

# A Discernibility-Based Approach to Feature Selection for Microarray Data

Zacharias Voulgaris and George D. Magoulas, *Member, IEEE*

**Abstract**— Feature selection has been used widely for a variety of data, yielding higher speeds and reduced computational cost for the classification process. However, it is in microarray datasets where its advantages become more evident and are more required. In this paper we present a novel approach to accomplish this based on the concept of discernibility that we introduce to depict how separated the classes of a dataset are. We develop and test two independent feature selection methods that follow this approach. The results of our experiments on four microarray datasets show that discernibility-based feature selection reduces the dimensionality of the datasets involved without compromising the performance of the classifiers.

**Index Terms**—classification problems, feature selection, dimensionality reduction, discernibility, microarray data.

## I. INTRODUCTION

Feature selection has been an active research area over the past few years, as it promises fast performance and reduced complexity in classification problems. In datasets of high dimensionality, particularly those encountered in bioinformatics applications, feature selection is not merely a plausible but often an essential part of the classification process. To this end, over the last few years a series of methods for feature selection have been introduced, most of them focusing on microarray data [1-9], where the number of data points is small whilst the dimensionality of the features is high. In addition, sometimes the accuracy rate is increased when the classifier is applied on the reduced feature set. This renders feature selection an important technique for this type of classification problems.

The methods used in the literature have been applied mostly for classification with SVMs [1, 2, 3], although other approaches have been also considered [4, 5, 6, 7, 8, 9]. The focus of feature selection on SVM classifiers is not random. Apparently, this type of classifier is the one that is mostly affected by dimensionality, since the relatively high number of features compromises their performance due to over fitting [10].

As regards the methods themselves, feature selection has been accomplished using Genetic Algorithms [1, 4], Locally Linear Embedding (LLE) [2], Recursive Salient Analysis (RSA) [3], the selection of a support set [5], a combination of PCA and the UKW clustering algorithm [6], t-tests [4, 7], Regression Splines (MARS) [7], Classification and Regression Trees (CART) [7], Random Forests [7], Linear Genetic Programs (LGP) [7], Neural

Networks as a similarity measure [8], and clustering analysis [9].

Most of these methods were applied on one or more of the following datasets: Lymphoma, Colon, Leukaemia, Prostate, and Brain Cancer. All of these datasets have microarray data, exhibiting an ultra high dimensionality and are publicly available at [11].

All of the methods of the literature appear to work well, for one or the other dataset, for SVMs or other classifiers used in each particular case. However, the underlying problem that all of them have is that there is no particular stopping criterion for the feature selection. In other words, the reduced feature sets come in a variety of sizes with no way of knowing beforehand anything regarding the quality of the selected features. Our approach is an attempt to rectify this situation.

This paper is structured as follows: in Section II the Index of Discernibility is presented, and its use for feature selection with the 2 methods introduced in this paper is discussed. In Section III our experiments are presented and the results are discussed. Finally, in Section IV, conclusions are drawn.

## II. FEATURE SELECTION USING THE INDEX OF DISCERNIBILITY

The Index of Discernibility (ID) is a measure developed for assessing how easily distinguishable the classes of a dataset are [14]. The ID makes use of (hyper-)spheres of fixed radius around each element of the dataset, which corresponds to the average distance between this and the rest of the elements of that class. Note that the radius depends on the class structure, so elements belonging to different classes may have different radii. Once the radius of an element is established, elements of the same class as the examined element that belong to its (hyper-)sphere are identified and counted. The discernibility of that element is calculated by dividing the number of these elements by the number of total elements in the (hyper-)sphere. The Index of Discernibility of the whole dataset is calculated as the number of elements having discernibility higher than the threshold  $th = 0.5$  divided by the total number of elements. The pseudocode of the Index of Discernibility can be seen below.

This heuristic measure should not be confused with the discernibility tables, or the indiscernibility concept, used in Rough Sets ([12, 13]). The latter are completely different both in function and in field of application.

