International Journal of Computational Intelligence Research.
ISSN 0973-1873 Vol.1, No.1 (2005), pp. 68–76
© Research India Publications http://www.ijcir.info

Aiding Classification of Gene Expression Data with Feature Selection: A Comparative Study

Changjing Shang and Qiang Shen

Department of Computer Science
University of Wales, Aberystwyth, UK

Abstract: **This paper presents an application of supervised machine learning approaches to the classification of the yeast S. cerevisiae gene expression data. Established feature selection techniques based on information gain ranking and principal component analysis are, for the first time, applied to this data set to support learning and classification. Different classifiers are implemented to investigate the impact of combining feature selection and classification methods. Learning classifiers implemented include K-Nearest Neighbours (KNN), Naive Bayes and Decision Trees. Results of comparative studies are provided, demonstrating that effective feature selection is essential to the development of classifiers intended for use in high-dimension domains. In particular, amongst a large corpus of systematic experiments carried out, best classification performance is achieved using a subset of features chosen via information gain ranking for KNN and Naive Bayes classifiers. Naive Bayes may also perform accurately with a relatively small set of linearly transformed principal features in classifying this difficult data set. This research also shows that feature selection helps increase computational efficiency while improving classification accuracy.**

I. Introduction

Recently developed DNA microarray experiment technology has resulted in expression levels of thousands of genes being recorded over just a few tens of different samples [3, 4, 9, 13, 16]. Such massive expression data gives rise to a number of new computational challenges. In particular, automated classification of gene expressions has become the subject of much research in order to determine the functionality of known or unknown genes [15, 24]. For example, significant work has been reported in the literature on classification of yeast S. cerevisiae gene expression vectors [4]. Amongst such work, an initial experiment was reported in [9, 10], with a conclusion that genes of a similar function yield similar expression patterns in microarray hybridization experiments. The results of this work showed that each of the five different predicted classes in [4] more or less forms a natural cluster. This and subsequent research (e.g. through the use of support vector machines [20] and linear discriminant analysis [17]) has shown that the yeast data set may be automatically classified if appropriate methods are exploited. Indeed, the existing literature has demonstrated that finding sets of genes with expression levels that allow class separation can be achieved, potentially, by the use of supervised or unsupervised learning mechanisms, including classifiers such as Naive Bayes, logistic regression, neural networks, and Gaussian mixture models [14, 18]. Although these methods are all candidates to perform pattern recognition tasks in general and yeast gene classification in particular, each has its own strengths and limitations. It is difficult, if not impossible, to predict which would give the best result. Therefore, it is useful to build different classifiers and to validate their performance on a common data set and subject to common criteria.

For the yeast data set that is under consideration in this paper, an important problem facing the development of a practically usable classifier is that there are a large number of genes with too many features included in the original data. Yet, some of these features may be irrelevant to the classification due to measurement redundancies or noise. Thus, selecting discriminatory genes can be critical to improving the accuracy and speed of the classification systems. This problem of *curse of dimensionality*, common in machine learning and pattern recognition, can be reduced via the assistance of dimensionality reduction, which is a process that chooses a smaller set of features from the set of original features or their transformed versions, according to certain criteria [6, 7, 22, 24]. For example, recent work of [24, 25] employed principal component analysis (PCA) [6, 8] to reduce the dimensionality of a different gene data set, showing that using only the first few so-called principal features to identify the most predictive genes works effectively.

Although PCA helps remedy the sensitivity of a classifier to high dimensionality, the transformation process of features involved irreversibly destroys the underlying meaning of the feature set. Further reasoning about the derivation from transformed principal features is almost always humanly impossible. As an alternative to transformation-based dimensionality reduction, feature selection offers a means of choosing a smaller subset of original features to represent the original data set. This approach has drawn much recent attention in data mining and knowledge discovery [12, 14]. For example, the work on prediction of molecular bioactivity for drug design [22] shows that feature selection without transformation may also help improve classification performance. It is, therefore, very interesting and necessary to investigate the impact of utilising strict feature selection techniques upon the task of classification. So is to reveal the additional benefits of reducing data dimensionality in the simplification of structural complexity of the learned classifiers. For these reasons, this paper presents a novel comparative study of different combinations of these dimensionality reduction techniques with popular pattern classifiers, in the context of gene expression vector classification.

