Classification by Ensembles from Random Partitions using Logistic Regression Models

A Dissertation Presented

by

NOHA LIM

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

December 2007

Stony Brook University

The Graduate School

Noha Lim

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

Hongshik Ahn – Dissertation advisor
Professor
Applied Mathematics and Statistics

Stephen J. Finch – Chairperson of Defense
Professor
Applied Mathematics and Statistics

Wei Zhu
Professor
Applied Mathematics and Statistics

Sangjin Hong – Outside member
Professor
Electrical & Computer Engineering
Stony Brook University

This dissertation is accepted by Graduate School

Lawrence Martin
Dean of Graduate School

Abstract of the Dissertation

Classification by Ensembles from Random Partitions using Logistic Regression Models

by

NOHA LIM

in

Applied Mathematics and Statistics

Stony Brook University

2007

A robust classification procedure is developed based on ensemble of classifiers. Each classifier is built using a logistic regression tree or logistic regression model fitted from a different set of predictors determined by a random partition of the entire set of predictors. The main goal of this study is to apply logistic regression models to a high-dimensional data set without variable pre-selection. For data with a smaller sample size than the feature space, variable selection is required to use a standard logistic regression model. The new method solves this problem by random partitioning of the feature space. The proposed method combines the results of multiple classifiers to achieve a substantially improved prediction compared to the optimal single classifier. This approach is designed specifically for high-dimensional genomic data sets for which a classifier is sought. We evaluate the performance of the proposed methods compared to widely used classification methods using five microarray data sets and simulation data sets. This study shows that the performance of the proposed method is consistently good in terms of overall accuracy. For unbalanced data, this approach maintains the balance between sensitivity and specificity more adequately than many other classification methods considered in this study. The proposed method can be applied to huge data sets of binary classification besides microarray data.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Data Mining and Classification

Data mining, so called knowledge discovery in database, is the technique of extracting meaningful information from large data sets. Due to the advancement of information technology, data mining of a huge database becomes an essential tool for making decisions. It is used in some areas as customer relation marketing using sales data, credit risk assessment using financial data, and diagnosis of patients using gene expression data.

One of the main functions of data mining is classification. Classification is the procedure to build a rule using the pre-defined classes and their features in historical data and apply this rule to a new data for discriminating each observation. Many problems in science and industry require this technique because of their complex and huge data sets.

Various tools and algorithms are used to perform classification. Three major sources are classical statistics, machine learning, and artificial intelligence. Due to the complexity of a problem, there is no superior approach that always performs best.

In statistics, Fisher's linear discriminant (1936) and logistic regression are classical and standard methods. Moreover, some modern techniques such as the

Bayesian approach and *k*-nearest neighbor have been developed. These statistical approaches are generally based on a probability model for each class.

Machine learning focuses on extracting rules from the data automatically. Decision tree algorithms belong to this category. They classify data using a series of logical splitters that divide the data into nodes. The genetic algorithm is another technique in this area.

A neural network is a classification algorithm in the field of artificial intelligence. It is a very powerful tool with the capability of pattern recognition. Many nodes are connected to each other and form a network. There are input, output, and interconnecting layers in this network, and they work like networks of neurons.

There are several issues to be addressed in classification problems. First of all, accuracy is a major issue because it represents the performance of the classifiers. However, we sometimes prefer a slightly less accurate model if it is much faster than a more computer-intensive model, because the time required to complete calculation may be a main concern in analyzing high-dimensional data. Another important issue is the balance between sensitivity and specificity. For example, positive response of the data is only 10% of sample size. We can get 90% accurate model if it classifies all the observations as negative. However this is not a good classification model. Although we may lose some correct classification of negative responses, we have to detect more positive cases carefully.

1.2 Classification of Microarray Data

Microarray is a widely used technique in cancer research. Recently it has been used for classification of cancers. In cancer treatment or therapy, early diagnosis and an accurate classification are very important. Thus, highly accurate statistical classification method is desired for these studies. However, one of the properties of microarray data is that there are many predictors with a small sample size. This makes microarray classification very difficult.

Support Vector Machines (SVM: Vapnik, 1995) were developed to bypass this difficulty of high-dimensional data. This approach extends the boundary by projecting the input space to a higher dimensional space. However, this method is sensitive to the choice of kernel and other specifications.