Next, we present two independent feature selection methods. Although of similar philosophy, they differ in their structure and in the way they employ the Index of Discernibility.

---

Authors are with the School of Computer Science and Information Systems, Birkbeck College, University of London, Malet Street, London, WC1E 7HX (telephone: +44 20 7763 2110, fax: +44 20 7242 2754, email: {zacharias, gmagoulas}@dcs.bbk.ac.uk)

- 1 Initialization: set  $l=0$  (class variable),  $i=0$  (pattern variable),  $th=0.5$  (threshold of discernible element)
- 2  $N \leftarrow$  number of patterns of dataset
- 3  $q \leftarrow$  number of classes of dataset
- 4 do  $l \leftarrow (l + 1)$
- 5  $C_l \leftarrow$  patterns belonging to class  $l$
- 6  $D_l \leftarrow$  distance matrix of class  $l$  using the Euclidean distance
- 7  $r_l \leftarrow$  average distance between 2 elements of the  $l$ -th class.
- 8 until  $l = q$
- 9 do  $i \leftarrow (i + 1)$
- 10  $b \leftarrow$  class of  $i$
- 11  $d \leftarrow$  distances of element  $i$  from all other elements of dataset
- 12  $n \leftarrow$  (number of elements for which  $d \leq r_b$ )
- 13  $c \leftarrow$  number of elements belonging to class  $b$  for which  $d \leq r_b$
- 14  $z_i \leftarrow c / n$
- 15 until  $i = N$
- 16 ID  $\leftarrow$  number of elements for which  $z_i \geq th$

Algorithm 1 – Pseudocode for calculating the Index of Discernibility

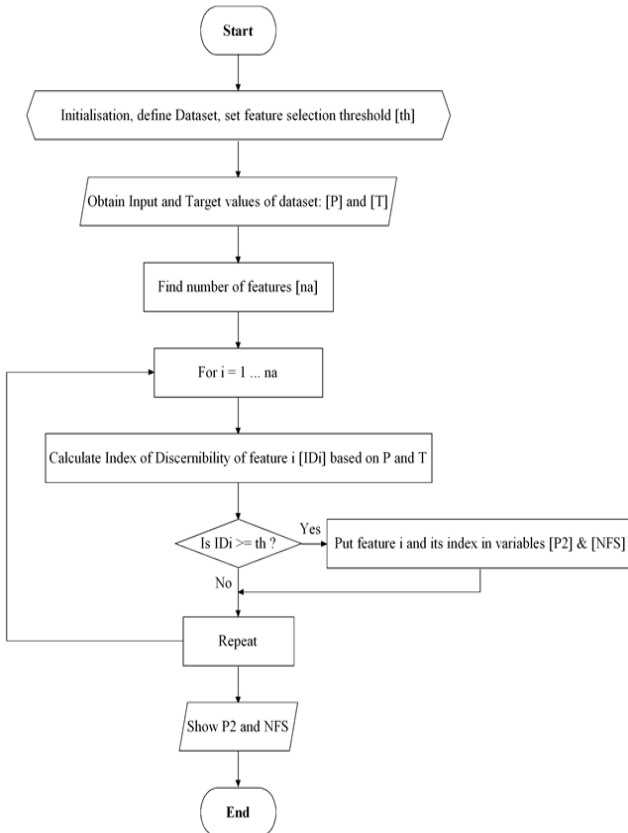


Fig. 1. Flow chart of the first ID-based feature selection method (fs3).

#### A. The fs3 Method

The first method, fs3, is the simplest and fastest in its operation. As it can be seen from its flow chart in Fig. 1, it

makes an assessment of each one of the features of the dataset, using the Index of Discernibility, and then selects the ones that are of a given standard. The latter is expressed by the threshold  $th$ , which is given by the user. Note that since this is an absolute parameter, the number of features at the new feature set heavily depends on the dataset itself. However, a value ranging from 0.7 to 0.85 is a good choice for the  $th$  parameter. The outputs of this method are the reduced feature set (P2), as well as a list of the indices of these features (NFS).

