Building classification models from microarray data with tree-based classification algorithms

Peter J. Tan, David L. Dowe and Trevor I. Dix

Clayton School of Information Technology, Monash University, Melbourne, Australia

Abstract. Building classification models plays an important role in DNA mircroarray data analyses. An essential feature of DNA microarray data sets is that the number of input variables (genes) is far greater than the number of samples. As such, most classification schemes employ variable selection or feature selection methods to pre-process DNA microarray data. This paper investigates various aspects of building classification models from microarray data with tree-based classification algorithms by using Partial Least-Squares (PLS) regression as a feature selection method. Experimental results show that the Partial Least-Squares (PLS) regression method is an appropriate feature selection method and tree-based ensemble models are capable of delivering high performance classification models for microarray data.

1 Introduction

DNA microarrays measure a large quantity (often in the thousands or even tens of thousands) of gene expressions of several samples simultaneously. The collected data from DNA microarrays are often called microarray data sets. Advancing statistical methods and machine learning techniques have played important roles in analysing microarray data sets. Results from such analyses have been fruitful and have provided powerful tools for studying the mechanism of gene interaction and regulation for oncological and other studies.

Among much bioinformatics research concerned with microarray data, two areas have been extensively studied. One is to design algorithms to select a small subset of genes most relevant to the target concept among a large number of genes for further scrutinising. Another popular research topic is to construct effective predictors which are capable of producing highly accurate predictions based on diagnosis or prognosis data.

However, due to the nature of the collection of microarray data, a microarray data set usually has a very limited number of samples. In a typical gene expression profile, the number of gene expressions (input variables) is substantially larger than the size of samples. Most standard statistical methods and machine learning algorithms are unable to cope with microarray data because these methods and algorithms require the number of instances in a data set to be larger than the number of input variables. Therefore, many machine learning articles have proposed modified statistical methods and machine learning algorithms tailored to microarray analyses. As such, many proposed classification algorithms

for microarray data have adopted various hybrid schemes. In these algorithms, the classification process usually has two steps, which we now outline.

In the first step, the original gene expression data is fed into a dimensionality reduction algorithm, which reduces the number of input variables by either filtering out a larger amount of irrelevant input variables or building a small number of linear or nonlinear combinations from the original set of input variables. The former approach is often known as *variable selection* while the latter is often known as *feature selection*. In the second step, classification models are trained on the data set with a reduced number of input attributes (created in the previous step) using an ordinary supervised classification algorithm.

In principle, many dimensionality reduction algorithms for supervised learning can be applied to the classification of gene expression data. Various two-step schemes have been presented and all of them reported improved classification accuracy. However, in practice, end results are dependent on the combination of the dimensionality reduction algorithm and the classification algorithm. There is no conclusion from previous studies so far which confirms superiority of any particular scheme for microarray data classification.

In this study, we attempt to improve predictive accuracy by building a hybrid classification scheme for microarray data sets. In the first step, we implement two different dimensionality reduction schemes: (i) Partial Least-Squares (PLS) regression [1, 2] as the dimensionality reduction algorithm, and (ii) an alternative and novel hybrid feature selection scheme which consecutively applies the discretization method from [3] on the original data sets followed by the PLS regression algorithm. Then in the second step, the two sets of filtered data with new features resulting from the two feature selection schemes described in the first step are separately fed into tree-based classification algorithms. We then use these two schemes (in Tables 3 and 4 respectively) to compare the results from four tree-based classification algorithms - C4.5 [4], AdaBoost [5, 6], Random Forests [7] and MML Decision Forests [8].

2 Dimensionality Reduction for Microarray data

As discussed in the introduction, various dimensionality reduction algorithms have been proposed for the task of dimensionality reduction of microarray data. Fayyad and Irani's discretization method [3] discretises continuous-valued attributes by recursively applying an entropy minimisation heuristic. Tan and Gilbert [9] applied this method to filter out irrelevant genes for classifications. They also compared the single decision tree-based classification algorithm C4.5 with ensemble classification algorithms Bagging and AdaBoost, they concluded that ensemble methods often perform better than a single classification algorithm, especially in classifying gene expression data. Similar claims can also be found in [10, 11].

Evolutionary algorithms have also been applied in some classification algorithms for microarray data sets. Jirapech-umpai [12] implemented an evolutionary algorithm proposed by Deutsch [13] for multiclass classification. The study

intended to investigate the problem of searching the optimal parameters for the evolutionary algorithm which will generate the optimal number of predictive genes among the initial gene pool. The performances of the algorithm are measured by testing on the leukemia and the NCI60 data sets. They concluded that good results can be achieved by tuning up the parameters within the evolutionary algorithms.