Ensemble method is another way to classify these high-dimensional data, and is gaining acceptance in the data mining community due to the significant improvement in accuracy (Breiman, 1996, 1998, 2001; Freund and Schapire, 1996). In ensemble methods, we can build a strong classifier by combining many weak classifiers (Hastie et al., 2001).

Recently two ensemble methods, boosting (Schapire, 1990, 2002) and bagging (Breiman, 1996), have been widely used. Both methods use a resampled learning set to build a base classifier. Boosting changes the distribution of the learning set based on previous classifiers and combines weak classifiers using weighted voting.

Quinlan (1996) mentioned that boosting sometimes fails, and the class distributions across the weight vectors become skewed.

Bagging uses a bootstrap sample to build each base classifier. Each sample is chosen randomly with replacement and the final decision is made by equal weight voting of these base classifiers. Random Forest (Breiman, 2001) is based on this algorithm and gaining recognition. It combines classification trees which are built by bagging and random subspace of the predictors. Due to their resampling algorithms, both bagging and boosting cause overlap of predictor variables among classifiers, and consequently high correlation among the base classifiers. Kuncheva et al. (2003) noted that low correlation between classifiers enables one to improve prediction accuracy in ensembles.

We propose a new ensemble-based approach for classification called CERP (Classification by Ensembles from Random Partitions). This method is designed specifically for high-dimensional data sets. By randomly partitioning the predictors to k mutually exclusive subspaces, we can avoid the problem of dimensionality. Since there is no overlap of chosen predictors among subspaces, CERP tends to have low correlation among classifiers. We use a logistic regression model or logistic regression tree as base classifiers. We combine the predicted values of these base classifiers by taking the average of the predicted values of the base classifiers.

In order to further improve the accuracy, we build multiple ensembles by randomly re-partitioning the feature space. As we build multiple ensembles, fresh new information is obtained by different partitions.

We investigate the performance of the CERP method, and compare it with commonly used methods including Random Forest (RF: Breiman, 2001), Support Vector Machines (SVM: Vapnik, 1995), Boosting (Schapire, 1990; Freund and Schapire, 1996, 1997), *k*-Nearest Neighbors (kNN), Shrunken Centroids (Tibshirani et al., 2002), Linear Discriminant Analysis (LDA) and single optimal trees (CART: Breiman et al., 1984; QUEST: Loh and Shih, 1997).

CERP is applied to several data sets: the prediction of estrogen receptor binding activity (Blair et al., 2000), detection of allelic expression of imprinted genes (Reik and Walter, 2001; Greally, 2002), classification of colon cancer (Alon et al., 1999), classification of acute leukemias into acute lymphoblastic leukemia (ALL) or acute myeloid leukemia (AML) based on each individual patient's gene-expression profile (Golub et al., 1999) and prediction of which breast cancer patients would benefit from adjuvant chemotherapy based on gene expression data (van't Veer et al., 2002).

1.3 Classification Trees and Logistic Regression

We introduce two well-known standard methods for classification of binary classes. They are CART (Brieman et al., 1984) and logistic regression. They perform reasonably well so that many researchers use them for analyzing their data. Furthermore, many new algorithms dealing with class prediction have been developed and derived from those two methods.

CART is a widely used decision tree algorithm developed by Brieman. It starts with a binary split which has exactly two branches from the parent node. When we make a rule for splitting the node in a tree, we need to evaluate the function for splitting. In CART, one calculates the GINI index for current node *c* given as,

$$Gini(c) = 1 - \sum_j p_j^2$$

where p_j is the probability of class j in c. The splitting rule is chosen to maximize the reduction in the GINI index.

If the variable has n distinct numerical values, there are n-1 possible splits. For a variable with n categories, there are $2^{n\text{-}1}$-1 possible splits. CART examines each and every possible split for all variables in each node. It takes one split from this exhaustive search based on GINI index.

To prevent over-fitting, CART employs a pruning method which is called minimal cost-complexity pruning. The purpose of this step is to build a right sized tree by estimating the true misclassification cost. First, CART builds a full grown tree and then cuts the pair of leaves sequentially. In each sub-tree, misclassification cost and cost-complexity value are calculated using 10-fold cross-validation. Finally, the CART algorithm chooses the final optimal tree using these values.

The advantage of CART is that one can obtain a very clear and explicit classification model. Furthermore, we can see the variables associated with the response using the attributes used to split. However, it is difficult to apply to high-dimensional data sets. Because there are too many features in high-dimensional data, CART often fails to capture some important features included in the data.