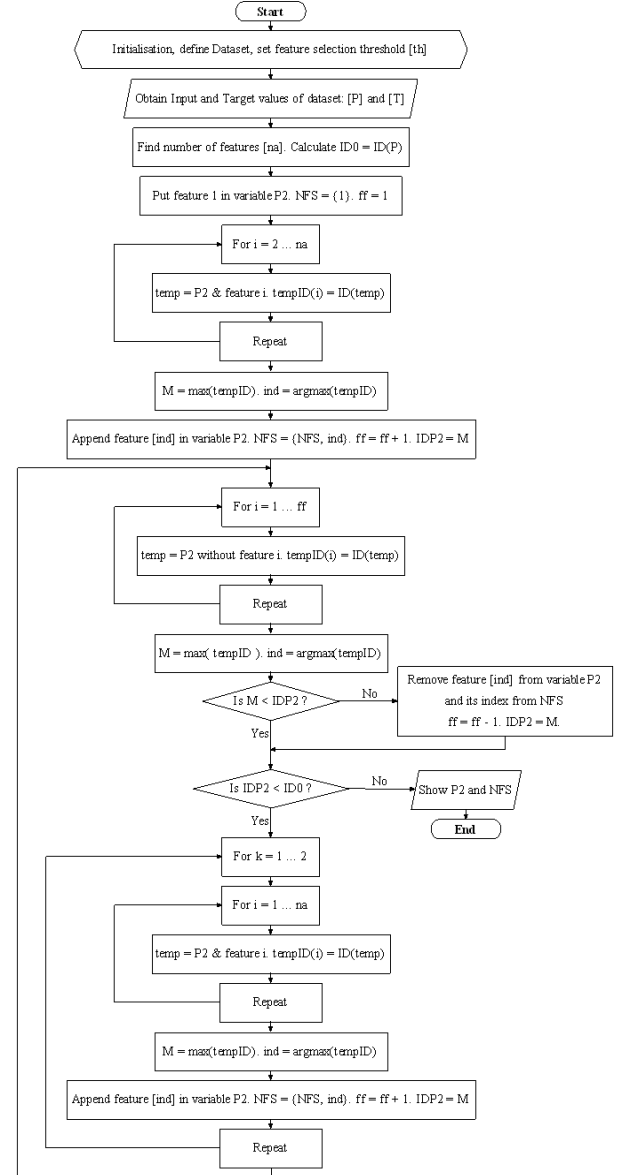


Fig. 2. Flow chart of the second ID-based feature selection method (fs10).

#### B. The fs10 Method

The second method, fs10, is quite different in its operation and somewhat more complex. However, it has the advantage that it is entirely automatic, as there is no need to set a threshold parameter for the selection of the features. As it can be seen in Fig. 2, where its flow chart is shown, fs10 gradually builds the new feature set (P2) by initially taking the first feature of the dataset and then adding two features at a time, so as to *maximise the Index of Discernibility of the whole feature set*. In other words, the features are not evaluated one by one, but as a group,

something which, although more time-consuming, is a better way of selecting the features, since it takes into account their relationship. Afterwards, the algorithm checks whether the Index of Discernibility is increased by removing one of the features of P2 or remains at least the same. This step helps prevent the accumulation of redundant features in the new feature set. By repeating this process (adding 2 new features and removing 1, if there is a redundancy) until the Index of Discernibility ceases to increase, this method goes through the rest of the available features. Alike the previous method, its outputs are the reduced feature set (P2), as well as a list of the indices of these features (NFS). This method tends to yield smaller reduced feature sets and often takes longer.

### III. EXPERIMENTS AND DISCUSSION

In this section, we present two sets of experiments. The first set concerns testing the proposed feature selection schemes in terms of their dimensionality reduction capabilities, whilst the second set investigates the performance of five classification algorithms based on the reduced feature sets.