Díaz-Uriarte and Alvarez de Andrés used random forests [7] as the dimensionality reduction algorithm as well as the algorithm for classification of microarray data. The proposed scheme trains random forests iteratively. At each iteration, the number of input variables is reduced by discarding those variables with the smallest variable importance. They showed that their new gene selection procedure selects small sets of genes while maintaining high predictive accuracy.

Statistical methods in multivariate analysis such as Partial Least-Squares (PLS) regression [1, 2] and Principal Component Analysis (PCA) have also been adopted for feature selection for microarray data. Nguyen and Rocke [14] conducted a numerical simulation study on the PLS and the PCA methods for microarray-based classification. They concluded that when being applied as the dimensionality reduction method for classification algorithms, PLS out-performs PCA with microarray data.

Although feature selection methods do not explicitly select a subset of genes most relevant to the target concept, attempts have been made to interpret the results of feature selection methods. Roden et al. presented a method [15] for identifying subsets of biologically relevant genes by using a combination of principal component analysis and information-theoretic metrics. Connection between PLS dimensionality reduction and gene selection was examined by Boulesteix [10]. The study found that the order of the absolute values of the coefficients for the first PLS component was identical to the order produced by the classical BSS/WSS ratio gene selection scheme.

3 Related Algorithms

3.1 Principal Component Analysis Regression and Partial Least Squares Regression

Principal Component Analysis (PCA) reduces the dimension of the original data space by projecting original data points to a new coordinate system of the same dimensionality, and then restricting this. The principle components (PC) are orthogonal and calculated by running the nonlinear iterative partial least squares (NIPALS) algorithm, which in turn maximizes the variance on each coordinate sequentially. So the i^{th} PC is given by

$$w_i = \underset{w^T w=1}{\operatorname{argmax}}\, var\{w^T x\},$$

subject to $t_i^T t_j = 0$, where $i \neq j, t_k = w_k^T x$. The idea behind PCA is to discover and retain those characteristics which contribute most to its variance. As such,

the dimension of the data set can be reduced by keeping the lower order (small i) PCs while omitting the higher order (large i) PCs.

Partial least squares (PLS) regression aims to reduce the data dimensionality with a similar motivation, but differs from PCA by adopting a different objective function to obtain PLS components. Whereas PCA maximises the variance of each coordinate and whereas both PCA and latent factor analysis will not take into account the values of the target (dependent) attribute, the PLS regression model attempts to find a small number of linear combinations of the original independent variables which maximise the covariance between the dependent variable and the PLS components. (PLS uses the entire data set: input and target attributes.) So the i^{th} PLS component is given by

$$w_i = \underset{w^T w=1}{\operatorname{argmax}} cov\{w^T x, y\},$$

subject to $t_i^T t_j = 0$, where $i \neq j, t_k = w_k^T x$.

The PLS method can be illustrated by examining the following relations. Assuming X is an n x m matrix representing a data set of n instances with p independent variables, then if the number of PLS components is K, then the matrix X can be written as the summation of K matrices generated by outer products between vector t_i (which is often known as the score vector) and p_i^T (which is often called the load vector). The optimal number of PLS components, K, is usually determined by applying cross-validation methods on training data. The details of choosing the optimal K for this study can be found in sec. 4.2.

$$X = TP^T + E = \sum_{i=1}^{K} t_i p_i^T + E$$

In effect, the relation in the PLS model projects the data vectors X from the original p-dimensional space into a (much lower than p) K-dimensional space. In the same way, when PLS components are used in the regression, the relation between dependent variable y and PLS component t_i can be written as

$$Y = TBQ + F$$

where T is PLS components matrix, B is the coefficients vector so that TB is orthogonal, Q is the regression coefficients matrix, F is the residual matrix and $\|F\|$ is to be minimised.

Partial least squares regression can be regarded as an extension of the multiple linear regression model. It has the advantage of being more robust, and therefore it provides a good alternative to the traditional multiple linear regression and principal component methods. The original PLS method was proposed by Wold in the late 1960s and initially applied in the field of econometrics. Since then the method had been adopted in other research disciplines and been widely applied in many scientific analyses. SIMPLS [16] is an algorithm for partial least squares regression proposed by de Jong [16]. Compared to conventional nonlinear

iterative partial least squares (NIPALS)-PLS, SIMPLS runs faster and is easier to interpret. In SIMPLS, the PLS components are calculated directly as linear combinations of the original variables, which avoids the construction of deflated data matrices. An implementation of SIMPLS by Mevik as an add-on package for the R statistical environment was used in this study.