There are several tree-structured classification algorithms such as CHAID (Kass, 1980), C4.5 (Quinlan, 1993), or QUEST (Loh and Shih, 1997), and many researchers continue developing or modifying these methods to achieve a better performance. All these methods commonly split the data by making branches and children nodes, but they have their own rules of splitting and stopping.

Logistic regression is a standard statistical procedure for analyzing binary response data (y=0 or y=1, for example). The form of the model is

$$\ln\left(\frac{p_i}{1-p_i}\right) = x_i\boldsymbol{\beta}$$

where p_i is the probability that the response is 1 and β is a vector of regression coefficients. The difference of the logistic regression compared to the least-squares regression is that the response of logistic regression is 0/1 variable, and the equation predicts the log odds that the observation will be 1. When we estimate the parameters, the likelihood function

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}$$

where $p_i = e^{x_i\beta} / (1 + e^{x_i\beta})$, is used. The parameter estimates are obtained by maximizing this likelihood function. Unlike least-squares regression, there is no explicit formula for the estimation of parameters. Thus, iterative calculation is applied to estimate the parameters. One starts with an initial guess of the parameter values, fits the model iteratively, and perturbs it over and over in order to improve the estimation of the parameters. Finally one stops the iterations when the improvement of the model is less than a pre-set tolerance.

There is a restriction on this logistic regression. The number of variables must be less than the sample size. Thus, one cannot apply logistic regression directly to high-dimensional data sets. Variable selection is necessary before we apply the model to high-dimensional data. However, it is difficult to find the most significant and appropriate predictors.

Chapter 2

Enhancement of Class Prediction by Ensemble Voting Methods

Hastie et al. (2001) documents that an ensemble of a combination of weak classifiers is a powerful committee. This section illustrates how the ensemble voting methods enhance the accuracy of class prediction.

Suppose that *n* classifiers are independent among each other, where *n* is odd. Let X_i denotes a random variable indicating a correct prediction by the *i*th classifier. If the prediction accuracy of each classifier is *p*, then X_i ~ Bernoulli(*p*), and the number of accurate classifications by the ensemble majority voting method is $Y = \sum_{i=1}^{n} X_i$ ~ binomial(*n, p*). Let $n = 2k + 1$, where *k* is a nonnegative integer. Define $A_n = P(Y \geq k+1)$. Then the prediction accuracy of the ensemble classification by majority voting is

$$A_n = \sum_{i=k+1}^{n} {}_nC_i \, p^i (1-p)^{n-i} \tag{1}$$

Lam and Suen (1997) showed that $A_{2k+1} = .5$ for k = 0, 1, ... when p = .5; the sequence $\{A_{2k+1}\}$ is strictly increasing when p > .5; and $\{A_{2k+1}\}$ is strictly decreasing when $p < .5$.

If *n* is large, then $Y \xrightarrow{d} N(np, np(1-p))$ by the central limit theorem. After the continuity correction, the approximate probability of the prediction accuracy by a majority voting is

$$A_{2k+1} = A_n = P\left(Y \ge \frac{n+1}{2}\right) \simeq P\left(Z \ge \frac{\frac{n+1}{2} - np - \frac{1}{2}}{\sqrt{np(1-p)}}\right) = P\left(Z \ge \frac{\sqrt{n}(1-2p)}{2\sqrt{p(1-p)}}\right) \quad (2)$$

where $Z \sim N(0,1)$. Define $f(p) = (1-2p)/\sqrt{p(1-p)}$, $0 < p < 1$. Since $f(p) < 0$ for $.5 < p < 1$ and thus $\lim_{n\to\infty} \sqrt{n} f(p)/2 = -\infty$, we see that $\lim_{n\to\infty} A_n = 1$ in (2). Function $f(\cdot)$ is decreasing in (0, 1) because $f'(p) = -\{2[p(1-p)]^{3/2}\}^{-1} < 0$. The second derivative is $f''(p) = 3(1-2p)\{4[p(1-p)]^{5/2}\}^{-1}$. Note that $\lim_{p\downarrow 0} f(p) = \infty$, $f(.5) = 0$ and $\lim_{p\uparrow 1} f(p) = -\infty$. Also, $f(p)$ is convex on (0, .5) and concave on (.5, 1). This implies that the prediction accuracy of the ensemble voting method converges quickly to 1 when p is close to 1, while it converges slowly to 1 if p is slightly larger than .5.