The datasets used in our experiments were: the Diffuse Large B-Cell Lymphoma (DLBCL), the Colon Tumor (colon), the Central Nervous System (CNS) and the ALL-AML Leukaemia (leukaemia) problems. All of them were taken from [11] and are representative of a class of bioinformatics applications that employ microarray data. Also, these datasets are often used in the literature for feature selection [1-9]. Their characteristics are shown in Table I (this table also presents the dimensionality of the reduced feature sets that will be discussed in the next subsection). The large number of feature in the original datasets stems from the fact that these data involve gene expression profiles, which are by nature of this size. Datasets with dimensionality between 2000 to 4000 features are actually already reduced. Lastly, all of the features are numeric and continuous.

All experiments were carried out on a 1.8 GHz CPU processor with 2 GB of RAM. The OS platform was a Linux Ubuntu system. During the experiments no other applications were running.

Below, we start with Section III.A where we present the results of the two feature selection methods introduced in the paper. The reduced feature sets are then used in the rest of the section to train and evaluate various classifiers: Section III.B describes the classifiers trained and the measures for their performance evaluation, whilst Section III.C discusses the results and compares them with results produced when the same classifiers were trained on the complete feature set.

#### A. Results of Feature Selection Methods

The aim of this set of experiments was to investigate whether the methods can cope with different datasets without fine tuning. So we decided not to experiment with different threshold values in order to establish some kind of “optimum” threshold for each dataset but to set the threshold value used in the fs3 method equal to 0.75 (this means that any feature having an ID less than 0.75 is omitted from the new feature set). Although this may not

be the optimum value, results exhibited in Table I were quite good in general.

As it can be seen from Table I, the reduction of the feature set for each one of the four datasets is dramatic. Particularly after the application of the fs10 method, the number (#) of features is small compared to the original ones. This is translated to smaller complexity of the datasets (as it will be seen later on), and significantly less need for storage space.

TABLE I  
DATASETS CHARACTERISTICS AND SIZES OF REDUCED FEATURE SETS

Dataset	Patterns #	Original features #	Features # after fs3	Features # after fs10
DLBCL	47	4026	45	4
Colon	62	2000	11	4
CNS	60	7129	19	4
Leukaemia	72	7129	112	2

#### B. Classification Experiments with the Reduced Feature Sets

In this set of experiments we investigated whether the use of the reduced feature sets affects the quality of the classification. To this end we used a variety of classifiers and evaluation measures.

The classifiers used in our experiments are the k Nearest Neighbour (kNN) [10], a variations of kNN (V-kNN) [14], the Linear Discriminant Analysis (LDA) method [10], the Gravity Model Classifier [15], and the Fuzzy kNN classifier [16] (we used the MATLAB implementation of the Fuzzy kNN that was produced by Emre Akbas in December 2006).

The kNN classifier is one of the simplest and oldest classification methods in the literature. It is very popular due to its simplicity and speed. It works by examining the elements that are in proximity to the test patterns, in the feature space of the dataset. Due to its nature, it is heavily dependent on the geometry of the feature space.

The V-kNN is a variation of the kNN classifier. It stands out from the other variations developed so far in that it does not need the k parameter which has been considered an essential input for kNN to operate. Also, compared to the other variations of kNN, it is fast and very accurate. Just like kNN, it significantly depends on the geometry of the feature space.

The LDA is a quite established classification method, which works by creating a linear model that covers all the training patterns and then applies it to the testing ones. As the procedure includes a matrix inversion at some stage (involving dimensionality that is equal to the number of features of the dataset), it is greatly influenced by the geometry of the feature space.

The Gravity Model Classifier (GMC) is an alternative classifier which is not similar to any of the other ones mentioned above. It involves finding the “gravitational pull” of each one of the elements of each class of the dataset and comparing their average, for deciding which class wins the classification. As the “gravity forces” are depended on the squares of the distances, this classifier depends a lot on the feature space of the dataset.

Lastly, the Fuzzy kNN is an interesting alternative of

kNN, employing the use of fuzzy sets in the classification process. The only parameter the user has to set is, like the original kNN, the number of neighbours used.

In both kNN and Fuzzy kNN, the k parameter used for all of the experiments was k=5. In preliminary experiments carried out, there has been observed little difference in the performance of the classifiers for different values of k.