The rest of this paper is organised as follows. Section 2 introduces the yeast data set, which both justifies the need for the kind of research carried out in this study and sets the scene for the experimental investigations reported later. Section 3 briefly reviews the dimensionality reduction and feature selection techniques used, namely, principal component analysis (PCA) and information gain-based feature ranking [18]. Section 4 presents the specification of three (supervised) learning algorithms adopted to build the classifiers in this work, namely, K-Nearest Neighbours [8], Naive Bayes [6, 8] and Decision Trees [18, 19]. Section 5 shows the results of applying the different classifiers, in conjunction with the use of dimensionality reduction and feature selection techniques, to the yeast data set. The comparative study of the experimental results is made: a) against the classification using the full set of original features, b) between the use of PCA and that of information gain-based feature selection, and c) between the three types of classifier, both in terms of overall classification accuracy and of classification performance at individual class level. Section 6 concludes the paper and points out further research.

II. Problem Case

The data set used for this research is obtained from the publicly available expression profiles maintained by Brown's group at Stanford University [4]. It consists of yeast S. cerevisiae gene expression vectors, involving 79 experiments on 2467 genes. As indicated in [4, 9, 15], this data set is quite noisy and contains a rather high number of missing values. That is, not all genes have the entire set of the 79 measurements because each experiment was performed on a different subset of genes. Within this paper, these missing experimental values are represented by 0.0 (as with what is done in the literature, see [15]). The database only defines a total of 224 gene functional classes and thus, most of them (about 90%) are undefined. As such, the data set is very unbalanced, there are only a few positive examples for each of the 6 classes (Histones, Proteasome, Cytoplasmic Ribosomal Proteins, Respiration Chain Complexes and Tricarboxylic-acid Pathway), and most of the genes do not belong to any of these six. Furthermore, there are some genes that belong to a certain class, but have different expression levels; and there are genes that do not belong to the class which they share prediction level patterns with. These cases will unavoidably lead to false negatives and false positives in classification. It is because of such difficulties possessed by this data set that it has been chosen to carry out the present study, with an aim to check the potential of combining different classification and feature selection techniques. A more detailed overview of this data set can be found in [4, 15].

Table 1 lists the definition of those defined six functional classes and the corresponding class labels for the first five classes (that are to be used by the classifiers later). The first five classes are selected for this work because they represent categories of genes that are expected, on biological grounds, to exhibit similar expression profiles. They count for a total of 208 functional defined genes. Along with those undefined classes, the sixth functional class of Helix-turn-helix is removed from the data set due to the fact that it does not constitute an actual function class [4, 15]. Otherwise, with these biased data being present, there would be little chance for any classification algorithm to obtain a good classification performance [21]. This would make comparative studies as carried out later pointless. Therefore, only 208 genes (interchangeably treated as attributes or features hereafter) that have 79 experimental values, which jointly involve 5 functional classes (the first 5 classes of Table 1), are used.

Functional classes	Class labels for classifiers
Histones	1
Proteasome	2
Cytoplasmic ribosomal proteins	3
Respiration chain complexes	4
Tricarboxylic-acid pathway	5
Helix-turn-helix	
Undefined functional classes	

Table 1: Functional classes and their corresponding class labels.

III. Feature Selection and Dimensionality Reduction

Feature selection refers to the process of selecting descriptors that are most effective in characterising a given domain. It addresses the specific task of finding a subset of given features that are useful to solve the domain problem, without disrupting the underlying meaning of the selected features. In this regard, it is related to, but different from, the processes of variable dimensionality reduction and of parameter pruning, although its effect to data set dimensionality is the same as that of the latter. Below is a brief introduction to feature selection via information gain-based ranking [18] and dimensionality reduction via principal component analysis (PCA)

[6, 8], both of which are employed in the comparative studies later.

A. Information Gain Based Feature Ranking

Let X be an attribute and C be the class variable. The following equations define the entropy of the class before and after observing the attribute, respectively:

$$H(C) = -\sum_{c \in C} p(c) log_2 p(c)$$

$$H(C|X) = -\sum_{x \in X} p(x) \sum_{c \in C} p(c|x) log_2 p(c|x)$$

where x is a feature value and c is a class label.