3.2 MML Oblique Trees and Decision Forests

MML oblique tree [17] is a multivariate decision tree classification algorithm. At internal nodes of MML oblique trees, the data is divided into two mutually exclusive sets by employing a linear discriminant function of input variables. An MML coding scheme encodes such a split, with the margin between the data and the separating hyperplane taken into account. The motivation behind such a scheme is to find a linear discriminant function with the optimal trade-off between fitting the data and simplicity. **Decision forests with MML oblique trees** [8] is an ensemble classification algorithm which at least matches and sometimes surpasses the "right"/"wrong" performance of Breiman's random forests [7]. The optimal candidate trees in decision forests (with overall lower MML coding) with high probabilistic prediction accuracy (low log-loss score) and smaller tree size (lower height with fewer leaf nodes) in MML Decision Forests are selected by the MML oblique trees algorithm. Compared to schemes with univariate trees (which cut on only one attribute at a time), using MML (multivariate) oblique trees offers potential to greatly increase the diversity of the inferred forest. A new weighted tree averaging scheme is also proposed. The scheme is based on Bayesian weighted tree averaging but uses a modified, smoothed prior on decision trees.

3.3 C4.5, AdaBoost and Random Forests

C4.5 [4] is a decision tree inference algorithm introduced by Quinlan. Similar to most decision tree learning algorithms, C4.5 adopts the divide-and-conquer approach to construct decision trees and the procedure is recursive in nature. The C4.5 classification tree algorithm runs fast and it is simple to implement. Therefore, C4.5 trees are often used as base learners in ensemble learning schemes like AdaBoost and random forests.

AdaBoost [6] iteratively re-samples the training set with adapted probabilities (or assigns adapted weights) over instances of the training set. In the end, the scheme gives a weighted average of the results returned by running the classification algorithms on the re-sampling sets. It works very well when the data is noise free and the number of training data is large. But when noise is present in the training sets, or the number of training data is limited, AdaBoost tends not to perform as well as Bagging and random forests.

Random forests [7] uses CART [18] as the base learner and employs several methods to generate a diverse ensemble. Each decision tree in a random forest is trained on a distinct and random data set re-sampled from the original training set, using the same procedure as bagging. While selecting a split at each internal

node during the tree growing process, a random set of features is formed by either choosing a subset of input variables or constructing a small group of variables formed by linear combinations of input variables. Random forests [7] have achieved "right"/"wrong" predictive accuracy comparable to that of AdaBoost and much better results on noisy data sets. Breiman also claimed and showed that AdaBoost is a form of random forest (algorithm) [7].

4 Experiments

4.1 Data sets

In this study, we select seven (mainly oncological) microarray data sets - Leukaemia, Breast cancer, Central nervous system (CNS), Colon tumour, Lung cancer, Prostate cancer and Prostate cancer outcome. All seven microarray data sets have binary output attributes and can be freely downloaded from the Gene Expression Datasets Collection. They have properties that are common in microarray data sets and have also been extensively tested in many previous studies. It makes comparisons with other approaches more convenient. Table 1 shows the summary of the data sets, which can also be found on the web site (http://sdmc.lit.org.sg/GEDatasets) and in [9].

Table 1. Summary of Datasets

Dataset	Number of attributes	Number of PLS components	Number of Instances (Training+Test)	(Binary) Class distribution
Leukaemia	7129	9	72	47:25
Breast cancer	24481	10	97	46:51
Central nervous system	7129	8	60	21:39
Colon tumour	7129	8	62	40:22
Lung cancer	12533	14	181	31:150
Prostate cancer	12600	12	136	77:59
Prostate cancer outcome	12600	5	21	8:13

4.2 Methodology

In this study, the dimensionality reduction scheme is implemented as follows. Each column of the training set is normalised, so that each column has a mean of zero and variance of one. The values of the binary target attribute are set to either 0 or 1. Specifying the number of components for the Partial Least Square Regression, then a PLS model for a training data set is built by feeding the original training set into the SIMPLS algorithm. The output scores of the PLS algorithm are regarded as the values of input variables and forms the training set for the classification algorithms. Similarly, the test sets for the classification

algorithm were obtained by feeding each instance of the original test data into the PLS model built from the training set.

Determining the optimal number of PLS components There is only one free parameter in the PLS algorithms - the number of components, m. There are extensive discussions on how to determine the optimal number of components. However, the goal for performing PLS on the training set in this study is not for regression, rather, the PLS method is applied as a procedure for data pre-processing for the decision tree-based classification algorithms. In our scheme, m is the number of input variables of the data sets to train decision tree and various ensemble learning algorithms.