Regarding the experiments, 50 runs were carried out, using the leave-one-out method. The reason for using this method of validation was the small number of patterns in each dataset (being 72 patterns at the most). Also, this method of validation is the one more widely used in the literature for datasets of this type.

The classifiers were evaluated with three different measures which are comprehensible and descriptive of their performance. These measures were Accuracy Rate, the Degree of Certainty and the CPU Time (measured in seconds). All of these measures were averaged over the 50 experimental rounds.

The Accuracy Rate is the widest used measure in the literature, expressing the ration of the classification hits over the total number of test patterns.

The Degree of Certainty (DC) is a generalisation of the concept of Certainty Factor, first presented in [17]. It applies to almost all types of classifiers and provides a measure of how certain a classification is, very much like the *a posteriori* probability yielded by probabilistic classifiers. For every test pattern it is calculated by the formula:

$$DC_i = \frac{\max(\text{classification score})}{\sum_{c=1}^{\# \text{ of classes}} \text{classification score}(c)}, \quad (1)$$

where  $i$  denotes the  $i$ -th pattern classified,  $c$  defines the class number and *classification score* is the score determining the “certainty” that the  $i$ -th pattern belongs to each one of the classes and is related to the structure of the classifier.

The DC takes values between 0 and 1 (inclusive) and the higher it is, the most certain the classification is, according to the classifier. Note that this is a relative measure and depends a lot on the classifier used.

The significance of the DC lies in the evaluation of the outputs of the classifiers and the comparison among different classifiers. This is because in some cases, the output may be more or less random, therefore having a relatively low DC. Also, the DC may give an insight about how “easy” to discriminate the dataset is, according to a particular classifier: if the average DC of the classification is quite high, this would suggest that the dataset is quite predictable, using that classifier. And if this happens for different classifiers, one can induce that it is predictable in general. The DC is quite easy to implement and is not computationally expensive. This makes it an efficient addition to many classifiers, making their outputs more useful.

The CPU Time is the interval between two consecutive checks of the CPU clock of the computer. It is considered to be more reliable than the time interval measured with a stopwatch by the user, because it takes into account only the CPU usage of the experiments themselves.

This way it is not influenced much by the other processes that may run simultaneously in the OS.

### C. Results of Classification and Discussion

Classification results with the original feature sets are presented in Tables II-VI. Note that for the LDA classifier, there are no results for the CNS and Leukaemia datasets. This is because due to very high computational demands, the system could not complete the relevant experiments.

As regards the overhead of the feature selection process, for the fs3 algorithm, the CPU time ranged from 6 to 23 seconds (depending on the dataset), while for the fs10 method, from 24 to 104 seconds.

TABLE II  
RESULTS FOR KNN USING THE ORIGINAL FEATURE SET

Dataset	Accuracy Rate	Degree of Certainty	CPU Time (sec)
DLBCL	0.5532	0.6511	1.0100
Colon	0.8387	0.8516	0.8900
CNS	0.6667	0.6767	3.0500
Leukaemia	0.9306	0.8861	4.6800

TABLE III  
RESULTS FOR V-KNN USING THE ORIGINAL FEATURE SET

Dataset	Accuracy Rate	Degree of Certainty	CPU Time (sec)
DLBCL	0.4894	0.8237	16.880
Colon	0.7903	0.9530	15.820
CNS	0.6333	0.8263	76.520
Leukaemia	0.9028	0.9746	144.45

TABLE IV  
RESULTS FOR FUZZY KNN USING THE ORIGINAL FEATURE SET

Dataset	Accuracy Rate	Degree of Certainty	CPU Time (sec)
DLBCL	0.5532	0.6548	0.9600
Colon	0.8548	0.8561	0.7200
CNS	0.6667	0.6806	3.1900
Leukaemia	0.9306	0.8871	4.7400