The amount by which the entropy of the class decreases after observing an attribute reflects the additional information about the class provided by that attribute and is called information gain: $IG = H(C) - H(C|X)$. In other words, it measures how well a given feature separates the observed instances according to their given class categories. Without losing generality, suppose that there are N attributes (or features): $X_1, X_2, ..., X_N$. Each attribute $X_i, i = 1, 2, ..., N$, is assigned a score based on the information gain over the class entropy due to observing itself:

$$IG_i = H(C) - H(C|X_i)$$

The ranking of the attributes is then done with respect to the values of IG_i in a descending order, reflecting the intuition that the higher an IG value, the more information the corresponding attribute has to offer regarding the class. Note that to compute the information gain, data sets with numeric features are required to be discretised (with continuous variables quantified using real-valued intervals). Many alternative methods can be applied for this. In the present work, the method given in [11] is used.

B. Principal Component Analysis (PCA)

Principal component analysis is a standard statistical technique that can be used to reduce the dimensionality of a data set. This is done by projecting the data of a dimensionality N onto the eigenvectors of their covariance matrix with, usually, the largest M eigenvalues taken ($M < N$). More formally, each so-called principal component $PC_i, i = 1, 2, ..., M$, is obtained by linearly combining the original attributes (or features) such that

$$PC_i = \sum_{j=1}^{M} b_{ij} X_j$$

where X_j is the jth original attribute, and b_{ij} are the linear factors (eignvectors) which are chosen so as to make the variance of the corresponding PC_i as large as possible.

In implementation, the transformation from the original attributes to principal components is carried out through a process by first computing the covariance matrix of the original attributes and then, by extracting its eigenvectors to act as the principal components. The eigenvectors specify a linear mapping from the original attribute space of dimensionality N to a new space of size M in which attributes are uncorrelated. The resulting eigenvectors can be ranked according to the amount of variation in the original data that they account for. Typically, the first few transformed attributes account for most of the variation in the data set and are retained, while the remainder are discarded.

Note that in contrast with the information gain-based feature ranking, PCA is an unsupervised method which makes no use of information embodied within the class variable. Also, what the PCA returns are linear combinations of the original features. Therefore, the meaning of the original features is not preserved. As opposed to this, selecting a subset of top-ranked features based on information gain ranking will preserve the original meaning of those features selected, performing feature selection in its strict sense.

IV. Classification Algorithms

As indicated previously, three supervised learning algorithms are adopted here to build models in order to perform gene classification, namely, K-Nearest Neighbours (KNN) [8], Naive Bayes [6, 8], and Decision Tree [18, 19]. This is workable because class labels of the training examples are available for use in the search for separating genes. To be self-contained, this section gives a brief overview of these algorithms.

A. K-Nearest Neighbour (KNN)

KNN is one of the simplest learning/classification algorithms, and has been successfully applied to a broad range of problems [15]. To classify an unclassified vector X, the KNN algorithm ranks the neighbours of X amongst a given set of N data (X_i, c_i), $i = 1, 2, ..., N$, and uses the class labels c_j $(j = 1, 2, ..., K)$ of the K most similar neighbours to predict the class of the new vector X. In particular, the classes of these neighbours are weighted using the similarity between X and each of its neighbours, where similarity is typically measured by the Euclidean distance metric (though any other distance metric may also do). Then, X is assigned the class label with the greatest number of votes among the K nearest class labels.

The KNN classifier works based on the intuition that the classification of an instance is likely to be most similar to the classification of other instances that are nearby to it within the vector space. Compared to other classification methods such as Naive Bayes', KNN does not rely on prior probabilities, and it is computationally efficient if the data set concerned is not very large. If, however, the data sets are large (with a high dimensionality), each distance calculation may become quite expensive. This reinforces the need for employing PCA and information gain-based feature ranking to reduce data dimensionality, in order to reduce the computation cost.

B. Naive Bayes

A Naive Bayes classifier can achieve relatively good performance on classification tasks [7, 18], based on the elementary Bayes' Theorem. It greatly simplifies learning by assuming that features are independent given the class variable. More formally, a Naive Bayes classifier is defined by discriminant functions:

$$f_i(X) = \prod_{j=1}^{N} P(x_j|c_i)P(c_i)$$

where $X = (x_1, x_2, ..., x_N)$ denotes a feature vector and $c_j, j = 1, 2, ..., N$, denote possible class labels.

The training phase for learning a classifier consists in estimating conditional probabilities $P(x_j|c_i)$ and prior probabilities $P(c_i)$. Here, $P(c_i)$ are estimated by counting the training examples that fall into class c_i and then dividing the resulting count by the size of the training set. Similarly, conditional probabilities are estimated by simply observing the frequency distribution of feature x_j within the training subset that is labelled as class c_i. To classify a class-unknown test vector, the posterior probability of each class is calculated, given the feature values present in the test vector; and the test vector is assigned to the class that is of the highest probability.

