Petra Perner (Ed.)

Poster Proceedings

IBaI CD-Report ISSN 1617-2671

Published July, 2006

www.data-mining-forum.de

Organized by:

ibai Institute of Computer Vision and Applied Computer Sciences, Leipzig/Germany
Dr. Petra Perner

Preface

The Industrial Conference on Data Mining ICDM-Leipzig was the sixth event in a series of annual events which started in 2000. We are pleased to note that the topic data mining with special emphasis on real world applications has been adopted from so many researchers all over the world into their research work. We received 156 papers from 19 different countries.
The main topics are data mining in medicine and marketing, web mining, mining of images and signals, theoretical aspects of data mining, and aspects of data mining that bundles a series of different data mining applications such as intrusion detection, knowledge management, manufacturing process control, time-series mining and criminal investigations.
The program committee was working hard in order to select the best papers. The acceptance rate was 30%. All these selected papers are published in this proceeding volume as long papers up to 15 pages. Besides that we installed a forum where work in progress has been presented. These papers are collected in a special poster proceeding volume and show once more the potentials and interesting developments for data mining for different applications.
Three new workshops have been established in connection with ICDM: 1. Mass Data Analysis on Images and Signals, MDA 2006, 2. Data Mining for Life Sciences, DMLS 2006, and 3. Data Mining in Marketing, DMM 2006. These workshops are developing new topics for data mining under the aspect of the special application. We are pleased to see how many interesting developments are going on under these topics.
We would like to express our appreciation to the reviewers for their precise and highly professional works. We appreciate the help and understanding of the editorial staff at Springer and in particular Alfred Hofmann, who supported the publication of these proceedings in the LNAI series.

We wish to thank all speakers, participants, and industrial exhibitors who contributed to the success of the conference.

We are looking forward to welcoming you to ICDM 2007 (www.data-mining-forum.de) and to the new work you will present there.

July 2006-05-10 Petra Perner

Citation

Petra Perner (Ed.)
Industrial Conference on Data Mining, ICDM 2006, Poster Proceedings,
IBaI CD- Report, ISSN 1617-2671, July 2006

Example Paper:

I. Gurrutxaga, A. Ansuategi, O. Arbelaitz, J. M. Perez, J. Muguerza, and J. I. Martin. *Comparison of two strategies, CTC and CMM, to combine m classifiers in a single comprehensible one*. In Poster Proceedings: Petra Perner (Ed.), Industrial Conference on Data Mining, ICDM 2006
IBaI CD- Report, ISSN 1617-2671, July 2006, p. 1-13.

Table of contents

IV. Image Mining

Comparison of two strategies, CTC and CMM, to combine *m* classifiers in a single comprehensible one

Ibai Gurrutxaga, Ander Ansuategi, Olatz Arbelaitz, Jesús Mª Pérez, Javier Muguerza, José I. Martín

Dept. of Computer Architecture and Technology, University of the Basque Country
M. Lardizabal, 1, 20018 Donostia, Spain
ibai.gurrutxaga@ehu.es, aansuategui001@ikasle.ehu.es,
{olatz.arbelaitz,txus.perez,j.muguerza,j.martin}@ehu.es
http://www.sc.ehu.es/aldapa

Abstract. Accurate prediction is probably the most pursued objective when solving real problems with machine learning, but there are situations where, added to the prediction, it is important to obtain a comprehensible output. The aim of this work is to compare the behaviour of two strategies to combine the knowledge of *m* classifiers in a single one in order to maintain the explaining capacity of the final classier: Consolidated Tree's Construction (CTC) algorithm and Combined Multiple Models (CMM) algorithm. The comparison is done from three points of view: accuracy, complexity and stability in explanation. Experimental results show that the use of CTC would be more recommendable than the use of CMM because, even if from the accuracy point of view the behaviour of CTC and CMM is similar, CT trees will give a more comprehensible (59.2% simpler) and steadier explanation (structure 50% steadier) than CMM classifiers.

1 Introduction

The aim of machine learning techniques when used to solve real world problems is to automate knowledge acquisition for performing useful tasks. Accurate prediction (error or guess) is probably the most pursued objective, but there are situations where, added to the prediction, it is of great value to obtain a comprehensible output. This way, the decision made by the system in an automatic way can be well-grounded. There are real domains such as illness diagnosis, fraud detection in different fields, marketing, etc., where the users of the machine learning algorithm wish to gain insight into the domain rather than obtain the right classification [5]. To solve this kind of problems, the learner's output needs to be comprehensible. In other situations where comprehensibility is not necessary, it will also be an important advantage for classifiers because it will help in processes of interactive refinement.

Related to the previous requirements, classifiers can be divided in two groups. There are classifiers with no comprehensible output such as artificial neural networks, support vector machines, multiple classifiers, etc., that, due to their complexity do not provide an explanation to the classification. And the second group, classifiers with

comprehensible output that focus in representations, such as decision trees and rule sets, and are more comprehensible than those that can usually be found in statistics or pattern recognition.

This focus on representation makes often models very dependent on the training data so that the built learners are unsteady or unstable: classifiers induced from slightly different subsamples of the same data set are very different in accuracy and structure [6].

The stability of the given explanation is important when the classifier gives an explanation to the classification made. As Turney found working on industrial applications of decision tree learning, not only to give an explanation but the stability of that explanation is of capital importance: "the engineers are disturbed when different batches of data from the same process result in radically different decision trees. The engineers lose confidence in the decision trees even when we can demonstrate that the trees have high predictive accuracy" [14].

Decision trees have been chosen as paradigm with comprehensive output in this work. Since in a decision tree the explanation is given by its structure, if we want to obtain a convincing explanation we need a way to build structurally steady classifiers (physical stability or structural stability) with small complexity (being the trees less complex, with smaller amount of nodes, will make the associated explanation simpler and as a consequence more comprehensible).

In paradigms such as bagging and boosting [1][2][4][7][13] this problem is not solved: some classifiers, normally weak classifiers such as classification trees, are combined to make a decision on the whole in order to reduce the error rate, but the solution is based on multiple trees, and, as a consequence, comprehensibility disappears. Domingos explained it very clearly in [5]: "while a single decision tree can easily be understood by a human as long as it is not too large, fifty such trees, even if individually simple, exceed the capacity of even the most patient".

We have developed a methodology for building classification trees based on several subsamples, Consolidated Trees' Construction Algorithm (CTC), which is less sensitive to changes in the training set from a structural point of view. Therefore the classification is contributed with a more steady explanation.

The aim of this work is to compare the behaviour of CTC algorithm with the Combined Multiple Models (CMM) algorithm proposed by Domingos [5] from three points of view: accuracy, complexity of the built classifiers and stability in explanation, that is to say, physical or structural stability.

The paper proceeds describing the algorithms we are going to compare, CTC and CMM algorithms, in Section 2. In Section 3 we describe the measures used to quantify stability and simplicity in explanation. Details about the experimental methodology are described in Section 4. In Section 5 we present an analysis of the experimental results: comparison in accuracy, complexity and structural stability of CTC and CMM algorithms. Finally Section 6 is devoted to show the conclusions and further work.

2 Description of the compared proposals

The proposals we are going to compare in this work, CTC and CMM, share the same idea: the accuracy and stability of a learner that produces a comprehensible representation can probably be improved by applying the learner $m>>1$ times. CTC and CMM propose different strategies to combine the knowledge of the m classifiers in a single one in order to maintain the explaining capacity of the final classier. Proposals such as bagging or boosting share the first idea but forget the importance of the comprehension in classification.

Both ideas could be applied to any classifier with comprehensible output but in this work classification trees have been selected as base classifiers. The algorithms will be described with this assumption but it would be easy to replace classification trees by any other learner.

2.1 CMM Algorithm

CMM proposes to recover the comprehensibility loss in multiple classifiers using the learning algorithm to model the data partitioning produced by them. The learning is done from randomly generated examples that are classified using the combined models. The new model will be comprehensible.

Algorithm 1. CMM Algorithm.

Inputs:
S training set
bagging procedure for combining models
N_S (*Number_Samples*) number of component models to generate
n number of new examples to generate

Procedure CMM (*S, C4.5, bagging, N_S, n*)

for i := 1 to N_S
Let S^i be a bootstrap sample of S
Let M^i be the model produced by applying *C4.5* to S^i
end for

for j := 1 to n
Let x be the randomly generated example
Let c be the class assigned to x by $bagging_{M1,\ldots,MN\_S(x)}$
Let $S = S \cup \{(x,c)\}$
end for

Let M be the model produced by applying *C4.5* to S

CMM is a general algorithm that can be used with different learners and combining models but in this work we will use (and describe) Domingos' implementation [5]. This way, the selected learning algorithm has been classification trees, specifically C4.5 release 8 of Quinlan [12], and, on the other hand, Domingos selected bagging to combine classifiers. As a consequence the base classifiers, C4.5 trees, will be built based on bootstrap subsamples and the resulting models will be aggregated by uniform voting [2]. The knowledge of this multi-classifier will be transmitted to CMM using it to artificially generate and label the examples that will be used to build it.

Algorithm 1 shows Domingo's CMM proposal adapted to the concrete implementation. As described in the algorithm *N_S* bootstrap samples are extracted from *S*, the original training set, and one C4.5 tree is built from each of them. *n* new examples are generated using the probability distribution implicit in the generated C4.5 trees (*n/Number_Samples* examples from each component C4.5 tree). The number of examples generated based on each individual branch of the C4.5 trees depends on the proportion of examples covered by it in the bootstrap sample. For each example this is done by ensuring that it satisfies the preconditions of the branches, and beyond that, by setting the values of its attributes according to a uniform distribution and generating missing values for each attribute in similar proportions to those found in the original training set, see [5]. The corresponding class (*c*) is assigned to each example based on the class the bagging of all the generated C4.5 trees assigns them ($c = bagging_{M1,\ldots,MN\_S}(x)$). This way, the examples will be representative of the combination of basic classifiers.

The CMM classifier will be the C4.5 tree built from the new sample obtained adding the *n* randomly generated examples to the original training set.

2.2 CTC Algorithm

CTC algorithm uses classifiers induced from several subsamples to build a single tree [11]. The consensus is achieved at each step of the tree's building process and only one tree is built. The different subsamples are used to make proposals about the feature that should be used to split in the current node. In order to make the CTC comparable to Domingo's CMM classifiers, the split function used in each subsample is the gain ratio criterion (the same used by Quinlan in C4.5 [12]). The decision about which feature will be used to make the split in a node of the Consolidated Tree (CT) is accorded among the different proposals by a not weighted voting process node by node. Based on this decision, all the subsamples are divided using the same feature. The iterative process is described in Algorithm 2.

The algorithm starts extracting a set of subsamples (*Number_Samples*) from the original training set. The subsamples are obtained based on the desired resampling technique (*Resampling_Mode*). For example, the class distribution of the original training set can be changed or not, examples can be extracted with or without replacement, different subsample sizes can be chosen, etc.

Algorithm 2. CTC Algorithm.

Generate N_S (*Number_Samples*) subsamples (S^i) from S with *Resampling_Mode* method
CurrentNode := *RootNode*
for i := 1 to N_S
$LS^i := \{S^i\}$
end for
repeat
for i := 1 to N_S
$CurrentS^i := First(LS^i)$
$LS^i := LS^i - CurrentS^i$
Induce the best split $(X,B)^i$ for $CurrentS^i$
end for
Obtain the consolidated pair (X_c,B_c) based on $(X,B)^I$, $1 \le i \le N\_S$
if $(X_c,B_c) \neq Not\_Split$
Split *CurrentNode* based on (X_c,B_c)
for i := 1 to N_S
Divide $CurrentS^i$ based on (X_c,B_c) to obtain n subsamples $\{S_1^i, ..., S_n^i\}$
$LS^i := \{S_1^i, ..., S_n^i\} \cup LS^i$
end for
else consolidate *CurrentNode* as a leaf
end if
CurrentNode := *NextNode*
until $\forall i$, LS^i is empty

In general, decision tree's construction algorithms divide the initial sample in several data partitions. In our algorithm, LS^i contains all the data partitions created from each subsample S^i. When the process starts, the only existing partitions are the initial subsamples.

The pair $(X,B)^i$ is the split proposal for the first data partition in LS^i. X is the feature selected to split and B indicates the proposed branches or criteria to divide the data in the current node. In the consolidation step, X_c and B_c are the feature and branches obtained by a voting process among all the proposals. In the different steps of the algorithm, the default parameters of C4.5 have been used as far as possible so that the made experimentation is as close to Domingo's as possible.

The process is repeated while LS^i is not empty. The Consolidated Tree's generation process finishes when, in the last subsample in all the partitions in LS^i, most of the proposals are not to split it, so, to become a leaf node. When a node is consolidated as a leaf node, the a posteriori probabilities associated to it are calculated averaging the a posteriori obtained from the data partitions related to that node in all the subsamples. Once the consolidated tree has been built it works the same way a decision tree does.

Previous works [8], [11] show that CT trees have larger discriminating capacity and structural stability than C4.5. In the experimentations made when subsamples with the original class distribution have been used, best results have been obtained for stratified

samples without replacement and of 75% of the training set (instance of *Resampling_Mode* parameter) [8]. This will be the selected option for this work.

3 Structural measures

CTC and CMM algorithms have been compared from three points of view: error, complexity, and structural stability. We need to define what we understand for complexity and structural stability.

Even if usually the complexity of classification trees is measured as the number of leaf nodes, we have used a related variant in our experimentation. The complexity has been measured as the number of internal nodes of the tree. There are two main reasons: explanation is related to the internal nodes of the trees, and this is an adequate measure to normalise the *Common* measure we will use to analyse the stability.

In order to evaluate the structural stability, a structural distance among the trees that are being compared has been defined: *Common*. This structural measure is based on a pair to pair comparison, *Similarity,* among all the trees of the set. This function (*Similarity*) counts the common nodes among two trees. It is calculated starting from the root and covering the tree level by level. If two nodes coincide in the feature used to make the split, the proposed branches or stratification and the position in the tree, they will be counted as common nodes. When a different node is found, the subtree under that node is considered as different. For a set of trees T_{set}, with m trees, the *Common* value is calculated as the average value of all the possible pair to pair comparisons (Equation 1):

$$Common(T_{set}) = \frac{2}{m(m-1)} \sum_{\substack{k,l=0 \\ k<l}}^{m-1} Similarity(T_k, T_l) \quad (1)$$

In order to take into account the parsimony principle we have normalised the *Common* value in respect to the complexity. We will denominate this measure *%Common* and it will quantify the identical fraction of two or more trees.

4 Experimental Methodology

Eleven databases of real applications from the well known UCI Repository benchmark [10] have been used for the experimentation. For the *KDDcup99* database, used to train Intrusion Detection Systems, we have not used the original data which has 5 million examples and 23 classes. In order to reduce the cost of the experimentation, we have used a stratified sample of 4,941 examples where the number of classes has been reduced to two (attack / not attack). Table 1 shows the wide range of characteristics of the used domains (extreme values are in italics and underlined) : the number of patterns (*N. of patterns*) goes from 148 to 4,941; the number of features (*N.*

of features) from 4 to 41; and the number of classes of the dependent variable (*N. of classes*) from 2 to 15.

Table 1. Description of experimental domains.

Domain	*N. of patterns*	*N. of features*	*N. of classes*
Breast-W	699	10	***2***
Iris	150	***4***	3
Heart-C	303	13	2
Glass	214	9	7
Segment	2310	19	7
Voting	435	16	2
Lymph	***148***	18	4
Hepatitis	155	19	2
Hypo	3163	25	2
Soybean-L	290	35	***15***
KDDcup99	***4941***	***41***	2

The validation methodology used in this experimentation has been to execute 5 times a 10-fold stratified cross validation [9]. In each of the folds of the cross-validation 400 subsamples have been extracted:

- 200 stratified subsamples of 75% of the training sample in the corresponding fold without replacement (T=75%).
- 200 bootstrap samples obtained with replacement and identical size to the training sample in the corresponding fold (Boot).

The first set of subsamples has been used to build CT trees and the second one for CMM classifiers. Thinking that the goodness of the algorithm could be due to the kind of used subsamples, we have also tried to build CMM classifiers with subsamples of 75% of the original training set, and CTC classifiers with bootstrap samples but in both cases results were worse than the ones we will present here.

Figure 1 shows a schema of the generation of subsamples.

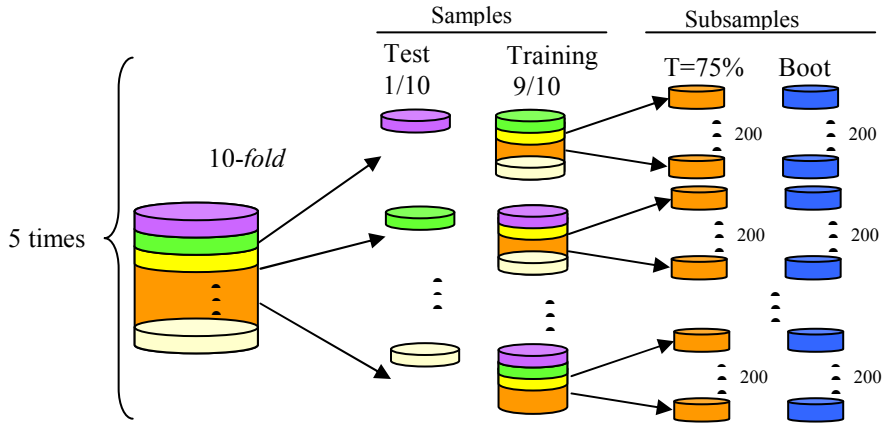


Figure 1. Schema of subsample generation to build CT trees (T=75%) and CMM classifiers (Boot).

The 200 subsamples have been used to explore in both algorithms 12 values for *Number_Samples* parameter (*N_S*): 3, 5, 10, 20, 30, 40, 50, 75, 100, 125, 150 and 200. In each one of the 50 folds of the cross validations 12 CT trees and 12 CMM classifiers have been built (each one with different number of samples or base classifiers). As a consequence, approximately 6,600 CT trees and 6,600 CMM classifiers have been built for the experimentation presented in this paper.

Domingos in his experimentation analysed the accuracy of CMM classifiers using 25 C4.5 trees, but we wanted to expand this experimentation to larger amount of subsamples in order to analyse the effect of this parameter for the two algorithms (CTC and CMM), and two points of view: accuracy and quality of the explanation.

The CTC methodology has been compared to the CMM algorithm using as base classifier in both of them the C4.5 Release 8 of Quinlan, with default parameter settings. In both algorithms trees have been pruned, using the error based pruning implemented in C4.5 R8 software, to situate both systems in a similar zone in the learning curve [9]. We can not forget that developing too much a classification tree leads to a greater probability of overtraining and to a less comprehensive output.

For building CMM classifiers the number of randomly generated examples (*n*) needs to be fixed. Taking into account the process used to generate examples (*n/Number_Samples* examples are generated from each component C4.5 tree) and that the number of component C4.5 trees goes from 5 to 200, this number needs to be large enough to generate a minimum set of examples from each one of the C4.5 trees and, as the original sample is added to these examples to build the CMM, it also needs not to be too small compared to it. Domingos generated 1,000 artificial examples but the sizes of the databases used for the experimentation where smaller than the sizes of the databases used in our experimentation. As a consequence, the number has been fixed to max (1,000; (*NPF* * 1.5)) being *NPF* the number of patterns of the training set in each fold.

5 Experimental results

CTC and CMM algorithms have been compared from three points of view: error, complexity, and structural stability (measured based on *%Common*).

From a practical point of view, the complexity quantifies how simple the given explanation is, *Common* quantifies structural stability of the classification algorithm, whereas the error would quantify the "quality" of the explanation given by the tree. Evidently an improvement in comprehensibility must be supported with a reasonable error rate. Our main goal has been to decrease complexity and increase stability with no loss in accuracy.

We will start the comparison of CTC and CMM algorithms from the accuracy point of view because if their behaviour is very different the rest of the comparison might not be worth. We show in Table 2 average (5 runs * 10 folds) error results for CMM algorithm in each one of the databases, and in Table 3 the same kind of results but for CTC algorithm. We have marked in bold in both tables the best value for each database and value of *N_S* parameter.

Table 2. Error values for CMM algorithm in 10 databases and different values of *N_S* parameter.

	CMM											
N_S	03	05	10	20	30	40	50	75	100	125	150	200
Breast-W	6.06	5.75	**5.60**	**5.44**	**5.41**	**5.26**	**5.44**	**5.29**	**5.23**	**5.26**	**5.41**	**5.38**
Iris	5.75	**6.41**	5.88	6.01	5.61	5.88	5.48	5.48	5.34	5.34	5.48	5.61
Heart-C	25.47	**24.03**	**22.90**	**22.77**	**22.22**	**23.10**	**22.80**	**22.71**	**22.85**	**22.49**	23.58	24.03
Glass	31.19	30.37	30.58	30.36	**29.00**	**29.04**	**29.10**	29.68	**28.16**	**27.92**	**28.74**	**29.17**
Segment	3.64	3.59	**3.24**	**3.51**	**3.28**	**2.99**	**3.30**	**3.27**	**3.27**	**3.06**	**3.21**	**3.35**
Voting	3.96	3.87	3.78	3.60	3.50	3.64	3.60	3.78	3.74	3.60	3.69	3.51
Lymph	**20.84**	**21.10**	21.05	20.51	20.66	20.37	21.14	20.73	**19.95**	20.99	20.78	20.77
Hepatitis	20.01	**19.02**	**19.43**	**19.56**	**19.82**	**19.45**	**18.78**	**19.08**	**19.95**	**18.72**	**18.51**	**20.17**
Hypo	0.80	0.78	**0.74**	0.75	0.73	0.75	**0.73**	0.75	0.75	**0.73**	0.74	0.74
Soybean-L	13.98	12.94	12.67	11.64	12.33	12.18	11.43	11.98	11.57	11.71	11.29	11.64
KDDcup99	0,50	0,50	0,57	0,56	0,50	0,56	0,54	0,48	0,52	0,50	0,45	0,50

Table 3. Error values for CTC algorithm in 10 databases and different values of *N_S* parameter.

	CTC											
N_S	03	05	10	20	30	40	50	75	100	125	150	200
Breast-W	**5.77**	**5.66**	5.63	5.52	5.49	5.60	5.63	5.60	5.58	5.60	5.60	5.58
Iris	**5.61**	5.75	**5.48**	**4.68**	**4.54**	**4.41**	**4.28**	**4.14**	**4.28**	**4.14**	**4.01**	**4.28**
Heart-C	**25.41**	24.49	23.77	22.85	22.92	24.02	23.75	23.55	23.42	23.36	**23.16**	**23.48**
Glass	**29.11**	**29.21**	**30.15**	**29.94**	30.22	30.13	**29.61**	29.43	29.85	30.16	30.06	30.00
Segment	**3.52**	**3.31**	3.54	3.56	3.54	3.56	3.59	3.33	3.44	3.50	3.49	3.57
Voting	**3.32**	**3.41**	**3.50**	**3.36**	**3.41**	**3.41**	**3.41**	**3.36**	**3.36**	**3.36**	**3.36**	**3.36**
Lymph	21.34	21.13	**19.67**	**19.91**	**19.65**	**19.93**	**20.06**	**20.18**	**20.18**	**20.19**	**20.31**	**20.19**
Hepatitis	**19.49**	21.55	20.99	21.11	20.97	21.08	21.25	20.31	20.58	21.25	20.95	20.70
Hypo	**0.73**	**0.73**	**0.74**	**0.72**	**0.72**	**0.72**	**0.73**	**0.73**	**0.74**	**0.74**	**0.73**	**0.73**
Soybean-L	**12.40**	**11.84**	**11.84**	**11.29**	**11.71**	**11.30**	**10.88**	**10.67**	**10.67**	**10.68**	**10.88**	**10.74**
KDDcup99	**0,47**	**0,42**	**0,42**	**0,44**	**0,43**	**0,48**	**0,49**	**0,43**	**0,44**	**0,47**	**0,45**	**0,46**

If we compare the results of both tables we can observe that in general the obtained accuracy is very similar in both systems. There are not large differences in the averages for none of the 120 comparisons (80 times out of 132 CTC behaves better, and the rest of the cases, CMM behaves better) and the minimum values are obtained for half of the databases with CTC algorithm and for the other half with CMM. Analysis for finding statistically significant differences (paired t-test [3]), with 95% confidence level, has been done, but not differences have been found. The average behaviour for the 10 databases can be observed in Figure 2. As Domingos stated in his work these error rates are situated among the error rates achieved with C4.5 algorithm and the ones achieved with bagging.

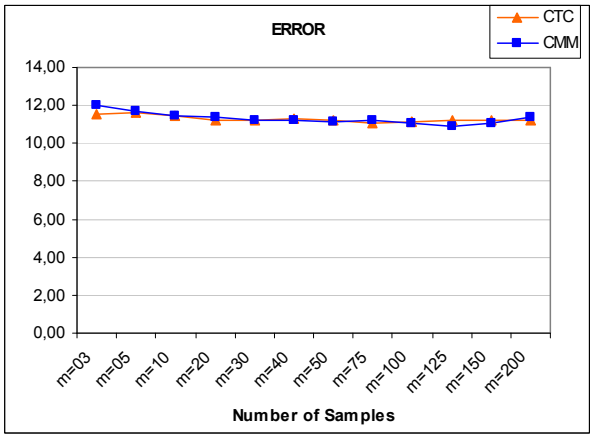


Figure 2. Average error values for CTC and CMM when *N_S* is varied (11 databases, 5 runs, 10 folds).

From the results in Table 2, Table 3 and Figure 2 we can state that from the accuracy point of view the behaviour of both algorithms, CTC and CMM, is similar, so, the convenience of using one or the other will depend on their explaining capacity.

As we mentioned in the introduction we will measure the comprehensibility of the classifier based in two parameters:

- Complexity, because when smaller the tree's complexity is more comprehensible the structure is.
- Structural stability because important changes in the structure of the obtained trees produce loss of confidence in the final users.

Figure 3 shows average complexity values (11 databases * 5 runs * 10 folds) when *N_S* parameter is varied for both algorithms, CTC and CMM. The average complexity for CTC algorithm is 13.86. This means that the explanation given to the classification made will be based more or less in 13 variables. On the other hand, average complexity for CMM classifiers is 33.94 and this means that in average the explanation given by a CMM classifier will be based on 34 variables. Evidently CT trees will give simpler and as a consequence more comprehensible explanation than CMM classifiers. We could say this explanation is 59.2 % simpler.

The stability of the explanation given to the classification is also important. We have measured the stability of the explanation by measuring the common structure of the built classifiers. Figure 4 shows the average values (11 databases * 5 runs * 10 folds) of %*Common* (percentage of common structure) when *N_S* parameter is varied for both algorithms, CTC and CMM.

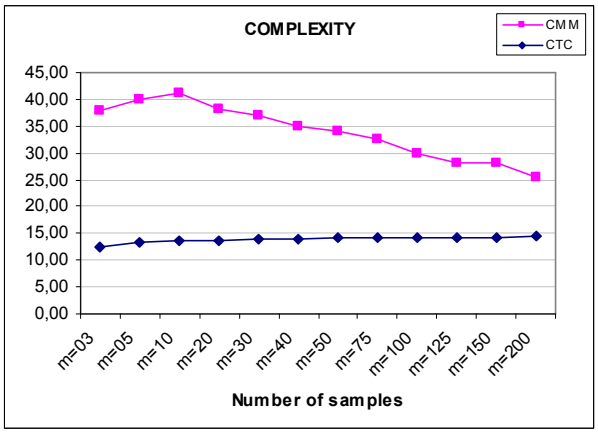


Figure 3. Average complexity values for CTC and CMM when *N_S* is varied (11 databases, 5 runs, 10 folds).

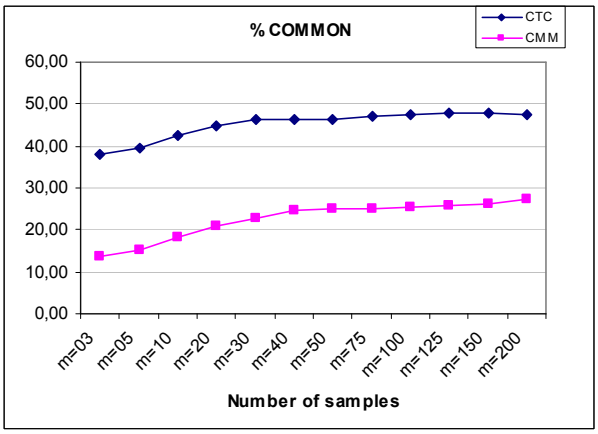


Figure 4. Average %*Common* values for CTC and CMM when *N_S* is varied (11 databases, 5 runs, 10 folds).

The curves in Figure 4 show that the common structure of CT trees is larger than the common structure of CMM classifiers. CT trees share in average 45.10% of their structure, whereas CMM classifiers share only 22.54%. Based on these measurements we could say that the explanation given by Consolidated Trees is nearly 50% more steady that the one given by CMM classifiers. As a consequence, the customer or the user of that tool will feel more confident when using CTC algorithm than when using CMM.

Related to the number of samples or number of base classifiers (*N_S*) used to build CT trees and CMM classifier, results confirm, in general, the outcomes of Domingos' work. From the accuracy point of view, no significant improvements are obtained in none of the algorithms when the number of used subsamples is larger than 30. From

the comprehensibility point of view, we could say that the behaviour of CTC algorithm is stabilized for N_S = 30 or larger, whereas for CMM algorithm the stability is smaller because complexity decreases when N_S is increased. As we concluded in previous works [8][11], between 30 and 50 samples would be a good trade off among quality of results and computational cost for CTC algorithm, and also for CMM.

6 Conclusions and Further Work

Being aware of the importance of classifiers to be comprehensible when using machine learning to solve real world problems, we have compared in this work the behaviour of CTC algorithm with the Combined Multiple Models (CMM) algorithm proposed by Domingos [5] from three points of view: accuracy, complexity of the built classifiers and stability in explanation, that is to say, physical or structural stability.

From the experimentalresults we can conclude that i t is recommendable the use of CTC rather than the use of CMM because, even if from the accuracy point of view, the behaviour of both algorithms, CTC and CMM, is similar, after analysing the complexity of both algorithms, we can say that CT trees will give simpler and as a consequence more comprehensible explanation than CMM classifiers. We could say this explanation is 59.2 % simpler. And besides, looking to how steady the structure of the built trees is maintained, we could say that the explanation given by Consolidated Trees is nearly 50% more steady that the one given by CMM classifiers. As a consequence, the customer or the user of that tool will feel more confident using CTC algorithm than using CMM.

Related to the number of samples or number of base classifiers used to build CT trees and CMM classifier, results confirm that between 30 and 50 samples would be a good trade off among quality of results and computational cost for CTC algorithm, and also for CMM.

There are many things that can be done in the future related to this work. Firstly the experimentation can be extended to more databases. On the other hand, the effect of the subsampling inside a fold could also be studied, that is to say, how the use of a set or another of subsamples of the same fold affects to CTC and CMM. Related to the measure of stability in explanation, other structural measures can be tried, as for example, taking into account that even if two variables appear in different order in the structure of the trees, many times their explanation does not change. Finally, both algorithms could be used with other classification paradigms with explaining capacity such as induction rules.

Acknowledgments

The work described in this paper was partly done under the University of Basque Country (UPV/EHU) project: 1/UPV 00139.226-T-15920/2004. It was also funded by the Diputación Foral de Gipuzkoa and the European Union.

The *lymphography* domain was obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. Thanks go to M. Zwitter and M. Soklic for providing the data.

References

1 Bauer E., Kohavi R.: An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants, Machine Learning, Vol. 36, (1999) 105-139.
2. Breiman L.: Bagging Predictors. Machine Learning, Vol. 24, (1996) 123-140.
3. Dietterich T.G.: Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms, Neural Computation, Vol. 10, No. 7, (1998) 1895-1924.
4. Dietterich T.G.: An Experimental Comparison of Three Methods for Constructing Ense mbles of Decision Trees: Bagging, Boosting, and Randomization, Machine Learning, Vol. 40, (2000) 139-157.
5. Domingos P.: Knowledge acquisition from examples via multiple models. Proc. 14th International Conference on Machine Learning Nashville, TN (1997) 98-106.
6. Drummond C., Holte R.C.: Exploiting the Cost (In)sensitivity of Decision Tree Splitting Criteria, Proceedings of the 17th International Conference on Machine Learning, (2000) 239-246.
7. Freund, Y., Schapire, R. E.: Experiments with a New Boosting Algorithm, Proceedings of the 13th International Conference on Machine Learning, (1996) 148-156.
8 Gurrutxaga I., Arbelaitz O., Pérez J.M., Martín J.I., Muguerza J.: The effect of the used resampling technique and number of samples in consolidated trees' construction algorithm. Accepted for presentation in IADIS Applied Computing 2006.
9. Hastie T., Tibshirani R., Friedman J.: The Elements of Statistical Learning. Springer-Verlang (es). ISBN: 0-387-95284-5, (2001).
10 Newman, D.J. & Hettich, S. & Blake, C.L. & Merz, C.J.: UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, (1998).
11 Pérez J.M., Muguerza J., Arbelaitz O., Gurrutxaga I., Martín J.I.: Consolidated Trees: an Analysis of Structural Convergence". Lecture Notes in Artificial Intelligence 3755, Data Mining: Theory, Methodology, Techniques, and Applications. Springer-Verlag. Graham, J. W. and Simeon J. S. (Eds.), (2006), 39-52.
12. Quinlan J.R.: C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc.(eds), San Mateo, California (1993).
13. Skurichina M., Kuncheva L.I., Duin R.P.W. Bagging and Boosting for the Nearest Mean Classifier: Effects of Sample Size on Diversity and Accuracy, LNCS Vol. 2364. Multiple Classifier Systems: Proc. 3th Inter. Workshop, MCS , Cagliari, Italy, (2002) 62-71.
14. Turney P. Bias and the quantification of stability. Machine Learning, 20 (1995), 23-33.

Adaptive Weights Calculation Procedure for Weighted Voting– Idea and Experimental Results

Michal Wozniak

Chair of Systems and Computer Networks, Wroclaw University of Technology
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
e-mail: michal.wozniak@pwr.wroc.pl

Abstract. The *Multiple Classifier Systems* are nowadays one of the most promising directions in pattern recognition. There are many methods of decision making by the group of classifiers. The most popular are methods that have their origin in vote methods, where the decision of the common classifier is a combination of single classifiers decisions. This work presents original method of weighted classifiers combination and experimental results of proposed algorithms from computer generated data.

1 Introduction

The concept of the *Multiple Classifier Systems* (MCS) is not new and it is known for over 15 years [1]. Some works in this field were published as early as the '60 of the XX century [2], when it was shown that the common decision of independent classifiers is optimal, when chosen weights are inversely proportional to errors made by classifiers. In many review articles this trend is mentioned as one of the most promising in field of the pattern recognition [3]. In the beginning in literature one could find only the majority vote, but in later works more advanced methods of receiving common answer of the classifier group were proposed. Attempts to estimate classification quality by the classifier committee are one of essential trends. Known in literature conclusions, derived on the analytic way, concern particular case of the majority vote [4], when classifier committee is formed from independent classifiers. Unfortunately this case has only theoretical character and is not useful in practice. On the other hand the weighted vote is taken into consideration [5]. In this work it was pointed that the optimal weight value should be dependent on the error of the single classifier and on the *prior* probability of the class, on which classifier points. One also has to mention many other works, that describe analytical properties and experimental results, like [6-8].
Paper presents new method of weighted voting procedure based on errors of simple classifiers. Proposed method has iteration character. Its' quality is compared to another voting concepts.

The content of the work is as follows: next section shortly introduces into necessary background. Section 3 proposes how calculate weights of classifiers by

original adaptive procedure. In section 4 the results of experimental investigation of methods under consideration are presented. The last section concluded the paper.

2 Problem statement

Let's assume that we have n classifiers $\Psi^{(1)}, \Psi^{(2)}, ..., \Psi^{(n)}$. Each of them decides if object belongs to $i \in \boldsymbol{M} = \{1, ..., M\}$. For making common decision by the group of classifiers $\Psi^{(1)}, \Psi^{(2)}, ..., \Psi^{(n)}$ let's use following common classifier $\overline{\Psi}$:

$$\overline{\Psi} = \arg\max_{j \in \boldsymbol{M}} \sum_{i=1}^{k} \delta(j, \Psi_i) w_i \Psi_i \,, \tag{1}$$

where w_i is the weight of ith classifier and

$$\sum_{i=1}^{n} w_i = 1 \tag{2}$$

and

$$\delta(i, j) = \begin{cases} 0 & \text{if} \quad i \neq j \\ 1 & \text{if} \quad i = j \end{cases}. \tag{3}$$

Lets note that the weights can perform the role of the quality of classifier $\Psi^{(i)}$. The accuracy of classifier $\overline{\Psi}$ is maximized by assigning weights

$$w_i \propto \frac{P_{a,i}}{1 - P_{a,i}} \,, \tag{4}$$

Where accuracies $P_{a,i}$ denotes probability of accuracy of ith classifier. Unfortunately it is not sufficient for guarantying the smallest classification error. The *prior* probabilities for each class have to be taken into account also[5]. In real decision problems the values of the *prior* probabilities are usually unknown. From this reason we propose the heuristic algorithm of classifier weights calculation.

3 Adaptive Weights Calculation Algorithm

Let us present AWL algorithm (ang. *Adaptive Weights Calculation*). It has heuristic character and its idea is based on perceptron learning method. This adaptive procedure increases the weights of classifier which accuracy is higher than accuracy of common classifier. The preudocode of procedure is shown in fig.1.

Procedure AWC
Input: $\Psi_1, \Psi_2, ..., \Psi_k$ – set of classifiers for the same decision problem
Learning set *LS*
T number of iterations
1.. For each classifier Ψ_i

b) compute weights of classifier $w_i(0) = \frac{1}{k}$

c) estimate accuracy probability of $\Psi^{(i)} - \hat{P}_{a,i}$

2. For *t*:=1 do *T*

a) estimate accuracy probability of $\overline{\Psi} - \hat{P}_a(t)$

b) sum_of_weights:=0

c) for each classifiers Ψ_i

$$w_i(t) = w_i(t-1) + \left(\frac{1}{(t+\gamma)} \left(\hat{P}_a(t) - \hat{P}_{a,i} \right) \right)$$

if $w_i(k) > 0$
then sum_of_weights:= sum_of_weights+ $w_i(k)$
else $w_i(t)$:=0;

d) if sum_of_weights >0
then
i) for each classifier Ψ_i

$$w_i(t) := \frac{w_i(t)}{sum\_of\_weights}$$

else
i) $t := T$
ii) for each classifier Ψ_i
$w_i(t) = w_i(t-1)$

end.

Fig. 1. Pseudocode of AWC algorithm

The γ is integer, constant value fixed arbitrary (usually given by expert or obtained via experiments) which is connected with the speed of the weights learning process.

4 Experimental investigation

The aim of the experiment was to compare the errors of the weighted combined classifiers with the quality of single classifiers and with the quality of combined classifiers that in combining rules do not take into account the quality of single classifiers [9,10]. The following classifiers were chosen:

1. Majority voting – denoted as MV,
2. Weighted voting which based on the weight proposed in (4);– denoted as PV,
3. Adaptive weighted voting classifier which uses weights obtained via AWC procedure described in section 3 – denoted as AV,
4. Single classifier based on the learning sequence generated according to the chosen unperturbed probability distribution – denoted as P1,
5. Single classifier based on the learning sequence generated according to the chosen probability distribution, perturbed 10% of elements of the uniform distribution – denoted as P2,
6. Single classifier based on the learning sequence generated according to the chosen probability distribution, perturbed 20% of elements of the uniform distribution – denoted as P3,
7. Single classifier based on the learning sequence generated according to the chosen probability distribution, perturbed 30% of elements of the uniform distribution – denoted as P4,
8. Single classifier based on the learning sequence generated according to the chosen probability distribution, perturbed 40% of elements of the uniform distribution – denoted as P5.

The conditions of experiments are as follow:

1. For experiments we chose the following probability distributions for conditional probability distribution in each of class:
   1.1. Highleyman's distribution,
   1.2. Normal distribution,
   1.3. Banana distribution.
2. All experiments were carried out in Matlab environment using the PRtools toolbox [11] and own software [15].
3. Probabilities of errors of the classifiers were estimated using the 10-cross-validation method.
4. In each experiment we testes the dependencies between classifiers' qualities and size of learning set.
5. For all experiments two-class recognition task was considered with equal values of *prior* probabilities.
6. We used three classifiers – two of them based on the estimation of the estimation of the probability density[12,13]:
   5.1. k_n *-NN* estimator
   5.2. the Parzen estimator were used
   5.3. C4.5 algorithm[14].

The results of experimental investigations are presented in Fig.2-10.

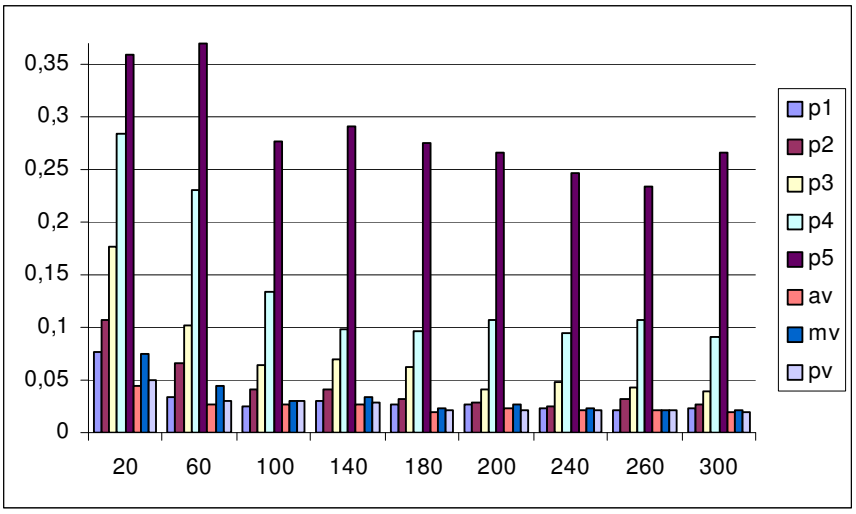


Fig. 2. Results of classification error [%] versus the number of learning sets for presented voting methods, *k*(*N*)-*NN* algorithm and banana distributions ($\gamma = 2$).

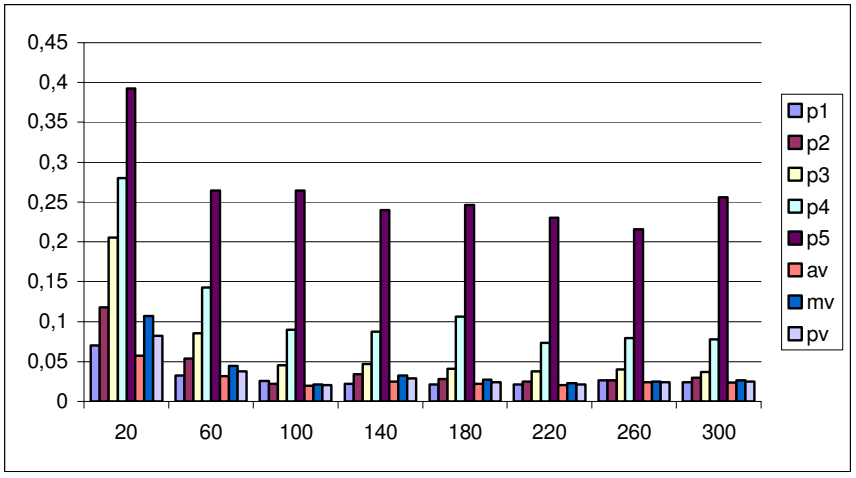


Fig. 3. Results of classification error [%] versus the number of learning sets for presented voting methods, Parzen algorithm and banana distributions ($\gamma = 0$).

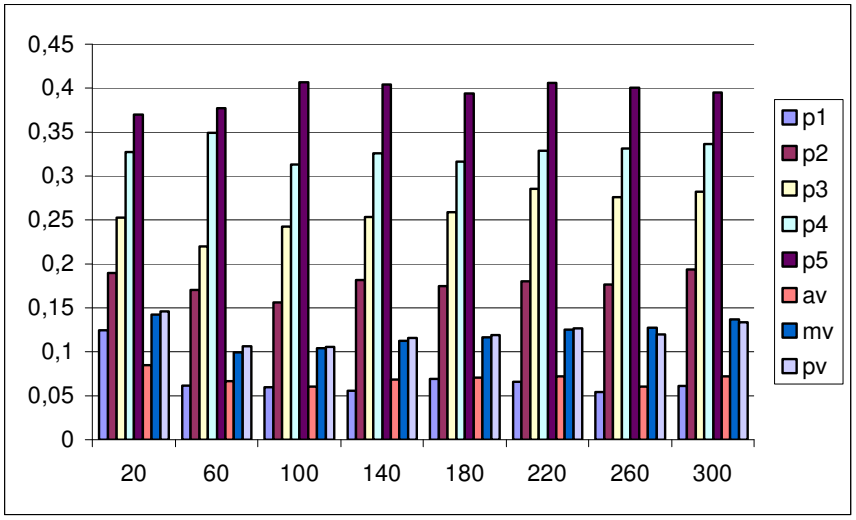


Fig. 4. Results of classification error [%] versus the number of learning sets for presented voting methods, *C4.5* algorithm and banana distributions ($\gamma = 10$).

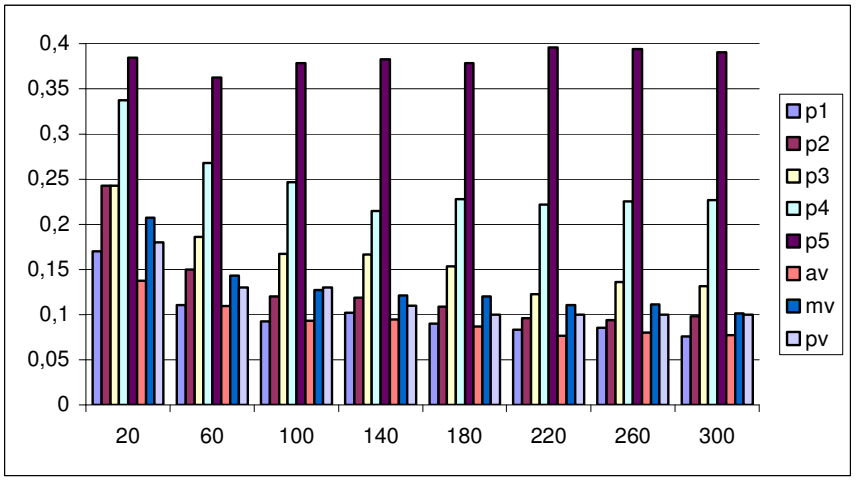


Fig. 5. Results of classification error [%] versus the number of learning sets for presented voting methods, *k*(*N*)-*NN* algorithm and Highleyman's distributions ($\gamma = 2$).

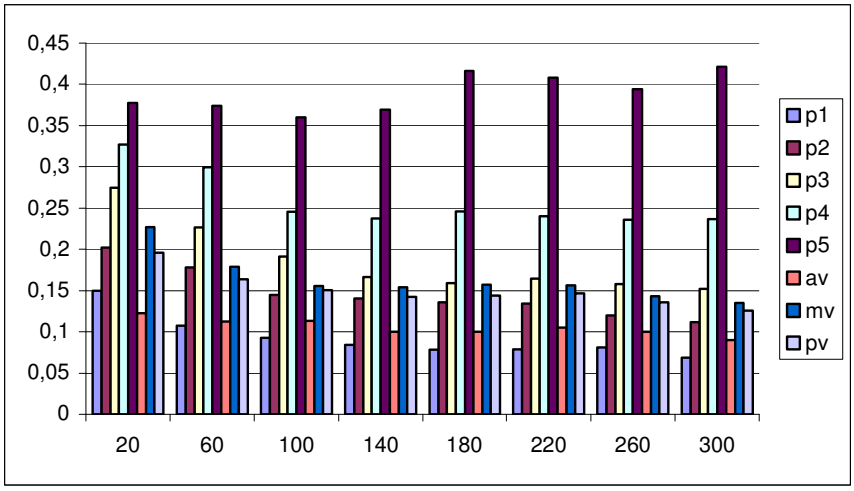


Fig. 6. Results of classification error [%] versus the number of learning sets for presented voting methods, *Parzen* algorithm and Highleyman's distributions ($\gamma = 3$).

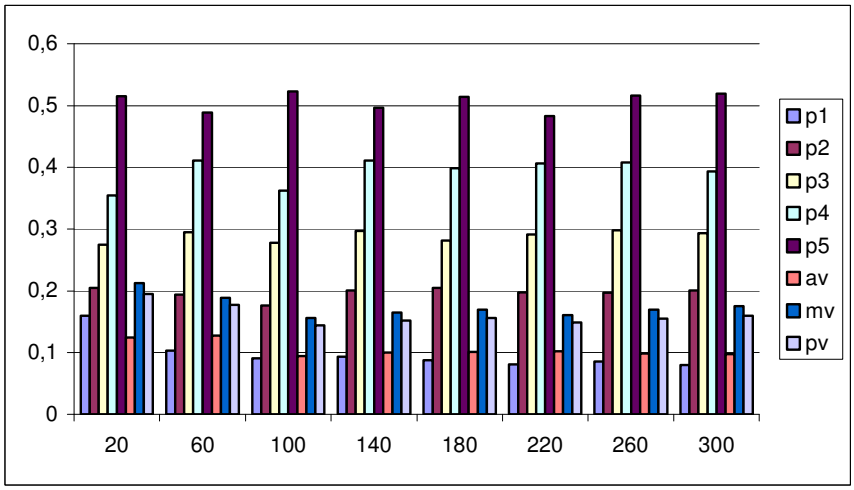


Fig. 7. Results of classification error [%] versus the number of learning sets for presented voting methods, *C4.5* algorithm and Highleyman's distributions ($\gamma = 10$).

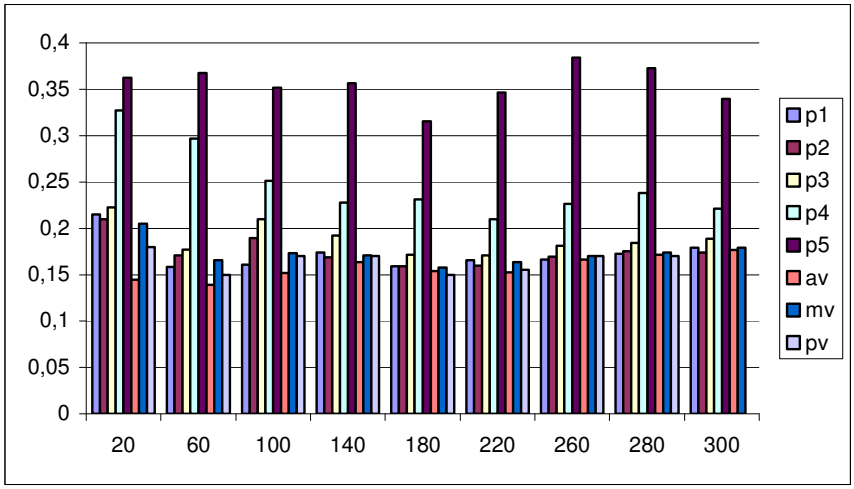


Fig. 8. Results of classification error [%] versus the number of learning sets for presented voting methods, *k*(*N*)-*NN* algorithm and Normal distributions ($\gamma = 1$).

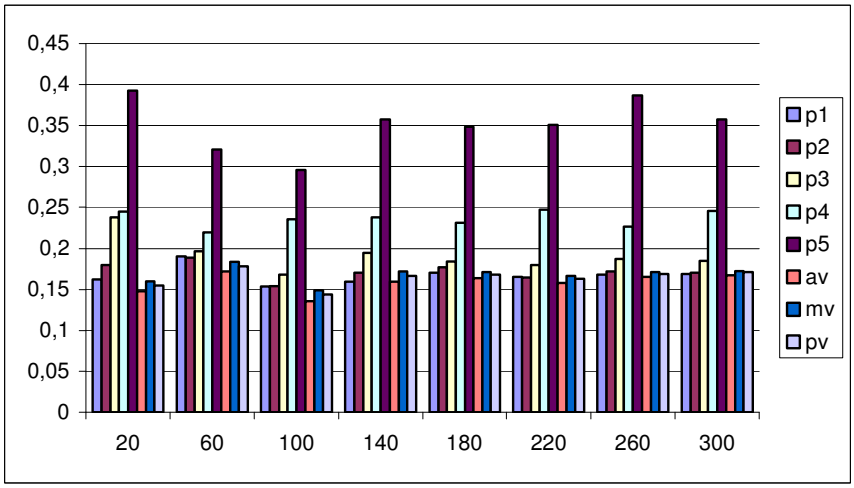


Fig. 9. Results of classification error [%] versus the number of learning sets for presented voting methods, *Parzen* algorithm and Normal distributions ($\gamma = 1$).

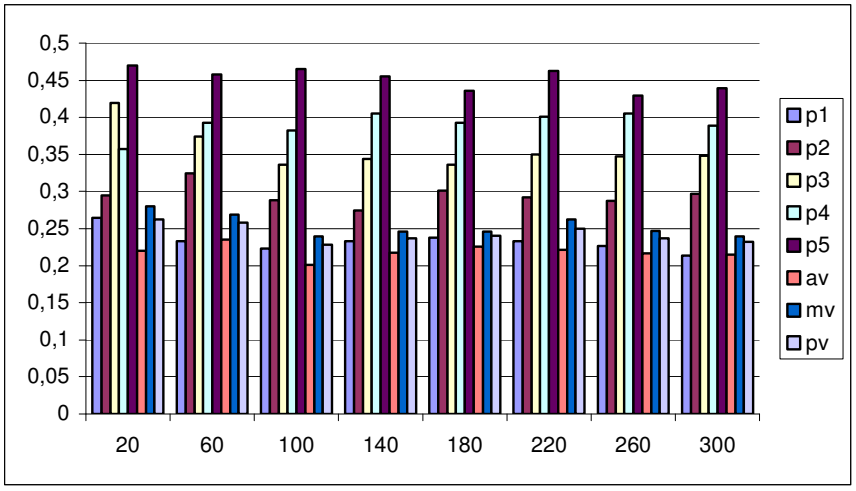


Fig. 10. Results of classification error [%] versus the number of learning sets for presented voting methods, *C4.5* algorithm and Normal distributions ($\gamma = 10$).

4.1 Experimental results evaluation

First one has to note that we are aware of the fact that the scope of computer experiments is limited. Therefore making general conclusions basing on them is very risky. In our opinion mentioned below statements should not be generalized at this point, but they should be confirmed by other experiments in much broader scope.
In the case of the presented experiment

1. combined algorithms gave results slightly worse than the best single classifier. It was also observed that classifiers, which for making decision used weighted answer of single classifiers (AV, PV), recognized objects definitely better than the rest of classifiers except the best single one;
2. adaptive weights calculation algorithm (AV) improve the quality of weighted voting (PV) which based on the weight proposed in (4);
3. in adaptive weights calculation procedure the values of each classifiers weights are probably closer their optimal value than weights proposed in (4);
4. the error of the best classifier is close the quality of combined classifiers but always slightly worse than proposed concept. We have to respect that for proposed experiment this classifier based on the learning set without noises, what we can find in real decision problem.

We were presenting only the piece of experimental results. During the experiments we observed that γ factor had been playing the important in the learning process. For presentation we chose only one value of γ with the best quality of classification.
We stated that the weights values established in the first 5-10 iterations. For some experiments we observed that after about 10 iterations classification error of combined classifier was increasing. In some cases the weights of the best single

classifier exceeded 0.5 what caused that quality of decision of committee of classifiers was the same as for the best classifier.
During learning we noticed that in the coincidence when the differences between the qualities of simple classifiers are quite big (how it took place e.g. in the case of the C4.5 algorithm), the AWD procedure assigned to the best classifier the weight bigger than 0.5. It means that decision of the group classifiers is *de facto* decision of the best classifier. It is not undesirable behavior because lets consider the following team of five independent classifiers $\Psi^{(1)}, \Psi^{(2)}, \Psi^{(3)}, \Psi^{(4)}, \Psi^{(5)}$ with accuracies $P_{a,1} = 0.55$, $P_{a,2} = 0.60$, $P_{a,3} = 0.65$, $P_{a,4} = 0.70$, $P_{a,5} = 0.80$. The probability of error of the committee of classifiers (voting according majority rule) $P_e^{\,5}$ is given by the following formula

$$
\begin{aligned}
P_e^{(5)} = {} & (1-P_{a,1})(1-P_{a,2})(1-P_{a,3})(1-P_{a,4})(1-P_{a,5}) + \\
& (1-P_{a,1})(1-P_{a,2})(1-P_{a,3})(1-P_{a,4})P_{a,5} + \\
& (1-P_{a,1})(1-P_{a,2})(1-P_{a,3})P_{a,4}(1-P_{a,5}) + \\
& (1-P_{a,1})(1-P_{a,2})P_{a,3}(1-P_{a,4})(1-P_{a,5}) + \\
& (1-P_{a,1})P_{a,2}(1-P_{a,3})(1-P_{a,4})(1-P_{a,5}) + \\
& P_{a,1}(1-P_{a,2})(1-P_{a,3})(1-P_{a,4})(1-P_{a,5}) + \\
& (1-P_{a,1})(1-P_{a,2})(1-P_{a,3})P_{a,4}P_{a,5} + \\
& (1-P_{a,1})(1-P_{a,2})P_{a,3}(1-P_{a,4})P_{a,5} + \\
& (1-P_{a,1})(1-P_{a,2})P_{a,3}P_{a,4}(1-P_{a,5}) + \\
& (1-P_{a,1})P_{a,2}(1-P_{a,3})(1-P_{a,4})P_{a,5} + \\
& (1-P_{a,1})P_{a,2}(1-P_{a,3})P_{a,4}(1-P_{a,5}) + \\
& (1-P_{a,1})P_{a,2}P_{a,3}(1-P_{a,4})(1-P_{a,5}) + \\
& P_{a,1}(1-P_{a,2})(1-P_{a,3})(1-P_{a,4})P_{a,5} + \\
& P_{a,1}(1-P_{a,2})(1-P_{a,3})P_{a,4}(1-P_{a,5}) + \\
& P_{a,1}(1-P_{a,2})P_{a,3}(1-P_{a,4})(1-P_{a,5}) + \\
& P_{a,1}P_{a,2}(1-P_{a,3})(1-P_{a,4})(1-P_{a,5}) = 0{,}20086 \quad .
\end{aligned}
\tag{5}
$$

As we see it will be better if we remove the worst classifier and reduce the ensemble to the single and the most accurate classifier $\Psi^{(5)}$ [5].

5 Final remarks

The original method of weights calculation for combined classifier was presented in this paper. This study presented the experimental quality evaluation of the combined classifiers recognition also. Obtained results seem to confirm the sense of using

combing methods. Unfortunately, as it was stated, it is not possible to determine their value in the analytical way. One should although hope that in case of application of above mentioned methods for real decision tasks, we have judgment of the expert, who can formulate the heuristic function of weights selection or be calculate by heuristic procedure like methods proposed in this paper. This function then can be verified and improved by the computer experiment.
We have to stress that proposed method suffers from overfitting. Therefore we have to be very careful when we fixed the number of procedure iterations.

Acknowledgement

This work is supported be The Polish State Committee for Scientific Research under the grant which is realizing in years 2006-2009.

Reference

1. Xu L., Krzyżak A., Suen Ch.Y., Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition, *IEEE Transactions on SMC*, l.22, no. 3, 1992, pp.418-435.
2. Chow C.K., Statistical independence and threshold functions, *IEEE Transaction on Electronic Computers*, EC-16, 1965, pp.66-68.
3. Jain A.K., Duin P.W., Mao J., Statistical Pattern Recognition: A Review, *IEEE Transaction on PAMI*, vol 22., No. 1, 2000, pp. 4-37.
4. Hansen L.K., Salamon P., Neural Networks Ensembles, *IEEE Transactions on PAMI*, vol. 12, no. 10, 1990, pp. 993-1001.
5. Kuncheva L.I., *Combining pattern classifiers: Methods and algorithms*, Wiley-Interscience, New Jersey, 2004.
6. Hashem S., Optimal linear combinations of neural networks, *Neural Networks*, 10(4), 1997, pp.599-614.
7. Tumer K., Ghosh J.,Linear and Order Statistics Combiners for Pattern Classification [in:] Sharkley A.J.C. [ed.] *Combining Artificial Neural Networks*, Springer, 1999, pp. 127-155.
8. Fumera G., Roli F., A Theoretical and Experimental Analysis of Linear Combiners for Multiple Classifier Systems, *IEEE Transaction on PAMI*, vol. 27, no 6, 2005, pp.942-956.
9. Kittler J., Alkoot F.M., Sum versus Vote Fusion in Multiple Classifier Systems, *IEEE Transaction on PAMI*, vol. 25, no. 1, 2003, pp. 110-115.
10. Kuncheva L.I., Whitaker C.J., Shipp C.A., and Duin R.P.W., Limits on the Majority Vote Accuracy in Classier Fusion, *Pattern Analysis and Applications*, 6, 2003, pp. 22-31.
11. Duin R.P.W., Juszczak P., Paclik P., Pekalska E., de Ridder D., Tax D.M.J. *PRTools4, A Matlab Toolbox for Pattern Recognition*, Delft University of Technology, 2004
12. Devijver P. A., Kittler J., *Pattern Recognition: A Statistical Approach*, Prentice Hall, London, 1982
13. Duda R.O., Hart P.E., Stork D.G., *Pattern Classification*, John Wiley and Sons, 2001.
14. Quinlan J.R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993
15. Koszalka L., Skworcow P., Experimentation system for efficient job performing in veterinary medicine area, *Lecture Notes in Computer Science*, vol. 3483, 2005, pp. 692-701.

Ranked patterns and structular linearisation of data sets[1]

Leon Bobrowski[a,b]

[a] *Faculty of Computer Science, Bialystok Technical University*

[b] *Institute of Biocybernetics and Biomedical Engineering, PAS, Warsaw, Poland*

Abstract: Linear separability of learning data sets is a fundamental concept in the theory of neural networks. Powerful methods of data mining originating from the support vector machines (*SVM*) are also linked to this concept.
We consider here the method of learning sets linearization based on data transformations by the ranked structures. This method is called "structural linearization" and it is also linked to the problem of ranked patterns extraction from learning sets. Extraction of ranked patterns or designing ranked structures from the formal neurons can be based on minimization of the convex and piecewise linear (*CPL*) criterion functions. The structural linearization discussed in the paper is also linked to the problem of the classifiers assembling (voting classifiers) or aggregation of information from different sources.

Key words: linear separabilty, ranked patterns, structural linearisation, convex and piecewise linear (CPL) criterion function

1. Introduction

The theory and applications of artificial neural networks began with the model of Perceptron [1]. Hierarchical layers of formal neurons (multilayer perceptrons) still belong to the most fundamental models of neural networks [2]. Linear separability of learning sets in a selected feature space was a big issue of the perceptron learning algorithms. The problem of linear separability is still very current in designing neural networks. Designing neural network means here a choice of a neural network structure (e.g. number of layers and number of elements in particular layers) and the weights of connections from elements of a lower layer to elements of the next, higher layer.

At present, the support vector machines (*SVM*) constitute the most powerful tools of data mining with important applications, among others in bioinformatics [3]. The *SVM* solution defines such optimal hyperplane which linearly separates the original or the transformed learning sets. The literalizing transformations in the *SVM* approach are based on the search and application of adequate kernel functions.

In this paper we consider the structural linearization of learning sets based on data transformations by the ranked structures ([4], [5], [6]). Application of this method to the problems of weighted voting of classifiers and an aggregation of information from different sources is also taken into focus.

---

[1] This work was partially supported by the W/II/1/2006 and SPB-M (COST 282) grants from the Białystok University of Technology and by the 16/St/2006 grant from the Institute of Biocybernetics and Biomedical Engineering PAS.

2. Separable learning sets

Let us assume that each of the m analysed objects O_j ($j = 1,\ldots,m$) is represented as the so - called feature vector $\mathbf{x}_j = [x_{j1},.....,x_{jn}]^T$, or as a point in the n-dimensional *feature space* $F[n]$ ($\mathbf{x}_j \in F[n]$). The components (*features*) x_{ji} of the vector $\mathbf{x}_j$ are numerical results of a variety of examinations of a given object O_j. The feature vectors $\mathbf{x}_j$ can be of mixed, qualitative-quantitative type with binary or real components x_{ji} ($x_{ji} \in \{0,1\}$ or $x_{ji} \in R$).

We assume that the database contains descriptions $\mathbf{x}_j(k)$ of m objects $O_j(k)$ ($j = 1,.....,m$) labelled according to their *category* (*class*) ω_k ($k = 1,\ldots,K$). The learning set C_k contains m_k feature vectors $\mathbf{x}_j(k)$ assigned to the k-th category ω_k

$$C_k = \{\mathbf{x}_j(k)\} \quad (j \in I_k) \tag{1}$$

where I_k is the set of indices j of such feature vectors $\mathbf{x}_j(k)$ from the class ω_k which belong to the set C_k.

Definition 1: The learning sets C_k (1) are *separable* in the feature space $F[n]$, if they are disjunctive in this space ($C_k \cap C_k = \varnothing$, if $k \neq k'$). It means that the feature vectors $\mathbf{x}_j(k)$ and $\mathbf{x}_{j'}(k')$ from different learning sets C_k and $C_{k'}$ cannot be equal:

$$(k \neq k') \Rightarrow (\forall j \in I_k) \; and \; (\forall j' \in I_{k'}) \;\; \mathbf{x}_j(k) \neq \mathbf{x}_{j'}(k') \tag{2}$$

We are also considering the separation of the sets C_k (1) by the hyperplanes $H(\mathbf{w}_k, \theta_k)$ in the feature space $F[n]$

$$H(\mathbf{w}_k, \theta_k) = \{\mathbf{x}: \mathbf{w}_k^T \mathbf{x} = \theta_k\}. \tag{3}$$

where $\mathbf{w}_k = [w_{k1},....,w_{kn}]^T \in R^n$ is the weight vector, $\theta_k \in R^1$ is the threshold, and $(\mathbf{w}_k)^T\mathbf{x}$ is the inner product.

Definition 2: The learning sets (1) are *linearly separable* in the n-dimensional feature space $F[n]$ if each of these sets C_k can be fully separated from the sum $\cup C_i$ ($i \neq k$) of the remaining sets C_i by some hyperplane $H(\mathbf{w}_k, \theta_k)$ (3):

$$\begin{aligned} &(\exists k \in \{1,...,K\}) \;\; (\exists \, \mathbf{w}_k, \theta_k) \;\; (\forall \mathbf{x}_j(k) \in C_k) \;\; \mathbf{w}_k^T \mathbf{x}_j(k) > \theta_k. \\ &\qquad\qquad \textbf{\textit{and}} \;\; (\forall \mathbf{x}_j(k) \in C_i, \, i \neq k) \;\; \mathbf{w}_k^T \mathbf{x}_j(k) < \theta_k \end{aligned} \tag{4}$$

In accordance with the relation (4), all the vectors $\mathbf{x}_j(k)$ belonging to the learning set C_k are situated on the positive side ($\mathbf{w}_k^T \mathbf{x}_j(k) > \theta_k$) of the hyperplane $H(\mathbf{w}_k, \theta_k)$ (3) and all the feature vectors $\mathbf{x}_j(i)$ from the remaining sets C_i are situated on the negative side ($\mathbf{w}_k^T \mathbf{x}_j(k) < \theta_k$) of this hyperplane.

The separation of data sets C_k by the hyperplanes $H(\mathbf{w}_k, \theta_k)$ (4) can by linked to data transformation by a layer of K formal neurons $NF(\mathbf{w}_k, \theta_k)$, The formal neuron $NF(\mathbf{w}_k, \theta_k)$ is defined by the threshold decision rule $r(\mathbf{w}_k, \theta_k; \mathbf{x})$

$$r = r(\mathbf{w}_k, \theta_k; \mathbf{x}) = \begin{array}{ll} 1 & \textit{if} \quad \mathbf{w}_k{}^T\mathbf{x} \geq \theta_k \\ 0 & \textit{if} \quad \mathbf{w}_k{}^T\mathbf{x} < \theta_k \end{array} \qquad (5)$$

where r is the output $\mathbf{w} = [w_1,....,w_n]^T \in R^n$ is the weight vector, $\theta_k \in R^1$ is the threshold and $\mathbf{x} = [x_1,....,x_n]^T$ is the input feature vector.

The feature vector $\mathbf{x}$ activates (r=1) the formal neuron $FN(\mathbf{w},\theta)$ if and only if $\mathbf{x}$ is situated on the positive side of the hyperplane $H(\mathbf{w},\theta)$ ($\mathbf{w}^T\mathbf{x} \geq \theta$).

Layer of K formal neurons $FN(\mathbf{w}_i,\theta_i)$ transforms the feature vectors $\mathbf{x}$ into the binary vectors $\mathbf{r} = \mathbf{r}(\mathbf{x})$, where $\mathbf{r} = [r_1,...,r_K]$, $r_i = r(\mathbf{w}_i,\theta_i;\mathbf{x})$ (5). Such layer can be used as the classifier with the allocation rule given below

$$\textit{if} \ (r(\mathbf{w}_k,\theta_k; \mathbf{x}) = 1) \ \textit{and} \ \ (\forall i \neq k) \ r(\mathbf{w}_i,\theta_i; \mathbf{x}) = 0 \ \textit{then} \ (\mathbf{x} \in \omega_k) \qquad (6)$$

A vector $\mathbf{x}$ is allocted to the class ω_k if only one neuron $FN(\mathbf{w}_k,\theta_k)$ in this layer is activated. We can remark that if the learning sets C_k (1) are linearly separable (4), then the layer of K formal neurons $FN(\mathbf{w}_i,\theta_i)$ with the rule (4) properly allocates all the feature vectors $\mathbf{x}_j(k)$.

3. Ranked classifiers

Let us take into account a ranked family $Q(m_r)$ of m_r classifiers (rules, neural networks) Q_i with the binary outputs q_i ($q_i \in \{0,1\}$). Each classifier Q_i is defined on feature vectors $\mathbf{x}$ by an individual decision rule $q_i(\mathbf{x})$:

$$q_i = q_i(\mathbf{x}) \quad (i = 1,....., m_r) \qquad (7)$$

The feature vector $\mathbf{x}$ activates the classifier $q_i(\mathbf{x})$ if and only if $q_i(\mathbf{x}) = 1$. The ranked relation "*prior to*" is defined in the below manner between any two classifiers Q_l and Q_i from the family $Q(n)$.

Definition 1: The classifier Q_l (7) is *prior to* the classifier Q_i if and only if $l < i$.

Definition 2: The *active field* S_i (i =1,…, m_r) of the ranked layer is a set of such feature vectors $\mathbf{x}_j(k)$ ($\mathbf{x}_j(k) \in C_k$) which activate the classifier Q_i ($q_i(\mathbf{x}) = 1$) and do not activate any of the prior classifiers Q_l ($l < i$).

$$S_i = \{\mathbf{x}_j(k): r_i(\mathbf{x}_j(k)) = 1 \ \textit{and} \ (\forall l < i) \ r_l(\mathbf{x}_j(k)) = 0\} \qquad (8)$$

Definition 3: The classifier $q_i(\mathbf{x})$ (7) is deterministically admissible for the k-th class ω_k (k = 1,…,K) if and only if the active fields S_i (8) contains the feature vectors $\mathbf{x}_j(k)$ from only one learning set C_k (1).

Definition 4: The classifier Q_i is statistically admissible at the level α ($0 < \alpha < 0.5$) for the k-th class ω_k ($k = 1,\ldots,K$) if and only if the active field S_i (6) contains the feature vectors not only from one set C_k and the fraction f_i of elements $\mathbf{x}_j(l)$ from other sets C_l is less than α ($f_i < \alpha$).

The fraction f_i of elements $\mathbf{x}_j(l)$ from other sets C_l is defined by the expression below:

$$f_i = m_i'(k) / (m_i(k) + m_i'(k)) \qquad (9)$$

where $m_i(k)$ is the number of elements $\mathbf{x}_j(k)$ from the set C_k in the active field S_i (6) and $m_i'(k)$ is the number of elements $\mathbf{x}_j(l)$ from other sets C_l (1) in this field.

It can be seen that the number m_r of the ranked classifiers R_i with the non-empty active fields S_i (6) fulfills the below conditions.

$$K \le m_r \le m \qquad (10)$$

where K is the number of the learning sets C_k (1), and m is the number of feature vector $\mathbf{x}_j(l)$ in these sets..

The lowest possible number $m_r = K$ appears when the active field S_i (6) is extremely large ($\forall k \in \{1,\ldots,K\}$ $S_k = C_k$ (1). The highest possible number $m_r = K$ appears when each active field S_i (8) contains only one feature vector $\mathbf{x}_j(l)$. It can be expected that classifiers R_i with large active fields S_i (8) have greater generalizing power than classifiers with small active fields.

Definition 5: The admissible classifiers Q_i with the decision rules $q_i(\mathbf{x})$ (5) form the *ranked layer* if and only if each feature vector $\mathbf{x}_j(k)$ from the sets C_k (1) belongs to some active fields S_i (8).

The ranked layer of m_r admissible classifiers Q_i transforms each feature vector $\mathbf{x}_j(k)$ into the vector $\mathbf{q}_j(k)$ with m_r binary components $q_i(\mathbf{v}_i;\mathbf{x}_j(k))$.

$$\mathbf{q}_j(k) = [q_1(\mathbf{v}_1; \mathbf{x}_j(k)),\ldots.., q_{mr}(\mathbf{v}_{mr};\mathbf{x}_j(k))]^T \qquad (11)$$

where $\mathbf{v}_i = [v_0, v_1,....,v_{n'}]^T$ is a vector of parameters.

The transformed vectors $\mathbf{q}_j(k)$ form the sets D_k:

$$D_k = \{\mathbf{q}_j(k)\} \quad (j \in I_k) \qquad (12)$$

The separability (2) of the sets C_k (1) is preserved by the ranked layer as it is specified below.

Lemma 1. If the sets C_k (1) are separable, then the sets D_k (12) are also separable.

Proof: The sufficient condition for the sets D_k separbility can have the form (2)

$$(k \neq k') \Rightarrow (\forall j \in I_k)\ and\ (\forall j' \in I_{k'})\ \ \mathbf{q}_j(k) \neq \mathbf{q}_{j'}(k') \qquad (13)$$

The above condition results directly from the definition of the *active field* S_i (8). Two vectors $\mathbf{q}_j(k)$ and $\mathbf{q}_{j'}(k')$ belong to active sets S_i and $S_{i'}$. related to different classes ω_k and $\omega_{k'}$. So these vectors cannot be equal ($\mathbf{q}_j(k) \neq \mathbf{q}_{j'}(k')$). □

Theorem 1: The ranked layer of m_r admissible classifiers Q_i transforms the separable sets C_k (1) into linearly separable (4) sets D_k (12).
Proof: Let us assign the following parameters α_i to each active field S_i (8).

$$(\forall i \in \{1,\ldots..,m_r\})\ \ \alpha_i = 1 / 2^i \qquad (14)$$

The hyperplane $H(\mathbf{w}_k,\theta_k)$ (3) used for separating the set D_k (12) from the sum $\cup D_i$ of the remaining sets D_i ($i \neq k$) can be defined by the weight vector $\mathbf{w}_k = [w_{k1},....,w_{kn}]^T$ with the following components w_{ki}

$$(\forall i \in \{1,\ldots..,m_r\})\ \ \textbf{\textit{if}}\ S_i \in D_k\ \ \textbf{\textit{then}}\ \ w_{ki} = \alpha_i\ \ \textbf{\textit{and}} \qquad (15)$$
$$\textbf{\textit{if}}\ S_i \notin D_k\ \ \textbf{\textit{then}}\ \ w_{ki} = -\alpha_i$$

By direct computations we can verify the below inequalities.

$$(\exists k \in \{1,...,K\})\ (\forall \mathbf{q}_j(k) \in D_k) \qquad \mathbf{w}_k^T \mathbf{q}_j(k) > 0 \qquad (16)$$
$$\textbf{\textit{and}}\ \ (\forall \mathbf{q}_j(i) \in D_i,\ i \neq k) \qquad \mathbf{w}_k^T \mathbf{q}_j(i) < 0$$

where $\mathbf{w}_k$ is the weight vector with the components w_{ki} (14). The inequalities (14) mean that the sets D_k (12) are linearly separable (4).□

The considerations above are similar to the proof given in the paper [4].

4. Linear aggregation of the ranked classifiers

Let us consider a two-layer hierarchical structure. The first layer of this structure is formed by m_r admissible systems (classifiers) Q_i (*Def.* 3) with binary outputs (decision rules) $q_i(\mathbf{v}_i;\mathbf{x})$ (11), where $\mathbf{v}_i = [\ v_1,....,v_{n'}]^T$ is a vector of parameters (Fiq. 1). The second layer is formed by K formal neurons $FN(\mathbf{w}_i,\theta_i)$ with the decision rules $r(\mathbf{w}_k,\theta_k;\mathbf{q})$ (5) based on the binary output vectors $\mathbf{q} = [q_1(\mathbf{v}_1,\mathbf{x}),\ldots\ldots,q_{mr}(\mathbf{v}_{mr},\mathbf{x})]^T$ (11) of the first layer. Each formal neurons $FN(\mathbf{w}_i,\theta_i)$ agregates linearly the classifiers Q_i of the first layer.

Figure 1: Linear aggregation of the ranked classifiers Q_i

In accordance with Theorem 1, the ranked layer transforms the separable sets C_k (1) into linearly separable (4) sets D_k (12). As a consequence, the layer of formal neurons $FN(\mathbf{w}_i,\theta_i)$ with the weights vectors $\mathbf{w}_i$ (15) fully separates the learning sets C_k (1). In other words, the classifier (6) based on the layer of formal neurons properly allocates all feature vectors $\mathbf{x}_j(k)$ from the learning sets C_k (1).

Designing two-layer ranked classifiers is based mainly on a search of adequate elements of the first layer. This term can mean classifiers defined by a relatively small number of parameters and characterized by active fields S_i (8) with many elements $\mathbf{x}_j(k)$ of the learning sets C_k (1). It could be expected that such classifiers will possess a sufficient genearalization power. In the paper [] an example was given of the first layer build from the formal neurons $FN(\mathbf{w}_k,\theta_k)$ (5) or the hyperplanes $H(\mathbf{w}_k,\theta_k)$ (3). Three other possibilities of the first layer designing are described below.

Example 1: The hyperplanes $H'(v_k, \theta_k)$ (3) defined by only two parameters v_k and θ_k are used in designing the active fields S_i (8) of the first layer

$$H'(v_k, \theta_k) = \{\mathbf{x}: v_k x_k = \theta_k\} \qquad (17)$$

The hyperplane $H'(v_k, \theta_k)$ (17) in the n-dimensional feature space $F[n]$ is parallel to the all axises x_i with the exeption of the k-th axis x_k. The decision rule based on the hyperplanes $H'(v_k, \theta_k)$ (17) can be given in the following form

$$\textbf{\textit{if}}\ (v_k x_k \geq \theta_k)\ \textbf{\textit{then}}\ (q_k(\mathbf{x}) = 1)\ \textbf{\textit{else}}\ (q_k(\mathbf{x}) = 0) \qquad (18)$$

Desiging the active fields S_i (8) means in this case a search for an adequate sequence of parameters (v_1,θ_1), (v_2,θ_2),……., (v_{mr},θ_{mr}).

Fig. 2. The active fields S_i (8) defined by the hyperplanes $H'(v_k,\theta_k)$ (3).