TABLE V  
RESULTS FOR LDA USING THE ORIGINAL FEATURE SET

Dataset	Accuracy Rate	Degree of Certainty	CPU Time (sec)
DLBCL	0.6383	0.7234	5381.4
Colon	0.5161	0.7736	882.40
CNS	-	-	-
Leukaemia	-	-	-

TABLE VI  
RESULTS FOR GMC USING THE ORIGINAL FEATURE SET

Dataset	Accuracy Rate	Degree of Certainty	CPU Time (sec)
DLBCL	0.5106	0.5771	1.7400
Colon	0.6452	0.6405	1.5300
CNS	0.6500	0.6227	5.7100
Leukaemia	0.6528	0.6926	8.3500

The results of the experiments using the reduced feature sets are exhibited in Tables VII-XVI and provide the average of the 50 runs. The first 5 of them show the results using the fs3 method whilst the rest using the fs10 method.

TABLE VII  
RESULTS FOR KNN USING THE REDUCED FEATURE SET OF FS3

Dataset	Accuracy Rate	Degree of Certainty	CPU Time (sec)
DLBCL	0.7234	0.8936	0.0600
Colon	0.8548	0.8871	0.0500
CNS	0.7167	0.7800	0.1200
Leukaemia	0.9444	0.9639	0.1700

TABLE VIII  
RESULTS FOR V-KNN USING THE REDUCED FEATURE SET OF FS3

Dataset	Accuracy Rate	Degree of Certainty	CPU Time (sec)
DLBCL	0.6809	0.9303	1.2900
Colon	0.9032	0.9480	2.2900
CNS	0.6833	0.8601	2.2200
Leukaemia	0.9444	0.9888	3.5500

TABLE IX  
RESULTS FOR FUZZY KNN USING THE REDUCED FEATURE SET OF FS3

Dataset	Accuracy Rate	Degree of Certainty	CPU Time (sec)
DLBCL	0.8723	0.8990	0.0900
Colon	0.9032	0.9011	0.0600
CNS	0.7000	0.7821	0.1400
Leukaemia	0.9444	0.9624	0.1900

TABLE X  
RESULTS FOR LDA USING THE REDUCED FEATURE SET OF FS3

Dataset	Accuracy Rate	Degree of Certainty	CPU Time (sec)
DLBCL	0.4255	0.7553	0.2000
Colon	0.7742	0.8104	0.1900
CNS	0.5667	0.6945	0.1100
Leukaemia	0.4861	0.7335	0.2600

TABLE XI  
RESULTS FOR GMC USING THE REDUCED FEATURE SET OF FS3

Dataset	Accuracy Rate	Degree of Certainty	CPU Time (sec)
DLBCL	0.7234	0.6951	0.1300
Colon	0.8387	0.6899	0.2500
CNS	0.6667	0.6437	0.2200
Leukaemia	0.6528	0.7954	0.3400

TABLE XII  
RESULTS FOR KNN USING THE REDUCED FEATURE SET OF FS10

Dataset	Accuracy Rate	Degree of Certainty	CPU Time (sec)
DLBCL	0.9362	0.8894	0.0700
Colon	0.8226	0.8710	0.0600
CNS	0.7833	0.7400	0.0900
Leukaemia	0.9861	0.9861	0.1400

As it can be observed from Tables VII-XVI the average CPU time of the classification is reduced. This is very important, considering that the feature selection itself takes a not neglectable amount of time. Thus, the reduction of the time involved in the classification (due to the simplicity of the new dataset) might make it worthwhile. Also, as one would expect, the second feature selection method (fs10) takes considerably more time, for all of the datasets. Yet,

this is understandable, as it is a more complicated method, working with groups of features instead of single features, at a time.