C. Decision Trees

Different methods exist to build decision trees, which summarise given training data in a tree structure, with each branch representing an association between attribute values and a class label. The most famous and representative amongst these is, perhaps, the C4.5 algorithm [19, 24]. It works by recursively partitioning the training data set according to tests on the potential of attribute values in separating the classes. The core of this algorithm is based on its original version, named the ID3 [18, 19]. So, to have a basic understanding of how this algorithm works, the ID3 method is outlined below.

The decision tree is learned from a set of training examples through an iterative process, of choosing an attribute (i.e. feature) and splitting the given example set according to the values of that attribute. The key question here is which of the attributes is the most influential in determining the classification and hence should be chosen first. Entropy measures or equivalently, information gains (see section 3.1) are used to select the most influential, which is intuitively deemed to be the attribute of the lowest entropy (or of the highest information gain). In more detail, the learning algorithm works by: a) computing the entropy measure for each attribute, b) partitioning the set of examples according to the possible values of the attribute that has the lowest entropy, and c) for each subset of examples repeating these steps until all attributes have been partitioned or other given termination conditions met. In order to compute the entropy measures, frequencies are used to estimate probabilities, in a way exactly the same as with the Naive Bayes approach. Note that although attribute tests are chosen one at a time in a greedy manner, they are dependent on results of previous tests.

V. Experimental Results

A large corpus of experiments has been carried out. To better organise the presentation of the results the experimental background is first given below, a comparative analysis of the results is then presented.

A. Experimental Background

Different classifiers are used to accomplish classification by mapping feature patterns of a different dimensionality onto their underlying functional class types. There are a total of five output classes for the present problem case (see Table 1). The classification performance is measured using three-fold-cross-validation. That is, the gene expression vectors are randomly partitioned into three equally-sized subsets, and each subset is used as a test set for a classifier trained on the remaining two subsets. The empirical accuracy is given by the average of these three subset classifiers. A specific point worth noting is that for the KNN classifiers, the results on the use of a varying number K of nearest neighbours are obtained, with K set to 1, 3, 5, 8, 10, 12 and 15. The actual K value of a certain classifier is selected amongst those which leads to the best classification performance.

B. Comparison with the Use of Unreduced Features

It is important to show that, at least, the use of those features selected does not significantly reduce the classification accuracy as compared to the use of the full set of original features. This forms the first part of this experiment-based investigation. For the given data set, the information gain ranking-based feature selector returns the original 79 features, ranked in the descending order of the size of their corresponding information gain. In particular, the top-most ranked feature is the original feature of index 79, which has the largest information gain compared with the others; and the original feature of index 73 is of the lowest rank.

Figure 1 illustrates the classification performance using KNN, Naive Bayes and Decision Tree classifiers, in conjunction with the use of information gain ranking-based feature selection method. Each bar indicates the classification accuracy using a different classifier and a different number of selected original features. In this figure, for example, the left-most case involves the use of five original features (79, 57, 61, 78 and 68, listed in order of their ranks), and the right-most case shows the results from the use of the original full feature set (a total of 79 features). For comparison, Figure 2 shows the classification performance of using the same classification methods, with respect to the sets of features returned by PCA.

Clearly, different feature sets significantly affect the classification performance. The best performance for each type of classifier with its corresponding feature set is summarised in Table 2 (where for the case of combining Naive Bayes and information gain ranking, parts of the feature indices selected are omitted due to space limit). It is very interesting to note that the classification performance using selected features can be better than that of using the full feature set. In particular, for KNN and Naive Bayes classifiers, much better results can be obtained if careful selection of a subset of features is carried out. For instance, the employment of the top-ranked 13 features for KNN classifier beats the case when the full set of features is used by around 1.5% in terms of classification rate. Similarly, the use of the first 11 principal components allows approximately 1.5% improvement over the classification rate of using the full feature set.

Importantly, this improvement of performance is obtained by structurally much simpler classifiers, as compared to a classifier that requires the full feature set. The best KNN and Decision Tree classifiers only need 13 and 7 features to achieve the equivalent performance, and the best Naive Bayes classifier only requires 11 transformed features to outperform the classifier that uses the full set of features.