Example 2: The Euclidean balls $K_E(\mathbf{v}_k,\rho_k)$ with the center $\mathbf{v}_k = [v_1,....,v_{n'}]^T$ and the radius ρ_k can be used in the first layer

$$K_E(\mathbf{v}_k,\rho_k)= \{\mathbf{x}: (\mathbf{x} - \mathbf{v}_k)^T(\mathbf{x} - \mathbf{v}_k) = \rho_k^2\} \qquad (19)$$

The decision rule based on the balls $K_E(\mathbf{v}_k,\rho_k)$ can have the following form

$$\textbf{\textit{if}}\ (\mathbf{x} - \mathbf{v}_k)^T(\mathbf{x} - \mathbf{v}_k) \leq \rho_k^2)\ \textbf{\textit{then}}\ (q_k(\mathbf{x}) = 1)\ \textbf{\textit{else}}\ (q_k(\mathbf{x}) = 0) \qquad (20)$$

Fig. 3. The active fields S_i (8) defined by the balls $K_E(\mathbf{v}_k, \rho_k)$ (19)

Example 3. The L1 balls $K_{L1}(\mathbf{v}_k, \rho_k)$ with the center $\mathbf{v}_k = [v_1,....,v_{n'}]^T$ and the radius ρ_k can be used in the first layer

$$K_{L1}(\mathbf{v}_k, \rho_k) = \{\mathbf{x}: | x_1 - v_1| +.....+ | x_n - v_n| = \rho_k^2\} \qquad (21)$$

The decision rule based on the balls $K_{L1}(\mathbf{v}_k, \rho_k)$ can be given in the following form

if $(| x_1 - v_1| +....+ | x_n - v_n| \leq \rho_k^2)$ ***then*** $(q_k(\mathbf{x}) = 1)$ ***else*** $(q_k(\mathbf{x}) = 0)$ (22)

Fig. 4. The active fields S_i (8) defined by the balls $K_{L1}(\mathbf{v}_k,\rho_k)$ (21)

Designing the active fields S_i (8) in the Examples 2 and 3 means a search for an adequate sequence of parameters $(\mathbf{v}_1,\rho_1)$, $(\mathbf{v}_2,\rho_2)$,……., $(\mathbf{v}_{mr},\rho_{mr})$.

5. Concluding remarks

Designing ranked layers for the purpose of classification was discussed in the paper. The proposed method is based on building a sequence of admissible active fields S_i (*Def*. 3).

The ranked layers have a fundamental property of data sets linearization. It means that the separable data sets C_k (1) are transformed by the ranked layer into linearly separable (4) sets D_k (12). In this way a simplified representation of a classification problem is reached. Linearization of data sets by the ranked layers could find important applications also in the methods originating from the Support Vector Machines (*SVM*) [3].

It can be expected that the statistical approach to the deterministic approach towards designing ranked layers (*Def*. 4) will increase the chance of obtaining classifiers with a large discrimnative power.

Bibliography

[1]. Rosenblatt F.: *Principles of neurodynamics*, Spartan Books, Washington 1962

[2]. O. R. Duda and P. E. Hart, D. G. Stork: *Pattern Classification,* J. Wiley, New York, 2001.

[3]. Vapnik V. N.: *Statistical Learning Theory*, J. Wiley, New York 1998

[4]. L. Bobrowski, "Design of piecewise linear classifiers from formal neurons by some basis exchange technique'" *Pattern Recognition*, **24**(9), pp. 863-870, 1991

[5]. L. Bobrowski, 'The ranked neuronal networks", *Biocybernetics and Biomedical Engineering*, Vol. 12, No. 1-4, pp. 61-75, 1992

[6]. Bobrowski L.: *Eksploracja danych oparta na wypukłych i odcinkowo-liniowych funkcjach kryterialnych* (*Data mining based on convex and piecewise linear (CPL) criterion functions*) (in Polish), Technical University Białystok, 2005

Using Latent Semantic Indexing for Data Deduplication

Michael Spiz

University of Washington, Institute of Tehnology
Tacoma, WA 98402, USA
spizm@u.washington.edu

Abstract. This paper presents a method for deduplicating records using latent semantic indexing (LSI). When merging two datasets from two different sources, there is often a problem with overlap between the records. Finding these duplicate records can be challenging since the format of the data is often different between databases. Existing methods for data deduplication focus primarily on using data cleaning and approximate string matching techniques. While these methods are effective for finding duplicates in records with few words, such as names and addresses, they do not work as well for records with more terms such as corporate names. The system described in this paper uses LSI techniques to discover duplicates in a dataset. This article shows the LSI deduplicator performs more accurately on test and real-world data than existing techniques, but at the expense of runtime and resource utilization.

1 Introduction

Many companies collect data from various independent sources, placing that information inside a data warehouse. When merging two datasets from two different sources, there is often a problem with overlap between the records. Finding these duplicate records can be challenging since the format of the data is often different between databases. Existing methods for data deduplication focus primarily on using data cleaning and approximate string matching techniques. While these methods are effective for finding duplicates in records with few words such as names and addresses, they do not work as well for records with more terms such as corporate names. The system described in this paper uses latent semantic indexing (LSI) techniques to discover duplicates in a dataset.

As an example, the company Diligenz Inc. acquires databases of corporate records from many states throughout the USA. Initially, these databases have different schemas, are in different formats, and adhere to different standards. All these databases are then carefully transformed into a standard schema to help ease the retrieval of data. However, searching for corporate entities in this data is still a challenging task. First, even though each corporate name has a unique identifier associated with it, there are no identifiers that associate a name with a corporate entity. This means that it is not possible to find all names belonging to a franchise operation, large institution, or any other entity with multiple

names. Secondly, it is difficult to find distinct corporate entities that do business in multiple states because of the differences in the underlying schemas for the individual state databases. The desire to find a way to be able to autonomously generate associations between these names is the motivation for this paper.

Section 1.1 defines LSI and how it works. The problem statement is outlined in Sect. 2. At this point, the LSI deduplication system is introduced in Sect. 3. It describes both the framework that was used and how the LSI indexer functions. The datasets used for testing are described in Sect. 4. Using those datasets, Sect. 5 shows an analysis of speed and accuracy of the indexer and compares it with two other methods. Similar existing projects are described in Sect. 6. Possible future improvements are discussed in Sect. 7. Lastly, the paper finishes with some conclusions from the project in Sect. 8.

1.1 Latent Semantic Indexing

Latent Semantic Indexing (LSI) is a method for uncovering the semantics in a body of text in such a way that it is easier to process and search by a computer. It is also called Latent Semantic Analysis (LSA). In bodies of text, it is common that different words are used to describe the same concept. Likewise, a single word can also have multiple meanings. This makes performing a search for a document by searching for instances of a word an inefficient way of finding all the documents related to that word [1, 2].

This is where LSI comes in. LSI creates a matrix whose columns are comprised of words and rows comprised of documents. The values of the matrix are weights that are proportional to the number of times those words appear in the document. The weights are structured so that words that are rarer have a greater weight. This is because words that are used commonly throughout a document such as "the", "and", or "this", do not carry as much relevance in the document as other words do. Afterwards, a singular value decomposition (SVD) is applied to the matrix. This turns each document into a multi-dimensional vector that represents the content inside the document. Documents with similar terms and content will end up having vectors which are closer to each other than unrelated documents. Performing queries for words are more successful after performing LSI on the set of data because the search results will also return documents that have words that are synonymous with the queried word.

2 Problem Statement

Consider an example where one has a data warehouse that contains millions of filled-out forms. These forms contain filings for bank loans made out by various corporations. In order to gather meaningful statistical data on the data filled out in these forms, one needs to make sure every form is entered in a consistent and uniform manner. This, unfortunately, is never the case. Figure 1 shows an example listing of what possible names for the University of Washington could look like. The names can be manually clustered into two, possibly three

different groups: two branches of the University of Washington and Washington University. A human can quickly discern the differences between the variations and abbreviations and properly cluster the list. Having a computer perform this kind of operation accurately and autonomously is a much more difficult task.

University of Washington
University of Washington, Tacoma
Univ of WA
Washington University
UW
UWT

Fig. 1. Sample list of university names.

Figure 2 shows an actual sampling of names from a database of corporate information in California. Again, just by glancing over the list, one can make educated guesses as to which names belong to the same corporate entity. For instance, "BANK ADMINISTRATION INSTITUTE" and all of its chapters should be grouped together. In contrast, all the "Bank of..." rows should be grouped separately except for "BANK OF CANTON OF CALIFORNIA" since each (minus the exception) describes a separate location for a bank and likely are separate corporations.

Current implementations of data deduplication systems focus mostly on the analysis of substrings and their similarity to one another [3–5]. Many use methods that involve more complexity, such as using stemming algorithms, soundex and metaphone, or Hidden Markov Models in order to clean and normalize the data as much as possible and to decrease word complexity. However, after performing all their preprocessing operations, these systems ultimately end up using some form of substring comparison in order to make the final determination of whether two records are duplicates.

This method for finding duplicates works very well with person names, addresses and other data that involves single words or simple structures. For longer, more complex strings, such as company names, the standard methods become less effective. The reason is that company name variations involve the addition, subtraction, and transposition of entire words. Existing deduplication methods focus more on the character and word level variations and not phrase-level changes. These also do not detect synonymous word replacements such as "company" and "corporation". Replacements of that sort occur often in corporate name databases.

Having a system that can deal with such information would be a valuable asset for finding distinct corporate entities within a raw dataset of business names. Using such information one can compile these associations and either sell this to customers or perform analysis on them.

BANK ADMINISTRATION INSTITUTE
BANK ADMINISTRATION INSTITUTE-DESERT-SEA CHAPTER
BANK ADMINISTRATION INSTITUTE-GOLDEN GATE CHAPTER
BANK AND GOLDBERG PRODUCTIONS, INC.
BANK AUDI (U.S.A.)
BANK AUSTRIA CREDITANSTALT AMERICAN CORPORATION
BANK COMPLIANCE ASSOCIATES, INC.
BANK OF BERKELEY
BANK OF BURLINGAME
BANK OF CANTON OF CALIFORNIA
BANK OF CANTON OF CALIFORNIA LEASING CORPORATION
BEVERLY MANOR INC. OF BURBANK
BEVERLY MANOR INC. OF BURBANK SOUTH
BURBANK ENTERTAINMENT VILLAGE, L.L.C.
BURBANK ENTERTAINMENT VILLAGE ASSOCIATES, L.L.C.
TERRA BURBANK PARTNERS ONE, LLC
TERRA BURBANK PARTNERS TWO, LLC
RIVERBANK DENVER, INC.
RIVERBANK PARAGON, INC.

Fig. 2. Sample of corporate names from the California corporate database.

3 LSI Deduplication System

3.1 Febrl

Febrl (Freely Extensible Biomedical Record Linkage) [6] is an open-source Python project designed to perform data cleaning, and deduplication of database information. Febrl was specifically designed by the biomedical community to help sort through patient name and address datasets. These datasets often contain typos and duplicate information. It is very tedious to sift through the data by hand, so this project was created to automate the process.

Febrl is divided into several modular components. The first component is the standardizer, which is responsible for cleaning and normalizing the incoming dataset. For example, there are several ways to write out the word "avenue" such as "ave." or "av." The standardizer uses lookup tables to convert all these word variations into a single word. Febrl comes with standardization lookup tables for names, titles, and addresses, however other industry-specific lookup tables can be added to this without much problem.

Often in a database, names and addresses are stored in only two columns. For more effective deduplication, the program needs to separate these into their individual components (first name, middle initial, last name, house number, street, etc.). The problem with parsing out these columns is that certain components of the name and address may be missing or transposed depending on how the data was originally entered. To handle this, Febrl uses Hidden Markov Models

(HMMs) to parse the names and addresses. The names and addresses are parsed out into tokens. Each token is assigned a tag based on information in lookup tables (known names, cities, postal codes, etc.). Once each token is tagged, they are passed into a trained HMM to determine what token most likely corresponds with what output field.

At this point, all the records are cleaned and partitioned into separate fields. The last step is to link similar records together. Comparing every record to every other record is quite computationally expensive especially when there are a large number of records to compare. To help this situation, each record should only be compared to a subset (or block) of the entire record set. There are several blocking algorithms in Febrl that can be used to perform this task. The trick to these is to avoid having a matching record outside of the comparison subset.

Finally, a comparison algorithm is used in order to get a measure of similarity between two records. Febrl implements several approaches, one of which, the Naïve Bayes classifier, adds up weights for each of the attributes and uses that as a similarity measure.

As is apparent, there are many pieces to making Febrl work. The design of the system is still very developer-oriented. There is no user interface, and the configuration of a project is done using a controller class. While this gives the program a high learning curve, it also makes it modular and capable of accepting enhancements easily. Febrl makes for an excellent platform for researching algorithms and techniques for data cleaning and record linkage.

3.2 LSI Indexer

Instead of using a traditional string similarity based approach to deduplicate records, the system presented here uses LSI to index and then deduplicate the test records. The reasoning for using such an indexer is that datasets containing strings of multiple terms, such as company names, may contain enough semantic meaning in them to allow one to cluster possible duplicate records together.

The LSI indexer is built on top of the Febrl framework [6]. Since Febrl is written in Python, the indexer is also written in the same language. Figure 3 shows an overview of how the data is processed within the system.

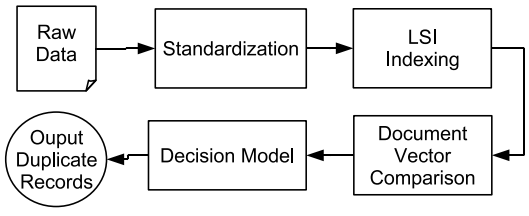


Fig. 3. Overall data flow for the LSI indexer.

The basic functionality of the LSI indexer is as follows. The program is controlled through a Python script. It first defines a schema for the data to be imported and then uses Febrl to import a comma separated data file. The fields that are to be indexed are defined in the script. As each record from the dataset is read, the parser combines the marked fields into a single large string labeled with an identifier (ID). This tuple is then passed to the indexer for processing. The indexer then tokenizes the string into words, stems each word using the Porter stemming algorithm [7], and then adds each unique stemmed word into a hash table.

Once all the records are read in, a matrix is constructed where each column represents a stemmed word in the hash table, and each row is a normalized vector of the word frequencies. A Singular Value Decomposition (SVD) transformation is then performed on the matrix. The resulting vectors in the matrix, at this point, represent the semantic position within the dataset. The cosine distance is then measured between each vector. If the distance is within a specified tolerance, those two vectors are marked as duplicates. The result is a collection of sets of vectors that are potential duplicates.

4 Datasets

The evaluation was performed on two very different datasets in order to compare the LSI indexer against the other indexers. The first dataset is a sample listing of corporation names and addresses in the state of California. This database holds a collection of all registered businesses in the state of California. It holds approximately 30 million records, taking up a total of 18GB of space. There is, however, much less than this amount of businesses in California. So why are there so many records? Many businesses have franchises, branches, or separate divisions. These are required by the state to have separate names. These names, however, are not linked in any way to the parent company, so it is difficult to resolve these to a list of distinct corporate entities. This problem becomes multiplied when merging this type of corporate data from different states into one database. For this reason, this data is a good candidate for deduplication. By having a list of distinct corporate entities, one can gain a better picture of who is in control of what businesses. Table 1 shows a sample of the dataset.

The second dataset that was chosen comes from a dataset generator that is included with Febrl. It is designed to create real-world name and address datasets that contain duplicate information [8]. The generator allows one to control the number of duplicates in the dataset and the number and type of errors that are introduced into the data. A major advantage for having such a dataset is that one is guaranteed to know which records are duplicates. This makes benchmarking different algorithms much simpler. Table 2 shows a sample of the dataset.

In order to be able to get a good gauge for the number of duplicates that exist in the corporate dataset, the sample was limited to 150 records. This way, it was feasible to manually look through the records and find the actual number of duplicates.

Table 1. Raw data from the California corporate database.

EntityID	EntityName	MailAddress2	MailCity	MailState
630	(THE) UNIVERSITY HEIGHTS IMPROVEMENT ASSOCIATION INC.			
6141	222 UNIVERSITY AVENUE - NORTH	P O BOX 4456	BURLINGAME	CA
6142	222 UNIVERSITY AVENUE CORPORATION	P O BOX 4456	BURLINGAME	CA
14061	810 UNIVERSITY AVENUE INC.	810 UNIVERSITY AVE	BERKELEY	CA
14858	921 UNIVERSITY INC.	921 UNIVERSITY AVE	BERKELEY	CA
35927	ABLE UNIVERSITY PRESS INC.	4084 N BURGE RD	STOCKTON	CA
38101	ACADEMIC CREDIT UNIVERSITY	5181 OVERLAND AVE	CULVER CITY	CA
38153	ACADEMIC RESEARCH UNIVERSITY	4860 LONG BEACH BLVD	LONG BEACH	CA
38445	ACADEMY OF INTERNATIONAL SOCIETY OF PEOPLE UNIVERSITY	2315 CYPRESS CIRCLE DRIVE	LOMITA	CA
42920	ACHIEVEMENT UNIVERSITY	AOKI 1102-1-8-10 HIGASHIGOTANDA SHINAGAWAKU	TOKYO	JAPAN
44588	ACP UNIVERSITY CORPORATION	30129 VIA RIVERA	RANCHO PALOS VERDES	CA
48350	ADAM SMITH UNIVERSITY	3463 STATE STREET SUITE 363	SANTA BARBARA	CA

Table 2. Raw data for generated address list (columns truncated)

rec_id	given_name	surname	street_number	address_1	...
rec-359-dup-0	joel	baynes	16	beasley street	...
rec-74-org	alaiyah	leslie	33	becker place	...
rec-305-dup-0	finn	maspwn	556	bisdee street	...
rec-195-org	flynn	lock	41	hallett place	...
rec-232-org		white	237	westgarth street	...
rec-316-org	april	grubb	38	chuculba crescent	...
rec-90-org	ellie	carich	31	wynn street	...
rec-368-org	hannah	george	50	bural court	...
rec-97-org	courtney	highet	46	sugarloaf circuit	...
rec-393-org	thomas	penno	7	arabana street	...
rec-172-dup-0	chloe	vreugdemburg	35	chaton place	...
rec-25-org	blade	coleman	144	britten-jones drive	...

5 Analysis

The following describes the procedures used for discovering duplicate records using the different methods. Both the LSI indexer and the other indexers use the same code for reading the data. Since the LSI indexer only focuses on term frequencies, all the data for each row is concatenated and analyzed as a single phrase during the reading process.

The LSI indexer groups semantically similar rows together based on the cosine distance between the reference row and its potential duplicates. The higher the cosine distance, the more similar the two rows are. For the analysis of the California corporate data, a cutoff cosine distance of 0.55 was chosen.

The indexers built into Febrl require more information in the setup process. Each column of data can be processed and indexed differently in order to maximize the usefulness of the indexes that are generated. The first "bigram" indexer represents Febrl's default settings. For the address list, names were encoded for comparison using metaphone encoding [9], addresses were compared using the Jaro-Winkler approximate string matching algorithm [9], and cities were compared using keying error distance [9]. The second "edit distance" indexer changes the blocking method to sorting, and changes the comparators to use edit distance.

Since the California data sample is from real-world data, the actual number of known duplicate records is unknown. Thus, the only way to measure the effectiveness of the algorithms is to compare the results against a human. After manually analyzing the sample dataset, 22 duplicate records were found.

The results in table 3 show the percentage of duplicate records found compared to a manual human-based search of the data. The algorithms that most closely match what a human would pick receives the highest percentage. The false-positive (FP) rate shows what percentage of marked duplicate records were not human-labeled as being a duplicate.

Table 3. Benchmark results from the LSI deduplicator.

	Percentage of duplicates detected	FP Rate	Run time (seconds)
Human (manual) comparison	100%	0%	900
LSI with address dataset	96%	0%	130
LSI with corporate dataset	36%	50%	190
Bigram with address dataset	54%	0%	3
Bigram with corporate dataset	9%	50%	3
Edit distance with address dataset	76%	0%	3
Edit distance with corporate dataset	13%	40%	13

The main drawbacks of the current implementation of the indexer are its execution time and memory consumption. All of the algorithms perform the same number of disk accesses. The LSI indexer needs to store and process its

information in matrices, so it scales in $O(n^2)$ time. The memory consumption has the same problem. A matrix needs to be built for the index that is $m * n$ in size where m is the number of unique terms in the dataset and n is the number of items in the dataset. Both of these factors make scaling the size of the dataset difficult. In contrast the other indexers only need to look at subsets of the datasets. This makes them scale in $O(n \cdot log(n))$ time.

Regardless, the performance of the LSI algorithm was lower than expected. The source of the problem was traced to the dot multiplication functions in Python. In order to get a more accurate gauge of the true performance of the algorithm, a version of the system was designed that piped all the matrix and vector operations through Mathematica 5.1. As is apparent in Table 4, the LSI indexer performance is faster than the Febrl indexers for the size of the dataset being used, but slows down quickly due to the complexity of the algorithm.

The LSI algorithm achieved the highest accuracy of the algorithms tested even though it needed the least amount of data cleaning and pre-processing. This means that there is still more potential to get greater accuracy by performing more complex data cleaning up front. Greater accuracy can also be achieved by normalizing the data using domain-specific transformations such as street names, surnames, or dates.

Table 4. Performance of LSI Indexer through Mathematica

Dataset Size	Run Time (seconds)
150	1.2
250	5.2
500	32.6

6 Discussion

6.1 ALIAS

Sunita Sarawagi created a system called ALIAS which provides a framework for interactive deduplication [10]. The goal of the system is to utilize an active learning approach to label duplicate records in a database. The system starts with a small initial training set. Duplicate training records are classified with a "1" and non-duplicate records with a "0". Each of these pairs are also run through a set of various similarity functions (edit-distance, soundex, etc.) that may also include domain-specific functions to enhance accuracy. The system then relies on the values generated from these various functions to determine the best way to find duplicate records within the dataset. The program finds record pairs which have the greatest level of uncertainty as to their similarity and presents those to the user for classification. Doing this iteratively, the program learns how to distinguish between similar and dissimilar records in the database. This

system provides a novel way for finding duplicates in a dataset using training data provided by a person. The system, however, only uses syntactic measures for similarity. In certain situations, it would likely benefit from semantic similarity measures that LSI can provide.

6.2 Iterative Record Linkage

Indrajit Bhattacharya and Lise Getoor worked on a method for deduplication that tries to find linked records by examining them in the context of the other records in the database [11]. To create these links, the authors use an iterative approach: they make several passes over all the records in the database, creating new associations between records each time. Again, the approach used is syntactic in nature. As described in Sect. 2, there are certain types of data that do not lend themselves well to being analyzed in such a manner. The LSI indexer attempts to provide a semantic approach to finding duplicates.

7 Future Work

The programming done for this project was meant as a proof-of-concept. Therefore, not much emphasis was placed on performance and scalability. A possible first step to improving the current program would be to incorporate a distributed LSI algorithm. This would help in dividing the work of creating the index among multiple computers. Work has already been done in this area [12, 2]. Some of these also use sparse matrices for the document-term matrix in order to reduce memory usage. There also exist much faster implementations of algorithms that approximate the SVD algorithm [13].

An additional solution to the scaling issue would be to use a two-stage approach to the indexing by using LSI as a blocking algorithm. In the first stage, entries would be indexed and clustered by their syntactical similarity, just like Febrl currently does. In the second stage, LSI would be performed only on subsets, or blocks of syntactically similar data. This would reduce the complexity of the system to $O(n \cdot log(n))$, but may have negative effects on accuracy.

A problem with the current implementation is that it is sensitive to character transpositions, misspellings, and alternate word forms. Since the base unit for LSI is words, it treats each spelling variation as a separate word with a separate semantic meaning. It would benefit the program to create a type of stemming algorithm that would stem words and also account for character transpositions and misspellings.

The datasets that this type of program analyzes contain a very diverse set of terms due to the large number of names that are present in addresses and businesses. The diversity of terms increases the size of the document-term matrix, which in turn slows down indexing. One way to improve the performance of the program would be to implement a pass prior to indexing that would remove all terms that only occur once in the document space. These terms do not provide the LSI algorithm any semantic information.

It would be interesting to see this algorithm included into a hybrid deduplication system such as ALIAS [10]. Having a system that can take both string similarity measures and semantic similarity into account at the same time would likely allow for finding results with greater accuracy.

8 Conclusion

This paper has shown that LSI provides an effective means for deduplicating records on both generated and real-world data. Both Febrl's existing algorithms and the LSI algorithm that were tested were not completely accurate, especially with the corporate data. However, the LSI deduplicator found a higher percentage of duplicates in both datasets than Febrl's built-in algorithms. The main drawback of using LSI is its performance and scalability. Future work will involve looking into ways to improve the scalability and to combine different deduplication approaches into a smarter system.

References

1. Landauer, T.K., Foltz, P.W., Laham, D.: Introduction to latent semantic analysis. Discourse Processes (25) (1998) 259–284
2. Letsche, T.A., Berry, M.W.: Large-scale information retrieval with latent semantic indexing (1996)
3. Bilenko, M., Mooney, R.J.: Adaptive duplicate detection using learnable string similarity measures. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, Washington, D.C., ACM Press (2003) 39–48
4. Chaudhuri, S., Ganjam, K., Ganti, V., Motwani, R.: Robust and efficient fuzzy match for online data cleaning. In: Proceedings of the 2003 ACM SIGMOD international conference on Management of data, San Diego, California, ACM Press (2003) 313–324
5. Christen, P., Churches, T.: Febrl - freely extensible biomedical record linkage (2005)
6. Group, A.D.M.: Febrl 0.3 (2005)
7. Porter, M.F.: An algorithm for suffix stripping. Program **14**(3) (1980) 130–7
8. Christen, P.: Probabilistic data generation for deduplication and data linkage. In: Sixth International Conference on Intelligent Data Engineering and Automated Learning, Brisbane (2005)
9. Black, P.E.: Dictionary of algorithms and data structures (2006)
10. Sarawagi, S., Bhamidipaty, A.: Interactive deduplication using active learning. In: Proc. of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Canada (2002)
11. Bhattacharya, I., Getoor, L.: Iterative record linkage for cleaning and integration. In: Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, Paris, France, ACM Press (2004) 11–18
12. Tang, C., Dwarkadas, S., Xu, Z.: On scaling latent semantic indexing for large peer-to-peer systems. In: SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (2004) 112–121

13. Gorrell, G., Webb, B.: Generalized hebbian algorithm for latent semantic analysis. In: Proc. Interspeech. (2005)
14. Navarro, G.: A guided tour to approximate string matching. ACM Computing Surveys **33**(1) (2001) 31–88

A commodity platform for Distributed Data Mining – the HARVARD System

Ruy Ramos, Rui Camacho and Pedro Souto

LIACC, Rua de Ceuta 118 - 6^o 4050-190 Porto, Portugal
FEUP, Rua Dr Roberto Frias, 4200-465 Porto, Portugal
{ruyramos, rcamacho, pfs}@fe.up.pt
http://www.fe.up.pt/~rcamacho

Abstract. Systems performing Data Mining analysis are usually dedicated and expensive. They often require special purpose machines to run the data analysis tool. In this paper we propose an architecture for distributed Data Mining running on *general purpose* desktop computers. The proposed architecture was deployed in the **HARV**esting **A**rchitecture of idle machines fo**R** **D**ata mining (HARVARD) system. The Harvard system has the following features. Does not require special purpose or expensive machines as it runs in general purpose PCs. It is based on distributed computing using a set of PCs connected in a network. In a Condor fashion it takes advantage of a distributed setting of available and idle computational resources and is adequate for problems that may be decomposed into coarse grain subtasks. The system includes a dynamic updating of the computational resources. It is written in Java and therefore runs on several different platforms that include Linux and Windows. It has fault-tolerant features that make it quite reliable. It may use a wide variety of data analysis tools without modification since it is independent of the data analysis tool. It uses a easy but powerful task specification and control language.

The HARVARD system was deployed using two data analysis tools. A Decision tree tool called C4.5 and an Inductive Logic Programming (ILP) tool.

keywords: Data Processing, Parallel, Distributed Computing, Problem Solving Environments

1 Introduction

As a result of more complex and efficient data acquisition tools and processes there is in almost all organisations huge amounts of data stored. Large amounts of money are invested in designed efficient data warehouses to store such amounts of data. This is happening not only in Science but mainly in industry. Existing OLAP techniques are adequate for relatively simple analysis but completely inadequate for in-depth analysis of data. The discipline of Knowledge Discovery

is Databases (KDD) is a valuable set of techniques to extract useful information from large amounts of data (data ware houses). However, KDD [2] is facing nowadays two major problems. The amounts of data are becoming so large that it is impractical (or too costly) to download the data into a single machine to analyse it. Also, due to the amounts of data or to its distributed nature in large corporations, it is the case that the data is spread across several physically located data bases. These two problems prompted for a new area of research called Distributed and Parallel Data Mining[3]. This new area addresses the problem of analysing distributed databases and/or making the analysis in a distributed computing setting.

This paper reports on the development and deployment of a computational distributed system capable of extracting knowledge from (very) large amounts of data using techniques of Data Mining (DM) based on Machine Learning (ML) algorithms. The system developed was designed to accommodate easily three of the main stages of a KDD process: pre-processing, Data Mining and post-processing. The system enables the use of different pre and post-processing tasks and the use of different Data Mining tools without any change to it.

Our proposal envisages the following objectives. To provide the user with a simple but powerful language to specify the tasks in the main stages of the KDD process of data analysis. To allow the use of common personal computers in the data analysis process. The computational system uses only idle resources in the organisation. The system will run on a large variety of platforms (Windows and Linux at least) and use parallel and distributed computation. It may run in a cluster or grid environment. The system may use data distributed among several physically separated databases. The system is *independent* of the data analysis (ML) tool. It has facilities to monitor the KDD process and facilities to recover from major system faults.

Fulfilling the above objectives will have the following advantages. The user may easily configure a data analysis process adequate for his specific needs. The analysis will be affordable to a wide range of organisations since the costs involved are quite low — the machines used are common desktop machines. The analysis process does not disturb the normal work of the organisation since it only uses idle computational resources. A large number of organisations may use it since it runs of a variety of platforms and accesses data that may be physically distributed among several databases.

To attain the objectives of the project we propose the HARVARD computational system. The system allows the user to describe each task of the KDD process in a XML format and to specify the workflow of their execution in a easy to use specification language. The system runs with a single Master node and an unlimited collection of Slave nodes. Both the Master and the Slaves are programmed in Java. The Slaves may access the data directly in a database (using JDBC) and all necessary software for the data analysis tool via HTTP. As will

be described later with more detail the Master reads the KDD process description and generates a workflow graph that is used by a scheduler. The scheduler uses also information concerning the computational resources available. A Slave node may download the data and data analysis tool, monitor its workload and executes the tasks determined by the master. Information concerning the status of the resources are updated regularly.

The rest of the paper is organised as follows. In the Section 2 we present the proposed architecture. In Section 3 we describe the event-driven working of the architecture. In Section 4 we describe how the sub-tasks of the KDD process may be specified by means of a simple but powerful language. We present the fault-tolerant features of HARVARD in Section 5. The deployment of the HARVARD system is described in Section 6. Section 7 compares other projects features with the HARVARD capabilities. We conclude in Section 8.

2 The Architecture

We consider the KDD process as composed of a set of tasks that are executed according to a workflow plan. Both the tasks description and the workflow plan are provided by the user in two files. One file contains the tasks description and the second one contains the workflow. The workflow is specified using a control description language that is presented in Section 4. The tasks specification is made using XML. The information concerning an individual task includes the name and location of the tool used in the task, the location of the data to be processed and the computational resources required (platform type, memory and disc needs, etc). An example of such a specification is presented in Figure 1.

The distributed architecture of the HARVARD system is composed of a Master node and a set of computing nodes called Slave nodes. The Master node is responsible for the control and scheduling the sub-tasks of the whole KDD process. Each Slave node executes application (sub-)tasks assigned by the Master node. Each node is composed by four modules that execute specific tasks to make the overall system working.

In what follows we refer to Figure 2 for the modular structure of both the Master and the Slave nodes. We now describe in detail each node type.

The Master node

The Master node is responsible for reading the KDD process specification and "executing it". Each task is of the KDD process is handle by the system as a **Working Unit** (**WU**). Each WU is assigned to one or more machines. The assignment of a WU to more than one machine makes the the system more tolerant to faults. It occurs when there are idle machines available and the task is expected to have long running times. There are other fault tolerant features that we will refer below. When a WU finishes, the results associated with it are stored

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<workunit>

  <identification> T1 </identification>
  <apllication>
    <urlapl>www.fe.up.pt/ilp/IndLog/indlog.tgz</urlapl>
    <script>www.fe.up.pt/ilp/IndLog/script-ilp.scp</script>
    <parameters>www.fe.up.pt/ilp/IndLog/parameters.txt
    </parameters>
  </apllication>
  <data>
    <dataset>kdd99</dataset>
    <DBserver>www.fe.up.pt/mysql</DBserver>
    <DB>kdd99</DB>
    <translationscript>toilp.scp</translationscript>
  </data>
  <requirements>
    <memory>1000</memory>              # MB
    <processor>Pentium</processor>
    <harddisc>1000</harddisc>          # MB
  </requirements>
  <estimatedtime> 30 </estimatedtime> # seconds
  <results>
   <filename>kdd99.out</filename>
   <DBserver>www.fe.up.pt/mysql</DBserver>
   <DB>kdd99</DB>
  </results>
</workunit>
```

Fig. 1. An illustrative simple example of a task specification in XML.

and the status of the workflow graph updated. When the graph is completely traversed, meaning that the KDD process has finished, the result is returned to the user.

The Master node is composed by four modules: the Task Manager; the Scheduler; the Resource Manager and; the Communications module.

The Task Manager Module The basic function of the **Task Manager** (TM) module is to store, update and provide information concerning the tasks of the KDD process. The TM module constructs a graph structure representing the workflow of the tasks.

It first reads and stores the specifications of all tasks composing the KDD process and then reads the workflow plan of the tasks and constructs a workflow graph structure. This module updates the status of the tasks in the graph and associates the results of each one when finished. At the end it informs the user of the results of the KDD process. It may also be used to monitor the whole KDD process providing the user with information about each task status.

The TM interacts with the Scheduler module. Looking at the workflow graph this module informs the Scheduler of ready to process tasks, provides a complete specification of each task and receives information concerning the terminations and results of each task.

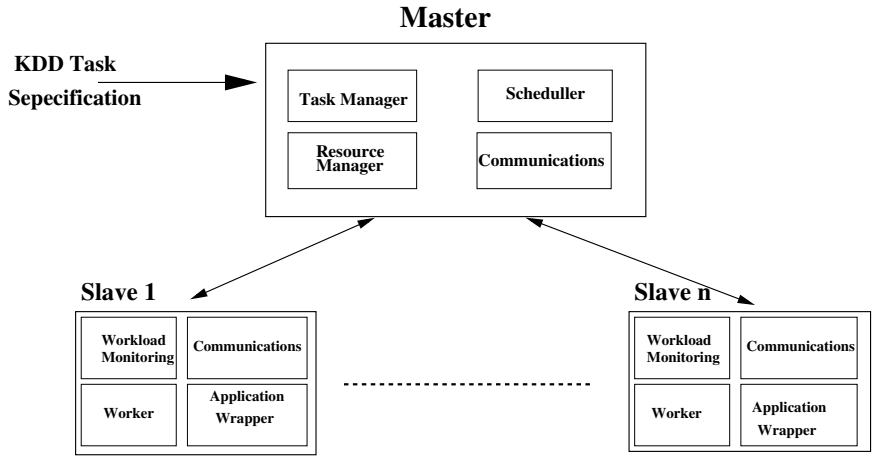


Fig. 2. Diagram of the architecture's components.

The Resources Manager Module The **Resources Manager (RM)** module stores and periodically updates information concerning the computational resources usable by the HARVARD system. When the system starts this module loads from a database the static information concerning all the computational resources usable by the system. That information is dynamically updated during the system execution. The information of each resource includes the type of platform and CPU, the available amount of memory and disc space and a time-table with the periods the machine may be used. The workload of each machine is communicated periodically to this module in order for the system to have an updated view of the resources. The RM module has a method (a match maker) to compute the "best" computational resource for a given request from the Scheduler. Each computing resource has a time-table of availability of the resource and the policy of use. This information indicates when the machine is available and under what conditions. The usage conditions may indicate that HARVARD may use the machine only when there are no users logged in or by specifying a workload threshold that must be respected at all times.

The Task Manager module receives, from the Scheduler, requests for available machines satisfying a set of resources requirements and returns the best match at the moment. This module alerts the TM whenever a task must be rescheduled in two situations: if the machine the task was runing in is severely delayed to notify the TM module of its workload and; if the pre-established period of use of the machine is expired[1].

[1] In this case the task running on the machine is terminated.

The Communications Module The **Communications (COM)** module is the only channel to access the world outside a node. All messages or requests concerning components or resources outside the node are processed by the COM module. This module exists in both Master and Slave nodes. To accomplish that task it implements several communication protocols that includes: RMI, socket, HTTP and JDBC. All these allows a slave to download the task's required software (HTTP), download the data (JDBC), send messages to the Master (sockets or RMI) and allows the Master to send messages to the Slaves (socket or RMI). It also allows the Master to keep a DB backup of its status and activities (JDBC) to allow a full recover in case of fault.

This Master COM module interacts via RMI or sockets with the COM module of the Slave to send messages. In a Master node the messages to be sent are received from the Scheduler module or the Resources Manager module. The former sends messages concerning task assignments and control directives whereas the later sends tasks status updated to be stored in a DB (fault tolerant purposes). The COM module receives and redirects the workload messages for the RM module. Received messages concerning tasks results are redirected to the TM module.

The Scheduler Module The **Scheduler** module controls the execution of the tasks composing the KDD process, launching, rescheduling or stopping the Work Units. The scheduler may also decide to assign a WU[2] to more than one Slave node. The scheduler inspects the workflow graph where the tasks interconnections and status are represented to decide what tasks to activate and when.

The Scheduler asks the Resource Manager module for the best match machine satisfying a given Work Unit requirements. With the results of such request the Scheduler assigns that WU to the given Slave and notifies the Slave via the Communications module. Whenever there is a change in the status of a WU the Scheduler is informed by the Task Manager of that event and triggers the (re)scheduling a new task.

A Slave node

A Slave node does the actual data analysis work by running the Data Ming tool. In order to have a distributed system that is independent of the Data Mining tool the DM tool is involved in a wrapper that directly controls it. Each Slave also reports periodically its workload to the Resource Manager module of the Master. It is through the Slave's Communications module that the Slave downloads the DM tool and the data to be processed, and stores the results of the local analysis.

[2] The ones considered more critical for some reason like training longer execution times.

Each Slave has four modules: the Workload Monitoring (WM); the Worker (WO); the Application Wrapper (WR) and; the Communications (COM) module.

The Worker module The WU message is interpreted in this module. A WU usually results in several steps to be performed. A typical WU for analysing data involves the downloading of the analysis tool, the download of the data, the processing and the return of the results. The Worker module controls all these steps by asking the Communications module to fetch the software and data and triggering the Application Wrapper module to execute the analysis. Finally it sends (via Communications module) the results to the Master.

The Application Wrapper module The WR module completely controls the DM tool. It supplies the DM tool input stream and collects whatever appears at the DM output stream. Through the input stream the module provides the commands for the DM tool. The commands are provided in a file indicated in the Working Unit specification. The output stream is stored in a file as the results file. The results file is uploaded to a database entry as indicated in the WU specification. For the time being all the analysis of the results files are done in other follow up WU where special scripts written by the user do the necessary analysis. This keeps the system independent of the DM tool.

The Workload Monitoring module This module monitors periodically the workload of the machine it is running and reports that information to the Resources Manager module of the Master. It also detects in a user has logged in the machine. In the later case the Master is informed that the task running will be terminated. The Slave enters a idle state where it just waits for the machine to be idle again.

Communications Module The slave Communicating module is the only channel to the outside world. It has capabilities to download software using HTTP or ftp protocol, it may download data from a DB using JDBC and it can send and receive messages to and from the Master using RMI or sockets.

The Communications module interacts with all modules of the Slave node delivering and receiving messages.

3 An event-driven implementation

The decomposition into modules according to functionality enabled an easier development of the system. Each module is "self-contained" and implement functionalities like: scheduling, resource management, task management, communication, application program control etc. A major goal in the proposed architecture design is that although there are a lot of modules and threads hat execute a

wide range of tasks they should not compete for the CPU unnecessarily. If all the modules and threads were running at the same time the system would be slow and the application program would take much more time to run and return the results. The proposed architecture is designed based on an event-driven and message passing implementation. Each module, on both the Master and Slave nodes, has an internal mail box. Each module reads its mailbox and processes the messages received. The read operation is blocking and therefor if there are no messages they don't need to work and are in a waiting state that does not compete for CPU. Whenever a message arrives (an event) the message is processed and its content may require some computations. It is only in this situation that the module uses the CPU. The processing of a message may require some computations and usually involves the exchange of messages with other modules activating this way other functionalities. After processing a message a module executes again a read operation that puts it in the waiting state, not computing for CPU, if there are no messages.

The overall result is that a module runs (uses CPU) only when required otherwise does not compete for CPU.

4 Sub-tasks workflow description language

The HARVARD system accepts as input a file describing the workflow of the sub-tasks composing the KDD process. The workflow is a graph with two kinds of nodes: sequential nodes and; parallel nodes. Each node stores a set of tasks to be executed or edges to other nodes. In a sequential node the set has an order and that order represents the sequential execution of the sub-tasks that must be respected. In a parallel node the tasks in the set may start all at the same time. In a parallel node there may be a *barrier* specifying the tasks that must terminate before the "execution" of the node is considered terminated. The graph has a root node where the tasks of the KDD process start executing.

An example of the use of the workflow description language is shown in Figure 3. The example represents a simplified KDD process where the pre-processing is a simple feature subset selection (T1 through T8) using a simplified version of parameter tuning (T3, T4, T6 and T7). After the pre-processing the predictive power of the model is estimated using a 5-fold Cross Validation (T9 through T14). Finally the model is constructed in sub-task T18. All of the T_is are described in a separate file in XML and available to the Task Manager module of the Master node of the HARVARD system. The example is explicitly made simple for illustrative purposes. It illustrates how sequential and parallel executions may be interleaved and how easy is to specify a KDD process.

Some of the steps in a KDD process are done quite frequent and most often are the same for a vast number of domains. For example feature subset selection or the DM tool parameter tuning are quite frequent pre-processing tasks in a KDD process. For these frequent tasks we intend to provide an interface (or

```
# This is the Sub-Tasks workflow description
# using the Task Control Language
seq
  T1  # use 70%/30% train/test set

  par # feature subset selection
    seq
     T2  # get dataset without Att1 (DS1)
     par
       T3  # eval dataset DS1 using m = 2
       T4  # eval dataset DS1 using m = 50
     endpar
    endseq

    seq
     T5  # get dataset without Att2 (DS2)
     par
       T6  # eval dataset DS2 using m = 2
       T7  # eval dataset DS2 using m = 50
     endpar
    endseq
  barrier T[3,4], T[6,7] # wait for tasks T3, T4, T6 and T7 to end

  T8  # choose best set of attributes
  T9  # choice of the blocks for a 5-fold CV

  par  # execute the 5 fold Cross Validation
    T[10-14] # do each CVi
  barrier T[10-14]  # wait for all CV folds (tasks T10 up to T14)