If there is a correlation among classifiers, we can use the over-dispersed binomial model to calculate the prediction accuracy. The beta-binomial model is commonly used for deriving an over-dispersed binomial model (Williams, 1975). Let Y be binomial(n, p) and p itself be a random variable with mean μ. Then we can calculate the accuracy of the ensemble classification $P(Y \ge k+1)$ using the beta-binomial distribution. This model is restricted to the positive correlation ρ in order to satisfy Var(p) > 0. However, Prentice (1986) showed that the beta-binomial may be extended to cases where $\rho < 0$ for certain values. His extended beta-binomial model is valid when $\rho \geq \max\{-p(n-p-1)^{-1}, -(1-p)[n-(1-p)-1]^{-1}\}$.

Ahn et al. (2007) address that negatively correlated classifiers improve their prediction accuracy more rapidly than the independent classifiers, while the improvement slows down when the correlation increases. This result implies that CERP can improve the accuracy by avoiding high correlation caused by an overlap of the predictor variables.

Table 1 shows the improvement of prediction accuracy by ensemble majority voting. We assume that the accuracy of each classifier is the same in this approach. Since we partition the predictor space randomly, this assumption is reasonable. However, because of the random variation of the accuracy among classifiers, there is a difference between the estimation in this approach and the estimation based on the assumption of unequal accuracies. For classifiers with unequal accuracies, we calculate the correlation between two binary classifiers as given in Kuncheva et al. (2003). We examined the beta-binomial or extended beta-binomial models to estimate the correlations for the examples used in this study and found that they are quite close to the estimates obtained using the correlation between two binary classifiers given in Kuncheva et al.

If there is no constraint of equal accuracy of base classifiers and equal correlation among classifiers, Breiman (2001) showed that there is an upper

Table 1: Enhancement of the prediction accuracy by ensemble majority voting.

		p (prediction accuracy of each base classifier)						
n	*ρ*	0.50	0.55	0.60	0.70	0.80	0.90	0.95
3	-0.05	0.50	0.58	0.66	0.80	0.91	0.98	NA[a]
	0.00	0.50	0.57	0.67	0.78	0.90	0.97	0.99
	0.10	0.50	0.57	0.64	0.76	0.87	0.95	0.98
	0.30	0.50	0.56	0.62	0.73	0.84	0.93	0.97
7	-0.03	0.50	0.62	0.73	0.90	0.98	NA	NA
	0.00	0.50	0.61	0.71	0.87	0.97	1.00	1.00
	0.10	0.50	0.59	0.67	0.81	0.92	0.98	0.99
	0.30	0.50	0.57	0.63	0.75	0.86	0.94	0.97
15	-0.01	0.50	0.67	0.81	0.96	1.00	NA	NA
	0.00	0.50	0.65	0.79	0.95	1.00	1.00	1.00
	0.10	0.50	0.60	0.70	0.85	0.95	0.99	1.00
	0.30	0.50	0.57	0.64	0.76	0.87	0.95	0.98
25	-0.01	0.50	0.72	0.88	0.99	NA	NA	NA
	0.00	0.50	0.69	0.85	0.99	1.00	1.00	1.00
	0.10	0.50	0.61	0.71	0.87	0.96	0.99	1.00
	0.30	0.50	0.57	0.64	0.77	0.87	0.95	0.98
101	0.00	0.50	0.84	0.98	1.00	1.00	1.00	1.00
	0.10	0.50	0.62	0.73	0.89	0.97	1.00	1.00
	0.30	0.50	0.57	0.64	0.77	0.88	0.95	0.98

[a] not available using the extended beta-binomial model by Prentice (1986)

bound of the generalization error $PE^* \leq \overline{\rho}(1-s^2)/s^2$, where $PE^* = P_{\boldsymbol{X},Y}[mr(\boldsymbol{X},Y) < 0]$ is the limit for the generalization error when the accuracy of each base classifier is higher than .5 and s is the strength of the set of classifiers defined as $E_{\boldsymbol{X},Y} mr(\boldsymbol{X},Y)$ and $\overline{\rho}$ is the average correlation of the tree classifiers. The $mr(\boldsymbol{X},Y)$ is defined as

$$mr(\boldsymbol{X},Y) = P_\Theta(h(\boldsymbol{X},\Theta) = Y) - \max_{j \neq Y} P_\Theta(h(\boldsymbol{X},\Theta) = j)$$