One major advantage of leave-one-out cross-validation is that it retains the maximum number of data as training sets. As the number of samples in a typical microarray data set is small, we use leave-one-out cross-validations to find the optimal number of components m which will result in classification models with highest "right"/"wrong" predictive accuracies. For each pair of data set and learning algorithm, the PLS methods were repeated with various numbers of PLS components m which ranged from 2 to $4\sqrt{N}$. To reduce the computational cost, the number of PLS components is increased by 2 instead of 1 in each iteration. Then the numbers of PLS components leading to classification models with highest predictive accuracies are regarded as the optimal numbers, as shown in table 4.2.

Ten-fold cross-validation For each original data set, 100 pairs of training and test data sets are generated by repeating the 10-fold cross-validation method ten times. Then these 100 pairs of data sets are pre-processed by using procedures described at the beginning of this section. Then for each of 100 pairs of training and test sets which resulted from the above process, classification models were built and tested by using the four classification algorithms (C4.5 [4], AdaBoost [5, 6], Random Forests [7] and MML Decision Forests [8]) described at the end of the introduction.

Leave-one-out cross-validation By selecting one instance from a data set as a test set and using the rest of the data as a training set, N pairs of training and test sets were obtained for a data set with N instances by selecting each data instance only once as a test set. Then for each of the N pairs of training and test set, the experiments were conducted as the procedures described in subsection 4.2 immediately above.

4.3 Results and Discussions

Table 3 shows the classification performances of the four classification algorithms on seven microarray data sets, with the lowest classification errors for each data set highlighted, MML oblique forest achieves the lowest classification error in 5

Table 2. The optimal number of PLS components

Dataset	Single C4.5	Random Forest	C5.0 AdaBoost	MML Oblique Forest
Leukaemia	2	8	8	2
Breast cancer	14	10	10	10
Central nervous system	4	6	4	8
Colon tumour	2	4	2	4
Lung cancer	2	2	2	4
Prostate cancer	10	24	12	32
Prostate cancer outcome	18	18	6	6

out of 7 data sets while random forest performs best in the other 2 sets. The MML oblique forests, which ensemble oblique trees with optimal probabilistic prediction performance (see e.g., [17, sec. 3.1][8, sec. 4.2]), return excellent predictive accuracy on noisy data such as microarray data. On the other hand, C5 AdaBoost did not perform well on such noisy data sets. In general, all three decision tree-based ensemble classification algorithms achieve higher predictive accuracies than the single model based decision tree algorithm C4.5. For all seven data sets, the best performing ensemble learning algorithms have classification errors 12.7% to 70% (e.g.,$\frac{0.6-2.0}{2.0} = -0.7$) lower than those of C4.5. It clearly indicates that ensemble algorithms are better candidates for building classification models for microarray data sets. Such a conclusion can also be found in [9, 11].

Table 3. Predictive error (%) of classification algorithms, using PLS dimensionality reduction scheme ((i) from section 1)

Dataset	Single C4.5	Random Forest	C5.0 AdaBoost	MML Oblique Forest
Leukaemia	5.7	3.8	4.3	**3.3**
Breast cancer	34.8	**28.8**	32.1	30.8
Central nervous system	38.8	35.5	36.8	**34.1**
Colon tumour	19.1	15.3	17.3	**11.2**
Lung cancer	2.0	3.8	1.8	**0.6**
Prostate cancer	17.0	9.4	11.9	**8.7**
Prostate cancer outcome	35.0	**30.7**	48.5	46.8

When applying the PLS method directly on the whole gene set from the original data, our tests returned improved classification accuracies on three (Leukaemia, Lung Cancer and Prostate Cancer) data sets. However, the other tests returned lower "right"/"wrong" classification accuracies on other four data sets than those reported in Tan and Gilbert's paper [9]. In [9], a subset of the original set of genes was selected by using Fayyad and Irani's discretization

method [3]. For each of the four data sets with worse results in our study, only less than 5% of the original gene set was selected and used to build classification models. For each of the three data sets with improved results in this study, at least 14% of the original gene set were retained. The observation suggests a two-stage dimensionality reduction scheme. In the first stage, irrelevant genes are filtered out by using Fayyad and Irani's discretization method [3]. In the second stage, dimension of the data is further reduced by applying the PLS method on the data with reduced numbers of genes. We processed data on each of seven data sets using the above scheme, then we re-ran the experiments. These experimental results show that, in going from the PLS scheme in Table 3 to the hybrid scheme in Table 4, there are significant across the board increases in classification accuracy. In some data set like the lung cancer data set, the predictive accuracies were extremely high (something like 99.9%).