 T18 # run with all data to produce the final theory
endseq
```

Fig. 3. An illustrative simple example of a sub-task workflow description.

macro in the workflow description language) where the user just indicates the task and their parameters. For example: do a top-down feature selection up to two attributes or tune the "p" parameter using the values p1, p2 and p3. The system will the "unfold" that macro into the corresponding graph structure.

5 Fault-tolerance features

The HARVARD system can recover from the failure of the Master node. During its activity the Master node stores status information in an external Database[3]. The stored information is enough to enable any node to restart as a Master node and continue the work from the point where the former Master failed. When starting the system the user may specify that one of the Slave nodes (called *backup Master* node) may take control should a Master failure occur. In such a case the *backup Master* receives periodically a "alive message" from the Master. If M[4] consecutive of such alive messages fail the *backup Master takes control*. It recovers the task status from the Data Base and sends all slaves a

[3] Located on a different machine.
[4] A system parameter

message announcing its new status as Master node.

In a normal execution context each Slave node, registered at the Master, does a periodic "alive confirmation" sending a message to the Resource Manager module of the Master. A Slave node that misses more than N[5] confirmation messages is considered unusable. The task running in a Slave that changes state from running to unusable is marked as requiring rescheduling. It will be assigned another Slave node.

6 Deployment of the HARVARD system

Just to test the feasibility of our approach and not to compare the system's performance on a specific problem we used a freely available large dataset. We have applied both C4.5 and IndLog on the analysis of a Intrusion Detection dataset. This dataset was part of the KDD 1999 conference challenge and is freely available[6]. It is a good test case because there are 24 types of attacks (24 class values) and 4898431 labelled firewall log entries (dataset cases) and 2984154 test cases and 39 attributes that results in 743 MB of data. The data and the status logs where stored in two MySQL databases in separate machines. We used a laboratory with 15 PCs where Master students have classes and use for developing their practical works. The machines have dual boot so sometimes they start with Linux and sometimes with Windows.

To analyse the data we used and ILP algorithm [10,11] implemented in Prolog called IndLog [6,7] and a Decision Tree tool called C4.5 . To use each of these tools we had to provide scripts to execute the tool, scripts to extract the results from the outputs of the tools and scripts to compare the results and choose the best settings based on those results. In order to be able to use different tools the user has to provide such scripts. We have available such scripts for these two tools (C4.5 and IndLog). We intend to produce such scripts for other popular data analysis tools such Apriori (Association Rules analysis), CART (decision and regression analysis) etc.

At this stage we successfully tested the execution of the system in Linux and Windows platforms. The fault-tolerant features of the system were tested by simulating the breakdown of the Master node and to establish that a Slave node would interrupt whenever a user logged in to that machine.

7 Related work

Our system is designed to take advantage of idle computers in an organisation and adequate for problems that may be decomposed into coarse grain subtasks.

[5] A system parameter

[6] from: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

We now present related projects that include some of the HARVARD objectives but none of them have all the functionalities we implemented in HARVARD. Related work includes the Condor system [9], the Boinc system [8] and the Knowledge Grid [1]. In the rest of this section we present, briefly, the related projects and compare them with the HARVARD system.

Condor operates in a university campus or organisation environment. The system aims at maximising the utilisation of workstations with as little interference as possible between the jobs it schedules and the activities of the people who own the workstations. "Condor is a specialised job and resource management system for computing intensive jobs. Like other full-featured systems, Condor provides a job management mechanism, scheduling policy, priority scheme, resource monitoring and resource management. Users submit theirs jobs to Condor when and where to run the based upon policy, monitors their progress, and ultimately informs the user upon completion" [9]. Condor allows almost any application that can run without user interaction to be managed. This is different from systems like Set@home and Protein Folding@home. These programs are custom written. Source code does not have to be modified in anyway to take advantage of these benefits. Code that can be re-linked with the Condor libraries gain two further abilities: the jobs can produce check-points and they can perform remote system calls [9].

The Boinc (Berkeley Open Infrastructure for Network Computing)[7] is a platform that makes it easy for scientists to create and operate public-resource computing projects. Workstations connected to the Internet by phone or DLS line can participate in some project and share its own computer power to solve scientific problem whenever the device is idle. The process is very simple since people interested to participate just instal a client software that connects to a project master server. So, when workstation is idle some tasks may be executing. Some projects like SETI@home, Folding@home use the Boinc platform [8].

The Knowledge Grid is a specialised architecture in data mining tools that uses basic global services from Globus architecture [1]. The architecture design of the Knowledge Grid follows the principles: data heterogeneity and large data sets handling; algorithm integration and independence; compatibility with grid infrastructure and grid awareness; openness, scalability, security and data privacy [5].

Like the Condor and Boinc projects, our architecture uses idle workstations and runs under heterogeneous environments with different operating systems (Linux, Windows, OS-X). Differently from Condor we have a much more powerful description language to specify the KDD process and the system just runs one KDD process a time. The HARVARD system differs from Boinc in the way the client receives information and where to find data and data analysis tool. In

[7] **University of California - Berkeley-** http://boinc.berkeley.edu/

HARVARD both data and the data analysis tool can be fetched in a DB using JDBC or in the Web via HTTP. HARVARD has also a much more sophisticated fault-tolerant functionalities.

Different from all the referred systems, our proposal implements a two-level language. A specific semantics for the administration of the data mining process, and other for specification of tasks of distributed processing. While one language is designed for the user to specify the process of the knowledge discovery, the other language is used by the system to manage the distributed computations.

In the like of Globus, the user sees a computation environment for knowledge extraction using algorithms Machine Learning in a virtual computer. In this way an analyst of businessman can use an interface that allows applying techniques of extraction of knowledge in a fast and efficient way without need of taking knowledge of the support platform. It is as if he had interacting in an environment of high performance commutating. The final result is a friendly and interactive platform with the user and efficient in terms of computational resources that makes use of distributed and idle resources.

8 Conclusions

We have proposed an architecture for Distributed Knowledge Discovery in Databases that is capable of using different Data Analysis tools without any modification. The architecture enables the use of general purpose desktop computers that are idle in an organisation. This latter feature makes the Data Mining process affordable to a wider range of organisations and the process of analysing the data does not interfere with the normal work of the organisation. The architecture features also allow the processing of data in distributed data bases and the migration of the data analysis tools to avoid the transfer of large volumes of data.

We have deployed the architecture in the HARVARD system and tested it using a Decision Tree data and Inductive Logic Programming system data analysis tools. We provide a workflow control language that enables the user to easily describe the KDD process with the sub-tasks and options of his choice. The HARVARD system is programmed in Java and may run in a wide range of platforms. It has fault-tolerant features that make it quite reliable.

As future work we envisage to develop to enrich the control description language to easy even more the specification of the Data Mining process by the user.

References

1. The Grid: Blueprint for a New Computing Infrastructure. eds. I. Foster and C. Kesselman. Morgan-Kaufmann Publishers, 1999, Ch. 11, pp 259–278.
2. Data Mining: Concepts and Techniques. eds. J. Han and M. Kamber. Morgan-Kaufmann Publishers, 2001.

3. Advances in Distributed and Parallel Knowledge Discovery. eds. Kargupta,H. and Chan, P. AAAI/MIT Press, 2000.
4. I. Foster and C. Kesselman. Globus: A Metacomputing Infrastructure Toolkit. International Journal of Supercomputer Applications, vol.11, N. 2, pp 115–128, 1997.
5. Mario Cannataro and Domenico Talia. The knowledge grid. Communications of the ACM, vol. 46, N. 1, pp 89–93, 2003.
6. Rui Camacho. Inducing Models of Human Control Skills using Machine Learning Algorithms. PhD thesis: Faculty of Engineering, University of Porto, Portugal, 2000.
7. Rui Camacho. IndLog –Induction in Logic. Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA 2004). Springer-Verlag, LNAI 3229, pp 718–721, 2004.
8. David P. Anderson BOINC: A System for Public-Resource Computing and Storage. GRID, pp 4–10, 2004.
9. M. J. Litzkow and M. Livny and M. W. Mutka Condor—A Hunter of Idle Workstations Proceedings of the 8th International Conference on Distributed Computing Systems, pp 104–111, 1988.
10. Stephen Muggleton. Inductive Logic Programming. New Generation Computing, vol. 8, N. 4, pp 295–318, 1991.
11. Stephen Muggleton and Luc De Raedt. Inductive Logic Programming: Theory and Methods. Journal of Logic Programming, vol. 19/20, pp 629–679, 1994.
12. Quinlan, J.R. Simplifying Decision Trees. International Journal of Man-Machine Studies, vol. 27, pp 221–234, 1987.

Regression trees and the evaluation of public goods

Angela Scaringella[1]

Facoltà di Sociologia, Università di Roma "La Sapienza"
via Salaria 113
I00198 Roma, Italy
Angela.Scaringella@uniroma1.it

Abstract. A method for the evaluation of public goods based on regression trees is proposed as an alternative to multivariate linear regression. The estimate is improved by making use of the splittings of the tree where of the relevant predictor appears as a surrogate variable.

1 Introduction

In [5] Harrison and Rubinfield analyzed the influence of air pollution concentration on housing values on data of housing in Boston area. They used for their analysis classical least squares multivariate regression for the price of homes, the dependent variable, as a function of independent variables among which was the nitrogen oxide concentation , the variable whose influence they wanted to study. Some tranformations were performed both on the dependent and the independent before computing the linear regression equation (see also [3] for an estensive analysis of this work). They used the linear regression equation to deal with the problem of contingent evaluation: in their case to give an economic value to reduction in air pollution.

The method of linear regression has the advantage of a great simplicity. However in Boston housing case so as in other analogous cases the analysis of data seems to indicate that nonlinear effects, non-considered in linear regression, may play a relevant role. It is worth to note that the transformations on the variables that are performed in [5] before linear regression have the goal of introducing some nonlinearity though in a non-systematic way. Moreover the method based on linear regression does not consider correlations that often subsist between the variable on which one is perforfming the evaluation analysis and other predictors. Therefore it is natural to try to find an alternative to linear regression. In particular one can try to use data mining methods that are designed to deal with complex models with a large number of variables. Among such methods classification and regression trees have often an efficiency comparable to other data mining methods but also present the advantage of higher perspicuity (see [7], [8], [9], [10], [2], [1]).

In particular CART program (see [1]), that is one of the most widely used programs for building classification and regression trees, can be applied to Boston

housing data. The chosen tree depends on some settings of the program and in particular on which method for choosing the splittings is used whether least mean square or least mean deviation.The result is that in the regression tree selected by the program the variable that was at the center of the investigation, i. e. the nitrogen oxide concentration, appears as a splitting variable just in few nodes of the selected tree. On the other side it appear as a surrogate variable in some of the splittings.

The problem arises then how to use regression trees in the framework of the problem of contingency evaluation where one has to detect the effect of one or some of the predictor variables on the response variable.

In this paper we discuss the problem of contingent evaluation in the framework of regression trees. We consider which procedures can be applied in examples such as that of Boston housing data in order to give an economic value to reduction in air pollution. In section 3 we deal with the problem of evaluating the effect of a variable taking also account of the fact that it appears as a surrogate variable in some of the nodes.

2 Contingent valuation and regression trees

Assume that we are dealing with a potentially infinite population of cases. Each case consists of a pair $(\mathbf{x}, y)$. $\mathbf{x}$ is a vector whose components are the predictor variables, whereas y is called the response variable. We have at our disposal a sample L taken from the population. The regression problem deals with devising a procedure that, starting from the set L, that is called the learning set, builds a function $d(\mathbf{x})$ defined on the vector of the predictor variables that should predict the value of the response variable y for a case with the first component equal to $\mathbf{x}$. Of course how good is a given function d in predicting y depends on the distance function we use for evaluating the distance between $d(\mathbf{x})$ and y. In classical linear regression this distance is evaluated by $(y - d(\mathbf{x}))^2$ mainly for mathematical convenience, but in general any other reasonable function can be used.

Regression trees deal with the problem of regression by means of a sequence of splittings based on sigle predictor variables or linear combination of them. We consider the case in which the splittings are based on a single predictor variable.

The problem of contingent evaluation of public goods based on surveys is linked with regression. We assume that the response variable represents a price or some averaged price for a certain family of cases with some given characteristics. We focus our attention on one of the predictor variables ξ in order to evaluate which is the total value of some change of it. This is of particular interest in the case of public goods that don't have a price on the market and must therefore be evaluated in some indirect way.

We can use the same cases of the learning set L that we assume to be representative of the whole population. We construct a new set of cases L' which is obtained from the set L by changing the value of the predictor variable ξ to a value ξ' by giving to it an increment corresponding to the amount that we want

to evaluate. The increase in value of the good we are interested in can then be estimated by

$$\frac{|P|}{|L|}\left(\sum_{x'\in L'} d(x') - \sum_{x\in L} d(x)\right), \tag{1}$$

where d is the function corresponding to the regression tree developed from the learning set L that assigns to each case the predicted value of the response variable, $|P|$ is the number of elements of the whole population and $|L|$ is the number of cases of the learning set.

One can ask whether the use of the learning set both for the construction of the tree and for the evaluation can lead a biased estimate. It seems to us that this procedure is legitimate in this case and does not present the same problems that arise when one uses the learning set for evaluating the prediction capability of a tree.

3 The use of surrogate variables

When one applies CART procedure for building the regression tree to the Boston housing data that were collected by Harrison and Rubinfield in order to evaluate the effects of changes in the level of athmospheric pollution, we obtain a tree such that the predictor variable related to pollution, i. e. nitrogen oxide concentration, appears as the the splitting variable of very few nodes. In this case if we apply formula 1 we get smply 0 or a very low evaluation.

This is unsatisfactory and could lead us simply to reject regression trees as a tool for evaluation in such cases. It must be said however that although the alternative method of linear regression gives us a definite answer, it has some arbitrariness in its application. In their analysis of Boston housing data Harrison and Rubinfield performed transformations on some of the predictor variables because in this way linear regression explains better the data. Without such trasformations or with different transformations the numerical result for contingent evaluation would be different. The linear regression method does not take account of the correlation between different predictor variables. Moreover in the case where we have many variables and strong non-linear effects are assumed to be present linear regression is certainly not adeguate.

The procedure that have described in previous section takes just account of the nodes where the relevant variable appears as the splitting variable. It is possible that it also appears as a surrogate variable in other nodes. This apperance implies that the splitting of these nodes can be simulated in some measure by using our variable because of some correlation subsisting between our variable and the splitting variable of that node. We have noted that possible correlations existing between variables are neglected in multivariate linear regression.

We propose therefore to deal with the problems we have exposed by suitably using surrogate variables. Surrogate variables are ordinarily introduced to treat cases for which some variable involved in the splittings of the tree is missing. In

this case one may try to use a different variable to simulate the splitting. One introduces an estimate based on the cases of the learning set for the probability that the original splitting and a new one based on the surrogate variable behave in the same way [1]:

$$p(s^*, \tilde{s}) = p_{LL}(s^*, \tilde{s}) + p_{RR}(s^*, \tilde{s}), \tag{2}$$

where s^* and $\tilde{s}$ are respectively the original and the surrogate splitting and $p_{LL}(s^*, \tilde{s})$ is the estimate based on the learning set that s^* and $\tilde{s}$S both send a case to the left and similarly $p_{RR}(s^*, \tilde{s})$ is the estimate for the probability that they both send a case to the right. Of course one is lead to choose among the splittings $\tilde{s}$ based on the surrogate variable the one for which $p(s^*, \tilde{s})$ reaches the maximum. Once this splitting has been chosen one needs to evaluate whether it can be reasonably used as a surrogate splitting. A necessary condition is that its error probability be small compared with a completely random procedure where a case is to the left or to the right with the same probabilities p_L and p_R as the original splitting s^* (estimated from the learning set) irrespectively of the values of the predictor variables. This can be expressed in terms of the *predictive measure of association* $\lambda(s^*|\tilde{s})$ between the splitting s^* and $\tilde{s}$:

$$\lambda(s^*|\tilde{s}) = \frac{\min(p_L, p_R) - (1 - p(s^*, \tilde{s}))}{\min(p_L, p_R)}. \tag{3}$$

If for some of the splittings the predictor variable we are interested in is one of the surrogate variable with a strictly possitive predictive measure of association, then we propose to substitute the original tree with one where these splittings are substituted with the surrogate ones. This new tree can be used for contingent evaluation by means of the formula 1 applied to the new tree.

Application of this method to Boston housing data give estimates that need to be analysed and compared to those obtained by means of different methods in order to to check their validity. It is also possible, as noted in [1], to combine regression trees and linear regression methods by first building a relatively small tree and then performing multiple linear regression in each of the terminal nodes (see also [2]). This could also be used to improve the estimates for evaluation.

References

1. Breiman, L., Friedman, J.H., Olshen, R. A., Stone, C. J.: Classification and Regression Trees. Chapman & Hall/CRC, Boca Raton London New York Washington D. C.(1984)
2. Fielding, A.: A binary segmentation: the automatic interaction detector and related techniques for exploring data structure. In: Muircheartaigh, C. A., Payne, C. (eds.): The analysis of survey data. Vol. I. Chichester; Wiley (1977)
3. Belsey, D. A., Kuh, E., Welsh R. E.: Regression diagnostics. Institute for Social Research, University of Michigan, New York (1980)
4. Friedman, J. H. : A tree-structured approach to nonparametric multiple regression. In: Gasser, T., Rosenblatt., M. (eds.) : Smoothing techniques for curve estimation. Springer-Verlag, Berlin (1979)

5. Harrison, D., Rubinfield, D. L.: Hedonic prices and the demand for clean air. J. Envir. Econ. and Management **5** (1978) 81–102
6. Mitchell, R. C., Carson, R. T. : Using Surveys to Value Public Goods: The Contingent Valuation Method. Resources for the Future, Washington, D. C. (1999)
7. Morgan, J. N., Sonquist, J. A. : Problems in the analysis of survey data, and a proposal. J. Amer. Statistic. Assoc. **58** (1963) 415–434
8. Sonquist, J. A., Morgan, J. N.: The detection of interaction effects. Ann Arbor: Institute for Social Research, University of Michigan (1964)
9. Sonquist, J. A.: Multivariate model building: the validation of a search strategy. Ann Arbor: Institute for Social Research, University of Michigan (1970)
10. Sonquist,J. A., Baker, E. L., Morgan, J. N.: Searching for structure. Rev. ed. Ann Arbor: Institute for Social Research, University of Michigan (1973)

Ranking the Rules and Instances of Decision Trees *

Yuh-Jye Lee and Yi-Ren Yeh

Dept. of Computer Science & Information Engineering,
National Taiwan University of Science & Technology, Taipei, Taiwan
{yuh-jye, M9315027}@mail.ntust.edu.tw

Abstract. Traditionally, decision trees rank instances by using the local probability estimations for each leaf node. The instances in the same leaf node will be estimated with equal probabilities. In this paper, we propose a hierarchical ranking strategy by combining decision trees and leaf weighted Naïve Bayes to improve the local probability estimation for a leaf node. We consider the importance of the rules, and then rank the instances fit in with the rules. Because the probability estimations based on Naïve Bayes might be poor, we investigate some different techniques which were proposed to modify Naïve Bayes as well. Experiments show that our proposed method has significantly better performance than that of other methods according to paired t-test. All results are evaluated by using AUC (Area under ROC Curve) instead of classification accuracy.

1 Introduction

In many machine learning problems, the goal is not only correctly classifying the instances. Ranking instances based on the class probabilities is more desirable for practical applications. For example, we would like to rank the resulting pages returned by a search engine to match the user's preference. In other words, the requirements of some applications and problems are not only the boolean response of "positive or negative", but also the ranking according to the probability of being positive. Most learning models provide the information of the probability estimation which might be used to rank instances. How to deal with those information from learning models is an interesting research issue [17, 7, 13, 15]. We will focus on ranking the rules and instances generated by decision trees model. We will use the term "local" to indicate that the estimation is based on a particular leaf node information. Thus the local probability estimation of $P(c|x)$ is given by $\frac{k}{n}$, where k is the number of training instances of class c in the leaf node and n is the total number of training instances in the leaf node. However,

* This work was supported in part by the National Science Council under the Grants NSC-93-2213-E-001-013, NSC-93-2422-H-001-0001, and NSC-93-2752-E-002-005-PAE, and by the Taiwan Information Security Center (TWISC), National Science Council under the Grants NSC 94-3114-P-001-001-Y and NSC 94-3114-P-011-001.

decision trees built by C4.5 [19] have been observed to produce poor probability estimations [18], even though they have a good performance in accuracy and an intelligible result. The local probability estimated by this way is rough and unreasonable. There are two main drawbacks of this intuitive approximation. The instances in the same leaf node will be estimated with the same probability. There is no difference between each individual instance in the same leaf node. Moreover, there is no difference between the leaf nodes which have the same proportion of positive instances but have different sizes. A more sophisticated way will be needed for the local probability estimation. In this paper, we describe a hierarchical ranking strategy by combining decision trees and Naïve Bayes. In the first step, we apply Laplace smoothing technique to re-estimate local probability for each leaf node. This will differentiate the leaf nodes which have the same proportion of positive class instances. We will rank the rules based on these local probabilities ordering. This also gives a grouping order for every instance. In the second step, we build a *leaf Naïve Bayes classifier*, the Naïve Bayes classifier generated by the instances in a leaf node, for each leaf node to rank the instances in the leaf node. However, building a Naïve Bayes classifier for a small training set tends to having a more serious *zero frequency* problem [21]. We will use the information provided by the global Naïve Bayes classifier which is generated by the entire training set as the prior knowledge in applying smoothing technique to the leaf Naïve Bayes classifier. Thus, we can rank every instance in a hierarchical way. We test our proposed methods on 7 public available datasets from UCI repository [2]. The experiment results show that our proposed method has significantly better performance than that of other methods according to paired t-test. All results are evaluated by using AUC (Area under ROC Curve) instead of classification accuracy.

The paper is organized as follows. In Section 2, we discuss some related works on improving decision trees ranking. Section 3 describes our hierarchical ranking method in details. We report the experiment results and discuss the details of tuning the parameters of our proposed method in Section 4. We conclude the paper with a brief discussion in Section 5.

2 Related Work

Traditional decision trees algorithms, such as C4.5 and ID3, have been observed to produce poor probability estimations. One of the reasons is that the algorithms aim at building a small and accurate tree which biases against good probability estimations [17]. Many techniques have been proposed to improve the probability estimations of decision trees [7, 16, 22, 12, 1]. Most of them apply the smoothing methods [7, 16, 22] to modify the local probability estimation. They correct the traditional probability estimation by some corrected ratio that shifts the probability toward the prior probability of the class. The most popular smoothing method is Laplace smoothing which estimates the probability $P(c|x)$ by $\frac{k+1}{n+|C|}$ at a leaf node, where k is the number of training instances of class c in the leaf node, n is the total number of training instances in the leaf

node, and $|C|$ is the number of classes. Laplace smoothing uses a uniform class distribution, $\frac{1}{|C|}$, as a prior knowledge to correct the reliability of probability estimation. The more instances a leaf node owns, the less the prior knowledge can affect. This can distinguish those leaf nodes which have the same proportion of positive instances but have different sizes. Although the smoothing methods improve the probability estimation by considering the reliability of the leaf node, the instances in the same leaf node still have the same probability estimations.

Other methods, like [12, 1], suggest that the probability estimations of the instances in the same leaf should be different. In [12], they average probability estimations from all leaves of the tree to produce different probability estimations in the same leaf node. However, this method only uses the attributes on the tree. It might easily produce duplicate points in pure nominal datasets, especially with a high dimensional dataset and a small tree. The method in [1] is similar to our proposed method. They propose a geometric score to distinguish the probability estimations of instances in the leaf. This method is limited to numerical attributes.

3 Hierarchical Ranking of Decision Trees

In order to keep the intelligibility and replace the same probability estimation in the leaf node, we describe a hierarchical ranking strategy by combining decision trees and Naïve Bayes. In first step, we rank the rules, and secondly we rank the instances in the leaf nodes.

3.1 Ranking Rules and Ranking Instances in a Leaf Node

In the first step, we adopt the smoothing techniques to rank the rules produced by decision trees. As described in section 2, many smoothing techniques have been proposed to improve the ranking of leaf nodes. The smoothing techniques will help us to differentiate the leaf nodes which have the same proportion of positive class instances without destroying the tree structure. We note that these smoothing techniques are often applied to unpruned trees in order to produce more different probability estimations. This might reduce the intelligibility of the tree because of the more complex rules. In our proposed method, Laplace smoothing is applied to the pruned trees directly. We then distinguish instances in the same leaf node via an embedded leaf Naïve Bayes.

Distinguishing instances in the same leaf node is an important part of decision trees ranking. In order to achieve this goal, we adopt the following strategies in the second step:

- Embedding a Naïve Bayes classifier in each leaf node,
- Combining leaf Naïve Bayes classifier and global Naïve Bayes in estimating $P(a_i|c)$, where a_i is the ith attribute and c is the given class,
- Using weighted Naïve Bayes classifier to incorporate the information provided by *non-tree* attributes.

We will discuss them in details in the following subsections.

3.2 Leaf Naïve Bayes Classifier

Naïve Bayes usually works well when tested on actual datasets, particularly combined with some of the attribute selection procedures [21]. Like decision trees, it also can deal with hybrid data types by the normal-distribution assumption of numerical attributes. For these reasons, we embed Naïve Bayes in each leaf node to rank instances. Since our goal is to distinguish the probability estimations of instances in a leaf node, the information of instances in the leaf node should be more important than that of the entire training set. Thus, we only use the instances in the leaf node to generate the leaf Naïve Bayes mainly and the information given by the global Naïve Bayes will be treated as prior knowledge for the leaf Naïve Bayes.

3.3 Estimating $P(a_i|c)$

Leaf Naïve Bayes can describe a good local distribution of $P(a_i|c)$ in the leaf node, but has less reliability due to the smaller size of training instances in a leaf node. In order to keep the local information and raise the reliability, we take the information from global Naïve Bayes into account when estimating $P(a_i|c)$ in leaf Naïve Bayes. For the flexibility, we also harmonize the information from leaf Naïve Bayes and global Naïve Bayes with a varied weight. How to determine the weight will be discussed more in section 4. Since Naïve Bayes adopts different strategies in estimating $P(a_i|c)$ for numerical and nominal attributes, we will describe our combination method for the two type attributes respectively.

In estimating $P(nominal_i|c)$, leaf Naïve Bayes will cause zero frequency more seriously. Traditionally, Naïve Bayes use Laplace smoothing to overcome zero frequency problem. Instead of using uniform prior knowledge in Laplace smoothing, we use m-estimate smoothing technique [7] to carry the prior knowledge from the entire training set as follows:

$$P(nominal_i|c) = \frac{k_i + w \cdot t_i}{N + w \cdot M} \tag{1}$$

where $\frac{k_i}{N}$ and $\frac{t_i}{M}$ are the estimations of $P(nominal_i|c)$ in the leaf Naïve Bayes and global Naïve Bayes respectively. The w is a nonnegative weight parameter for controlling the importance of the prior knowledge given by global Naïve Bayes. It can be varied for each leaf node.

The smoothing technique is only suitable for nominal attributes. It means that we need a new strategy to combine the information of the leaf Naïve Bayes and global Naïve Bayes in estimating the probability density measure for the numerical attributes. The probability density measure of ith attribute is denoted by $P(numerical_i|c)$ if the ith attribute is numerical type. We abused the notation a little bit here. Generally, Naïve Bayes conducts numerical attributes with normal-distribution assumption, which can be determined by the sample mean and sample standard deviation of the numerical attribute. The local probability is replaced by a local probability density measure. Inspired by the idea in dealing with nominal attributes, we combine the sample mean and sample standard

deviation of $P(numerical_i|c)$ in the leaf Naïve Bayes , denoted by Mean_{local} and Std_{local}, and that in the global Naïve Bayes, denoted by Mean_{global} and Std_{global}, with convex combination. That is:

$$\begin{aligned} \text{Mean}_i &= (1-w)\cdot \text{Mean}_{local} + w \cdot \text{Mean}_{global} \\ \text{Std}_i &= (1-w)\cdot \text{Std}_{local} + w \cdot \text{Std}_{global} \end{aligned} \tag{2}$$

where w is the varied weight of the prior knowledge similar to the case of nominal attribute. Finally, the normal distribution of the ith attribute given the class c in the leaf node is determined by Mean_i and Std_i.

3.4 Using Weighted Naïve Bayes

Attributes independent and equally important are the assumptions of Naïve Bayes, but these assumptions will not always hold in real data especially in high dimensional data. In fact, many research focus on breaking these assumptions to improve the performance of Naïve Bayes, like Bayesian network and weighted Naïve Bayes. Bayesian networks focus on breaking the attribute independence assumption, but it still supposes attributes are equally important. On the other hand, weighted Naïve Bayes focuses on breaking attributes equally important and keep the independent assumption. Here, we will describe how weighted Naïve Bayes can break the equally important assumption. The original Naïve Bayes classifier is defined as follows:

$$\text{NB}(x) = \arg\max_c P(c)\prod_{i=1}^{n} P(a_i|c),\ x = (a_1, a_2, \cdots, a_n),\ c : class \tag{3}$$

If we increase the degree of $P(a_i|c)$ in Naïve Bayes classifier, that will enlarge the influence of the attribute a_i(see Table 1). Thus, we can extend Naïve Bayes to weighted Naïve Bayes as follows [10]:

$$\text{WNB}(x) = \arg\max_c P(c)\prod_{i=1}^{n} P(a_i|c)^{w_i},\ x = (a_1, a_2, \cdots, a_n),\ c : class \tag{4}$$

In the weighted Naïve Bayes classifier, we break the equally important assumption by using different weights.

Decision trees are usually built by the subset of all attributes. Intuitively, only using the attributes on the tree in leaf Naïve Bayes should be more representative. In fact, removing redundant attributes will be good for the performance of Naïve Bayes. In other words, only using the attributes on the tree is similar to the attribute selection. However, only using the attributes on the tree will easily produce many duplicate points in pure nominal datasets, especially with a high dimensional dataset and a small tree. We have mentioned this phenomenon in section 2. In order to overcome the duplicate points problem, we use all attributes to build up the leaf Naïve Bayes but give tree attributes more weight. The attribute weighting is closely related to attribute selection. Thus, we also can emphasize the importance of tree attributes by using attribute weighting.

	Degree 1	Degree 2	Degree 3
$P(a_i\|c_1)$	0.5	0.25	0.125
$P(a_i\|c_2)$	0.1	0.01	0.001
difference	5 times	25 times	125 times

Table 1. The effect of different degrees of $P(a_i|c)$

4 Numerical Results and Comparisons

Using accuracy in evaluating models will completely ignore probability estimations produced by learning models. As the purpose is probability estimations or ranking, accuracy is not sufficient in measuring and comparing learning models. The area under the ROC (Receiver Operating Characteristics) curve, or simply AUC, has been shown as an measurement for the quality of ranking [3, 11, 12]. The ROC curve was first used in signal detection theory to represent the tradeoff between the hit rates and false alarm rates [6, 9]. It has been extensively studied and applied in medical diagnosis since 1970s [14, 20]. The AUC can be expressed in a simpler form: if the sample contains m positives and n negative examples, we can denote AUC simply by the following Wilcoxon-Mann-Whitney statistic [4]:

$$AUC = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} I_{P(x_i)>P(x_j)} + \frac{1}{2} I_{P(x_i)=P(x_j)}}{mn} \tag{5}$$

where $P(x)$ is the probability estimation from learning models and I_π is defined to 1 if the predicate π holds and 0 otherwise.

In our experiment, we evaluate our proposed method and compare with other methods on 7 two-class datasets from the UCI repository. The datasets are described in Table 2. In Lymphography dataset, we throw away the 4 instances to reduce the classes. In numerical results, we repeat 5-fold cross validation on each dataset 5 times and report the mean of them.

4.1 Comparing Our Proposed Method with Other Methods

In this section, we will compare our proposed method with C4.4 [17], C4.5, and C4.5-L. The C4.5 is the most popular decision trees algorithm which incorporates with some pruning strategy for a better classification result. While the C4.4 does not prune decision trees, it might cause the over-fitting problem but will give a better ranking result. Another difference between the C4.5 and the C4.4 is that the C4.4 ranks the instances with Laplace smoothing and C4.5 does not. In our experiment the C4.5-L denotes C4.5 with Laplace smoothing. Table 3 shows the numerical results of them. The number before the slash is the AUC score and the other number is the quantity $\delta = \frac{A_1 - A_2}{1 - A_2}$ [5], where A_1 and A_2 are the

Dataset	Size	Nom. Attributes	Num. Attributes
Tic Tac Toe	958	9	0
House Vote	435	16	0
SPECT	267	22	0
Bupa	345	0	6
Ionosphere	351	0	34
Sonar	208	0	60
Lymphography	142	15	3

Table 2. Descriptions of datasets

AUC scores of $Method_1$ and $Method_2$ respectively. The performance metric δ indicates what percentage of $Method_2$'s missing ROC area $(1 - A_2)$ is covered by $Method_1$. In Table 3, we use C4.5 as the counterpart method ($Method_2$) and compute the δ values of using C4.4, C4.5 and our proposed method as the $Method_1$.

As we mentioned above, we need to determine two parameters, weights of prior knowledge and weights of attributes, in our method. The details of tuning heuristics and procedures will be discussed in Section 4.2. In Table 3, we use the optimal parameters followed by Section 4.2, and we can observe that C4.4 is better than C4.5 and C4.5-L. Therefore, we use a paired t-test on our proposed method and C4.4 with a 95% confidence coefficient. The test result shows our method is significantly better than C4.4 in AUC scores, except the dataset Tic Tac Toe (see Table 4). This shows that our method can significantly improve the ranking of decision trees. In fact, our proposed method is C4.5 with smoothing techniques and leaf weighted Naïve Bayes. In other words, we not only can rank the leaf nodes with smoothing techniques but also have the ability to rank the instances in the leaf nodes. Moreover, we also can show the necessity for ranking instances in the leaf nodes because instances with equal probability estimations in the same leaf node will be treated as random ranking within the leaf node (see Fig. 1).

4.2 Parameters Tuning

In order to increase the reliability, we take the information from the global Naïve Bayes into account when estimating $P(a_i|c)$ in a leaf Naïve Bayes. From the idea, there is a point of view to regard for the weight of the global Naïve Bayes. If there are enough instances in the leaf nodes, the weight of prior knowledge should be less. Therefore, we will take a varied weight for each leaf node when adjusting $P(a_i|c)$ via the m-estimate smoothing technique and convex combination. We

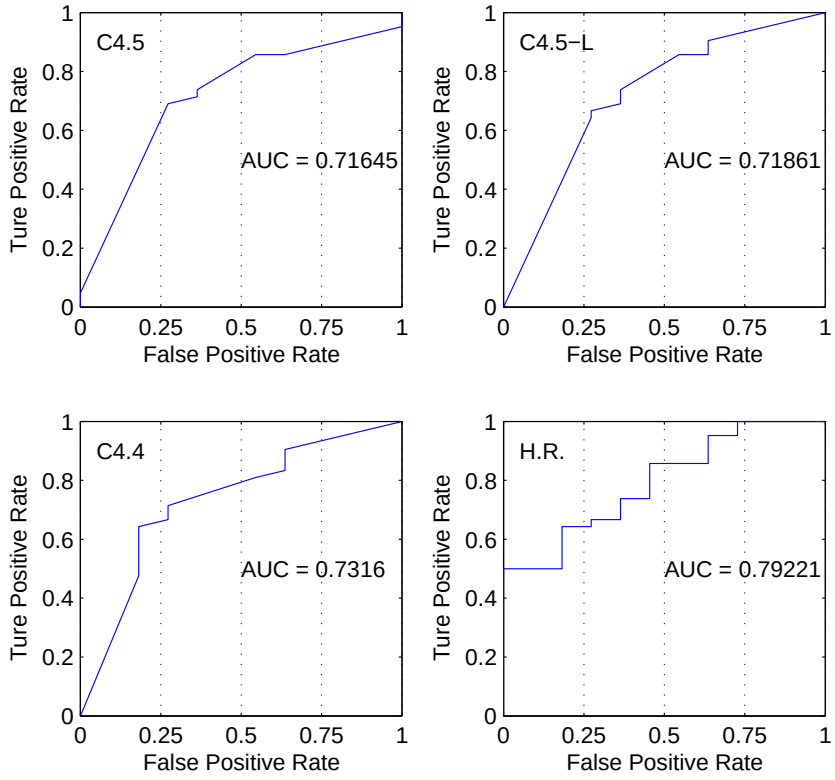


Fig. 1. ROC curves of C4.5 (left top), C4.5-L (right top), C4.4 (left bottom), and our proposed method (right bottom) of a particular fold of SPECT dataset.

define our varied weight to achieve our idea as follows:

$$w_i = 1 - \alpha_i, \text{ where } \alpha_i = \frac{\text{number of instances in the leaf node } i}{\text{number of the entire training instances}} \tag{6}$$

Table 5 shows the results. In the second and third columns, we fixed $w_i = 0$ and $w_i = 1$ for all leaf node i. They represent only using the leaf Naïve Bayes and the global Naïve Bayes respectively. Obviously, the varied $w_i = 1 - \alpha_i$ is more appropriate of them. The result also shows that combining the leaf Naïve Bayes and global Naïve Bayes is good for ranking instances in the leaf node.

As we mentioned in Section 3, removing redundant attributes will be good for the performance of Naïve Bayes. Unfortunately, removing redundant attributes may easily produce duplicate instances in pure nominal datasets. In order to

Dataset	C4.5	C4.5-L	C4.4	Hierarchical R.
Tic Tac Toe	0.8896	0.9050 / 13.9%	**0.9281 / 34.8%**	0.9111 / 19.5%
House Vote	0.9603	0.9706 / 25.9%	0.9746 / 36.0%	**0.9885 / 71.0%**
SPECT	0.7769	0.7978 / 9.4%	0.7782 / 0.5%	**0.8433 / 29.8%**
Bupa	0.6743	0.6884 / 4.3%	0.6942 / 6.1%	**0.7045 / 9.3%**
Ionosphere	0.8922	0.9326 / 37.5%	0.9314 / 36.3%	**0.9526 / 51.6%**
Sonar	0.7322	0.7836 / 19.2%	0.7836 / 19.2%	**0.8137 / 30.4%**
Lymphography	0.8263	0.8471 / 12.0%	0.8554 / 16.8%	**0.8877 / 35.4%**

Table 3. Numerical results of C4.5, C4.5-L, C4.4, and our proposed hierarchical ranking method on seven public available datasets in UCI repository

Dataset	p-value
Tic Tac Toe	0.9996
House Vote	0.0108
SPECT	0.0434
Bupa	0.0287
Ionosphere	0.0125
Sonar	0.0238
Lymphography	0.0193

Table 4. The p values of the paired t-test on our proposed method and C4.4

overcome this problem, we use all attributes but decrease the weight of non-tree attributes in leaf weighted Naïve Bayes. Table 6 shows the results of different weights on non-tree attributes. We fix the weight of tree attributes and vary the weight, w, of non-tree attributes to 0, 0.01, 0.1, 0.5, and 1. $w = 0$ and $w = 1$ means only using tree attributes and all attributes with equal weights respectively. $w = 0.5$ had the best performance in our experiment. It shows that our strategy of weighting is good for the performance. Note that we have better improvement in two pure nominal datasets, House vote and SPECT, by comparing with $w = 0$. In our observation, only using tree attributes will result in producing some equal probability estimations in the leaf node because of dimension reduction. It means that using weighted Naïve Bayes will not only raise the ability of Naïve Bayes, but also solve the problem of dimension reduction.

Dataset	$w = 0$	$w = 1$	$w = 1 - \alpha$
Tic Tac Toe	0.9110	0.9079	0.9111
House Vote	0.9792	0.9885	0.9885
SPECT	0.8286	0.8437	0.8433
Bupa	0.7020	0.7042	0.7045
Ionosphere	0.9441	0.9489	0.9526
Sonar	0.8046	0.8135	0.8137
Lymphography	0.8770	0.8902	0.8877

Table 5. The effect of different prior knowledge weights in estimating $P(a_i|c)$, where α is defined by (6). In this table, we use the same weight 0.5 of non-tree attributes for all.

Dataset	$w = 0$	$w = 0.01$	$w = 0.1$	$w = 0.5$	$w = 1$
House Vote	0.9790	0.9883	0.9881	0.9885	0.9880
SPECT	0.8310	0.8336	0.8416	0.8433	0.8403
Ionosphere	0.9471	0.9497	0.9507	0.9526	0.9518
Sonar	0.8089	0.8096	0.8113	0.8137	0.8140
Lymphography	0.8846	0.8882	0.8882	0.8877	0.8769

Table 6. Different weights of non-tree attributes. In this table, we use the same varied weights of prior knowledge for all.

5 Conclusion and Future Work

In this paper, we present a hierarchical ranking method for decision trees. We rank the rules as well as the instances fit in with the rules. This method combines decision trees and Naïve Bayes via embedding leaf weighted Naïve Bayes in each leaf node. The hierarchical ranking strategy will retain the intelligibility of decision trees. Another important feature of our method is that it can deal with hybrid datasets as well. Experiment results show that our proposed method improves the AUC score over other methods. It means that ranking instances in the leaf node works in improving the ranking performance.

Our method needs to set two parameters, weights of prior knowledge and attributes. For the weight of attributes, we simply assign two different weights for tree attributes and non-tree attributes in our experiments. In fact, how to determine the weight of weighted Naïve Bayes is an interesting issue [8]. In our future work, we will study how to determine different weights for each attribute and take the depth of the split attribute in decision trees into account.

References

1. I. Alvarez and S. Bernard. Ranking cases with decision trees: a geometric method that preserves intelligibility. In *Proceedings of 18th Internationa Conference on Artificial Intelligence(IJCAI-2005)*, 2005.
2. C. Blake and C. Merz. Uci repository of machine learning databases. *niversity of California (http://www.ics.uci.edu/∼mlearn/MLRepository.html)*, 1998.
3. A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30:1145–1159, 1997.
4. C. Cortes and M. Mohri. Auc optimization vs. error rate minimization. In *Advances in Neural Information Processing Systems(NIPS 2003)*, 2004.
5. D. Dash and G. F. Cooper. Model averaging for prediction with discrete bayesian networks. *Journal of Machine Learning Research*, 5:1177–1203, 2004.
6. J. Egan. Signal detection theory and roc analysis. *New York:Academic Pres*, 1975.
7. C. Ferri, P. Flach, and J. Hernández-Orallo. Improving the auc of probabilistic estimators trees. In *Proceedings of 14th European Conference on Machine Learning(ECML'2003)*, pages 121–132, 2003.
8. T. Gärtner and P. A. Flach. Wbcsvm: Weighted bayesian classification based on support vector machines. In *Proceedings of the 18th International Conference on Machine Learning*, pages 207–209, 2001.
9. D. Green and J. Swets. Signal detection theory and psychophysics. *New York:Wiley*, 1966.
10. S. Sheng H. Zhang. Learning weighted naive bayes with accurate ranking. In *Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 567–570, 2004.
11. D. J. Hand and R. J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45:171–186, 2001.
12. C. X. Ling J., Hunag, and H. Zhang. Auc: a astatistically consistent and more discriminating measure than accuracy. In *Proceedings of 18th Internationa Conference on Artificial Intelligence(IJCAI-2003)*, 2003.
13. C. X. Ling and J. Yan. Decision tree with better ranking. In *Proceedings of 2003 International Conference on Machine Learning (ICML'2003)*, pages 480–487, 2003.
14. C. Metz. Basic principles of roc analysis. *Seminars in NuclearMedcine*, 8:283–298, 1978.
15. J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. *in Advances in Large Margin Classifiers, A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA: MIT Press*, 2000.
16. F. Provost and P. Domingos. Well-trained pets: Improving probability estimation trees. *Technical Report CDER #00-04-IS*, 2000.
17. F. Provost and P. Domingos. Tree induction for probability-based rankings. *Machine Learning*, 52:199–215, 2003.
18. F. Provost, T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the fifteenth international conference on machine learning*, pages 445–453, 1998.
19. J. R. Quinlan. Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.
20. J. Swets. Measuring the accuracy of diagnostic systems. *Science*, 240:1285–1293, 1988.

21. I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Francisco, CA, 2000. http://www.cs.waikato.ac.nz/∼ml/index.html.
22. B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proceedings of 18th International Conference on Machine Learning*, pages 609–616, 2001.

A Research on Data Mining Techniques Based on Ant Theory for Path-Type Association Rules

Nai-Chieh Wei[1], Hao-Tien Liu[1], and Yang Wu[2]

[1] Department of Industrial Engineering and Management, I-Shou University
No. 1, Sec. 1, Syuecheng Rd., Dashu Township, Kaohsiung County 840, Taiwan, R.O.C.
ncwei@isu.edu.tw
[2] Far East College
No. 49, Chung Hwa Rd., Hsin-Shih, Tainan County 744, Taiwan, R.O.C.
sflin@cc.fec.edu.tw

Abstract. In an increasingly diverse market, customer behaviors should not be ignored in order to get more competitive advantages than other competitors. Therefore, a useful pattern in effectively extracting database is needed for companies to discover practical knowledge of decision making. The presented study is to develop a combination method of data mining by integrating Principal Component Analysis, Association Rules, and Ant Theory. Based on the floor layout and sales data, the integrated method is capable of finding the effective path-type product association rules, and overcoming the drawbacks of data overloading which is caused mainly by more than two products used in generating a large number of association rules. The path-type association rules may be indicated as purchase rules utilized to discover the customer's shopping paths. Research results show that the integrated method can effectively analyze knowledge of purchase rules hidden in the database for decision makings of future promotion campaigns, product selections, and product displays in the store. Finally, the sales data of a large chain store is taken as a case study, and a comparison between the data mining software, XpertRule Miner, and the integrated method is made.

1. Introduction

Companies are bound to have better chances to be successful as long as they are capable of discovering in the database information that is unknown in the past, or obtaining more information than their competitors in the competitive environment (Sung and Sang 1998). Data mining techniques, therefore, play a significant role in accomplishing this goal and have become a popular topic in numerous articles in recent years, including Association Rules, Decision Tree, and Genetic Algorithms etc. (Fayyad 1996, 1997). As one of the appealing data mining techniques, Association Rules describe the possibility of what products that would be purchased simultaneously by customers. Hence, the Association Rules of product purchasing can explain the reasons behind a customer purchasing product X and product Y at the same time. Would it be possible that the concept of Association Rules mentioned above match the viewpoint of path finding probability in Ant Theory?

Ant Theory was proposed by Marco Dorigo (1996), observed the actual behavior of ants searching for paths and developed the named theory. Ants are able to locate the shortest path between formicary and food. Instead of vision, they use a type of secretion, Pheromone, left behind on their passageways to help deliver messages to other ants during food hunting for finding the paths to the food location. So as time goes by, and as more and more ants take the same path to food, more Pheromone is deposited on the path, and in turn ants would have higher frequency in selecting the direction with higher concentration of Pheromone. Hence, ants will gradually take the same path between formicary and food location. Similarly, if several items of products present higher frequency in sales data, that means these paths may be discovered and used often on the sales floor by consumers; therefore, the probability that these products are chosen at the same time is relatively high. Although Ant Theory and its extended researches have been applied to the Traveling Salesman Problem (Dorigo, Maniezzo, and Colorni, 1996), vehicle routing, and scheduling, there is still lack of information about the application of Ant Theory on data mining and path finding probability of sales floor.

The research as presented here is going to use fewer variables to explain more variations in the data. It might help to overcome the interferences caused by too many variables while extracting information in sales data in the past to avoid producing unreliable Association Rules. Further, the main purpose of the research is to apply Ant Theory for finding out the probability of path selection and identifying the paths customers take among the product sections. These shopping paths, corresponding to product purchasing, are discover by the purchase rules of path-type association. Within these obtained paths, decision makers can identify the hot paths which can be used in looking for the useful information hidden in the sales data effectively.

2. Research Method

Important research steps are as follows:

Step 1 Principal Component Analysis

Product items are categorized firstly by Principal Component Analysis (PCA) in this step. PCA is a statistical technique proposed by Pearson in 1901, and then further developed by Hotelling in 1933. PCA is based on the concept that has fewer variables to interpret more variations in the original data to analyze all product items (variations) in order to eliminate some unnecessary items or to categorize them. Take sales data for instance. It can be found in Association Rules that the first rule can encompass the following few products (A, B, D) while the second rule can cover the following ones (A, B, C, D). It is possible that item D is purchased after items A and B have been purchased. It is also possible that item D is purchased after item A has been purchased; therefore, item B is redundant information in the rule, or else can be used for item (variation) categorization. Hence, this research performs PCA to categorize some product items (variations) and this makes categorized items (variations) more significant.

The biggest problem with PCA is that the unit of each variable is not quite the same. Therefore, variables are to be standardized (to render the new average value 0 and the standard deviation 1) when they are different units.

For example, there are p($x_1 \ldots x_p$) original variables and up to p principal components ($y_1 \ldots y_p$), and the j-th principal component may be expressed as follows:

$$y_j = PC_j = a_{j1}x_1 + \ldots + a_{jp}x_p = \vec{a}'_j \vec{x} \quad (1)$$

The number of variables in each principal component is equal to its corresponding eigenvalue λ, and can be represented as:

$$Var(Y_j) = Var(\vec{a}'_j \vec{x}) = \vec{a}' \sum \vec{a}_j = \vec{a}'_j (\lambda_j \vec{a}_j) = \lambda_j (\vec{a}'_j \vec{a}_j) = \lambda_j \quad (2)$$

and the factor loading of the j-th principal component is:

$$f_j = \frac{y_j}{\sqrt{\lambda_j}} = \frac{a_{j1}x_1 + \ldots + a_{jp}x_p}{\sqrt{\lambda_j}} = l_{j1}x_1 + \ldots + l_{jp}x_p \quad (3)$$

Moreover, in order to make factor loading easier to be interpreted, orthogonal rotations is also used to calculate the loading of each variable. In orthogonal rotations, there are no relations between each factor; that is, its correlation is equal to 0, and the angle between the axes is 90 degrees. The revolution axes are used to adjust the scale of factor loading based on the association between the variables and the factor.

Step 2 Sectional Distance

In Step 2, a store floor plan of each section is drawn up, a center point of each section is used to represent the entire area, and the rectilinear distance between two sections is used to represent the distance. The formula is as follows:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (4)$$

Step 3 Parameter Settings

The parameter hypothesis of α and β is absolutely important in Ant Theory because appropriate parameters can generate optimal solutions in a shorter time frame. It was mentioned in the research of Dorigo et al. (1997) that the ant chooses the location with the shortest distance for the next visit if $\alpha = 0$; and this corresponds to a classical random Greedy Algorithm. On the contrary, if $\beta = 0$, the ant chooses the next visit based solely on the influence of Pheromone, regardless of the distance. This technique will lead to stagnation in the solution generating process; in other words, all ants are

very likely to take the same traveling route. Therefore, α and β must be taken into consideration that they are generally set bigger and not equal to zero.

Step 4 Path Selection Probability

After Step 1 and Step 2 are completed, the formulae of ant's path selection probability in Ant Theory can then be used to find out the probability of occurrence between sections. The ant will then use probability function to select the next section. Therefore, in the t-th iteration, the function representing the probability of the k-th ant choosing to go from section i to section j can be expressed as follows:

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in allowed_k}[\tau_{il}(t)]^{\alpha}[\eta_{il}]^{\beta}}, & if \quad j \in allowed_k \\ 0 \qquad\qquad , otherwise \end{cases} \tag{5}$$

Where

$p_{ij}^{k}(t)$: represents the probability function for the k-th ant to choose to go to section j from section i.

τ_{ij}: represents the Pheromone concentration left between section i and section j.

η_{ij}: represents the visibility between section i and section j.

$allowed_k$: represents the set of sections for k-th ant to choose as the next section when in section i (namely the sections which k-th ant has not yet visited).

α and β: represent parameters that control the relative influences between $\tau_{ij}(t)$ and η_{ij} respectively. α and β are both ≥ 0, and larger α symbolizes preferences of selecting paths based on the magnitude of $\tau_{ij}(t)$; while larger β represents preferences of selecting paths based on the magnitude of η_{ij}.

t: represents the number of iteration.

It is known from the formulae shown above that the probability of the ant selecting the next section to visit is based on the Pheromone left between the two sections: τ_{ij} and visibility, η_{ij}, in which the visibility between sections in each iteration is the same. However, the concentration of Pheromone renews after all ants complete the journey (one iteration) and is used as a reference for ants to choose their paths. The Pheromone concentration renewal formula is as follows:

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij} \tag{6}$$

Where

$\tau_{ij}(t)$: represents the Pheromone concentration left between section i and j in the t-th iteration, and the general assumption is that the initial value is $\tau_{ij}(0) = c$ where c is a constant.

ρ : represents the residual coefficient of Pheromone ($1-\rho$: the evaporation coefficient of pheromone), where the value of ρ is voluntarily set between [0, 1].

$\Delta\tau_{ij}$: represents the sum of the Pheromone concentration left between section i and j by all ants and may also be expressed as

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{7}$$

Where $\Delta\tau_{ij}^{k}$ is the Pheromone concentration left by the k-th ant in sections i and j, and may be expressed as

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k} & , \ if\,(i,j) \in tour \quad done \quad by \quad ant \quad k \\ 0 & , \ otherwise \end{cases} \tag{8}$$

Where Q represents the Pheromone concentration secreted by each ant and is a constant, and L_k is the total distance between all sections taken by the k-th ant in one iteration. And this research hypothesizes that the Pheromone concentration on the initial path is 0 (namely $\tau_{ij}(0) = 0$).

Step 5 Result Analysis

At the end, the results obtained by means of data mining software of XpertRule Miner (1999) will be used to check the feasibility of integrating Ant Theory with data mining techniques proposed by this research to discover the rules.

3. Case Study

This study firstly uses actual sales data from a domestic wholesaler in Kaohsiung, a typical wholesaler in Taiwan and takes 350 sales transactions from the database. It targets 35 food product items in the store and uses PCA to study each item to eliminate interference factors among them. PCA removes unnecessary product items or groups them differently, and then determines the sections of all product items in each group. Furthermore, this study applies the path selection probability, concept of Ant Theory to determine the probability and the paths of customers in all sections. Finally, this study establishes possible purchase rules corresponding to product items.

In order to simplify calculation, this study only considers the path taken by customer starting from the store entrance, onto all sections to select products, and finally to the store exit. It temporarily leaves out the consideration of going to the cashier before leaving. Furthermore, in the process of calculating the path selection probability in Ant Theory, the fixed number of ants (k), starting point, and end condition are also pre-determined; that is, the calculation is stopped when certain iteration is reached, and the results obtained from such condition is used for analysis.

Step 1 Principal Component Analysis

SPSS software is applied to conduct PCA, and the categorization is eventually done based on factor loading, as shown in Table 1.

Table 1. Categorization results

Group	Products	Group	Products
1	Marinated salty melon, Gluten	9	Mini sushi, Canine dry food, One-liter ice creams, Yogurt
2	Naturally fermented soy sauce, Olive oil, Cheese slices	10	Deep-fried products
3	Instant noodles, Eel, Other crisps, Leafy greens	11	Sport beverages, Cakes
4	Hot pot dumplings, Grade A pork	12	Toast
5	Sandwiched cookies, Domestic cigarettes, Instant coffee	13	Tropical fruits, Sliced fish
6	Cold salads, Stews, Milk	14	Rice crackers
7	Baked goods, Dumpling, Dried peaches and plums	15	Kaoliang, Tofu
8	Instant fish and pork jerky, Noodles		

Step 2 Sectional Distance

In Step 2, a section layout of the store floor is drawn up, as shown in Figure 1, a center point of each section is used to represent the entire area, and the rectilinear distance between two sections is used to represent the distance.

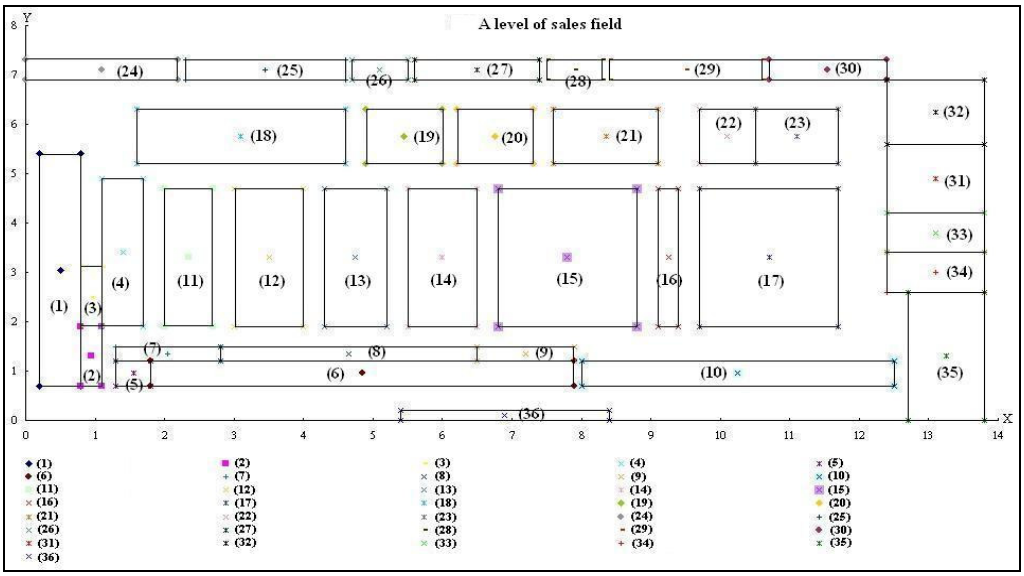


Figure 1. A section layout of the store floor

Moreover, the sections covered by each group are shown in Table 2.

Table 2. Sections covered by each group

Group	Covered Sections	Group	Covered Sections
1	(15) Canned food, Seasonings, Miscellaneous section	9	(26) Sushi section (34) Pet section (17) Frozen and refrigerated food section (31) Dairy products section
2	(15) Canned food, Seasonings, Miscellaneous section, (33) Cheese section	10	(27) Ready-to-go food section
3	(09) Instant noodles section (32) Fish products (12) Domestic biscuits (20) Vegetables section	11	(13) Beverages section (24) Freshly-made cakes and bread section
4	(17) Frozen and refrigerated food section, (22) Meat product section	12	(24) Freshly- made cakes and bread section
5	(12) Domestic biscuits section (04) Cigarettes and alcohol section (11) Coffee and milk powder section	13	(19) Fruits section (23) Seafood section

6	(21) Restaurant section (28) Grill and marinated food section (31) Dairy products section	14	(14) Imported products section
7	(28) Grill and marinated food section (17) Frozen and refrigerated food section (19) Fruits section	15	(04) Cigarettes and alcohol section (17) Frozen and refrigerated food Section
8	(25) Dry products section (09) Instant noodles section		

Step 3 Parameter Settings

The parameter combination found in Dorigo's study (1996) is perhaps only suitable under specific condition. Therefore, the models that match the conditions in this study are established firstly (that is, set the start and stop conditions), and 5 cities with coordinates as (1, 2), (3, 4), (5, 1), (8, 5) and (6, 3) are set to test the results of few parameter combinations (fix ρ = 0.5 and disregard the influence of parameter Q because it is a constant and its effect can be ignored. Set Q = 100). The results are shown in Table 3. It is discovered that when α = 0 or β = 0, the obtained distance is comparatively longer. And when α and β values are considered together, and α = 1 and β = 4, the acquired distance is more favorable (shorter) than when α = 1 and β = 5. And when the quantity of ants is set to be 10, the obtained distance is more favorable (shorter) than when the number of ants is set to be 5. Therefore, this research determines to adopt the parameter combination of α = 1, β = 4, and k = 10 to be the parameter settings for all groups.

Table 3. Some of parameter combination tests

(α, β)	k=5	k=10	(α, β)	k=5	k=10	(α, β)	k=5	k=10
(0,4)	20.843	20.843	(4,0)	21.335	24.843	(1,4)	19.114	17.942
(0,5)	20.843	20.843	(5,0)	21.335	24.843	(1,5)	20.843	20.843

Step 4 Path Selection Probability

First, deciding on the number of ant (k=10) and the stop condition (T=5), the fixed start point is the entrance and all customers must enter by that entrance and exit by the same location after completing the entire shopping process. The path selection probability parameter in Ant Theory is used to determine probability of selecting path in each section and path distance when stop condition is reached. This study assumes the Pheromone concentration on the initial path to be 0 (that is $\tau_{ij}(0) = 0$). Therefore, the results obtained from 5 round trips with continuously updated

Pheromone concentration (that is, reach the pre-determined t = 5 stop conditions), are shown in Table 4:

Table 4. Results obtained at the stop conditions

Group	Path (Section) Selection	Distance	Customer Purchase Rules
2	(36) => (15) => (33) => (36)	19.800	Entrance => (Naturally-fermented soy sauce, Olive oil) => Cheese slices => Exit
3	(36) => (9) => (20) => (12) => (32) => (36)	37.000	Entrance => Instant noodles => Leafy greens => Other crisps => Eel => Exit
4	(36) => (17) => (22) => (36)	18.900	Entrance => Hot pot dumplings => Grade A pork => Exit
5	(36) => (12) => (11) => (4) => (36)	17.600	Entrance => Sandwiched cookies => Instant coffee => Domestic cigarettes => Exit
6	(36) => (21) => (28) => (31) => (36)	27.300	Entrance => Cold salads => Stews => Milk => Exit
7	(36) => (17) => (28) => (19) => (36)	24.500	Entrance => Dumplings => Baked products => Dried peaches and plums => Exit
8	(36) => (9) => (25) => (36)	21.500	Entrance => Noodles => Instant fish and pork jerky => Exit
9	(36) => (17) => (34) => (31) => (26) => (36)	30.600	Entrance => One-liter ice creams => Dry dog food => Yogurt => Mini sushi => Exit
11	(36) => (13) => (24) => (36)	25.600	Entrance => Sport beverages => Cakes => Exit
13	(36) => (19) => (23) => (36)	22.600	Entrance => Tropical fruits => Sliced fish => Exit
15	(36) => (17) => (4) => (36)	25.200	Entrance => Tofu => Kaoliang => Exit

The customer purchase rules shown in Table 4 can be described as hot paths that the customers select at a high frequency. These hot paths, corresponding to products, can be identified as the base of possible purchase rules; thus, decision makers can look for the useful information hidden in the sales data effectively in order to use as references for making more appropriate and efficient decisions.

The aforementioned shows the purchase rules are constituted by covering three more sections. On the other hand, the simpler purchase rules are established by covering three or fewer sections, and the results are shown in the following table:

Table 5. Results that cover three sections or less

Group	Path (Section) Selection	Customer Purchase Rules
1	(36) => (15) => (36)	Entrance => (Marinated salty melon, gluten) => Exit
10	(36) => (27) => (36)	Entrance => Deep-fried products => Exit
12	(36) => (24) => (36)	Entrance => Toast => Exit
14	(36) => (14) => (36)	Entrance => Rice crackers => Exit

Step 5 Result Analysis

The results show that the integrated data mining method established in this study is easier to be understood by using fewer rules than XpertRule Miner. The integrated method is capable of providing information for finding out the paths customers select among the product sections. On the other hand, compared with the integrated method, the scope of Association Rules discovered by XpertRule Miner is wider and more difficult to be interpreted, since some irrelevant information is easily generated under the influence of the data contents (such as the samples in this study). The comparison between the integrated data mining method and XpertRule Miner is shown in Table 6:

Table 6. Result comparisons

	Integrated Data Mining Model	XpertRule Miner
Rules Formation	First find out the combination of products that might be purchased together. Then use Ant Theory in each group and section to discover the path taken between sections when stop condition is reached. After that, establish customer purchase rules corresponding back to product items.	Find out all conditions satisfying the minimum support and confidence to establish the rules.
Rules Pattern	Find out the path-type rules formed by several items (A=>B=>C=>...=>A) and the product items amongst these rules are not likely to be unique.	Find out the rules formed by the two items ($A \cap B$) and the product items included in the rules are unique.
Summary	Fewer rules, therefore easier to understand. Decision makers can use this data as reference in product display or promotion combinations in the future.	The scope of rules is wider and it is possible that some irrelevant information is generated under the influence of the data contents. That is, the drawback of Association Rule is that the rules of lower occurring rate are left out when

		higher thresholds are set for the minimum support and confidence.

4. Conclusions and Recommendations

The integrated data mining method used to discover path-type product association rules in sales data is developed in this study. Ant theory is applied to an extended area to identify the path type rules for sales management. The resulted path-type rules can be viewed as hot paths, and each of them might indicate higher frequency of customer movement path on the floor. So, if store management could utilize this knowledge and make decisions on the adjustment of product mix, location, promotion, and advertising, it might not only increase the chance of product exposed to customer resulting in higher sales but also improve the efficiency of customer flow. However, due to a lack of detailed analysis in certain aspects of this research, the following can serve as future research focus:

1. The customer quantity of flow in all covered sections will need to be analyzed and critical information of improving store product display also needs to be provided. Then it is desirable to adapt the developed method for comparing conditions before and after the improvement to see whether the satisfaction level is increased in these sections.
2. The shaping up of product association rules itself can be categorized either as symmetric or transitive. The former describes the possibility of customer purchases product B after purchasing product A, whereas the latter depicts the chance of customer purchasing product C after making the purchases of items A and B. This aspect of the study is also worth further investigation in the future.

Acknowledgement

This research is partially supported by National Science Council of Taiwan, R. O. C. (NSC 94-2622-E-214-014-CC3).

References

1. Attar Software Limited, XpertRule Miner, 1999, pp1-154.
2. Berry, M. J. A., and Linoff, G. S., Data Mining Techniques: For Marketing, Sales and Customer Support, John Wiley & Sons, Inc., 1997.
3. Dorigo, M., Maniezzo, V., and Colorni, A. , The Ant System: Optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics–Part B, Vol. 26, No. 1, 1996, pp.1-13.

4. Dorigo, M. and Gambardella, L. M., Ant colony system: A cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1, 1997, pp. 53-66.
5. Fayyad, Data Mining and Knowledge Discovery：Making sense out of data, IEEE Expert, Vol. 11, No. 5, 1996, pp 20-25.
6. Fayyad, U., Piatetsky-Shapiro, G., Padhraic, S., From Data Mining to Knowledge Discovery in Databases, AI magazine, 1996, pp. 37-54.
7. Fayyad, U., Piatesky-Shapiro, G. and Smyth, P., and Uthurusamy, R. Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1996.
8. Fayyad, U., Stolorz, P., Data mining and KDD: Promise and Challenges, Future Generation Computer Systems, Vol. 13, No. 2-3, 1997, pp. 99-115.
9. Sung, H. H. and Sang, C. P., Application of data mining tools to hotel data mart on the intranet for database marking, Expert System with Applications, 15, pp. 1-31.

Effective Clustering of High-Dimensional Data

Emin Erkan Korkmaz*[1], Reda Alhajj[2 3], and Ken Barker[3]

[1] Department of Computer Engineering, Yeditepe University, İstanbul, Turkey
ekorkmaz@cse.yeditepe.edu.tr
[2] Department of Computer Science, Global University, Beirut, Lebanon
[3] Department of Computer Science, University of Calgary, Calgary, Alberta, Canada
{alhajj,barker}@cpsc.ucalgary.ca

Abstract. Genetic algorithms (GAs) have been successfully integrated into the clustering process. However, GAs are criticized for being slow. So, this paper presents a novel clustering approach to overcome the difficulty of using GA for clustering large datasets. When the GA is applied to the clustering problem, the length of the chromosomes used in the search is in general determined by the number of instances that will be clustered. Hence, GA performance decreases dramatically with large datasets. So, the contribution of this paper is a special preprocessing step to decrease the number of instances that would go through the clustering phase. This facilitates for more effective clustering of high dimensional data. The performance of the proposed approach has been tested using the *Dermatology Database*; and it has been observed that the dataset can be clustered accurately with the new approach; the reported results demonstrate the effectiveness of the pre-processing step.

Keywords: classification, clustering, genetic algorithms, pre-processing, data mining.

1 Introduction

Clustering can be described as the grouping of a set of data cases or instances so that instances that fall into the same cluster are more similar to each other than to instances of other clusters. Clustering is different from classification in the sense that clustering is unsupervised process while classification is supervised. Unsupervised clustering of data has received the attention of many researchers, and the developed techniques in the area can be used effectively in a variety of disciplines, ranging from image processing and knowledge discovery to market research and molecular biology [1]. They may be classified into different categories, including hierarchical clustering, grid-based methods, partitioning and methods based on co-occurrence of categorical data [1].

Hierarchical clustering builds a tree of clusters, namely a hierarchy where sibling clusters partition the elements contained by their common parent [2].

* Emin Erkan Korkmaz is partially supported by TUBITAK (The Scientific and Technological Research Council of Turkey) under grant number 105E027.

Partitioning methods form another approach where algorithms are used to divide data into subsets. Checking all possible subset organizations would have an exponential complexity. Hence, the key point is to use a heuristic or an optimization method to find a good partitioning. Different alternatives exist for modeling the partitioning problem. In probabilistic clustering, data is assumed to be a sample independently drawn from several probability distributions [3]. Approaches based on the definition of an objective function, also exist. The objective function is to be used in searching for the best partition. K-means [4] is one of the well known examples of this approach. It is also possible to partition the data based on the notion of density [5].

Other clustering approaches involve utilizing genetic algorithms (GA), which have been successfully used to solve many optimization and search problems [6]; different techniques have been proposed for using GA to search for the optimal clustering of a given data set. However, existing techniques have some drawbacks, and redundancy seems to be a problem for the representations used [7]. Ensuring the validity of the chromosomes that appear throughout the search is another problematic issue for GA usage [8]. Finally, the number of clusters has to be specified beforehand in many of the proposed methods. The application of GA to the clustering problem can be considered in the framework of partitioning approaches. Mainly, two different schemes are used for the representation of the clustering problem in the field of GA. The first one allocates each instance to a different gene of a chromosome and the value of the gene indicates its cluster. *Group Number Encoding* presented in [9] is a classical example of this scheme. In the second scheme, instances are represented by gene values and the position of a gene specifies its cluster. *Permutation with Separators* encoding [9] is an example of this scheme. The work described in [10] adapted the *Augmented Group Number Encoding* so that it can be applied to the general clustering problem. However, the representation used cannot encode different number of clusters in a fixed length chromosome. Thus, a variable length representation [11] is used to encode the different number of clusters that might appear throughout the search.

To sum up, most of the current clustering algorithms expect the number of clusters to be specified in advance by an expert, who may not be very familiar with the dataset and hence may give a biased and sometimes misleading estimate. Such an estimate may negatively affect the whole clustering process. Motivated by this, we have already demonstrated the effectiveness of utilizing GA in automatically finding the number of clusters. We achieved this by utilized a different scheme, named as *Linear Linkage* (LL) encoding, for encoding clustering solutions into chromosomes [12, 13]. The utilized representation forms a linked-list structure for instances in the same cluster. The genetic operators modify the chromosomes by altering the links. Also, we deal with the partitional clustering problem by using a multi-objective GA to minimize *Total Within Cluster Variation (TWCV)*, together with the number of clusters. TWCV [8] is a measure which denotes the sum of the average distance of cluster elements to cluster center. If TWCV measure is used as the sole objective in the search,

GA will tend to reduce the size of the clusters and eventually will form clusters with single elements where the variation turns out to be zero. This is effectively handled by the number of clusters objective.

We have already shown [12, 13] that LL encoding is superior to the other representational schemes by conducting experiments on two well-known datasets, namely Iris and Ruspini, which are considered as small datasets. Such small datasets could not highlight weaknesses of the fact that a single gene is reserved for each element to be clustered in LL encoding, just like the previously used encoding schemes. In other words, the chromosome length is equal to the size of the dataset used for LL, too. Consequently, GA performance decreases dramatically with large chromosomes. It is possible to have an unsatisfactory convergence, even if LL encoding is used on large datasets. To overcome this major shortcoming, a special preprocessing is proposed in this paper to decrease the number of instances that would go through the clustering phase. The aim is to facilitate for more effective and efficient clustering of high dimensional data using the LL encoding based method empowered with the preprocessing step. The immediate outcome of the preprocessing step is considerably better clustering with obviously improved performance.

Dermatology Database [14] is the testbed used in the experiments. This dataset is included in order to comment on and demonstrate the scalability of the approach. It is a high dimensional real-world dataset, which contains different types of *Eryhemato-Squamous* Disease. This dataset has been widely used as a benchmark for supervised techniques. The Dermatology dataset is not specifically designed for unsupervised (clustering) techniques and it is difficult to find unsupervised applications on it. Without the preprocessing step, the LL encoding based GA performed inefficiently on this dataset and could not produce the intended results. To the best of our knowledge, no significant success is obtained by a clustering technique on this dataset, but the results obtained with the LL representation scheme combined with the preprocessing step described in this paper are outstanding on this dataset. The output clustering accuracy is close to the results obtained by the supervised techniques.

The rest of the paper is organized as follows. The proposed approach to facilitate high-dimensional clustering is described in Section 2. The experimental results on the Dermatology dataset are reported in Section 3. Section 4 is conclusions.

2 Combining Preprocessing with Multi-objective GA for High-Dimensional Clustering

2.1 The Preprocessing Step

We have already mentioned above that a special preprocessing is used to decrease the number of instances that would go through the clustering phase. Note that we follow the approach where the length of a chromosome in the GA search is determined by the number of instances that will be clustered. It is a well-known

fact that the performance of the GA decreases dramatically with increasing chromosome length. Consequently, we decided on using a heuristic to filter out some of the instances from the clustering process in order to shorten the chromosome length without negatively affecting the overall overcome from the process. The idea is to find out groups of instances which are closer to each other more than a threshold value (δ) (to be specified experimentally) and filter out from the clustering process all of the elements of each group except one. The unfiltered elements are clustered with the GA and then the filtered elements are placed into the cluster that the unfiltered element of their group belongs to. The algorithm used can be defined as follows:

0. Mark all instances as unused.
1. Choose a random instance I_k, $0 \leq k \leq n$, where n is the number of instances to be clustered, and I_k has not yet been marked as used instance Mark I_k as unfiltered instance.
2. For all instances I_i, where $i \leq n$ and $i \neq k$
   2.1. If $distance(I_k, I_i) \leq \delta$ then
   Match I_i with I_k and filter out I_i from the clustering process.
3. Mark all I_i and I_k as used.
4. Repeat steps $[1-3]$ until no unused elements can be found.
5. Perform the clustering process with the unfiltered instances.
6. For each instance I_i filtered out from the clustering process.
   6.1. Find out the instance I_k with which I_i was matched at step 2.1 and place I_i into the same cluster with I_k.

The threshold (δ) is determined experimentally based on characteristics of the utilized dataset. For the *Dermatology dataset* used in the experiments, it has 366 instances. In the experiments, the value is increased until 174 instances of the set can be excluded from the clustering process. This δ is set as 0.038, which is equal to sixteen times the minimum distance that exists in the dataset. Hence, the clustering process is carried out on the remaining 192 instances which results in a considerable reduction in the chromosome length needed. Note that, as δ increases the risk of matching instances belonging to different classes appears. Certainly, this is a factor which would decrease the overall clustering accuracy at the end of the process. Hence, we tried to use a value which would minimally affect the clustering quality. In support of this, we decided on the value of δ experimentally and demonstrated the effectiveness of such selection process on the Dermatology dataset utilized in testing the proposed combined approach.

2.2 The Utilized Objectives

Multi-objective GA has been effectively utilized for optimization purposes that simultaneously involve multiple objectives which are mostly heterogeneous and competing. Several effective multi-objective GA methods have been developed, e.g., the *Niched Pareto Genetic Algorithm* presented in [15]. The method is based on the notion of *Pareto Domination*, which is used during the selection

operation. An element is considered to be *Pareto-dominant* over another one only if it is superior in terms of at least one objective, and without being inferior in terms of any of the objectives used. Hence, instead of a single solution, a set of *Pareto-optimal* solutions is obtained at the end of the search. None of the elements in this set would be *Pareto-dominant* over other elements. Another effective operation defined in [15] is called niching, which applies a pressure on the search to spread the genetic population along the pareto optimal surface.

The approach described in this paper is simply an integration of a preprocessing step into the *Niched Pareto Genetic Algorithm*, which optimizes the following two objectives: TWCV, and number of clusters. The former objective is articulated in the following intra-cluster distance formula:

$$TWCV = \sum_{n=1}^{N} \sum_{d=1}^{D} X_{nd}^2 - \sum_{k=1}^{K} \frac{1}{Z_k} \sum_{d=1}^{D} SF_{kd}^2 \tag{1}$$

where $X_1, X_2, .. , X_N$ are the N instances, X_{nd} denotes feature d of pattern X_n (n= 1 to N). SF_{kd} is the sum of the $d-$th features of all the patterns in cluster $k(G_k)$, and Z_k denotes the number of patterns in cluster $k(G_k)$ and SF_{kd} is computed as:

$$SF_{kd} = \sum_{\overrightarrow{x_n} \in G_k} X_{nd} \ , \quad (d = 1, 2, ...D). \tag{2}$$

TWCV is an effective objective function for the k-clustering problem. However, using TWCV alone is impossible for the general clustering problem. Comparing TWCVs for partitions of different sizes would be misleading. It is more probable for TWCV to decrease as the number of clusters increases. Hence, GA based clustering approach which uses TWCV only tends to find solutions with smaller clusters and larger partition sizes. This bias causes the GA search to end up with clusters consisting of single instances - TWCV for such trivial partition is zero. After realizing such a major restriction, we decided on using additional objectives to offset the bias towards trivial clustering. It is straightforward to set another objective as to favor more coarse-grained clustering, which is to minimize the partition size. The reported test results simply demonstrate that targeting these two objectives in a multi-objective GA leads to solutions for all different partition sizes in the Pareto optimal set.

2.3 The GA based Clustering Process

After the preprocessing step, the length of each chromosome will be k (the number of unfiltered instances), i.e., one gene is reserved for each unfiltered instance. These k instances are to be distributed into m clusters. Under the linkage encoding scheme, each gene stores an integer denoting its fellowship and not its membership. This is the fundamental difference between the group number encoding and the linkage encoding. Each gene is a link from an instance to another instance of the same cluster. Given k instances, any partition on them can be described as a chromosome of length k. Two instances are in the same group if either instance can be directed to the other instance via the links.

Without any constraint, the state of redundancy is just as bad as that of the group number encoding because the number of feasible chromosomes is still k^k.

Let the k genes be indexed inside a chromosome from 1 to k. The value of each gene in LL chromosome denotes the index of a fellow gene where the instances corresponding to these two genes would be in the same cluster. We can also treat the stored index as an out-link from a node, and if a gene stores its own index, it depicts an ending node. So, LL encoding gets its name because instances in a cluster construct a pseudo linear path with the only loop allowed being a self loop link to mark the last node. To qualify an unrestricted linkage chromosome as a valid LL encoding chromosome, the chromosome must comply with two constraints. First, the integer value stored in each gene is greater than or equal to its index, but less than or equal to k. Second, no two genes in the chromosome have the same value with the exception that at most two genes can have the same integer value if the integer is the index of an ending node.

We have already shown that LL encoding makes a one-to-one mapping between the chromosomes and clustering solutions [12, 13]. Although LL encoding keeps only fellowship in genes, it also implies the membership of each instance. Since each cluster must have one starting node and one ending node, both nodes can be used to identify a cluster. In practice, ending nodes are treated as the membership identifier for clusters because they are easier to detect.

Steps of the GA Process:

The initial population should include diverse chromosomes. It is intuitive to achieve this goal by generating random chromosomes, where each gene in a chromosome is assigned an integer randomly selected from the range 1 to k, where k is the number of instances to be clustered, and such that the first LL encoding constraint is satisfied an multiple links are prevented in order to achieve diversity.

Note that multiple links are allowed during the initialization process. Later, we will have backward links in a chromosome emerging in the process of the mutation operation. Therefore, a recovery process is needed after the constructors, and later other GA operators are employed to rectify a chromosome into its legitimate format. The Rectifying algorithm used for the recovery process involves two correction steps. First, backward links are eliminated from a chromosome. Then, multiple links to a node (except for the ending nodes) are replaced with one link in and one link out.

When two randomly selected chromosomes competing for a spot in the parent pool, they are not directly compared with each other. Rather, each is compared to a comparison set of chromosomes sampled from the current generation. If one of the competing chromosomes, say A, is dominated by the comparison set and the other chromosome, say B, is not dominated, then B advances to the parent pool. However, when both A and B are either dominated or not dominated by the set, the niche count of each chromosome is compared. The chromosome with the smaller niche count gets advantage. Niche count is an indicator of the

solution density around a chromosome in a certain solution population. This approach encourages even distribution of solutions in the GA population [15].

In each generation, the Pareto dominant set is achieved through a search in the whole population. Every individual is compared with the rest. If a chromosome is not dominated by any other chromosome, it is copied to the Pareto dominant set. The Pareto dominant set of the last generation contains the optimal solution.

We use one point crossover, which we have adjusted in a way to maintain the linked lists in a correct way. Then, the mutation operation modifies the value of a gene, and may have two different effects on the chromosome: 1) a sub-group of instances can be moved to a new cluster, or 2) a cluster can be split into two. In other words, the utilized mutation operator changes the membership of a set of instances rather than just a single instance.

3 Experimental Results

In this section, we report the result of our experiments on the *Dermatology Database* [14], which is a high dimensional real-world dataset that contains different types of *Eryhemato-Squamous* Disease. The diagnosis of the disease is a real problem in dermatology. The database contains the clinical features of six different disease types in the erythemato-squamous group. These are *psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, cronic dermatitis* and *pityriasis rubra pilaris*. Each instance of the dataset is labeled from 1 to 6 denoting its disease type. The database contains 366 instances and the majority class covers about %30 of the data. The dimension of the database is 34; 12 of these features are obtained by clinical observation, and the other 22 are *histopathological* ones obtained by the examination of the skin samples. All of the features are integer valued.

This dataset is a challenging real world problem which can be used to determine the effectiveness of the state of the art clustering and classification methods. The huge dimensionality of the data set is also appropriate to demonstrate the scalability of our new clustering scheme empowered with the preprocessing step.

This dataset has been widely used as a benchmark for supervised techniques. Since the labels of the disease types are specified in the dataset, researchers usually use a portion of the data for the training phase and the best observed accuracy is reported as 96.9% [16] for the test cases. Due to the difficulties mentioned above, the dataset turns out to be infeasible for most of the unsupervised (clustering) techniques. To the authors knowledge, no significant success is obtained by a clustering method on this dataset.

The experiments are carried out with the following GA parameters: Number of Experiments=30, Number of Generations=3000 population size=800 Nitch Radius=5 Crossover Rate=0.7 Mutation Rate=0.02. These values have been determined by running some initial tests. All data elements are scaled ranging from 0 to 1. The *cosine* metric has been used to determine the distances between instances of the dataset. This is a better choice compared to the *eu-*

clidean metric, since 33 features of the instances are nominal values. Note that the cosine metric is independent of the vector magnitudes.

Table 1. The reported Clusters

Cluster 1	**Cluster** 2	**Cluster** 3	**Cluster** 4	**Cluster** 5	**Cluster** 6
2 2 2 2 2 2 2 2 2	1 1 1 1 1 1 1 1 1	3 3 3 3 3 3 3	6 6 6 6 6 6	4 4 4 4 4 4 4	5 5 5 5 5 5
2 **4** **4** **1** 2 **4** 2 2 2	1 1 1 1 1 1 1 1 1	3 3 3 3 3 3 3	6 6 6 6 6 6	4 4 4 4 4 4 4	5 5 5 5 5 5
2 **4** 2 2 2 **4** **4** **4** 2	1 1 1 1 1 1 1 1 1	3 3 3 3 3 3 3	6 6 6 6 6 6	4 4 4 4 4 4 4	5 5 5 5 5 5
2 2 2 2 **4** **4** **4** 2 2	1 1 1 1 1 1 1 1 1	3 3 3 3 3 3 3	6 6	4 **2** 4 4 4 4	5 5 5 5 5 5
2 2 2 2 2 2 2 2 2	1 1 1 1 1 1 1 1 1	3 3 3 3 3 3 3		4 **2** 4 4 4 4	5 5 5 5 5 5
2 2 **1** 2 2 2 **4** **4** 2	1 1 1 1 1 1 1 1 1	3 3 3 3 3 3 3		4 4 4 4	5 5 5 5 5 5
2 **4** 2 2 2 2 2 2 2	1 1 1 1 1 1 1 1 1	3 3 3 3 3 3 3			5 5 5 5 5 5
2 2 2 **4** 2 2 2 2 2	1 1 1 1 1 1 1 1 1	3 3 3 3 3 3 3			5 5 5 5 5 5
2 2 2	1 1 1 1 1 1 1 1 1	3 3 3 3 3 3 3			5 5 5 5
	1 1 1 1 1 1 1 1 1	3 3 3 3 3 3 3			
	1 1 1 1 1 1 1 1 1	3 3			

It has been observed that the straightforward application of GA using LL encoding can not obtain a satisfactory convergence on the Dermatology dataset. When the preprocessing method presented in the previous section is used, it became possible to obtain convergence of GA in a practical time limit. The average GA execution time is calculated to be around 47.6 minutes on a $3Ghz$ pentium IV linux machine. More than this, the results obtained with this unsupervised technique are outstanding. The six clusters obtained at the end of the best GA-run are presented in Table 1. Each instance in a cluster is shown with the class label assigned to it in the dermatology database. The success criteria are certainly based on the match between the clusters and the existing classes. If instances in the same cluster have the same class labels then we can say that the cluster is formed correctly. For instance, the first cluster has collected the instances with label 2, the second one with label 1 and so on. The erroneous placements are shown with bold face. It can be easily seen in Table 1 that the majority class (labeled as class 1) is formed in cluster 2 with a hundred percent of correctness. The only problematic classes seem to be 2 and 4. Some instances of these two classes have been grouped in opposite clusters. The overall clustering accuracy is defined as the ratio of the misplaced instances to the total number of instances and turns out to be %95.01. This is an outstanding success for an unsupervised clustering technique, which competes with the results obtained using supervised techniques.

The average TWCV values up to 20 clusters are presented in Figure 1, from which the results clearly point out the optimum number of clusters for this dataset. In consistent with the results obtained on the previous datasets, the change in TWCV is quite stable down to the optimum number of clusters (6), but the TWCV increases dramatically when the number of clusters is smaller than 6. Furthermore, the same leap can be observed for the standard deviation bars and the variance is high in partitions with number of clusters smaller than the optimum.

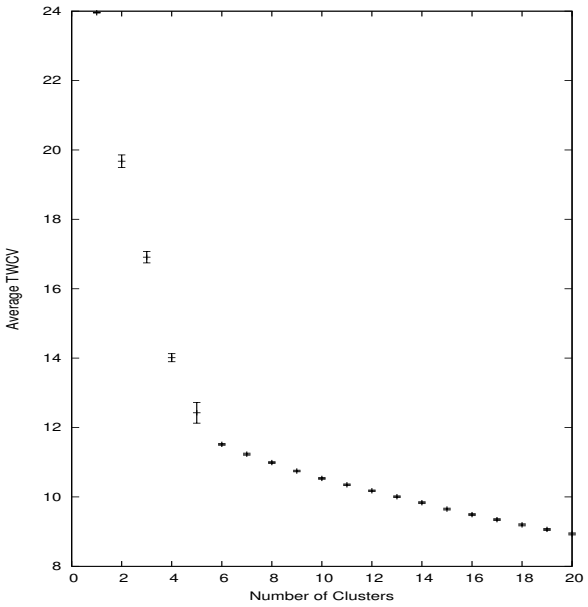


Fig. 1. The average TWCV values obtained on Dermatology-data. The results shown are the average of 30 different runs

4 Conclusions

In this paper, we extended the power of the linear linkage encoding scheme with a preprocessing step that reduces the number of instances to be used in the encoding process. This produced a more successful and effective multi-objective GA based clustering approach. This new scheme has been successfully used with the GA which is a powerful optimization technique. In other words, the special preprocessing step is introduced to reduce the size of the dataset used and hence to obtain a satisfactory convergence on large datasets. The results obtained on the Dermatology set provide a good insight about the importance and effectiveness of the new scheme. It has been observed that the preprocessing used is critical for achieving convergence on this dataset. The results clearly point out that the method is feasible for difficult real-world problems. The same GA process failed to achieve comparable success without the preprocessing step. To correctly read the results, it is necessary to note that the change in TWCV points out the optimal number of clusters for the tested Dermatology dataset. This result is consistent with the result obtained using supervised techniques.

The approach proposed in this paper is promising and open to further developments. We are planning to test the methodology with other well known real-world datasets to stress and elaborate more on its effectiveness and applica-

bility. Also, new genetic operators suitable to be used with LL might be designed in order to increase the performance further.

Acknowledgement

The authors would like to thank Jun Du for his participation to the implementation of the initial phase of the methodology.

References

1. Berkhin, P.: Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA (2002)
2. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data. John Wiley & Sons (1990)
3. McLachlan, G.J., Basford, K.E.: Chapters 1 and 2. In McLachlan, Basford, eds.: Mixture models: inference and applications to clustering. Marcel Dekker, Inc. (1988) 1–69
4. Dhillon, I., Fan, J., Guan, Y.: Efficient clustering of very large document collections. In Grossman, R., Kamath, C., Naburu, R., eds.: Data Mining for Scientific and Engineering Applications. Kluwer Academic Publishers (2001)
5. Han, J., Kamer, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann (2001)
6. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, Mass. (1989)
7. Falkenauer, E.: Genetic Algorithms and Grouping Problems. John Wiley&Sons (1998)
8. Krishna, K., Murty, M.: Genetic k-means algorithm. IEEE Transactions on Systems, Man, and Cybernetics - PartB: Cybernetics **29**(3) (1999) 433–439
9. Jones, D.A., Beltramo, M.A.: Solving partitioning problems with genetic algorithms. In Belew, Richard K.; Booker, L.B., ed.: Proceedings of the 4th International Conference on Genetic Algorithms, San Diego, CA, Morgan Kaufmann (1991) 442–449
10. Falkenauer, E.: A new representation and operators for genetic algorithms applied to grouping problems. Evolutionary Computation **2**(2) (1994) 123–144
11. Burke, D.S., Jong, K.A.D., Grefenstette, J.J., Ramsey, C.L., Wu, A.S.: Putting more genetics into genetic algorithms. Evolutionary Computation **6**(4) (1998) 387–410
12. Du, J., Korkmaz, E.E., Alhajj, R., Barker, K.: Alternative clustering by utilizing multi-objective genetic algorithm with linked-list based chromosome encoding. In: MLDM. (2005) 346–355
13. Korkmaz, E.E., Du, J., Alhajj, R., Barker, K.: Combining advantages of new chromosome representation scheme and multi-objective genetic algorithms for better clustering. Intelligent Data Analysis **10**(2) (2006)
14. Blake, C.L., Merz, C.J.: UCI repository of machine learning databases (2000)
15. Horn, J., Nafpliotis, N., Goldberg, D.E.: A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In: Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence. Volume 1., Piscataway, New Jersey, IEEE Service Center (1994) 82–87
16. Torrez, F.: (Machine learning datasets, universite charles de gaulle,groupe de recherche sur l'apprentissage automatique, cedex, france)

The OCS testing data analysis of hard spot based on data-mining technique

Chen Tanglong, Xiao Jian

School of Electrical Engineering, Southwest Jiaotong University
610031 Chengdu, China
tl_chen@sina.com

Abstract. The hard spot of OCS (Overhead Contact System) is a keen technical parameter which is used to evaluate the safety of electrified railway transportation and the quality of current collection between pantograph and catenary. Since the test of hard spot dependeds on the velocity of the train and the OCS conditions, studying on the kinetic characteristics of hard spot is of great importance. This paper proposes a method for OCS testing data analysis based on the technique of data mining. First the testing data are preprocessed through centering, de-dimension and standardization. Then the classification of the testing data has been made by the combination of hierarchical clustering and k-means clustering algorithm in terms of spatial location. Finally, the mathematical models of hard spot are obtained by carrying on the linear regression analysis for each classes of the data set. Both the statistics analysis and the simulation results confirm the feasibility of the approach proposed.

1 Introduction

The OCS is a large investment and high quality required system. It is essential for carrying out OCS on-line testing to guarantee the safe running and the quality of current collection.

The testing of OCS is to obtain the geometric and kinetic parameters of the contact line such as hard spot, pull-off value, contact pressure and velocity of train etc., through the sensors fixed up under the pantograph when train is running. The hard spot is an important parameter used to reflect the dynamic impact of pantograph and catenary. The bigger the value of hard spot is, the less smooth of the contact between the pantograph and catenary is, and the bigger offline of the dynamic contact between the catenary and pantograph is. This would leads to the unstableness of the current collection of the locomotive [2],[3]. Thus, it will affect the traction speed of the train, and increase the abrasion of pantogragh's slippery strip and contact line, which, in consequence, will reduce their service life.

The hard spot value tested by vehicular testing devices differs when train run at different speed. The question of at which value of hard spot, the catenary and pantograph needing to be repaired, and at which value of hard spot, the train speed needing to be restricted, is still open. All over the world, there is a lack of a technical

criterion for the power supply and maintenance departments in railway companies [1], [5].

In this paper, the OCS data, which were sampled from the section from Beijng-Guangzhou Railway Line in Dec.2004, are used to carry out analytical research by data-mining technique. Our goal is to determine the implied relationship among hard spot, speed and other parameters, so as to provide an approach to set up the mathematical model for the data processing of hard spot. By this way, it could normalize the hard spot values sampled at different speed to the corresponding values at specified speed of electrified section so one can evaluate the physical characteristics of the hard spot comprehensively, and is helpful for power supply and maintenance departments to adopt the most effective maintenance plan.

Furthermore, through the mathematical model established in this paper, the value of hard spot when train running at high speed can be estimated from the hard spot data measured by OCS test equipment, when the train running at low speed. This can supply an important theoretical basis for the reconstruction of low speed railway to meet the needs of high speed train, which is a heavy investment in China now. Besides, it can be used to evaluate the OCS condition so as to determine the highest speed allowable in a specified section of railway; Under which, the quality of current collection can be guaranteed.

2 Data preparation

There are hundreds of properties associated with mass measured data, most of them, however, are redundant, since a large part of them have nothing to do with the mining mission [6]. If analyzing these complex data directly, it will not only take a lot of time but also affect the accuracy of data-mining adversely. Therefore, it's necessary to eliminate noise, flaw and duplicate record of original data and to carry out de-dimension, centering and standardization processing in the stage of data preparation [7], under the premise of keeping integrity of the original data.

The centering processing of data is to carry out phase transition transform, defined as:

$$x_{ij}^{*} = x_{ij} - \overline{x}_j \tag{1}$$

where $i = 1,2,...,n$, $j = 1,2,...,p$, x_{ij}^{*} is the transformed coordinate and $\overline{x}_j$ is the center-point (mean) of the column vector. This transform could make the origin of new coordinate system coincide with the center of gravity of the sample data set. Neither the mutual position of every sample points nor the correlation of each variables will be changed by the above transform.

After centering, the variance of the data can be written as :

$$Var(x_j) = \frac{1}{n}\left\| x_j \right\|^2 = \frac{1}{n} x_j' x_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}^2 \tag{2}$$

In this paper, Euclidean distance is used to measure the distances of the center-points in the sample data space, we have:

$$d^2(e_i - e_k) = \|e_i - e_k\|^2 = \sum_{j=1}^{p} (x_{ik} - x_{jk})^2 \quad \textbf{(3)}$$

In practical problems, different measurement units are generally adopted for different variables, the numeric value of testing data differ with each other extremely. Simply using the Euclidean distance is inappropriate. For instance, the altitude-difference of contact line is generally ten millimeters more or less, while the voltage-difference is about ten kilovolts. Since the variation of the numerical value of altitude-difference is comparatively large, while the variation of the pressure-difference is insignificant, adopting common distance operator $d^2(i, j)$ will exaggerate the effect of the pressure variable and could not reflect the change of data truthfully. In order to eliminate the adverse effect of pseudo-variation, the technique of de-dimension is usually adopted to carry out compression processing for various of variables, viz. to make every variables has same variance 1, namely:

$$x_{ij}^* = x_{ij} / s_j \quad \textbf{(4)}$$

where. $s_j = Var(x_j)$.

The standardization processing of data is to carry on centering and compression processing simultaneously, namely:

$$x_{ij}^* = \frac{x_{ij} - \bar{x}_j}{s_j} \quad \textbf{(5)}$$

where $i = 1,2,...,n$, $j = 1,2,...p$; take the new data list as. $X^* = (x_{ij}^*)_{n\times p} = (x_1^*, x_2^*, ..., x_p^*)$. It could be proved that the property exists in variable space *E*, all of the data have same variance 1, namely: $Var(x_j^*) = \frac{1}{n}\|x_j^*\|^2 = \frac{1}{n}(x_j^*)' x_j^* = \frac{1}{n}\sum_{i=1}^{n}(x_{ij}^*)^2 = \frac{1}{n}\sum_{i=1}^{n}\frac{(x_{ij} - \bar{x}_j)^2}{s_j^2} = \frac{s_j^2}{s_j^2} = 1.$

Therefore, the points of variable set distribute on a super-sphere whose diameter is $\sqrt{n}$, viz. $\|x_j^*\|^2 = n$.

3 The realization of Data-mining

According to the characteristics of testing data, this paper adopts a bottom-to-up algorithm of hierarchical clustering to determine the number of class, viz. to search the original prototypes, then use k-means algorithm to optimize the results of

clustering. Through adjusting the threshold of the algorithm sequentially, the optimum results of clustering can be obtained.

There are 648 groups of OCS testing data. Merge every two groups into one if they are close enough, and use their center of gravity to represent the new group. Go on merging until the distances of data are greater than the specified threshold value *T*, and complete the classification of all testing data. Through experiments several times, the optimal clustering number is obtained to be 5,and the single points number is 6, from 648 groups of data. The distance of these single points are far from other points and far from each other too.

The effect of clustering can be judged by the Ward method. When selecting the group used to merge, Ward method can be used to minimize the object function. The principle is to produce some clusters which can meet the requirement of realizing interior aggregation and exterior separation to the most extent.

The total bias of *p* variables is *T*, which can be classified to 2 classes: intra-class bias *W* and inter-class bias *B*, and $T=W+B$. Suppose that a division is composed of *g* classes, and the total bias *T* of *n* variables is equivalent to the sum of biases of the deviation of single variable against their mean value $\overline{x}$:

$$T = \sum_{j=1}^{p} \sum_{i=1}^{n} (x_{ij} - \overline{x})^2 \tag{6}$$

The intra-class bias *W* is the sum of biases:

$$w = \sum_{k=1}^{g} w_k \tag{7}$$

where w_k represents the bias of *p*th variable in the *k*th class (The serial number is n_k , center is $\overline{x}_k = (\overline{x}_{1k}, \overline{x}_{2k}, \Lambda, \overline{x}_{pk})$), as shown bellow:

$$w_k = \sum_{j=1}^{p} \sum_{i=1}^{n_k} (x_{ij} - \overline{x}_{jk})^2 \tag{8}$$

The inter-class bias *B* is the sum of weighted biases which is the mean-value against the total mean-value in each group:

$$B = \sum_{j=1}^{p} \sum_{k=1}^{g} n_k (\overline{x}_{jk} - \overline{x}_j)^2 \tag{9}$$

With Ward method, if one can merge classes to make the value of *W* increase bigger than the value of *B*, the maximization of intra-class aggregation and inter-class separation is accomplished.

The k-means algorithm regards *k* as the coefficient and classifies *n* object into *k* clusters, so as to make there is a high degree of similarity between data within intra-class, and low similarity degree between inter-classes. The computation of similarity is based on the mean value (which is considered as center) of *k* clusters. Take the

clustering number $k=5$ as example to explain the procedure of k-means algorithm as follows:

Step 1: The processed data can be considered as a sample space *X*, and 5 points are randomly selected to be the prototypes: $\{w_1, w_2, w_3, w_4, w_5\}$. The selection principle is let the similarity degree of each point be small as far as possible.

Step 2: Suppose the number of clusters is 5, denoted as $\{C_1, C_2, C_3, C_4, C_5\}$, and are corresponding to 5 prototypes respectively, viz. the center-point of the first class C_1 is w_1, and w_2 for the second class C_2, etc. By removing 5 prototypes from original sample space *X*, forms new sample space X_1.

Step 3: Read the first group of data x_1' from the sample space X_1, figure out the distance $d_1', d_2', d_3', d_4', d_5'$ from x_1' to each prototypes $\{w_1, w_2, w_3, w_4, w_5\}$ respectively, then determine the minimum distance. Thus, consider that x_1' belongs to the class associated with minimum distance. Compute the center of gravity of the new cluster added with x_1' as the new prototype.

Step 4: Repeat the step 3, until the sample space is empty. Terminate the circulation. Then obtain the final prototypes $\{w_1^*, w_2^*, w_3^*, w_4^*, w_5^*\}$, which are the center of gravity of these classes.

Step 5: Read a group of data x_i from the original sample space X , figure out the distances d_1, d_2, d_3, d_4, d_5 from x_i to each prototypes $\{w_1^*, w_2^*, w_3^*, w_4^*, w_5^*\}$ respectively, and determine the minimum distance $d_{\min}$. For example, suppose $x_i = (0.0022, 1.3972, 0.1235, 3.9485)^T$, and the distances to each prototypes are $d_1 = 5.6672$, $d_2 = 1.5747$, $d_3 = 4.3201$, $d_4 = 4.6354$, $d_5 = 4.5462$. Then the minimum distance $d_{\min} = d_2 = 1.5747$, so x_i belongs to the second class. Continue process in this way, and complete the classification.

The object of clustering is to group the points that shared the same properties, while *k*-means method takes distance as the criterion. Therefore, we can use the intra-class distance and inter-class distance as a means to evaluate the effect of clustering. Let the distance from any point of a class to another point of the same class is x, and denote the distance from this point to any point of another class as y , if the distances from any fixed point to any other point in the same class are less than the distance from this point to any point of other class, then we believe the clustering is a good one. Otherwise, consider the clustering is not good.

Again take the clustering of 5 classes as an example. Let x-axis represent the appraisal range, whose interval is $[-1,+1]$, where $+1$ denotes that the distance from considered point of a specified class to the adjacent classes is very far, 0 represents that it is not clear that the considered point belongs to which class, -1 represents the result of clustering may be wrong. This computation can be done by:

$$S_i = \frac{\min(b(i,:) - a(i))}{\max(a(i), \min(b(i,:)))} \tag{10}$$

where a_i represents the average distance from the *i*th point to another point of the same class; $b(i,k)$ represents the average distance from the *i*th point of a certain class to another class (the *k*th class). Y-axis represents the number of classes and the number of data within a class.

Fig 1 is the results of clustering of OCS testing data with 5 clusters, it can be seen that the clustering result is desirable.

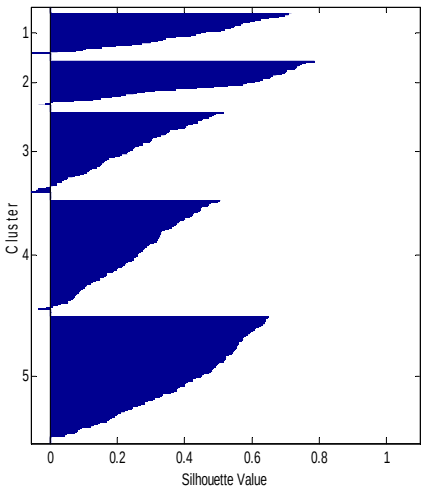


Fig. 1. The simulated plot of clustering with 5 clusters

4 Regression Analysis

Through the analysis above, the OCS testing data are clustering to 5 classes in terms of spatial location in order to determine the relationship of hard spot with the altitude-difference, velocity and pressure-difference etc. from the OCS testing parameters. Experiments indicate that the testing data exhibit linear distribution in the space. Therefore, it is reasonable to set up the mathematical model by linear regression method [10],[11].

The Original OCS testing data are classified into 5 classes, and a technique of system identification is provided to determine the relationship between hard spot and other parameters as follows.

In the process of regression analysis, dependent argument *y* is assigned to denote the hard spot, while other parameters are expressed with arguments x_1, x_2, Λ, x_p. The general form of linear model is shown as below:

$$\begin{cases} Y = X\beta + \varepsilon \\ \varepsilon \sim N_n(0, \sigma^2 I_n) \end{cases} \quad \textbf{(11)}$$

Where *y* is a $n \times 1$ dependent variable vector; *X* is a $n \times p$ argument regression matrix; β is the regression coefficient of a $p \times 1$ parameter vector; ε is a $n \times 1$ random disturbance vector. β and ε are independent, and submit Gaussian distribution $N(0, \sigma^2)$ and σ^2 is unknown. $N_n(\mu, \Sigma)$ represents a *n*-dimensional Gaussian distribution with the mathematical expect value μ and covariance matrix Σ. Observing y for *n* times independently, viz. to obtain the observed value y_1, Λ, y_n under the condition of x_{i1}, Λ, x_{ip} and $i = 1, \Lambda, n$. According to Eq.(11), it could be written as:

$$\begin{cases} y_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip} + \varepsilon_i \\ \varepsilon_i \sim N(0, \sigma^2), i = 1, ..., n \end{cases} \quad \textbf{(12)}$$

That is:

$$\hat{\beta} = (X'X)^{-1} X'Y \quad \textbf{(13)}$$

$$\hat{\sigma}^2 = \frac{1}{n-p-1} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad \textbf{(14)}$$

Where $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \Lambda + \beta_p x_{ip}$ is the estimated value of y_i, $\hat{Y} = X\hat{\beta} = X(X'X)^{-1} X'Y = HY = (y_1, \Lambda, y_n)'$ is the estimated value of Y. X' is the transposed matrix of X, $H = X(X'X)^{-1} X'$ is the projection matrix, $e = Y - \hat{Y}$ is called residual error, Q is the sum of residual error with degree of freedom (*n-p-1*).

$$Q = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad \textbf{(15)}$$

U is called regression squares sum, whose degree of freedom is p.

$$U = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 \tag{16}$$

It's can be proved that $\hat{\beta}$ is β 's optimum linear unbiased estimation and $\hat{\beta} \sim N_{p+1}(\beta, \sigma^2 (X'X)^{-1})$.

The equation below is called general variation squares sum, $S = Q + U$, whose degree of freedom is (*n-1*).

$$S = \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{17}$$

Obviously, we have $\frac{S}{\sigma^2} \sim \chi^2(n-1)$.

To test whether there exists linear relationship between dependent variable *y* and argument $x_1, x_2, ..., x_p$ as shown in Eq.(12), we need to examine the assumption as follows:

$$H_0 : \beta_1 = \Lambda = \beta_p = 0 \tag{18}$$

The statistic used here is:

$$F = \frac{U/P}{Q/(n-p-1)} \tag{19}$$

If H_0 exists, $\frac{Q}{\sigma^2} \sim \chi^2(n-p-1)$, $\frac{U}{\sigma^2} \sim \chi^2(p)$, and are independent with each other. So, the rejection field with confidence level α is:

$$P\{F > F_{1-\alpha}(p, n-p-1)\} = \alpha \tag{20}$$

This means that if observed value *F* is greater than $F_\alpha(p, n-p-1)$, we reject H_0 under confidence level $1-\alpha$, and believe that the linear relationship is significant.

Carrying the regression analysis for the first class, which contains 66 groups of data in all, we get regression model

$$y = -0.3631 + 0.0109x_1 + 0.0376x_2 + 0.0069x_3 \tag{21}$$

F statistic is used here to verify the hypothesis. From Eq.(15), one can obtain the squares sum of residuals $Q = 3.0931$. From Eq.(16), one can get the regression squares sum $U = 0.49976$. From Eq.(19), one can obtain $F = 3.3411$. Given

$\alpha = 0.05$, by looking up the table we can get $F_\alpha = 2.76$. $F = 3.3433 > 2.76 = F\alpha$, so we reject the assumption H_0 and regard that the regression model exists linear relationship.

Table 1. The residual error list of the first class

x_1	x_2	x_3	y	$\hat{y}$	e	e/y
1.8132	1.3307	0.1847	-0.1912	-0.29203	0.010083	0.10083
1.5868	-1.7287	0.1235	-0.4672	-0.40995	0.057249	0.057249
-0.0544	-1.3296	0.0624	-0.4672	-0.41326	0.053945	0.053945
1.3605	-1.7287	-0.3657	-0.4672	-0.41579	0.051407	0.051407
2.0396	-1.9947	-0.3657	-0.4672	-0.41839	0.048808	0.048808
2.0396	-2.1942	0.6739	-0.4672	-0.41872	0.04848	0.04848
1.1907	-1.8617	0.0624	-0.4672	-0.41969	0.047509	0.047509
0.9643	-1.8617	0.2459	-0.4672	-0.42089	0.046308	0.046308
-1.3561	-1.8617	0.7963	-0.3982	-0.44239	0.044187	0.044187
-0.6204	-1.4627	0.2459	-0.4672	-0.42316	0.044037	0.044037
-0.2808	-1.6622	0.6128	-0.4672	-0.42443	0.042769	0.042769

Residual error is also an effective means for verification. The residual of each classes is shown in Table.1, where x_1 represents the altitude-difference of contact line; x_2 represents the velocity of the train; x_3 represents the pressure-difference of catenary; y is the testing data of hard spot,; $\hat{y}$ is the estimated value of the hard spot. $e = y - \hat{y}$ is the residual error. In practical use, what we concern with is the variation ratio of residual associate with the field testing data, e/y , viz. the error rate. As can be seen from the table, only one error rate of a point reaches 0.1, while others are constrained within the range of 6%. In consideration of the complexity of testing environment, this result is acceptable.

Fig .2 shows the simulated curve of the hard spot of the first class, where the dash line represents the tested data of the hard spot, and the solid line represents the estimated value of the hard spot. In order to demonstrate the curve more clearly, arrange the hard spot in ascending order before simulation. Take into account of the complexity of the testing condition, the results indicate the regression equation is credible.

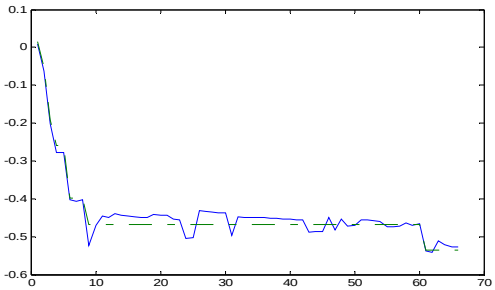


Fig. 2. The simulated curve of the 1st class

By carrying out regression analysis of 73 groups of data in all for the second class, we could get the regression equation as follows:

$$y = 2.5250 - 0.0825x_1 - 0.0514x_2 + 0.3051x_3 \quad \textbf{(22)}$$

Fig.3 shows the simulated curves of the testing data and the estimated value, which belong to the second class. As can be seen from the figure, there are two points whose estimated value exist severe dissociation, the reason is most probably that there are some faults in the the data collected from railway section. In practical use, these points need to be undergone a smooth process.

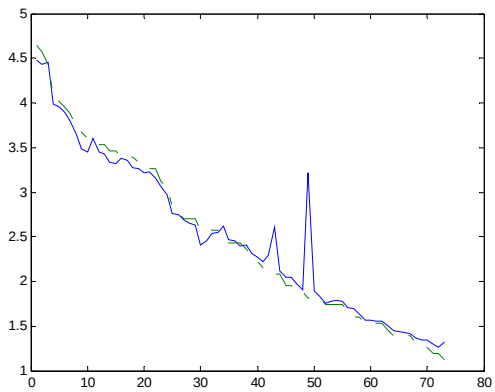


Fig. 3. The simulated curve of the 2nd class

By carrying out regression analysis of 132 groups of data in total for the third class, we can get the regression equation as follows. Fig.4 shows the simulated curve of the third class.

$$y = -0.6497 - 0.1699x_1 + 0.4789x_2 - 0.0885x_3 \quad \textbf{(23)}$$

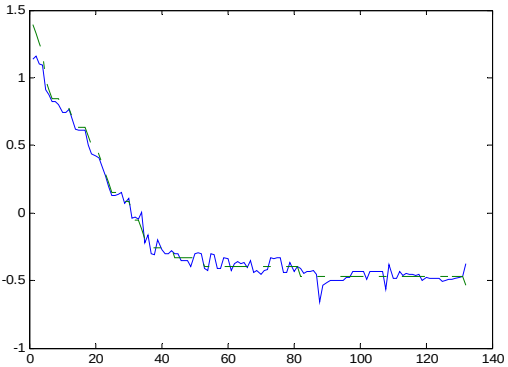


Fig. 4. The simulated curve of the 3rd class

There are 181 groups of testing data in class 4. Fig.5 shows the simulated curve of the fourth class. We can also obtain the regression equation as follows:

$$y = -0.3954 + 0.0377x_1 - 0.0950x_2 + 0.0152x_3 \quad \textbf{(24)}$$

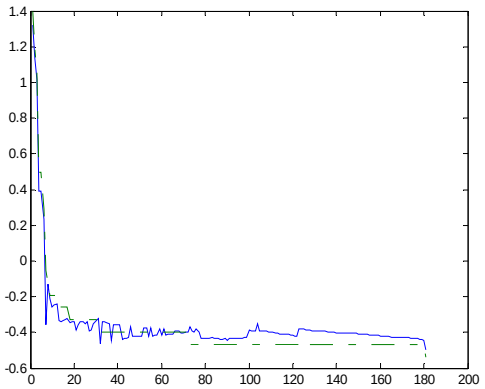


Fig. 5. The simulated curve of the 4th class

There are 196 groups of testing data in class 5. Fig.6 shows the simulated curve of the fifth class, and the regression equation is as follows:

$$y = -0.4372 + 0.1069x_1 + 0.0564x_2 + 0.0257x_3 \quad \textbf{(25)}$$

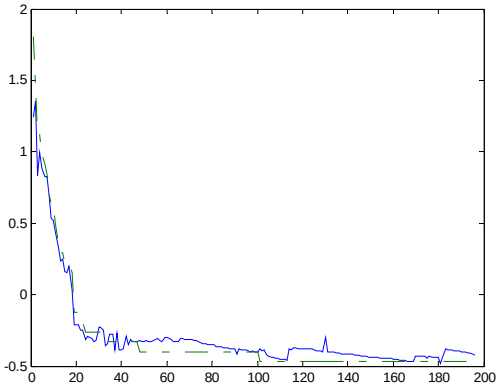


Fig. 6. The simulated curve of the 5th class

From the statistics analysis and simulation results mentioned above, it is clear that the linear regression model of the hard spot, which is obtained through data-mining technique and regression analysis, is acceptable. This shows the powerfulness of the data-mining technique in dealing with mass data.

5 Conclusion

This paper presents an OCS testing data analysis method by data mining technique. Mass data are collected by OCS testing equipment. The original OCS testing data are then separated into 2 parts, one for analysis so as to get the mathematical model, and the other part for the verification of the model. Through the preprocessing and classification of the original data and determining the distances of each class's center of gravity respectively, we have determined the classes for each groups of data, by the combination of hierarchical clustering and k-means clustering algorithm. These steps optimize the results of clustering that can summarize certain number of different physical characteristics of electrified railway section effectively, which contains a lot of information such as curve section, straight-line section, newly built electrified railway section etc.. Then regression analysis is used to find corresponding regression equations that represent the mathematical model of the hard spot. Statistics analysis and Simulation results indicate that the regression equations are effective. Preprocessing of the original data is also necessary to ensure the accuracy of the clustering and regression analysis.

Since the OCS testing data considered here obtains relatively less quantity of the data tested at some fixed points of railway section and observed at different time by the OCS testing car, the mathematical model established here needs to go on further improvement and enhancement. We believe, however, that data-mining is an effective method to dealing with OCS testing data.

References

1. Wan-Ju Yu，M.: Catenary systems of high-speed electrified railway. Southwest Jiaotong University Publishing House, Chengdu (2003.7)
2. Gukow, Kiessling Puschmann, Schmieder, Schmidt, B.C.: Fahrleitungen elektrischer Bahnen. Teubner, Stuttgart (1977)
3. Vinayagalingam T. Computer, J.: Evaluation of Controlled Pantographs for Current Collection from Simple Catenary Overhead Equipment at High Speed. Journal of Dynamic System, Measurement and Control. (1983.12)
4. Belyaev I A，Vologine V A，Frief A V, J.: Improvement of Pantographs and Catenary and Method of Caculation Their Mutual Interactions at High Speeds. Journal of Dynamic System, Measurement and Control. (1983.12)
5. Wormley D, R.: Dynamic Performance Characteristics of New Configuration Pantogragh-Catenary System. Final Report DOT/OST/P. (1984)
6. Sharon Bjeletich，Greg Mable., M.: Microsoft SQL Server 7.0 GuideLines. Tsinghua University Publishing House, Beijing (2000.9)
7. Keim D A, J.: Pixel-oriented Visualization Techniques for Exploring very large Databases, Journal of Computational and Graphical Statistics (1996.3)
8. Erik Thomsen, M.: OLAP Solution：Creating multi-dimensional Information System. Electronic Industry Publishing House, Beijing (2004.9)
9. Agrawal., C.R.: Automatic Subspace Clustering of High Dimensional Data for Data Mining Application. Proc. ACM SIGMOD'98 Int. Conf. On Management of Data, Seattle, WA, (1998)
10. Richard J.Roiger, Michael W.Geatz, M.: A Tutorial-bases Prime, Data Mining. (2003.12)
11. Margaret H. Dunham, M.:. Data Mining Introductory and Advanced Topics. Tsinghua University Publishing House, Beijing (2003)
12. J. David Irwin, M:. Basic Engineering Circuit Analysis. Department of Engineering Auburn University (1985)

Toward Intrinsic Gene Identification Using Random Forest with Dynamic Feature Selection

Ha-Nam Nguyen[1], and Syng-Yup Ohn[1]

[1] Department of Computer Engineering
Hankuk Aviation University, Seoul, KOREA
{nghanam,syohn}@hau.ac.kr

Abstract. Determining the relevant features is a combinatorial task in various fields of machine learning such as text mining, bioinformatics, pattern recognition, etc. Several scholars have developed various methods to extract the relevant features but no one is really superior. Recently, Breiman proposed Random Forest to classify a pattern based on CART tree algorithm and his method turns out good results compared to other classifiers. Taking advantages of Random Forest and using wrapper approach which was first introduced by Kohavi *et al*, we propose an algorithm named Dynamic Recursive Feature Elimination (DRFE) to solve the feature selection problem. In our method, we use Random Forest as induced classifier and develop our own defined feature elimination function by adding extra terms to the feature scoring. We conducted experiments with two public datasets: Colon cancer and Leukemia cancer. The experimental results of the real world data showed that the proposed method has higher prediction rate compared to the baseline algorithm and has comparable and sometimes better performance than the widely used classification methods in the same literature of feature selection.

1 Introduction

Machine learning techniques have been widely used in various fields such as text mining, network security and especially in bioinformatics. There are wide ranges of learning algorithms have been studied and developed, i.e. Decision Trees, K Nearest-Neighbor, Support Vector Machine, etc. The existing learning algorithms do well in most cases. However, as the number of features in dataset is large, the performance of those algorithms is degraded. In that case the whole set of features of a dataset usually over-describes the data relationships. Thus, an important issue is how to select a relevant subset of features based on their criteria. A good feature selection method should heighten the success probability of the learning methods [1, 2]. In other words, this mechanism helps to eliminate noises or non-representative features which can impede the recognition process.

Recently, Breiman proposed Random Forest (RF) based on an ensemble of CART tree classifications [3]. This method turns out better results compared to other classifiers including Adaboost, Support Vector Machine and Neural Network. Researchers applied RF as a feature selection method [4, 5]. Some tried RF directly [4] and others

adapted it for relevance feedback [5]. The previous approach [5] attempts to address this problem with correlation techniques. In this paper, we introduce a new method of feature selection based on Recursive Feature Elimination. The proposed method reduces the set of features via feature ranking criterion. This criterion re-evaluates the importance of features according to the Gini index [6, 7] and the correlation of training and validation accuracy which are obtained from RF algorithm. By that way, we take both feature contribution and correlation of training error into account. We applied the proposed algorithm to classify several datasets such as Colon cancer and Leukemia cancer. The DRFE showed better classification accuracy than RF and sometime it showed better results compared to other studies.

The rest of this paper is organized as follows. In section 2 we describe feature selection approaches. In Section 3 we briefly review RF and its characteristics that will be used in proposed method. The framework of proposed method is presented in Section 4. Details of the new feature elimination method will be introduced in Section 5. Section 6 explains the experimental design of proposed method and the analysis of obtained results. Some concluding remarks are given in Section 7.

2 Feature Selection Problem

In this section, we briefly summarize the space dimension reduction and feature selection methodologies. Feature selection approach has been regarded as a very effective way in removing redundant and irrelevant features, so that it increases the efficiency of the learning task and improves learning performance such as learning time, convergence rate, accuracy, etc. A lot of studies have focused on feature selection literature [1, 2, 8-11]. As mentioned in [1, 2], there are two ways to determine the starting point in a searching space. The first strategy might start with nothing and successively adds relevance features called *forward selection*. The other one, named *backward elimination*, starts with all features and successively removes irrelevant ones. Another heuristic strategy is *Bi-directional* Selection [12]. In this case, the feature subsets starts with null, full or randomly produced feature subset, then adds the currently best feature into or removes the currently worst feature from it, so that a given guideline values best at each iteration, until a prearranged performance requirement is met.

There are two different approaches used for feature selection, i.e. Filter approach and Wrapper approach [1, 2]. The Filter approach considers the feature selection process as precursor stage of learning algorithms. The most disadvantage of this approach is that there is no relationship between the feature selection process and the performance of learning algorithms. The second approach focuses on specific machine learning algorithm. It evaluates the selected feature subset based on goodness of learning algorithms such as accuracy, recall and precision values. The disadvantage of this approach is high computation cost. Some researchers tried to propose methods that can speed up the evaluating process to decrease this cost. Some studies used both filter and wrapper approaches in their algorithms called hybrid approaches [9, 10, 13-15].

Each feature subset should be evaluated belong to the approaches. Filter model usually uses evaluation functions which try to evaluate the classification performances of features. There are many evaluation functions such as Feature Importance [1-3, 7], Gini [3, 6, 7], Information Gain [6], the Ratio of Information Gain [6], etc. Wrapper model uses learning accuracy for the evaluation. In this way, all samples should be divided into two sets, training set and testing set with the same feature subset. Then, the algorithm of the model runs on the training set, and applies the result system on the testing set to give out the learning accuracy. They usually use cross validation to avoid the affection of the sample division. The feature subset is found in a search space. In this space, each state represents a feature subset, and the size of the search space for n features is $O(2^n)$, so it is impractical to search the whole space exhaustively, unless n is small. We should use heuristic function to find the state with the highest evaluation. Some techniques introduced for this purpose are Hill-climbing, Best-first search, etc.

In these methods, the feature criteria or randomly selection methods are used to choose the candidate feature subsets. The cross validation mechanism is employed to decide the final best subset among the whole candidate subsets [6].

3 Random Forest

Random Forest is a special kind of ensemble learning techniques [3]. It builds an ensemble of CART tree classifications using bagging mechanism [6]. By using bagging, each node of trees only select a small subset of features for the split, which enables the algorithm to create classifiers for high dimensional data very quickly. Ones have to specify the number of randomly selected features (*mtry*) at each split. The default value is *sqrt*(*p*) for classification where *p* is number of features. The Gini index [6, 7] is used as the splitting criterion. The largest possible tree is grown and not pruned. One should choose the big enough number of trees (*ntree*) to ensure that every input feature gets predicted at least several times. The root node of each tree in the forest keeps a bootstrap sample from the original data as the training set. The out-of-bag (OOB) estimates are based on roughly one third of the original data set. By contrasting these OOB predictions with the training set outcomes, one can arrive at an estimation of the predicting error rate, which is referred to as the OOB estimate of error rate.

To represent what is the out-of-bag (OOB) estimate method, we assume a method for build a classifier from training set. We can construct classifiers $H(x, T_k)$ based on bootstrap training set T_k from given training set T. The out-of-bag classifier of each sample (x, y) in training set is defined as the aggregate of the vote only over those classifiers for which T_k does not containing that sample. Thus the out-of-bag estimate for the generalization error is the error rate of the out-of-bag classifier on the training set.

The Gini index is defined as squared probabilities of membership for each target category in the node.

$$gini(N) = \frac{1}{2}\left(1 - \sum_j p(\omega_j)^2\right) \quad \textbf{(1)}$$

where $p(\omega_j)$ is the relative frequency of class ω_j at node N. It means if all the samples are on the same category, the impurity is zero; otherwise it is positive value. Some algorithm such as CART, SLIQ, and RF were used Gini index as splitting criterion [3, 6, 7, 16]. It tries to minimize the *impurity* of the nodes resulting from split based on following formula

$$gini_{split} = \sum_{i=1}^{k} \frac{n_i}{n} gini(i) \quad \textbf{(2)}$$

where k is number of partitions when a node N is split into, n_i is number of samples at child i and n be the total number of samples at node N. In Random forest the Gini decreases for each individual variable over all trees in the forest gives a fast variable important that is often very consistent with the permutation importance measure [3, 7]. In this paper we used the Gini decreases of variable as a component of importance criteria.

4 Proposed approach

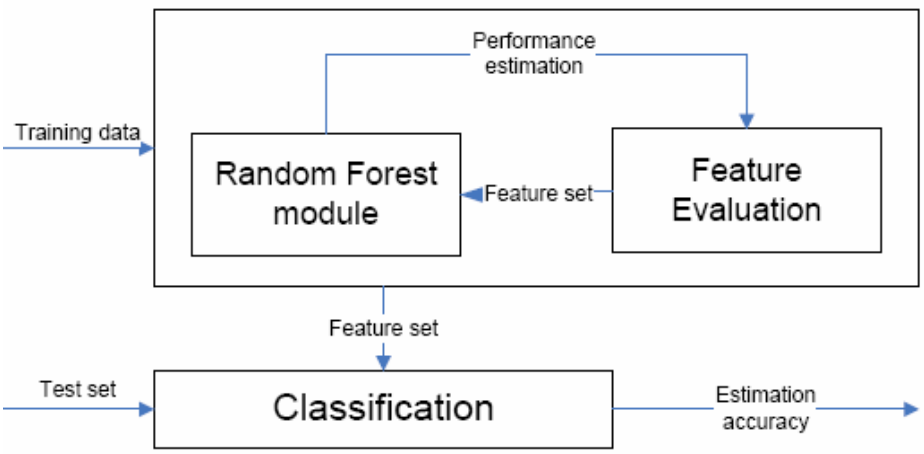


Fig. 1. The main procedures of proposed approach

The overall procedure of our approach is shown in Fig. 1.

The proposed method used *Random Forest module* to estimate the performance consisting of cross validation accuracy and the importance of each feature in training data set. Even though RF robust against over-fitting problem itself [3, 6], the pro-

posed approach can not inherit this characteristic. To deal with over-fitting problem, we use n-fold cross validation technique to minimize generalization error [6].

The *Feature Evaluation* module computes the feature importance ranking values according to the obtained results from *Random Forest module* (see formula 3). The irrelevant feature(s) are eliminated and only important features are survived by mean of feature ranking value. The survival features are again used as input data of *Random Forest module*. This process is repeated until it satisfies the desired criteria.

The set of features, which is a result of learning phase, is used as filter of Test dataset in classification phase. The detail of proposed algorithm will be presented in next section.

5 Dynamic Recursive Feature Elimination Algorithm

When compute ranking criteria in wrapper approaches, they usually concentrate on the accuracies of the features, but not much on the correlation of the features. A feature with good ranking criteria may not create a good result. Also the combination of several features with good ranking criteria, may not give out a good result. To remedy the problem, we propose a procedure named Dynamic Recursive Feature Elimination (DRFE).

1. Train data by Random Forest with cross validation
2. Calculate the ranking criterion for all features F_i^{rank} where i=1..n (n is the number of features).
3. Remove feature by using *DynamicFeatureElimination* function (for computational reasons, it may be more efficient if we remove several features at a time)
4. Back to step 1 until reach the desired criteria.

In step 1, we use Random Forest with n-folders cross vadilation to train the classifier. In the j[th] cross validation, we will obtain turtle (F_j A_j^{learn} $A_j^{validation}$) are the feature importance, the learning accuracy and the validation accuracy respectively. We will use those values to compute the ranking criterion in step 2.

The cores of our algorithm are presented in step 2. In this step, we use the results from step 1 to build ranking criterions which will be used in step 3. The ranking criterion of feature i^{th} is computed as follow

$$F_i^{rank} = \sum_{j=1}^{n} F_{i,j} \times \frac{(A_j^{learn} + A_j^{validation})}{|A_j^{learn} - A_j^{validation}| + \varepsilon} \quad \textbf{(3)}$$

where j=1,.., n is number of cross validation folders, $F_{i,j}$, A_j^{learn} and $A_j^{validation}$ are the feature importance in terms of the node impurity which can be computed by Gini inpurity, the learning accuracy and the validation accuracy of feature j-th obtained from *RandomForest* module respectively. ε is real number with very small value.

The first factor ($F_{i,j}$) is presented the Gini decreases for each feature over all trees in the forest when we train data by RF. Obviously, the higher decrease of $F_{i,j}$ is obtained, the better rank of feature we have [3, 6] . We use the second factor to deal

with the overfitting issue [6] as well as the desire of high accuracy. The numerator of the factor presents for our desire to have high accuracy. The more this value we got, the better the rank of the feature is. We want to have a high accuracy in learning and also want not too fit the training data which called overfitting problem [6]. To solve this issue, we applied the n-folder cross validation technique [6]. We can see that the less difference between learning accuracy and validation accuracy, the result is the more stability of accuracy. In the other words, the target of denominator is to reduce overfitting problem. In the case that learning accuracy is equal to validation accuracy, the difference is equal to 0, we use ε with very small value to avoid the fraction coming to ∞.

We want to choose the feature with both high stability and high accuracy. To deal with this problem, the procedure choose a feature subset only if the validation of this selected feature subset is higher than validation of the previous selected feature set. This heuristic method ensure that the feature set we chose always have better accuracy. As a result of step 2, we have an ordered-list of ranking criterion of features.

In step 3, we propose our feature elimination strategy based on backward approach. The proposed feature elimination strategy depends on both ranking criterion and validation accuracy. The ranking criterion making the order of features elimination and the validation accuracy is used to decide whether the chosen subset of features is permanently eliminated. In normal case, our method eliminates features having the smallest value of ranking criterion. The new subset is validated by *RandomForest* module. The obtained validation accuracy plays a role of decision making. It is used to evaluate whether the selected subset is accepted as new candidate of features. If the obtained validation accuracy is lower than previous selected subset accuracy, it tries to eliminate other features based on their rank values. This iteration is stopped whenever the validation accuracy of the new subset is higher than the previous selected subset accuracy.

If there is no feature to create new subset and no better validation accuracy, the current subset of features is considered as the final result of our learning algorithm. Otherwise the procedure goes back to step 1.

6 Experiments

We tested the proposed algorithm with several dataset include two public datasets (Leukemia and Colon cancer) to validate our approach. In this section, we represent the description of used datasets, our experimental configurations, and some evaluations about the obtained results.

6.1 Datasets

The colon cancer dataset contains gene expression information extracted from DNA microarrays [1]. The dataset consists of 62 samples in which 22 are normal samples and 40 are cancer tissue samples, each having 2000 features. We randomly choose 31

samples for training set and the remaining 31 samples were used as testing set. (Availble at: http://sdmc.lit.org.sg/GEDatasets/Data/ColonTumor.zip).

The leukemia dataset consists of 72 samples divided into two classes ALL and AML [17]. There are 47 ALL and 25 AML samples and each contains 7129 features. This dataset was divided into a training set with 38 samples (27 ALL and 11 AML) and a testing set with 34 samples (20 ALL and 14 AML) (Availble at: http://sdmc.lit.org.sg/GEDatasets/ Data/ALL-AML_Leukemia.zip).

6.2 Experimental Environments

Our proposed algorithm was coded using R language (http://www.r-project.org; R Development Core Team, 2004), using *RandomForest* packages (from A. Liaw and M. Wiener) for random forest module. All experiments are conducted on a Pentium IV 1.8 GHz personal computer. The learning and validation accuracies were determined by means of 4-fold cross validation. The data were randomly split into training set and testing set. In this paper, we used RF with original dataset as the base-line method. The proposed method and base-line method were executed with the same training and testing dataset to compare the efficiency of the two methods. Those implementations were done 50 times to test the consistency of obtained results.

6.3 Experimental Results and Analysis

6.3.1 Colon Cancer

The data was randomly divided into a training set of 50 samples and testing set of 12 for 50 times, and our final results were averaged over these 50 independent trials (Fig. 2). In our experiments, we use the default value for the *ntry* parameter (see Sec. 3) and *ntree* parameter was tried with some different values 500, 1000, 1500, and 2000.

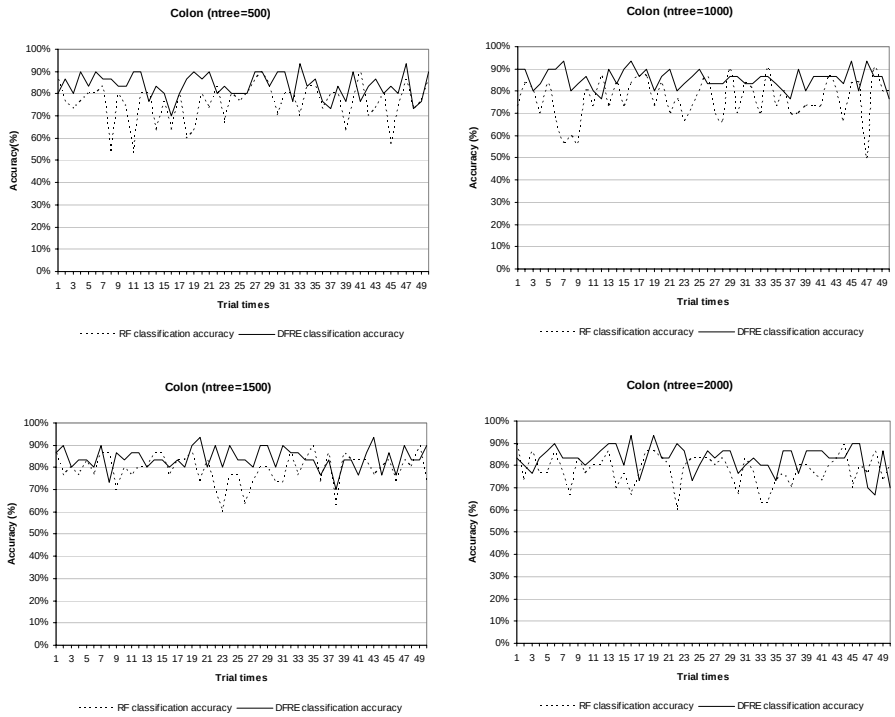


Fig. 2. The comparison of classification accuracy between DRFE (dash line) and RF (dash-dot line) via 50 trials with parameter *ntree* = {500, 1000, 1500, 2000} in case of Colon dataset

Table 1. The average classification rate of Colon cancer over 50 trials (average % classification accuracy ±standard deviation)

Tree number	500	1000	1500	2000
RF only	75.6±8.9	76.0±9	79.3±6.8	78.0±7.1
DRFE	83.5±5.6	**85.5±4.5**	84.0±5.1	83.0±6.0

The summary of classification results are depicted in Table 1. The classification accuracies of the proposed algorithm are significantly better than the baseline one. Table 2 presents the average number of selected features obtained from all experiments. As mentioned above, several features are eliminated each iteration because of speed up reason (Sec. 5). The proposed method achieves accuracy of 85.5% when performing on about 141 genes predictors retained after using the DRFE procedure. This number of genes only makes up about 7.1% (141/2000) of the overall genes. The method not only increases the classification accuracies but also reduces the standard deviation values (Table 1).

Table 2. Number of selected feature with different tree number parameters in case of Colon cancer over 50 trials (average number ± standard deviation).

Tree number	500	1000	1500	2000
Number of selection features	172±70	141±91	156±83	129±96

Some studies have done in term of feature selection approaches. The comparison of those studies' results and the proposed approach result are depicted in Table 3. Our method showed sometimes better results compared to the old ones. In addition, the standard deviation values of the proposed method are much lower than both RF (see Table 1) and other methods including GA\SVM [9], Bootstrapped GA\SVM [10] and Combined kernel for SVM [18] (see Fig. 3). It shows that the proposed method turned out more stable results than previous ones.

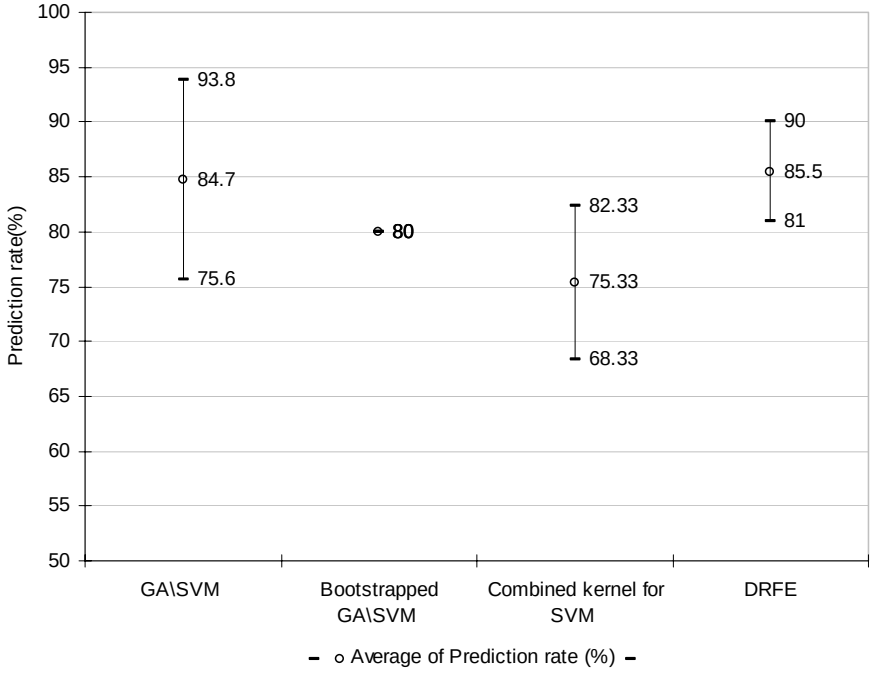


Fig. 3. The best prediction rate and it's standard deviation of some studies in case of Colon dataset

6.3.2 Leukemia Cancer

As mentioned in Sec. 6.1, the Leukemia dataset is already divided into training and testing set. To setup the 50 independent trials, we randomly selected 4000 features among 7129 given set of features. In this experiment, the *ntree* parameter was set to

1000, 2000, 3000, and 4000. By applying DRFE, the classification accuracies are significantly improved in all 50 trials (Fig. 4).

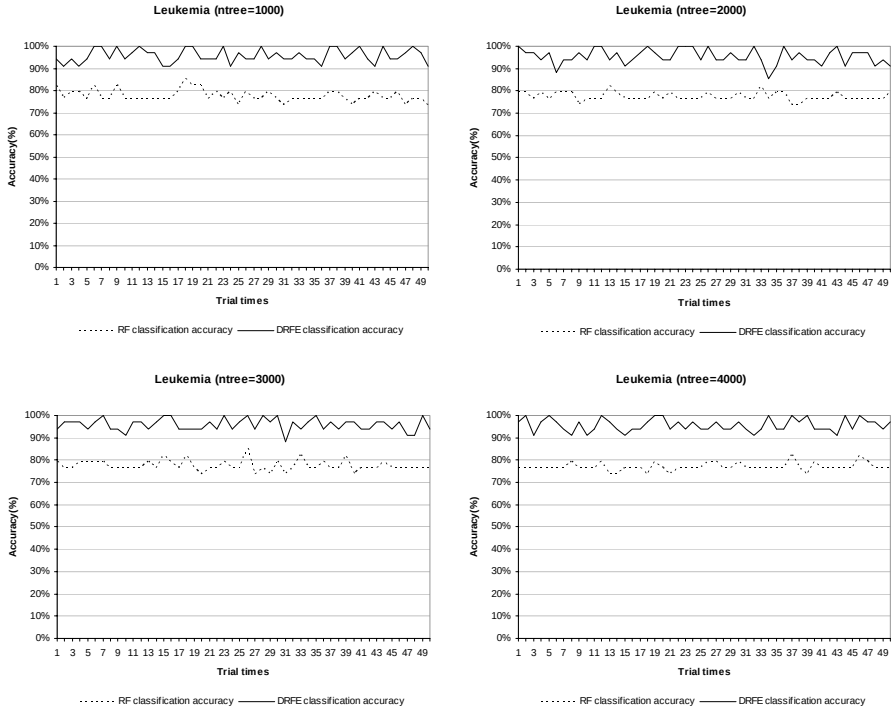


Fig. 4: The comparison of classification accuracy between DRFE (dash line) and RF (dash-dot line) via 50 trials with parameter *ntree* = {1000, 2000, 3000, 4000} in case of Leukemia dataset

The summary of classification results are depicted in Table 3. Table 4 shows the average number of selected features obtained from all experiments. In those experiments, the tree number parameters do not noticeably affect the classified results. We selected 50 as the number of feature elimination which is called *Step* parameter (Step=50). Our proposed method achieved the accuracy of 95.94% when performing on about 55 genes predictors retained by using DRFE procedure. This number of obtained genes only makes up about 0.77% (55/7129) of the whole set of genes.

Table 3. Classification results of leukemia cancer (average % classification accuracy ±standard deviation)

Tree number	1000	2000	3000	4000
RF only	77.59±2.6	77.41±1.9	77.47±2.5	76.88±1.9
DRFE	95.71±3.1	95.53±3.3	**95.94±2.7**	95.76±2.8

Table 4. Number of selected feature with different tree number parameter in case of Leukemia cancer over 50 trials (average number ± standard deviation)

Tree number	1000	2000	3000	4000
Number of selection features	147±21	138±41	55±21	74±51

And again, we compare the prediction results of our method and some other studies' performed on Leukemia dataset such as Weighted voting [8], Bootstrapped GA\SVM [10], Combined kernel for SVM [16] and Multi-domain gating network [19] (see Fig. 5). The Fig. 5 shows the classification accuracy of our method is much higher than these studies'.

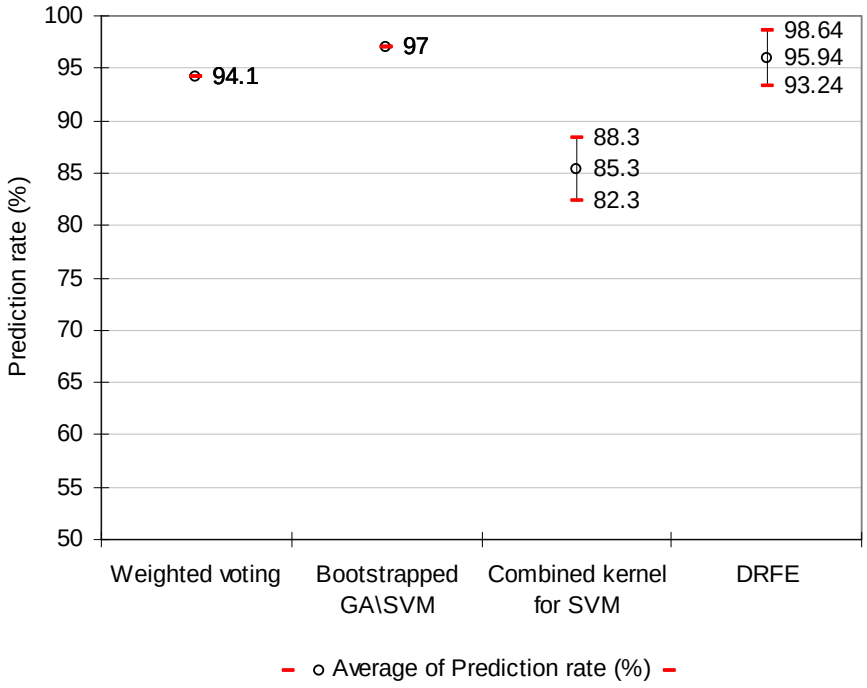


Fig. 5. The best prediction rate and it's standard deviation of some studies in case of Leukemia data set

7 Conclusions

In this paper, we introduced the novel method in term of feature selection. The RF algorithm itself is particularly suited for analyzing high-dimensional dataset. It can easily face with a large number of features as well as a small number of training samples. Our method not only employed RF by mean of conventional REF but also made

it fluently adapt to feature elimination task by using the DRFE procedure. Based on the defined ranking criterion and the dynamic feature elimination strategy, the proposed method obtains higher classification accuracies and more stable results than original RF. The experiments achieved a high recognition accuracy of 85.5%±4.5 when performing on Colon cancer dataset with only a subset of 141 genes and the accuracy of 95.94%±2.7 in case of Leukemia cancer using a subset of 67 genes. The experimental results also shown a significantly improvement in classification accuracy compare to the original RF algorithm especially in case of Leukemia cancer dataset.

Acknowledgement

This research was supported by RIC (Regional Innovation Center) in Hankuk Aviation University. RIC is a Kyounggi-Province Regional Research Center designated by Korea Science and Engineering Foundation and Ministry of Science & Technology.

References

1. Kohavi, R. and John, G.H.: Wrappers for Feature Subset Selection, Artificial Intelligence (1997) pages: 273-324
2. Blum, A. L. and Langley, P.: Selection of Relevant Features and Examples in Machine Learning, Artificial Intelligence, (1997) pages: 245-271
3. Breiman, L.: Random forest, Machine Learning, vol. *45* (2001) pages: 5–32.
4. Torkkola, K., Venkatesan, S., Huan Liu: Sensor selection for maneuver classification, Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (2004) Page(s):636 - 641
5. Yimin Wu, Aidong Zhang: Feature selection for classifying high-dimensional numerical data, Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2 (2004) Pages: 251-258
6. Duda, R. O., Hart, P. E., Stork, D. G.: Pattern Classification (2nd Edition), John Wiley & Sons Inc. (2001)
7. Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J.: Classification and Regression Trees, Chapman and Hall, New York (1984)
8. Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, J. P., Mesirov, J., Coller, H., Loh, M. L., Downing, J.R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E.: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," Science, vol. 286 (1999) pages: 531-537.
9. Fröhlich, H., Chapelle, O., and Schölkopf, B.: Feature Selection for Support Vector Machines by Means of Genetic Algorithms, 15th IEEE International Conference on Tools with Artificial Intelligence (2003) pages: 142
10. Chen, Xue-wen: Gene Selection for Cancer Classification Using Bootstrapped Genetic Algorithms and Support Vector Machines, IEEE Computer Society Bioinformatics Conference (2003) pages: 504
11. Zhang, H., Yu, Chang-Yung and Singer, B.: Cell and tumor classification using gene expression data: Construction of forests, Proceeding of the National Academy of Sciences of the United States of America, vol. 100 (2003) pages: 4168-4172

12. Doak. J: An evaluation of feature selection methods and their application to computer security, Technical Report CSE-92-18, Department of Computer Science and Engineering, University of Carlifornia (1992)
13. Das, S.: Filters, wrappers and a boosting-based hybrid for feature selection, Proceedings of the 18th ICML (2001)
14. Ng, A. Y.: On feature selection: learning with exponentially many irrelevant features as training examples", Proceedings of the Fifteenth International Conference on Machine Learning (1998)
15. Xing, E., Jordan, M., and Carp, R.: Feature selection for highdimensional genomic microarray data", Proc. of the 18th ICML (2001)
16. Mehta M., Agrawal R., Rissanen J.: SLIQ: A Fast Scalable Classifier for Data Mining, Proceeding of the International Conference on Extending Database Technology (1996) pages: 18-32
17. Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., and Levine, A.: Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays, Proceedings of National Academy of Sciences of the United States of American, vol 96 (1999) Pages: 6745-6750.
18. Nguyen, H.-N, Ohn, S.-Y, Park, J., and Park, K.-S.: Combined Kernel Function Approach in SVM for Diagnosis of Cancer, Proceedings of the First International Conference on Natural Computation (2005)
19. Su, T., Basu, M., Toure, A.: Multi-Domain Gating Network for Classification of Cancer Cells using Gene Expression Data, Proceedings of the International Joint Conference on Neural Networks (2002) pages: 286-289

3D Visualization of Brain Slices by Using Computer Techniques

Baki Koyuncu[1], Alper Pahsa[1]
[1]Computer Engineering Dept, Ankara University,
06500, Besevler, Ankara, Turkey
{bkoyuncu@ankara.edu.tr, apahsa@eng.ankara.edu.tr}

Abstract. In this study, A software was developed to obtain the 3D mesh of a Computerized Tomograhy (CT) image of a brain slice. Conventional CT brain slice image was digitized with a CT film scanner and the digitized image has been converted to a gray scale bitmap image. The gray scale intensity distributions of the bitmap image were stored in an ASCII intensity matrix and 2D intensity profiles of this matrix were plotted. Collection of these 2D intensity profiles were used in constructing a 3D mesh . Zooming, angular rotation , cropping facilities on the bitmap image were also included in the developed software.

Keywords: Intensity Distribution, Intensity Profiling, 3D Mesh graphs , Image Sampling, Image Cropping,

1 Introduction

From space to medical science research , there exist many examples of 3D visualization of 2D images. The aim of 3D visualization is to view the minute details from different orientations. For instance surface tissue structures which can not be seen on a 2D- CT image will be seen better in 3D form. Since 1972, CT imaging created an essential radiological solution for a wide range of clinical applications. It used X-rays to produce cross-sectional and two-dimensional images of the body tissues. Images were taken by rapid rotation of the X-ray tube around the patient. A ring of sensitive radiation sensors, located all around the patient, measured the radiation and a central system reconstructed the 2D image of a body slice from multiple X-ray projections [4].

Different approaches were used to generate 3D mesh from 2D CT image slices. For instance volume rendering algorithms , used with CT image systems, produces 3D mesh body structure of a patient. However complexity, memory usage and time consumption of these kind of algorithms are the side effects . Hence researchers need to develop simpler, fast and memory efficient 3D meshing algorithms.

The aim of this work was to develop a software to generate 3D mesh of the intensity distributions of a CT brain slice image . The development was in three steps :

a) Gray scale intensity distribution of a Bitmap brain slice image was converted into ASCII form. The intensity amplitude information in ASCII form was stored in a discrete matrix . Each entry of the matrix was the intensity amplitude of the image pixel at its spatial coordinate on the CT image[2].

b) Elements of the intensity matrix were plotted against their corresponding spatial coordinates to form 2D intensity profile graphs . Each 2D profile represented a row of intensity amplitudes along x direction on the CT image .

c) Sets of rows were plotted along y direction to generate the 3D mesh formation in XY plane of the CT image . The intensity amplitudes were along the Z axis and this procedure was called 3D intensity profiling of the image surface in XYZ coordinate system .

2 Theory

There are many methods to create 3D meshes. Most common meshing algorithms are volume rendering, surface graphics via triangulation and marching cubes. 3D mesh generation algorithms generally collected in two groups; structured meshes and unstructured meshes. A structured mesh is usually a warped grid of boxes, while an unstructured mesh is typically a triangulation. There are some advantages of structured meshes such as simplicity and suitability for multi grid and finite difference methods. Unstructured meshes conform to the domain more easily and allow element sizes to vary dramatically.[1]. On the other hand surface graphics meshing is generated via collection of polygon surfaces in closed boundaries. [10]

Volume rendering algorithms generate the objects in discrete 3D cubic forms. It is a technique for the exploration of datasets. When there exist no raw data, it is difficult to build up the right structured mesh.[10]

In marching cubes method everything is done over the volumetric element (voxel) information. In this method modification on the object voxels can be done with different meshing methods such as triangulation of distinct parts of the object starting from an interactive point[9]

In this study, an unstructured mesh technique is used to generate a 3D mesh surface by using the intensity amplitude distribution information.

2.1 Intensity Distribution

Images are denoted as a two dimensional functions of the form f(x,y). The amplitude of f at spatial coordinates (x,y) is a positive scalar quantity whose physical meaning is determined by the source image.[3]. f(x,y) has a nonzero and finite value. These functions are also known as gray level or picture elements (pixels).

Each pixel has a pixel value which describes its brightness level and/or its color. In the simplest images, (BMPs), the pixel values represent gray levels or intensity values of the image[8]. For gray scale images, intensity values are stored as an 8-bit integer from 0 to 255. 0 shows color black and 255 represents color white. Values in between give the different shades of gray.

2.2 Intensity Profiling

An image was defined as a 2D function of f(x,y) . [5]. This 2D function is also known as intensity amplitude function. When x,y and the amplitude values are finite or discrete quantities, the image is known as digital image. An image is digitized according to three parameters ; Pixel size, Number of gray levels and Loss of information.

Intensity profiling was used to construct the pixel intensity plots against pixel positions on a selected image row. A group of these rows were plotted with a finite distance between them to generate lines of intensity profiles of the image.

As in sampling and quantization theorem , intensity profiling use the same approach.

In sampling, equally spaced samples are taken on any row of the image. [3] These samples are then plotted against their corresponding locations on spatial coordinate axis to form a 2D intensity profile graph. An example intensity profile graph that was generated by the developed software was given in Fig.1.

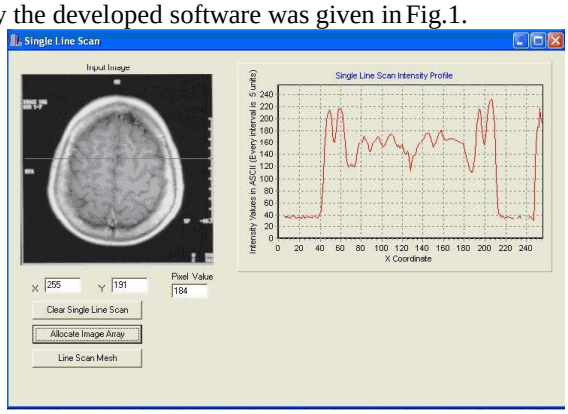


Fig.1 Single Intensity profile of a CT brain image

2.3 3-D Mesh Analysis

A mesh is a discretization of a geometric domain into simple elements such as partition of a polygon into small triangles or constructing a 3D graph from combination of 2D lines. Meshes find use in computer graphics, geographic information systems and medical sciences.[7] Although there exist many different mesh generation algorithms in literature it is generally true that a good mesh will have nicely shaped elements for accuracy. For a given 2D or 3D domain, a mesh is created by generating nodes and connections between these nodes [5].

In this study, 3-D mesh surfaces were constructed by using image intensity profiles. These meshes involved two kinds of data, spatial coordinates x , y and their corresponding pixel amplitude values. [4] A typical 3D mesh surface of a CT brain slice is shown in Fig.2 Developed software menu is presented in Fig.3 .

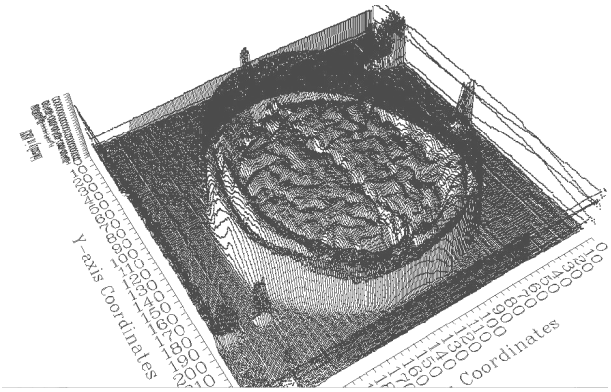


Fig.2 3-D Mesh plot of CT image in Figure.3

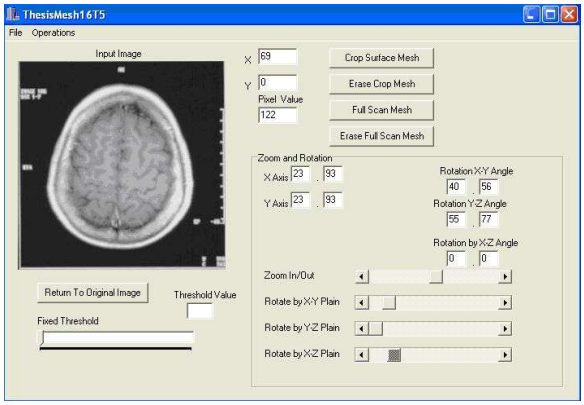


Fig.3 software menu

(showing an image of CT brain slice)

3 Procedure

In this study , a menu driven visual software was developed. Borland C++ Builder V5.0, application developer for MS Windows's 32bit platforms was used [6]. Every task in the software menu was designed according to end user view.

Every function in the menu displayed an event function. These functions were activated according to required parameters such as crop surface , full scan , zoom , rotate L/R etc …Extraction of intensity distribution, intensity profiling and 3-D mesh graphs of the intensity profiles were implemented with the software. The software read a digitized 256x256 pixel sized BMP image of CT brain slice from an image file. The image files had 8 bit black & white images with gray levels between 0 and 255.

The software scanned the bitmap file and generated a 256x256 array to store the intensity values of pixels in ASCII form . This was done by converting 8 bit intensity values into numerical forms between 0 and 255. Initially, a 2-D profile graph was plotted for each row of pixels against their X position coordinates on the CT image. Rows of 2-D profile graphs were plotted sequentially in the plot area. verification of the software for 2D plots was made with a black & white test pattern.2-D profile graph of the test pattern generated a single square pulse train which was consistent with the test pattern .Verification of the 3-D mesh plots were also checked with a test pattern. Test pattern was a black & white image of multiple black points on a white background. The expected 3-D mesh consisted cylindirical columns with a top surface . In the menu driven software , zooming , rotation and crop facilities were also included in the menu .

For zooming, points of 3D mesh was scaled according to the user request. If the user increased or decreased the zoom scale , intensity values on the intensity profile graphs interpolated themselves. Intensity values multiplied or divided themselves with incremental values. Hence the resultant 3D mesh graph expanded or contracted on the screen. For rotation , intensity profile graphs were rotated around the user specified XY, XZ or YZ planes according to the user defined angles between 0^0 and 360^0 in the menu. Software algorithm calculated the polar coordinate vectors and transformed the coordinate planes around the given angle. Polar coordinates were computed for x,y coordinates and z intensity points. ($x=r\cos\theta$, $y=r\sin\theta$, $z=\tan\theta$). During runtime , a new rotation matrix was generated for the pixel arrays and they were transitioned over the rotation matrix to their new coordinate location. For cropping , User marked (cropped) an area on the CT image and generated the 3D mesh plot of only this area . Rotation example was given for 60 degree in XY plane with 45^0 in XZ plane and zoom=20 in Fig.5. for an original CT image in Fig.4.

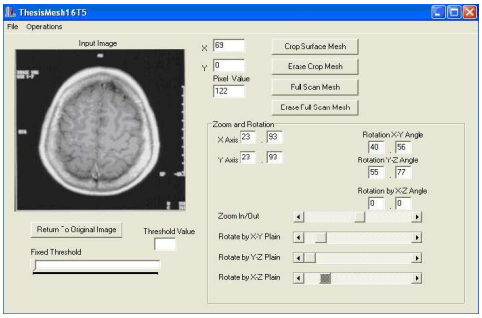


Fig.4 Original CT Image

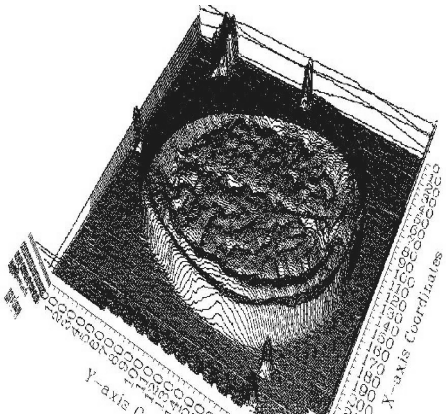


Fig.5 3D Mesh plot with 60 degree in XY plane

4 Results & Conclusion

A menu driven visual program was developed to display a 256x256 Bitmap image in 8bit gray scale . 256x256 sized 8 bit gray scale image was scanned and the pixel information was extracted in ASCI form. The software displayed the X ,Y position coordinates of the mouse motion together with the image pixel value at these coordinates.Pixel intensity amplitudes in ASCII form were plotted against the X coordinates and these plots were identified as 2-D intensity profile graphs.

3-D mesh graphs were also plotted by using the above 2-D profile graphs with an incremental distance between them. This incremental distance was used to increment the Y coordinates of the pixel values. Image intensity distribution information was used to generate 3D mesh graphs by using MatLab and developed software in this study [11]. MatLab used a surface graph algorithm to generate 3D meshes. Developed software used an unstructured meshing algorithm based on intensity profile lines to generate 3D meshes. A random cropped region from a CT brain slice image was selected and its mesh plots were generated by Matlab and the developed software. The examples were shown in Fig 6 and Fig 7.amples were shown in Figure 5 and Figure 6.

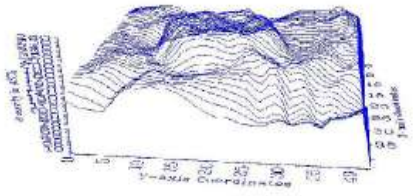


Fig.6 3-D Mesh plot of Cropped Region with developed software

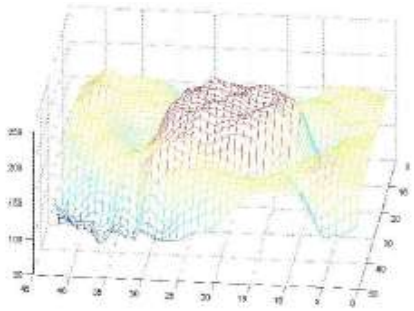


Fig.6 3-D Mesh plot of Cropped Region with MatLab mesh() function

As it was seen in the figures, 3D mesh generated by MatLab surface graphics had suppression effect on the actual data. This effect was multiplied for rotated meshes and it had to be corrected by advanced additional software . However developed software generated 3D mesh directly from input data without any modification. MatLab rotation capabilities always modified the 3D meshes either by pressing effect or by morphing effect on the shape whereas the developed software carried out the rotation of the 3D meshes without any modification. In conclusion , the developed software would be preferred in detailed mesh surface analysis .

Another advantage of the developed software was being a purpose built software platform. The extraction and conversion of pixel intensity amplitudes into ASCII form, 2-D intensity profiling, generation of 3D meshes were all in one package . It had additional facilities, zooming, rotating etc. like other software . This platform was open to include any other additional facility as the user required . It was very convenient to visualize a 2D intensity image in a 3D mesh formation. This software was a general platform and it could also be used in many other image processing areas. Intensity distributions could be displayed for any optical event in 3-D graphical form. This way any anomality which could not be seen by simply looking at the 2D image would now be observed as a topological identity on the mesh graphs .

There were many algorithms which were used for 3D meshing in diagnosis of malignant tissues and different tissue characterizations. CT brain imaging was used for 3D mesh construction. In literature most common algorithms used for 3D meshing were surface graphics via triangulation, volume rendering and marching cubes.

3-D meshing algorithms of surface graphics (OPEN GL) had advantages among the other meshing algorithms. Disadvantages of these algorithms were the exclusion of the object's inner structures and maintaining only the object's outer shell. Cutting, slicing and disection operations could not be applied on surface 3D mesh forms. Artificial viewing like semitransparent structures could not be seen in surface 3D meshes.Intensity distribution based 3D meshing had some advantages over the surface graphics 3D meshing . It was seen that both algorithms were generating

3-D meshes but Cropping, slicing and disection operations were more prominent in intensity distribution matrices due to the fact that these matrices could be divided into sub intensity distribution matrices. Length and size of the sub intensity distribution

matrices depended on the user's region of interest . Artificial viewing was possible in intensity distribution based 3D mesh graphs. Consequently , hidden details of the objects that could not be seen on 2D intensity image was obvious in 3D mesh plots.

Volume rendering 3D mesh algorithms provided full discrete 3D views of the objects in more details. 3D meshes produced with the volume rendering algorithms had results close to human eye perspec- tive. But this technique used large computer memory and processor time during the operations. It was one of the reasons to favor the intensity distribution based 3D mesh plots in the software . In this study a simple, time and memory efficient technique was used to build a 3-D mesh from a CT brain slice image.

Considering the complexity, memory efficiency and menu facilities ; The purpose built software presented here could be used in many biomedical applications such as diagnosis, surgical operations and therapeutically purposes.

References

1. Seth, "Mesh Generation" , 1996, Graphics at MIT, web site source: http://people.csail.mit.edu / seth/pubs/taskforce/section3_7.html1
2. Seth, "Mesh Generation" , 1996, Graphics at MIT, web page source "www.ise.ufl.edu/ahuja/PAPERS/ Romeijn- Ahuja-IMRT-OR2005.pdf"

site source: http://people.csail.mit.edu / seth/pubs/taskforce/section3_7.html

3.Gonzales C. R. , Richard E. W., "Digital Image Processing", second edition, 2001, Prentice Hall publications pp. 52-53

4.Glasbey, C. A. , Robinson C. D., "Estimation of tissue proportions in X-ray CT images using a new mixed pixel distribution", web page source: http://www.bioss.sari.ac.uk/image/task.pdf

5.Heckbert P., "Mini Glossary of Mesh Gene- ration", web page source: http://www.cs.cmu.edu /~ph/meshgloss.html

6.Borland C++ Builder 5.0 Enterprise Suite Version 5, 1983-2000, Borland Corporation, web site:http://www.borland.com

7.Cetin Nurhan, "Mesh Generation", 2000, web page source: http://www.inf.ethz.ch/personal / cetin/thesis/thesis/node18.html

8.Fisher R., Perkins S., Walker A., Wolfart E. , "Pixel Values", 2003, HIPR2, web site source: http://homepages.inf.ed.ac.uk/rbf/HIPR2/value.htm

9.Sakas Georgios, "3D Visualisation in Medical & Biological Applications", Fraunhofer Institute for Computer Graphics, Darmstadt, Germany, 2002, web page source: http://www.gdv.informatik.unifrankfurt.de/events/BestPractice/011123_Visualisierungstechnologien_Medizin_Pharmazie/Vortraege/Sakas.pdf

10.Mueller Klaus, "CSE:564 Scientific Visualisation, Lecture 10: Introduction to Volume Rendering", Stony Brook University Computer Science Department, 2003, web page source: http://www.cs.sunysb.edu/~mueller/teaching/cse564/volumeIntro_2006.pdf

11.Mesh, meshc, meshz, Matlab Function Reference, 1994-2006, MathWorks Inc., web page source: http://www- ccs.ucsd.edu/matlab /techdoc /ref/mesh.html

Analysis Of The Features Extracted from Sequence for Prediction of Protein's Subcellular Localization Using Fourier Transform

Guoqi Li[1] and Huanye Sheng[1]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University.
800 Dong Chuan Rd., Shanghai 200240, China
{liguoqi, hysheng}@sjtu.edu.cn

Abstract. The identification of a protein's subcellular localization(SCL) is valuable for biology research. Many efforts have been focused on how to predict the protein's SCL by computational method. It is necessary to analyzed the features extracted from protein sequences by Fourier transform to look for clues for deeper research. We divided the features extracted from sequence using Fourier transform into four frequency domain ranges and used KNN(k-nearest neighbor) method to test the classification ability of features in each frequency domain range. We were surprised that the highest and lowest frequency domain ranges are crucial in the classification. The result has meanings in both computational and biological view. It can used to reduce the dimension of features extracted from sequence using Fourier transform and also give some clues to discover the mechanism of protein's SCL. It also show that the frequency domain analysis is a valuable tool in the research of SCL.

1 Introduction

Identification of a bacterial protein's subcellular localization (SCL) provides valuable clues regarding its biological function. For example, surface-exposed or secreted proteins are of primary interest due to their potential as vaccine candidates, diagnostic agents (environmental or medical) and the ease with which they may be accessible to drugs. [1]. The high throughput genome sequencing projects are producing an enormous amount of raw sequence data, which need to be cataloged and synthesized. Since experimental determination of subcellular location is time consuming and costly, a computational, fully automatic and reliable prediction system for protein's SCL method is very useful.

Many efforts have been focused on how to predict the protein's SCL by computational method and have made great achievements on the performance of prediction. The computational prediction needs two steps. Extract feature vectors and then input them to classifier. Of course, the classifier must be trained by inputting feature vectors with known SCL at first. Various pattern classification and machine learning methods have been used, such as Mahalanobis distance [2], neural network [3], hidden Markov model (HMM) [4] and support vector machine [5]. It is

reasonable that the feature extraction method is more important in the problem, because it is close to the biological essential. Researchers have developed a lot of feature extraction methods. Most of these methods can be classified into two categories: one is based on information encoded in the sequences and the others based on function domain composition[6] or ontology methods[7]. With the help of the second kind of features the prediction accuracy was improved significantly. But the shortcoming of them is also obvious. Only a part of proteins' corresponding information can be available. It is difficult to get knowledge like functional composition and gene ontology for new sequences. Extracting features from protein sequence data is the more popular and reasonable method. Besides amino acid composition, Emanuelsson et al. made use of the N-terminal sorting signals [8], which is an efficient method but depends strongly on the leader sequences and often makes mistake when the leader sequences are unreliable. Chou introduced the quasi-sequence-order approach [9] and pseudo-amino-acid-composition [10] to incorporate sequence order information. Other methods combine new features, such as hydrophobic [11] information and Zp parameters [12].

Although human beings have accumulated many knowledge in the problem of protein's SCL, the biological mechanism of protein's CSL is still unknown. All the former methods only extracted features from some specific points. However, biological characters sometimes depend on the influences of small quantity molecules or local features, in computational view, which are faint signals. it is need to analyzing all the features in every dimension and select crucial points for the classification. The frequency domain analysis can meet the requirements. Zhengdeng Lei and Yang Dai have presented a method to extract the features using Fourier transform[13]. Our research is based on their result. We divided the features extracted from sequence using Fourier transform into four frequency domain ranges and used KNN(k-nearest neighbor) method, a simple classifier to test the classification ability of features in each frequency domain range. We were surprised that the highest and lowest frequency domain ranges are crucial in the classification for single localization site protein's SCL. The result has meanings in both computational and biological view. It can used to reduce the dimension of features extracted from sequence using Fourier transform and also give some clues to discover the mechanism of protein's SCL. It also show that the frequency domain analysis is a valuable tool in the research of SCL.

2 Dataset used in the research

The dataset we used in the research was gotten from PSORTdb. It is a database of SCL for bacteria that contains both information determined through laboratory experimentation (ePSORTdb dataset) and computational predictions(cPSORTdb dataset)[1]. The set of proteins from Gram-negative bacteria used in ePSORTdb is considered in this work. It consists of 1597 proteins with experimentally determined localizations(until December, 2005), in which 1448 proteins are of single localization site. Just single localization samples were selected, because they are easy to test in order to deduce the conclusion. And more, for convenience in the computing, 21

proteins whose amino acid length are longer than 1024 were discarded. Now the dataset is 274 cytoplasmic, 305 cytoplasmic membrane, 183 extracellular, 388 outer membrane and 277 periplasmic.

3 Extract features using Fourier transform

Original protein sequences are strings with the alphabet of 20 characters. before convert them into a sequence in the frequency domain with Fourier transform, they should be encoded into a numerical format. There are many ways to describe amino acids, most of which are correlated to some degree. For example, the AAindex database contains indices representing 434 different physicochemical and biological properties of amino acids. We concentrate on the amino-acid hydrophobicity in this work, as it is the one of major properties influencing the structure and function of a protein. A simple three-state hydrophobicity scale is used to map hydrophobic residues to 1, hydrophilic residues to -1, and "neutral" residues to 0. More precisely,[13]

(A, C, F, I, L M, V)->1,(D, E, H, K, N, Q, R)->-1, and (G, P, S, T, W, Y)->0.

Prior to the application of the Fourier transform, the numerical sequences have to be lengthened by padding with zeros, since the length of the input sequences is required to be a power of two. Let denote the smallest number that is greater than or equal to the length of the longest protein sequence in a given set, where is some integer.[13] In this research , for the longest length of sequence is limited to 1024. Let be the numerically encoded sequence after padding. The discrete Fourier transform will transform it into another sequence $\{X(1),\cdots,X(n)\}$:

(**1**)

and (**2**)

We used the Fast Fourier Transform(FFT) function in Matlab to do the implementation. $\{X(1),\cdots,X(n)\}$ is a complex number vector. Let is the complex modulus of , and is the feature vector for protein's SCL. It is clear than the number of extracted features is almost the same as the length of the most longest sequence in the data. Accurately in this experiment, it is 1024.

A common use of Fourier transform is to find the frequency components of a signal buried in a noisy time domain signal. And Fourier transform is a traditional method to extract features from signal in pattern recognition. In the problem of predicting protein's SCL, the biological mechanism is still unknown. If only select

features from some specific points, the features can't cover all the information of protein sequence usable for SCL. Maybe it can be said that concentrating on the amino-acid hydrophobicity is also unilateral, but in fact all the relationship between every amino acid is considered, although the information is simplified. Simplification is inevitable in feature extraction and you can use other concentrating method without changing the following analysis. Fourier transform is a good idea for extracting features in the SCL prediction, but it is not enough just select all the frequency domain range as a feature vector. In some sense, biological characters sometimes depend on the influences of small quantity molecules or local features, in computational view, which are faint signals. It is need to analyzing all the features in every dimension and select crucial points for the classification. The whole frequency domain range vector can represent all the character of proteins, but what we need is just the part who is usage for the classification of the proteins on the SCL.
So we divided the features extracted from sequence using Fourier transform into four average frequency domain ranges. We call them 1 to 4 range from high to low in the frequency domain. Then input them to a classifier respectively to find their ability for the classification. We used KNN(k-nearest neighbor) method as classifier.

4 KNN method used in the test

KNN(k-nearest neighbor) method is a basic classifier. The algorithm is simple and clear. Firstly, select labeled samples and add them to multi-dimensional space. Then find neighbors of the test sample and the test sample belongs to the class which has maximum members in the test sample' s neighbors. KNN method has various implementation. To described clearly, crucial points of the algorithm are given program list.

In this research, the feature vectors just should be tested and find out their classification ability. So we selected 150 sequences from every kind of SCL samples arbitrarily. All the feature vectors of the four frequency range were tested respectively. The following description is concentrate to one of the four frequency range vectors. Others are similar.

Let is one of the feature vector.

is feature matrix. From top to bottom, there are 150 feature vectors belong to cytoplasmic, cytoplasmic membrane, extracellular, outer membrane and periplasmic respectively.

Then calculate the distance between every vectors. Let the distance is:

(3)

We calculate the distance matrix with the following Matlab program.