The above results are indicative of the power of feature selection in helping to reduce redundant feature measures and also the noise associated with such measurement (as fewer features may even lead to higher classification accuracy). This, in combination with the observation that the information gain-based feature selection preserves the underlying meaning of the selected features, also shows that information loss can be minimised and even avoided in building the classifiers if feature selection is carefully carried out.

C. Comparison between the Use of Information Gain-based and PCA-based Features

This study aims at examining the performance of using different dimensionality reduction techniques. Results given in Figures 1 and 2 and in Table 2 are reused for this analysis.

The fundamental advantage of preserving attribute meanings through the use of a strict feature selection approach like the information gain-based ranking, over the use of a transformation-based approach such as PCA, has been pointed out earlier. Here, experimentally, the effects of using either approach are evaluated in terms of classification accuracy. Overall, using a subset of original features seems to perform better than that of a set of transformed features. This can be seen by comparing Figures 1 and 2, noting the scale differences between the two figures. In general, both methods considerably reduce the computation cost for the classifiers (except for the integration of Decision Tree and PCA). When examined across the whole range of simulation results, Table 2 confirms that the classifiers using the features selected via information gain ranking have a higher classification accuracy. Additionally, it is worth recalling that PCA alters the underlying meaning of the original features during its transformation process. That is, for example, those features marked with 1, 2, ..., 5, with regard to the case of conjunctive use of KNN and PCA in Table 2, are not part of the original features, but their linear combinations.

Note that when the number of principal components becomes larger than 11, there is virtually no further improvement across all classifiers. This is different for classifiers that use features selected by the information gain-based method, although when the dimensionality of the feature subsets increases to a large number, say 40 (see Figure 1), the variation between the effects of selected features upon the individual classifiers also becomes less obvious.

The locally best classification results of Table 2 (with respect to individual types of classifier) reinforce the observation that using the information gain-based approach generally leads to a better classification. For instance, comparing with classifiers that use the original full feature set, the Naive Bayes classifier with 60 features results in the best performance (95.19%), and so does locally the KNN classifier which utilises only 13 originals (93.27%). However, for the local best amongst those classifiers which use PCA-returned features, the classifiers use an even smaller number of principal features (5 and 11, respectively). The reason that more selected features are needed for classification when working with the information gain-based method is probably because each of the principal components is itself a linear combination of all the original features already. Hence, the use of a seemingly smaller set of principal components actually includes information contributed by many original features. This of course shows the power of PCA, despite the fact that it irreversibly destroys the underlying meaning of the original features.

D. Comparison between Classifiers

This final part of the comparative study is set to investigate the differences between different classifiers, in terms of their classification ability. It is clear from Figures 1 and 2 that on average, over the large corpus of experiment carried out, Naive Bayes and KNN classifiers tend to significantly outperform the Decision Tree classifiers. This is probably due to the fact that the former two types of classifier work directly on numerically-valued data without the need for discretisation, whilst Decision Tree learning requires partitioning the underlying domain into a set of symbolic values (even though they may be represented as real-valued intervals).

More particularly, the very best classification result (95.19%) using features chosen by either information gain ranking or PCA are achieved by Naive Bayes'. This, together with the above observed overall performance, demonstrates the effectiveness of Naive Bayes classifiers. In addition to having a higher classification rate, this kind of classifier in general has a computational advantage over its KNN and Decision Tree

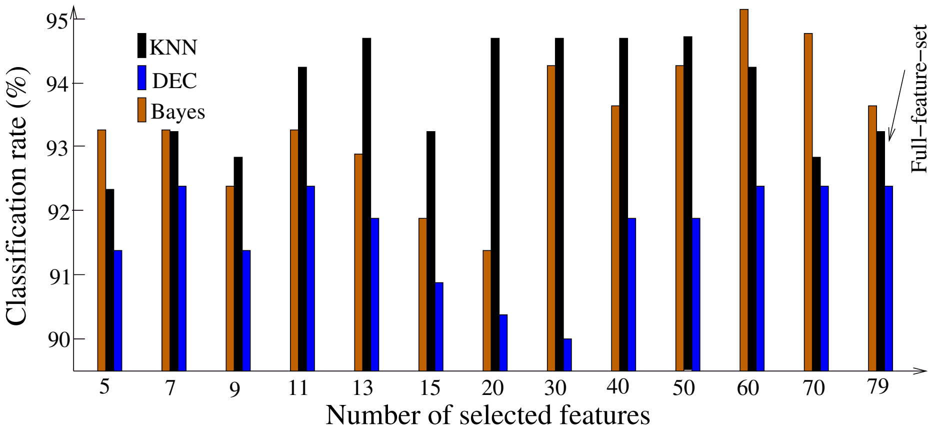

Figure. 1: Performance of KNN, Decision Tree and Naive Bayes over a different number of information gain-based features.