TABLE XIII  
RESULTS FOR V-KNN USING THE REDUCED FEATURE SET OF FS10

Dataset	Accuracy Rate	Degree of Certainty	CPU Time (sec)
DLBCL	0.9574	0.9811	1.2700
Colon	0.7581	0.9568	2.3200
CNS	0.7667	0.9285	2.1700
Leukaemia	1.0000	1.0000	3.0300

TABLE XIV  
RESULTS FOR FUZZY KNN USING THE REDUCED FEATURE SET OF FS10

Dataset	Accuracy Rate	Degree of Certainty	CPU Time (sec)
DLBCL	0.9362	0.9246	0.1000
Colon	0.8548	0.8915	0.0600
CNS	0.8500	0.7925	0.1100
Leukaemia	0.9861	0.9885	0.1800

TABLE XV  
RESULTS FOR LDA USING THE REDUCED FEATURE SET OF FS10

Dataset	Accuracy Rate	Degree of Certainty	CPU Time (sec)
DLBCL	0.8085	0.7872	0.1300
Colon	0.7903	0.8304	0.1900
CNS	0.6833	0.6896	0.0700
Leukaemia	0.9167	0.9004	0.0700

TABLE XVI  
RESULTS FOR GMC USING THE REDUCED FEATURE SET OF FS10

Dataset	Accuracy Rate	Degree of Certainty	CPU Time (sec)
DLBCL	0.9574	0.7384	0.0300
Colon	0.7419	0.7349	0.2200
CNS	0.7333	0.6770	0.2100
Leukaemia	0.9722	0.9036	0.2500

Another important point is that the average accuracy rate is significantly increased in most of the datasets, for most of the classifiers. This is something expected, since the original feature set contains a large number of redundant features which, for many of the classifiers, might make the classification process more difficult. So, by eliminating these features we end up with a relatively "easier" dataset to classify. This change is also depicted at the ID, which often is greater for the new datasets, something that rarely happens for smaller datasets (on the contrary, if from a dataset having a small number of features you diminish the feature set by merely one feature, the ID of the whole is bound to drop). This can be seen in Table XVII.

TABLE XVII  
INDEX OF DISCERNIBILITY VALUES FOR EACH DATASET, BEFORE AND AFTER THE FEATURE SELECTION PROCESS, FOR BOTH METHODS USED

Features	DLBCL	Colon	CNS	Leukaemia
Original	0.7872	0.8226	0.3833	0.8333
fs3-reduced	0.9362	0.8871	0.6500	0.9306
fs10-reduced	0.8936	0.8871	0.8667	1.0000

The fact that the datasets become simpler is backed up by another point, which can be observed in Tables II-XVI: the increase in the average Degree of Certainty, for almost all of the classifiers, in the various datasets. Particularly in the DLBCL dataset, the increase is quite significant.

Also it is noteworthy that the feature selection methods introduced allowed the LDA classifier to be applied to the CNS and Leukaemia datasets, which were unmanageable with the original feature set. This is also important when considering that many of the produced microarray datasets are of this dimensionality or even of a higher one.

A summary of the above results depicting which feature set was the best, for each one of the datasets and for each one of the classifiers is provided in Table XVIII. Note that a feature set was considered the best for a particular classifier when it outperformed the other feature sets in two or more of the evaluation measures used.

TABLE XVIII  
SUMMARY OF BEST FEATURE SET FOR EACH CLASSIFIER

Classifier	DLBC	Colon	CNS	Leukaemia
kNN	fs10	fs3	fs10	fs10
V-kNN	fs10	fs3	fs10	fs10
Fuzzy kNN	fs10	fs3	fs10	fs10
LDA	fs10	fs10	fs10	fs10
GMC	fs10	fs3	fs10	fs10

From Table XVIII, it can be seen that the reduced feature sets of both of the feature selection methods introduced here outperformed the original feature set, for all of the classifiers used. Particularly the feature set of the second method, fs10, dominated the other ones for three of the datasets, in spite of the fact that it is slower than the other feature selection method. However, as the reduced feature sets it created were significantly smaller, it managed to yield quite low CPU times on average.

The feature selection of the fs3 method appeared to be weaker in the case of LDA for one particular dataset (DLBCL), however the reduced feature set it yielded for this dataset seemed to work very well with all the other classifiers. So the weakness was because of a problematic generalisation of the LDA classifier, probably due to the presence of one or more useless features in the reduced feature set. So this raises the issue of whether the threshold of 0.75 for the fs3 method is reliable. It could be the case that different datasets require different threshold values in order to yield appropriately reduced feature sets. This would render the fs3 method a quite flexible alternative, which when fine-tuned, could perform equally well to the fs10 method. This issue will be investigated in our future work.