```
%A is feature matrix.
column = 750;
row = 256;
y = 0;
B = zeros(column, column);
for i = 1: column
  for j = i: column
    x = (A(i,:) - A(j,:)).^2; %select the i-row of two
      for k = 1:row
        y = y + x(k);
      end
    B(i,j) = y;
    B(j,i) = y;
    y = 0;
  end
end.
```

The distance matrix is square and is . It is clear that B is symmetric and the diagonals is zero vector. And then the elements from every row vector of B is sorted from small to large on its number. Let the result matrix is . The first column of is zero vector of course. Discard it and get the .

Now, for th row vector of the Matrix , look vector as test sample and the rests are training samples.

Then can test the classification using the following Matlab program:

```
row=750; D = zeros(row, row-1)
for i = 1:row
  num = zeros(1,5);
  for j = 1:row-1
      if C(i, j)<151
        num(1)=num(1)+1;
      elseif (C(i,j)>150)&&(C(i,j)<301) %150<C(i,j)<300
        num(2)=num(2)+1;
      elseif (C(i,j)>300)&&(C(i,j)<451) %300<C(i,j)<451
        num(3)=num(3)+1;
      elseif (C(i,j)>450)&&(C(i,j)<601) %450<C(i,j)<601
        num(4)=num(4)+1;
      elseif (C(i,j)>600)&&(C(i,j)<751) %600<C(i,j)<751
        num(5)=num(5)+1;
      end
      index = 1; %mark the maximum
```