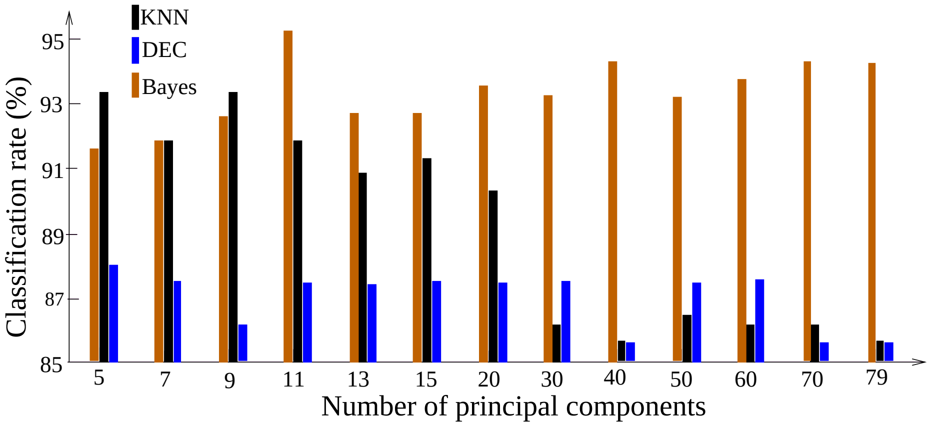

Figure. 2: Performance of KNN, Decision Tree and Naive Bayes over a different number of principal components.

counterparts in terms of complexity. However, there is “no free lunch”. Such a superior performance requires the use of many more features.

No matter which type of classifier to use, employing a smaller number of selected features significantly reduces the computational cost, for both training and classification phases. This is especially important to the KNN classifiers since they require measuring distances between a new instance and the existing data points. Fortunately, as discussed previously, dimensionality reduction does not necessarily reduce the accuracy of learned models. For the present data set, reduced feature sets actually can lead to higher classification rates. As a matter of fact, the locally best classification results are all obtained using reduced feature sets for Naive Bayes and KNN classifiers. Even for the relatively poor classifiers obtained by Decision Tree learning, using a mere 7 selected features can equal the performance of using the full 79 features.

The discussion above has considered the overall classification performance using different classifiers and feature selection methods. It is, however, useful to investigate the classification performance on each class in further detail. For this, Table 3 lists the classifiers’ performance over each class (only the locally best results for a particular type of classifier are given). To facilitate comparison, the results using the full set of original features are also listed in this table.

Clearly, at individual class level, except for class 3, Decision Tree-based classifiers had the worst performance (which jointly led to the worst overall performance observed earlier). For classes 2 and 4, KNN classifiers gave the best results (94.23% and 100.0%). Both KNN and Naive Bayes classifiers provided the same best result for class 1 (90.9%). KNN and Decision Tree gave quite poor results for class 5, but Naive Bayes (with 60 selected features using information gain) provided a 100.0% correct classification rate. Finally, it is worth noting that the classifiers with full original features only perform as well as the classifiers that employ a smaller set of features for classes 1 and 3. Incidentally, these results also help to reveal the relative difficulties in classifying instances which may belong to different classes of a data set.

VI. Conclusion

This paper has presented an experiment-based comparative study of three classification methods applied to the yeast S. cerevisiae gene expression data set. The work is itself novel as feature selection methods are, for the first time, employed in conjunction with the learning process of each classifier considered to address the difficulties in handling real problems represented by this data set. It has shown that in general, attribute selection is beneficial for improving the performance of these common learning algorithms. It has also shown that, as with the learning algorithms, there is no single best approach for all situations involving dimensionality reduction or feature selection. This investigation has helped to reinforce the fact that when building a practical classifier, what is needed is not only an understanding of how different learning algorithms work, but also when they work the best with what kind of support attainable from feature selection, as well as what background knowledge is available about the data in the given domain.