where $h(\boldsymbol{X},\Theta)$ is a set of classifiers and Θ is the predictor space. This shows that the limit of the generalization error depends on the average correlation. Moreover, the ensemble accuracy converges to 1 if there is no correlation among the classifiers. However, there is a limitation in improvement of the accuracy. Because CERP uses a fixed number of disjoint subsets, the number of base classifiers is limited and there is a bound which is lower than 1 for the ensemble accuracy. Furthermore, when we increase the number of disjoint subsets (*n*), the accuracy (*p*) decreases due to a smaller number of predictors used in each subset. Thus the improvement of the ensemble accuracy is expected to be slower than Table 1. However, we can achieve a reasonably fast improvement in high-dimensional data, because a large number of base classifiers can be generated.

Chapter 3

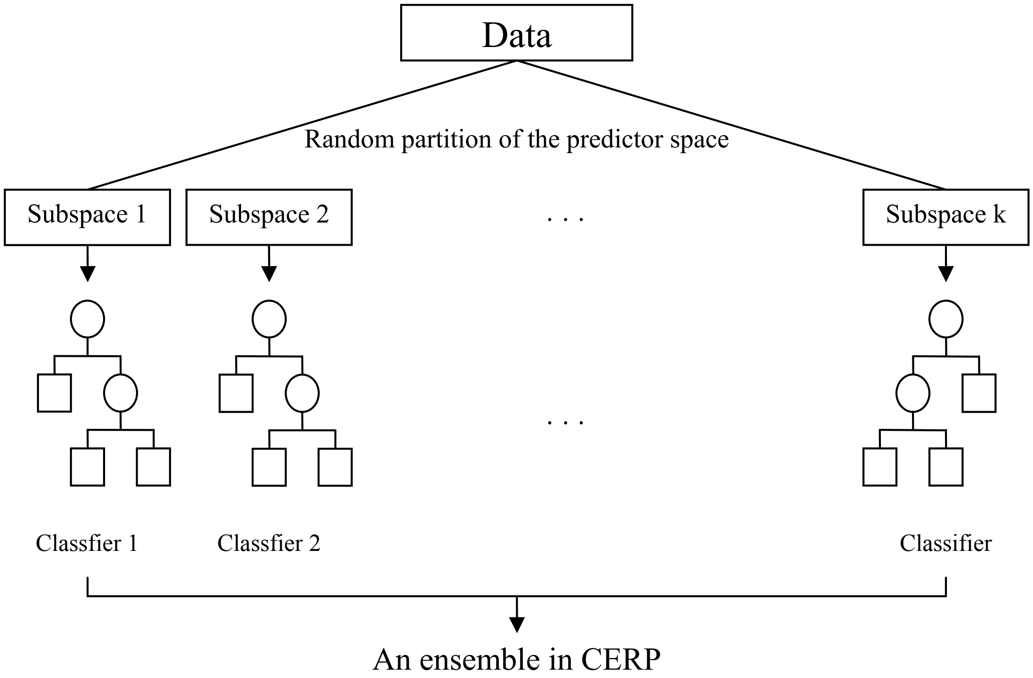
CERP: Classification by Ensembles from Random Partitions

3.1 Introduction

We propose a method for constructing CERP. The main idea of this method is to gain high accuracy of prediction by combining weak classifiers. Figure 1 illustrates the overall scheme of our approach. Let Θ be the space of the predictors. In order to minimize the correlation among the ensemble of classifiers, Θ is randomly partitioned into K subspaces ($\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$, …, $\boldsymbol{\theta}_\mathrm{K}$) with roughly equal sizes. Since the mutually exclusive subspaces are randomly chosen, we assume that there is no bias in selection of predictors in each subspace. In each of these subspaces, we build a single classification model using classification tree, logistic regression or logistic regression tree. CERP combines these weak classifiers by averaging the fitted values to gain an improvement of accuracy. Accuracies of these classifiers are expected to be similar due to the randomness of the partition. Thus we expect an improvement of the prediction accuracy as illustrated in Chapter 2. In order to improve further performance, we repeat this random partition. We classify each observation by a majority voting of these multiple ensembles.