```
        index2=1; %mark the second maximum
        for t=2:5
          if num(ret)<num(t);
            index 2= index;
            index = t;
          end
        end
        if (index ~= index2)&&(num(index)==num(index2))
          D(i,j-1)=0;
        else
          D(i, j-1)= index;
        end
    end
  end
```

$D_{750\times749}$ is the classification result matrix.

5 Result and conclusion

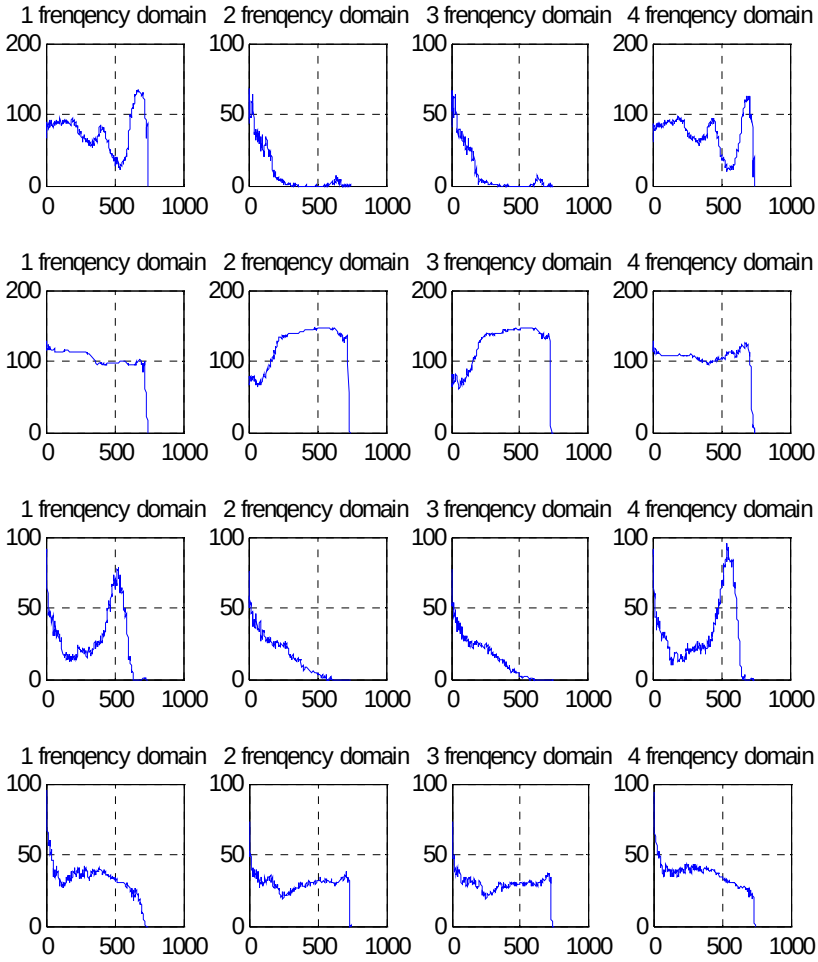
The following figure 1 show the testing results. each row presents a subcellular location, referring to cytoplasmic, cytoplasmic membrane, extracellular, outer membrane and periplasmic respectively. In the subgraph, the X axis means the k in the KNN algorithm and the Y axis represents the number of positive ones in the total 150 samples in the subclass. Detail data and program are available by email.

The result show that the feature vectors of highest and lowest frequency domain ranges have similar characters in the classification of single labeled protein's SCL. It is similar too that the characters of feature vectors of the two middle frequency domain ranges. The features in highest and lowest frequency domain ranges have more classification ability when k is not too big. In global view the two terminal ranges are more suit to be used as features in the prediction of protein's SCL. The information maybe used to reduce the dimension of features extracted from sequence using Fourier transform. And from the other view, the frequency domain characters of the 5 kinds subcellular location sample are not same. Maybe we should develop specific classification method according the typical subcellular location. In fact, the popular used tools for prediction of protein's SCL had integrated many strategies designed individually, so that all the advantages could be taken. Hope our research can promote these tools. And what's more, we will attempt to enhance the performance of the prediction of protein's SCL in the following research.

The result also has meanings in biological view. It can give some clues to discover the mechanism of protein's SCL. If the features in highest and lowest frequency domain ranges have more classification ability, we can imagine that the maximum features and minimal ones are crucial for classification. On the contrary, the middle dimension features are similar in every kind of subcellular location. With the clue gotten from the above analysis, we give a hypothesis to describe the mechanism of protein's SCL. The hydrophobicity is the one of major properties influencing the structure and function of a protein [13] . The features of lowest frequency domain ranges represent global structure of proteins, such as the shape of a

kind of polygon or a snowflake. And the features of highest frequency domain ranges represent the texture of proteins. The global shape and texture are crucial for the protein's SCL. Consider the pattern recognition problem in the macroscopical world. How to distinguish furniture from books? Important features are global structure and texture. However, the middle dimension features are useful to distinguish the individual from subclasses. For example from the middle dimension features we can easily find out what is table and what is chair, but furniture and books have definitely different global shape and texture. The middle dimension features are noise signal in the pattern recognition problem. Is similar mechanism in the macroscopical world or in molecular level? It seems so according to the above analysis.

With the help of the refined analysis qualification and the mature tools of frequency domain analysis, bioinformatics researches could mining more treasure underlying in the mess data.



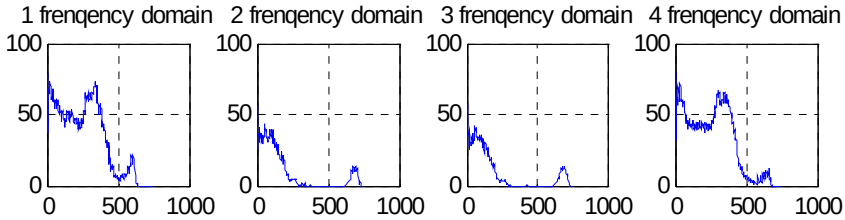


Fig. 1. each row presents a subcellular location, referring to cytoplasmic, cytoplasmic membrane, extracellular, outer membrane and periplasmic respectively. In the subgraph, the X axis means the k in the KNN algorithm and the Y axis represents the number of positive ones in the total 150 samples in the subclass.

6 Acknowledgments

The research is supported by the doctoral fund of Education Ministry of China. Project No. 20040248001.

References

1 PSORT-DB: Rey, S., M. Acab, J.L. Gardy, M.R. Laird, K. deFays, C. Lambert, and F.S.L. Brinkman (2005). PSORT-DB: A Database of Subcellular Localizations for Bacteria. Nucleic Acids Research. 33:D164-168. (Database issue)

2 J. Cedano, P. Aloy, J.A. Perez-Pons and E. Querol, "Relation between amino acid composition and cellular location of proteins," J. Mol. Biol., vol. 266, February 1997, pp. 594-600.

3 A. Reinhardt and T. Hubbard, "Using neural networks for prediction of the subcellular location of proteins," Nucleic Acids Research, vol. 26, 1998, pp. 2230-2236.

4 Y. Fujiwara, M. Asogawa and K. Nakai, "Prediction of mitochondrial targeting signals using hidden Markov models," Genome Informatics, 1997, pp. 53-60.

5 S. Hua and Z. Sun, "Support vector machine approach for protein subcellular localization prediction," Bioinformatics, vol. 17, 2001, pp. 721-728.

6 K. C. Chou and Y. D. Cai, "Using functional domain composition and support vector machines for prediction of protein subcellular location," J. Biol. Chem., vol. 277, 2002, pp. 45765-45769.

7 K. C. Chou and Y. D. Cai, "A new hybrid approach to predict subcellular localization of proteins by incorporating gene ontology," Biochemical and Biophysical Research Communication, vol. 311, 2003, pp. 743-747.

8 O. Emanuelsson, H. Nielsen, S. Brunak and G. von Heijne, "Predicting subcellular localization of proteins based on their N-terminal amino acid sequence," J. Mol. Biol., vol. 300, 2000, pp. 1005-1016.

9 K. C. Chou, "Prediction of protein subcellular locations by incorporating quasi-sequence-order effect," Biochemical and Biophysical Research Communications vol. 278, 2000, pp. 477-483.

10 K. C. Chou, "Prediction of protein cellular attributes using pseudoamino-acid-composition," Proteins, vol. 43, 2001, pp. 246-255.

11 Z. P. Feng and C. T. Zhang, "Prediction of the subcellular localization of prokaryotic proteins based on the hydrophobicity index of amino acids", Int. J. Biol. Macromol., vol. 28, 2001, pp. 255-261.
12 Z. P. Feng and C. T. Zhang, "A graphic representation of protein sequence and predicting the subcellular localizations of prokaryotic proteins," Int. J. Biochem. Cell Biol, vol.34, 2002, pp. 298-307.
13 Z. Lei and Y. Dai, "A novel approach for prediction of protein subcellular localization from sequence using fourier analysis and support vector machines," Proc. of 4th ACM SIGKDD Workshop on Data Mining in Bioinformatics, Seattle, August 22, 2004, pp. 11-17.

Evolving Oblique Decision Trees For Survival Analysis

Christian Setzkorn, Azzam F. Taktak, Bertil Damato, Jessica Grabham

Royal Liverpool University Hospital, UK

Abstract. Survival analysis, which is also known as failure time analysis, is an important research area. Its main objective is to extract models from data that approximate lifetime distributions. These models can be used to predict, for example, the probability of events occurring to systems depending on their features and time. Methods of survival analysis can also be used to determine whether particular groups of systems exhibit different survival/failure behaviour. For example, it is important to quantify the difference between a treatment and placebo group. Given a dataset one might also ask whether it contains unknown groups with different survival behaviour. The discovery of as yet unknown patient/customer groups, with different survival/churning behaviour could help to formulate new 'treatments' to improve health or increase profits. This paper proposes a multi-objective evolutionary approach that automatically determines groups with different survival behaviour. Oblique decision trees describe the groups to improve their transparency. The approach was successfully evaluated on several benchmark datasets and a medical dataset.

1 Introduction

A plethora of publications show that evolutionary algorithms (EAs) are valuable approaches for extracting models from data. They have been used, for example, to tackle regression and classification problems whilst overcoming some of the shortcomings of existing approaches. The interested reader is referred to [28] for an extensive review of evolutionary approaches for the problem of classification. Surprisingly, however, only very few evolutionary approaches have been proposed to tackle survival analysis problems.

One aim of survival analysis is to model life/failure time distributions to approximate the probability of an event occurring to a systems (e.g. patients) depending on the time and features of the system [18]. Features could be, for example, demographic information and/or physiological information. Events may correspond to the recurrence of a disease or death. To model lifetime distribution, or conversely the probability of survival, has a number of benefits. It allows clinicians to devise suitable treatment regimes and counsel patients about their prognosis. Hence, it helps patients to plan their lives and provide future care for their dependents. Survival analysis is also widely used in the social and economic sciences, as well as in engineering. Here the systems being observed are, for example, machines or customers. An event might be the failure of a machine or the churning of a customer. Survival analysis is therefore also referred to as failure time and reliability analysis in these domains [1, 17].

Previous papers have shown that EAs can successfully be used to approximate lifetime distribution [31, 32]. This paper, however, focuses on another important task of survival analysis. It proposes a method to automatically determine groups of, for example, patients that exhibit different lifetime distributions. These groups are described by oblique decision trees, which are fitted using a multi-objective evolutionary algorithm. The determination of subsets with different survival behaviour is tremendously important. It may help to assign particular treatments and limited resources more appropriately and thus save life and money. Furthermore, it may generate new insights.

The difference between lifetime distributions can be measured with the log-rank test [24], which is illustrated using a dataset taken from [10]. It contains the survival times of 42 leukaemia patients depending on a dichotomous feature that indicates whether or not the patient received a particular treatment. Table 1 contains the data separated according to the treatment feature.

Group 1 (Treatment)	**Group 2** (Placebo)
6, 6, 6, 7, 10, 13, 16, 22, 23, 6+, 9+, 10+, 11+, 17+, 19+, 20+, 25+, 32+, 32+, 34+, 35+	1, 1, 2, 2, 3, 4, 4, 5, 5, 8, 8, 8, 8, 11, 11, 12, 12, 15, 17, 22, 23

Table 1. Leukaemia data taken from [10]. The numbers correspond to the survival times in weeks after the patient entered the study. The plus sign indicates censoring.

The maximum time horizon of the study (the follow up time) is 35 weeks. Patients shown without plus signs died during the follow up time, whereas patients shown with plus sign were censored. In essence, censoring occurs if one knows the status of the patient only for a particular period of time, but not for the complete follow up time. Figure 1 depicts the survival curves of the two patient groups. They were estimated using the Kaplan-Meier method. Its computation is summarised in equation 1.

$$\hat{S}(t) = \prod_{j=1}^{k}(\frac{n_j - d_j}{n_j}) = \prod_{j=1}^{k}(1 - \frac{d_j}{n_j}) = \hat{S}(t-1)(1 - \frac{d_t}{n_t}) \tag{1}$$

Here d_j corresponds to the number of events (*e.g.* deaths) at time j where one or more events occurred and n_j corresponds to the number of objects (*e.g.* patients) that are still observed at time j. In the medical domain, these patients are also called the risk set. Given the structure of equation 1, it is not surprising that the Kaplan-Meier method is also often referred to as the Product-limit estimator.

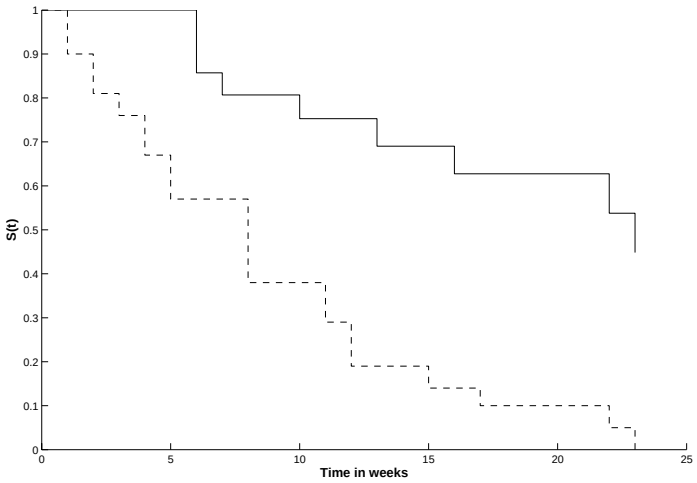


Fig. 1. Kaplan-Meier curves for the Leukaemia data summarised in Table 1. The solid line corresponds to the Kaplan-Meier curve for patients who were treated and the dashed line to those without treatment.

Figure 1 shows that the treatment group has a better survival experience in comparison to the placebo group. As mentioned earlier, this difference can be quantified using the log-rank test. The data shown in table 1 achieve a log-rank statistic of 16.79 when grouped according to the dichotomous feature *treatment*. The computation of the log-rank test for G groups is summarised in equation 2.

$$L = d^{'} V^{-1} d \tag{2}$$

Vector d corresponds to the sum of the k vectors for the event times: $j = 1, 2, \ldots, k$. [1] The $G-1$ values for each vector are computed according to equation 3.

$$d_j = (O_1 - E_1, O_2 - E_2, \ldots, O_{G-1} - E_{G-1}) \tag{3}$$

The difference between observation O_i and expectation E_i for event time j for group i is computed according to equation 4.

$$O_i - E_i = \sum_{j=1}^{k} m_{ij} - (\frac{n_{ij} m_j}{n_j}) \tag{4}$$

Here m_{ij} denotes the number of events at event time j in group i, n_{ij} the number of samples at risk at event time j in group i, m_j the number of events at event time j, and n_j the number of sample at risk at event time j.

Matrix V corresponds to the sum of k variance matrices for the event times. Each matrix consists of $i = 1, 2, \ldots, G-1 \times l = 1, 2, \ldots, G-1$ elements, which are computed according to equation 5.

$$V(i,l) = \begin{cases} \sum_{j+1}^{k} \frac{n_{ij}(n_j - n_{ij}) m_j (n_j - m_j)}{n_j^2 (n_j - 1)} & i = l \\ \sum_{j+1}^{k} = \frac{-n_{ij} n_{lj} m_j (n_j - m_j)}{n_j^2 (n_j - 1)} & i \neq l \end{cases} \tag{5}$$

One obvious requirement of the log-rank statistic is that the samples are already assigned to groups. However, in some situations it might be beneficial to assign the samples to groups automatically. This task is similar to unsupervised learning/clustering, where the samples are automatically classified/clustered. However, instead of finding clusters that maximise the intra-cluster similarity of samples, and minimise their inter-cluster similarity, one is interested in finding subgroups of samples that maximise the log-rank statistic.

This paper uses a multi-objective evolutionary algorithm (MOEA) to search for oblique decision trees that maximise the log-rank statistic. A similar MOEA has already been used successfully to tackle classification problems [30] and to estimate lifetime distributions [31, 30]. Oblique decision trees were chosen as they describe mutually exclusive subgroups within the data while overcoming some of the shortcomings of standard decision trees [14].

MOEAs are powerful optimisation algorithms, which can optimise several incommensurable objectives without making any assumptions about their importance. This is important in the context of model extraction, which is a multi-objective problem. It has at least two objectives: to maximise the fit of the model to the data and secondly to minimise its complexity. In this particular case 'model fit' refers to the log-rank statistic of the model (it has to be maximised). Simple models are required to understand the data generation process or in this case the produced groups. In fact, it is often argued that only simple models are adopted in practice due to their transparency [8, 15, 27]. The extraction of simple models is also important because complex models tend to require longer execution times, and more storage space. Overfitting is also a problem when one tackles classification and regression problems. However, it has to be noted that the problem being tackled is different to classification and regression problems.

There are additional reasons for the preference towards MOEAs for model extraction. For example, MOEAs are less prone to feature interactions and can cope better with noisy data. This is in contrast to other greedy search algorithms [7, 11, 33]. In addition, MOEAs can extract several models (trade-off solutions for the given objectives) in a single run. This is because MOEAs deploy a number of candidate solutions/models to explore the search space [25]. This

[1] If there are more than two groups, d corresponds to a vector and V to a matrix, where $d^{'}$ is the transpose vector and V^{-1} an inverse matrix.

has the advantage that, if the preferences of the decision maker changes, (s)he could choose another trade-off solution/model from the solution set (*e.g.* (s)he might prefer models with a higher log rank statistic over simpler models). This saves valuable (computational) resources, because the search does not have to be repeated.

2 The Implemented Evolutionary Approach

This section describes the implemented MOEA. It begins with a brief summary of the algorithm and then details its components. Figure 2 depicts the structure of the algorithm. It is similar to other evolutionary algorithms (*e.g.* [25]).

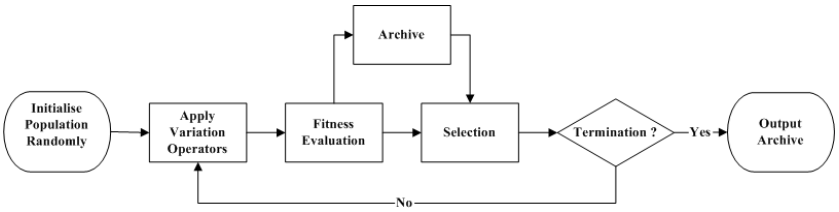


Fig. 2. Structure of the implemented MOEA.

In broad terms, the algorithm proceeds as follows. First, a number of candidate solutions/individuals (*i.e.* a population of oblique decision trees) are initialised randomly. After this, the variation operators are applied to some of the decision trees to recombine and/or change them. The fitness evaluation determines the performance (fitness/objective values) of each oblique decision tree.

The selection process generates a new population of individuals by sampling from the current population and the archive. The archive stores the best (elite) individuals found by the MOEA. This prevents the loss of good candidate solutions due to the randomness of the selection process [39]. The use of an archive is a form of elitism, which can help to create better individuals [6]. In fact, the implemented MOEA uses an archiving strategy that ensures diversity within the population and prevents premature convergence of the algorithm [21]. The selection process is followed by the termination test, which either terminates the MOEA or transmits the current population (generation) to the process that applies the variation operators.

Better individuals (decision trees) will be produced over time, as the above sequence is repeated several times. If the MOEA terminates (*e.g.* after a maximum number of 'generations') the decision trees within the archive are presented as the final output of the system. Here follows a more detailed description of the components of the MOEA.

2.1 The Representation Scheme

The representation scheme corresponds to a tree as shown in Figure 3. It was inspired by the representation scheme of genetic programming (GP) [3, 19], which has already been successfully used to induce decision trees from data. However, these decision trees were only used to tackle classification and regression problems [2, 12, 26, 37, 38].

Each non-terminal node (circles in Figure 3) corresponds to a linear combination of the features. It returns the value of its left child, if the value of the linear combination is greater or equal zero, otherwise it returns the value of the right child. Terminal nodes (squares in Figure 3) correspond to integer values that indicate the group membership. The tree is initialised randomly. A predefined lower and upper bound restrict the number of nodes. The coefficients of the linear combination are sampled from a predefined interval with a uniform probability.

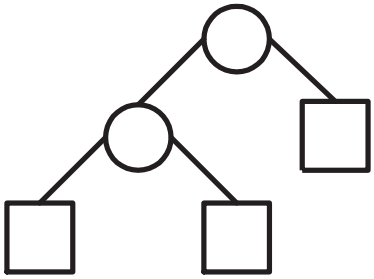


Fig. 3. Basis function tree.

The representation scheme also contains a binary vector that determines whether or not a feature is used within the linear combinations of the non-terminal nodes. This vector is initialised randomly (each value can be either zero or one with equal probability). This enables the algorithm to perform an implicit feature selection.

2.2 The Fitness Evaluation

The implemented MOEA currently optimises four objectives and deploys the fitness assignment of the second Strength Pareto Evolutionary Algorithm (SPEA2) [39]. The first objective has to be maximised whereas the other objectives have to be minimised.

- Objective 1: log-rank statistic of the decision tree (see equation 2 in section 1.
- Objective 2: number of features that the decision tree uses.
- Objective 3: number of nodes within the decision tree.
- Objective 4: number of different terminal node integers.

To assign a scalar fitness to an individual with several objective values, the fitness evaluation makes use of the Pareto dominance relation, which is explained in definition 1.

Definition 1. ***(Pareto Dominance Relation)*** *A solution x_1 is said to dominate a solution x_2, also expressed as $x_1 \succ x_2$, if x_1 is at least as good as x_2 in all objectives and better with respect to at least one objective. This can be expressed more formally as:* $\forall i \in \{1 \dots n\} : f_i(x_1) \leq f_i(x_2) \wedge \exists j \in \{1 \dots n\} : f_j(x_1) < f_j(x_2)$

The deployment of the Pareto dominance relation during the fitness selection enables the algorithm to optimise several incommensurable objectives without making any assumptions about their importance. The fitness $F(i)$ of an individual i is computed according to equation 6 [39].

$$F(i) = R(i) + D(i) \tag{6}$$

The value of $R(i)$ captures dominance information (see equation 7 and 8) and $D(i)$ captures density information (see equation 9) of the i-th individual.

$$R(i) = \sum_{j \in P_t + \overline{P_t}, j \succ i} S(j) \tag{7}$$

$$S(i) = \mid \{j \mid j \in P_t + \overline{P_t} \wedge i \succ j\} \mid \tag{8}$$

Here, P_t and $\overline{P_t}$ refer to individuals from the population and the archive respectively. The expression $i \succ j$ denotes the dominance relation between individuals i and j. Equation 7

determines the strength of the dominators of the i-th individual. A high value means that the i-th individual is dominated by many individuals, which in turn dominate other individuals. If the value of R_i is zero the individual i is non-dominated. The density information is computed according to Equation 8 and is an adaptation of the k-th nearest neighbour method [34].

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{9}$$

Here, σ_i^k is the Euclidean distance between the objective values of the k-th and the i-th individual. The value for k is equal to the square root of the sample size: $k = \sqrt{N + \overline{N}}$ [34]. The values N and $\overline{N}$ denote the number of individuals in the population and archive respectively.

2.3 The Selection

The selection process produces a new population of individuals from the current population and the archive using binary tournament selection [39]. Two individuals are randomly sampled without replacement from either the population or the archive. Whether an individual is selected from the archive or the population is determined by the 'elitism degree' (*ED*). The value of ED is computed according to Equation 10 [23].

$$ED = \begin{cases} 1 - \frac{|P_t|}{|\overline{P_t} \cup P_t|} & \text{if } |\overline{P_t}| \geq 2 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Here, $|P_t|$ is the size of the current population and $|\overline{P_t} \cup P_t|$ is the size of the archive and the current population. Hence, the larger the archive, the more likely it is that an individual is sampled from the archive. The individual with the lowest fitness value (see Equation 6) is declared as the winner of the 'binary tournament' and inserted into the new population. If a tie occurs, an individual is chosen with a uniform probability. This procedure is repeated until the new population has reached the size of the old population.

2.4 The Variation Operators

It is well known that the deployment of several problem-specific variation operators (VOs) can improve the evolutionary search [13, 16, 36]. It was therefore decided to implement several problem-specific VOs. These VOs work on different parts of the representation scheme to achieve an appropriate exploitation and exploration of the search space.

Two types of VOs were implemented. The first type (VO1) can change one individual and is also known as 'mutation operator'. The second type (VO2) can change two individuals and is also known as 'crossover operator'. There are several operators of each type. Each VO is applied to an individual with a low probability, which is determined by the parameters 'crossover probability' and 'mutation probability'. A particular VO is chosen with a uniform probability.

$VO1_1$ Operator The $VO1_1$ operator reinitialises an individual as described in section 2.1. This operator is expected to be very disruptive but may help to prevent the premature convergence of the algorithm by introducing new 'genetic material' into the population. The archive also alleviates possible detrimental effects of this operator.

$VO1_2$ Operator The $VO1_2$ operator reinitialises one node of an individual tree. The node is chosen with a uniform probability. A new node or sub-tree of randomly determined size (depending on the maximum/minimum number of nodes) replaces the node.

$VO1_3$ Operator The $VO1_3$ operator inverts a bit of the binary vector of an individual that determines which features are used within the model. The position of this value is chosen with a uniform probability.

$VO1_4$ Operator The $VO1_4$ operator changes one coefficient of the linear combination of one non-terminal node of the individual's tree. This is achieved by adding/subtracting a small value to/from the coefficient, which is obtained from a Gaussian random generator.

$VO1_5$ Operator The $VO1_5$ operator changes the value of a terminal node by sampling from a predefined range of integer values with a uniform probability.

$VO2_1$ Operator The $VO2_1$ operator performs an exchange (crossover) of randomly chosen parts between the trees of two individuals as illustrates in Figure 4.

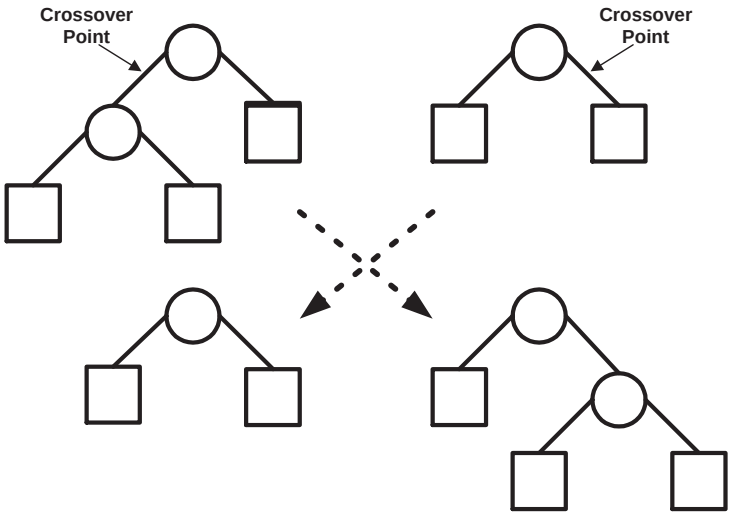


Fig. 4. Crossover between two trees.

The upper part of Figure 4 depicts the trees before the application of this operator. The lower part of Figure 4 depicts the resulting trees. Figure 4 also shows the crossover points which mark the parts that were exchanged. Crossover points are chosen such that the resulting trees do not contain fewer or more nodes than allowed.

2.5 The Archive

As mentioned earlier, an archive contains the best (elite) individuals found by the MOEA. It ensures that the best individuals are preserved, as they could otherwise get lost due to the randomness of the selection process [39]. For practical reasons, an archive should only store a limited number of individuals (large numbers of individuals increase the memory demands and the execution time of the algorithm). However, this can result in the loss of non-dominated solutions, which is a problem known as partial deterioration [22]. Laumanns *et al.* [22] have proposed an archiving strategy that uses an archive of bounded size, but does not exhibit the problem of partial deterioration. For the present purposes, the implemented MOEA deploys this archiving strategy.

3 Results and Discussions

The implemented MOEA is evaluated on three artificial datasets and one medical dataset. As each run of the MOEA produces several trade-off solutions, the model with the maximum log-rank statistic (see equation 2) was chosen as the final output of the algorithm. It should be noted, that if there were several models with the same log-rank statistic, the model with the smallest number of nodes and group assignments (see objective 4 in section 2.2) was chosen. During the experiments the parameters summarised in Table 2 were used.

Parameter	**Parameter Value**
Population Size	100
Number of Generations	1000
Crossover Probability	0.7
Mutation Probability	0.4
Minimum number of tree nodes	3
Maximum number of tree nodes	21

Table 2. MOEA parameter values used for each MOEA run.

3.1 Evaluation on the artificial dataset 1

This artificial dataset was inspired by the well-known XOR classification problem. It has two binary features: $X0$ and $X1$ and a third binary feature that has to be predicted. Two new features replaced the third feature in order to simulate a survival analysis problem rather than a classification problem. The first feature (I) indicated whether the event occurred to the sample and the second feature ($Time$) determined the observation time of the sample. The maximum observation time was set to a value of ten. The actual data consisted of fifty samples for each feature value combination (see Table 3).

x_0	x_1	μ	σ
0	0	1.1	0.15
0	1	1.8	1.1
1	0	1.8	1.1
1	1	1.1	0.15

Table 3. Possible combinations of the feature values (two left columns) and the parameters of the inverse lognormal distribution (two right columns).

Two values were generated to obtain the observation time for a sample. The first value (α) was sampled from the interval $[0 \ldots 10]$ with a uniform probability. The second value (β) was sampled from a inverse lognormal distribution [9], which is characterised by the parameters μ and σ. The parameter values for a particular sample (feature value combination) are summarised in Table 3. To simulate censoring the actual observation time and the indicator value were determined according to equation 11.

$$(I, Time) = \left| \begin{array}{ll} (1, \beta) & for\ \alpha \geq \beta \\ (0, \alpha) & \text{otherwise} \end{array} \right. \tag{11}$$

Figure 5 depicts the probability density functions (left) and the survival function (right) for the parameters in Table 3.

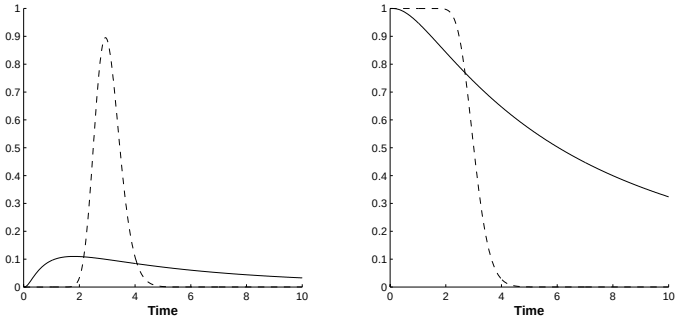


Fig. 5. Probability density functions (left) and the survival functions (right) for the parameters $\mu = 1.1$, $\sigma = 0.15$ (dashed line) and $\mu = 1.8$, $\sigma = 1.1$ (solid line).

The dashed lines correspond to the parameters $\mu = 1.1$ and $\sigma = 0.15$ and the solid lines to the parameters $\mu = 1.8$ and $\sigma = 1.1$. It can clearly be seen that this problem does not exhibit proportional hazards as the survival curves cross. This constellation usually causes problems to classical models and was therefore chosen. In fact, the log-rank statistic also presumes proportional hazards.

The model achieved a log-rank statistic of 80.76 and recovered the original grouping of the data accurately. Figure 6 depicts the Kaplan-Meier curves that the model produces according to its grouping.

3.2 Evaluation on the artificial dataset 2

This artificial dataset was generated in the same manner as the first artificial dataset. However, it contains an additional 'noise' feature, which was generated by sampling from the interval $[0 \ldots 1]$ with a uniform probability. As the dataset contained more features, 100 instead of 50 samples were produced for each feature combination. The generated model achieved a log-rank statistic of 202.51 and recovered the underlying grouping of the data correctly. In addition, the additional 'noise' feature was dropped. This shows that the approach can perform feature selection during the model extraction, while recovering the underlying grouping.

3.3 Evaluation on the artificial dataset 3

This artificial dataset was generated in a similar manner as the first artificial dataset. However, it contains three features that can have the value one or zero. Hence, there are eight possible

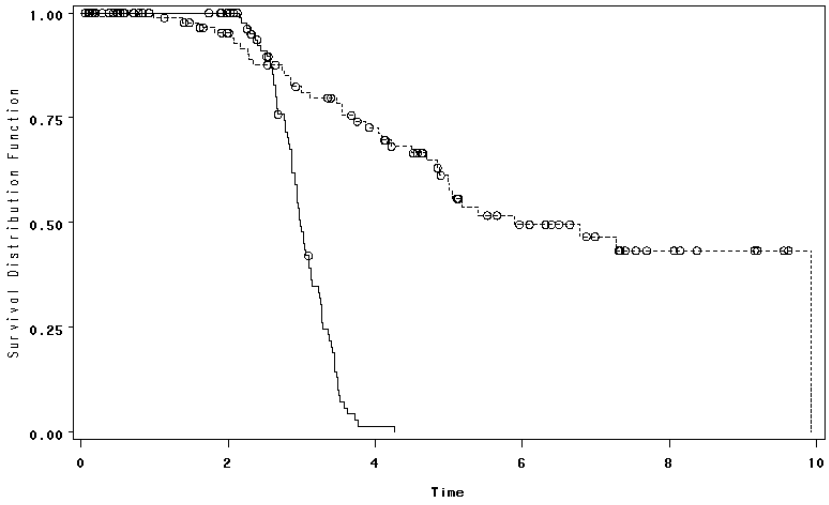


Fig. 6. Kaplan-Meier curves according to the grouping of the model.

combinations as shown in Table 4 with the corresponding parameters for the inverse lognormal distributions. We produced 150 samples for each feature value combination.

x0	**x1**	**x2**	μ	σ
0	0	0	1.10	0.15
0	0	1	3.50	1.70
0	1	0	1.80	1.10
0	1	1	3.50	1.70
1	0	0	1.80	1.10
1	0	1	3.50	1.70
1	1	0	1.10	0.15
1	1	1	3.50	1.70

Table 4. Possible combinations of the feature values (two left columns). Parameters for the inverse lognormal distribution (two right columns).

The probability density functions and survival functions for the three lognormal distributions are shown in Figure 7.

The model achieved a log-rank statistic of 425.01. However, 12.5% of the samples were grouped incorrectly. Samples with the feature combination 110 were wrongly assigned to the group with the parameters $\mu = 1.80$ and $\sigma = 1.10$. Furthermore, samples with the feature combination 010 and 100 were assigned to two different groups, although they belong to the same (parameter group) group. The Kaplan-Meier curves of the four groups that the model generated are depicted in figure 8.

3.4 Evaluation on a medical dataset

This section evaluates the implemented MOEA on a 'real-world' medical dataset that contains feature values of uveal melanoma patients [4]. Uveal melanomas, which have an occurrence rate of six per million per year [5], arise from melanocytes in the uvea. Patients with

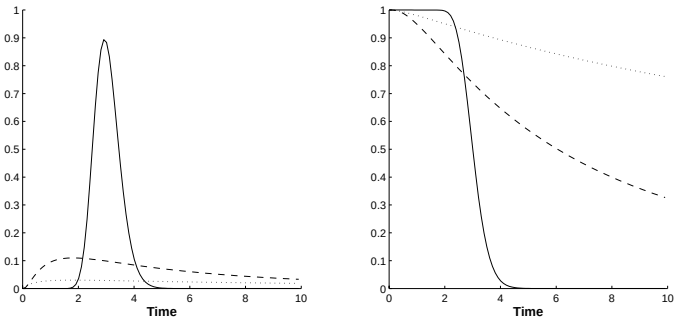


Fig. 7. Probability density functions (left) and the survival functions (right) for the parameters $\mu = 1.1$, $\sigma = 0.15$ (solid line); $\mu = 1.8$, $\sigma = 1.1$ (dashed line); $\mu = 3.5$, $\sigma = 1.7$ (dotted line).

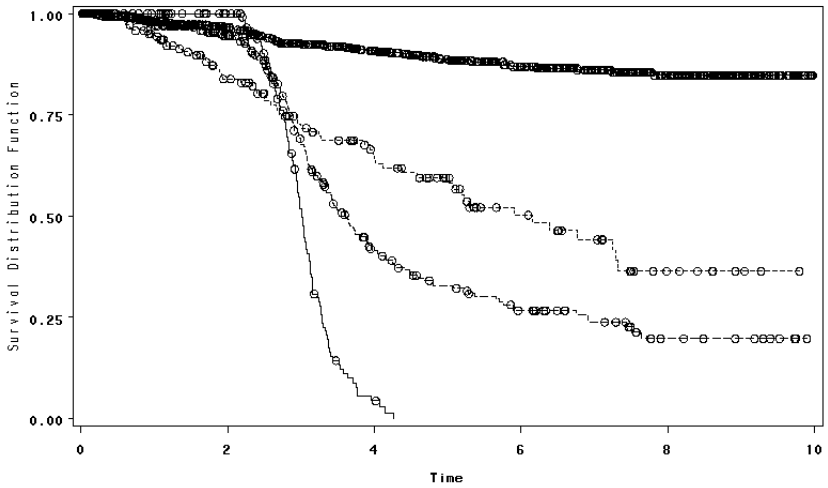


Fig. 8. Kaplan-Meier curves according to the grouping of the generated model.

uveal melanoma usually have symptoms such as blurred vision, flashing lights and visual field loss. Without treatment many eyes become blind, painful and cosmetically unsightly. Approximately 50% of all patients with uveal melanoma ultimately die of this disease, nearly always as a result of haematogenous spread of tumour (*i.e.* through the blood circulation) to the liver. Assigning patients to risk groups has a number of important benefits. For example, clinicians can treat patients differently depending on their risk group. This may help to assign constrained resources accordingly. Furthermore, it might help informed patients to plan their lives and provide future care for their dependents.

The data contained 1820 samples with four features that are summarised in Table 5.

Name	Type	Description
Antora	Dichotomous	Indicates whether the tumour is at the front or the back of the eye (anterior choroid or posterior choroid).
Age	Continuous	Age of the patient when (s)he entered the study.
Ludb	Continuous	Tumour dimension as measured by ultrasonography.
Gender	Dichotomous	n/a

Table 5. Features of the uveal melanoma dataset.

The generated model achieved a log-rank statistic of 204.31. It only uses two of the original four features: *Antora* and *Ludb*. This agrees with the medical consultant who provided the data and deemed the features *Antora* and *Ludb* as more important. Figure 9 depicts the Kaplan-Meier curves according to the grouping of the generated model.

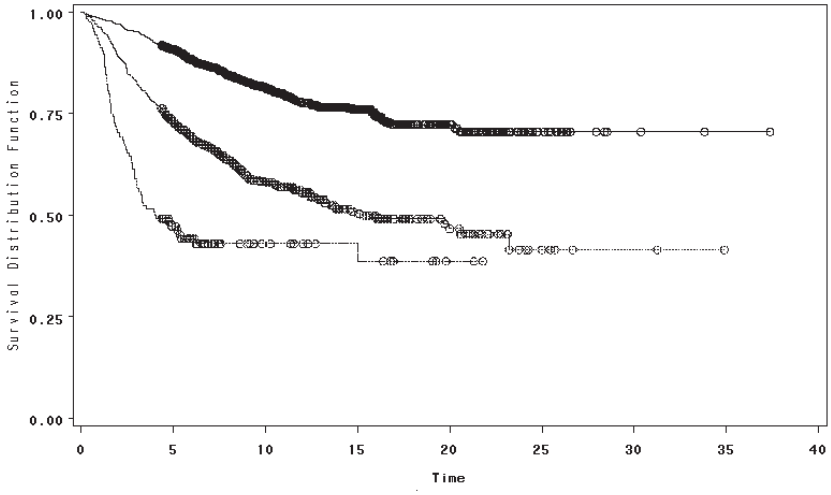


Fig. 9. Kaplan-Meier curves according to the grouping of the model.

The decision surface of the model is depicted in Figure 10. The best Kaplan-Meier curve corresponds to samples that were assigned to group '1' (1060 samples). The second best Kaplan-Meier curve corresponds to samples that were assigned to group '2' (636 samples). The worst Kaplan-Meier curve corresponds to samples that were assigned to group '3' (124 samples).

The decision surface was shown to a medical consultant, who agreed with this assignment to risk groups. However, whether or not the model is better than other models, such as tumour nodes metastasis staging (TNM) [35], remains to be investigated.

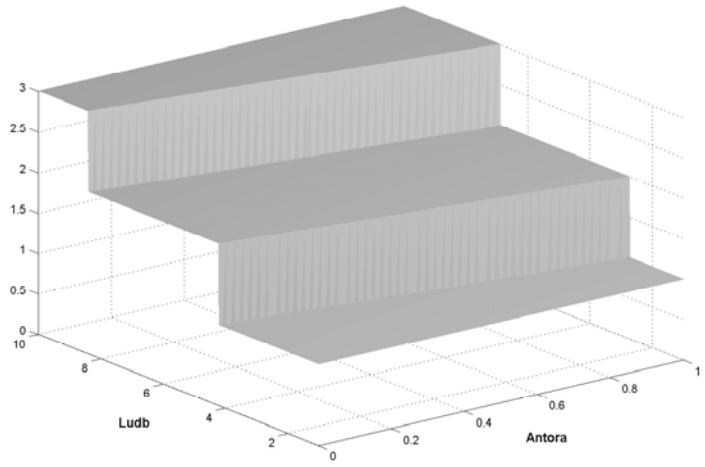


Fig. 10. Decision surface of the generated model.

4 Conclusions and further work

A multi-objective evolutionary algorithm for the extraction of oblique decision trees for survival analysis has been proposed and evaluated. The application of the MOEA to several artificial benchmark datasets and one medical dataset has shown that the MOEA can be used to identify groups with different survival behaviour. The evaluation on the artificial dataset also emphasised that the approach can cope with interaction effects and noisy non-proportional hazard distributions.

The current execution time of the MOEA is acceptable. It takes about one hour to extract a set of models from a dataset consisting of 2000 samples with four features. However, the execution time of the approach could be improved, by executing it in parallel whilst harvesting the computational power of idle computers in an institution. This was suggested in [29]. To investigate this further, would be especially interesting if one intends to apply the approach to larger datasets.

The present work uses the log-rank statistic to determine whether the groups described by the oblique decision tree have significantly different survival behaviour. However, the log-rank statistics assumes that the underlying hazard/lifetime distributions are proportional. Although we have shown that the statistic can be used to recover groups with non-proportional lifetime distributions, it might be worthwhile to investigate other statistics. The Peto statistic [18] is a suitable candidate.

Future work will also evaluate the approach on more complicated and larger artificial and medical datasets.

References

1. A. Afifi, V.A. Clark, and S. May. *Computer-aided multivariate analysis*. Chapman and Hall, 2003.
2. E. Cantú-Paz and C. Kamath. Inducing oblique decision trees with evolutionary algorithms. *IEEE Transactions on evolutionary computation*, 7(1):54–68, 2003.
3. N.L. Cramer. A representation for the adaptive generation of simple sequential programs. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 183–187. Lawrence Erlbaum Associates, Inc., 1985.

4. B.E. Damato. *Ocular tumours : diagnosis and treatment*. Butterworth Heinemann, 2000.
5. B.E. Damato. Current management of uveal melanoma. *European Journal of Cancer Supplements*, 3(3):433–435, 2005.
6. K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley Europe, 2001.
7. V. Dhar, D. Chou, and F. J. Provost. Discovering interesting patterns for investment decision making with glower - a genetic learner overlaid with entropy reduction. *Data Mining and Knowledge Discovery*, 4(4):251–280, 2000.
8. J.F. Elder and D. Pregibon. A statistical perspective on knowledge discovery in databases. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 83–113. AAAI Press / The MIT Press, 1996.
9. M. Evans, N. Hastings, and B. Peacock. *Statistical Distributions*. John Wiley and Sons Inc, 1993.
10. E.J. Freireich, E. Gehan, E. Frei, L.R. Schroeder, I.J. Wolman, R. Anbari, E.O. Burgert, S.D. Mills, D. Pinkel, O.S. Selawry, J.H. Moon, B.R. Gendel, C.L. Spurr, R. Storrs, F. Haurani, B. Hoogstraten, and S. Lee. The effect of 6-mercaptopurine on the duration of steroid-induced remissions in acute leukemia. *Blood*, 21:699–716, 1963.
11. A. Freitas. *Data Mining and Knowledge Discovery With Evolutionary Algorithms*. Springer Verlag, 2002.
12. Z. Fu, B.L. Golden, S. Lele, S. Raghavan, and E. Wasil. Diversification for better classification trees. *Computers and Operations Research*, 2005.
13. J.J. Grefenstette. Lamarckian learning in multi-agent environments. In Rick Belew and Lashon Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 303–310, San Mateo, CA, 1991. Morgan Kaufman.
14. T.J. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning : data mining, inference, and prediction*. Springer, 2001.
15. M. Humphrey, S. Cunningham, and I. Witten. Knowledge visualization techniques for machine learning. *Intelligent Data Analysis*, 2:333–347, 1998.
16. C.Z. Janikow. A knowledge-intensive genetic algorithm for supervised learning. *Machine Learning*, 13:189–228, 1993.
17. J.D. Kalbfleisch and R.L. Prentice. *The statistical analysis of failure time data*. Wiley, 1980.
18. D.G. Kleinbaum. *Survival analysis: A self-learning text*. Springer, 1996.
19. J.R. Koza. Genetic programming. In J.G. Williams and A. Kent, editors, *Encyclopedia of Computer Science and Technology*, volume 39, pages 29–43. Marcel-Dekker, 1998.
20. L.I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley and Sons Inc, 2004.
21. M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Archiving with guaranteed convergence and diversity in multi-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 439–447. Morgan Kaufmann Publishers, 2002.
22. M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multi-objective optimization. *Evolutionary Computation*, 10:263–282, 2002.
23. M. Laumanns, E. Zitzler, and L. Thiele. A unified model for multi-objective evolutionary algorithms with elitism. In *Proceedings of the 2000 Congress on Evolutionary Computation (CEC 2000)*, pages 46–53, Piscataway, New Jersey, 2000. IEEE Press.
24. N. Mantel and W. Haenszel. Statistical aspects of the analysis of data from retrospective studies of disease. *Journal of the National Cancer Institute*, 22(4):719–748, 1959.
25. Z. Michalewicz and D.B. Fogel. *How to Solve it: Modern Heuristics*. Springer, Berlin, 2000.
26. N.I. Nikolaev and V. Slavov. Inductive genetic programming with decision trees. *Intelligent Data Analysis*, 2(1-4):31–44, 1998.
27. M. Pazzani, S. Mani, and W. Shankle. Comprehensible knowledge discovery in databases. In M.G. Shafto and P. Langley, editors, *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, pages 596–601. Lawrence Erlbaum, 1997.
28. C. Setzkorn. *On the Use Of Multi-Objective Evolutionary Algorithms For Classification Rule Induction*. PhD thesis, University of Liverpool, Department of Computer Science, Liverpool, United Kingdom, 2005.
29. C. Setzkorn and R.C. Paton. Javaspaces - an affordable technology for the simple implementation of reusable parallel evolutionary algorithms. In J.A. López, E. Benfenati, and W. Dubitzky, editors, *Knowledge Exploration in Life Science Informatics - KELSI 2004 (LNAI 3303)*, pages 151–161. Springer-Verlag New York, Inc., 2004.

30. C. Setzkorn and R.C. Paton. On the use of multi-objective evolutionary algorithms for the induction of fuzzy classification rule systems. *BioSystems*, 81(2):101–112, 2005.
31. C. Setzkorn, A.F. Taktak, and B. Damato. Survival analysis using a multi-objective evolutionary algorithm. In *Proceedings of the Second International Conference on Computational Intelligence in Medicine and Healthcare - CIMED 2005*, pages 224–230, 2005.
32. C. Setzkorn, A.F. Taktak, and B. Damato. On the use of multi-objective evolutionary algorithms for survival analysis. *BioSystems*, 2006. (in press).
33. Y. Shi, R. Eberhart, and Y. Chen. Implementation of evolutionary fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 7(2):109–119, 1999.
34. B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1999.
35. L.H. Sobin and C. Wittekind. *Classification of Malignant Tumours*. Wiley-Liss, 2002.
36. W.M. Spears. Adapting crossover in evolutionary algorithms. In J. R. McDonnell, R.G. Reynolds, and D.B. Fogel, editors, *Proceedings of the Fourth Annual Conference on Evolutionary Programming*, pages 367–384, Cambridge, MA, 1995. MIT Press.
37. A. Tsakonas. A comparison of classification accuracy of four genetic programming-evolved intelligent structures. *Information Sciences*, 176(6):691–724, 2006.
38. Y.S. Yeun, K.H. Lee, and Y.S. Yang. Function approximations by coupling neural networks and genetic programming trees with oblique decision trees. *Artificial Intelligence in Engineering*, 13(3):223–239, 1999.
39. E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm. In K.C. Giannakoglou, editor, *Proceedings of the EUROGEN2001 Conference*, pages 95–100, Barcelona, Spain, 2002.

Improving Network Intrusion Detection System By Decision Tree And Exclusion-Condensation Based Pattern Matching

Xin Jin[1], Ronghuai Huang*[2], Rongfang Bie[1]

[1]College of Information Science and Technology
Beijing Normal University, Beijing 100875, China
[2]School of Education Technology, Beijing Normal University, Beijing, China
xinjin796@126.com
huangrh@bnu.edu.cn
rfbie@bnu.edu.cn

Abstract. Signature-based detection approach is followed by most deployed Network Intrusion Detection Systems (NIDS). The performance of such systems is dominated by comparing the input network data to signature rules, which are usually described in a specification language. In our project, we varied and tested three of the most commonly used attribute selection criteria to find the best decision tree building for intrusion detection system rules; also, we developed ECPM, a extension of the fastest pattern matching algorithm E2xB which is especially designed for NIDS, then integrated decision tree and ECPM forming a new detect engine DTECPM to speed up the performance of state-of-art intrusion detection systems. Experimental results on DARPA intrusion detection benchmark datasets show that the improvement is great.

1 Introduction

There is no disputing the fact that the number of hacking and intrusion incidents is increasing year on year as technology rolls out. Intrusion Detection Systems (IDS) are monitoring programs aiming at detecting intruders who are acting illegally in a computer system. An intruder can be generally defined as a person or process performing unauthorized operations on a network node (workstation, server, router...) in an attempt to gain more control over it, though the signification of intruder and illegal depends on the limits set by a security model.

There are two main categories of IDS: host based (HIDS) and network based (NIDS). A host based IDS is running as a process on a host computer and monitors sensitive activities on this computer, such as unauthorized access or modification of files. A network based IDS consists in a sniffer program listening to the network traffic in an attempt to detect suspicious activity over network protocols, Snort (the de facto standard for network intrusion detection) and Firestorm are two examples of

* Corresponding Author.

such systems [1,18]. Both host and network based IDS will generate alarms when detecting suspicious activity. These alarms should trigger a response from the network administrator or some automated response tool. A host based IDS will usually be located on critical computers such as servers, while a network based IDS should be located at strategical points in a network, in order to have access to relevant traffic. In this paper, we focus on NIDS since NIDS can detect intruders before they have entered the target while HIDS can only detect intrusions when the target has already been intruded.

Most deployed NIDS monitor packets on the network and compare them against a database of signature rules from known malicious threats. As is a state-of-art NIDS, Snort has been extensively used and studied in the literature [4]. An example of a Snort rule is: *alert tcp $EXTERNAL_NET any -> $HOME_NET 23 (msg:"TELNET livingston DOS"; flags:A+; content:"|fff3 fff3 fff3 fff3 fff3|"; classtype:attempted-dos; sid:713; rev:4;)*. A rule contains fields that can specify a suspicious packet's protocol, IP address, Port, Flags, content and others. The "content" field contains the pattern that is to be matched, written in ascii, hex or mixed format, where hex parts are between vertical bar symbols "|". The simplest technique utilized to compare a packet with a set of rules is to consecutively check every defined feature of a rule against the data and then advance to the next one, eventually determining all matching rules. Snort-like IDS optimized such time-taking procedure by constructing a two-dimension list for all the rules. However, this optimizing is not good enough for real-time highly speed networks. In [17] the authors use a variant of ID3 to build detection engine Snort-NG for Snort rules. They use Information Gain as the attribute selecting criteria to construct the decision tree, however, it is widely accepted that no single kind of attribute selection criteria that performs the best in all cases [9,16]. In this study we implemented three decision tree detection engines based on Information Gain, GINI and CSS respectively and found that GINI is the best when rule sizes are small and medium (suitable for special purpose NIDS whose rule sizes are usually not too large), while Information Gain is the best when rule sizes are very large (suitable for general purpose NIDS which normally has a large amount of rules). The findings are valuable since that we can choose different kinds of decision tree detection engine for different situation.

Pattern matching (or string matching), which is an important part of NIDS, generally means matching text or binary sequences in the packet payload against known signature strings (the string in the "content" field of the rule mentioned above is an example). The major problem of string matching in network intrusion detection is that it is generally expensive: finding a single pattern in an string imposes computation which is at least linear to the size of the string [13] and NIDS rule-sets often contain thousands of such strings. The computational burden of string matching is significant: recent measurements on a production network suggest that Snort spends roughly 30% of its total processing time in string matching, while for Web-intensive traffic, this cost increase to as much as 80% [20]. In order to speed up the string matching process, many algorithms have been proposed, including Boyer-Moore [3], Aho-Corasick [19], Wu-Manber [14], setwise Boyer-Moore-Horspool [20] and E2xB [2]. Among them, E2xB has proved to be the best especially for NIDS [6]. E2xB is based on the fact that most packet are normal thus are not likely to be matched to any attack signa-

ture pattern, so E2xB use fixed-size sub-strings to quickly filter out non-attack packets. However, one problem of E2xB is that it only consider the "most-normal" characteristics of the packets and does not the other part of detection task: the rule signatures. In this study, we propose a new algorithm Exclusion-Condensation based Pattern Matching (ECPM) which extends E2xB by adding a condensation based module to optimize signatures. Then we integrate ECPM and Decision Tree for Snort forming a detection engine Snort-DTECPM.

The paper is organized as follows. Section 2 presents the ECPM algorithm. Section 3 describes and the decision tree structure for NIDS rules and how to integrate ECPM and rule decision trees. Section 4 presents three attribute selection criteria and how to tailor them for NIDS rules. Section 5 presents the experiments on Snort-DTECPM, the results are compared with Snort and Snort-NG. The final section briefly concludes the paper.

2 Exclusion-Condensation based Pattern Matching

According to the specific characteristics of NIDS (that is, most network packets are normal network traffic), and to optimize signatures. We developed a new fast pattern-matching algorithm called Exclusion-Condensation based Pattern Matching (ECPM).

Suppose that we want to check whether a packet payload *P* contains a signature string *S*. If there is at least one character in *s* that is not in *P*, then *S* is not in *P*. This idea is used to determine quickly when a given signature *S* does not appear in the payload. In order to efficiently determine whether a character *c* in signature *S* belongs to *P*, an 256-element array b[256] is used to hold the ASCII character vocabulary. While the NIDS is set up, the program condense the signature set (see procedure signature_condense() in Figure 2 for details); then, when comes a new packet, we initialize its payload *P*, and, for each character *c* that appears in *P*, we mark the corresponding element on the array. If every character of a signature pattern belongs to *P*, we then use a standard string-searching algorithm Boyer-Moore to confirm whether it is really a sub-string of *P*. The pseudo-code for condensation, initializing and string searching is presented in Fig. 1.

In order to reduce false matches, the algorithm can be generalized for pairs of characters. We first condense the signature set by pairs of consecutive characters, and when comes a new packet we record the appearance of each pair of consecutive characters in its payload *P*. Then, the algorithm checks whether each distinct pair of consecutive characters of *S* appears in *P*. If a pair is found that does not appear in *P*, we know that *S* is not in *P*. When a pattern is made up of only one character, search() will directly go to standard string searching algorithm.

```
Procedure signature_condense(S)
(1)   count the number of distinct characters in S;
(2)   for the ith unique character c              //ordered by ASCII
(3)      SC[i]= c;                 //this array holds the condensed S
(4)   return SC;
Procedure payload_initialize(payload)
```