In particular, this work has investigated the following three classification algorithms: K-Nearest Neighbours, Naive Bayes and Decision Trees; and the following two methods for making choice of features: information gain-based ranking and linear transformation-based principal component analysis. Comparative studies have been performed between the use of full feature set and that of a subset; between the employment of different types of learning algorithm in building classifiers; and of course, between the utilisation of dimensionality reduction techniques that choose features in the strict sense (i.e. not altering any form of the original features) or through transformation (i.e. changing the representation of original features).

Amongst a large corpus of systematic experimental studies carried out, the best classification accuracy is achieved by using a subset of features chosen by information gain-based method for KNN and Naive Bayes classifiers. Naive Bayes can also do well with a relatively small set of features linearly transformed by PCA, in performing classification of this difficult data set. This may be due to the use of PCA; by transferring features into linear combinations of original attributes, PCAhelps to alleviate the independence assumption made by this type of classifier. Another point worth noting is that not only classification accuracy but also computational efficiency is improved through dimensionality reduction or feature selection. Results of such studies from this realistic application show the success of this research.

This work, nevert.5heless, only considered three types of classifier and two types of feature selection method (even though the choice of the latter two techniques have been carefully done such that one follows the strict feature selection approach and the other works via variable transformation). There are many possible alternatives (e.g. SVM [4, 20]) that have been applied to this data set among others, though no feature selection was involved. It is very interesting, and would be potentially very beneficial to the relevant research communities, to investigate such alternatives and compare their performance with the ones studied here, including the use of most recently developed feature selection techniques (e.g. [12]). In addition, the present work is focussed on a highly specific and unbalanced data set. It would also be useful to examine this data set more carefully, in terms of missing values and inconsistent relationships. This will help extend the useful findings to other problem domains.

References

[1] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer and Z. Yakhini. Tissue classification with

gene expression profiles. Journal of Computational Biology, 7:559–584, 2000.

[2] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. Artificial Intelligence, 97(1-2):245-271, 1997.

[3] A. Brazma and J. Vilo. Gene expression data analysis. FEBS Letters, 480:17-24, 2000.

[4] M. Brown, W. Grundy, D. Lin, N. Cristianini, C. Sugnet, T. Furey, M. Ares Jr. and D. Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. Proceedings of National Academy of Science, 97(1):262-267, 2000.

[5] M. Dash and H. Liu Feature selection for classification. Intelligent Data Analysis, 1(3):131-156, 1997.

[6] P. Devijver and J. Kittler. Pattern Recognition: A Statistical Approach. Prentice Hall, 1982.

[7] P. Domingos and M. Pazzani. Feature selection and transduction for prediction of molecular bioactivity for drug design. Machine Learning, 29:103-130, 1997.

[8] R. O. Duda and P. E. Hart. Pattern Classification and Scene Analysis. Wiley, 1973.

[9] M. Eisen, P. Spellman, P. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. Proceedings of National Academy of Science, 95:14863–14868, 1998.

[10] M. Eisen, P. Spellman, P. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. On-line supplement: http://genome-www.stanford.edu/clustering/Figure2.txt

[11] U. M. Fayyad and K. B. Irani. Multi-interval discretisation of continuous-valued attributes. Proceedings of the 13th International Joint Conference on Artificial Intelligence, 1022-1027, 1993.

[12] R. Jensen and Q. Shen Semantics-preserving dimensionality reduction: rough and fuzzy-rough approaches. IEEE Transactions on Knowledge and Data Engineering, 16(12):1457-1471, 2004.

[13] G. Getz, E. Levine and E. Domany Coupled two-way clustering analysis of gene microarray data. Proceedings of National Academy of Science, 97(22):12079-12084, 2001.

[14] D. Hand, H. Mannila and P. Smyth. Principles of Data Mining. MIT Press, 2001.

[15] M. Kuramochi and G. Karypis. Gene classification using expression profiles: a feasibility study, International Journal on Artificial Intelligence Tools, 14(4):641-660, 2005.

[16] D. Lashkari, DeRisi, J. McCusker, A. Namath, C. Gentile, S. Hwang, P. Brown and R. Davis. Yeast microarrays for genome wide parallel genetic and gene expression analysis, Proceedings of National Academy of Science, 94:13057-13062, 1997.

[17] G. J. McLachlan. Discriminant Analysis and Statistical Pattern Recognition. Wiley, 2004.

[18] T. M. Mitchell. Machine Learning. McGraw-Hill, 1997.

[19] J. R. Quinlan. C4.5. Programs for Machine Learning. Morgan Kaufmann, 1993.