## REFERENCES

[1] H. Frohlich, and O. Chapelle, "Feature selection for support vector machines by means of genetic algorithms," *15th IEEE International Conference on Tools with Artificial Intelligence*, pp. 142-148, 2003.

[2] S. Chao, and C. Lihui, "Feature dimension reduction for microarray data analysis using locally linear embedding," *3rd Asia-Pacific Bioinformatics Conference*, pp. 211-217, Jan. 2005.

[3] L. Cao, C. K. Seng, Q. Gu, and H. P. Lee, "Saliency analysis of support vector machines for gene selection in tissue classification," *Neural Computing & Applications*, vol. 11, pp. 244-249, 2003.

[4] M. Lecoche, and K. Hess, "An empirical study of univariate and GA-based feature selection in binary classification with microarray data," *UT MD Anderson Cancer Center Department of Biostatistics Working Paper Series*, Working Paper 5, Mar. 2005. Available online at <http://www.bepress.com/mdandersonbiostat/paper5> (last accessed February 2008).

[5] G. Alexe, G. Bhanot, and B. Venkataraghavan, "A robust meta-classification strategy for cancer diagnosis from gene expression data", *IEEE Computational System Bioinformatics Conference*, pp. 322-325, 2005.

[6] D. K. Tasoulis, V. P. Plagianakos, and M. N. Vrahatis, "Unsupervised clustering in mRNA expression profiles," *Computers in Biology and Medicine*, vol. 36, pp. 1126-1142, 2006.

[7] S. Mukkamala, Q. Liu, R. Veeraghattam, and A. H. Sung, "Computational intelligent techniques for tumor classification (using microarray gene expression data)," *International Journal of Lateral Computing*, vol. 2(1), pp. 38-45, 2005.

[8] T. Sawa, and L. Ohno-Machado, "A neural network-based similarity index for clustering DNA microarray data," *Computers in Biology and Medicine*, vol. 33, pp. 1-15, 2003.

[9] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays", *Natl. Acad. Sci. USA*, vol. 96, pp. 6745-6750, Jun. 1999.

[10] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification (2nd ed.)*, John Wiley and Sons, 2001.

[11] Kent Ridge Bio-medical Data Set Repository, Available online at <http://sdmc.lit.org.sg/GEDatasets/Datasets.html> (last accessed January 2008).

[12] S. Tan, Y. Wang, and X. Cheng, "Text feature ranking based on rough-set theory," *IEEE/WIC/ACM International Conference on Web Intelligence*, 2007.

[13] J. Komorowski, Z. Pawlak, L. Polkowski, and A. Skowron, "Rough sets: a tutorial," *Rough-Fuzzy Hybridization: A New Method for Decision Making*, Springer-Verlag, Singapore, 1998. Available online at: <http://citeseer.ist.psu.edu/komorowski98rough.html> (last accessed February 2008).

[14] Z. Voulgaris, and G. D. Magoulas, "Extensions of the k nearest neighbour methods for classification problems," *26th IASTED International Conference on Artificial Intelligence and Applications*, CD Proceedings ISBN: 978-0-88986-710-9, pp. 23-28, Feb. 2008.

[15] D. Ruta, and B. Gabrys, "Physical field models for pattern classification," *Soft Computing Journal*, vol. 8(2), pp. 126-141, 2003.

[16] J. M. Keller, M. R. Gray, and J. A. Givens, Jr., "A fuzzy k-nearest neighbor algorithm", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15(4), pp. 580-584, 1985.

[17] T. Aydin, and H. Guvenir, "Modeling interestingness of streaming classification rules as a classification problem," *14th Turkish Symposium on Artificial Intelligence and Artificial Neural Networks*, pp. 168-176, 2006.