Performance of CERP depends on the number of features in one partition, and

Figure 1: The proposed CERP approach.



the optimal partition size varies with the data. Thus, instead of using a fixed partition size, we search for an optimal partition size based on a nested 3-fold cross-validation (CV) in each learning sample in CV. We assume that the accuracy is unimodal as a function of the partition size. First, we partition the predictor space as each subspace has around *n/2* predictors, build a CERP model, and calculate its accuracy. In the same way, we try *n/3*, *n/4*, ..., *n/10* and *n/12*. The partition size resulting in the highest overall accuracy is chosen among these. Thus *n/i* will be chosen for some integer *i* = 2, ..., 10 or 12. The second step is to search the optimal size of the subspace by a dual bisection search between *n/i* and *n/(i-1)* and between *n/i* and *n/(i+1)* based on the overall accuracy. After this, we have two candidates. We take the one with higher overall accuracy.

In order to improve the balance between sensitivity and specificity, we search an optimal decision threshold for classification. This approach shares the same principle as the methods by Pazzani et al. (1994) and Domingos (1999). Instead of a threshold of 0.5, a high misclassification cost may be assigned by using the rate of the positive responses in the data as a threshold. When *r* is the rate of the positive responses, we classify a sample as 1 if the fitted value is larger than *r*, and classify it as 0 otherwise. The rate of the positive responses is not necessarily the optimal choice of the threshold in terms of balancing sensitivity and specificity. We found in this study that the optimal threshold usually lies between 0.5 and the rate of positive response.

To search for an optimal threshold, a nested 3-fold CV is performed in each learning set L_i, $i = 1, \ldots, 10$, of a 10-fold CV as follows:

Within L_i, we use a finite grid with an increment of 0.02 between 0.5 and r.

1. By applying each of the thresholds $ts_j = 0.50, 0.52, \ldots, r$ (or $ts_j = r, r + 0.02, \ldots, 0.48, 0.50$), conduct the following nested 3-fold CV: Construct a CERP classifier with one ensemble in each of the learning samples $L_{i(1)}$, $L_{i(2)}$, $L_{i(3)}$ and evaluate the accuracy using the corresponding test samples $T_{i(1)}$, $T_{i(2)}$, $T_{i(3)}$ by applying ts_j.
2. Choose a threshold with the highest prediction accuracy from part 1, say ts_i.
3. Apply ts_i to the test sample T_i corresponding to L_i.

Only one ensemble is used in this nested CV because of the tendency that the optimal threshold for CERP is similar for one or multiple ensembles.

3.2 C-T CERP: Classification Tree CERP

C-T CERP (Classification Tree CERP) was developed by Ahn et al. (2007). An optimal classification tree based on the CART algorithm is used as a base classifier. By using the C-T CERP, we can overcome the problem of a single CART tree applied to high-dimensional data as discussed in Section 1.3. C-T CERP can capture many features due to the random partitioned data, while single CART utilizes a limited number of variables. Moreover, fresh new information can be obtained by a different partition of the variables in each additional ensemble.

In CART, a tree is fully grown until the number of samples in each node is less than or equal to 5 or there is only one class left in the node. This tree is pruned by progressively deleting branches and the misclassification cost is calculated in each subtree using 10-fold cross-validation. A sub-tree with the smallest size whose misclassification cost is less than the smallest estimated CV error plus 1-standard error is chosen as the final tree (1-SE rule: see Breiman et al., 1984). In Ahn et al. (2007), a tree program written in C is used to build each CART tree in C-T CERP. In this study, *rpart* (Therneau and Atkinson, 1997) package in R is used to build a CART classifier in C-T CERP for a comparison of classification methods.

3.3 LR-T CERP: Logistic Regression Tree CERP

As an alternative to C-T CERP, we developed LR-T CERP (Logistic Regression Tree CERP) using a combination of the CART algorithm and logistic regression.

In LR-T CERP, we build a classification tree using the 1-SE rule. The classification tree (*rpart*) in the R package is used for splitting and pruning an optimal size of the base tree for each subspace. We use the same option of *rpart* as in C-T CERP. We add another option for pruning the fully grown tree. If the terminal node contains only one class, we prune that node with the sibling node together. This step is to prevent failure of fitting the logistic regression model.

In each terminal node, we fit the full logistic regression model. When there are fewer observations than predictors, univariate logistic regression models including the intercept term is fit with each predictor, and the *n-2* predictors with smaller deviances plus the intercept term are chosen to be included in the model. However, we observed that the sample size was larger than the number of predictors in the terminal node most the time.