[20] V. Vapnik. Statistical Learning Theory. Wiley, 1998.

[21] G. Weiss and F. Provost. Learning when training data are costly: the effects of class distribution on tree induction. Journal of Artificial Intelligence Research, 19:315-354, 2003.

[22] J. Weston, F. Perez-Cruz, O. Bousquet, O. Chapelle, A. Elisseeff and B. Scholkopf. Feature selection and transduction for prediction of molecular bioactivity for drug design. Bioinformatics, 19:764-771, 2003.

[23] D. Wu, K. Bennettw and N. Cristianini Large margin decision trees for induction and transduction. Proceedings of the 6th International Conference on Machine Learning, 474-483, 1999.

[24] E. P. Xing, M. L. Jordan and R. M. Karp Feature selection for high-dimensional genomic microarray data. Proceedings of the 18th International Conference on Machine Learning, 601-608, 2001.

[25] M. Xiong, L. Jin, W. Li and W. Boerwinkle. Computational methods for gene expression-based tumor classification. Journal of Bio Techniques, 29(6):1264-1270, 2000.

Author Biographies

Changjing Shang received her B.Sc. degree in Communications and Electronic Engineering from the National University of Defence Technology, China, her M.Sc. degree (with distinction) in Informatics from the University of Edinburgh, UK, and her Ph.D. degree in Computing and Electrical Engineering from Heriot-Watt University, UK. Dr Shang is currently conducting her research within the Department of Computer Science at the University of Wales, Aberystwyth. Her research interests include statistical and neural computation, adaptive algorithms, and pattern recognition, with their applications in large-scale signal and image modelling and analysis. She has published 30 peer-refereed articles in academic journals and conferences.

Qiang Shen received his B.Sc. and M.Sc. degrees in Communications and Electronic Engineering from the National University of Defence Technology, China, and his Ph.D. degree in Knowledge Based Systems from Heriot-Watt University, UK. Prof. Shen is the Director of Research with the Department of Computer Science at the University of Wales, Aberystwyth, UK, and an Honorary Fellow at the University of Edinburgh, UK. His research interests include descriptive uncertainty modelling, model-based inference, pattern recognition, and knowledge refinement and reuse. He serves as an associate editor or editorial board member of several academic journals in his research area, including IEEE Transactions on Fuzzy Systems, Fuzzy Sets and Systems, and Computational Intelligence Research. Prof. Shen has published 160 peer-refereed articles in international journals and conferences.

Classification Method	Selection Method	Dimension -ality	Feature Index	Classification Rate
KNN(K=1)	InfoGain	13	79,57,61,78,68,60,58,49,50,67,56,62,51	94.71%
KNN(K=1)	PCA	5	1, 2, 3, 4, 5	93.27%
KNN(K=1)	Full	79	From 1 to 79	93.27%
DEC. Tree	InfoGain	7	79,57,61,78,68,60,58	92.31%
DEC. Tree	PCA	5	1, 2, 3, 4, 5	87.98%
DEC. Tree	Full	79	From 1 to 79	92.31%
NaiveBayes	InfoGain	60	79,57,61,78,68, ..., 20, 28, 27, 7, 65	95.19%
NaiveBayes	InfoGain	30	79,57,61,78,68, ..., 3, 9, 40, 25, 14	94.23%
NaiveBayes	PCA	11	From 1 to 11	95.19%
NaiveBayes	Full	79	From 1 to 79	93.75%

Table 2: Classification rate vs. feature sets.

Classification Method	Selection Method	Dimen- sionality	c1	c2	c3	c4	c5	Overall Rate
KNN(K=3)	InfoGain	13	90.9%	94.3%	100.0%	100%	42.9%	94.71%
KNN(K=1)	Full	79	81.8%	94.3%	100.0%	85.2%	57.1%	93.27%
DEC. Tree	InfoGain	7	81.8%	88.6%	100.0%	81.5%	64.3%	92.31%
DEC. Tree	Full	79	81.8%	88.6%	100.0%	85.2%	57.1%	92.31%
NaiveBayes	InfoGain	60	90.9%	88.6%	98.3%	88.9%	100.0%	95.19%
NaiveBayes	PCA	11	90.9%	85.7%	100.0%	96.3%	78.6%	95.19%
NaiveBayes	Full	79	90.9%	88.6%	97.5%	85.2%	92.9%	93.75%

Table 3: Classification rates over individual classes.