Table 4. Predictive error (%) of classification algorithms, using a hybrid dimensionality reduction scheme ((ii) from section 1)

Dataset	Single C4.5	Random Forest	C5.0 AdaBoost	MML Oblique Forest
Leukaemia	3.3	**1.9**	3.6	**1.9**
Breast cancer	21.9	18.4	20.4	**17.9**
Central nervous system	25.8	**21.7**	27.6	23.1
Colon tumour	12.3	15.7	15.8	**11.6**
Lung cancer	1.8	**0.1**	1.8	0.2
Prostate cancer	10.9	7.1	8.6	**6.6**
Prostate cancer outcome	32	**20.3**	30.3	28.1

5 Conclusions and Future Research

We conducted an extensive survey in the area of building classification models from microarray data with various tree-based classification algorithms. Experimental results show that in most cases, tree-based ensemble learning algorithms delivered classification accuracies equivalent to or better than those on the same data sets reported by other studies. Combined with the Partial Least-Squares (PLS) regression method, which is proved to be an appropriate feature selection method, tree-based ensemble learning algorithms are capable of building classification models with high predictive accuracies from microarray data. As the study shows that our hybrid feature selection scheme improves classification accuracies, one question immediately arises: will there be better hybrid schemes for the feature selection process for building tree-based classification models? Since the number of instances in the studied microarray data is small and the performances of many classification algorithms are sensitive to the number of training data, another interesting question is raised: when comparing predictive performances of various classification algorithms on microarray data, what is

the impact of adopting different methodologies such as ten-fold cross-validation, leave-one-out cross-validation and bootstrap [19]?

This work was funded by a Monash University Faculty of I.T. Small Grant. We thank Robert Jorissen of the Ludwig Institute for Cancer Research for valuable feedback.

References

1. Geladi, P., Kowalski, B.: Partial least-squares regression: a tutorial. Analytical Chimica Acta **185** (1986) 1–17
2. Höskuldsson, A.: PLS regression methods. Journal of Chemometrics **2**(3) (1988) 211–228
3. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: IJCAI. (1993) 1022–1029
4. Quinlan, J.R.: C4.5 : Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA, U.S.A. (1992) The latest version of C5 is available from http://www.rulequest.com.
5. Freund, Y.: Boosting a weak learning algorithm by majority. Information and Computation **121**(2) (1995) 256–285
6. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: International Conference on Machine Learning (ICML). (1996) 148–156
7. Breiman, L.: Random forests. Machine Learning **45**(1) (2001) 5
8. Tan, P.J., Dowe, D.L.: Decision forests with oblique decision trees. In: Lecture Notes in Artificial Intelligence (LNAI) 4293 (Springer), Proc. 5th Mexican International Conf. on Artificial Intelligence, Apizaco, Mexico (2006) 593–603
9. Tan, A.C., Gilbert, D.: Ensemble machine learning on gene expression data for cancer classification. Applied Bioinformatics **2** (2003) S75–S83
10. Boulesteix, A.L.: PLS dimension reduction for classification with microarray data. Statistical Applications in Genetics and Molecular Biology **3**(1) (2004)
11. Díaz-Uriarte, R., de Andrés, S.A.: Gene selection and classification of microarray data using random forest. BMC Bioinformatics **7** (2006) 3
12. Jirapech-umpai, T.: Classifying Gene Data Expression using an Evolutionary Algorithm. Master thesis, University of Edinburgh (2004)
13. Deutsch, J.M.: Evolutionary algorithms for finding optimal gene sets in microarray prediction. Bioinformatics **19** (2003) 45–52
14. Nguyen, D.V., Rocke, D.M.: On partial least squares dimension reduction for microarray-based classification: a simulation study. Computational Statistics & Data Analysis **46**(3) (2004) 407–425
15. Roden, J.C., King, B.W., Trout, D., Mortazavi, A., Wold, B.J., Hart, C.E.: Mining gene expression data by interpreting principal components. BMC Bioinformatics **7** (2006) 194
16. de Jong, S.: SIMPLS: an alternative approach to partial least squares regression. Chemometrics and Intelligent Laboratory Systems **2**(4) (1993) 251–263
17. Tan, P.J., Dowe, D.L.: MML inference of oblique decision trees. In: Lecture Notes in Artificial Intelligence (LNAI) 3339 (Springer), Proc. 17th Australian Joint Conf. on AI, Cairns, Australia (2004) 1082–1088
18. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification And Regression Trees. Wadsworth & Brooks (1984)
19. Efron, B.: Bootstrap methods: Another look at the jackknife. The Annals of Statistics **7**(1) (1979) 1–26