```
(1)  bzero(b, 256/8);              // clear array
(2)  for each character c in the payload
(3)     b[c] = 1;
(4)  return b;
Procedure search(SC, S, payload)
(1)  for each element i in condensed signature SC
(2)     if (b[SC[i]] == 0)
(3)        return 0;
(4)  return boyer_moore(S, payload);
```

Fig. 1. Pseudo-code for ECPM (the *one* character version)

ECPM is better than E2xB in that ECPM takes into account both signature and payload, while E2xB only considers payload. Take the following signature pattern for example:

ABABABCCCABABABABDDDABABABE

With ECPM, this pattern will be condensed as a 5-element array SC[5], with element SC[0]=A, SC[1]=B,…, SC[4]=E, while in E2xB, a 27-element array is needed. Suppose that A, B, C and D are all in the payload, then E2xB will take 27 matches to complete the searching routine, while ECPM will need only 5 matches. When rule sizes are large, which means that there are thousands of signatures to be matched to a payload, condensing the signature patterns can do speed up the matching process.

3 Decision Tree Structure for Rules

Decision tree is traditionally used for classification [5]. It builds a flow-chart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes or class distributions.

When using the fields/features in the network packet header as nodes, decision tree built for NIDS rules can be used to quickly check whether a packet satisfy any rule by starting at the root of the tree and traverse down to a leaf node, which provides the classification of the packet. In this way, the detection mechanism is changed from rule-to-rule comparison to a feature-to-feature approach. This can quicken the rules matching procedure significantly. Table 1 shows a simplified example of the network attack signature rules set. Fig. 2 shows the corresponding decision tree constructed for it.

Table 1. A simplified example of network attack signature rules (field "content" are included). A rule specifies a packet from a source address to a destination address and destination port, the class label is given in the last column. The source and destination address fields have the type IPv4 while the destination port field is of type integer

Rule	Source IP	Destination IP	Destination Port	Class
R1	10.11.0.2	10.11.0.5	23	Attack 1
R2	10.11.0.3	10.11.0.1	25	Attack 2

R3	10.11.0.2	10.11.0.4	83	Attack 3
R4	10.11.0.2	10.11.0.5	83	Attack 4

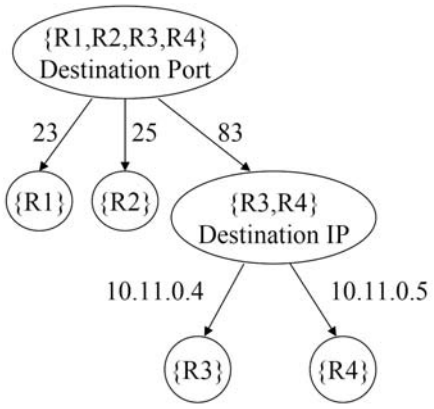


Fig. 2. Decision tree constructed for rules. The arrows that are leading from a node to its children are annotated with the value of the packet field that is specified by the rules of each respective child node.

The decision tree is built in a top-down manner. At each non-leaf node, the program selects a field/attribute to partition the rules of the node. The tree's root node represents the set that initially contains all rules while its children nodes are associated with the direct subsets created by partitioning them according to a selected field. When a node in the tree contains more than one rule and these rules have different values in their fields except "content" (this field is not used for any node partition), then they are subsequently partitioned and the node is labeled with the field that has been used for this partitioning step. Finally, every leaf node of the tree contains only a single rule and one attack class associated with the rule, or multiple rules which have different values only in their field "content".

After rule tree construction (this is done when the system starts up), the detection process progress as follows (take the tree in figure 2 for example): for any incoming packet, the detection engine first find whether its Destination Port has a value in the set {23, 25 or 83}, if no, we can quickly decide that the packet is not an attack, if yes, and if the vale is 23, then we can quickly know that the packet is a attack with the class label *Attack 1*. If the value is 83, then the engine has to further check the Destination IP of the packet in the same way. If a rule at the leaf node has "content" field (this field is not included in the tree structure), a string-matching algorithm is called to check whether the packet's payload contain a string specified in the rule and then decide whether the packet is really an attack.

To integrate ECPM and decision tree, the simplest way is to call ECPM as the string matching procedure. This mechanism is suitable for E2xB and other algorithms, but since ECPM uses *signature_condense*() to build condensation structures to optimize signature patterns, this sub-procedure will be repeatedly called if using such mechanism directly, so we extract the condensing sub-procedures to be done when

the decision tree is built and when calling ECPM we just skip the condensing procedure, this can further improve the matching process.

The actual rule set has many field/attribute and since the rule decision tree does not involve pruning which is common in traditional decision tree, when many rules have to be processed, the tree will become to big to be manageable. In order to deal with the problem, multiple trees are needed to reduce the tree size. However, this solution proposes another problem when we integrate ECPM and decision trees. Some rules may have the same "content" field and if they are distributed to different decision trees the system will build many copies of the same condensation structure, we solve the problem by using a tab to label the condensing procedures for each decision tree, when a content have already be condensed in one decision tree, the same content will not need to be condensed in any other tree.

4 Attribute Selection Criteria

Many kinds of criteria for attribute selection have been proposed for decision tree building [7,8,10]. However, as mentioned in the introduction section, there appears to be no single kind of attribute selection criteria that performs the best in all cases. In order to create an optimized best (i.e., best for network data) tree structure for NIDS rules, we varied and tested three of the most commonly used attribute selection criteria: Information Gain, Gini Index of Diversity (GINI) and Chi-Squared Static (CSS).

Suppose that there are a total of m classes denoted by $C=\{C_1, C_2,\ldots, C_m\}$, at a particular node in the tree, let there be N training examples represented by,

$$(a(1), b(1),\ldots; t(1)), (a(2), b(2),\ldots; t(2)),\ldots, (a(N), b(N),\ldots; t(N))$$

where, $a(i)$, $b(i)$,…are vectors of n attributes and $t(i) \in C$ is the class label. Of the N examples, N_{C_k} belong to class C_k. The decision rule at the node splits these examples into V partitions, or V child nodes, each of which has $N^{(v)}$ examples. In a particular partition, the number of examples of class C_k is denotes by $N_{C_k}^{(v)}$.

For NIDS rules, each rule itself is considered to be a class on its own; therefore m is the total number of rule itself (with N being equal to m), both N_{C_k} and $N_{C_k}^{(v)}$ are equal to 1.

4.1 Information Gain

Partition on the basis of information gain is one of the most commonly used node splitting criteria. It forms the basis of the popular ID3, C4.5 algorithms and is based on choosing the attribute that results in the largest decrease in entropy [11,12]. More specifically the information gain resulting from splitting the rules bases on attribute A can be written as,

$$Gain(A) = [\sum_{k=1}^{m} -(\frac{N_{C_k}}{N})\log(\frac{N_{C_k}}{N})] - [\sum_{v=1}^{V}(\frac{N^{(v)}}{N})\sum_{k=1}^{m} -(\frac{N_{C_k}^{(v)}}{N^{(v)}})\log(\frac{N_{C_k}^{(v)}}{N^{(v)}})] \quad (1)$$

For NIDS rules, the equation above is varied as,

$$Gain(A) = \log(N) - [\sum_{v=1}^{V}(\frac{N^{(v)}}{N})\log(N^{(v)})] \quad (2)$$

The attribute chosen is the one with the largest value of *Gain*(*A*).

4.2 Gini Index of Diversity

Gini Index of Diversity (GINI for short), originally proposed by Breiman et al refbreiman, is based on,

$$D(A) = \frac{1}{N}[\sum_{k=1}^{m}\sum_{v=1}^{V}\frac{N_{C_k}^{(v)^2}}{N^{(v)}} - \sum_{k=1}^{m}\frac{N_{C_k}{}^2}{N^{(v)}}] \quad (3)$$

Usually we would like a node in the decision tree to be pure, i.e., have instance of a single class. Similar to the decrease in entropy used in the information gain based attribute selection methods, here the decrease in impurity as given by the equation above is used.

For NIDS rules, the equation above is varied as,

$$D(A) = \sum_{v=1}^{V}\frac{1}{N^{(v)}} - \frac{1}{N} \quad (4)$$

The attribute chosen is one which results in the largest decrease in impurity (or the largest value of $D(A)$).

4.3 Chi-Squared Static (CSS)

The Chi-Squared Statistic (χ^2) based partitioning is based on comparing the obtained values of the frequency of a class because of the split to the *a priori* frequency of the class. More specifically,

$$\chi^2 = \sum_{k=1}^{m}\sum_{v=1}^{V}\frac{(N_{C_k}^{(v)} - \widetilde{N}_{C_k}^{(v)})^2}{\widetilde{N}_{C_k}^{(v)}} \quad (5)$$

where , $\widetilde{N}_{C_k}^{(v)} = (N^{(v)}/N)N_{C_k}$ denotes the *a priori* frequency. Clearly, a larger value of χ^2 indicates that the split is more homogeneous, i.e., has a greater frequency of instances from a particular class.

For NIDS rules, the equation above is varied as,

$$\chi^2 = \sum_{v=1}^{V} \frac{(N - N^{(v)})^2}{N^{(v)}} \tag{6}$$

The attribute chosen is the one with the largest value of χ^2.

5 Experiments and Results

This section presents the experimental results of our evaluation of three attribute selection criteria tailored for NIDS rules decision tree building, and the results that we have obtained by integrating decision trees and ECPM to replace the detection engine of Snort. We have implemented patches named Snort-DTECPM (Decision Tree and Exclusion-Condensation based Pattern Matching) for Snort. Our performance results are compared to the results obtained with Snort and Snort-NG.

For all the experiments we used a PC with an Intel Pentium III Coppermine Processor running at 600 MHz, with a L1 cache of 16 KB and L2 cache of 256 KB, and 192 Mbytes of main memory. The operating system is RedHat Linux 9.0. Programs read tcpdump log files from disk. (For simplicity, data are read from local files by using the appropriate Snort option, which is passed to the underlying pcap(3) library. Replaying data from a remote host provided similar results.) When performing the measurements, all preprocessor and logging plug-ins of Snort have been disabled to have our results reflect mostly the processing cost of the detection algorithms themselves. Although the overhead of the operating system to read from the file and the parsing functionality of Snort still influences the numbers, it does so for both approaches.

We used two different types of data sets. One is an *inside* tcpdump file with a size of 351.0MB (1945538 packets) which was produced by MIT Lincoln Labs for their 1999 DARPA intrusion detection evaluation [15]. The other one with 351.7 MB (1616713 packets) is an *outside* tcpdump file from the same resource. Fig. 3 shows the inside and the outside data, which was collected in the simulated military network of Lincoln Lab. The simulated system consists of four real victim machines running SunOS, Solaris, Linux, and Windows NT, a Cisco router, and a simulation of a local network with hundreds of other hosts and thousands of users and an Internet connection without a firewall.

In a single run, we measured the time that programs needed to complete the analysis of the test data sets. For each of these data sets, we performed five single runs for an increasing number of rules and averaged the results. Three programs were executed consecutively and did not influence each other while running.

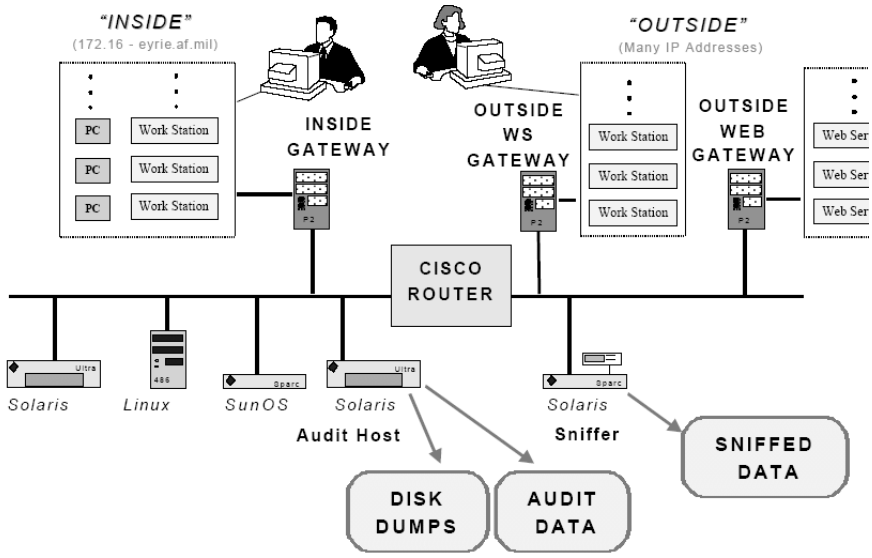


Fig. 3. The simulated military network of Lincoln Lab for DARPA intrusion detection evaluation datasets, from [15].

The comparison of the results for the test data sets is shown in Fig. 4 and 5. Snort-DTECPM performed better than Snort for every test case and yielded an average speed up of 213%. The maximum speed up seen during the tests was 278%.

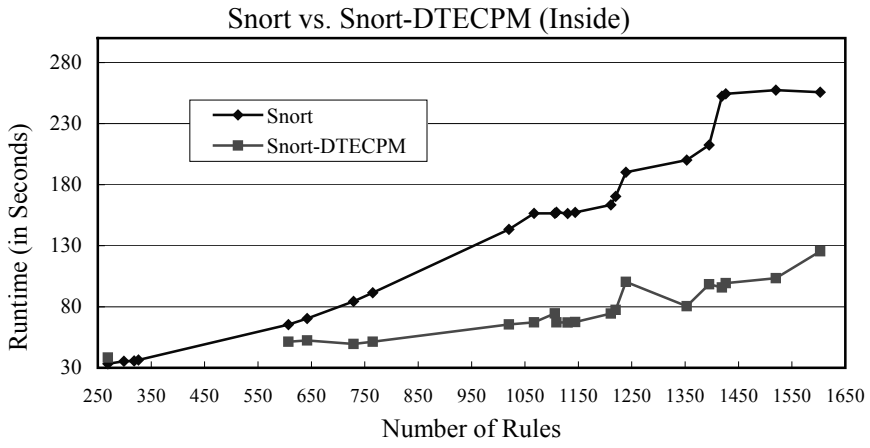


Fig. 4. A comparison of Snort and Snort-DTECPM on runtime performance for inside.tcpdump file.

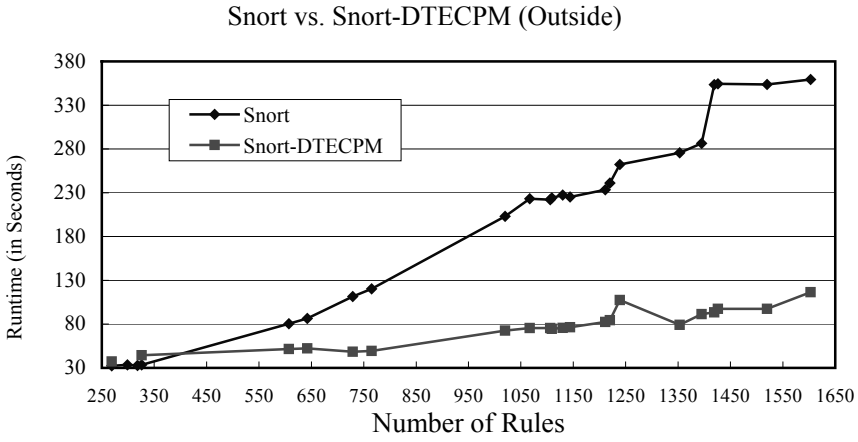


Fig. 5. A comparison of Snort and Snort-DTECPM on runtime performance for outside.tcpdump file.

In order to find the best attribute selection criteria for network data decision tree building, we disabled the ECPM, implemented and tested three different detection engines: Information Gain, CSS and GINI decision tree.

Fig. 6 and 7 show that for both *inside* and *outside* network data, GINI based decision tree engine is better than Information Gain and CSS based decision engines when rule sizes are small and medium (less than 1400). For the inside traffic, the maximum speed up is 4% while for the outside traffic 8%. Information gain is better than CSS and GINI when rule sizes are very large (over 1400). For the inside traffic, the maximum speed up is 14% while for the outside traffic 19%. CSS performed worst for nearly every test case.

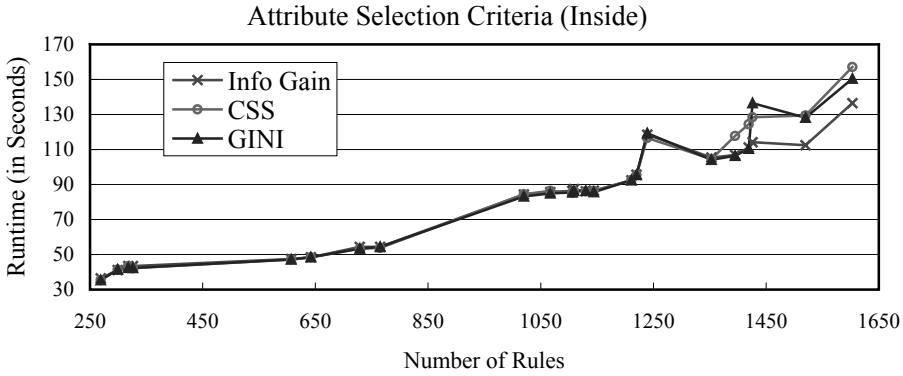


Fig. 6. A comparison of different attribute selection criteria on runtime performance for different rule sizes. (inside.tcpdump file)

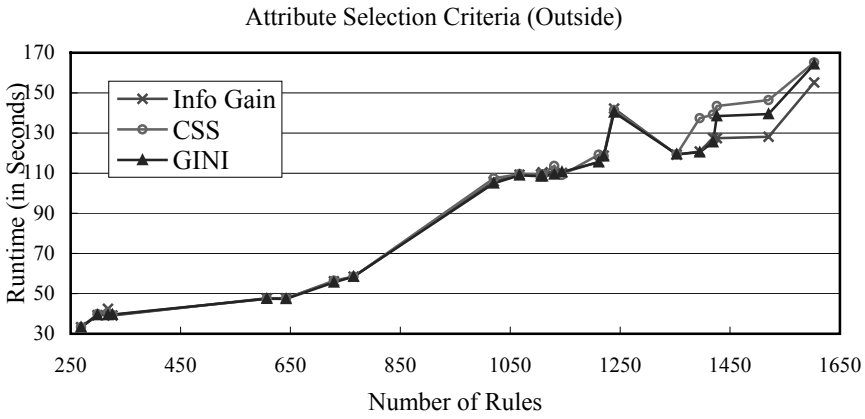


Fig. 7. A comparison of different attribute selection criteria on runtime performance for different rule sizes. (outside.tcpdump file)

In order to find how great ECPM can serve improving the detect engine we compare our Snort-DTECPM with Snort-NG.

Fig. 8 and 9 show that Snort-DTECPM offers good overall improvement compared to the Snort-NG. The improvement of Snort-DTECPM is typically between 25% and 48%, and can be as high as 51%. Snort-DTECPM is faster because, in the common case, it can quickly decide that a given signature string is not contained in a network packet's payload.

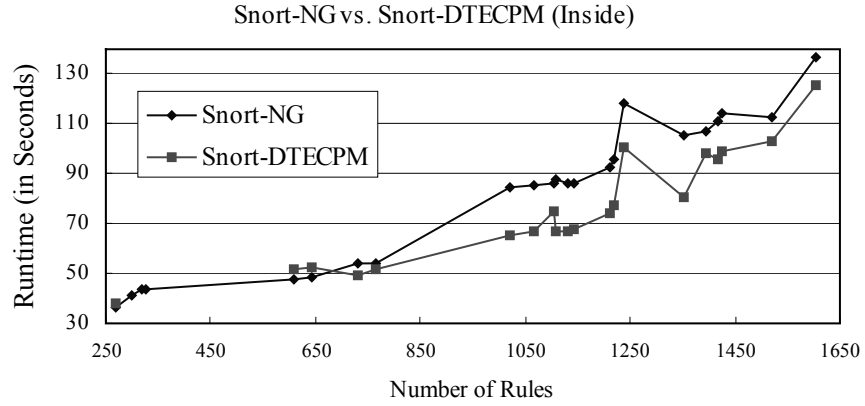


Fig. 8. A comparison of Snort-NG and Snort-DTECPM on runtime performance for inside.tcpdump file.

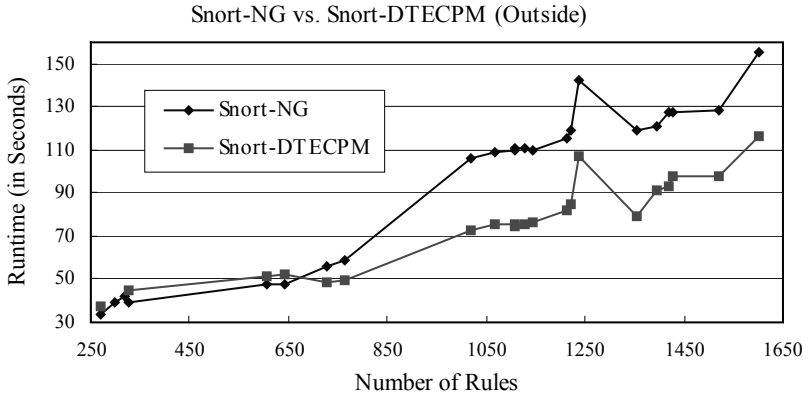


Fig. 9. A comparison of Snort-NG and Snort-DTECPM on runtime performance for inside.tcpdump file.

6 Conclusions

Decision tree detect engine is a very great choice for signature-based NIDS. When rule sizes are small and medium (suitable for special purpose NIDS whose rule sizes are usually not too large), GINI is better than Information Gain and CSS when building decision tree detection engine, while Information Gain is better than CSS and GINI when rules size are very large (in the circumstance of general purpose NIDS). So we can choose different kinds of decision tree detection engine for different situation. In addition, we proposed a fast exclusion and condensation based pattern matching algorithm ECPM which extend the best NIDS payload matching algorithm E2xB by adding a module to condense the large signature patterns. We also demonstrate that ECPM and decision trees can be integrated successfully. The resulting detection engine Snort-DTECPM can improve NIDS performance greatly. Since Snort-DTECPM is virtually independent of Snort itself, it is easy to adapt it to other similar signature-based IDS.

Acknowlegments

This research was supported by the Foundation of Chinese National 985 Project "Educational Information and Technology Platform" in Beijing Normal University, and by the Undergraduate Science Research Foundation of Beijing Normal University.

References

1. Martin Roesch: Snort: Lightweight Intrusion Detection for Networks. In USEBIX Lisa 99 (1999)
2. K. G. Anagnostakis, E. P. Markatos, S. Antonatosj, M. Polychronakis: E2xB: A Domain Specific String Matching Algorithm for Intrusion Detection. In Proceedings of the 18th IFIP International Information Security Conference (SEC03), May (2003)
3. Boyer R., Moore J.: A Fast String Searching Algorithm. Communications of the ACM, 20(10):762–772 (1977)
4. Firestorm NIDS: http://www.scaramanga.co.uk/firestorm (2005)
5. Jiawei Han, Micheline Kamber: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, August (2000)
6. Håvard Bakke, Pål Erik Eng, Erik Mellem, Frode Olsen: S. Antonatos, K. G. Anagnostakis, E. P. Markatos, M. Polychronakis: Performance Analysis of Content Matching Intrusion Detection Systems. Proceedings of the International Symposium on Applications and the Internet (2004)
7. J. K. Martin. An Exact Probability Metric for Decision Tree Splitting and Stopping. Machine Learning, 28:257—297(1997)
8. U. M. Fayyad, K. B. Irani: The Attribute Selection Problem in Decision Tree Generation. In Proceedings of the 10th National Conference on Artificial Intelligence. MIT Press, Cambridge, Massachusetts 104-110 (1992)
9. J. Mingers: An Empirical Comparison of Selection Measures for Decision Tree Induction. Machine Learning, 3:319—342 (1989)
10. W. Buntine, T. NiblettL: A Further Comparison of Splitting Rules for Decision-tree Induction. Machine Learning, 8:75—85 (1989)
11. J. R. Quinlan: Induction of Decision Trees. Machine Learning, 1(1):81—106 (1986)
12. J. R. Quinlan: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Manteo, California (1993)
13. R. L. Rivest: On the Worst-Case Behavior of String-Searching Algorithms. SIAM Journal on Computing, 6(4): 669–674, December (1977)
14. S. Wu and U. Manber: A Fast Algorithm for Multi-pattern Searching. Technical Report TR-94-17, University of Arizona (1994)
15. DARPA: http://www.ll.mit.edu/IST/ideval/ MIT Lincoln Labs. (2005)
16. R. Kothari, M. Dong: Decision Trees for Classification: A Review and Some New Results. In Lecture Notes in Pattern Recognition, S.R. Pal and N.R. Pal, (Eds.), Singapore, World Scientific Publishing Company (2001)
17. Christopher Kruegel, Thomas Toth: Using Decision Trees to Improve Signature-based Intrusion Detection. In Proceedings of the 6th International Symposium on the Recent Advances in Intrusion Detection, Lecture Notes in Computer Science (2003)
18. SNORT: http://www.snort.org (2004)
19. A. Aho and M. Corasick: Efficient String Matching: An Aid to Bibliographic Search. Communications of the ACM, Vol. 18, No. 6, pp. 333-343 (1975)
20. M. Fisk, G. Varghese: An Analysis of Fast String Matching Applied to Content-based Forwarding and Intrusion Detection. Technical Report CS2001-0670 (updated version), University of California - San Diego (2002)

Development of Users Distribution in Enterprise Systems with limited Buffer Size in Application Servers

Ping-Ho Ting[1], Kuan-Ching Li[2], and Chun Chung Wei[3]

[1] Dept. of Hospitality Management, TungHai University
Taichung 40704, Taiwan
ding@thu.edu.tw
[2] Dept. of Computer Science and Information Management, Providence University
Shalu, Taichung 43301, Taiwan
kuancli@pu.edu.tw
[3] Dept. of Information Management,ChungChou Institute of Technology
Yuanlin, Chunghwa 51003, Taiwan
ccwei@ms15.url.com.tw

Abstract. As enterprises worldwide race to improve their real-time management turnaround, which is essential requirement to improvements in productivity and service deployments, and therefore, large amount of resources have been invested into Enterprise Systems (ESs). All modern and robust ESs adopt a n-tier client-server architecture, which includes several application servers to hold users and applications. Currently, most web systems are stateless, which means that each request is routed independently to a different server at each time. However, for ESs, each request from same user is routed to the same application server.

Distributions in application and web servers are different in granularity. In the former scenario, a user represented by a set of transactions is the atomic element, while in the latter scenario, single request is the atomic element and different requests issued by the same user can be directed to different web servers. Until present time, few researches have been devoted in the user distribution to application servers in n-tier architecture.

In this paper, it is proposed a Heuristic Buffer Constraint Clustering Algorithm, namely HBC^2A , which is a Greedy-based strategy algorithm. The algorithm give suggestions of user distributions, the number of servers needed, and the similarity of user requests in each server. In addition, this algorithm is applied on a set of real data which is derived from the access log of an Enterprise System, in order to evaluate the quality of suggested distributions.

1 Introduction

For ESs that process daily business transactions, users typically have low tolerance on system performance. If a system responds to data entries or queries too slowly, users lose patience and complain loudly. Yet, the number of ES users keeps growing in most companies as the number of business processes incorporated into ESs increases. Therefore, keeping response time under control is a vital issue for most system administrators. To boost performances, activating more than one application servers become common practices in the industry.

When an ES has multiple application servers, distributing users to similar applications and application servers plays an important role in tuning overall system performance, as pointed out by documents of major ES systems [1, 13]. Throughout our text, an application in an ES corresponds to an atomic and unbreakable transaction, transactions and applications are used interchangeably.

In current practices, ESs do not automatically switch users to other application servers, due to the resources involved in the transmissions. As a user logs onto an application server, all related data such as authorizations, preferences, and created data are collected in the server's virtual memory, in order to create time-sharing working environment and reduce user effort in keying data. Besides, all applications executed by users are compiled and stored in memory. Data accessed by applications are also cached in memory to

improve the efficiency of systems. In many cases, the amount of data cached are huge, and as consequence, transferring a user to a different application server may trigger a transmission of huge amount of data. Thus, users are not switched automatically among servers in current practice.

In ESs, the dispatching mechanism needs to consider two criteria to gain reasonable performance: the number of users log on in each application server and the collection of applications executed in servers. Therefore, as each user consume hardware resources and the n-tier architecture has more than one application server, user distribution becomes one of the important issues in tuning ES performance [2]. In ESs, each application is evoked by a user who logs on an application server, and stays connected to the server for an entire working session, which can last for several hours and includes the execution of a set of applications. Therefore, admitting a user into an application server is equivalent to admitting a set of transactions into an application server, which marks a sharp difference between the distribution of application severs and traditional web servers.

In traditional web servers, requests are examined individually and those issued by the same user can be routed to different web servers. Commercial products, such as SAP R/3, equipped with a simple dispatching algorithm, considers only number of users and server response time. The task of grouping users is left to system administrators [1, 13]. In addition to the rough guideline of grouping financial users into one server and logistic users into another, system administrators need specific suggestions, such as explicit user distributions, the number of servers needed and the similarity of user requests in each server. To address the needs, this research paper proposes algorithm to suggest distribution based on user profiles. The distribution algorithm can the least number of servers needed that satisfy all the constraints of a system.

The scheme of the proposed research is shown in figure 1. The procedure is started with the collection of user profiles from an enterprise system. The profile is consisted of a set of transactions accessed by users in a given period of time. The transactions that are accessed frequently are labelled as regular transactions. The frequent accesses are compared against profile support threshold and user support threshold. The purpose of profile support is to screen transactions that are seldom used by all users and user support threshold is to find transactions which are accessed frequently by each user. The regular transactions are further analyzed to form associated regular transaction in the third step with confidence threshold. Associated regular transactions are designed to predicted the behavior of new and not frequent users, who do not have enough records in the user profile. In the distribution, regular transactions are used to cluster users with the novel algorithm prosed in this research paper, namely HBC^2A .

To explain the algorithms and related procedures, the rest of the paper is organized into the following sections. Applications are grouped into large itemsets with traditional Apriori algorithm[7, 11] to find frequent patterns. The process is explained in section 2. A group of users forms a cluster if the union of the users' transaction sets has an Application Match Ratio(AMR) exceeds a given threshold. AMR is a similarity measure of user patterns grouped in the same set. The definition of AMR and related properties are proved in section 3. An example of AMR based hybrid distributing approach is shown in section 4 Simulations with real data and comparisions with Round-Robin users distribution are shown in section 5. A review of distributed web server architectures and clustering of categorical sets is shown in section 6. Conclusion and possible extension of HBC^2A are discussed in section 7.

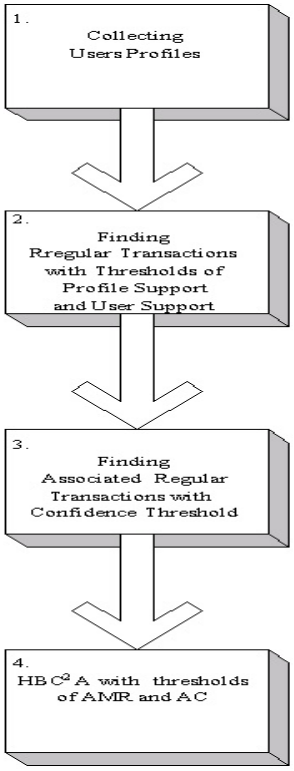


Fig. 1. Research scheme.

2 Finding Users' Regular Transactions

To record system and user statuses, most enterprise systems include various tracing mechanisms. Among the various recordable data are user sessions and applications executed in sessions. For the purpose of the paper, these data are transformed into user profiles. A user profile is a set of $\langle user - id, transaction - set \rangle$, where user-id is the account name of a user and transaction-set is the set of transactions accessed by the user in a session. A sample user profile is shown in Table 1, which records the sessions of ten users. User 1, 3 and 6 have more than one sessions in the profile. User 1 access transaction A, B, E, F, and H in one session and A, B, E, and F in another session.

Table 1. User Profiles

User-Id	Transaction-Set
1	{A, B, E, F, H}
1	{A, B, E, F}
2	{A, B, E, F, G}
2	{A, B, E, H}
3	{A, B, E}
3	{B, E, F, H}
4	{I, J, K, L}
5	{B, I, J, K}
6	{B, I, J, L}
6	{B, I, J, K}
7	{O, P, Q, R}
8	{O, P, Q, R}
9	{P, Q, R, K}
10	{W, X, Y}

As careful readers may have found that the transactions accessed by user 10 in the profile shown in Table 1 is special because most of his/her transactions are unique and are not shared by others. Transaction G of users 2 in the first session is also unique. If the rarely used transactions are all stored in buffers, large sizes of buffers are needed and the utilization rates of these buffers are low. Therefore, only regularly accessed transactions are considered. A user's regularly accessed transactions,termed as regular transactions, are transactions which occur in enough number of sessions in the corresponding user profile and are accessed often enough by the user.

Definition 1 *Given a user,u, a user profile, U, and a transaction, t, t is one of u's regular transaction in U if*

$$\frac{|\{s|t \in s.\textit{transaction-set}, s \in U\}|}{|U|} \geq \textit{profile support threshold}, \textit{and}$$

$$\frac{|\{s|s \in U, s.\textit{user-id} = u \wedge t \in s.\textit{transaction-set}\}|}{|\{s|s \in U, s.\textit{user-id} = u\}|} \geq \textit{user support threshold}.$$

Profile support threshold and user support threshold are given by system administrators. The higher the threshold, the fewer the regular transactions users have.

To compute or estimate regular transactions for each user, three steps are employed. The first one computes large itemsets with any existing set oriented pattern discovering algorithm, such as [4, 14]. The large itemsets computed from the algorithms have supports higher than the profile support threshold in the associated user profile. In the second algorithm, each large 1-itemset is examined against each user to form users' regular transactions. For new users who do not have accumulated enough entries to computer personal regular transactions, the paper propose to predicate their regular transactions with the association rules computed with known algorithms. Figure 2 shows the stages in computing regular transactions.

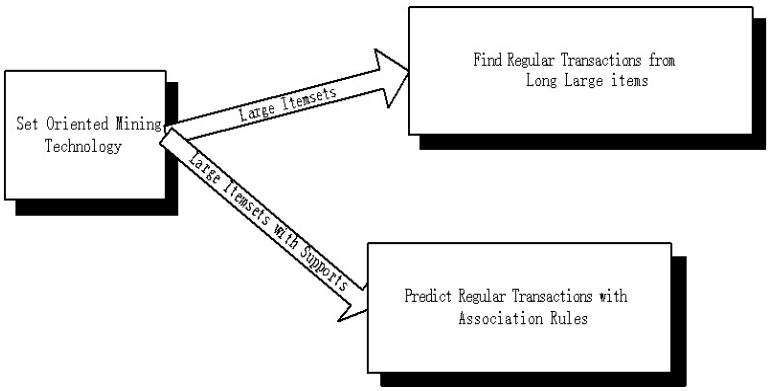


Fig. 2. The Stages of Computing Regular Transactions

If profile support threshold is set at 20%, the set of level 1 large itemsets of the sample user profile is {A, B, E, F, H, I, J, K, P, Q, R}; the level 2 set is {AB, AE, AF, BE, BF, BH, BI, BJ, EF, EH, IJ, IK, JK, PQ, PR, QR}; the level 3 set is {ABE, ABF, AEF, BEF, BEH, BIJ, IJK, PQR}; the level 4 set is {ABEF}. Therefore, the set of patterns generated from the Apriori-Like Algorithm is {A, B, E, F, H, I, J, K, P, Q, AB, AE, AF, BE, BF, BH, BI, BJ, EF, EH, IJ, IK, JK, PQ, PR, QR, ABE, ABF, AEF, BEF, BEH, BIJ, IJK, PQR, ABEF}.

The second step in computing users' regular transactions is to map transactions in large itemsets to users. A transaction is a user's regular transaction if it happens in enough number of the user's sessions. One obvious way to do so is taking every Level 1 large itemsets and check it against each users' transaction sets. The itemset is one of the user's regular transaction if the item occurs in enough number of the user's transaction sets.

Assume the user support threshold is set at 40%, the regular transactions of the the running example is shown in Table 2.

Table 2. Regular Transactions

User-Id	Regular Transactions
1	{A, B, E, F, H}
2	{A, B, E, F, H}
3	{A, B, E, F, H}
4	{I, J, K}
5	{B, I, J, K}
6	{B, I, J, K}
7	{P, Q, R}
8	{P, Q, R}
9	{P, Q, R}
10	∅

New users do not have any records in the user profiles and do not have associated regular transactions. However, dispatching programs still need to dispatch them in run-time. Therefore, help for dispatching programs to guess the patterns of new users are in order.

If each new user provides one of the transactions she/he wishes to access after logging on, the dispatching program can check if the transaction has high association with any large itemsets. If so, the union of the large itemsets dentoe the user's Predicted Regular Transaction set.

Definition 2 *The Associated Regular Transactions of a transaction, t, under a set of large itemsets, P, a user profile, U, is*

$$AT(t) = \cup\{p \in P \mid t \in p, CP_U(p|t) \geq \textit{confidence threshold}\},$$

where $CP_U(p|t) = \frac{|\{s|s \in U, p \in s.\textit{transaction set}\}|}{|\{s|s \in U, t \in s.\textit{transaction set}\}|}$

By setting the confidence threshold at 80%, the Associated Regular Transactions of transactions in large-1 itemsets in the running example is shown in Table 3.

Since the algorithms needed to find the Associated Regular Transactions are trivial when large itemsets are ready. The paper does not include the algorithm either.

3 Clustering and Distributing by *HBC²A*

Systems with multiple servers gain performance speed at the cost of keeping duplicated programs and data in more than one servers. In sophisticated application servers with hundreds or thousands of users on-line all the time, the memory needed are considerable [2]. Therefore, users share similar transactions are grouped

Table 3. Associated Regular Transactions with Confidence Threshold at 80%

Transaction	PT	Confidence
A	ABE	100%
B	AB	100%
E	ABE	83%
F	BEF	100%
H	BEH	100%
J	IJK	100%
K	IJK	100%
P	PQR	100%
Q	PQR	100%
R	PQR	100%

into one cluster, which is then assigned to an application server. This section proposes HBC^2A to cluster users and a straightforward algorithm to distribute clusters.

Definition 3 *A cluster is a set of users that share common applications in an Enterprise system.*

The quality of a cluster is measured by AMR , Application Match Ratio. The AMR of a cluster is defined as the ratio of AC versus the applications in the cluster, where AC denotes the number of applications that can be hosted in an application server without causing buffer swap. AMR is smaller than one when users in the cluster have more regular transactions than the buffers can hold. In this case, buffer swap occurs and the smaller the AMR is, the more the buffer swap will occur.

Definition 4 *The AC of an enterprise system is an integer number. The number denotes the number of applications that can resides in application servers of the enterprise systems without causing buffer swap.*

The regular transactions in a cluster are defined as the union of regular transactions of users grouped in the cluster.

Definition 5 *The number of regular transactions in a cluster, c, is defined as*

$$\|c\| = |\cup_{u \in c}\, u.regular\ transactions|$$

The AMR of a cluster, c, is defined as the ratio of AC to $\|c\|$. $AMR\ (c) = \frac{AC}{\|c\|}$.

Lemma 1 *The AMR of each cluster has a value between 0 and AC .*

Proof

AMR 's are positive and therefore are always greater than to 0.

Given a cluster, c

$$\begin{aligned} AMR\ (c) &= \frac{AC}{\|c\|} \\ &\leq \frac{AC}{1} \\ &\leq AC \end{aligned}$$

AMR of a cluster, therefore has values between 0 and AC .

□

Hence, system administrators can assign an AMR threshold between 0 and AC . By setting the threshold is between 0 and AC , the system administrators can tune the tolerance degree of buffer overflow.

Theorem 1 Anti-Monotonicity of AMR *AMR of a cluster decreases with the addition of any user with non-empty regular transaction set to the cluster.*

Proof

If a cluster, c, has the AMR of $\frac{AC}{p}$ *where p is the number of different transactions in the cluster. If a user with q new transactions is added to the cluster then the new AMR is* $\frac{AC}{p+q}$.

$$\begin{aligned}\frac{AC}{p} - \frac{AC}{p+q} &= \frac{AC * (p+q) - AC * p}{p * (p+q)} \\ &= \frac{AC * q}{p * (p+q)} \\ &\geq 0\end{aligned}$$

The case of $\frac{AC*q}{p*(p+q)} = 0$ *occurs when q=0, which means the regular transaction set of the new user does not contain any new transactions.*

□

Therefore, AMR has the property of Anti-Monotonicity, which means that adding a user to a cluster can only reduce the AMR of the cluster, unless the new transaction set does not contain any new transactions. The property can be used to prune hapless candidate clusters that have AMR under a threshold in the cluster forming algorithm, HBC^2A . In this paper, system administrators are requested to supply an AMR threshold. Candidate clusters with AMR smaller than the threshold are discarded.

Theorem 2 *The threshold of AMR must be smaller than or equal to* $\frac{AC}{|t_{max}|}$ *, where* t_{max} *is the largest regular transaction set in the user profile , to have all users grouped into at least one cluster.*

Proof
Any cluster c containing users with t_{max} *has AMR (c)* $\leq \frac{AC}{|t_{max}|}$. *If the threshold is larger than* $\frac{AC}{|t_{max}|}$, *then the users can not be included in any cluster.*

□

Definition 6

– *A qualified cluster is a cluster whose AMR exceeds a given threshold.*

- *A set of clusters is comprehensive under a user profile, U, if the union of the clusters includes all users with regular transactions in U.*
- *A set of clusters is disjoined if the intersections of any two clusters are empty.*
- *A set of qualified clusters is a distribution under a user profile, U, if they are comprehensive under U and disjoined.*

In the running example, if *AC* is set at 3, and *AMR* threshold at 0.5 , then the cluster of {1,2,3} ,{4,5,6} and {7,8,9} have *AMR* of 0.6, 0.75 and 1, respectively. The set composed by the three clusters is comprehensive, disjoined and forms a valid distribution. The running example is shown in Table 4.

Table 4. A set of qualified clusters when AC=3 and AMR=0.5

Qualified cluster	Users	Regular Transactions	AMR
Cluster 1	1,2,3	A,B,E,F,H	0.6
Cluster 2	4,5,6	B,I,J,K	0.75
Cluster 3	7,8,9	P,Q,R	1

We propose a Heuristic BC^2A , namely HBC^2A , HBC^2A returns distributions that satisfy constraints with the fewest number of clusters, and the rules associating single transactions to predicted regular transactions. The constraints include *AC* , an *AMR* threshold, profile support threshold, user support threshold, and rule confidence threshold. The recommendations guarantee that when all frequent users logging on the system and accessing all regular transactions, each server still has an *AMR* above the given *AMR* Threshold. Information included in the recommendations are number of servers, clusters of users, and *AMR*s of clusters.

The HBC^2A includes three steps in computing the recommendations - computing the set of qualified clusters and selecting clusters to form distribution. The main steps are listed as following:

Initialization: for each user with regular transactions, and these users form qure, Q. Sort Q on users by the number of their regular transactions and form new queue, Q'.

Composing C_i from Q': A user u_i in Q' is added to C_i by the user in Q' from C_i if the new cluster C_i has an *AMR* value exceeding the given threshold. In the mean time, Removing the new user from Q'. Repeating the step until C_i has an AMR value lower than the given threshold.

Repeating the Last Step Until Q' is emptyset : If Q' is empty then HBC^2A has found all qualified clusters in C_1, ..., and C_i; Otherwise, HBC^2A has to repeat the last step.

The algorithm returns all the distributions that satisfy the requirements with the least number of application servers and let system administrators to decide which distribution they prefer.

4 An *AMR* Based Hybrid Dispatching Approach

Each ES typically has a dispatching program listening to networks and accepts user requests. The program resides an application server, intercepts user requests, and direct them to application servers.

Assuming the system administrator in our running example picks the distribution of {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}. The case of user 1, 2, 4, 7, and 8 have logged on and user 5 and 6 are waiting in the web server is depicted in Figure 3.

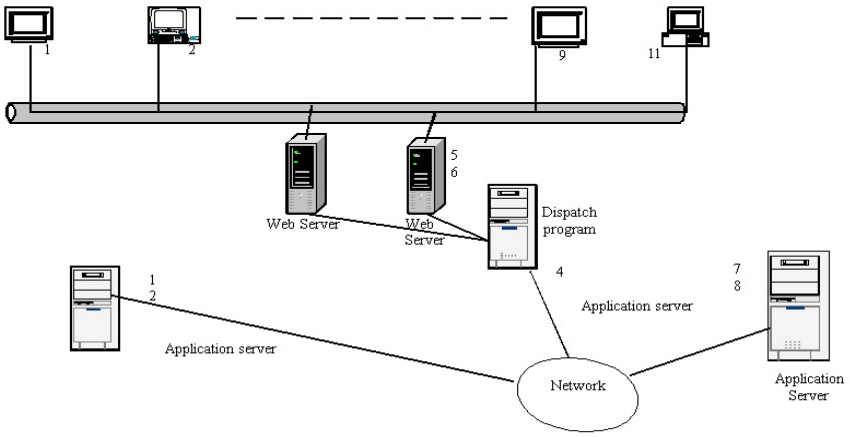


Fig. 3. Users are Distributed through a Dispatching Program

The distributions suggested by HBC^2A bases on frequent patterns in user profiles. For new and infrequent users, HBC^2A does not suggest their distributions directly but returns association rules, PR (Prediction Rules), in the output to help dispatching program make the decision. To apply the rules, a new user only needs to provide a transaction he/she plan to evoke after logging on the ES. With the association rules, a dispatching program can distribute a user according to its associated predicted regulation transactions. If the first transaction does not lead to any predicted regular transactions, then the single transaction works as the basis for dispatching.

The running example is shown in figure 3. An *AMR* Based Hybrid dispatching algorithm distributes users while keeping the *AMR* of each server as high as possible. In the dispatching procedure, users are distributed to a server according to one of the three alternatives:

- If a regular user logs on, then send the user to the recommended server and return to listening mode.
- If an infrequent user logs on with a transaction, then find the predicted regular transactions implied by the transaction. If no entry matched then the single transaction is treated as the predicted transaction.
- Compute the potential new *AMR* in each server with the addition of the user. Assign the user to the server with the highest *AMR* , and update the *AMR* in the corresponding server.

The distribution in the running example has *AMR* s of 3/5, 3/4, and 1 in the three servers. If a new user with user-id 11 wishes to log on the system and submits an A as the first transaction then the user has a predicted regular transaction set of ABE, according to Table 3. The *AMR* after adding ABE to the three servers would be 3/5, 3/6, and 3/6, respectively. Because the first server has the highest AMR value, the new user is distributed to the first server, and the distribution becomes {1, 2, 3, 11}, {4, 5, 6}, and {7, 8, 9}.

5 Simulation

Several experiments are conducted on real data collected from a mid size machinery company based in Taichung, Taiwan. The company has their SAP system up and running since 2002. Five weeks of user access logs are extracted from the system to perform the experiment. Four weeks of the data are used to suggest distributions. The fifth week of data are used to evaluate the quality of the suggested distributions.

In the experiment, 1,853,689 access logs are collected which include 56 users have regular patterns. The average number of transactions in user profiles is 7.7. The quality of suggested distributions are measured by Application Hit Ratios and Entropy. The Application Hit Ratio of a server is defined as the number of transaction accesses hits a stored version of the transactions in the memory over the total number of transactions accessed in the server. The Application Hit Ratio of a distribution is the average Application Hit Ratios of servers suggested in the distribution. The entropy of a server is defined as $-\sum p_i \log_2(p_i)$, where p_i is the probability of transaction i being accessed by users in the cluster. Since AR and AMR thresholds are typically smaller than 1, some frequent transactions are not stored in the memory. In the experiment, we assume that servers automatically store the applications that are accessed the most in the training data in the memory. Infrequent users appearing in the testing data are assigned to servers according to the hybrid distribution algorithm.

The Experiment of *HBC²A* and Round-Robin are implemented on Matlab 6.1 and executed on a Pentium 4-1.8 GHz Microsoft XP Server system with 256 Megabytes of main memory. With the improved algorithm, distributions can be suggested within one minites.

5.1 Experimental Results of *HBC²A*

Seven distributions are suggested against the collected data. These distributions have *AMR* threshold set at 0.8 with AC ranging from 21 to 28, respectively. These simulations all have profile support threshold at 0.1 and user support threshold set at 0.3.

The number of servers needed for each distribution is shown in Figure 4. With *AMR* =0.8, *HBC²A* suggests a distribution with five servers when AC=21, four servers when AC = 22, three servers when AC = 23 , two servers when AC=24, 25, 26 and 27, and 1 server when AC = 28.

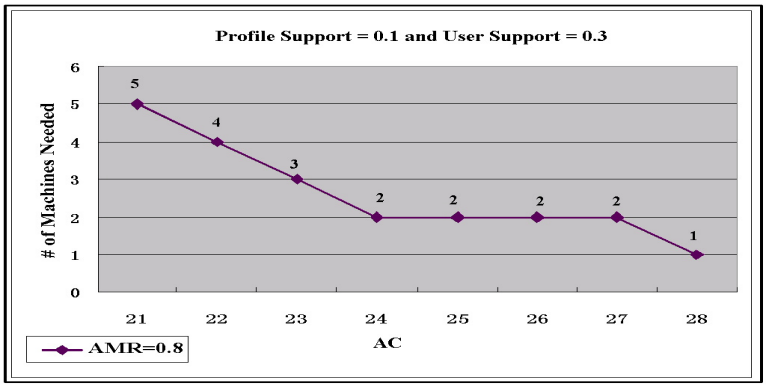


Fig. 4. Distribution Experiments with *HBC²A*

The quality of each distribution is evaluated in Figure 5 and Figure 6. The Application Hit Ratio of the distributions with AMR =0.8 is 0.91714 when AC = 21, 0.94924 when AC = 22, 0.95687 when AC = 23, 0.95674 when AC = 24 , 0.96696 when AC = 25, 0.96565 when AC = 26, 0.96474 when AC = 27 and 0.9559 when AC = 28. The Entropy of the distributions with AMR =0.8 is 6.6595 when AC = 21, 8.3807 when AC = 22, 9.542 when AC = 23, 10.0652 when AC = 24 , 10.0263 when AC = 25, 10.435 when AC = 26, 8.7664 when AC = 27 and 11.755 when AC = 28.

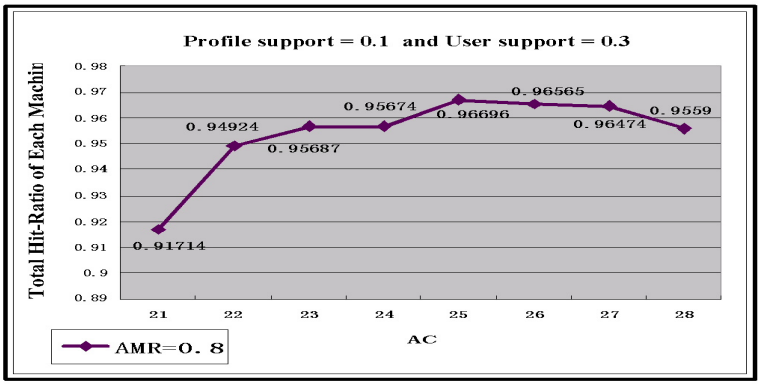


Fig. 5. The Application Hit Ratios of Distributions with HBC^2A

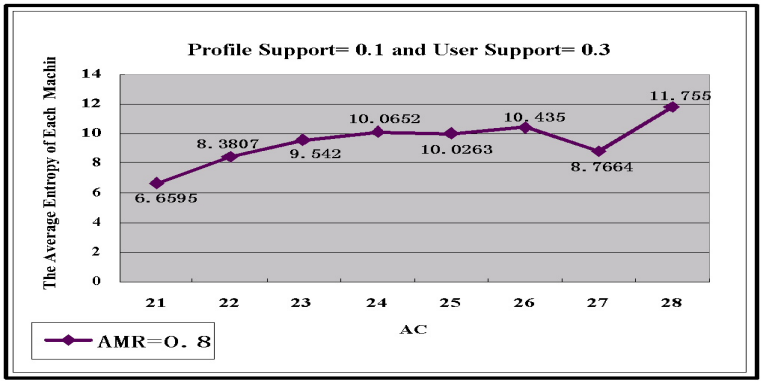


Fig. 6. The Entropy of Distributions with HBC^2A

From data shown in Figure 4, Figure 5, and Figure 6, we find that the hit-ratios of the distributions ranging from 0.91714 to 0.96696. From figure 5, we find that the more AC , the more average the higher hit-ratios machines have, because each machine can hold more transactions. The more number transactions manchine can hold, The more chances the high-ratios machine have. From figure 6, we find that the more AC, the more average entropy machines have. We conclude that the company should use one server to hold all users if hardware capacity is large enough. The second to the best distributions have Application Hit Ratios of 0.96696 which occurs when AMR =0.8, AC = 25(two machined need). Since the former settings requires fewer memory resource, system administrators are advised to adapt the former distribution.

5.2 Comparision of HBC^2A and Round-Robin User Distribution

HBC^2A considers the constraints and tries to find groups of users whose combinations of accessed transactions do not cause too many page faults if they are clustered into one application. Round-Robin distributes user to one of the several application servers in a server group by a rotated order. The approach ensure users are fairly distributed in a server group.

For the purpose of comparision, We set the same memory constraints to HBC^2A and Round-Robin. In terms of users and transactions allocations, HBC^2A can get better result than Round-Robin since given the same number of machines, the transaction distributions in HBC has lower entropy than Round-Robin, the hit-ratio of user distributed in each machine by HBC^2A get better result than Round-Robin. (Please refer figure 7, figure 8).

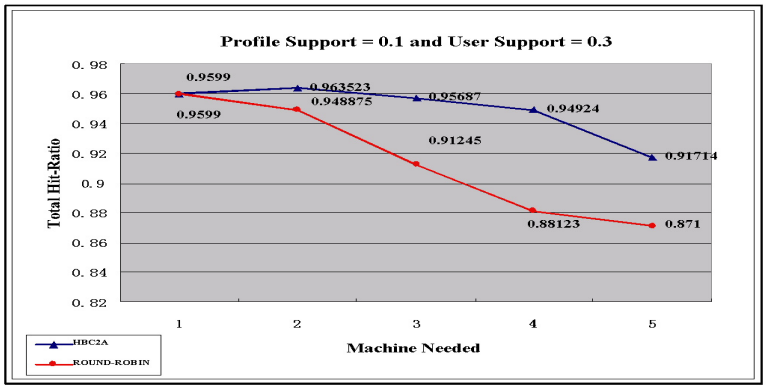


Fig. 7. Comparision of HBC^2A and Robin in Hit Ratios of the Experiment

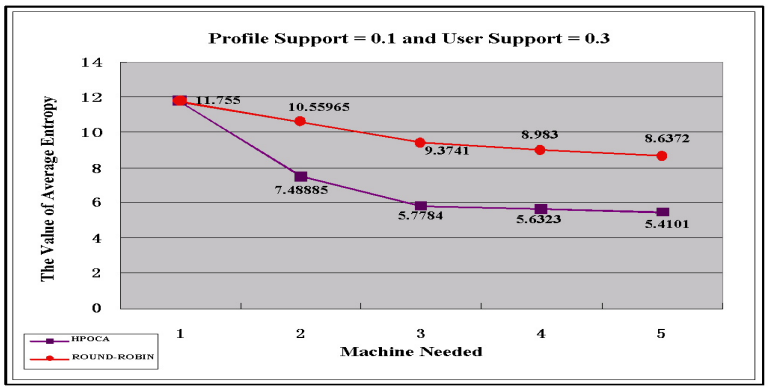


Fig. 8. Comparision of HBC^2A and Robin in Entropy of the Experiment

In summary, if the memory of each machine is seriously limited , HBC^2A should be used to distribute users, because the result generated by them is guaranteed to satisfy the memory consumption criteria.

6 Related Work

With the Internet rush, many researches have been devoted to distribute user requests in Distributed Web Server Architecture, in order to improve the performance of web servers. Depending on the locations where

request distributions happen, these researches are classified in client-based, DNS (Domain Name Server)-based, dispatcher-based, and server-based, as in [6, 5, 18, 8, 20]. Since current `Http protocol` is stateless, each request is routed independently to a web server[5, 3, 16, 17, 19]. All of the above researches assume that requests can be independently routed to different servers, where as in the application servers of ESs, requests from the same users have to be routed to the same server.

Clustering literatures are classified into two models: partitioning clustering and hierarchical clustering [15, 9, 12]. If k clusters are needed, partitioning clustering choose k centroids initially and gradually, tune the constituents of each clusters or centroids with some criteria function until a locally optimized characteristic is reached. Hierarchical clustering can be further divided into agglomerative and divisive clustering. As the name suggested, agglomerative clustering gradually merge smaller clusters into larger clusters until k clusters are found. Divisive clustering, on the other hand, splits larger clusters into smaller clusters until k clusters are found.

Most clustering algorithms employ Euclidean distances to compute similarity. The shorter the distances the more similar the data points in the clusters are. However, Euclidean distances are not ideal for clustering categorical data. For example, to cluster transaction sets with Euclidean distances, each set has to be translated into a sparse binary vector. In the running example, the second session of user 1, {A, B, E, F} is translated into $\langle 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \rangle$. The huge number of zeros can easily skew the distances between transaction sets. For example, a transaction set of {A} is translated into $\langle 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \rangle$ and {I} is translated into $\langle 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0 \rangle$. Since {A} and {I} have a distance of two bits, and {A} and {A, B, E, F} have a distance of three bits, the former pairs has shorter distance than the latter. The conclusion violates the general perception of set operations. Therefore, Euclidean distances are not ideal for clustering categorial data.

Many set oriented algorithms use Jaccard coefficient [15] to compute distances. Given two sets T_1 and T_2, their Jaccard coefficient is $\frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$. However, Jaccard coefficient has two drawbacks for our application. The first is that it cannot describe the number of elements in each cluster, which are important to calculate the buffer efficiency. The second is that Jaccard coefficient is not accurate in computing the similarity between transactions sets. For example, the Jaccard coefficient of {A,B,C},{A} and {A,B,C},{B,C,D} are 1/3 and 2/4, respectively. However, in *HBC*^{2}*A* , the distance of the former pair is 0, since {A, B, C} include {A}. Another major work in clustering categorical data is ROCK [10], which proposes to cluster transaction sets based on links between nodes, which are composed by common neighbors between any pair of nodes. A common neighbor of two transaction sets is a transaction set sharing similar items with the two sets. ROCK puts two elements into the same cluster if the count of common neighbors exceed certain threshold. ROCK also has the same drawbacks as Jaccard coefficient. For instance, if a profile includes transaction sets {A}, {A, B, C},{A, C, D},{B, C, D},{B, C, E} and the threshold of a qualified common neighbor(link) is set at 1/3 of Jaccard coefficient. The ROCK coefficient of {A,B,C},{A} and {A,B,C},{B,C,D} are 1 (due to the common neighbor {A, C, D}) and 2 (due to the common neighbor {A, C, D} and {B, C, E}), respectively. From view of ROCK, the similarity of the former pair is lower than later pair. On the other hand, the distance of the former pair is 0 in *HBC*^{2}*A* , which is more close to our intuition in the application of user distribution, since {A} is a subset of {A, B, C}. Many set oriented algorithms use Jaccard coefficient and ROCK. However, Jaccard coefficient and ROCK along cannot describe the number of elements in each cluster, which are

important to calculate the buffer efficiency. Hence, common categorial clustering technology is not suitable for clustering users in the application.

7 Conclusion

Managers in enterprises often add users to ESs, as they extend E-business practices to various divisions of corporate operations. With the addition of each user, new pressures on performances are brought upon to the systems. Yet, system response time is one of the most important factors in measuring user satisfactions.

Since ESs tend to consume considerable amount of hardware memory, application servers can easily run out all memory available, which induce to hardware limitations. When this happens, a common procedure adopted in boosting performance is adding application servers to ESs. With multiple application servers in the scene, distributing users with similar application requirements to the same application servers increases buffer utilization and lead time to next hardware upgrades.

The procedure of HBC^2A roots its development on AMR , which is a similarity measure of user transactions grouped in the same cluster. A cluster with high AMR means users in the cluster share similar applications under a given buffer limitation. AMR has the property of Anti-Monotonicity, which states that AMR of a cluster decreases with the addition of each new transaction set. With the property, HBC^2A can prune hopeless search branches and stop the iterations when an empty cluster set is found. Distributions are combinations of clusters which cover all users with regular transactions and each user is included in only one cluster. The distributions composed of fewest number of clusters are returned as suggestions.

Although frequent users and regular transactions are stable in ESs, new users are added to the systems from time to time. These users have no entries in user profiles and are distributed by a hybrid dispatching program that distributes frequent users according to a selected distribution and new users with dynamic A AMRs. A transaction of the new user is checked to find its predicted regular transactions. If an entry is found, the dispatching program associating the user with the predicated transactions, otherwise, the single transaction is associated with the user. The associated transactions are then used to decide the target server for the new user. The user goes to the server with the highest AMR after accepting the user.

As future work, HBC^2A is among a series of study in distributing users with historical user profiles, and are by no means the last two. Several issues require further studies, such as modelling user profiles with sequences, dynamically updating user patterns, incorporating CPU and systems loads into dispatching and distribution algorithms.

Acknowledgements

This study is supported by National Science Council, Taiwan, Republic of China, through the Project No.NSC94-2416-H-029-010-. We would like to thank anonymous referees for their invaluable comments on this work.

References

1. SAP AG. *System R/3 Technicale Consultant Training 1 - administration*, chapter R/3 WorkLoad Distribution. SAP AG, 1998.
2. SAP AG. *System R/3 Technicale Consultant Training 3 - Perf. Tuning*, chapter R/3 Memory Management. SAP AG, 1998.
3. Woo Hyun Ahn, Woo Jin Kim, and Daeyson Park. Content-aware cooperative caching for cluster-based. *The Journal of system and software*, 69(1):75–86, 2004.
4. R. Argawal and R. Srikant. Fast algorithms for mining associations rules. In *Proceedings of International Conference in Very Large Data Bases*, pages 487–499, 1994.
5. H. Bryhni, E. Klovning, and O. Kure. A comparison of load balancing techniques for scalable web servers. *IEEE Network*, 14:58–64, 2000.
6. V. Cardellini, M. Colajanni, and P.S. Yu. Dynamic load balancing on web-server systems. *IEEE Internet Computing*, 3:28–39, 1999.
7. Yen-Liang Chen, Ping-Yu Hsu, and Chun-Ching Ling. Mining quantitative assocation rules in bag databases. *Journal of Information Management*, 7:215–229, 2001.
8. Gianfranco Ciardo, Alma Riska, and Evgenia Smirni. Equiload:a load balancing policy for cluster web servers. *Performance Evaluation*, 46:101–124, 2001.
9. R. O. Duda and P. E. Hard. *Pattern Classification and Scene Analysis*. Wiley-Interscience Publication, 1973.
10. S. Guha, R. Rastogi, and K. Shim. Rock: A robust clustering algorithm for categorical attributes. *Information Systems*, 25(5):345–366, 2000.
11. J. Han and M. Kamber. *Data Mining: Concepts and Techniques*, chapter Mining association rules in large databases. Morgan Kaufmann Publisher, 2001.
12. J. Han and M. Kamber. *Data Mining: Concepts and Techniques*, chapter Clustersing. Morgan Kaufmann Publisher, 2001.
13. J.A. Hernándes. *The SAP R/3 Handbook*, chapter Distributing R/3 Systems. McGraw-Hill, 2 edition, 2000.
14. J. Pei J. Han and Y. Yin. Mining frequent patterns without candidate generation. In *Proceedings of ACM-SIGMOD International Conference on Management of Data*, pages 1–12, 2000.
15. A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.
16. P. Mohapatra and H. Chen. A framework for managing qos and improving performance of dynamic web content. In *Proceedings of Global Telecommunications Conference*, volume 4, pages 2460–2464, 2001.
17. S. Nadimpalli and S. Majumdar. Techniques for achieving high performance web servers. In *Proceedings of International Conference on Parallel Processing*, pages 233–241, 2000.
18. B. C-P. Ng and C-L. Wang. Document distribution algorithm for load balancing on an extensible web server architecture. In *Proceedings of International symposium on cluster computing and the Grid*, pages 140–147, 2001.
19. Victor Safronov and Manish Parashar. Optimizing web servers using page rank prefetching for clustered accesses. *Information Sciences*, 150:165–176, 2003.
20. Zhiguang Shan, Chuang Lin, and Dan Marineslu. Modeling and performance analysis of qos-aware load balancing of web-server cluster. *Computer Networks*, 40(2):235–244, 2002.

Temporal Mining of Recorded Collaborative Production of Artefacts

Matt-Mouley Bouamrane and Saturnino Luz

Department of Computer Science
Trinity College Dublin, Ireland
{Matt.Bouamrane,Saturnino.Luz}@cs.tcd.ie

Abstract. Identifying salient parts of time-based media in multimodal meeting recordings is particularly challenging, as they often contain important information which can not be easily visualised (audio) or summarised. In scenarios involving online collaborative activities which result in production of artefacts, certain self-contained information items are likely to be regularly manipulated. By capturing and timestamping space-based interactions among collaborators and the actions they perform on these items over time, one can define *semantic objects* with persistent histories. This information which is usually lost in common recording of multimodal meetings offers new possibilities for the data mining of recorded meetings. Potential links between semantic objects, while not necessary obvious when looking at a meeting's outcome can be uncovered by investigating temporal relationships between objects. Space-based actions also generally have strong associated semantics and are therefore appropriate for quick visual scanning. Drawing from our experience with the development of an integrated online collaborative writing environment with structured activity logging and post-meeting browsing system, we present a generic time-based artefact history model and a novel time-based data mining paradigm for extracting information from recordings of collaborative artefact-producing activities. A prototype browser, Meeting Miner, is presented which builds on these techniques.

1 Introduction

The complexity of most projects performed in the workplace means that on a daily basis many tasks need to be carried out by teams involving people with various responsibilities and fields of expertise, sometimes residing in different places. This often means that projects are completed through an iterative process, during which they are sent back on forth between different experts. Phases of individual work are punctuated by meetings of some sort to discuss progress, share ideas, take decisions or allocate tasks. As computers have become ubiquitous tools for communication, synchronous collaboration can be greatly enhanced by the possibility of recording meetings thus freeing participants from distracting and time consuming tasks such as note taking and meeting minutes

production. However, as the number of stored meetings grow, so does the complexity of extracting meaningful information from the recordings. Therefore, in order to be truly effective, a conferencing capture system needs to offer users efficient means of navigating recordings and accessing specific information.

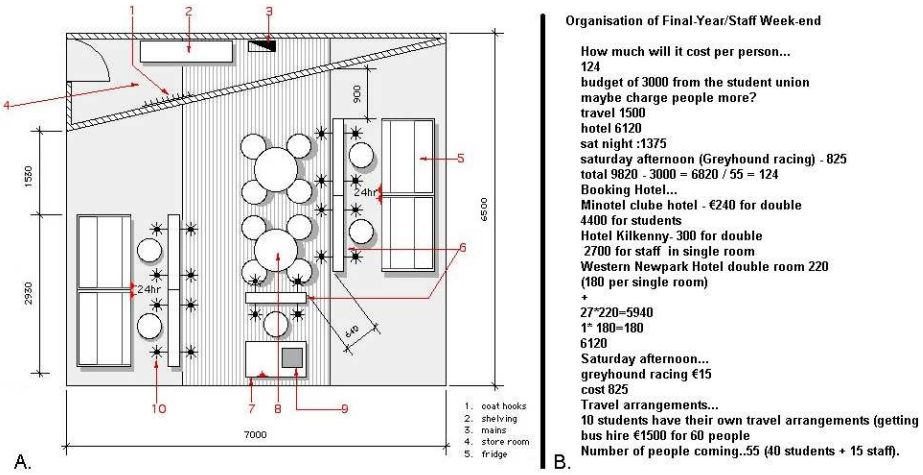


Fig. 1. Two examples of artefacts produced as outcomes of collaborative activities

In a specific scenario, which we will refer to as *artefact focused meetings*, one or many *space-based* items such as a textual documents, sketches, drawings, plans are either mentioned or produced during the meeting, either to support the decision-making process or in some cases, as the *goal* or focal point of the meeting (collaborative design). Fig. 1 illustrates two examples of possible outcomes of artefact focused meetings: a plan and a text document. Although the outcome is the obvious *product* of the collaborative meeting activities, it offers no clue about the often laborious *process* by which it was achieved. To illustrate this, consider the text of Fig. 1: it is a one sided A4 page document, yet it is the result of a collaborative writing task involving more than two hundred edits performed on a basic text skeleton. The document production process is either lost (outside the meeting participants' individual memories) or needs to be recorded in a content rich continuous medium such as audio or video. In the latter case, the problem now consists in accessing relevant parts of the content rich media. It is a common workplace practice to have many ongoing projects at the same time, some of which might be put on hold while some condition outside the remit of the office is resolved. Consider a person who did not attend a meeting or needs to find a specific information months after the last meeting took place. While presented with the meeting outcome, one can clearly see a set of information but

can be left wondering as to what are the reasons behind the choice of a specific item:

- *Does this choice of material comply with fire regulations?*
- *Why did they decide to book a hotel outside of town?*

To answer these questions without listening to the entire recording one needs means to access parts of the meeting recording where this information is most likely to be found. In other words, one needs efficient indexing and navigation tools. There has been growing research interest in the development of applications for visual mining of multimodal meeting data in order to support users' meeting browsing requirements [1,2]. In previous work we exploited temporal linkage patterns between text and speech to support meeting browsing [3]. In this paper we further generalise that approach to cover artefact focused meetings.

The paper is organised as follows: we first introduce related research work and existing systems developed for meeting browsing. We present a novel generic model for temporal mining of space-based artefact producing meetings. We then present the result of our own experience in developing an integrated online meeting capture and information mining architecture based on the design requirements earlier presented: the Meeting Miner.

2 Related Work

In recent years, modality translation from sequential data into the space domain: transcripts from audio through automatic speech recognition (ASR) and images from video (keyframes) has emerged as the dominant paradigm for continuous media navigation [4,1]. The MeetingBrowser [5,2] displays meeting transcripts time aligned with corresponding sound or video files. The browser comprises a number of components, including a speech transcription engine and automatic summarizer. The summarizer attempts to identify salient parts of the audio and present the result to the user as a condensed script, or *gist* of the meeting. The SCAN (Spoken Content based Audio Navigation) [6,7] system uses acoustic and prosodic features for audio segmentation and a number of information retrieval techniques applied on ASR transcripts for speech recording indexing. The SCAN user interface has three components: search, overview and transcript. The search component retrieves audio documents based on users' queries match against the ASR transcripts of the documents contained in the database. The ten highest ranking documents are displayed along with the number of hits (number of terms of the query contained in the document transcript). The overview displays audio segments as rectangles colour coded according to the terms from the user query and where the height is proportional to term Frequency. The MeetingViewer [8] is a client application for browsing meetings recorded with the TeamSpace [9,10] online conferencing system. TeamSpace's MeetingClient provides low-bandwidth video for participants awareness as well as supporting the use of a number of

artefacts such as sharing and annotating slide presentations, creating and editing agenda and meeting action items and inserting bookmarks. In addition to participants information (joining, leaving meeting) all interactions events performed on the clients artefacts are automatically recorded and timestamped by the server. These events are subsequently used to index the meeting and are displayed on a timeline on the MeetingViewer interface to facilitate navigation. COMAP (COntent MAPper) [11,12] is a system for browsing captured online speech and text meetings. The user interface displays the textual outcome of the co-authoring task along with mosaic timeline views of participants' speech and editing activities. An *Interleave factor* (IF) metric [13] measures levels of concurrent media activity, with intervals of greater activity deemed of greatest significance. A summary view of a recording can be generated through IF ranking. The Ferret Media Browser [14] is a client-server application for browsing recorded collocated multimodal meetings, with a combination of any available media for display and synchronised play-back. ASR transcripts, a key-word search and speech segmented according to speakers' identity are also available. Media streams can be dynamically added to or removed from the display during the browsing task. A detailed description of meeting browsing techniques and applications can be found in [15].

3 Object-based Temporal Mining

3.1 Key Concepts

The meeting scenario assumed in this paper is one where geographically dispersed users interact with a space-based "document" (see definition below) which acts as the focal point of the meeting, and can communicate through continuous media communication channels (audio and/or video). Furthermore, it is assumed that interactions with the spaced-based document will be computer mediated. This is important in the fact that participants' interactions need to be automatically detected and recorded. Before proceeding, we define the following key terms whose usage is somewhat peculiar:

Document The set of space-based artefacts used during the meeting. This is to a large extent the focal point of the meeting, either because the document supports the decision making process or is the meeting intended final outcome (work plans, technical drawings). The document can be collaboratively written text, slides for a presentation, graphs, drafts and plans, for collaborative design, audio or video clips, or any combination of these.

Data Objects Within the document, smaller data objects which can be treated and manipulated as individual semantic entities. The granularity of the semantic data objects is best defined according to the application scope. To illustrate this point, in common applications, a pixel or character would have no intrinsic semantics as opposed to a word, sentence, paragraph, a shape or image.

Primitive Operations Primitive operations will typically modify some property of a data object. Example of primitive operations are *insertion*, *deletion* of characters in text, modifying the texture, size, of an image or shape.

Manipulation Rules Manipulation rules need to associate an unambiguous and definite outcome when a data object manipulation affects other objects. Examples are *cutting* and *pasting* paragraphs or shapes, occlusions, etc.

Timestamp A timestamp records information about all primitive and manipulation operations previously defined. Information recorded are: the *agent* who performed the operation, the *nature* of the operation, the *time* of operation and unless unfeasible, the exact *content* of the operation. Note that this is often partially the case in many existing applications: the nature and content of a number of past operations are stacked for undos and in collaborative applications, some form of timestamping generally needs to be implemented to address concurrency and consistency issues [16,17,18].

Object Log For each object, a list of all timestamped actions (primitive and manipulation operations) which affected the object from creation throughout the meeting.

3.2 Object-based Mining

To provide access to multimodal recordings, one is faced with the challenge of structuring and integrating various orthogonal modalities (space-based vs. time-based) in an intuitive way for users. Continuous media such as audio and video, with time as inner structure, are difficult to access for lack of natural reference points, navigation is time consuming and can be confusing and summarisation is a non-trivial process. A study of users browsing and searching strategies when accessing voicemail messages, sometimes of very short duration (30s), showed that people had serious problems with local navigation of messages, and remembering messages' content [19,20]. Many users performed time-consuming sequential listening of messages in order to find relevant information and often reported taking notes to remember messages' content. In contrast, users displayed improved browsing performance, playing less audio when speech recognition transcripts were available as audio indexes in the user interface [21]. However, ASR currently suffers from a number of limitations. Disfluencies in spontaneous human-to-human dialogues, lack of word or sentence boundaries, poor recording conditions, crosstalk, inappropriate language models, out-of-vocabulary items and variations in speaking styles and pronunciations mean that for a certain percentage of people, some systems may have very low recognition rates [22]. In cases of meetings lasting an hour or longer, even if the transcripts were of good quality, they may still represent quite a voluminous amount of information to scan through. Also, as spoken language is significantly different to written language, the transcripts may be difficult to decipher (due to style, repetitions, false starts, etc). Text-based information retrieval techniques can be applied to the ASR transcripts and one could argue that a key-word search can be an appropriate way of overcoming these shortcomings to quickly find specific information. However, in some cases a person looking for a specific information

and the meeting participants who actually mentioned the relevant information during the meeting may use *different words* to refer to the *same object*. This would frustrate a key-word search approach and the user would need to reverse to reading the full transcripts.

More importantly, in the case where a meeting's outcome is a space-based artefact, visual browsing may be a more appropriate way of navigating the meeting recording. Consider the outcome of Fig. 1a. Due to the graphical nature of the document, visual scanning is almost instantaneous. In a collocated meeting scenario, one possible way of querying information from other participants would be to simply point at a particular item and say *"what about this?"* Ideally, one would want to have access to *all* segments of the recording where the item in question is *mentioned*. For the reasons discussed above, current ASR technology cannot guarantee that this will be done reliably. In contrast, the alternative proposed in this paper is based on a simple assumption, namely that by associating all space-based *data objects* with a log of all the actions which affected them during the course of the meeting, one can provide access to all segments of the recording during which a particular item was *manipulated*. We refer to this information mining paradigm as information retrieval from the *data object perspective*.

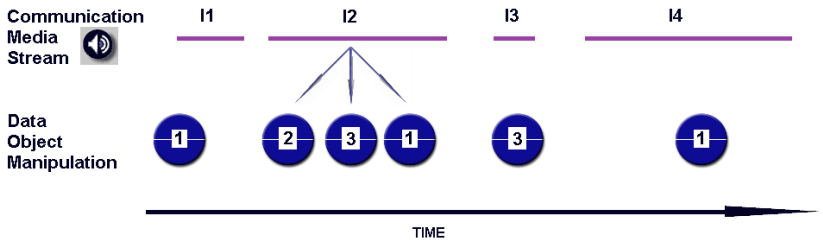


Fig. 2. The data stream or timeline perspective

Common approaches to building visualisation and retrieval interfaces for browsing multimodal meetings emphasise linear access (whether sequential or random) due to the structuring role time naturally plays in multimedia data. Segmentation and indexing according to some features of the time-based media (speaker transition, shot detection) are used to define a number of media intervals [23]. Access to specific media intervals is provided by some persistent representation of the time-based media (keyword, speaker identity, keyframes). By synchronised play-back of multiple media streams, a number of browsing systems [8,14,24] will ensure that space-based artefact manipulations concurrent

with the current media interval will be visible to the user. In Fig. 2[1] selecting the media interval I_2 will not only play the corresponding audio and video but will also display the nature of manipulations on the three objects: O_1, O_2 and O_3 which were modified within the interval duration.

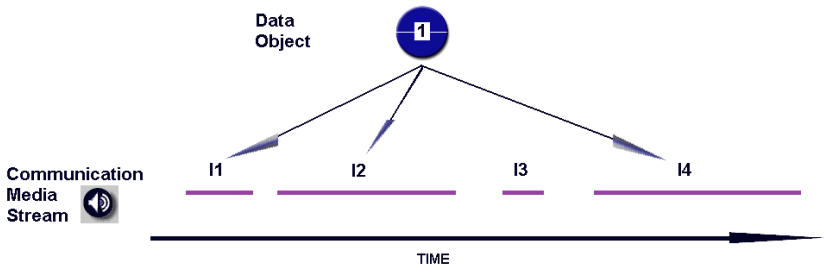


Fig. 3. Data Object perspective

The paradigm shift we propose involves looking at inter media relationship from the object perspective. By logging all (relevant) information relating to the manipulation of a specific artefact, one can link these with all concurrent time-base media segments. Thus, access to the time-based media is now done through selecting a specific *object*, or specific *actions* performed on an object. This is illustrated by Fig. 3: selecting object O_1 provides access to all time-based segments concurrent with a manipulation of O_1: the three related intervals I_1, I_2 and I_4. Our assumption is that in many cases, object manipulations will coincide with meeting participants focusing on the specific object. From an information retrieval perspective, this paradigm shift from time to object seems quite intuitive, shifting the emphasis from *"what were people doing when they were discussing this?"* to *"what were they saying when they did this?"*

3.3 Object Temporal Associations

One immediate property of having access to individual objects' history logs is that it enables us to discover potential associations between specific objects just by investigating the concurrency of actions performed on these objects. The following scenario illustrates how analysing temporal patterns of object manipulation uncover potential hidden information. The outcome of Fig. 1a is the result of several meetings, each concentrating on resolving a specific issue. The final plan is *flat*: relationships between the different objects are not a-priori obvious. However, by analysing temporal information for objects, we discover

[1] Artefact manipulations have been represented as punctual objects to emphasise their nature as separate abstract entities. In reality, artefact manipulations are time intervals themselves (duration of manipulations)

that in one of these meetings, the manipulation of the *"Table"* objects seems to be often concurrent to manipulations of the *"Exit"* item. Listening to audio segments where these objects are manipulated in close time proximity, the user discovers that in this particular project, the client's preferred table layout is not compatible with fire regulations. As a result, the client's *original* layout (no longer visible in the final outcome) had to be modified in order to accommodate the existing *"Exit"* and meet fire regulations, which explain the reasons behind the position of the *"Table"* objects in the *final* layout.

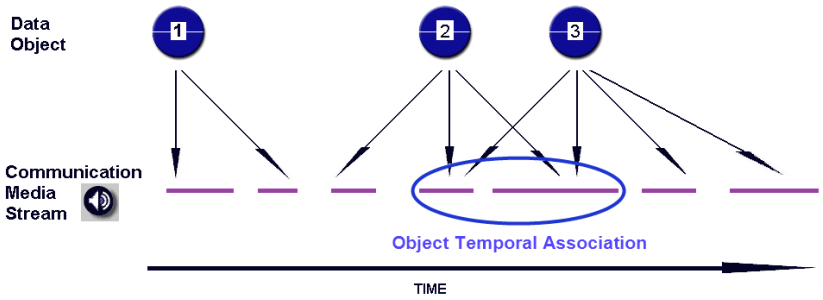


Fig. 4. Data Objects Temporal Associations

Fig. 4 illustrates object temporal associations: objects O_2 and O_3 were on several occasions manipulated in close time proximity and as a result share a number of concurrent time-based media segments in their respective history log, indicating a potentially useful information link between these two distinct objects. Object associations enable us to go beyond the information pattern illustrated in Fig. 3. Specific objects are not only linked to relevant segments of the time-based media, the wider context in which a specific object was manipulated during the meeting can be investigated within the context of other data object manipulations. We thus define the concept of an object's *temporal neighbourhood* as the set of (i) time-base media segments concurrent with the object manipulation and (ii) actions performed on other objects within the previous time-based segments' duration. We propose the following algorithm for retrieving an object temporal neighbourhood:

Definition 1. *Temporal neighbourhood retrieval:*

1. *retrieve the set of all space-based actions performed on a specific object*
2. *retrieve the set of all time-base segments concurrent with these actions*
3. *retrieve all actions performed on* different *objects which took place within the duration of the previous set of time-base segments*
4. *iterate through the 2 previous steps until no new actions or time-base intervals can be found*

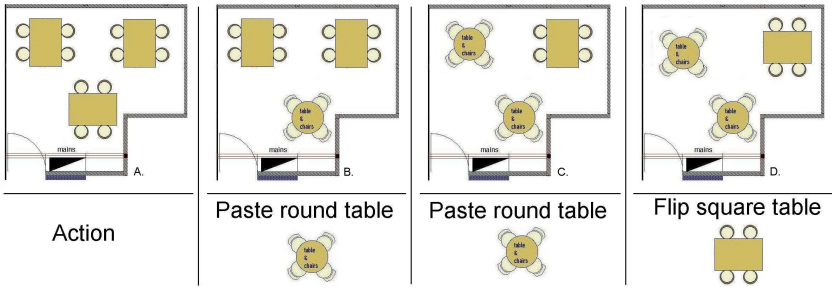


Fig. 5. Sequence of Actions

3.4 Action-based Browsing

We have so far defined space-based objects as potential information extraction and retrieval units. Another potential use of a log of space-based actions is as a navigation tool into the time-based media recording. Consider the simple sequence illustrated in Fig. 5: the upper part of the figure illustrates the evolution of the document during the course of the meeting while the bottom part shows the corresponding actions. One might wonder what prompted the choice of two different sets of table in the final outcome Fig. 5D. By visually observing the sequence of actions, a user may identify the exact moment when an action of *interest* was performed (i.e: pasting round table), thus identifying a region of the time-based media where a potential explanation is likely to be found, as illustrated in Fig. 6.

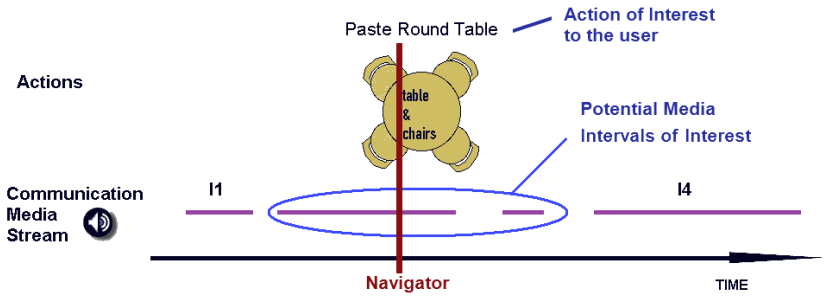


Fig. 6. Using space-based actions as a navigation tool

For this navigation method to be useful, only certain actions should be visible during browsing: for most applications, pixel or character-based operations are meaningless out of context. Operation filtering can be done at two stages. During the action logging stage, only actions which are potentially useful for information retrieval purposes are captured, while atomic operations are either discarded

or buffered (into a more comprehensive operation). As previously mentioned, a potentially useful action entirely depends on a particular application's scope. Information filtering can also be done at the post-meeting processing stage, where the user can dynamically choose what type of actions he is interested in (i.e: display only *"paste" specific "type" of object*). The definite appeal of such a navigation method is that space-based actions will generally have strong associated semantics and are appropriate for quick visual scanning, thus potentially offering a powerful indexing method into the time-based media. Interactions are discrete, generally sparse enough so as not to overload a user with information, and tend to form natural semantic clusters over time (when a specific topic is discussed) allowing for discrimination and segmentation of topics within a meeting recording. This indexing method is also perfectly accurate in timing and content as it is not subject to recognition errors.

3.5 Advanced Query Model

The object history model presented in this paper offers a large choice of granularities of retrieval. If a user does not exactly know what he is looking for, a general object neighbourhood retrieval, where all time-based media intervals and objects operations related to a specific object are retrieved may be appropriate, or, an action-based navigation, as detailed in the previous section. However, if the user is looking for a specific information, more advanced queries can be formulated by selecting a type of object with conjunction and disjunction of actions (primitive and manipulation) types and nature of actions attribute. An example of advanced query is: select all objects *round table*, where operation is *hatch* and attribute is *hatch pattern*. The general form of an advanced query is:

$$\text{retrieve } \textit{object type} \wedge \textit{action type} \wedge \textit{action attribute}.$$

4 Implementation: The Meeting Miner

The general space-based history model presented in this paper is the result of our experience in building an online collaborative writing environment with structured activity logging [25,26] and audio communication channel (Real Time Protocol multicast). In this environment, the chosen granularity of space-based artefact units for capturing operation logs are the paragraphs of text. These are self-contained information items with persisting histories when the segments are moved or altered. While paragraphs can be the subject of a number of physical manipulations like the ones previously described (cutting, pasting, moving, merging) the required number of primitives and manipulation rules applicable to paragraphs in a general usage scenario are limited, and therefore presented us with an ideal and manageable case study for the development of a space-based artefact history capture and management architecture. A detailed description of the timestamping model designed to manage paragraph history in case of modifications to document structure can be found in [27] and paragraph level

retrieval and browsing as well as preliminary evaluation results can be found in [3]. The Meeting Miner [28] user interface is shown in Fig. 7. The main user interface components are described below:

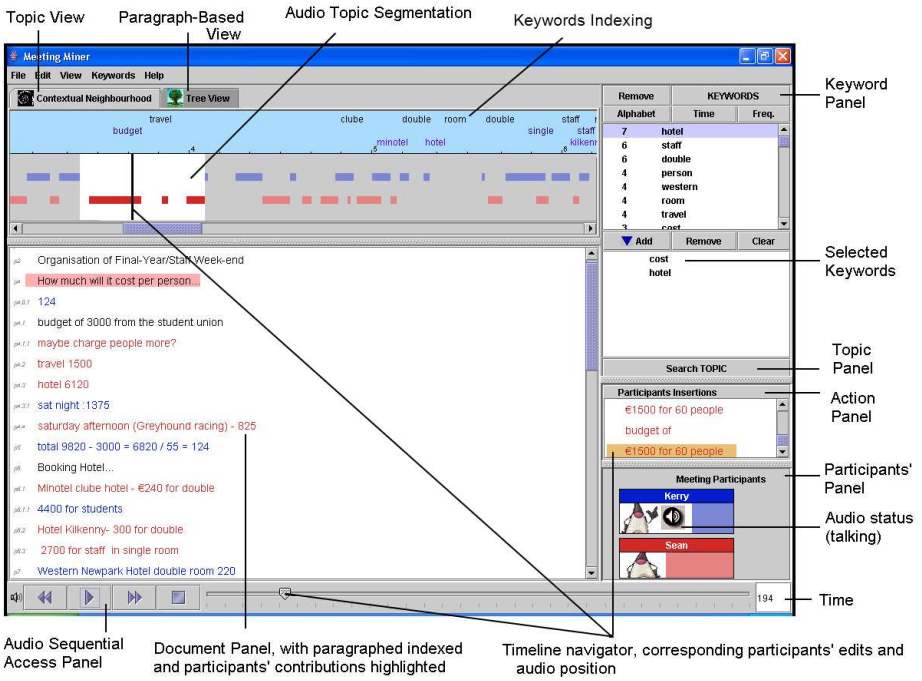


Fig. 7. MeetingMiner

Document View displays the final text document in the lower pane. Each paragraph in the document view is indexed for easy cross referencing. Each participants' individual contributions can be highlighted according to a colour code. When the paragraph-based view is selected, clicking on an individual paragraph will display the paragraph temporal neighbourhood.

Upper panel The upper pane can be one of two views, depending on how the user wishes to browse the meeting. In a paragraph-based retrieval, clicking on a specific paragraph will prompt the display of the tree-structured paragraph retrieval unit consisting of the content of editing nodes, and corresponding audio nodes, with the name of all the active participants within the duration of these temporal intervals (Fig. 8). An alternative view is the topic view, or *contextual neighbourhood* view. When this mode is selected, regions where audio contributions are likely to be related to the topic selection (a set of keywords selected in the topic panel) are highlighted. Clicking on a particular interval will play the corresponding audio.

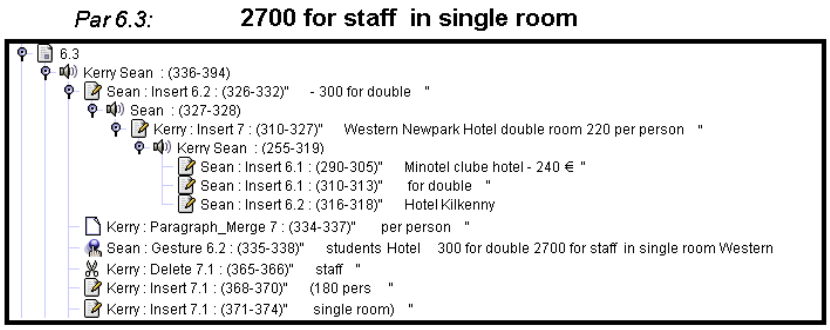


Fig. 8. A paragraph Temporal Neighbourhood

Keyword Indexer. Above the audio view showing speech according to speakers, it displays significant keywords in order to offer hints of audio content.

Keyword Panel. Displays all the potential keywords from the text document identified by the system. The list of keywords can be displayed in alphabetical order, frequency ranking or simply time of appearance. The user can dynamically update the list (removing words under a certain frequency or only select keywords associated with a certain type of action, etc.)

Topic Panel. The user can dynamically choose a set of keywords to create a specific topic. A subsequent topic search will highlight audio segments associated with these keywords. The audio intervals selected by the topic search are segments in the neighbourhood of participants' edits which contain the keywords.

Action Panel. Used in conjunction with the timeline navigator (slider) bar, it displays the nature of concurrent participants' edits for action-based browsing.

Participants' Panel. Displays the names of the participants. Each participant is assigned a unique colour code which highlights on the interface the ownership of the various text and audio contributions. A little audio icon is also displayed to show participants current activities (speaking, idle, etc).

Audio Panel Provides sequential and random access to the audio file. The browser's audio mode settings offers the user several navigation options such as skipping silences, or, if the topic mode is selected, jumping to the next topical segments. Similar functionalities were implemented in the SpeechSkimmer [29].

Timeline Navigator (slider) The navigator's purpose is twofold: first, it offers a reference point into the audio recording. It also offers random access to the audio file. While moving the slider, participants' concurrent actions are displayed in the Action Panel, so the user can decide to stop and listen to a specific section of the recording if he were to see an action of particular interest (as described in 3.4).

5 Conclusion and Future Work

Based on our experience with the development of an integrated online collaborative writing environment with structured activity logging and post-meeting browsing system, we have presented a generic time-based artefact history model and novel time-based data mining paradigms for mining information from artefacts producing meetings. Capturing and timestamping participants' space-based interactions with data objects offers new possibilities for meeting mining. Investigating temporal relationships between objects uncovers potential semantic links which are not necessary obvious when looking at a meeting's outcome. The definite appeal of such a navigation method is that space-based actions will generally have strong associated semantics and are appropriate for quick visual scanning, thus offering a powerful indexing method into the time-based media. Interactions are discrete, generally sparse enough so as not to overload a user with information, and tend to form natural semantic clusters over time (when a specific topic is discussed) allowing for discrimination and segmentation of topics within a meeting recording. This indexing method is also perfectly accurate in timing and content as it is not subject to recognition errors. Future work will integrate ASR technology within the temporal mining architecture presented. A full evaluation of the Meeting Miner as a meeting mining tool will be performed.

Acknowledgments

This work has been supported by Enterprise Ireland through a Basic Research Grant.

References

1. Tucker, S., Whittaker, S.: Accessing multimodal meeting data: Systems, problems and possibilities. In Samy Bengio, H.B., ed.: Machine Learning for Multimodal Interaction: First International Workshop, MLMI 2004. Volume 3361., Martigny, Switzerland, Springer-Verlag GmbH (2005) 1–11
2. Waibel, A., Bett, M., Metze, F., Ries, K., Schaaf, T., Schultz, T., Soltau, H., Yu, H., Zechner, K.: Advances in automatic meeting record creation and access. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing. (2001) 597–600
3. Bouamrane, M.M., Luz, S., Masoodian, M.: History based visual mining of semi-structured audio and text. In: Proceedings of Multimedia Modelling, MMM06 Beijing, China, IEEE Press (2006) 360–363
4. Smeaton, A.F.: Indexing, browsing, and searching of digital video and digital audio information. LNCS Lectures on information retrieval (2001) 93–110
5. Waibel, A., Bett, M., Finke, M., Stiefelhagen, R.: Meeting browser: Tracking and summarizing meetings. In Penrose, D.E.M., ed.: Proceedings of the Broadcast News Transcription and Understanding Workshop, Lansdowne, Virginia, Morgan Kaufmann (1998) 281–286

6. Whittaker, S., Hirschberg, J., Choi, J., Hindle, D., Pereira, F., Singhal, A.: Scan: designing and evaluating user interfaces to support retrieval from speech archives. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '99, New York, NY, US, ACM Press (1999) 26–33
7. Whittaker, S., Hirschberg, J., Amento, B., Stark, L., Bacchiani, M., Isenhour, P., Stead, L., Zamchick, G., Rosenberg, A.: Scanmail: a voicemail interface that makes speech browsable, readable and searchable. In: Proceedings of the SIGCHI conference on Human factors in computing systems,CHI '02, New York, NY, US, ACM Press (2002) 275–282
8. Geyer, W., Richter, H., Abowd, G.D.: Making multimedia meeting records more meaningful. In: Proceedings of International Conference on Multimedia and Expo, ICME '03. Volume 2. (2003) 669–672
9. Geyer, W., Richter, H., Fuchs, L., Frauenhofer, T., Daijavad, S., Poltrock, S.: A team collaboration space supporting capture and access of virtual meetings. In: Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work, GROUP '01, New York, NY, US, ACM Press (2001) 188–196
10. Richter, H.A., Abowd, G.D., Geyer, W., Fuchs, L., Daijavad, S., Poltrock, S.E.: Integrating meeting capture within a collaborative team environment. In: Proceedings of the 3rd international conference on Ubiquitous Computing, UbiComp '01, London, UK, Springer-Verlag (2001) 123–138
11. Masoodian, M., Luz, S.: Comap: A content mapper for audio-mediated collaborative writing. In Smith, M.J., Savendy, G., Harris, D., Koubek, R.J., eds.: USbility Evaluation and Interface Design. Volume 1., Lawrence Erlbaum (2001) 208–212
12. Luz, S., Masoodian, M.: A model for meeting content storage and retrieval. In: Proceedings of the 11th International Multimedia Modelling Conference, MMM'05, IEEE Press (2005) 392–398
13. Luz, S.: Interleave factor and multimedia information visualisation. In Sharp, H., Chalk, P., LePeuple, J., Rosbottom, J., eds.: Proceedings of Human Computer Interaction 2002. Volume 2., London (2002) 142–146
14. Wellner, P., Flynn, M., Guillemot, M.: Browsing recorded meetings with ferret. In Bengio, S., Bourlard, H., eds.: Proceedings of Machine Learning for Multimodal Interaction: First International Workshop, MLMI 2004. Volume 3361., Martigny, Switzerland, Springer-Verlag GmbH (2004) 12–21
15. Bouamrane, M.M., Luz, S.: Meeting browsing, state of the art review. to appear in User-Centered MultiMedia, special issue of Multimedia Systems Journal, Susan Boll and Gerd Utz Westermann eds., Springer (summer 2006)
16. Ellis, C.A., Gibbs, S.J., Rein, G.: Groupware: some issues and experiences. Communications of the ACM **34**(1) (1991) 39–58
17. Sun, C., Jia, X., Zhang, Y., Yang, Y., Chen, D.: Achieving convergence, causlity preservation, and intention preservation in real-time cooperative editing systems. ACM Transactions on Computer-Human Interaction **5**(1) (1998) 63–108
18. Sun, C., Chen, D.: Consistency maintenance in real-time collaborative graphics editing systems. ACM Transactions on Computer-Human Interaction **9**(1) (2002) 1–41
19. Whittaker, S., Hirschberg, J., Nakatani, C.H.: All talk and all action: strategies for managing voicemail messages. In: conference summary on Human factors in computing systems, CHI 98, New York, NY, USA, ACM Press (1998) 249–250
20. Whittaker, S., Hirschberg, J., Nakatani, C.H.: Play it again: a study of the factors underlying speech browsing behavior. In: CHI '98: conference summary on Human factors in computing systems, New York, NY, USA, ACM Press (1998) 247–248

21. Hirschberg, J., Whittaker, S., Hindle, D., Pereira, F., Singhal, A.: Finding information in audio: A new paradigm for audio browsing and retrieval. In Mani, I., Maybury, M.T., eds.: Proceedings of the ESCA workshop: Accessing information in spoken audio, Cambridge (1999) 117–122
22. Furui, S.: Automatic speech recognition and its application to information extraction. In: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, Morristown, NJ, US, Association for Computational Linguistics (1999) 11–20
23. Allen, J.F.: Maintaining knowledge about temporal intervals. Communications of the ACM **11**(26) (1983) 832–843
24. Brotherton, J.A., Bhalodia, J.R., Abowd, G.D.: Automated capture, integration, and visualization of multiple media streams. In: Proceedings of the IEEE International Conference on Multimedia Computing and Systems, ICMCS '98, Washington, DC, US, IEEE Computer Society (1998) 54
25. Bouamrane, M.M., King, D., Luz, S., Masoodian, M.: A framework for collaborative writing with recording and post-meeting retrieval capabilities. IEEE Distributed Systems Online (2004) Special issue on the 6th International Workshop on Collaborative Editing Systems.
26. Masoodian, M., Luz, S., Bouamrane, M.M., King, D.: Recoled: A group-aware collaborative text editor for capturing document history. In: Proceedings of WWW/Internet 2005. Volume 1., Lisbon (2005) 323–330
27. Bouamrane, M.M., Luz, S., Masoodian, M., King, D.: Supporting remote collaboration through structured activity logging. In Hai Zhuge, G.C.F., ed.: 4th International Conference in Grid and Cooperative Computing, GCC 2005. Volume 3795 / 2005., Beijing, China, Springer-Verlag GmbH (2005) 1096–1107
28. Bouamrane, M.M., Luz, S.: Navigating multimodal meeting recordings with the meeting miner. In et al., H.L., ed.: Proceedings of Flexible Query Answering Systems, FQAS'2006. Volume 4027/2006., Milan, Italy, Springer-Verlag (2006) 356–367
29. Arons, B.: Speechskimmer: a system for interactively skimming recorded speech. In: ACM Transactions on Computer-Human Interaction. Volume 4,1., New York, NY, US, ACM Press (1997) 3–38

Improving Organizational Efficiency by Combining Tier Analysis and Clustering Method

Sung Ho Ha[1], Han Kook Hong[2]

[1] School of Business Administration, Kyungpook National University, 702-701 Sangyeok-dong, Buk-gu, Daegu, Korea
hsh@mail.knu.ac.kr
[2] Management Information System, Dong-Eui University, 614-714 Kaya-Dong, Pusanjin-ku, Pusan, Korea

Abstract. In this paper, we propose an Intelligent Data Envelopment Analysis (IDEA) system that utilizes a hybrid methodology combining the Tier Analysis with the machine learning technology. We aim to show that the IDEA system can be used to evaluate the intra-organizational efficiency in the system integration projects and the inter-organizational efficiency in the life insurance companies. The application is unfolded in two phases. In the first phase, DEA is repetitively used to evaluate the efficiency of DMUs and cluster them together according to their efficiency level (Tier Analysis). In the second phase, the IDEA system utilizes self-organizing map to group similar DMUs, selects efficient DMUs within a reference set (benchmarking target), and provides the guidelines on the stepwise enhancements for the inefficient ones.

1 Introduction

The DEA has been introduced in operational research [8] and economic literatures [13] as a method for assessing the efficiency of activity units. It is a linear programming based method that has been used extensively for assessing the relative efficiency of activity units of non-profit (e.g. education [6][16], courts [18], hospitals [7][9]) and for-profit (e.g. banks [19][20][23], hotel [10], restaurants [1], public houses [2], corporate performance [21]) organizations. The full technical details of DEA will not be discussed here, but reviews can be found in Boussofiance et al., [4] and Fried et al., [15].

As the earlier list of applications suggests, DEA can be a powerful tool used widely. But, despite of its extensive applications and merits, some features of DEA remain bothersome. So, we present an intelligent DEA system that utilizes a hybrid methodology combining the conventional DEA with the machine learning technology in order to complement drawbacks of the conventional DEA. The application is divided into two phases.

In the first phase, the IDEA system applies DEA to evaluate the efficiency of DMUs with their multidimensional inputs and outputs. After that, the system clusters the DMUs together through the Tier Analysis, which applies the DEA again to the

remaining inefficient DMUs. Then the system generates several classification rules by using a decision tree classifier, which identifies and describes the characteristics of each tier. Those rules are used to classify new DMUs into each tier without disrupting the relative efficiency structure of existing DMUs. The system predicts the efficiency level of a new DMU by applying the production rules obtained.

To verify the usefulness of the first phase in the proposed system, we apply the system to evaluate the efficiency of SI (System Integration) projects, which were performed by one of the biggest SI companies in Korea. The companies have suffered competitive environment intensively. As a natural consequence, they have re-engineered their inefficient and out-dated management styles in order to survive in such a rough condition. To make the business profitable and to improve the business performance, it is inevitable that the efficiency of projects should be evaluated objectively (the intra-organizational efficiency). Nowadays, it becomes more visible trends that customers, including government and client companies, gradually want to verify the project-performing capabilities of SI companies in advance. Besides, overseas information technology companies have aggressively tried to enter the domestic market. In the age of globalization and high competition, it is imperative that domestic SI companies need to introduce the performance evaluation models of SI projects, including Capability Maturity Model and Software Process Improvement and Capability Determination, to gain a competitive advantage. Therefore, it makes our research regarding evaluation of SI projects very opportune.

In the second phase, the IDEA system can be used to derive the stepwise strategies improving the efficiency of a DMU and can be helpful in finding, so-called, the improvement path for any inefficient DMU. The conventional DEA offers no guidelines about the efficiency improvement, since a reference set for inefficient DMUs just contains several efficient ones. Hence, the system utilizes a technique for dividing DMUs into similar segments. The basic idea is that DMUs within the same segment share common domain-specific knowledge and, therefore, it is easier for a less inefficient DMU to become more efficient if it tries to follow the management strategy or operation of more efficient ones in the same segment. With the tiers identified by the Tier Analysis, the segmentation knowledge is used to find improvement paths for inefficient life insurance companies.

To verify the usefulness of the second phase of the proposed system, we apply the system to evaluate the inter-organizational efficiencies of 29 life insurance companies in Korea. The market for life insurance has become saturated. Participation of foreign life insurance companies into Korean market has made the management environment worse. In fact, small life insurance companies became bankrupt during last couple of years. Therefore, in order to survive in such a highly competitive market, they are eagerly pursuing the productivity improvement in the management and the management strategies, which result in improving the efficiency of operation and gaining a competitive advantage. In doing so, life insurance companies need an appropriate tool to precisely measure their operational efficiencies. Based on these measurements, they set up their improvement strategies to make themselves more efficient.

2 Data Envelopment Analysis (DEA)

DEA was developed by Charnes et al. as a generalization of the framework of Farrell [14] on the measurement of productive efficiency. DEA, as a non-parametric approach, evaluates relative efficiency of inputs and outputs and determines a set of Pareto-efficient DMUs with an objective of calculating a discrete piecewise frontier. Details of the methodology as well as description of DEA can be found in Charnes et al. [8].

Several characteristics that make DEA powerful are as follows: First, it can handle simultaneously multiple inputs and multiple outputs of a DMU. Second, it does not require an assumption of a functional form relating inputs to outputs. Third, DMUs are directly compared against a peer or combination of peers and it provides managers with a procedure to differentiate between efficient and inefficient DMUs. Fourth, it pinpoints the sources and the amount of deficiency for each of the inefficient DMUs. Fifth, it can be used to detect specific inefficiencies that may not be detectable through other techniques such as linear regression or ratio analyses. Finally, inputs and outputs can have different units of measurement.

Despite of its extensive applications and merits, some features of DEA remain bothersome. First, through DEA is good at estimating 'relative' efficiency of a DMU, it only tells us how well we are doing compared with our peers but not compared with a 'theoretical maximum'. Thus, in order to measure efficiency of a new DMU, we have to entirely develop new DEA with the data of previously used DMUs. We cannot predict the efficiency level of the new DMU without another DEA analysis. Second, for DMUs directly compared with a peer or combination of peers, DEA offers no guidelines where relatively inefficient DMUs improve. Finally, it does not provide stepwise paths for improving the efficiency of each inefficient DMU.

3 Intelligent DEA System

In this section, we present an IDEA system that utilizes a hybrid methodology combining the conventional DEA and the machine learning technology in order to complement drawbacks of the conventional DEA.

3.1 Phase I – Tier Analysis and Classification Rules for Each Tier

The IDEA system uses DEA to evaluate the efficiency of DMUs. DEA determines the most productive group of the DMUs and the less-productive group. That is, the DMUs are clustered into an efficient group or an inefficient one by DEA. A similar approach for clustering DMUs by DEA was presented by Thanassoulis [22]. However, the clusters on that study were made by the characteristics of the input resource mix not by their efficiency levels. Tier Analysis here is a kind of technique that can be used to cluster DMUs according to their efficiency levels.

In the first application of DEA, the IDEA system obtains the efficiency scores of entire DMUs. The results reveal the most efficient group by indicating their scores are

equal to 1. We call this group 'Tier 1'. Then, the IDEA system proceed DEA again only with the inefficient DMUs which are not on Tier 1. DMUs whose efficiency scores in the second application are equal to 1 are set Tier 2. The same procedure is repeated while the number of remaining inefficient DMUs is at least three times multiple of that of input plus output variables, as Banker and Kemerer [3] have proposed. We call this procedure 'Tier Analysis'. The IDEA system divides DMUs into several tiers by applying the Tier Analysis (refer to Fig. 1).

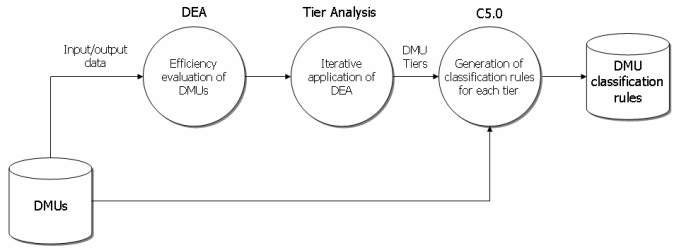


Fig. 1. A procedure of Tier Analysis and generating classification rules for each tier.

After that, by using the C5.0 with the DMU tiers, the system generates rules for classifying new DMUs into each tier. These rules can determine the input variables, the output variables, and their value ranges that discriminate each tier best.

3.2 Phase II – Finding the Stepwise Improvement Paths

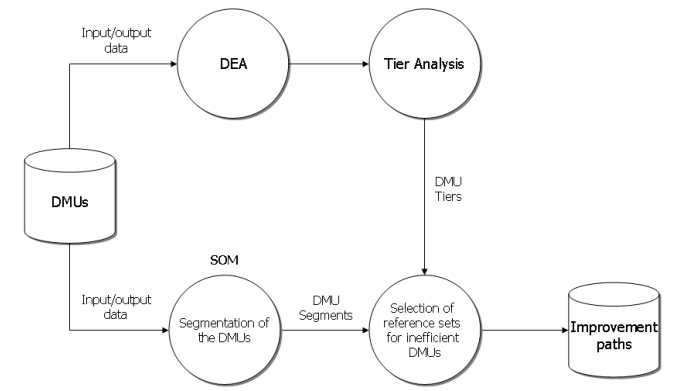


Fig. 2. A procedure of finding stepwise improvement paths.

In the second phase, we determine the stepwise path for improving the efficiency of DMUs except the most efficient DMUs on the Tier 1. In doing so, the set of DMUs used in the first phase is clustered into a number of segments by using SOM. With the DMU segments by SOM and the DMU tiers by DEA, a set of benchmarking target

DMUs are determined. We call this set 'Improvement Path', which any inefficient DMUs can follow in order to improve their efficiency (refer to Fig. 2).

4 Application Results

In order to verify the usefulness of our IDEA system, we apply it to two different cases: the evaluation of efficiency of SI projects (Intra-organizational efficiency) and the evaluation of the efficiency of 29 life insurance companies (Inter-organizational efficiency).

4.1 Case I: The SI Projects

SI projects perform all the activities that are necessary to build and maintain various kinds of information systems in response to their client needs. SI companies mainly carry out their works on project basis. Upon receiving the project request from the client, the SI company organizes a team for the project. The quality of the company is determined by the efficiency of projects. Precise evaluation of projects, therefore, becomes an important issue for the SI companies.

SI companies are interested in assessing how well a project uses its resources to obtain a desired outcome. The companies are also interested in defining the main resources (inputs) and the relevant products (outputs) of the process, and in finding appropriate measures for their efficiency (an efficiency measurement framework). Measuring the project, however, has not been easy. Most researchers and practitioners have had different point of view on what to measure and how to measure it.

Deephouse et al. [12] used software quality, meeting targets, and rework after delivery as output variables of software development projects. Software quality covers the extent to which the software system meets the diverse needs of the intended users (customer satisfaction index). Meeting targets centers around the fact that, to be successful, a project should be on time and within budget. Rework is observed on many SI projects and it may occur as a result of poor understanding of user requirements or poor technical design. Banker and Kemerer [3] used budget performance, schedule performance, user satisfaction, and maintenance complexity as output variables of information system development projects. They used either labor hours or labor cost as main input variables.

In this paper, we propose a project management model with four input and four output variables, as shown in Table 1.

Table 1. Input and output variables for SI project management.

	Variable	Measurement
Input	Labor hours A (*La*)	Amount of total man-months, who have a career over 10 years
	Labor hours B (*Lb*)	Amount of total man-months, who have a career between 6 and 10

		years
	Labor hours C (*Lc*)	Amount of total man-months, who have a career below 5 years
	Material and equipment resources (*MER*)	Total monetary amount of hard-ware, software tools, and other materials
Output	Customer satisfaction index (*CSI*)	Customer questionnaire
	Schedule performance (*SP*)	Ratio of planned project period to actual period
	Budget performance (*BP*)	Difference between actual devel-opment cost and planned budget
	Rework hours after delivery (*RAD*)	Amount of total man-months for rework or additional service

DMUs that we gathered are 50 SI projects, which were carried out during 1997-2004. The IDEA system uses the Charnes-Cooper-Rhodes (CCR) ratio model of DEA. In our application of 50 projects, the fourth tier is the last one derived by the Tier Analysis. In the Tier Analysis, what is important is which tier each project belongs to. Results of the Tier Analysis are summarized in Fig. 3. 13 DMUs belong to Tier 1, 10 DMUs to Tier 2, 11 DMUs to Tier 3, and 16 DMUs to Tier 4.

Tier	DMU (SI project)	Reference set
1	$P_1$2 $P_1$7 $P_1$8 $P_1$14 $P_1$16 $P_1$21 $P_1$29 $P_1$32 $P_1$40 $P_1$41 $P_1$45 $P_1$46 $P_1$47	No reference set

Tier	DMU (SI project)	Reference set in Tier 1
2	$P_2$3	$P_1$40 $P_1$41 $P_1$45
	$P_2$5	$P_1$16 $P_1$41 $P_1$45
	$P_2$6	$P_1$21 $P_1$29 $P_1$40
	$P_2$24	$P_1$7 $P_1$14 $P_1$16 $P_1$29
	$P_2$28	$P_1$7 $P_1$16 $P_1$29 $P_1$41
	$P_2$37	$P_1$8 $P_1$21 $P_1$32 $P_1$40
	$P_2$38	$P_1$21 $P_1$46
	$P_2$39	$P_1$8 $P_1$16 $P_1$46
	$P_2$43	$P_1$21 $P_1$45
	$P_2$48	$P_1$7 $P_1$32

Tier	DMU (SI project)	Reference set in Tier 2
3	$P_3$1	$P_2$6 $P_2$24
	$P_3$9	$P_2$5 $P_2$6 $P_2$37
	$P_3$17	$P_2$3 $P_2$6 $P_2$37 $P_2$43
	$P_3$18	$P_2$28 $P_2$37 $P_2$48
	$P_3$19	$P_2$6
	$P_3$20	$P_2$5 $P_2$24 $P_2$37
	$P_3$25	$P_2$5 $P_2$6 $P_2$37
	$P_3$26	$P_2$28 $P_2$37 $P_2$48
	$P_3$30	$P_2$6
	$P_3$36	$P_2$5 $P_2$6 $P_2$37 $P_2$43
	$P_3$50	$P_2$5 $P_2$6 $P_2$37

Tier	DMU (SI project)	Reference set in Tier 3
4	$P_4$4	$P_3$1 $P_3$18
	$P_4$10	$P_3$17 $P_3$50
	$P_4$11	$P_3$9 $P_3$20 $P_3$50
	$P_4$12	$P_3$1 $P_3$18 $P_3$25 $P_3$36
	$P_4$13	$P_3$1 $P_3$9 $P_3$30 $P_3$36 $P_3$50
	$P_4$15	$P_3$25 $P_3$36 $P_3$50
	$P_4$22	$P_3$20 $P_3$25 $P_3$50
	$P_4$23	$P_3$1 $P_3$20 $P_3$25
	$P_4$27	$P_3$9 $P_3$30 $P_3$36 $P_3$50
	$P_4$31	$P_3$1 $P_3$50
	$P_4$33	$P_3$25 $P_3$50
	$P_4$34	$P_3$9 $P_3$25 $P_3$30 $P_3$36
	$P_4$35	$P_3$1 $P_3$18
	$P_4$42	$P_3$1 $P_3$25 $P_3$50
	$P_4$44	$P_3$1 $P_3$25 $P_3$50
	$P_4$49	$P_3$1 $P_3$18 $P_3$20 $P_3$25 $P_3$50

Fig. 3. Clustering of SI projects by the Tier Analysis.

Classification. A classification using C5.0 starts from preparing training set of cases, each described in terms of the eight attributes (the input and output factors) and a known class (a tier number).

The induction process of C5.0 attempts to classify a case, expressed as a function of the attributes, which explains the training cases and may also be used to classify unseen cases. The 50 cases are prepared. As moving from left to right, eight factor values and a tier number are arranged in rows in Table 2.

Table 2. Sample cases for training C5.0.

Factors / DMUs	*La*	*Lb*	*Lc*	*MER*	*CSI*	*SP*	*BP*	*RAD*	Tier
$P_3$1	2.0	7.0	7.0	14.5	90.8	0.96	8.1	4.11	3
$P_1$2	1.1	3.3	2.0	4.6	89.5	1.2	4.1	4.87	1
$P_2$3	1.0	2.0	4.0	5.8	86.6	0.99	2.7	4.91	2
$P_4$4	10.6	31.4	16.4	30.9	79.6	0.84	12.1	3.29	4
...	...	...	...	...	...	...	...	...	

Decision trees produce comprehensible classification rules and discover major input and output variables most affecting the efficiency of DMUs. The IDEA system found such order of influence as *La*, *Lb*, *SP*, *RAD*, *MER*, *Lc*, *BP*, and *CSI*. The labor resource A (La) has the greatest effect on the efficiency of DMUs.

4.2 Case II: The Life Insurance Companies

In general, a method of analyzing productivity of a life insurance company is to represent the relationship of inputs and outputs to be a generalized Leontief profit function and to estimate parameters of the function [25]. However, the life insurance industry has such an uncertain management environment as inaccuracy of price information on inputs and outputs, unbalance of the amount of inputs and outputs due to monopoly or duo-poly, the exit from or entry into the industry, and government regulations on insurance rate. These limitations prevent the parametric method, which needs strict assumptions on a population, from being used.

Several researches have been made to measure the efficiency of life insurance companies by using DEA [5][11][24]. However the difficulty of those efficiency studies lies in measuring the productivity of the insurance industry. As Hornstein and Prescott [17] explain, there is not even a conceptual definition of the output to guide the construction of a reasonable measure of its product. Without it, it is not clear what data should be collected and how they should be used to compute an output measure. Therefore two alternatives are often suggested: on one hand, premiums or incurred losses, and on the other hand, the number of policies contracted appropriately. In

recent papers, losses and financial investments, and premiums earned are used as a proxy for nominal output.

In this paper, we propose an evaluation model of life insurance companies with four input and two output variables, as shown in Table 3.

Table 3. Input and output variables for life insurance industry evaluation.

Variable		Measurement
Input	Net operating expenses (NOE)	Subtracting income expenses from such expenses as labor wages, general administration, welfare, and salesman recruiting expenses
	Number of office workers (NOW)	The number of persons who manage sales persons and staffs in the head office
	Number of sales persons (NSP)	The number of persons who do a business with customers directly
	Number of branch offices (NBO)	The number of branch offices geographically dispersed
Output	Reciprocal of Loss Rates (LR)	The ratio of premium receipts to claims paid
	Working assets (WA)	Sources of property investment (cash, deposits, trust, securities, and real estate)

Note that we do not include the number of insurance contracts as an evaluation factor. Because domestic companies sell various types of life insurance products and their prices are different among them, considering the number of insurance contracts could introduce uncertainty in measurement.

DMUs used in the second case are the 29 life insurance companies in Korea. The IDEA system divides 29 companies into four different tiers according to their efficiency levels. Tier 1 contains four DMUs (C_13, C_15, C_117, C_125), Tier 2 has six DMUs (C_21, C_212, C_213, C_227, C_228, C_229), Tier 3 consists of ten DMUs (C_32, C_34, C_37, C_315, C_316, C_318, C_319, C_322, C_324, C_326), and Tier 4 comprises nine DMUs (C_46, C_48, C_49, C_410, C_411, C_414, C_420, C_421, C_423).

Segmentation. SOM utilized for clustering DMUs has six input variables and three by three output nodes. It has 29 DMUs as a training set. Choosing nine output nodes is appropriate since it is manageable for a manager to handle with. Training was performed during 20,000 epochs and terminated when the change of weight was less than a pre-specified threshold (0.01).

Table 4 shows the results of segmentation and summarizes the characteristics of each segment. Four segments came out. Segment 1 has one member company (C_13), segment 2 has seven companies (C_32, C_34, C_46, C_48, C_49, C_212, C_414), segment 3 contains 2 companies (C_21, C_15), and segment 4 contains 19 companies (C_37, C_410, C_411, C_213, C_315, C_316, C_117, C_318, C_319, C_420, C_421, C_322, C_423, C_324, C_125, C_326, C_227, C_228, C_229).

Table 4. Characteristics of each DMU segment.

Segment	NOE (Avg)	NOW (Avg)	NSP (Avg)	NBO (Avg)	LR (Avg)	WA (Avg)
1	1,411,258	7,912	58,415	1,711	1.07	33,016,684
2	113,017	1,554	8,197	374	0.762	1,769,286
3	758,874	6,769	53,760	1,604	1.02	16,180,759
4	28,436	396	1,656	82	1.35	300,244

Improvement Path. DMUs on the lower tiers can find the way for improving their efficiency by finding and following the reference DMUs on the upper tiers, which reside in the same segment.

For example, $C_3$24 on the Tier 3 has a reference set that consists of $C_2$12, $C_2$13, and $C_2$27 on the upper efficient frontier 2 (Tier 2). Among them, the IDEA system chooses $C_2$13 as a benchmarking target, since it belongs to the same segment with $C_3$24.

Based on the results from the Tier Analysis and SOM, the IDEA system can at last identify the stepwise improvement path for each DMU on each tier (except Tier 1). For example, the system found an improvement path for $C_4$10 like $C_4$10 → $C_3$24 → $C_2$13 → $C_1$17.

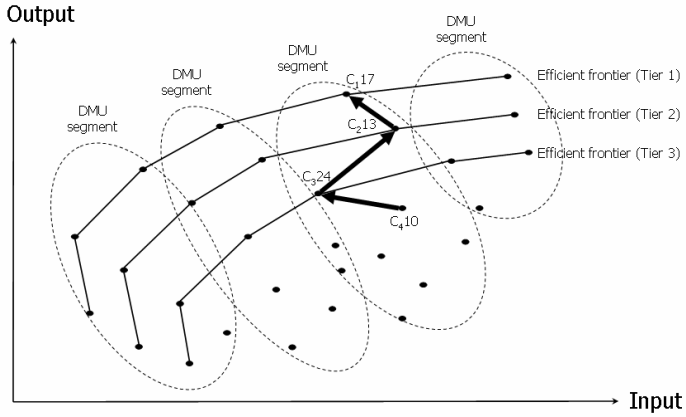


Fig. 4. An improvement path for a DMU $C_4$10 on the Tier 4.

As shown in Fig. 4, $C_3$24 is the first benchmarking target on the improvement path toward $C_1$17. Management of $C_4$10 could target $C_1$17 from the very beginning. However, because there is a resource limitation on running the business, the strategy pursing a stepwise improvement is plausible.

According to Table 5, the company, $C_3$24, consumes less net operating expenses (NOE) and operates less number of branch offices (NBO) than $C_4$10. Although $C_2$13, as the second improvement target, spends a similar level of input resources with $C_3$24, the level of working assets (WA) is much higher than that of $C_3$24. At last, $C_1$17 generally spends fewer inputs, especially the number of office workers (NOW), than $C_2$13, but it generates much more outputs, especially in the reciprocal of loss rate (LR). This

means that our intelligent DEA system can suggest more important input and output variables to management who considers improving the efficiency of his or her company.

Table 5. Characteristics of the target DMUs on the improvement path for $C_4$10.

Company	Tier	Input factors				Output factors	
		NOE	NOW	NSP	NBO	LR	WA
$C_1$17	1	26,909	259	1,723	98	0.686	339,621
$C_2$13	2	26,521	401	1,892	101	0.450	349,576
$C_3$24	3	25,761	389	1,672	97	0.449	251,910
$C_4$10	4	32,265	394	1,571	131	0.456	253,137

5 Conclusion and Discussions

In conventional DEA, it simply identifies inefficiencies, identifies comparable efficient units, and locates slack resources. But, the IDEA system we proposed provides more information about discriminant descriptors among input and output variables, which affects the efficiency of DMUs, and about rules for classifying new DMUs, and about stepwise improvement paths.

The IDEA system utilized a hybrid methodology combining the conventional DEA with the machine learning technology. It was unfold in two phases. To verify the usefulness of the first phase of our proposed methodology, the IDEA system applied the methodology to evaluating 50 SI projects, which were performed by one of the biggest SI companies in Korea. After Tier Analysis which clustered 50 projects into four tiers, the IDEA system generated classification rules by using a decision tree classifier in order to classify any new SI project without perturbing existing relative efficiency structure.

To verify the usefulness of the second phase, the IDEA system applied the methodology to evaluating the efficiencies of 29 life insurance companies in Korea. The conventional DEA cannot provide any guidelines about efficiency improvement to relatively inefficient companies. The IDEA system, however, can choose benchmarking targets for each inefficient company from the reference set. The system can provide information about stepwise improvement path by using SOM as a segmenting tool.

However, the present research has limitations. They can be also the topics for further researches. Environmental factors, including project complexity, the quality of available hardware and software tools, may also affect the efficiency of the SI projects. Unfortunately, due to the unavailability of data, these variables could not be included in this research. Future IDEA system may incorporate exogenous, uncontrollable variables or categorical variables into the production model.

Current practice of management evaluation on life insurance companies in Korea have focused on their capability of growth, productivity, profitability, and soundness and publicity. Therefore an intelligent DEA model including qualitative as well as quantitative data is needed to measure the efficiency of DMUs more accurately.

References

1. Athanassopoulos, A., Thanassoulis, E.: Performance improvement decision aid system in retail organizations using data envelopment analysis. Journal of Productivity Analysis 6 (1995) 153-170.
2. Athanassopoulos, A., Thanassoulis, E.: Separating market efficiency from profitability and its implications for planning. Journal of operational research society (1995) 46 30-45.
3. Banker, R.D., Kemerer, C.F.: Performance Evaluation Metrics for Information Systems Development: A Principal-Agent Model. Information Systems Research 3 (1992) 379-398.
4. Boussofiane, A., Dyson, R., Thanassoulis, E.: Applied data envelopment analysis. European journal of operational research 51 (1991) 1-15.
5. Brockett, P.L., Cooper, W.W., Golden, L.L., Rousseau, J.J., Wang, Y.: DEA evaluations of the efficiency of organizational forms and distribution systems in the US property and liability insurance industry. International Journal of Systems Science 29 (1998) 1235-1247.
6. Brockett, P.L., Cooper, W.W., Lasdon, L., Parker, B.R.: A note extending Grosskopf, Hayes, Taylor and Weber, "Anticipating the consequences of school reform: A new use of DEA". Socio-Economic Planning Sciences 39(4) (2005) 351-359.
7. Butler, T.W., Li, L.: The utility of returns to scale in DEA programming: An analysis of Michigan rural hospitals. European Journal of Operational Research 161(2) (2005) 469-477.
8. Charnes, A., Cooper, W.W., Rhodes, E.: Measuring the efficiency of decision making units. European Journal of Operational Research 2 (1978) 429-444.
9. Chen, A., Hwang, Y., Shao, B.: Measurement and sources of overall and input inefficiencies: Evidences and implications in hospital service. European Journal of Operation Research 161(2) (2005) 447-468.
10. Chiang, W.E., Tsai, M.H., Wang, L.S.M.: A DEA evaluation of Taipei hotels. Annals of tourism research 31(3) (2004) 712-715.
11. Cummins, J.D., Weiss, M.A., Zi, H.: Organizational form and efficiency: the coexistence of stock and mutual property-liability insurers. Management Science 45 (1999) 1254-1269.
12. Deephouse, C., Mukhopadhyay, T., Goldenson, D.R., Kellner, M.I.: Software Processes and Project Performance. Journal of Management Information Systems 12 (1996) 187-205.
13. Fare, R., Grosskopf, S., Lovell, K.: The Measurement of Efficiency of Production. Kluwer Nijhoff, Boston (1985).
14. Farrell, M.J.: The measurement of productivity efficiency. Journal of the Royal Statistical Society Series A 120(3) (1957) 253-281.
15. Fried, H., Lovell, K., Schmidt, S.: The Measurement of Productive Efficiency: Techniques and Applications. Oxford University Press, New York (1993).
16. Grosskopf, S., Hayes, K.J., Taylor, L.L., Weber, W.L.: Anticipating the consequences of school reform: A new use of DEA. Management Science 45 (1999) 608-620.

17. Hornstein, A., Prescott, E.C.: Measuring of the insurance sector output. The Geneva Papers on Risk and Insurance 59 (1991) 191-206.
18. Lewin, A.Y., Morey, R., Cook, T.: Evaluating the administrative efficiency of courts. OMEGA 10 (1982) 404-411.
19. Manandhar, R., Tang, J.C.S.: An empirical study on the evaluation of bank branch performance using data envelopment analysis. International Journal of Services Technology and Management 5(2004) 111-139.
20. Paradi, J.C., Schaffnit, C.: Commercial branch performance evaluation and results communication in a Canadian bank - a DEA application. European Journal of Operational Research 156 (2004) 719-735.
21. Smith, P.: Data Envelopment Analysis applied to financial statements. Omega: the international journal of management science 18 (1990) 131-138.
22. Thanassoulis, E.: A data envelopment analysis approach to clustering operating units for resource allocation purposes. Omega: the international journal of management science 24(4) (1996) 463-476.
23. Thompson, R.G., Brinkmann, E.J., Dharmapala, P.S., Gonzalez-Lima, M.D.: DEA/AR profit ratio and sensitivity of 100 large U.S. banks. European Journal of Operational Research 98 (1997) 213-229.
24. Tone, K., Sahoo, B.K.: Evaluating cost efficiency and returns to scale in the Life Insurance Company of India using Data Envelopment Analysis. Socio-Economic Planning Sciences 39 (2005) 261-285.
25. Weiss, M.A.: Efficiency in the Property Liability Insurance Industry. The Journal of Risk and Insurance 58 (1991) 452-479.

Using Predicted Outcome Stratified Sampling to Reduce the Variability in Predictive Performance of a One-Shot Train-and-Test Split for Individual Customer Predictions

Geert Verstraeten[1] and Dirk Van den Poel[1]

[1] Ghent University, Department of Marketing, Hoveniersberg 24, 9000 Ghent, Belgium
{Geert.Verstraeten, Dirk.Vandenpoel}@UGent.be

Abstract. Since it is generally recognised that models evaluated on the data that was used for constructing them are overly optimistic, in predictive modeling practice, the assessment of a model's predictive performance frequently relies on a one-shot train-and-test split between observations used for estimating a model, and those used for validating it. Previous research has indicated the usefulness of stratified sampling for reducing the variation in predictive performance in a linear regression application. In this paper, we validate the previous findings on six real-life European predictive modeling applications for marketing and credit scoring using a dichotomous outcome variable. We find confirmation for the reduction in variability using a procedure we describe as predicted outcome stratified sampling in a logistic regression model, and we find that the gain in variation reduction is – also in large data sets – almost always significant, and in certain applications markedly high.

1 Introduction

In the latest decades, due to the increasing usage of customer identification cards and loyalty programs, companies in very diverse industries have been able to proceed in building large transactional databases, recording all detailed interactions on an individual customer basis. Such interactions often include purchasing behavior, information requests, complaint behavior and subsequent complaint handling, survey information, etc. While this information serves for a large number of applications, in this study, we focus on the use of the transactional database for the predictive modeling of individual customer behavior, i.e. *individual customer predictions*. Indeed, ample previous research has proven that the historical information that resides in customer databases can aid in predicting future customer behavior on an individual level. For example, using the purchasing history of a given customer, companies have tried to assess e.g. whether this customer will (i) cease purchasing, (ii) respond to folders, (iii) be interested in certain products, (iv) increase their spending over their lifetime, (v) be able to refund granted credit, etc. Summarized, individual customer predictions mainly serve for targeted marketing and consumer credit scoring applications.

An intriguing concept in predictive modeling lies in the existence of overfitting. It is well established that predictive models have the tendency to be overly optimistic when their performance is measured on the same data used to build the models. Hence, adequate validation of such models require – at least – the usage of an independent holdout sample, a sample of data unseen by the classifier, that can be used to evaluate the true performance of the classifier [1]. As a practical solution to this, practitioners and researchers often start from a table of analysis, which is then split into two partitions: one used for estimating the model, and one used for validation. Very frequently, this split is performed using a random data partitioning. In his research, however, [2] provided evidence that, in this approach, the results are highly dependent on the particular split of the data used. Accordingly, replicating the test using a different random partitioning might produce very different performances of the particular estimation and validation sets. Additionally, he examined the use of Winsorization and stratified sampling to a multiple linear regression problem in an attempt to reduce the variability of the results. While Winsorization focuses on imposing boundaries for outliers in the target variable, stratified sampling ensures that the data is split in such a way that the distribution of the target variable of the estimation and validation sets is as similar as possible.

Building on Malthouse's study, we note that a large number of applications in the domain of individual customer predictions do not imply the use of a continuous variable, but instead attempt to predict a binary output variable. For example, we assess whether or not a customer will respond to an offer, will leave the company in a given time period, will purchase a certain product, will repay his credit, etc. In this study, we assess the usefulness of adapting the ideas in [2] to accommodate the use of a binary target variable, and we evaluate the benefits in terms of variance reduction on six real-life predictive modeling data sets.

The remainder of this paper is structured as follows. In the following section we describe the methodology that will be used in this paper, and defend the choices we make to perform the analyses. The next section covers a description of the data sets used in this study. Next, the results of the study are discussed, and in the last sections, the reader is offered conclusions, limitations, and suggestions for further research.

2 Methodology

2.1 One-Shot Train-and-Test Validation

Recently, the literature surrounding the assessment of a predictive model's performance has evolved drastically. To the best of our knowledge, current state-of-the-art domain knowledge prescribes that – in order to compare the predictive performance of different models – ten iterations of tenfold cross-validation should be applied. In tenfold cross-validation, the data is randomly split into ten subsamples. Subsequently, each sample serves iteratively as the holdout sample, while the other samples are used for model estimation. In order to compute the accuracy of the model, model perform-

ances are then averaged over the validation sets. In 10 x 10 cross-validation, the previously described procedure is performed ten times using a different random partitioning, and [3] have proven that this test shows, a high degree of replicability of the test besides acceptable Type I and Type II errors. However, because the samples used in this test are not independent, in order to correct for the resulting increased Type I error, they apply the 'corrected resampled t-test' suggested by [4].

However, for a variety of reasons, the widespread use of the one-shot train-and-test validation for predictive modeling is not without merit. The fact that the 10 x 10 cross-validation test requires 100 models to be built and validated might be responsible for the fact that only few applications involve in such rigourous testing. Indeed, in a number of situations, model builders require a more straightforward insight into the absolute performance of their models, while they would not necessarily proceed in testing whether significant differences occur between different model architectures. For example, a company that realizes that its customers are leaving will want to apply a predictive model in a timely manner in order to address the customers at risk. Hence, this company might continue to lose a lot of customers during a very extensive validation procedure, so time efficiency translates seamlessly into cost efficiency, and the company might choose to adopt a more straightforward validation procedure. Additionally, also in scientific readings, the use of the one-shot train-and-test validation is still popular. For example, many recent well-appreciated predictive modeling studies in *Marketing Science* report the use of a single split (see, e.g. [5, 6, 7]) for model validation. However, since it has been proven that the results of such a validation procedure are highly dependent on the particular split of the data used [2], in this study, we will consider the use of stratified sampling in an attempt to reduce this variability.

2.2 Predictive Modeling Technique

In the domain of individual customer predictions, given the variety of data available, it is possible to generate a large number of predictors that can serve in the model. It is not uncommon that such analyses are based on several hunderds of thousands of observations using several hunderds of candidate predictive variables. While at the advent of statistical theory, data sets of such magnitude were most likely beyond imagination, statistical techniques such as linear and logistic regression have been proven to show adequate predictive performance in such settings when benchmarked to other classifiers such as neural networks, decision trees, k-nearest neighbour, discriminant analysis and support vector machines [8, 9, 10]. They have become the standard method of analysing data with a discrete outcome variable in many fields in the eighties [1], and plausibly due to their ease-of-interpretation, regression models are still one of the main stalwarts of today's predictive model builders in industry [11]. Hence, in this study, we will use multiple logistic regression to predict customer behavior.

However, the use of a large number of candidate predictor variables implies that caution should be used when applying such models. First, the fact that the predictor variables are often closely related – often described as multicollinearity – has often

been accused of influencing parameter signs and greatly boosting the variance of the parameter estimates, rendering them uninterpretable [1]. Still, it has been well documented that this phenomenon need not hamper predictive performance on the condition that the multicollinearity persists in the validation set, and in the future population at large [12]. A second result of the large dimensionality is given by the existence of overfitting, implying that the inclusion of a large number of predictor features might lead to increases in the performance on the data used for calibrating the model, whereas *real* predictive performance – as measured when the model is applied to unseen data - does not increase, or even decreases. While we previously focused on the necessity of adequate model validation, feature selection can serve as a tool to reduce overfitting [1] and hence improve the predictive performance while at the same time reducing unnecessary or even unwanted complexity. In this study, we will apply a stepwise variable selection procedure, implying that features are entered iteratively according to the maximal contribution to the chi-square statistic, but the effects entered do not necessarily remain in the model. Each introduction of a new feature is followed by any possible removal of insignificant features, according to the Wald test for individual parameters [1]. In the analysis of the effect of stratification on the variance of predictive performance, we will compare the results of a model using all parameters, henceforth the *full* model, with the results of a model using only those parameters selected during a stepwise variable selection procedure.

2.3 Stratified sampling

In his recent study, [2] described the use of stratified sampling to reduce the variance of the estimates in a linear multiple regression problem. In this procedure, the author first sorts the data set according to the dependent variable. Next, strata are created by grouping consecutive observations, e.g. stratum one groups the first two observations, stratum two the following two observations, etc. Finally, the split between estimation and validation is performed by randomly assigning one of the observations in each stratum to the estimation set, and the remaining observation to the validation set. Hence, they use stratification to ensure that the distribution of the dependent variable is similar in both the training and test sets.

However, as already indicated, the domain of predictive modeling for targeted marketing and consumer credit scoring contains a number of core applications where the target variable is dichotomous. In its minimal form, in such cases, stratification implies that the sampling should ensure that the proportion of cases where the signal occurs (i.e. the incidence) is equal in both training and validation sets. It should be clear that this stratification procedure is far less stringent that the procedure offered by [2], and will not necessarily imply that the variation is adequately reduced.

In their study on variable selection in logistic regression models, [13] illustrate a convienient way of transforming a logistic regression problem into a linear regression problem. Several steps suffice in this procedure. First, the logistic procedure is performed, and the predicted probabilities (which we label *pred*) are registered. Next, the binary outcome variable (y) is transformed via the equation $z = \log(pred / (1 - pred)) + ((y - pred) / (pred * (1 - pred)))$. Let the observations be weighed by a variable

defined as $w = pred * (1 - pred)$. A regression procedure that uses the same predictors, yet using z instead of y as a dependent variable, and uses the weight variable w, will then obtain the same least squares estimates as the logistic regression. Interestingly, while designed for a very different application, this procedure does result in the creation of a new dependent variable, z, that is continuous, and that can serve to adapt the stratification procedure to resemble the procedure for multiple linear regression described in [2]. In the remainder of this paper, we will call this procedure *predicted outcome stratified sampling*, or shorter, POS sampling.

In this study, we will compare the variability of the predictive performance of a random partitioning into training and validation set with a stratified splitting as described in the procedure above. To this end, we will perform both the random and the stratified splitting 100 times using a different random number sequence. Note, however, that in both procedures, we ensure to control for the incidence, so that for example in a credit scoring problem where 1% of the customers fail to repay their debts, the defaulters are proportionally distributed across estimation and validation sets, so that the percentage of defaulters is constant over the different sets. Another difference in comparison with [2] is that, in our study, the estimation set will be twice as large as the validation set, since it is more common that a smaller amount of the observations are held out for validation purposes. To summarize, this implies that strata of three consecutive observations will be created based on a ranking according to the z values described in [13], whereby two observations of each stratum are randomly chosen for estimating the model, while the remaining observation is used in validating the model.

For model evaluation purposes, because we do not always possess profit information in every application, we will not use the gains chart used by [2], but instead we report the area under the receiver operator characteristics curve (AUC), since this measure evaluates the performance of a given classifier regardless of the choice of a particular discretisation cutoff. An intuitive interpretation of the AUC is that it provides an estimate that a randomly chosen instance of class 1 is correctly rated higher than a randomly selected instance of class 0 [14].

3 Data

In this study, we make use of six real-life proprietary European predictive modeling data sets. All data sets were constructed for company-driven applications, and hence represent a sizeable test bed for comparing alternative predictive models. All cases are binary classification cases, and applications lie in the domains of targeted marketing and credit scoring. In Table 1, we present some descriptive statistics about the datasets used, namely (i) the case description, (ii) the industry of the application, (iii) the incidence of the target feature, e.g. the percentage of churners, buyers, defaulters, etc present in the data set, (iv) the number of observations, (v) the condition index, representing the degree of multicollinearity present in the data set. All data sets involved show high degrees of multicollinearity, considering the fact that condition indexes of 100 or more appear to be large, causing substantial variance inflation and

great potential harm to regression estimates [12], and (vi) the number of predictive features in the data set.

Table 1. Descriptive statistics of the data sets used

(i) Case	(ii) Industry	(iii) Incid	(iv) Obs	(v) C.I.	(vi) Features
Loyalty	Retail	0.4738	878	111	35
Spending	DIY retail	0.2814	3 827	1 442	15
Partial Churn	Retail	0.2515	32 371	241	45
Churn	Subscription services	0.1307	143 198	767	167
Targeting	Retail	0.3082	741 234	4030	100
Credit Scoring	Mailorder	0.0089	38 064	114 593	137

4 Results

Table 2 presents some descriptive statistics of the predictive performance of the different models. In this table, we distinguish between the predictive performance on the estimation sample, the validation sample, and overfitting, which is defined as the difference between estimation and validation sample within a single split. For each of these samples, we report the mean, minimal, maximal, the range (being the difference between the maximal and the minimal), and the standard deviation of the AUC performance measure. The most important conclusion from this table is that the range as well as the standard deviation of the AUC is *in all cases* reduced by performing the stratified sampling procedure. We also note that these findings are consistent across the estimation and validation samples, and are also reflected in the variation of overfitting. Additionnally, no large differences can be found when a full model is computed versus a model that uses a stepwise variable selection procedure, implying that the results are not sensitive to the particular variables used in the different models. It is clear, however, that the improvements vary in size across the different data sets. Hence, Table 3 presents a summarized overview of the reduction in variance that can be reached by using POS sampling in a logistic regression model.

Table 2. Overview of descriptives of the variability in the predictive performance of the different models

Loyalty			Estimation					Validation					Overfitting		
Model	Mean	Min	Max	Range	Stddev	Mean	Min	Max	Range	Stddev	Mean	Min	Max	Range	Stddev
Full, Random	0.7852	0.7558	0.8142	0.0584	0.0120	0.7223	0.6645	0.7869	0.1224	0.0253	0.0629	-0.0285	0.1497	0.1782	0.0367
Full, Stratified	0.7827	0.7758	0.7934	0.0176	0.0033	0.7282	0.6895	0.7532	0.0637	0.0119	0.0545	0.0226	0.1019	0.0793	0.0143
Stepwise, Random	0.7610	0.7302	0.7918	0.0616	0.0122	0.7413	0.6765	0.7976	0.1212	0.0252	0.0197	-0.0674	0.1154	0.1828	0.0371
Stepwise, Stratified	0.7584	0.7473	0.7691	0.0218	0.0047	0.7477	0.7162	0.7720	0.0559	0.0105	0.0107	-0.0218	0.0511	0.0729	0.0143
Spending			Estimation					Validation					Overfitting		
Model	Mean	Min	Max	Range	Stddev	Mean	Min	Max	Range	Stddev	Mean	Min	Max	Range	Stddev
Full, Random	0.7621	0.7483	0.7795	0.0312	0.0064	0.7587	0.7260	0.7865	0.0604	0.0126	0.0035	-0.0376	0.0535	0.0911	0.0189
Full, Stratified	0.7635	0.7612	0.7679	0.0067	0.0011	0.7562	0.7425	0.7611	0.0186	0.0030	0.0072	0.0014	0.0218	0.0204	0.0034
Stepwise, Random	0.7620	0.7464	0.7764	0.0300	0.0063	0.7619	0.7303	0.7928	0.0625	0.0131	0.0001	-0.0439	0.0461	0.0899	0.0190
Stepwise, Stratified	0.7635	0.7593	0.7673	0.0079	0.0016	0.7600	0.7454	0.7656	0.0202	0.0033	0.0035	-0.0046	0.0171	0.0217	0.0036
Partial Churn			Estimation					Validation					Overfitting		
Model	Mean	Min	Max	Range	Stddev	Mean	Min	Max	Range	Stddev	Mean	Min	Max	Range	Stddev
Full, Random	0.8197	0.8147	0.8251	0.0104	0.0019	0.8159	0.8052	0.8244	0.0191	0.0038	0.0038	-0.0096	0.0199	0.0295	0.0057
Full, Stratified	0.8191	0.8187	0.8196	0.0010	0.0002	0.8171	0.8158	0.8179	0.0020	0.0004	0.0020	0.0009	0.0034	0.0025	0.0005
Stepwise, Random	0.8190	0.8136	0.8245	0.0108	0.0019	0.8157	0.8047	0.8246	0.0198	0.0038	0.0033	-0.0109	0.0197	0.0307	0.0057
Stepwise, Stratified	0.8184	0.8179	0.8192	0.0012	0.0002	0.8170	0.8160	0.8179	0.0019	0.0004	0.0014	0.0004	0.0031	0.0028	0.0005
Churn			Estimation					Validation					Overfitting		
Model	Mean	Min	Max	Range	Stddev	Mean	Min	Max	Range	Stddev	Mean	Min	Max	Range	Stddev
Full, Random	0.7759	0.7717	0.7803	0.0086	0.0016	0.7711	0.7618	0.7805	0.0187	0.0032	0.0048	-0.0085	0.0185	0.0270	0.0047
Full, Stratified	0.7758	0.7751	0.7766	0.0015	0.0003	0.7714	0.7697	0.7727	0.0030	0.0006	0.0044	0.0029	0.0063	0.0033	0.0006
Stepwise, Random	0.7737	0.7673	0.7777	0.0103	0.0018	0.7700	0.7601	0.7785	0.0184	0.0032	0.0036	-0.0091	0.0169	0.0260	0.0048
Stepwise, Stratified	0.7737	0.7684	0.7751	0.0067	0.0008	0.7701	0.7672	0.7719	0.0047	0.0009	0.0035	-0.0001	0.0060	0.0061	0.0011
Targeting			Estimation					Validation					Overfitting		
Model	Mean	Min	Max	Range	Stddev	Mean	Min	Max	Range	Stddev	Mean	Min	Max	Range	Stddev
			0.7414	0.0027	0.0004				0.0054	0.0009				0.0081	0.0013
Full, Random	0.73997	0.73868	3	4	6	0.7401	0.73721	0.74262	1	1	-0.0001	-0.0039	0.0042	3	7
			0.7401	0.0001					0.0002		0.0002	0.0001	0.0004	0.0002	
Full, Stratified	0.7401	0.74003	9	5	2.8E-05	0.73982	0.73965	0.73992	7	4.7E-05	8	6	1	6	4.8E-05
			0.7413	0.0027	0.0004				0.0053	0.0009			0.0040	0.0080	0.0013
Stepwise, Random	0.73986	0.73861	2	1	6	0.74009	0.73724	0.74263	9	1	-0.0002	-0.004	2	4	6
			0.7400	0.0001					0.0002		0.0001		0.0003	0.0002	
Stepwise, Stratified	0.73999	0.73991	8	7	3.7E-05	0.73982	0.73967	0.73992	5	5E-05	8	4.9E-05	1	6	5.5E-05
Credit Scoring			Estimation					Validation					Overfitting		
Model	Mean	Min	Max	Range	Stddev	Mean	Min	Max	Range	Stddev	Mean	Min	Max	Range	Stddev

Full, Random	0.9031	0.8823	0.9179	0.0356	0.0065	0.8456	0.7889	0.8948	0.1058	0.0193	0.0575	-0.0044	0.1274	0.1318	0.0243
Full, Stratified	0.9026	0.8889	0.9121	0.0232	0.0048	0.8480	0.7974	0.8745	0.0771	0.0138	0.0546	0.0233	0.1146	0.0914	0.0161
Stepwise, Random	0.8554	0.6709	0.9004	0.2295	0.0567	0.8297	0.6052	0.8983	0.2931	0.0600	0.0257	-0.0302	0.0942	0.1243	0.0235
Stepwise, Stratified	0.8582	0.6658	0.8901	0.2243	0.0541	0.8333	0.6234	0.8780	0.2545	0.0583	0.0249	-0.0140	0.0712	0.0852	0.0172

Table 3. Factor with which the variance decreases by using stratification

	Full Model			Stepwise Model		
	Estima-tion	Valida-tion	Overfit-ting	Estima-tion	Valida-tion	Overfit-ting
Loyalty	**13.03**	**4.54**	**6.56**	**6.83**	**5.75**	**6.74**
Spending	**35.01**	**17.43**	**29.87**	**14.94**	**15.87**	**27.97**
Partial Churn	**102.53**	**81.39**	**130.49**	**80.00**	**81.64**	**114.21**
Churn	**22.35**	**29.17**	**52.49**	**4.75**	**11.73**	**19.79**
Targeting	**262.71**	**373.60**	**813.55**	**150.78**	**322.24**	**605.13**
Credit Scoring	**1.87**	**1.95**	**2.29**	1.10	1.06	**1.86**

The numbers in Table 3 should be interpreted as follows. The upper left figure is reached by dividing the variance in the predictive performance of the estimation set of the full model of the 'Loyalty' application when a random partitioning is used, by the corresponding variance when POS sampling is used. Hence, in that particular situation, the variation of the random partitioning is over 13 times as large as the variation of the POS sampling. Levene's test for the homogeneity of variance [15] was applied in order to analyse the significance of the differences in variation. Significant drops in the variance (at $p < 0.01$) are indicated in bold face. We conclude that, in all models but the stepwise model of the 'Credit scoring 1' data set, the drop in variance is statistically highly significant.

5 Conclusions

In business as well as academia, the use of a single shot train-and-test split to perform model assessment is not uncommon. Surprisingly, even when large data sets are used, the results of the models can vary strongly when data is partitionned into an estimation and a validation sample on a random basis. The ongoing use of a single split as a validation procedure implies that model builders may benefit from a reduction of variability in model performance. In this study, we provide evidence that the insights of [13] regarding the similarities between linear and logistic regression can be used to adapt the stratification procedure suggested in [2] in order to apply a variance reduction heuristic that can accommodate predictive models with a dichotomous outcome variable. In this study, we have computed the reduction in variance of the predictive performance on six real-life European predictive modeling applications for marketing and credit scoring. The predicted outcome stratified (POS) sampling used consistently succeeds in reducing the variance of the predictive performance in the estimation and validation samples, but also in the overfitting, and this effect was confirmed across a model containing all variables, and a model containing only those variables selected by a stepwise variable selection procedure. However, across the different applications, the gains that can be used in terms of variance reduction vary. In the least successful case, the gains are non-significant, whereas in the most successful application,

the variation in overfitting is over 800 times lower when POS sampling is used instead of random sampling.

This has important implications. In those situations that time is only available to compute one (or a limited amount) of validation iterations, the use of a random split seems never more justified than the use of the stratified split procedure suggested here. Indeed, the only requirement to perform a stratified split is the outcome of a run of the logistic regression model on the total data set. The gain exists in the fact that it is (sometimes far) more likely that the resulting performance assessment will be more accurate.

6 Limitations and Issues for Further Research

This study has a number of limitations. In contrast to the paper of [2], in this study, our focus lies on the absolute question instead of the relative question. Indeed, the center of attention in this study lies in model assessment, whereas [2] focusses on model selection, i.e. deciding which model offers the best predictive qualities. Due to the data complexities involved in an analysis on a test bed of large data sets, we did not compare different classifiers, variable selection techiques, etc. Hence, future research might be directed towards assessing the usefulness of POS sampling in logistic regression in order to compare the differences in performance of alternative predictive models.

Acknowledgements

The authors would like to express their highest gratitude to Wouter Buckinx, Jonathan Burez, Bart Larivière and Bernd Vindevogel, who contributed the data sets they gathered during their PhDs in order to enable the experiments performed in this study.

References

1. Hosmer D.W. and Lemeshow S.: Applied Logistic Regression, John Wiley & Sons, New York (1989)
2. Malthouse E.C.: Assessing the performance of direct marketing scoring models. Journal of Interactive Marketing **15(1)** (2001) 49-62
3. Bouckaert R. and Frank E.: Evaluating the Replicability of Significance Tests for Comparing Learning Algorithms. In: Dai H., Srikant R. and Zhang C. (eds.): Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'2004). Springer (2004) 3-12
4. Nadeau C. and Bengio Y.: Inference for the generalization error. Machine Learning **52** (2003) 239-281
5. Montgomery A.L., Li S., Srinivasan K. and Liechty J.C.: Modeling Online Browsing and Path Analysis Using Clickstream Data. Marketing Science **23(4)** (2004) 579-595

6. Park Y.-H. and Fader P.S.: Modeling Browsing Behavior at Multiple Websites. Marketing Science **23(3)** (2004) 280-303
7. Swait J. and Andrews R.L.: Enriching Scanner Panel Models with Choice Experiments. Marketing Science **22(4)** (2003) 442-460
8. Baesens B., Van Gestel T., Viaene S., Stepanova M., Suykens J., and Vanthienen J.: Benchmarking State of the Art Classification Algorithms for Credit Scoring. Journal of the Operational Research Society **54** (2003) 627-635
9. Davis R.H., Edelman D.B. and Gammerman A.J.: Machine-Learning Algorithms for Credit-card Applications. IMA Journal of Mathematics Applied in Business and Industry **4** (1992) 43-51
10. Dasgupta C.G., Dispensa G.S. and Ghose S.: Comparing the predictive performance of a neural network model with some traditional market response models. International Journal of Forecasting **10(2)** (1994) 235-244
11. Thomas L.C.: A Survey of Credit and Behavioral Scoring: Forecasting Financial Risk of Lending to Customers. International Journal of Forecasting **16** (2000) 149-172
12. Belsley D.A.: Conditioning diagnostics, collinearity and weak data in regression. John Wiley & Sons, New York (1991)
13. Hosmer D.W., Jovanovic B., & Lemeshow S.: Best subsets logistic regression. Biometrics **45** (1989) 1265-1270
14. Hanley J.A. and NcNeil B.J.: The meaning and use of the area under a receiver operating characteristic (roc) curve, Radiology **148** (1983) 839-843
15. Levene, H.: Robust Tests for the Equality of Variance. In Olkin I. (ed): Contributions to Probability and Statistics. Stanford University Press, Palo Alto, CA (1960) 278-292

A text mining system for bioinformatics: requirements and architecture

Ilkka Karanta, Antti Pesonen, Lauri Seitsonen, and Paula Silvonen

VTT Digital Information Systems, Box 1000 (VM3),
FI-02044 VTT, Finland
{Ilkka.Karanta, Antti.Pesonen, Lauri.Seitsonen, Paula.Silvonen}@vtt.fi
http://www.vtt.fi/

Abstract. We describe OAT, a new information extraction system under development. It extracts relevant (subject, predicate, object) triplets from natural language texts. It uses ontologies extensively: the results are saved in an ontology, and ontologies are used in the information extraction process itself. It is adaptable both to a domain of discourse and within a domain of discourse (finding new concepts). This paper concentrates on the requirements and architecture of OAT.

1 Introduction

Text mining has received a lot of attention in bioinformatics lately [3]. The main reason is that the number of publications in the biosciences is vast and growing at an accelerating pace. Thus, all the forms of text mining [7] – information retrieval (IR), information extraction (IE), summarization, categorization, clustering, topic tracking, information visualization and question answering being the most important – are being utilized in the field.

IR systems aim to identify and retrieve texts concerning a particular topic. IE systems extract pre-specified types of facts from texts [3]; the goal is usually to find unexpected connections between entities (e.g. genes and diseases).

Co-occurrence and natural language processing (NLP) are two approaches that are currently being used for extracting relationships from biological texts. In co-occurence, the idea is to identify entities that co-occur repeatedly in sentences or abstracts and assume there is a relationship between them. NLP methods analyse both syntax and semantics of the sentences, which makes them more reliable than methods based on simple co-occurence.

Some current IE and text mining systems are PreBind [1], an IE system for locating protein-protein interaction information in PubMed abstracts based on support vector machine; PubGene [4] IE system, based on simple name co-occurrence; Pathway Studio [5], for navigation and analysis of biological pathways, gene regulation networks and protein interaction maps; iProLink [2], a resource to facilitate text mining and NLP research in literature based database curation, named entity recognition, and ontology development; and Journal Mine, a portal on top of GeneWays [6], that mines and organises data from scientific journals.

Impressive progress has been made in information extraction, but certain issues still need to be addressed. In particular, existing systems generally have the following properties:

- they underutilize the already extracted knowledge. Thus, for instance, term lists used in retrieval and extraction aren't updated with the results of the IE process.
- they leave the extracted information in textual form, and it has to be post-processed for use in e.g. ontologies. This is usually done manually.
- they are usually very large. This makes their use clumsy.
- connecting them to other software is usually awkward.
- they don't learn or adapt to new findings.

OAT (Ontology Aided Text Mining) is an IE system that aims to address these issues.

2 Requirements

OAT extracts (subject, predicate, object) triplets from sentences in a text corpus.

The input of a text mining process is a corpus of text. At this stage, OAT uses PubMed, a collection of biomedical abstracts, but can be used with other information sources, too.

Another starting point is a set of interesting concepts. These concepts determine what concepts should occur at least somewhere in an extracted triplet.

The IE process uses various natural language processing methods. It can handle synonyms, anaphora, and other complications of natural language.

The end result of a single run is a set of (subject, predicate, object) triplets. The subject and object are relevant concepts and their attributes, and predicate is a verb phrase connecting them. All of these are standardized: e.g., the subject and object contain the name of the concept (which is not necessarily the phrase occurring in the text); and the verbs are resolved to their basic forms. This allows for the automatic processing of the triplets later. Each triplet also contains a pointer to the publication where the triplet was extracted from.

The main purposes of the extraction of triplets are twofold. The triplets form a directed graph, where the subject (node) points to the object (node) via the predicate (arc). If a new connection (path) between two concepts (concept nodes) – say, a genetic trait and a disease – can be found, it might be of great scientific importance, possibly redirecting further research. The concept graphs can also be visualized, giving experts insights to the relationships between concepts in their field.

3 Architecture

OAT is a client-server software developed with Java. The server side of the system is divided to a number of well defined processes, each handling a specific task of the OAT processing. Figure 1 shows the overall architecture of the system.

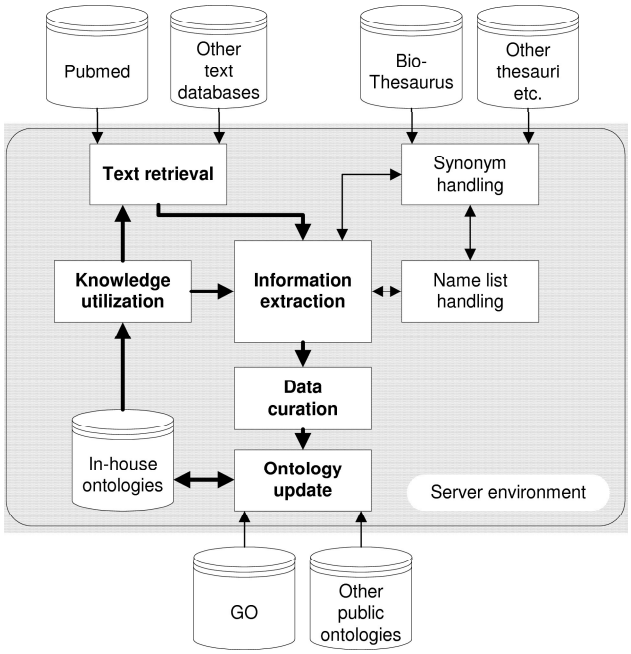


Fig. 1. The main modules of OAT architecture

The architecture relies on the thin client / extensive server side processing paradigm. Thin clients are browser run applets that can be easily run in a lightweighted hardware like small, inexpensive notebooks. All heavy processing and interaction with external data sources occurs on the server. Information extraction processing is computationally very demanding already with relatively small corpora; the thin client approach is thus well justified.

3.1 Server side

OAT processing starts with text retrieval. The interesting scientific articles are filtered from the target (external) datasource by giving a boolean search clause. At first only conjunctive (AND) concept lists are accepted as search clauses. The external datasource interface is implemented on an abstract Java interface to facilitate additional datasource connection tailoring.

IE processing is implemented on top of GATE - General Architecture of Text Engineering by the NLP group at the University of Sheffield (http://gate.ac.uk/). GATE offers a pipeline approach for NLP. Sub-processes of the pipeline are tuned for OAT. Furthermore, the pipeline is extended with our propietary modules.

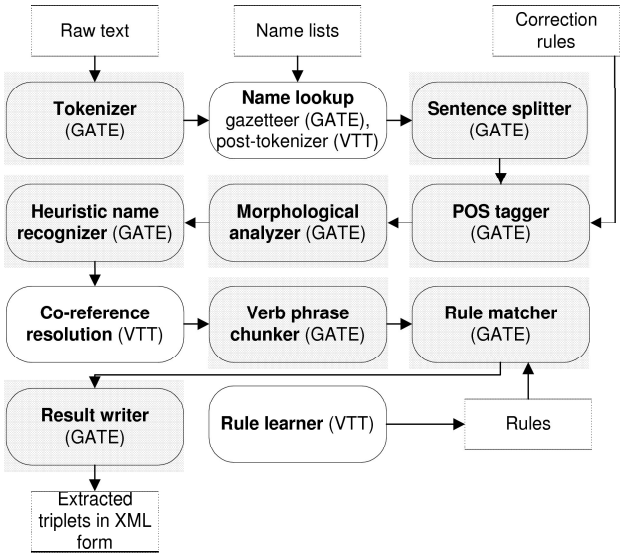


Fig. 2. The information extraction process of OAT

In figure 2, the NLP modules are furnished with the origin of the code (GATE or VTT). The name lists, the correction rules for the POS tagger and the rules for the rule matcher are formed in collaboration with the biologists at VTT. Additionally, the name lists are supplemented with synonym hits found from expernal synonym databases (like BioThesaurus).

The generated triplets are curated by a scientist with sufficient knowledge on the subject. The data curation phase is crucial for preventing false positive triplets from corrupting the resulting ontology.

The curated triplets form the input for updating the in-house ontologies. Each biological concept in our ontology is furnished with unique identifier. A cross reference database is built on top of the ontology to connect the concepts in our ontology to other, relevant datasources (like Gene Ontology). The cross-reference system is constantly kept up to date.

A key feature of OAT is in the knowledge utilization module (see figure 1). The module compares the input data of former text retrieval and information extraction processes to the knowledge captured by the in-house ontology. If the ontology contains associations that link concepts in former input data to concepts that were thought irrelevant (or were unknown) the text retrieval / information

extraction process is re-started with updated input. The number of iterations in the processing loop is controlled by rules restricting the validity of the novel concepts (for example, concepts that lie behind several association links from the original concepts of the original name lists are not considered valid).

3.2 Client side

The user controls the OAT system with a client side applet. The GUI is divided to "sheets", each of them offering control over text retrieval, name and synonymy handling, information extraction or data curation.

Additionally, the GUI provides controls, for example, to schedule the ontology update process (either calendar based or manual, controlled by the data curator) and to set the parameters of the knowledge utilization module.

4 Further work

Many open research issues remain. The input of OAT, scientific papers, contain findings that often are by their nature uncertain. How to quantify this uncertainty, how to extract a measure for it from the texts (automatically, if possible), and how to combine the measures of uncertainty remains an open issue. Natural language, including scientific texts, always contains some imprecision; how to model and manage this is still unresolved. Yet another issue is entity recognition (how to find the biological entities that are mentioned within a text).

References

1. Donaldson, I., Martin, J., de Bruijn, B., Wolting, C., Lay, V., Tuekam, B., Zhang, S., Baskin, B, Bader, G.D., Michalickova, K., Pawson, T., Hogue, C.W.V.: PreBIND and Textomy - mining the biomedical literature for protein-protein interactions using a support vector machine. BMC Bioinformatics **4** (2003) 11
2. Hu, Z.-Z., Mani, I., Hermoso, V., Liu, H., Wu, C.H.: iProLINK: an integrated protein resource for literature mining. Computational Biology and Chemistry **28** (2004), 409-416.
3. Jensen, L.J., Saric, J., Bork, P.: Literature Mining for the Biologist: from information retrieval to biological discovery. Nature Rev. Gen. **7** (2006) 119–129
4. Jenssen, T.-C., Laegreid, A., Komorowski, J., Hovig, E.: A literature network of human genes for high-throughput analysis of gene expression. Nature Genetics, May 2001.
5. Nikitin, A., Egorov, S., Daraselia, N, Mazo, I.: Pathway Studio - the analysis and navigation of molecular networks. Bioinformatics applications note, **19** (2003), 1-3
6. Rzhetsky, A., Iossifov, I., Koike, T., Krauthammer, M., Kra, P., Morris, M., Yu, H., Duboue, P.A., Weng, W., Wilbur, W.J., Hatzivassiloglou, V., Friedman, C.: GeneWays: a system for extracting, analyzing, visualizing, and integrating molecular pathway data. Journal of Biomedical Informatics **37** (2004), 43-53
7. Sullivan, D.: Document Warehousing and Text Mining. John Wiley & Sons, New York (2001)

Differential Voting in Case Based Spam Filtering

Deepak P, Delip Rao, Deepak Khemani

Department of Computer Science and Engineering
Indian Institute of Technology Madras, India
deepakswallet@gmail.com, delip@cse.iitm.ernet.in, khemani@iitm.ac.in

Abstract. Case-based reasoning (CBR) has been shown to be of considerable utility in a spam-filtering task. In the course of this study, we propose that the non-random skewed distribution of the cases in a case base is crucial, especially in the context of a classification task like spam filtering. In this paper, we propose approaches to improve the performance of a CBR spam filter by making use of the non-random nature of the case base. We associate each case in the case base with a voting power, which is essentially a function that incorporates the knowledge of the local neighborhood of the case. We show that the performance of the spam filter can be considerably improved by making use of such techniques that incorporate the voting powers.

1 Introduction

Spam or unsolicited email is a major concern to the industry and the end users. Finding a legitimate email in a deluge of spam mails can become a daunting task. Automated techniques for spam filtering have been used with considerable success to distinguish spam messages from the non-spam ones. One of the dangers of using spam filtering is to classify a legitimate email as spam. A pessimistic approach is usually adopted, i.e., when in doubt leave the message as legitimate. This however lowers the accuracy of such systems. Automatic spam filtering systems are continuously striving to improve accuracy. Case-based reasoning has been shown to be of considerable value in a spam-filtering task [2, 3]. It is interesting to note that the distribution of the spam cases in the case space is not uniform. Data mining techniques can be used to exploit this skewed distribution (hereafter referred to as patterns) for our advantage. We associate every case in the case base with a voting power, which incorporates the knowledge of the neighborhood of the case, which can then be made use of to classify a test case. We propose various algorithms for computing the voting power of a case, and show that many of them perform better in comparison with the traditional techniques. Further, the complexity of re-computing the voting powers when an addition or deletion occurs to the case base is linear in the number of cases in the case base.

Section 2 reviews the related work in the area. Section 3 lays down the motivation behind the work and justifies the motivation by some experiments. Section 4 describes the techniques that we propose (and the intuition behind them) and the techniques with which we compare our approaches. Section 5 lists the performance

measures used and Section 6 lists the experiments conducted, the results and their implications.

2 Related Work

The work on spam filtering (as a classification task) has warranted lot of investigation in the past using several classifiers with Support Vector Machines giving the best performance [9]. In this paper we describe a memory based approach to spam filtering. It has been shown [1] that memory based approaches for spam filtering work significantly better than well studied naïve Bayesian approaches. Further, they go on to say that it might probably due to the fact that there are many more types of messages rather than just spam and legitimate. For instance, the common classes of spam mails that the authors receive include mortgage mails, free university degree mails and so on and so forth. A more recent work [2] proposes new methods of feature selection based on spam and non-spam vocabularies and asserts that a CBR approach to spam filtering can effectively track and adapt to the changing behavior of spammers and legitimate mails (concept drift) and provides methods for the same. It uses the conventional and intuitive CBR approach of majority voting (the voters being the neighbors of the test case in the case base) to flag a message as either spam or legitimate. It may be noted that the tracking of concept drift can be easily incorporated in a CBR system, and that the incremental (online) model acquisition is a natural process in a CBR system. This may be contrasted with other conventional classification systems, where the model is built based on only the training data. Another work [3] incorporates various ideas specific to spam filtering. Firstly, it incorporates the notion that a legitimate message classified as spam is much costlier than a spam message labeled legitimate. Conventional spam filtering systems, either label a mail as spam (labeling systems), or send it to a special folder such as "bulk" or "spam" (redirection systems). Thus a legitimate mail is classified as spam may well escape the attention of the user in systems that employ the latter method. It has been estimated empirically estimated that a false positive (legitimate mail classified as spam) is 9 times costlier than a false negative in a labeling system, whereas the ratio is 999 for a redirection system. This error disparity is hereafter referred to in the paper as "Severity Disparity". Further, the same work presents a significant departure from the conventional CBR model in that it incorporates differential weighting of votes, viz., the closer a neighbor is, to the test case, the higher would be the value of it's vote (we call this technique Diff-CBR hereafter in this paper). It also incorporates differential weighting of features in the vector space. In this paper, we compare our techniques to the approaches used in the latter two papers.

3 Motivation and Justification

In the day-to-day life, an average web user comes across a wide range of spam and legitimate mails. Intuitively, most mails fall into more categories than just two classes as spam and legitimate. Many of the spam mails that the authors receive fall into categories such as "interest free home loans", "mortgage", "easy university degree"

and pornographic spam mails. We will later show that these spam mail fall into well defined clusters or patterns in the case-base of the CBR spam filter. Our hypothesis is that by making use of such patterns we should be able to improve the performance of a CBR spam filter considerably. It may well be re-emphasized here that the need for incremental model acquisition (to tackle concept drift) and the presence of more (logical) classes than the ones to which a case is to be classified as, makes CBR the natural choice for a spam filtering system. In the following subsections, we describe the dataset used and justify our assertion that spam mails form clusters experiments on the corpus.

3.1 Dataset Used

We choose to use the SpamBase Database (hereafter referred to as the corpus) compiled by George Forman of HP Labs. It is available through the University of California Irvine Machine Learning Repository[1]. SpamBase is a collection of 4601 pre-processed messages, each message represented as a labeled (as either spam or legitimate) vector of 57 selected features and contains 39.4% spam messages.

3.2 Justification

We now show the existence of clusters in the mail corpus and for suitable values of k we can get sufficiently "pure" clusters that we can use for our filtering purpose. Purity can be defined as the ratio of the sum of the cardinalities of the maximally represented class for each cluster (across all clusters) to the total number of cases clustered [8]. We applied the traditional k-means clustering algorithm with k = 15 with varying initial cluster centers on the corpus. Several values of k were used and we observed empirically that a large value of k yielded clusters of high purity. Note that our method is parameterized on k. It may be noted that fixing k at 15 is motivated by the need to choose a high value of k to illustrate the clustering in the corpus and is not related to any background knowledge of the corpus whatsoever. The presence of clusters would reveal a skewed distribution of messages among clusters and that is exactly what we ran into. Among the 15 clusters that we obtained, one of them had 42% of the corpus and only 5 clusters had more than 5% of the corpus. The percentages of the corpus in each of the 15 clusters are listed in table 1.

Table 1. Distribution in k-Means Clusters with k = 15

Percentages of the Corpus in Clusters														
11	1	20	3	6	4	3	1	1	2	42	1	1	0	6

Having justified our claim that there are clusters in the corpus, we now show that the clusters are pure enough (using the labels in the corpus). The overall purity of a clustering is the weighted average of the purities of the different clusters. Figure 1 shows the weighted average of the purity of the k-means clusters against k. As can be seen, the purity plot meanders around 0.78, which shows that the clusters are pure

[1] http://www.ics.uci.edu/~mlearn/MLRepository.html

enough. We now propose various techniques to make use of the skewed distribution in the CBR framework and compare it with the state-of-the-art techniques such as simple CBR and diff-CBR.

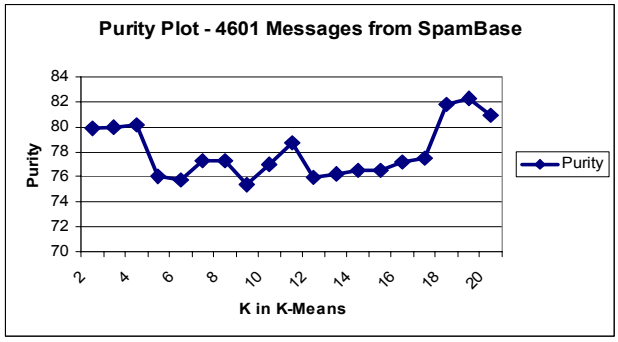


Figure 1. Total Purity of Clusters with varying k

3.3 Broad Methodology

In each of the techniques that we propose (in sections that follow), we use the concept of voting powers for a case in a case base. Given a case base, we can associate each case with a voting power depending on the cases in its vicinity and their labels. In order to compute voting power, we can either consider

- The k nearest neighbors (k-NN) of the case (and their labels) in the case base (k may be an input parameter)
- All cases which are no farther from the case than a given distance (which would be taken as an input parameter)

Given a test case and the ultimate aim of classifying it as either spam or legitimate, we need two functions:

- Confidence_Spam(Test_Case T, k-NN of the test case in the Case Base) which returns the confidence of the test case being spam and
- Confidence_Legitimate(Test_Case T, k-NN of the test case in the Case Base) which returns the confidence of the test case being legitimate

The intuitive algorithm for filtering is given by:

- Flag(Test_Case T) = Confidence_Spam(T,k-NN) > Confidence_Legitimate(T,k-NN) ? “Legitimate” : ”Spam”

In cases where the confidences are equal, we argue that we should “play safe” and label it as legitimate.

The Confidence functions that we propose make use of the voting powers of each of the k nearest neighbors, and optionally their distances from the test case. We use Confidence_X to denote two functions Confidence_Spam and

Confidence_Legitimate. Both the confidence functions use the same algorithm, except for the fact that one considers only spam elements in the k-NN and vice versa. We present a broad framework of the classifier algorithm.

```
Confidence_X(Test_Case T, k-NN of T in the Case Base)
{
    confidence = 0.0;
    for each C among the k-NN neighbors
    {
         if(label(C) == X)
         {
           add_pwr = Voting_Power(C);
optionally,
           add_pwr = add_pwr / distance(Test Case, C);
           confidence = confidence + add_pwr;
         }
     }
     return confidence;
}
```

Having introduced as many primitives as has been done, each algorithm can be specified by the voting power computation function and as to whether it involves the optional step in the algorithm. We call algorithms that include the optional step as *distance-weighting algorithms* for the sake of brevity hereafter in this paper.

4 Techniques for Spam Filtering using CBR

We present two techniques that have already appeared in literature followed by six techniques that we propose, to improve the performance of the CBR Spam Filter. All the descriptions use the primitives introduced in the preceding section. The intuition behind each technique that we propose has been detailed therein.

Simple CBR. A simple CBR [2] is a non-distance-weighting algorithm that uses a constant voting power function. Presenting it in another fashion, it takes the majority vote for classification.

Diff-CBR. This technique [3] which has been shown to be much better than Simple CBR is a distance-weighting algorithm that uses a constant voting power function.

C1 CBR. A case in the case base which is part of a spam cluster would have mostly spam cases among its k-NN (and vice versa). Such a case surrounded by spam cases being among the k-NN of a test case, intuitively gives a higher confidence that the test case is part of or in the near vicinity of the spam cluster (and vice versa). There is a host of clustering algorithms which rely on finding elements with a dense neighborhood and use them as seed points for identifying clusters [5,6,7]. C1 CBR is a variation of Simple-CBR which incorporates this notion in a straightforward manner. It is a non-distance-weighting algorithm in which the voting power of a case

is the number of cases with the same label as the case (in question) among its k-NN. In order that no case is assigned a voting power of zero, we include the case itself among the k-NN neighbors of the case to compute its voting power.

C2 CBR. This is a variation of Diff CBR along the same lines as C1 CBR and is a distance weighting algorithm, where the voting power function is exactly the same as in C1 CBR.

C3 CBR. C1 and C2 CBR are suspected to suffer from a serious drawback. Consider a dense spam cluster and a singleton point in an isolated area of the case base. All points in the spam cluster would get a voting power of k when k nearest neighbors are considered (unsurprisingly), and the singleton point would also get a voting power of k (surprisingly!) if all its k-NNs are spam (possibly, they are part of the dense cluster) although they are a considerable distance away compared to the former case. C3 CBR tries to rectify this problem by introducing an additional parameter, which we hereafter refer to as the radius. The voting power of a case in the case base is computed as 1 + (number of cases with the same label as the case in question, and which fall within a distance of radius from the case). The addition of 1, once again is to ensure that no case gets a zero voting power. This is a non-distance-weighting algorithm. We would like to clarify at this point that determining an optimal value for radius is a non-trivial task.

Better Voting Power Function (BVPF). A bit of thought is more than sufficient to come up with the insufficiencies of the C3 CBR voting power function. Consider a highly noisy space where a spam case has 50 spam cases and 50 legitimate cases within its radius. On the contrary, consider a pure space which has a sparse cluster where a spam case has just 5 spam neighbors within its radius. Intuitively, the second case deserves a better voting power whereas the C3CBR voting power function assigns a voting power of 51 to the former and 6 to the latter; a huge disparity indeed. Although the frequency of such hostile cases have to be studied, the disparity introduced by the C3CBR voting power function is so high that we can't let it go unattended. In this context, we choose to lay down some of the more intuitive desiderata for a voting power function.

Assume that the total number of cases in the radius of the case in question is t, let the number of cases with a matching label among them be m_1 and those with mismatching labels among the t cases be m_2. Firstly, the voting power function should be directly related to m_1. Secondly, the voting power function should be directly related to t. Thirdly, it should be inversely related to m_2. Given that t is $(m_1 + m_2)$, one might reasonably argue that any two of the above relations should be sufficient. Such a function is highly non trivial. We propose a voting power function, hereafter referred to as BVPF which we define as follows.

$$BVPF(t, m_1, m_2) = (m_1 - m_2) * log(t) / t$$

A closer look at the function would reveal that it favors dense areas compared to sparse ones. To illustrate the aspect, $BVPF(70,50,20) > BVPF(7,5,2)$. Although it is easy to create a hostile situation to BVPF, we argue that a hostile situation for C3CBR function is much more probable than one for BVPF. In the remaining algorithms that we propose, we consistently use BVPF as the voting power function.

C4 CBR. This is a non-distance-weighting algorithm which uses the BVPF as the voting power function.

C5 CBR. This is the distance-weighting algorithm which uses BVPF as the voting power function.

C6 CBR. As mentioned earlier, determining an optimal value for radius is a non-trivial task. C6 CBR tries to make the process as much insensitive to the value of the radius parameter. We distort the confidence function a bit and redefine it as following:

$$\text{Confidence\_X\_C6CBR}_{\text{radius=r}}\text{(T, k-NN)} = \sum_{i=1}^{n} \text{Confidence\_X}_{\text{radius=i*r}}\text{(T, k-NN)}$$

$\text{Confidence\_X}_{\text{radius=r}}$ (T, k-NN) denotes Confidence_X(T, k-NN) computed with BVPF as the voting power function and r taken as the radius for the BVPF computations. We consistently set n = 5 (the number of terms in the right-hand-side of the above equation) in the course of our experiments with C6 CBR.

5 Performance Measures Used

We use various performance measures for evaluating the different techniques for spam filtering described above. *Spam precision* is the percentage of messages classified as spam that truly are. *Spam recall* (interchangeably referred to as spam accuracy) is the proportion of actual spam messages that are classified as spam. Non-spam messages are usually called solicited messages or legitimate messages. *Legitimate precision*, analogously, is the percentage of messages classified as legitimate that truly are. *Legitimate recall* (interchangeably referred to as legitimate accuracy) is the proportion of actual legitimate messages that are classified as legitimate [4]. The *total accuracy* is the total number of messages classified correctly. Intuitively, as discussed earlier, the severity of the error of classifying a spam message as legitimate is much less than the severity of classifying a legitimate message as spam. Taking these into account, we define an *error cost function* as the sum of the errors with differential weighting for the two kinds of errors. The obvious parameter to this cost function would be the difference in severities. It has been shown [1] that the cost of the latter error is 999 times that of the former in a setting where spam messages are blocked from the user. In a scenario where messages are just flagged as spam by the filter, the disparity in severity comes down to 9. The error cost is hence calculated as (Severity Disparity)*(# of false positives)+(# of false negatives). We analyze the techniques with both values for the severity disparity parameter.

6 Experiments, Results and Implications

In this section, we walk through (in chronological order) the results of the various experiments that have been conducted. As is typical in any experiment in this context,

we divide the corpus into the seed case base (the training set) and the test set (which in the real world context, would be a stream of incoming mail messages). The case base forms the training set and each message in the test set is classified by the CBR making use of the case base. In cases where we choose the seed case base to have a size of 50% of the corpus, we start off putting every other message from the corpus into the seed case base. As the order of the corpus is not representative of the order of messages coming in (the SpamBase corpus does not have time-stamped messages), we choose not to add processed test cases to the case base. It may well be appreciated that adding processed test cases to the case base would be helpful only in cases where the order of consideration of cases is representative of the order of arrival of messages. Given that the entire corpus (and hence the test set too) is labeled, we can get a feel of the performance of the algorithms in this regard.

6.1 Performance of non-radius based techniques

As explained in the preceding section, Diff-CBR, Simple CBR, C1 and C2 CBRs don't require a radius parameter. We experimented with them on varying case base sizes. We present the spam precision and total accuracy charts for those techniques.

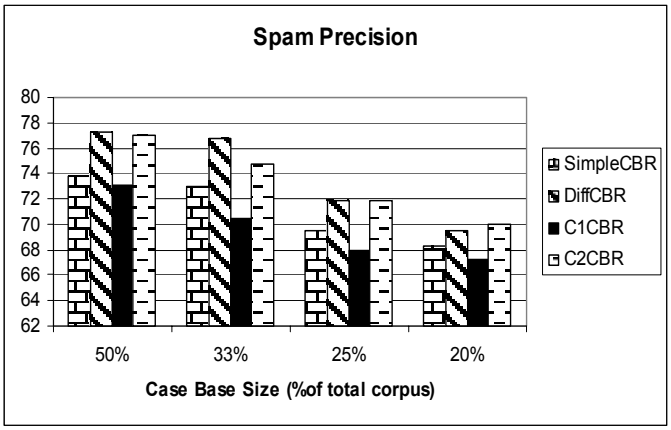


Figure 2. Spam Precision Chart

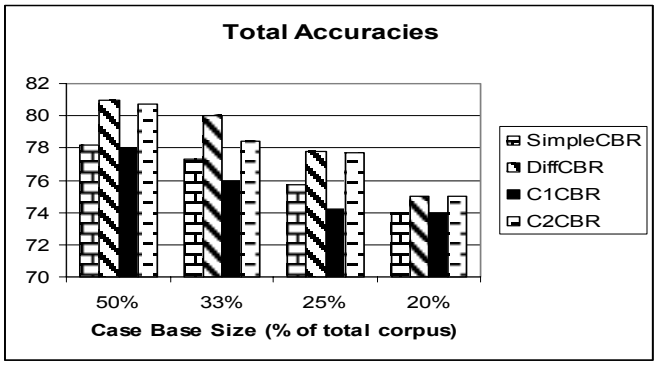


Figure 3. Total Accuracy Chart

As can be seen, Diff-CBR works the best in each of the cases, whereas C2 CBR does approach it closely in performance compared to other methods. Given that C2 CBR seems to be brushing shoulders with Diff-CBR and applies the common technique of distance weighting, we decided to look at the number of common errors that they make to gather an insight as to how much influence the voting power function actually had. The number of common errors was a surprisingly high 88% on the average which indicates that the performance of C2 CBR was more due to the distance-weighting component than the voting power function. As the comparison between C1 and Simple-CBR shows, the voting power function is actually worse than the constant power function used by the latter. Although these results are clearly disheartening, we choose to examine the extent of the effect caused by the drawback of the C1 and C2 voting power functions as mentioned in an earlier section.

6.2 C3CBR

We choose to analyze C3 CBR separately as all others to follow use the same voting power function. We chose to use a 50% case base, and increasing values of radius. We present the accuracy chart as below.

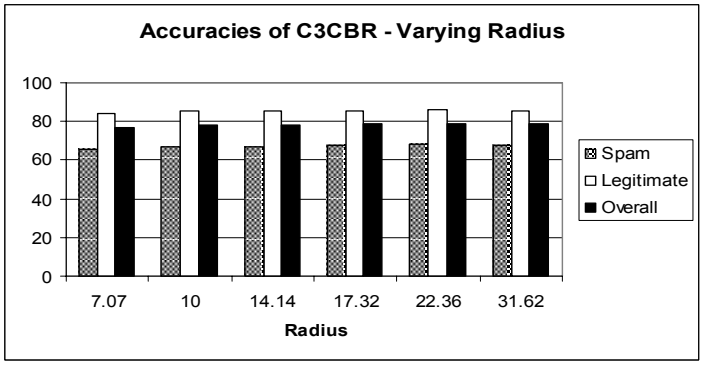


Figure 4. Accuracies of C3CBR, Varying Radius

As is evident from the results, the accuracies don't approach that of the conventional techniques such as Diff and Simple CBR. But one interesting thing worth mentioning in this context is that, although C3 CBR accuracies are (slightly) lesser than Diff-CBR, the fraction of common errors between C3 and Diff were 75% (compared to the figure of 88% for the C2-Diff pair) on the average. This gives us enough confidence that we are not searching in the dark and hence, we proceed to quantify the extent of the drawback discussed in the previous section.

6.3 Performance of Techniques that use BVPF

In our experiments with C4, C5 and C6 CBR, we were able to arrive at significantly better results (with varying values of radius). Even C4 CBR, the non-distance weighting BVPF CBR, gave much better accuracies compared to earlier techniques. C5 and C6 CBRs performed exceedingly well in comparison with others on spam precision. Although we do not include the charts of all the experiments that were conducted, we present a list of representative charts to show the performances of each of the techniques discussed so far so as to assert that BVPF works exceedingly well as a voting power function. All these were done with 50% of the corpus as the case base.

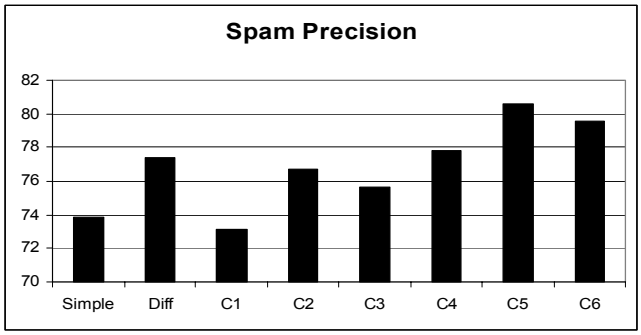


Figure 5. Spam Precision Chart

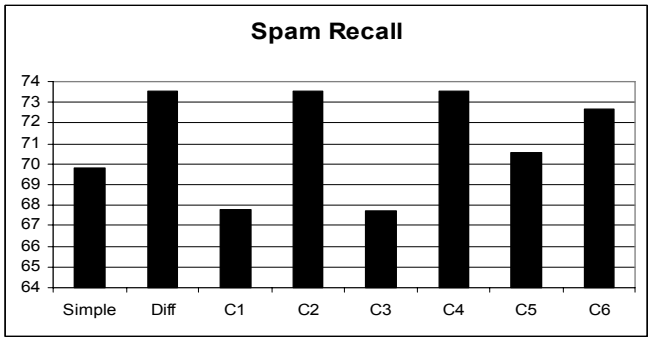


Figure 6. Spam Recall

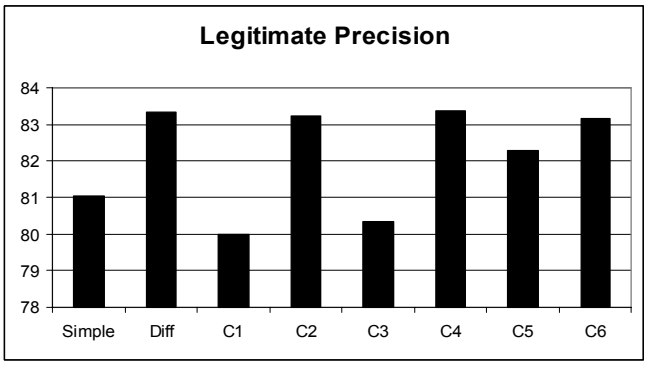


Figure 7. Legitimate Precision

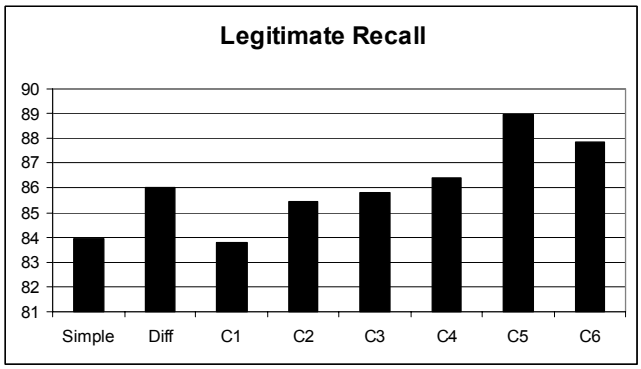


Figure 8. Legitimate Recall

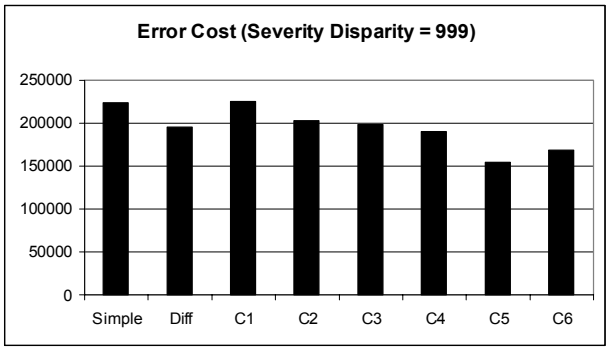


Figure 9. Error Cost Chart (Severity Disparity = 999)

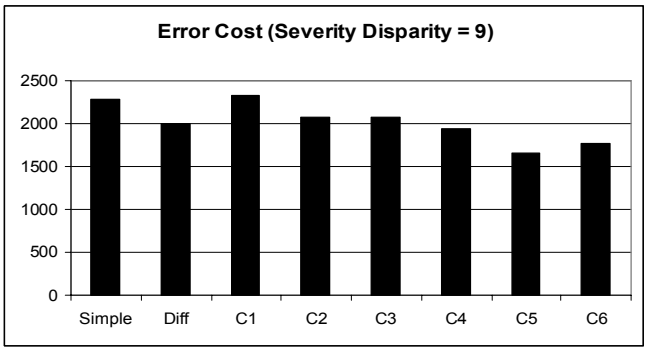


Figure 10. Error Cost Chart (Severity Disparity = 9)

As can be seen, all techniques that use BVPF, viz., C4, C5 and C6 CBRs work much better than the others on the parameters that count the most, i.e., spam precision, legitimate recall, and hence error costs. C6 CBR performs better than Diff on all measures except for Spam Recall where Diff-CBR works slightly better. C5 CBR gives the lowest error costs, and gives the highest Spam Precision and Legitimate Recall. On the whole, C5 and C6 techniques are way ahead of the conventional techniques. This proves our point that making use of the patterns in the case base is does improve performance very much.

7 Voting Powers and a Dynamic Case Base

Given that all our experiments have been on static case bases, it is reasonable enough to devote a section on how the computation of voting powers would be in a real-world scenario where cases get added and deleted from the case base. We provide a straightforward method to update the voting powers when a case gets added or deleted from the case base. We describe an algorithm to show how the BVPF powers can be updated on addition of a case base and omit other details as they would be a straightforward modification of the algorithm. We propose storing the <m_1, m_2, *BVPF* power> triplets for each case in the case base. The linear algorithm described below updates these triplets for relevant cases in the case base when a case gets added.

```
Update_On_Addition(NewCase n, Case Base C)
{
   m1 = m2 = 0;
   for(each case c in C)
   {
      if(distance(c,n) < radius)
      {
         if(label(c) == label(n))
         {
            increment m1;
            increment m1 of case n;
            re-compute BVPF of n;
```

```
            }
            else
            {
              increment m2;
              increment m2 of case n;
              re-compute BVPF of n;
            }
          }
      }
      Store <m1,m2,BVPF(m1+m2,m1,m2)> for the new case n;
}
```

8 Contributions and Future Work

We have, by means of this paper, provided approaches to make use of the patterns in the case base by means of associating each case with a voting power to improve spam filtering using CBR. This is, to the best of our knowledge, the first work on making use of the skewed distribution in the case base for a classification task. We have laid down the concerns on the design of a voting power function. Further, we have experimented exhaustively and made the implications of the voting power functions explicit.

As a part of future work in this regard, we propose to look deeper into the BVPF function and hostile cases to it. Further, as mentioned earlier, BVPF favors dense clusters over sparser ones. The implications of such a bias have to be examined in detail. BVPF is just our first approach in satisfying the desiderata for a voting power function and we have to look to find better variants of BVPF. Secondly, clustering is a data mining task which has been receiving a lot of attention of late. We would like to explore the feasibility of actually clustering the case base and making use of the clusters for the CBR classification task at hand. Further, we would like to look into other domains and test the applicability of BVPF and variants for classification tasks therein.

References

1. Androutsopoulos, Paliouras, Karkaletsis, Sakkis, Spyropoulos and Stamatopoulos, 2000, *Learning to filter spam e-mail: a comparison of a naive Bayesian and a memory-based approach*, PKDD Workshop on Machine Learning and Textual Information Access, 2000
2. Cunningham, Nowlan, Delany and Haahr, 2003, *A case-based approach to spam filtering that can track concept drift*, Proceedings of the ICCBR Workshop on Long-Lived CBR Systems, Norway, 2003

3. Sakkis, Androutsopoulos, Paliouras, Karkaletsis, Spyropoulos and Stamatopoulos, 2003, *A memory-based approach to anti-spam filtering for mailing lists*, Journal of Information Retrieval, Kluwer, 2003
4. Sahami, Dumais, Heckerman & Horvitz, 1998, *A bayesian approach to filtering junk e-mail*, AAAI-98 Workshop on Learning for Text Categorization, 1998
5. Ester, Kriegel, Sander, Xu, 1996, *A Density based algorithm for discovering clusters in large spatial databases with noise*, International Conference on Knowledge Discovery in Databases, KDD-1996
6. Hinneburg and Keim, 1998, *An efficient approach to clustering in large multimedia databases with noise*, International Conference on Knowledge Discovery in Databases, KDD-1998
7. Ankerst, Breunig, Kriegel and Sander, 1999, *OPTICS: Ordering Points to Identify the Clustering Structure*, Proceedings of the ACM SIGMOD Conference
8. Zhao, Karypis, "Criterion Function for Document Clustering: Experiments and Analysis", Department of Computer Science, University of Minnesota, TR#01-40
9. Drucker, H.D., Wu, D., Vapnik, V.: Support Vector Machines for spam categorization. IEEE Transaction on Neural Networks, Vol. 10 (5). 1999 1048-1054

Using Rough Set to Induce More Abstract rules from Rule Base

Feng Honghai [1, 2], Huang Yong [3], Zhao Shuo [3], Yang Bingru [3], Li Yueli [3]

[1]Urban & Rural Construction School, Hebei Agricultural University, 071001 Baoding, China
honghf@mail.hebau.edu.cn
[2] Information Engineering School, University of Science and Technology Beijing, 100083 Beijing, China
[3]Hebei Agricultural University, 071001 Baoding, China

Abstract. In fault diagnosis and medical diagnosis fields, often there is more than one fault or diseases that occur together. With the standard rough set method or other machine learning methods, the factors that cause a single fault to change to multi-faults cannot be induced. In order to obtain this kind of knowledge, the standard rough set based methods should be rebuilt. In this paper, we propose a decernibilty matrix based algorithm with which the cause of every single fault to change to multi-faults can be revealed. Additionally, we propose another rough set based algorithm to induce the common causes of all the single faults to change to their corresponding multi-faults, which is a process of knowledge discovery in rule base, i.e., not the usual database. Inducing more abstract rules in knowledge base is a very challenging problem that has not been resolved well.

1 Introduction

In fault diagnosis and medical diagnosis fields, often there is more than one fault or disease that occur together. With the multi-faults or multi-diseases, there may be more symptoms or the symptoms may be aggravated in comparison to that of the single fault or disease occurring, which is a kind of important knowledge for experts. But the standard rough set method only induces the rules for classification. The rules are simplified and generalized, namely there is no redundancy, and so the rules are not easier understood. Furthermore, with the standard rough set method, the cause of the formation of multi-faults cannot be induced. In order to obtain this kind of knowledge, the standard rough set based methods should be rebuilt.

Decernibility matrix is a valid method for attribute reduction and rule generation whose main idea is to compare the examples that are not in the same class. However, generally, the decernibility matrix is used to the whole data set [1], can it be used to partial data to induce the knowledge with which we can find the factors that cause multi-faults? The answer is affirmative. In this paper, we propose a decernibilty matrix based algorithm with which the cause of every single fault to change to multi-faults can be revealed.

Generating comparative knowledge is the advantage of rough set theory. Other machine learning theories such as SVM [2], ANN [3] and Bayesian networks cannot be used in this case, because these are all black-box ones.

Inducing more abstract rules from knowledge base is a very challenging problem that has not been resolved well. Yang Bingru has proposed an induction logic based schema for inducing the new knowledge [4]. Generally, as an inductive learning method, rough set theory can be used to discover knowledge in database, whereas has not been used to induce new knowledge in rule base. In this paper, we propose another rough set based algorithm to induce the common causes of all the single faults to change to their corresponding multi-faults, which is a process of knowledge discovery in knowledge base, i.e., not the usual database.

2 Basic Concepts of Rough Set

2.1 Indiscernibility Relation

In a decision table DT=$\langle U, A, V, f \rangle$, to every subset of attributes $B \subseteq A$, a binary relation, denoted by $IND(B)$, called the B-indiscernibility relation, is associated and defined as follows:

$$IND(B) = \left\{ \begin{array}{l} (x_i, x_j) \in U \times U : a \in B, f(x_i, a) \\ = f(x_j, a) \end{array} \right\} \qquad (1)$$

2.2

The rough sets approach to data analysis hinges on two basic concepts, namely the lower and the upper approximations of a set, referring to:

-The elements that doubtlessly belong to the set, and

-The elements that possibly belong to the set.

Let X denotes the subset of elements of the universe U ($X \subset U$). The lower approximation of X in B ($B \subseteq A$), denoted as $IND(B)$, is defined as the union of all these elementary sets that are contained in X . More formally:

$$POS_B(X) = B_-(X) = \{x_i \in U \left| [x_i]_{IND(B)} \subset X \right.\}$$

The upper approximation of the set X , denoted as $B^-(X)$, is the union of these elementary sets, which have a non-empty intersection with X :

$$B^-(X) = \{x_i \in U \left| [x_i]_{IND(B)} \bigcap X \neq 0 \right.\}$$

The difference:

$$\mathrm{BN}(X) = B^-(X) - POS_B(X)$$

is called a boundary of X in U .

2.3 Discernibility Matrix

Discernibility matrix of DT, denoted $M_{\mathrm{DT}}(m_{ij})$ a $n \times n$ matrix is defined as

$$m_{ij} = \left\{ \begin{array}{c} \left\{ a \in C : f(x_i) \neq f(x_j, a) \wedge (d \in D, f(x_j, d) \neq f(x_i, d)) \right\} \\ 0, d \in D, f(x_i, d) = f(x_j, d) \\ \phi, f(x_i, a) = f(x_j, d) \wedge (d \in D, f(x_i, d) \neq f(x_j, d)) \end{array} \right\} \quad (2)$$

Where $i, j = 1, 2 \cdots, n$

Thus entry m_{ij} is the set of all attributes that classify objects x_i and x_j into different decision classes in U . From formula (2), all of the distinguishing information for attributes is contained in above discernibility matrix.

3 Algorithm

3.1 Algorithm for Deriving the Factors that Make the Single Fault to Multi-faults

(1) Select the examples with multi-faults and the examples with a single fault that will coexist with other faults.

(2) Use discernibility matrix method to compare the single fault examples and the coexistent faults examples and induce the rules for multi-faults.

3.2 Algorithm for Inducing More Abstract Rules in Rule Base

(1) Select the rules with which we want to get more abstract rules.
(2) Transform the rules into data, and get a decision table.
(3) Use the concept of positive region to generate more abstract rules.

4 Fault diagnosis in power systems

4.1 Fault Diagnosis Decision System of Power Systems

Table 1 is a fault diagnosis decision system of power systems. The states of relays and circuits breakers are used as the attributes describing the related faults examples. In Table 1, there are 19 condition attributes and 16 fault classes. Table 1 consists of 26 faults cases, including 5 single faults and 10 double faults with no failure relay and breaker; 5 single faults with one failure breaker; 5 single faults with one failure breaker; 1 special case- "NO" represents no fault occurred.

The system consists of 5 components, 15 protective relays and 4 circuit breakers. The 5 components are bus bars A, B, C and lines L_1, L_2. The circuit's breakers are denoted by CB_1-CB_4. The 15 protective relays are Am, Bm, Cm, L_1Am, L_1Bm, L_2Bm, L_2Cm, L_1Ap, L_1Bp, L_2Bp, L_2Cp, L_1As, L_1Bs, L_2Bs and L_2Cs.

In Table 1, "1" means that breakers open or relays operate, "0" means that breakers close or relays do not operate, A, B, $\cdots$, L_2 denotes the single fault occurred on bus bar A, B, $\cdots$, line L_2 respectively, AB, AL_1, $\cdots$, denote the double faults on bus bar A and B, bus bar A and line L_1, $\cdots$, respectively.

Table 1 system fault cases and associated alarm patterns

U	CB_1	CB_2	CB_3	CB_4	Am	Bm	Cm	L_1Am	L_1Bm	L_2Bm	L_2Cm	L_1Ap	L_1Bp	L_2Bp	L_2Cp	L_1As	L_1Bs	L_2Bs	L_2Cs	Fault
1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A
2	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	B
3	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	C
4	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	L_1
5	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	L_2
6	1	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	L_1
7	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	L_1
8	0	0	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	L_2
9	0	0	1	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	L_2
10	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	B
11	1	0	0	1	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0	L_2
12	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	A
13	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	A
14	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	C
15	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	B
16	1	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	L_1
17	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	AB
18	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	AC
19	1	1	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	AL_1
20	1	0	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	AL_2
21	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	BC
22	1	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	BL_1
23	0	1	1	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	BL_2
24	1	1	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	CL_1
25	0	0	1	1	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	CL_2
26	1	1	1	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	L_1L_2
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NO

4.2 Induce the Factors that Cause Multi-faults

(1) For examples 1, 12 and 13, the fault is A, and for example 17, 18, 19 and 20, the faults include A. Which are the factors that cause the multi-faults including A? Now, with the partial decernibility matrix we can find the factors.

Firstly, the examples with faults A, AB, AC, AL_1 and AL_2 are selected. Secondly, generate the decernibility matrix with these examples. Thirdly, use the decernibility matrix to extract the factors that cause multi-faults including A.

Table 2 is the information table for inducing the causes of the formation of multi-faults that include A. Table 3 gives the results of the decernibility matrix.

The factors that cause fault A to change to multi-fault AB can be induced as follows (see example 17 in Table 2):

$(CB_2 \vee CB_3 \vee Bm) \wedge (CB_1 \vee CB_3 \vee Bm \vee L_1Bs) \wedge (CB_1 \vee CB_3 \vee Am \vee Bm \vee L_1Bs) = CB_3 \vee Bm$

The factors that cause fault A to change to multi-fault AB are $CB_3 \vee Bm$, in other words, CB_3=1 or Bm=1 are the facors that cause fault A to change to fault AB.

Table 2 The information table for inducing the causes of multi-faults that include A

CB_1	CB_2	CB_3	CB_4	Am	Bm	Cm	L_1Am	L_1Bm	L_2Bm	L_2Cm	L_1Ap	L_1Bp	L_2Bp	L_2Cp	L_1As	L_1Bs	L_2Bs	L_2Cs	Fault
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A
0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0		A
1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0		AB
1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0		AC
1	1	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0		AL_1
1	0	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0		AL_2

Table 3 Decernibility matrix for the examples with faults A, AB, AC, AL_1 and AL_2

	1	12	13
17(AB)	CB_2 CB_3 Bm	CB_1 CB_3 Bm L_1Bs	CB_1 CB_3 Am Bm L_1Bs
18 (AC)	CB_4 Cm	CB_1 CB_2 CB_4 Cm L_1Bs	CB_1 CB_2 CB_4 Am Cm L_1Bs
19(AL_1)	CB_2 L_1Am L_1Bm	CB_1 L_1Am L_1Bm L_1Bs	CB_1 Am L_1Am L_1Bm L_1Bs
20(AL_2)	CB_3 CB_4 L_2Bm L_2Cm	CB_1 CB_2 CB_3 CB_4 L_2Bm L_2Cm L_1Bs	CB_1 CB_2 CB_3 CB_4 Am L_2Bm L_2Cm L_1Bs

The factors that cause fault A to change to multi-fault AC can be induced as follows (see example 18 in Table 2):

$(CB_4 \vee Cm) \wedge (CB_1 \vee CB_2 \vee CB_4 \vee Cm \vee L_1Bs) \wedge (CB_1 \vee CB_2 \vee CB_4 \vee Am \vee Cm \vee L_1Bs) = CB_4 \vee Cm$

So we can conclude that:

The factors that cause fault A change to multi-fault AC are CB4 $\vee$Cm.

Similarly, we can induce that L_1Am =1 or L_1Bm =1 are the factors that cause fault A to change to fault AL_1 (see example 19 in Table 2). And CB_3=1, CB_4=1, L_2Bm=1 or L_2Cm=1 are the factors that cause fault A to change to fault AL_2 (see example 20 in Table 2).

To sum up, the factors that cause faults A, B, C, L_1 and L_2 to change into corresponding multi-faults are as follows:

(1) $CB_3 \vee Bm \rightarrow (A \Rightarrow AB)$
(2) $CB_4 \vee Cm \rightarrow (A \Rightarrow AC)$
(3) $L_1Am \vee L_1 Bm \rightarrow (A \Rightarrow A\ L_1)$
(4) $CB_3 \vee CB_4 \vee L_2Bm \vee L_2 Cm \rightarrow (A \Rightarrow A\ L_2)$
(5) $CB_1 \vee Am \rightarrow (C \Rightarrow AC)$
(6) $CB_2 \vee CB_3 \vee Bm \rightarrow (C \Rightarrow BC)$
(7) $CB_1 \vee CB_2 \vee L_1Am \vee L_1 Bm \rightarrow (C \Rightarrow CL_1)$
(8) $CB_3 \vee L_2Bm \vee L_2 Cm \rightarrow (C \Rightarrow C\ L_2)$
(9) $Am \rightarrow (L_2 \Rightarrow A\ L_2)$
(10) $CB_2 \vee Bm \rightarrow (L_2 \Rightarrow B\ L_2)$
(13) $Am \rightarrow (B \Rightarrow AB)$
(14) $Cm \rightarrow (B \Rightarrow BC)$
(15) $L_1Am \vee L_1 Bm \rightarrow (B \Rightarrow B\ L_1)$
(16) $L_2Bm \vee L_2 Cm \rightarrow (B \Rightarrow B\ L_2)$
(17) $Am \rightarrow (L_1 \Rightarrow A\ L_1)$
(18) $CB_3 \vee Bm \rightarrow (L_1 \Rightarrow B\ L_1)$
(19) $CB_4 \vee Cm \rightarrow (L_1 \Rightarrow C\ L_1)$
(20) $CB_3 \vee L_2Bm \vee L_2 Cm \rightarrow (L_1 \Rightarrow L_1\ L_2)$

(11) Cm $\rightarrow (L_2 \Rightarrow C\ L_2)$
(12) $CB_2 \vee L_1Am \vee L_1\ Bm \rightarrow (L_2 \Rightarrow L_1\ L_2)$

4.2 Induce More Abstract Rules from the above Rules

From rule (1) mentioned above, we can conclude that: if CB_3=1 or Bm=1 the fault B is added, and from rule (11), if Cm=1 the fault C is added etc. So we can get the following table that can be regarded as the varietal form of information table of rough set.

Table 4 Using information table to represent the rules

U	CB_1	CB_2	CB_3	CB_4	Am	Bm	Cm	L_1Am	L_1Bm	L_2Bm	L_2Cm	A^+	B^+	C^+	L^+_1	L^+_2
1			1			1						1				
2				1			1						1			
3								1	1					1		
4			1	1						1	1				1	
5	1				1							1				
6		1	1			1							1			
7	1	1						1	1					1		
8			1							1	1				1	
9					1							1				
10		1				1							1			
11							1							1		
12		1						1	1					1		
13					1							1				
14							1						1			
15								1	1					1		
16										1	1				1	
17					1							1				
18			1			1							1			
19				1			1						1			
20			1							1	1				1	

In Table 4, U is the set of rules. CB_1, CB_2, CB_3, CB_4, Am, Bm, Cm, L_1Am, L_1Bm, L_2Bm and L_2Cm are the condition attributes whose values only take 1 which means that breakers open or relays operate. A^+, B^+, C^+, L^+_1 and L^+_2 are decision attributes whose values only take 1 that means that the fault has been added. The condition attributes values and decision attributes values are only take the value 1 is because that we only induce the causality between breakers openning and faults or between relays operating and faults.

With rough set theory, we can conclude that:

$IND(Am) =$ $IND(A) =$ $POS_{Am}(A) =$ {5,9,13,17},

$IND(Bm) =$ $IND(B) =$ $POS_{Bm}(B) =$ {1,6,10,18},

$IND(Cm) =$ $IND(C) =$ $POS_{Cm}(C) =$ {2,11,14,19},

$IND(L_1Am) = IND(L_1Bm) = IND(L_1) = POS_{L_1Am}(L_1) = POS_{Bm}(L_1) =$ {3, 7,12,15},

$IND(L_2Bm) = IND(L_2Cm) = IND(L_2) = POS_{L_2Bm}(L_2) = POS_{L_2Cm}(L_2) =$ {4,8,16,20}

Whereas $IND(CB_1)$, $IND(CB_2)$, $IND(CB_3)$ and $IND(CB_4)$ cannot be a positive region of anyone of A^+, B^+, C^+, L^+_1 and L^+_2.

So the following more abstract rules can be induced:

(21) Am is the common factor that causes fault B to change into BA, C to change into CA, L_1 to change into L_1A and L_2 to change into L_2A (see rules (13), (5), (17) and (9)).

(22) Bm is the common factor that causes fault A to change into AB, C to change into CB, L_1 to change into L_1B and L_2 to change into L_2B (see rules (1), (6), (18) and (10)).

(23) Cm is the common factor that causes faults A to change into AC, B to change into BC, L_1 to change into L_1C and L_2 to change into L_2C (see rules (2), (14), (19) and (11)).

(24) L_1Am or L_1Bm are the common factors that cause fault A to change into AL_1, B to change into BL_1, C to change into CL_1 and L_2 to change into L_1L_2 (see rules (3), (15), (7) and (12)).

(25) L_2Bm or L_2Cm are the common factors that cause faults A to change into AL_2, B to change into BL_2, C to change into CL_2 and L_1 to change into L_1L_2 (see rules (4), (16), (8) and (20)).

5 Discussions

(1) The idea of decernibility matrix can be used to not only the whole data set but also the partial data to generate special knowledge.

(2) Obviously, rules (21)-(25) are more abstract than the rules (1)-(20), which means that the rough set theory can be used to induce more abstract knowledge in knowledge base.

(3) For inducing easily understood knowledge, the rough set theory has an advantage over the black-box based machine learning methods such as ANN, SVM etc.

(4) These kinds of knowledge help experts understand the correlation of rules or of attribute values better and more clearly.

References

1. Pawlak Z. Rough sets. Int. J. of Computer and Information Science. 1l (1982): 341-356
2. V.N. Vapnik. Statistical Learning Theory. John Wiley & Sons, (1998).
3. S.B.Cho."Pattern recognition with neural networks combined by genetic algorithm", Fuzzy sets and systems 103 (1999) 339-347

4. Yang Bingru, Shen Jiangtao and Chen Hongjie. Recearch on the Structure Model and mining Algorithm for Knowledge Discovery Based on Knowledge Base (KDK). Engineering Science (Chinese) 5 (2003): 49-53

Pen-Based Retrieval in Handwritten Documents

Sascha Schimke[1], Claus Vielhauer[1]

[1] ITI Research Group on Multimedia and Security,
Universitätsplatz 2, 39106 Magdeburg, Germany
{sascha.schimke, claus.vielhauer}@iti.cs.uni-magdeburg.de
http://wwwiti.cs.uni-magdeburg.de/iti_amsl/

Abstract. This paper describes techniques for searching in handwritten and handdrawn documents. We assume that more and more handwritten documents will accrue, as pen-based computers like PDA or TabletPC get increasingly popular. In order to manage the anticipated amount of handwritten documents, powerful abilities for searching within the documents are needed. The techniques, which we propose, are well researched in the field of bioinformatics, where they are used for finding parts within sequences of amino acids or genes. This paper documents a work in progress research activity.

1 Introduction

While pen-based computers like PDA or TabletPC get popular, a large amount of handwritten documents will be produced. Often it's easier and less disturbing in a meeting to minute using a pen instead of typing with a keyboard. And sometimes for rapid drawing or sketching the pen is the only possible opportunity. To handle and manage a large amount of such documents, consisting of handwritten and handdrawn parts, the ability for retrieval is absolutely necessary.

By document retrieval, we mean the following: having a set D of documents $\{d_1, d_2, d_3, \ldots, d_n\}$ and using a query word q, the result of the retrieval is a list D' of documents $\{d'_1, d'_2, \ldots, d'_m\}$ out of D, where the retrieval query q is contained one or more times within the documents of D'. For each document of D', even the position(s) of the occurrence(s) of the query q is available. In the case of pen-based retrieval in handwritten and –drawn documents, the query q as well as the elements of D is a result of writing or drawing process. (See Fig. 1)

From the users' point of view, the retrieval task can be performed in two manners: a) by writing or drawing the query or b) by manually selecting one occurrence of the query within a document.

The handwritten documents in our work are no scanned images of sheets of paper, as used in the field of off-line handwriting processing. We confine ourselves to the field of so called on-line data. This means, that the pen-movement data are acquired directly during the writing process, using special hardware. The resulting data are sequences of sampled pen tip positions and pressure values: $x(t)$, $y(t)$, $p(t)$. From these directly measurable data, further data can be derived, e.g. velocities in direction of x- or y-axis or the track velocity ($v_x(t)$, $v_y(t)$, $v(t)$).

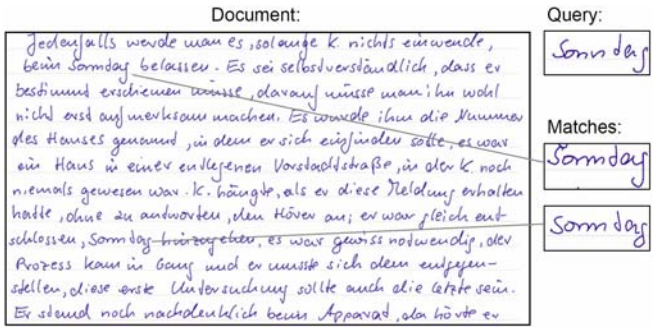


Fig. 1. Illustration of the search process for a query word "Sonntag" and two marked matches in a text from Kafka ("Der Prozeß")

The most intuitive idea would be to perform a textual recognition for the documents of D and for the query q. Using the resulting textual data, a simple string search function could be used, as is available in every word processor. The problem is that the textual recognition is never absolute perfect and therefore a simple search would fail in most cases. Another problem with using textual recognition is the case, where no text at all exists to be recognized, namely when using handdrawn images or sketches instead of words. To solve these problems, we try to use a kind of *direct handwriting matching* instead of textual recognition, as described in section 3.

2 Related Works

A large amount of related work has been done before by other researchers. But most of them had different approaches or different goals for their pen-based retrieval.

Srihari et al. presented a search engine for handwritten documents [8]. Contrary to our approach, their documents were acquired off-line. Due to the nature of off-line handwriting data, their processing steps are more complex then ours. A similar approach is for example presented in [3] for historical documents.

In [5] Landay and Davis describe a set of experiments for shared note taking. The participants of these experiments used PDAs and paper-based digitizer devices to write notes and other documents. Later, management of the accumulated documents should be possible, for example browsing and searching.

Besides this text oriented retrieval in pen based data, there are several approaches for pen aided retrieval in image databases. In [7] a system is described, which extracts shapes from images and makes a comparison of these shapes with pen drawn query inputs. A similar idea is described in [1]. Here the pen is used for retrieval in ClipArt data, i.e. vector graphic files.

An approach, which is slightly similar to our work, was described in [9]. The goal of the authors is the same as ours – to search in on-line captured handwritten documents – but they use different features and different matching algorithms. We expect to reduce the computational complexity of their system by using our own approach.

3 On-line Comparison for Retrieval

Our idea is to use an algorithm for finding substring in noisy data. Similar algorithms are used in genetics and bioinformatics for example to find sequences of amino acids or genes. The problem is that often there is no 100%-match between the query sequence and a part of the complete sequence. Instead, only a certain similarity can be ascertained.

3.1 Approximate String Searching

The goal of *approximate string searching* is to find those substrings (approximate matches) $m_1, m_2, \ldots, m_n$ of a document d, so that the distance of these substrings and a query string q is smaller than a threshold τ. Here the distance is defined as the *edit distance* [4]. The approximate string searching problem is to find those positions of d, where the approximate matches $m_1, m_2, \ldots, m_n$ end.
May the query string q consist of k characters $q[1], q[2], q[3], \ldots, q[k]$. Furthermore, may the document d consist of l characters $d[1], d[2], d[3], \ldots, d[l]$. The approximate string searching problem is equal to find those values j, so that $D(k, j)$ is smaller than the threshold τ:

$$D(i,j) = \begin{cases} 0 & \text{if } i = 0, \\ D(i-1,0) + 1 & \text{if } i > 0 \text{ and } j = 0, \\ \min \begin{cases} D(i,j-1) + 1 \\ D(i-1,j) + 1 \\ D(i-1,j-1) + \delta(i,j) \end{cases} & \text{else.} \end{cases}$$

with $0 \le i \le k$ and $0 \le j \le l$. The character-wise distance δ is defined as follows:

$$\delta(i,j) = \begin{cases} 0 & \text{if } q[i] = d[j], \\ 1 & \text{else.} \end{cases}$$

The asymptotic computational complexity of this recursion D is $O(k \cdot l)$, but the required memory can be reduced to $O(k)$ by smart implementations.

3.2 Textual Recognition

For retrieval in handwritten documents, we interpret each document as a long sequence of elements. These sequences are the basis for estimation of local similarity regarding a (shorter) query sequence q. One problem is to define the elements of the sequences, i.e. the alphabet. Using text recognition algorithms, the alphabet would be the set of characters and signs, which are the output of the algorithm and which form the words, sentences and the whole text. The similarity search is used to overcome the problem of misrecognised characters.

To limit the computational complexity, which is implied by using text recognition algorithms, in our first experiments we use more rudimental features from handwritten documents. Two kinds of features are described in the following two subsections.

3.3 Stroke Direction Features

In [2] a method is described for converting a line drawing into a sequence of stroke directions. The idea of Freeman is to have a very short and compressible description of line drawings, but we try to adapt his method for substring search in handwritten documents.

The background of Freeman's approach is to code a line drawing using a chain of strokes with a certain direction, as can be seen in Fig. 2. Following the quantized strokes and coding every direction with a certain symbol, we obtain a chain of direction codes. The granularity of the resulting code string is influenced by the size of the quantizing grid.

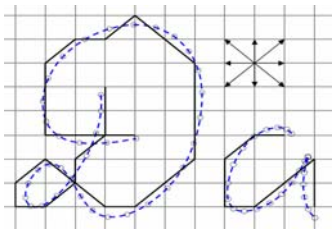


Fig. 2. A writing sample (dashed line) and the corresponding sequence of stroke directions (solid line), using a square grid quantization

3.4 Biometric Features

Another idea of coding handwritten inputs as strings is presented in [6]. Here, from the handwriting signals ($x(t)$, $y(t)$, $p(t)$, $v_x(t)$, $v_y(t)$, $v(t)$) the local extrema are extracted, i.e. points of minimal or maximal x-/y-value, pressure or velocity. That way for a given pen-based input a string is created, consisting of successive local extrema of the signals. The original purpose of this approach was biometric authentication using handwritten signatures.

3.5 Fusion of Single-Feature Results

Having different searching results for a query q within a document d by using different base features (and/or different parameters), derived from the original pen-based input, it is necessary to make a kind of fusion of these different results.

For example, by using the direction coded string (3.3) we could get two matches m_1 and m_2 (positions in the original document d). By using the minima and maxima for creating a string (3.4) to describe the pen movement we could get three matches m_3, m_4 and m_5, where m_2 and m_5 are identical. Two trivial solutions to handle this situation are the conjunction and the disjunction. In conjunction, only those matches, which are identical for both kinds of feature strings, are taken as final matching result. Disjunction means, that all the matches are taken as the result. The choice of conjunction or disjunction affects the recall and precision rates of the retrieval. Using

the disjunction approach, the recall will presumably increase, but in the same moment, the precision will decline.

4 Test Settings

For testing of our system, we collect handwriting data using TabletPCs and special pens. In our case the formers ones have a spatial resolution of 1,000 points/cm, temporal resolution of 100 Hz and can distinguish 1,024 degrees of pressure. The latter devices are pens, which are equipped with an optical sensor to read a special pattern at paper, which allows them to get precise position information [10]. The spatial resolution is about 280 points/cm, sampling rate is up to 50 Hz and 128 degrees of pressure can be distinguished. The documents, acquired with these devices, will be used to obtain recall and precision results for our handwriting retrieval system.

Acknowledgements

This work has been partly supported by the EU Network of Excellence SIMILAR (FP6−507609). The contents of this publication are the sole responsibility of the authors and can in no way be taken to reflect the views of the European Union.

References

1. M. Fonseca, B. Barroso, P. Ribeiro, J. Jorge, “Sketch-based retrieval of ClipArt drawings”, *Proceedings of the working conference on Advanced visual interfaces*, 2004, pp. 429–432.
2. H. Freeman, “Computer Processing of Line-Drawing Images”, *Computer Surveys*, Vol. 6, No. 1, March 1974, pp. 57–97.
3. V. Govindaraju, H. Xue, “Fast Handwriting Recognition for Indexing Historical Documents”, *Proceedings of First International Workshop on Document Image Analysis for Libraries (DIAL'04)*, 2004, pp. 314–320.
4. D. Gusfield, *Algorithms on Strings, Trees, and Sequences*, Cambridge University Press, ISBN 0-521585-19-8, 1997.
5. J. A. Landay, R. C. Davis, “Making sharing pervasive: Ubiquitous computing for shared note taking”, *IBM Systems Journal*, Vol. 38, No. 4, 1999, pp. 531–550.
6. S. Schimke, C. Vielhauer, J. Dittmann, “Using Adapted Levenshtein Distance for On-Line Signature Authentication”, *Proceedings of the 17th International Conference on Pattern Recognition*, 2004.
7. L. Schomaker, E. de Leau, L. Vuurpijl, “Using Pen-Based Outlines for Object-Based Annotation and Image-Based Queries”, *Lecture Notes in Computer Science*, Vol. 1614, 1999, pp. 585–592.
8. S. Srihari, C. Huang, H. Srinivasan, “A Search Engine for Handwritten Documents”, *Proceedings of SPIE-IS&T Electronic Imaging*, 2005, pp. 66–75.
9. K. Sun, J. Wang, “Similarity based matching method for handwriting retrieval”, *Proceedings of SPIE 2003 – Document Recognition and Retrieval X*, 2003, pp. 156–163.
10. Anoto Group AB, http://www.anotofunctionality.com/, 2005.

The Creation of KANSEI-Vocabulary Scale by Shape

Sunkyoung Baek[1], Kwangpil Ko[2], Hyein Jeong[3], Namgeun Lee[4], Sicheon You[5], Pankoo Kim[1,*]

[1] Dept. of Computer Science, Chosun University, Gwangju 501-759 Korea
{zamilla100, pkkim}@chosun.ac.kr
[2] Dept. of Design, Chungang University, Seoul 155-756 Korea
kkp0825@korea.com
[3] Dept. of English Language and Literature, Chosun University, Korea
hyeinjeong@chosun.ac.kr
[4] Dept. of English Education, Chosun University, Korea
nglee@chosun.ac.kr
[5] Dept. of Design, Chosun University, Korea
scyou@chosun.ac.kr

Abstract. In the oncoming generation of computing, the demand for tools of information retrieval for human sensibilities or tastes and of sensibility recognition and extraction has been increasing rapidly. The study of KANSEI-based image retrieval tools has especially come in the spotlight. Colors, meaningful similarities among colors, and sensibilities related to color have been the main themes in most of such studies. Those studies are based on the retrieval of lower-level visual information, for example, color, shape and texture. However, retrieval of such lower-level information has difficulty catching higher-level information such as intention or sensibility of users. To solve this problem, first we suggest a KANSEI-Vocabulary Scale by associating human sensibilities with shapes. The final goal in the future of our study is the creation of a KANSEI-Ontology based on visual information such as color, shape, texture, and pattern, to form a new KANSEI-Information system that can understand, retrieve, and recognize human beings' sensibilities and for an ontological system based on visual information and KANSEI-Vocabulary. For the purpose this study develops a vocabulary scale with shapes and sensibilities which is as one of a series of studies, and the scale will be part of the basis of intelligent image retrieval techniques depending on the user's intention and sensibility.

1 Introduction

Recently the demand for image retrieval and recognizable extraction corresponding to sensibility has been increasing, and the studies focused on establishing those KANSEI-based systems have been progressing more than ever. KANSEI in Japanese means by sensibility that is to sense, recall, desire and think of the beauty in objects [1]. In addition, the attempt to understand, measure and evaluate, and apply KANSEI to situational design or products will be required more and more in the future. How-

* Corresponding author.

ever, it is not a simple question. Because, in general, most data in computing processes are conducted on a binary scale over objectivity, monotony, universality, and reproducibility, while the data used in sensibility-based computing processes are conducted over subjectivity, polysemy, ambiguity, and situational dependence. So far most KANSEI-based experiments have produced specific results. On the basis of the data we prepare the ground for a KANSEI-based information system capable of understanding, retrieval, and recognizing human sensibilities and propose a study to apply to the area of intelligent human computing.

In experimentation with human sensibilities, the area of KANSEI-based image retrieval system has been limited to content-based using visual features such as texture, shape, pattern, and especially color, which are the most popular sources for experiment, or feature-based such as those using recognition system, but such retrievals have had some trouble in checking and recognizing images appropriate for the user's purposes or tastes in higher meaning.

In order to cope with these limiting barriers, we have attempted to associate visual information with human beings' sensibilities through a relational sample scale, which is made by linking the visual information (color, shape, texture, and pattern) with the KANSEI-Vocabulary of human beings'. First, for the scale we collected and classified the most common shapes and defined what the most standard shapes are. On the other hand, we found a relationship between shape and KANSEI-Vocabulary. As a result we were able to produce a scale for shapes and the related KANSEI-Vocabulary. We believe that such a result will allow us to realize a KANSEI-based information system and will be helpful for ontological construction based on visual information and KANSEI-Vocabulary.

2 Background and Related Works

2.1 KANSEI and Vocabulary

KANSEI of human beings is a psychological state based on the five senses, which also has distinctive differences between sensitivity and experience of each individual. To specify this in detail, sensitivity is indicated as pleasure, sorrow, anger, happiness, love and hate. It is knowledge used by everyone. KANSEI is expressed usually with emotional words for example, beautiful, romantic, fantastic, comfortable etc [2]. On the other hand, KANSEI is knowledge based on individual experience and differentiates each person. In other words, sensitivity is a good feeling of health and the best conditions, but the feeling of looking out the window and realizing it is showing is different in individual experience and knowledge [3]. KANSEI is a reflex and intuitive reaction. However, it includes many inconstant characteristics that make it difficult to be objective and typical. Therefore we use natural language for the representation of KANSEI because it is includes the image structure of human ideas that we cannot observe. In the natural language, it is represented by the adjective.

2.2 Visual Information and KANSEI-Vocabulary

The image is related to human beings' KANSEI in visual information because we get feelings in the same visual information, through knowledge and experience.

It is color that most researchers use as an experimental material and have been focusing on. Color can be used as a communication tool. Color imagery is also being treated as an objective method due to symbol. In a research center for color design in Japan, there have been ongoing researches on the relationship between color image and color scale with a database of coloration for the answers to the questions, "How many adjectives are needed to deliver someone's entire picture to others freely?", "How can the coloration indigenous to the implication of each adjective be expressed?", "What is the minimum number of adjectives that can be used?".

Kobayashi researched the relation between color and language in color image standardization, and Haruyoshi stated that many colors included images in a questionnaire in Japan [4] [5]. Color Wheel Pro explains the meaning of "The basic colors including a red and local color". Hewlett-Packard defines "The meaning of colors" in the USA [6]. In the Republic of Korea, IRI developed the IRI adjective image scale at a visibility and symbol of Koreans' sensibility [7]. Recently there has been research in how to change the meaning of Color-KANSEI into adjective words.

2.3 The Other Research based KANSEI

Mitsuteru Kokubun shows the techniques of collecting, estimating the importance of, and arranging keywords with a KANSEI-Vocabulary networking for a system for visualizing individual KANSEI-Information [8]. In Tagaki, to research KANSEI-based image he shows an image retrieval using sensibilities and a mapping structure between image and KANSEI-Scale with a nerve network and gene algorithm [9].

In relation to expressing visual information, terminology dictionaries such as a thesaurus were used to develop a rewriting method, query expanding, and relevance feedback [10]. Lexical ontology such as WordNet, similar to a thesaurus, was also used for matching visual information with sensibility lexical items.

3 The Theoretical Bases for This Research

3.1 Definition of Form

Shape means a plane two-dimensional space made by lines and indicates either a 'silhouette' or 'outline'. It is formed with both external angles and a frame axis, and its feelings are expressed at seeing its inner-shape. Shape has the dimensions of length and width by definition, but not of depth.

Form is the shape of a thing, its look and bearing, or a body and obtained figure, and is a unity, unified wholeness, or organization which creates a partial order for the

entire body of a thing. Form is a three-dimensional trace of the wheels in a dynamic definition. A form is a final shape made by points and lines; a shape is an original feature of a form, and an appearance is a thing made by elaborated combination of the surfaces of the form and its angles. In other words, form means the volume, the three-dimensional mass, or the outline of a thing that can be caught visually. It is also proposed in philosophy that it is the outward pattern of a given thing in a substantial nature.

3.2 Geometrical Form

Geometry is defined as 'the science of dealing with the size and shape of a thing' or 'an area of mathematics limited to mathematical features of space'.

In Rudolf Archaism, a geometrical form is a natural metaphoric or refined form created by ideological thoughts of human beings [11]. This form can be changed into a circle, triangle, square, etc., each of which can be computed mathematically with a ruler and a compass. It can also be called an artificial abstract form changed simply from a complex nature. In other words, this geometrical form gives us systematic, simple and plain feelings, but it originates from nature. Therefore it is occasionally considered to be a concept against natural form.

The precise geometrical forms were made with a method which has a mathematical or physical structure. The method became a standard of measuring beauty, which is now used in related researches and help make sense of space intuitively. Although these forms seem to be a bit complex, they can be recognized and reproduced in the same way as the original form. It might not be able to objectively explain the whole contents of the original forms, but it can comprehend their implications in the forms, because the simplicity and the rationality of geometrical form are strict rules in themselves.

3.3 Changing Elements of Form

In general, design consists of conceptual elements, visual elements, relational elements, and constructional elements [12].

Table 1. Elements of Form

Elements of form	**Constituent elements**
Conceptual Elements	Point, Line, Plane, Volume
Visual Elements	**Shape**, Size, Color, Texture
Relational Elements	Position, Direction, Spatiality, Gravity
Constructional Elements	Vertex, Edge, Face

Conceptual elements are basic to form, and invisible in the case of a whole form, while visual elements are both conceptual and sensible. Relational elements support inner interrelations. Constructional elements realize conceptual elements. In other

words, they are constructed with conceptual elements by the relational elements. For this relationship the style of a form can be realized only if the elements of the design are satisfied. We used “shape” as constituent element among visual elements of form. As from this section, we named the using of form “shape”.

4 The Experimental Procedures

4.1 The Model of Space Formation of Shape

P. Klee and W. Kandinsky suggested that line starts from point and that the point indicates the core and the start, which means the root of all the shapes and motions. As point is a principle element of image and an image starts with it, it develops into line, plane, and then body (three-dimensional shape). The various forms in the image scales are divided into soft and hard shapes, and cool and warm shapes.

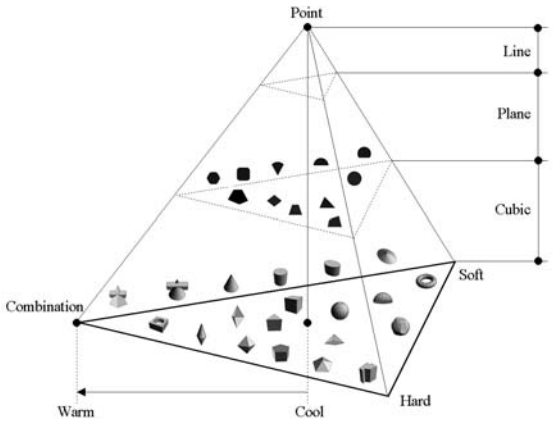


Fig. 1 Space Formation of Shape

4.2 The Classification of Geometrical Shape

In visual information there are three shapes, conceptual, geometrical, and natural shapes. Geometrical shape means an artificial abstract shape transformed from a natural shape through simplicity. The geometrical and natural shapes are included in organic shape but natural shapes have no defined standard. Therefore all the shapes can be classified on the basis of geometrical shape, not natural shape. A shape is created with planes, that is, a shape and a form are created with planes. To show the forma-

tion of shape and form, the planes and the lines of shape are put into the main elements in the first experiments of our study.

Table 2. Classification of Shape

		Soft	Hard	Combination
Plane	Cool	Circle	Triangle Equilateral Triangle Isosceles Triangle Isosceles Right Triangle Obtuse Triangle Quadrangle Square Rectangle Rhombus Parallelogram Trapezoid	Segment of a Circle Half Circle Sector Rounded Triangle Rounded Rectangle
	Warm	Ellipse	Polygon	Rounded Polygon
Cubic	Cool	Sphere Hemisphere	Cube Triangular Pyramid Square Pyramid Pentagonal Pyramid Triangular Prism Cuboid Pentagonal Prism Triangular Dipyramid Square Dipyramid Pentagonal Dipyramid	Cone Elliptic Cone Cylinder Elliptic Cylinder
	Warm	Torus	Prism Pyramid Dipyramid Polyhedron	Concurrence Form Opposition Form Piercing Form

4.3 KANSEI-Vocabulary Creation and KANSEI Scale Measurement according to Shape

Now experimental subject shapes are extracted on the basis of the forms analyzed in the previous section. They include four lines, and twenty shapes and KANSEI words are based on the forms collected. The process for the creation of a KANSEI-Vocabulary scale has two steps.

First, after the sample subject groups composed of 280 people were shown images of line and shape, we asked them to express adjectives which they felt in looking at each given image. Second, all the adjectives collected from step 1 we are classified according to their frequency of use. Third, among the words of section 2, those that are not included in the feelings or express demonstratively we removed despite their high frequency, finally gaining eight words according to their frequency.

Table 3. The Part of 1st KANSEI-Vocabulary by Shape

Shape image	KANSIE-Vocabulary
	balanced, boring, cold, comfortable, endless, honest, stable, static
	cold, direct, exclusive, high, initiative, rigid, selfish, strong
	active, dangerous, endless, initiative, jumping, slippery, speedy, uneasy
	beautiful, bright, elegant, free, peaceful, smooth, sunny, swollen
	plentiful, full, mild, perfect, relaxed, safe, satisfied, warm
	offensive, cold, crooked, dizzy, irritative, retrogressive, strange, uncomfortable
	ambiguous, confused, nervous, strange, unique, unstable, vague, worried
	abundant, balanced, comfortable, flexible, liberal, soft, tender, soft and yielding

※ The images shown in the above table can be different from the images used in the real experiment for vocabulary selection.

In the second step, another sample group of 250 people was employed to measure what degree of KANSEI is shown in vocabulary. The standard was five interval scales. The effects of size, color, aftereffect, and outward environments were controlled to reliably measure the KANSEI scales of each image of plane and line. We asked them to check their own KANSEI scale on the answer sheet after looking at each of the 24 images. Figure 2 shows a sample of measurement of the KANSEI scale: the KANSEI scale of 'plentiful' is 75%, the fourth in the five-degree section dividing 0~100 scale, where '100' means there is no difference between the KANSEI degree and the given KANSEI word while '0' means there is no similarity between them.

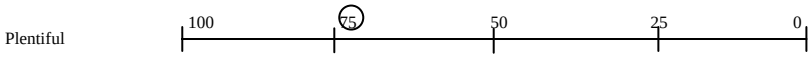


Fig. 2 Example of the KANSEI-Scale Measure

The time that it takes to show each shape image to the subject is represented in table 4. This time includes a few seconds that help the subject recognize each image without both depending on his/her own acquired knowledge and being affected by the previous image that he/she has seen before seeing the next image, plus a few more seconds for which they are allowed to write down the adjective of KANSEI that comes first to each person's mind.

Table 4. Standard Time of Scale Measure

	Object	Image(5sec./1image)	Response(20sec./1image)
Line	4	20 sec	80 sec
Shape	20	100 sec	400 sec

The KANSEI-Vocabulary Scale by shape is produced with the scale data measured over the 250 subjects by factor analysis. The purpose of factor analysis is to account for variables with the common underlying dimensions consisting of the elements of variables by analyzing the correlations of the multiple variables. In this study the method is employed because it minimizes the information loss of many vocabularies, narrows into minor factors, and gives a result in essential factors of KANSEI-Vocabulary by analyzing the relations among all the KANSEI-Vocabulary words and the relations between each shape and its related KANSEI-Vocabulary.

5 The Experimental Results

In Section 4 the procedures for the KANSEI-Vocabulary Scale by shape are represented. The procedures produced the result of the KANSEI-Vocabulary Scale as shown in Figure 3 and 4.

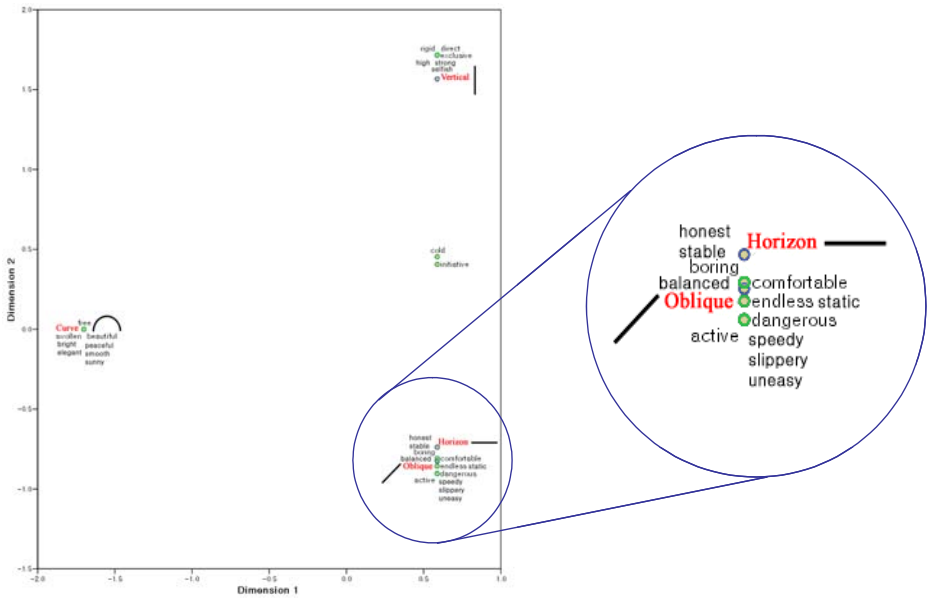


Fig. 3 KANSEI-Vocabulary Scale of Line

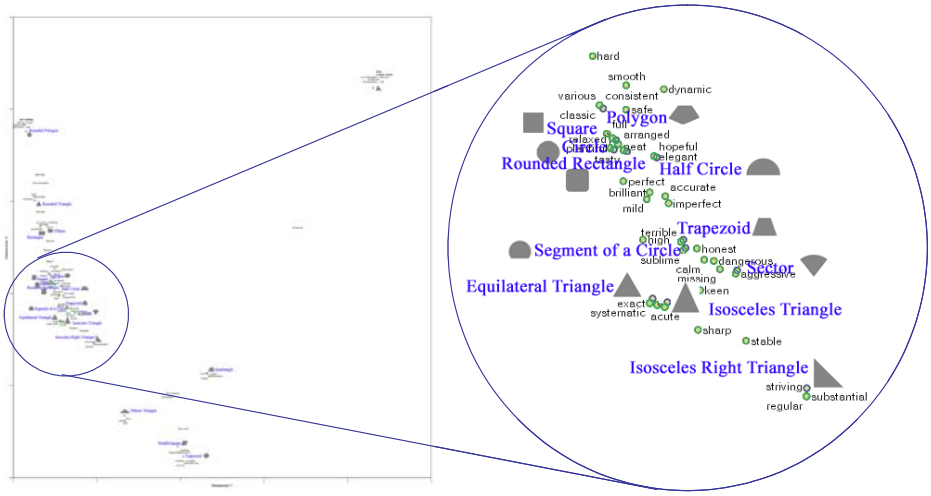


Fig. 4 KANSEI-Vocabulary Scale of Plane

Figure 3 represents the KANSEI-Vocabulary Scale of line and Figure 4 the KANSEI-Vocabulary Scale of plane. Lines and planes are treated separately and their scales are also produced separately, because in geometrical theory lines form a plane and planes a cubic, which indicates they exist in different areas and therefore the feelings they give are not consistent with one another. In other words, it should not be said that the feelings from a plane made of several lines are extended from the feelings from a vertical.

The result from the procedures of the KANSEI-Vocabulary Scale by shape shows that the features of each shape are closely related to human beings' KANSEI and decide the distance of dimension among the words of the KANSEI-Vocabulary. Figure 3 shows that the values of 'Dimension 1' are almost the same, but those of 'Dimension 2' are not. However, in the KANSEI words for a curve they are placed independently. This means that the words for a vertical are different from those for a curve.

Figure 4 shows that those of the sample group are more sensitive to curved shapes than to the other shapes and felt similar sensibilities to the given shapes. As the values of Shape-KANSEI of the given shapes are similar in spite of the different shapes, they are in close proximity. In other words, the factors of curve rather than those of line influence people's sensibilities as well as the factors of shape.

Table 5 represents the KASNEI words and some coordinates of shape on the KANSEI-Vocabulary Scale, which are applied to various areas by means of the distances between the shapes and each KANSEI word and the measure of the distance among the shapes.

Table 5. Scores in KANSEI-Vocabulary Scale by Shape

Name of Shape	Score in Scale		KANSEI-Vocabu-lary	Score in Scale		KANSEI-Vocabu-lary	Score in Scale	
	1	2		1	2		1	2
Circle	-0.537	0.103	plentiful	-0.547	0.107	irritative	3.361	2.325
Ellipse	-0.585	0.720	full	-0.556	0.145	retrogressive	3.361	2.325
Triangle	3.305	2.234	mild	-0.451	-0.028	strange	2.352	0.716
Equilateral Triangle	-0.435	-0.291	perfect	-0.513	0.019	uncomfort-able	3.361	2.325
Isosceles Right Triangle	-0.031	-0.527	relaxed	-0.547	0.107	exact	-0.443	-0.303
Isosceles Triangle	-0.397	-0.301	safe	-0.505	0.208	sharp	-0.316	-0.374
Obtuse Triangle	0.332	-1.326	satisfied	-0.547	0.107	threatening	-0.423	-0.308
Quadrangle	1.320	-0.858	warm	-0.547	0.107	precise	-0.443	-0.303
square	-0.532	0.128	cozy	-0.595	0.750	pricking	-0.443	-0.303
Rectangle	-0.663	0.697	crushing	-0.595	0.750	systematic	-0.443	-0.303
Rhombus	1.037	-1.768	dynamic	-0.406	0.262	missing	-0.259	-0.213
Parallelogram	0.967	-1.657	flexible	-0.728	1.291	regular	-0.031	-0.549
Trapezoid	-0.355	-0.135	natural	-0.595	0.750	honest	-0.319	-0.159
Polygon	-0.565	0.211	smooth	-0.505	0.272	stable	-0.190	-0.402
Segment of a Circle	-0.349	-0.157	wonderful	-0.595	0.750	striving	-0.031	-0.549
Half Circle	-0.424	0.082	recursive	-0.595	0.750	substantial	-0.031	-0.549
Sector	-0.213	-0.216	offensive	3.361	2.325	unstable	0.787	-1.123
Rounded Triangle	-0.742	1.012	cold	3.361	2.325	active	-0.404	-0.313
Rounded Rectangle	-0.503	0.098	corrked - curved	3.361	2.325	acute	-0.404	-0.313
Rounded Polygon	-0.846	1.761	dizzy	3.361	2.325	destructive	-0.404	-0.313

6 Conclusions

In this study we suggested a KANSEI-Vocabulary Scale by observing the relationships among KANSEI words and shapes which are those of the visual information. This scale can be used efficiently in KANSEI-based image retrieval according to the user's intentions and in part on the basis of intelligent information research to KANSEI. This experiment will help construct our KANSEI-based image retrieval system, which will be applied to various sections including product design and object production, whose main property is shape, to measure a user's recognition degree and evaluation.

This study is under work in texture and pattern as well, which will be also contributed to the final construction of KANSEI-Ontology based on the relation of visual information and KANSEI-Vocabulary. Such a result will allow knowledge retrieval, ontology-based information retrieval, and intelligent image retrieval because the

KANSEI-Vocabulary relation obtained from visual information induces a fixed quantity of sensibility data.

Acknowledgement

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment). (IITA-2005-C1090-0502-0009)

References

[1] Hideki Yamazaki, Kunio Kondo, “A Method of Changing a Color Scheme with KANSEI Scales,” Journal for Geometry and Graphics, vol. 3, no. 1, pp.77-84, 1999
[2] Shunji Murai, Kunihiko Ono and Naoyuki Tanaka, "KANSEI-based Color Design for City Map," ARSRIN 2001, vol. 1, no. 3, 2001
[3] Young-Joon Nam, “The Construction of Sensibility Thesaurus Based on Color,” Journal of Information Management, vol. 34, no. 4, pp. 43-61, 2003.
[4] Kobayshi Singenobu, “Color Image Scale,” Kodansha America, 1990
[5] Haruyoshi Nagumo, “Color Image Chart,” Chohyung Publishing Co., 2000
[6] Hewlett-Packard, http://www.hp.com/united-states/public/color/meaning.html, “Color Printing Center-The Meaning of Color”
[7] I.R.I, http://www.iricolor.com/04_colorinfo/colorsystem.html, “Adjective Image Scale”
[8] Mitsuteru KOKUBUN, "System for Visualizing Individual Kansei Information", Industrial Electronics Society, IECON 2000, 1592-1597 vol.3, 2000.
[9] Yungyoung Chong, Sungbae Cho, " Mapping Wavelet Feature Space to KANSEI Space in Image Using Neural Networks ", Korea Information science Society, The proceeding of Spring conference, vol. 27, no. 01, pp. 532-534, 2000.
[10] Hyunjang Kong, Wonpil Kim, Kunseok Oh, Pankoo Kim, " Building the Domain Ontology for Content Based Image Retrieval System ", Korea information processing Society, The proceeding of fall conference, vol. 9, no. 2, 2002.
[11] Rudolf archaism, Art and visual perception, Mijin Publishing Co., 1995
[12] Wucius Wong, Principles of Two-Dimentional Design, Van Nostrand Reinhold, pp.5-8, 1972.

The Characteristics of Filter Algorithm for Random Number Generator

Jinkeun Hong

Division of Information and Communication, Baekseok University,
115 Anse-dong, Cheonan-si, Chungnam, 330-740, South Korea
jkhong@bu.ac.kr

Abstract. For the hardware random number generator (RNG) in a crypto module,it is important that the RNG hardware offers an output bit stream that is always unbiased. However, even though the hardware generating processor generates an output bit stream quickly, if the software filter algorithm is inefficient, the RNG becomes time consuming, thereby restricting the conditions when an RNG can be applied. Accordingly, this paper proposes an efficient method of software filtering for an RNG processor in a crypto module. To consistently guarantee the randomness of the output sequence from a RNG, the origin must be stabilized, regardless of any change in circumstances. Therefore, a software filter model is analyzed to gain the optimum window depth of the random number generator. When it is considered the gathering quantities of output bit stream during the given the unit time, the consumed time, and the loss rate according to the probability of pass, in case of the designed random generator with software filtering, the decision of window depth is fitted at the unit level of 64bits or 128bits.

1 Introduction

Recent advances in wireless and wire communications and electronics have provided the emergence of several technologies and standards. Secure communication is the basis for multimedia and web technologies, such as online gaming technology, command and control technology. An Ubiquitous computing is continuing to grow, resulting in the construction of massively distributed computing environments[1-2]. However, the particular constraints imposed by ubiquitous computing, including computational power and energy consumption, raise significantly different security issues, such as authentication, authorization, accessibility, confidentiality, integrity, and non repudiation, along with more general issues, such as convenience, speed, and so on. A digital noise source that can provide a continuous stream of random binary numbers is very useful for encrypting digitized video, modem, or voice data. although mathematical algorithm and digital circuits can be used to generate pseudo random sequences for the purposed of simulation or circuit test and measurement, pseudo random source is inadequate for the more stringent requirements of data encryption. An H/W random number generator uses a non-deterministic source to produce randomness, and more

demanding random number applications, such as cryptography, a crypto module engine, and statistical simulation, then benefit from the sequences produced by an RNG, a cryptographic system based on a hardware component [1]. As such, a number generator is a source of unpredictable, ir-reproducible, and statistically random stream sequences, and a popular method for generating random numbers using a natural phenomenon is the electronic amplification and sampling of a thermal or Gaussian noise signal. However, since all electronic systems are influenced by a finite bandwidth, 1/f noise, and other non-random influences, perfect randomness cannot be preserved by any practical system. Thus, when generating random numbers using an electronic circuit, a low-power white noise signal is amplified, then sampled at a constant sampling frequency. Yet, when using an RNG with only a hardware component, as required for statistical randomness, it is quite difficult to create an unbiased and stable random bit stream. The studies reported in [3-5] show that the randomness of a random stream can be enhanced when combining a real RNG, LFSR number generator, and hash function. Hence, in previous studies about RNG schemes in the security area, Nicholas Riley and Craig Zilles (2005) presented probabilistic counter updates for predictor hysteresis and bias, Sergio Callegari, Riccardo Rovatti, and Gianluca Setti (2005) designed embedded analog digital converter based true random number generator for cryptographic applications exploiting nonlinear signal processing and chaos, Chua Chin Wang, et al. (2005) investigated switched current 3 bit CMOS 4.0MHz wide band random signal, and random numbers from meta stability and thermal noise was considered (D.C. Ranasinghe, et al. 2005 [6]). However, the randomness of such combined methods is still dependent on the security level of the hash function and LFSR number generator. Thus, a previous paper proposed a real RNG that combines an RNG and filtering technique that is not dependent on the security level of the period. Therefore, controlling a stable input voltage for an RNG is an important aspect of the design of an RNG. In particular, it is important that the RNG hardware offers an output bit stream that is always unbiased. Even though the hardware generating processor generates an output bit stream quickly, if the software filter algorithm is inefficient, the RNG becomes time consuming, thereby restricting the conditions when an RNG can be applied. When it is considered the gathering quantities of output bit stream during the given the unit time, the consumed time, and the loss rate according to the probability of pass, in case of the designed random generator with software filtering, the decision of window depth is efficient the unit level of 64bits or 128bits. Hereinafter, section 2 reviews the framework of the hardware RNG in crypto module, then section 3 examine the filter characteristics of software filter algorithm. Experimental results and some final conclusions are given in sections 4 and 5.

2 Framework of Random Number Generator

An H/W random number generator includes common components for producing random bit-streams, classified as follows: characteristics of the noise source,

amplification of the noise source, and sampling for gathering the comparator output. The applied noise source uses Gaussian noise, which typically results from the flow of electrons through a highly charged field, such as a semiconductor junction [7-8]. Ultimately, the electron flow is the movement of discrete charges, and the mean flow rate is surrounded by a distribution related to the launch time and momentum of the individual charge carriers entering the charged field. The Gaussian noise generated in a PN junction has the same mathematical form as that of a temperature-limited vacuum diode. The noise seems to be generated by the noise current generator in parallel with the dynamic resistance of the diode. The probability density of the Gaussian noise voltage distribution function is defined by Eq.(1):

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{(-\frac{x^2}{2\sigma^2})} \tag{1}$$

where σ is the root mean square value of Gaussian noise voltage.

However, for the proposed Gaussian noise random number generator, the noise diode was a diode with white Gaussian distribution. The power density for the noise was constant with a frequency from 0.1Hz to 10MHz and the amplitude had a Gaussian distribution. The noise comes from the agitation of the electrons within a resistance, which sets a lower limit on the noise present in a circuit. Thus, when the frequency range is given, the voltage of the noise is decided by a factor of the frequency. The crest factor of a waveform is defined as the ratio of the peak to the root mean square value, and here a crest value of approximately 4 was used for the noise. However, with the proposed real random number generator, since the noise diode is a noise diode with a white Gaussian distribution, the noise must be amplified to a level where it can be accurately thresholded with no bias using a clocked comparator. The applied noise source uses Gaussian noise, which typically results from the flow of electrons through a highly charged field, such as a semiconductor junction. Ultimately, the electron flow is the movement of discrete charges, and the mean flow rate is surrounded by a distribution related to the launch time and momentum of the individual charge carriers entering the charged field. The Gaussian noise generated in a PN junction has the same mathematical form as that of a temperature-limited vacuum diode. The noise seems to be generated by the noise current generator in parallel with the dynamic resistance of the diode.

$$V_n(rms) = \sqrt{2eI_{dc}B} \tag{2}$$

Where e is electron charge (1.6 x 1E-19 coulombs), I_{dc} is average dc current (A), B is noise bandwidth (Hz). The Gaussian noise voltage can be determined by applying Ohms Law.

$$E = \sqrt{2eI_{dc}Br_d^2} \tag{3}$$

Where, rd is the dynamic resistance of the junction. It is known that the dynamic resistance of a PN junction depends on the temperature and direct

current flowing through the junction. The dynamic resistance represents the ratio of a small change in the diode voltage to the corresponding change in the diode current.

$$E = kT\sqrt{(2/eI_{dc}} \tag{4}$$

Where K is Boltzmanns constant (1.38 x 1E-23 Joules/deg. Kelvin), T is temperature in degrees Kelvin. B is bandwidth in Hertz, rd is dynamic resistance, and e is electron charge (1.6 x 1E-19 coulombs). The dynamic resistance is inversely proportional to the direct current and decreases as the direct current increases, thereby causing the Gaussian noise voltage across the junction to decrease. Yet, if the direct current increases, the dynamic resistance decreases more quickly than the Gaussian noise current increases. As such, the Gaussian noise voltage becomes inversely related to the direct current. A hardware random number generator includes common components for producing random bit-streams, classified as follows: characteristics of the noise source, amplification of the noise source, and sampling for gathering the comparator output. The applied noise source uses Gaussian noise, which typically results from the flow of electrons through a highly charged field, such as a semiconductor junction. Ultimately, the electron flow is the movement of discrete charges, and the mean flow rate is surrounded by a distribution related to the launch time and momentum of the individual charge carriers entering the charged field.

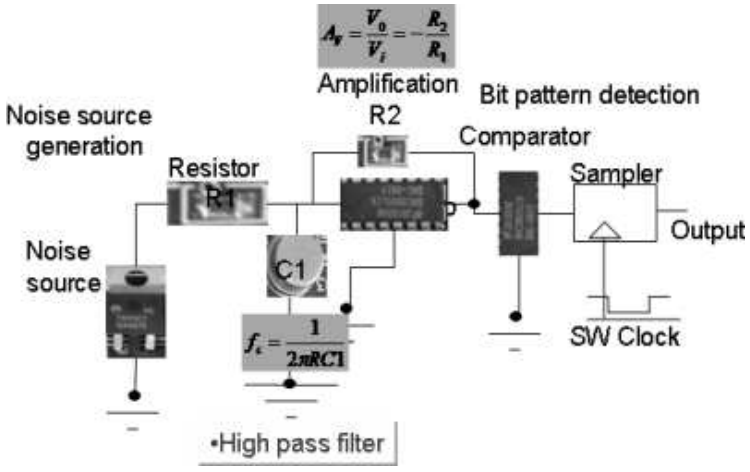


Fig. 1. The Architecture of Random Number Generator

3 The Filter Characteristics of Random Number Generator

The filter algorithm is applied in the next process of the output stream of the sampler to reduce the biased statistical randomness. If the buffer size [32bits] and significance level [γ] are established, this supports unbiased and stable randomness. In the filter model, a static buffer[S] memory of 32bits is used to buffer the "pass data" in the decision boundary, and the significance level for the P value is between 0.9995 and 1.0005.

$$P = t \setminus T \tag{5}$$

Where the total sum t is the sum of the number of "1"bit patterns and the total length T is the half value of the static length.

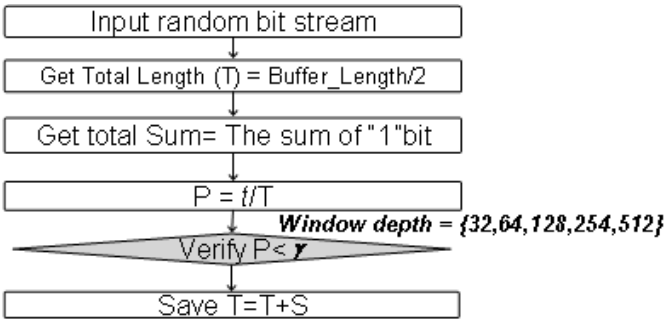


Fig. 2. Process of Filter Algorithm Model

Algorithm: SoftwareFilter
1.Let Threshold levelγ: 0.9995$\leq \gamma \leq$1.0005, Window depth W:16,32,64,256,512;
2.Given RNGSequence Total Length: Buffer Length/2;
3.Total Sum(t):the sum of pattern"1"bit
4.Let be P=t/T;
5.Decision $P < \gamma$;
6.If it is passed, then T=T+S;
7.Else Discard S;

Where w is the window step size (32bits). When the static buffer is fixed at 64bits, the half-value of the static length is 32 bits. If the value of the number of a pattern '1'bit / the half-value of the static length within the total length is included within the significance level, the decision will be the state of "pass". If the condition of "pass" is decided, this is added as pass data to the buffer memory. If "fail" is decided, through the conventional filtering process, this is included in the decision process. The process is then completed when the size of

the desired bit stream is gathered. The failed bits (32bits, S) are then made uniform by conventional filter (i.e., the duty distribution of the bit stream "0" and "1" is normalized). In conventional model, the output bit stream is expanded in steps of 32bits, while simultaneously evaluating the significance level. If the value of the duty cycle of the collected output bit stream, P, satisfies the condition of significance level, the 32bit stream is added, otherwise it is discarded. A tree filter model is applied in the next process of the output stream of the sampler to reduce the characteristics of a biased bit stream and as an efficient method that can reduce the time consumption. In the evaluation of randomness of random number, to provide the security level in respect of randomness of the gathered output bit stream, it is important to decide the optimal threshold level and window depth. In case of the decision of threshold level, the factor of threshold level is a critical element to detect the non biased bit stream. If the threshold level is increased, the passed probability is enhanced and the loss rate is degraded. In Fig. 3, when the value of threshold level sets between 0.9995 and 1.0005, for the evaluation of randomness of 20 rounds, the passed probability is provided the level of 97.2 percentage. If the bound level of passed probability is between 0.96 and 1.04, then it is the level of 92 percentage. Also if the bound level of pass is between 0.9999 and 1.0001, then the passed probability provides the value upper to 99 percentage.

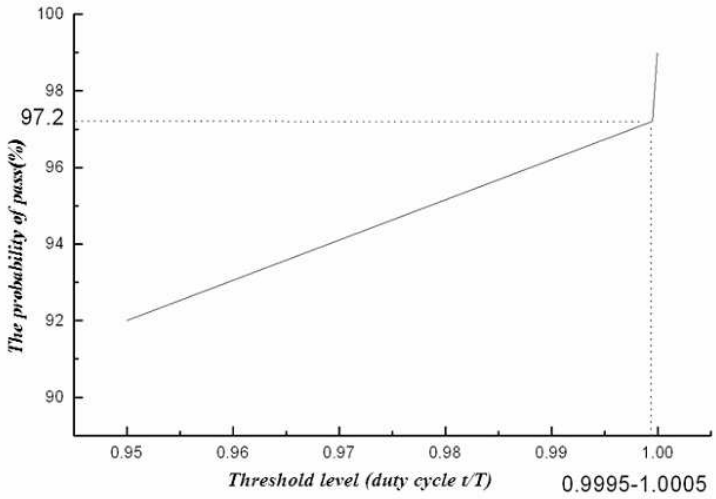


Fig. 3. The Pass probability of randomness according to threshold level

As the consideration of randomness of bit stream according to threshold level, the major point is that if the bound of the passed level is reduced, then the passed probability is increased, but loss rate is degraded. In Fig. 4, it is considered the loss of random bit stream. If the threshold level set between 0.9995 and 1.0005,

then the loss rate of output bit stream is the upper hand of 10 percentage. In relatively, if the threshold level is widened (the bound is between 0.95 and 1.05), the loss rate is reduced. If the bound of the threshold level is between 0.95 and 1.05, the loss rate of random bit stream is the upper hand of 20-30percentage. In the designed random number generator,if the result of statistical randomness is averagely the upper hand of 92 percentage, then it is guaranteed the random bit stream. Therefore when it is decided the bound of threshold level, it is fitted the bound level between 0.9995 and 1.0005, in respect of passed probability and loss rate.

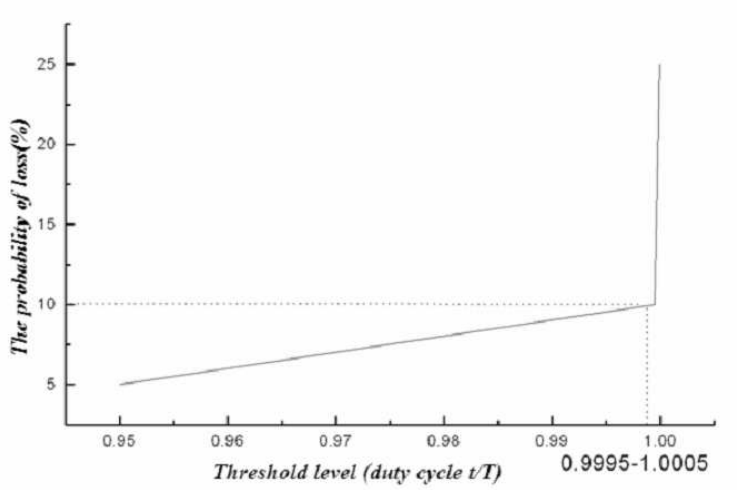


Fig. 4. The Loss probability of output bit streams according to threshold level

4 Experimental Results

A multiple bit stream of consecutive bits as the output from the RNG was subjected to a mono bit test (such as Eq.(5)), et. al.[9-10]. If any of the tests fail, the module then enters an error state. The statistical RNG test method of FIPS140-1 is used, on the basis of the statistical RNG randomness. When it is gathered to output bit stream, the loss rate of biased output bit stream is affected by the decision rule of threshold level, and the loss bit stream is affected by the window depth, which is a critical factor to gather bit stream. The applied window depths are based on a window depth of 16bits, 32bits, 64bits, 128bits, 256bits, and 512bits. If the window depth W is 16 bits, instead of 512bits, the probability of loss is reduced and the quantity of gathered bit stream is reduced, when it is considered to the filter process of one round. To generate the random number and test randomness, when the window depth is 512 bits, it is difficult to

pass the threshold level, in comparison with a window depth of 16 bits. The bit stream, is gathered by the filter process one round, due to a window depth of 512 bits, is increased. Also the probability of loss, relatively, is increased and does not satisfy the condition of significance level, which is the level of randomness. Therefore the window depth of software filter is decided in consideration of the characteristics of random number generator.

Table 1. The total bits according to the window depth of software filter

Depth[1]	Generation bits	Loss bits	Pass Prob.[2]
16	1.6E+07	1.6E+04	99.9
32	3.2E+07	6.4E+04	99.8
64	6.4E+07	2.56E+04	99.6
128	1.28E+08	1.02E+06	99.2
256	2.54E+08	4.07E+06	98.4
512	5.12E+08	1.61E+07	96.8

[1]Depth: filter window depth
[2]Pass Prob.: in condition of W=16,Prob. is 99.9percentage

When it is processed the filtering of 1 million round during 10 seconds, the total generation bits, are gathered according to the window depth, are presented in Table2. If the window depth is 16 bits, the probability of pass is 99.9 percentage, but in case of the window depth, which is 32 bits, the probability of pass is 99.8 percentage. In general condition, if the window depth is increased, the probability of pass is decreased, relatively. In experimentally, if the window depth is decided 64bits, the probability of pass is 81.5 percentage. Whether it is the biased bit stream or not, if the window depth, which is 512 bits, is applied, the failed probability and loss probability are increased. From the review of this result, it does not fit the window depth, which is 512 bits. It is needed to review the relation of the probability of pass according to window depth and the consumed time. When the window depth set 16 bits, the gathered bits by the filter process of one round is the unit of 16 bits, in case of the window depth 512 bits, the gathered bits is the unit of 512 bits. If it is not considered the loss rate, then the gathered bits are increased a degree of the unit. In Table3, it is presented the probability of pass according to the window depth. If the probability of pass is 99 percentage and the window depth set 512 bits, it is consumed 1 minute to process the same quantities. But if the window depth set 16 bits, it is consumed 23.5 minutes, in case of 256 bits, it is consumed 1.7 minutes. But if the probability of pass is degraded, as much as 95 percentage, the gathering speed of the smaller window depth is enhanced, in contrast of that of the larger window depth.In case of the window depth 256 bits, it is consumed 0.9 minute. The window depth is the unit of 512 bits, then,it is needed 1 minute. Therefore,

in condition of passed probability, which is reduced, the window depth is larger, the loss rate is more. In this case, it is efficient that the window depth is the unit of 256 bits than the unit of 512 bits.

Table 2. The probability of pass according to the window depth of software filter

Depth	Prob.[3]										
16	99.9	99.7	99.5	99	97	95	93	91	90	85.0	80.0
32	99.8	99.4	99	98	94	90	86	82	81	73.2	64.0
64	99.6	98.8	98	95	88.5	81.5	74.8	68.6	65.6	52.2	41.0
128	99.2	97.6	96.1	92.3	78.4	66.3	56	47	43	27.2	16.8
256	98.4	95.3	92.3	85.1	61.4	44	31.3	22.1	18.5	7.41	2.8
512	96.8	90.8	85	72.5	37.7	19.4	9.8	4.9	3.4	0.51	0

[3]Prob: The probability of pass per 1 min

When it is considered the gathering quantities of output bit stream during the given the unit time, the consumed time, and the loss rate according to the probability of pass, in case of the designed random generator with software filtering, the decision of window depth is efficient the unit level of 64bits or 128bits.

Table 3. The comparison of probability of pass and consumed time according to the window depth of software filter

Depth	Case 3	Case 4
16	23.5min	6.5min
32	11.8min	3.4min
128	6.0min	1.9min
256	3.1min	1.2min
512	1.0min	1.0min

Case 3: When depth is 16, Pass prob. is 99, Case 4: When depth is 16, Pass prob. is 95

5 Conclusions

A previous paper proposed a real RNG that combines an RNG and filtering technique that is not dependent on the security level of the period. It is important

that the RNG hardware offers an output bit stream that is always unbiased. In the random number generator, if the result of statistical randomness is averagely the upper hand of 92then it is guaranteed the random bit stream. Therefore when it is decided the bound of threshold level, it is fitted the bound level between 0.9995 and 1.0005 in respect of passed probability and loss rate. When it is considered the gathering quantities of output bit stream during the given the unit time, the consumed time, and the loss rate according to the probability of pass, in case of the designed random generator with software filtering, the decision of window depth is efficient the unit level of 64bits or 128bits. Accordingly, this paper proposes an optimal window depth of software filtering for an RNG processor in a crypto module. To consistently guarantee the randomness of the output sequence from a RNG, the origin must be stabilized, regardless of any change in circumstances.

References

1. H. Alireza and V. Ingrid. :High-Throughput Programmable Crypto-coprocessor. *IEEE Computer Society*(2004)
2. A. M. Jalal, R. Anand, C. Roy, and M. D. M. Cerberus. :A Context - Aware Security Scheme for Smart Spaces. *IEEE PerCom'03*(2003)
3. Riley, N. and Ziles, C. :Probabilistic counter updates for predictor hysteresis and stratification. *HPCA 2006*, Feb.(2006)
4. Callegari, S., Rovatti, R., and Setti, G. :Embeddable ADC based true random number generator for cryptographic applications exploiting nonlinear signal processing and chaos. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol.53, Issue 2, Part 2, pp.793-805, Feb.(2005)
5. Chua Chin Wang, Jian Ming Huang, Hon Chen Cheng, and Hu R. :Switched current 3 bit CMOS 4.0MHz wideband random signal generator. *IEEE journal of Solid State Circuits*,Vol.40, Issue 6, pp.1360-1365, June(2005)
6. Ranasignhe, D. C., Lim, D., Devadas, S., Abbott, D., and Cole, P. H. :Random numbers from metastability and thermal noise. *Electronics Letters*, Vol.41, Issue 16, pp.13-14, Aug.(2005)
7. http://webnz.com/robert/true rng.html.
8. Boris Ya, Ryabko and Elena Matchikina. :Nonlinear Switched-Current CMOS IC for Random Signal Generation. *IEE Electronic Letters*, vol.29 (1993)
9. M. Delgado-Restituto, F. Medeiro, and A. Rodriguez-Vasquez. :Fast and Efficient Construction of an Unbiased Random Sequence. *IEEE Trans. on Information Theory*, vol.46, No.3 (2000)
10. Diehard. http://stat.fsu.edu/ geo/diehard.html. (1998)

Reduced Quantized Colors for Content Based Image Retrieval

Jong-An Park[1] , Muhammad Bilal Ahmad[2], Tae-Sun Choi[2], Sung-Bum Pan[1] and Young-Eun An[1]

[1] College of Electronics & Information Engineering,
Chosun University, Gwangju, Korea.
japark@chosun.ac.kr

[2] Signal and Image Processing Lab, Dept. of Mechatronics,
Gwangju Institute of Science and Technology,
Gwangju, Korea
bilal@gist.ac.kr

Abstract. Content based image retrieval (CBIR) usually uses color features for representing images. The most common features of images are color histogram and color correlogram. The color correlogram was proposed to characterize not only the color distribution of pixels as in the case of color histogram, but also the spatial correlation of pairs of colors. Simple RGB values and their normal sequence r,g,b of pixels are used for color correlogram. In this paper, dominant r,g,b components for color correlogram are found and presented as the features of images. Simulation results show better results with the proposed features as compared to normal RGB features.

1 Introduction

It is hard to access or make use of the huge amount of information in the order of terabytes available on the web unless it is organized so as to allow efficient browsing, searching, and retrieval. Content-based image retrieval (CBIR) [1], [2], a technique which uses visual contents to search images from large scale image databases according to users' interests, has been an active and fast advancing research area since the 1990s. During the past decade, remarkable progress has been made in both theoretical research and system development. However, there remain many challenging research problems that continue to attract researchers from multiple disciplines.

CBIR systems adopt the following two-step approach to search image databases: (1) (indexing) for each image in a database, a feature vector capturing certain essential properties of the image is computed and stored in a feature base, and (2) (searching) given a query image, its feature vector is computed, compared to the feature vectors in the feature base, and images most similar to the query image are returned to the user. The idea of CBIR is to extract characteristic features from target images which are then matched with those of the query image. These features are typically derived

from shape, texture, or color properties of query and target images [3], [4]. After matching, images are ordered with respect to the query image according to their similarity measure and displayed for viewing. Color is widely used to represent an image [5-14]. The color composition of an image, which is usually represented as a histogram of intensity values [5], is a global property which does not require knowledge of the component objects of an image. Moreover, color distribution is independent of view and resolution, and color comparison can be carried out automatically without human intervention. However, it has become clear that color alone is not sufficient to characterize an image. Two very different images might have very similar color distribution (histogram). Therefore, spatial distribution of colors is also very important. To facilitate a more accurate retrieval process, integrated color-spatial retrieval techniques that employ both the color information as well as the knowledge of the colors' spatial distribution for image retrieval has been explored in the color correlogram [6], [7], [8], [15].

In color histogram and color correlogram, RGB values and their normal sequence R, G, B of pixels are used. In this paper, we propose different operations on pixels based on dominancy of R, G, B values within the pixels before finding color correlogram. The results are compared with the previous color based feature extraction methods.

The organization of the paper is as follows. Section 2 describes prior features extraction method based on color histogram and correlogram. The proposed algorithm is described in section 3. Simulation results are shown in section 4.

2 Prior Color Feature Extraction Algorithms

Color is very important feature and is widely used in content based image retrieval (CBIR) system. Color space is first selected before describing color description. RGB color space is very popular, because of its three dimensions, which makes RGB color space more superior than single dimension of gray values of images. Each pixel of the image can be represented as a point in a 3D color space, namely, red, green, and blue.

2.1 Color Histogram

The most simple and popular method of color feature extraction is to find the color histograms of images. Color histogram can be defined as follow:

Let I be an *n x n* image (a square image is considered for simplicity). The image I is quantized into q colors $c_1,..,c_q$. For a pixel $p=(x,y)\in I$, let $C(p)$ denote its color. Let $I_c \equiv \{p|C(p)=c\}$. The histogram *h* of I is defined as:

For a color pixel c_j, $j\in[q]$

$$H_{c_j}(I) \equiv \left\| I_{c_j} \right\| \tag{1}$$

Equation (1) gives the number of pixels of color c_j in I. The probability of randomly taking any pixel p having color c_j from the image I is then given as

$$h_{c_j}(I) \equiv \Pr\left[p \in I_{c_j}\right] = \frac{H_{c_j}(I)}{n^2} \tag{2}$$

The histogram is computed in $O(n^2)$. Color histogram serves as an effective representation of the color content of an image if the color pattern is unique compared with the rest of the data set. The color histogram is easy to compute and effective in characterizing both the global and local distribution of colors in an image. In addition, it is robust to translation and rotation about the view axis and changes only slowly with the scale, occlusion and viewing angle. However, color histograms have many inherent problems in indexing and retrieving images.

Since any pixel in the image can be described by three components in a certain color space (for instance, red, green and blue components in RGB space, or hue, saturation, and value in HSV space), a histogram, i.e., the distribution of the number of pixels for each quantized bin, can be defined for each component. Clearly, the more bins a color histogram contains, the more discrimination power it has. However, a histogram with a large number of bins will not only increase the computational cost, but will also be inappropriate for building efficient indexes for image databases. The most common size of histograms consists of from 64 to 256 bins. Furthermore, a very fine bin quantization does not necessarily improve the retrieval performance in many applications. One way to reduce the number of bins is to use clustering methods to determine the K best colors in a given space for a given set of images. Each of these best colors will be taken as a histogram bin.

In addition, color histogram does not take the spatial information of pixels into consideration, thus very different images can have similar color distributions. This problem becomes especially acute for large scale databases. To increase discrimination power, several improvements have been proposed to incorporate spatial information. A simple approach is to divide an image into sub-areas and calculate a histogram for each of those sub-areas. Increasing the number of sub-areas increases the information about location, but also increases the memory and computational time.

2.2 Color Correlogram

To take the spatial information into account, color correlogram was proposed. Color correlogram characterizes color distributions of pixels as well as the spatial correlation of pairs of colors. The first and second dimensions of the three-dimensional histogram are the colors of any pixel pair and the third dimension is their spatial distance. A color correlogram is a table indexed by color pairs, where the d-th entry for (c_i, c_j) specifies the probability of finding a pixel of color c_j at a distance

d from a pixel of color c_i in the image. Let a distance d be fixed a priori. Then, the color correlogram is defined as:

$$\Phi^{d}_{c_i,c_j}(I) = \Pr\left[p_2 \in I_{c_j}, |p_1 - p_2| = d \middle| p_1 \in I_{c_i}\right] \tag{3}$$

The order of correlogram is $O(q^2 d)$. If we consider the correlation between the identical colors, we get autocorrelogram defined as:

$$\phi^{d}_{c}(I) = \Phi^{d}_{c,c}(I) \tag{4}$$

Autocorrelogram has order of only $O(qd)$. High value of d would result in expensive computation and large storage requirements. A small d might compromise the quality of the feature. But the main concern is the quantized colors q. Generally q is chosen from 64 to 256 quantized colors, which is quiet big number for correlogram. Compared to the color histogram, the color autocorrelogram provides the best retrieval results, but is also the most computational expensive due to its high dimensionality.

3 The Proposed Algorithms

In this paper, two algorithms are proposed. One is single pixel based while the second is based on the group of pixels.

3.1 Max/Min Color Component Based Color Correlogram

The maximum component of a color pixel is defined as the color component that has maximum value between the three components (R,G,B) of a pixel. Similarly the minimum component of the same color pixel is defined. The color correlogram is redefined based on the maximum/minimum of color component of a pixel for different spatial distances.

Let the color of a pixel be expressed into its three components as triplet order $C(p) \equiv (r_p, g_p, b_p)$. The maximum color component of a pixel $p = (x, y) \in I$ is determined as:

$$\max(I(p)) = \arg\max_{r,g,b}\left(r_p, g_p, b_p\right) \tag{5}$$

Equation (5) returns the index of maximum component. Red is represented by index '1', while green by '2' and blue by '3'. Similarly the minimum color component of a pixel $p = (x, y) \in I$ is determined as:

$$\min(I(p)) = \arg\min_{r,g,b}(r_p, g_p, b_p) \tag{6}$$

Equation (6) returns the index of minimum component i.e., 1,2,3 for red, green and blue, respectively. The quantized colors q in the cases of the color histogram and the color correlogram is here reduced to set

$$S = \{12,13,21,23,31,32\} \tag{7}$$

The meaning of quantized colors is; 12 means red is the maximum and green is the minimum of the color pixel. Similarly, 23 means green is the maximum and blue is the minimum, and son. The modified color correlogram is now a table indexed by color pairs, where the d-th entry for (c_i, c_j) specifies the probability of finding a pixel of color c_j from set S at a distance d from a pixel of color c_i from set S in the image. The modified color correlogram has the complexity of the order $O(6x6xd)$.

3.2 Color Correlogram for sub-images

The whole image I is divided into M x M sub-images. Instead of obtaining color features at pixel level; we consider it at block-level. In other words, we determine the colors that can be used to represent a block or sub-image. The RED component from each pixel in the sub-images is added together. Similarly, the GREEN and BLUE components of pixels are added in the sub-images. The maximum component color of a sub-image is defined as the color component that has maximum value between the three added components (R,G,B) of the sub-image. Similarly the minimum of the same sub-image is defined.

The color correlogram is redefined based on the maximum/minimum of color component of a sub-image for different spatial distances with other sub-images. Figure 1 shows the three components red, green and blue of an image. The shaded part shows one sub-image/block. Each block is represented by the sum of their component values. For example, the sum of red components within a block can be defined as:

$$R^{sum}(I^k) = \sum_{x=1}^{M}\sum_{y=1}^{M} I^k(x,y) \tag{8}$$

where I^k represents the *kth* block of image I. After dividing the image I into sub-images/blocks, and then summation of pixel values within the block's component, color correlogram is found in the same manner as explained in section 3.1.

The color correlogram is now a table indexed by color pairs, where the *k-th* entry for (c_i, c_j) specifies the probability of finding a sub-image/block of color c_j from set S at a distance d from a sub-image of color c_i from set S in the image. The color

correlogram has the complexity of the order $O(6x6xd/K)$, where K represents the number of blocks in an image I.

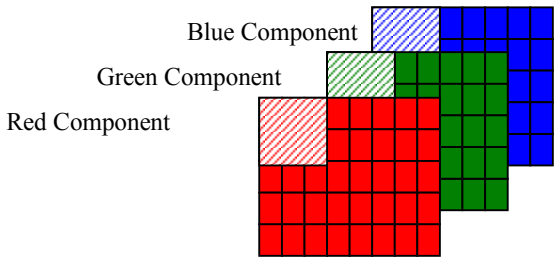


Fig. 1. Image division into components and sub-images.

4 Simulation Results

There are three steps to use color histogram/correlogram for image retrieval. The three basic steps are: (i) color space quantization for a choice of color space (we use RGB color space). The division of colors into q levels, $c_1,..,c_q$; (ii) histogram/correlogram binning – a scan of the image to count the number of color pixels in each color bin; and (iii) histogram/correlogram matching – choosing a distance function D to measure the similarity of two histograms/correlograms. The choice of color space, and the choice of the distance function D, determines the performance of a CBIR system that uses color histograms. Swain et. al [6] describes the popular distance function D as:

$$D(h(I),h(Q)) = \frac{\sum_{j=1}^{q} \min\left(h_{c_j}, h_{q_j}\right)}{\sum_{j=1}^{q} h_{q_j}} \tag{9}$$

where h_{c_j}, h_{q_j} are color counts of color j in image I (database image) and Q (query image). We use the same distance function D for our proposed algorithms.
For performance measure, the criterion proposed by Gevers and Smeulders [4] is used. Let rank r^{Q_i} denote the position of the correct match for query image $Q_i, i = 1,..., N_2$,

in the ordered list of N_1 match values. The rank r^{Q_i} ranges from $r = 1$ from a perfect match to $r = N_1$ for the worst possible match. The average ranking $\bar{r}$ and its percentile are defined by

$$\bar{r} = \frac{1}{N_2}\sum_{i=1}^{N_2} r^{Q_i} \text{ , and} \tag{10}$$

$$\bar{r}\% = \left(\frac{1}{N_2}\sum_{i=1}^{N_2}\frac{N_1 - r^{Q_i}}{N_1 - 1}\right)100\% \tag{11}$$

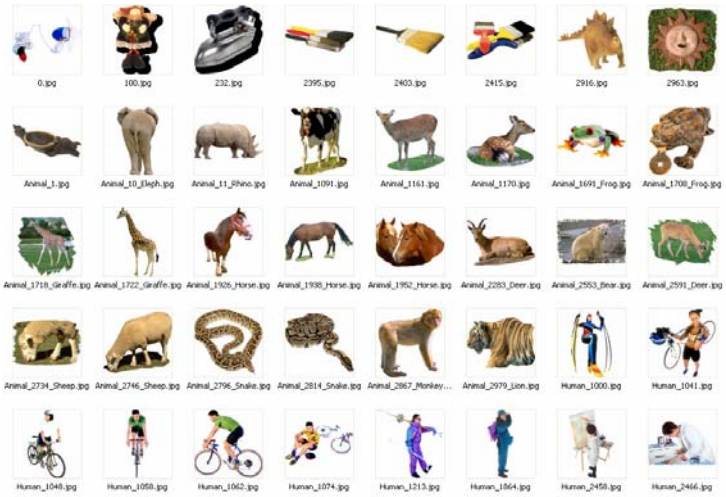


Fig. 2. Few images from the database

The performance measures of the proposed algorithms are compared with color histogram and color correlogram based algorithms in the literature. Few images from the database are shown in Fig. 2. The number of query images selected are *100* from 3500 database images. The retrievals based on the proposed algorithms T^{pixel_based}, T^{block_based} with the block size *16x16*, color histogram T^{Hist}, and color correlogram T^{Corr} are used for the comparison. The spatial distance $d = 1$ is selected for all correlogram based algorithms. Table 1 shows the performance measures of different image retrieval algorithms.

Table 1 Image retrieval performance measures

Algorithm	$\bar{r}$ measure	$\bar{r}$ % measure
T^{pixel_based}	57.5	93.1
T^{block_based}	53.0	95.4
T^{Hist}	60.5	92.1
T^{Corr}	55.5	95.8

5 Conclusions

The color contents based feature extraction algorithms are studied for image retrieval. The color histogram does not take into account the spatial relationship among the pixels. The color correlogram takes care of the spatial correlation of pairs of colors. In this paper, the color features extraction based on the maximum and minimum of color components are presented. Instead of obtaining color features at pixel level; we also consider it at block-level. The color correlogram is defined based on the maximum/minimum of color component of a pixel or a sub-image for different spatial distances.

Acknowledgements

This study was supported by Ministry of Culture & Tourism and Culture & Content Agency in Republic of Korea.

References

1. Yong Rui and Thomas S. Huang, "Image retrieval: Current technologies, promising directions, and open issues," Journal of Visual Communication and Image Representation, vol. 10, pp. 39-62, 1999.
2. Arnold W.M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain, "Content-based image retrieval at the end of the early years," IEEE Transactions of Pattern Analysis and Machine Intelligence, vol. 22, No. 12, pp. 1349-1380, December 2000.
3. W. Hsu, T.S. Chua and H.K. Pung, "An integrated color-spatial approach to content-based image retrieval," 3rd ACM Multimedia Conf., pp. 305-313, Nov. 1995.
4. S. F. Chang, W. Chen, H. Weng, H. Sundaram and D. Zhong, "VideoQ: an automatic content-based video search system using visual cues," Proc. ACM Multimedia, pp. 313-324, 1997.
5. B. Funt and G. Finlayson, "Color constant color indexing," IEEE Trans. On Pattern Analysis and Machine Vision, vol. 17, pp. 522-529, 1995.
6. J. Huang, S.R. Kumar, M. Mitra, W.J. Zhu and R. Zabih, "Image indexing using color correlograms," Proc. 16th IEEE Conf. on Computer Vision and Pattern Recognition, pp. 762-768, 1997.

7. Jing Huang, S Ravi Kumar, Mandar Mitra, Wei-Jing Zhu and Ramin Zabi, “Spatial color indexing and applications,” International Journal of Computer Vision, 35(3) pp. 245-268, 1999.
8. V. Kovalev, S. Volmer, “Color co-occurrence descriptors for querying-by-example,” Multimedia Modeling (MMM’98), p. 32-38, 1998.
9. Theo Gevers and Arnold W.M. Smeulders, “PicToSeek: Combining color and shape invariant features for image retrieval,” IEEE Transactions on Image Processing, vol. 9, No. 1, pp. 102-119, January 2001.
10. A.K. Jain and A. Vailaya, “Image retrieval using color and shape,” Pattern Recognition, vol. 29, No. 8, pp. 1233-1244, 1996.
11. M. Swain and D. Ballard, “Color indexing,” International Journal of Computer Vision, 7(1) pp. 11-32, 1991.
12. Brian V. Funt and Graham D. Finlason, “Color constant color indexing,” IEEE Transactions on Image Processing, vol. 17, No. 5, pp. 522-529, May 1995.
13. Jan-Mark Geusebroek, Rein van den Boomgaard, Arnold W.M. Smeulders, and Hugo Geerts, “Color invariance,” IEEE Transactions of Pattern Analysis and Machine Intelligence, vol. 23, No. 12, pp. 1349-1380, December 2001.
14. D.A. Forsyth, “A novel algorithm for color constancy,” International Journal of Computer Vision, vol. 5, No. 1, pp. 5-36, 1990.
15. G. C. Gotlieb and H.E. Kreyszig, “Texture descriptors based on co-occurrence matrices,” Computer Vision, Graphics, and Image Processing 51, 1990.

Institute of Computer Vision and Applied Computer Sciences IBaI

Director: Dr. Petra Perner

Address: Körnerstr. 10
04107 Leipzig
Germany

Phone: +49 341 8612273
FAX: +49 341 8612275
E-Mail: info@ibai-institut.de

Personal Homepage:
www.ibai-research.de

Institute`s Homepage:
www.ibai-institut.de

International Conference on Data Mining and Machine Learning MLDM
www.mldm.de

Industrial Conference on Data Mining ICDM
www.data-mining-forum.de

BioMedVision Center
www.biomedvision.de

Data Minng Tutorial
www.data-mining-tutorial.de