In CART, a proportion of positive responses in the leaf node is assigned as a fitted value. Thus, the samples in the same node have the same fitted values, and the result may not be influenced by thresholds for decision in C-T CERP. However, the fitted values are distinct in logistic regression tree in the same node. Thus, the performance of LR-T CERP depends on the decision threshold.

3.4 LR CERP: Logistic Regression CERP

We developed LR CERP as an alternative to C-T CERP. As a parametric counterpart to the classification tree, logistic regression can be used. Logistic regression is the most widely used method in statistics for binary classification. However, there is a restriction that the number of predictors must be less than the number of observation. Thus, we encounter a problem to select the variables among thousands of variables in high-dimensional data. CERP can be used to solve this problem because it partitions the predictor space into smaller subsets.

The goal of this study is, by combining results from a widely used logistic regression model, to develop a classifier which is comparable to other aggregation methods in terms of the prediction accuracy and the balance between sensitivity and specificity.

Based on the CERP algorithm, we developed LR CERP by using logistic regression models as base classifiers. Since we partition the data so that each partition contains fewer number of variables than the sample size, and consequently we do not need to select variables. The full logistic regression model can be fitted in each partition. We of course can perform the variable selection using AIC or other criteria. However, it makes the computation more complex and slower, and the improvement of accuracy may not be substantial. The *glm* function in R is used to fit logistic regression model.

LR CERP combines the results of multiple logistic regression models to achieve an improved accuracy of class prediction by taking the average of the predicted values within an ensemble. The predicted values from all the base classifiers in an ensemble are averaged and classified as either 0 or 1 using a threshold discussed in Section 3.1. Although a majority voting and averaging methods are fundamentally similar, the latter gave slightly better prediction accuracy for LR CERP in this study.

Chapter 4

Existing Methods

We compared the performance of CERP with the existing classification methods listed in this chapter. Twenty repetitions of 10-fold CV are performed in the comparison. Details of the comparison are given in Chapter 6.

4.1 Random Forest (RF)

RF was developed by Breiman (2001) and is available as a package named RandomForest in R. RF is an ensemble of single CART trees which are based on the values of a random vector of the feature space sampled independently and with the same distribution for all trees in the forest. It takes each bootstrap sample to generate a tree and uses this as a training set (bagging). The remaining sample serves as a test set. It also chooses a random subset of the predictors to find a split at each node. There is an option called *ntree* which is the number of trees to grow. The default value of *ntree* = 500 works well. The number of features selected randomly at each node may also vary. Ahn et al. (2007) show that square root of the number of predictors gives good results in many data sets. The threshold for a decision does not notably change the prediction.

4.2 Support Vector Machines (SVM)

SVM (Vapnik, 1995) projects the input space into a higher dimensional feature space. In this high dimensional space, SVM builds a linear classifier. We used the SVM function of the e1701 package in R. The choice of kernel and the parameters for the transformations are sensitive to the prediction. We selected the linear kernel and the default option of the radial basis kernel.

4.3 Boosting Methods

AdaBoost is introduced by Freund and Schapire (1997). It makes many base trees (decision “stumps”: trees with a single split) and gathers them with majority voting. The base trees are constructed from the same data set by giving weights based on the previous classifier’s fit. If the current classifier is wrong, the weight given to this classifier is required in the next iteration. Schapire et al. (1998) found that boosting tends to increase the margins which are related to the generalization error, and a large positive margin means a greater probability of correct classification. LogitBoost algorithm is similar to AdaBoost. The main difference is that it uses the logit loss function.

AdaBoost, LogitBoost, L2Boost and BagBoost are available in the R boosting package. Since the result of BagBoost is similar to that of L2Boost, we do not include it in the results discussed here. The *mfinal* option is the number of iterations of weighted voting. We tried 30 to 100 iterations in this study.

4.4 kNN: *k*-Nearest Neighbor Classifiers

kNN is based on the distance among the data points, and the Euclidean distance is used in this study. The kNN algorithm decides the class of a new data point using the *k*-Nearest Neighbors in the learning set. The class from the majority voting of these *k* neighbors is assigned to a new observation. kNN has been shown to be a consistent classifier (Friedman 1997; Dudoit, Fridlyand and Speed 2002).

We used the kNN function of “class” package in R. For a good prediction, variable pre-selection is required. We follow the method of Dudoit et al. (2002) by taking the highest BW ratio, which is the ratio of between group to within group sums of squares for each feature. For a variable *j*, BW ratio is defined as

$$\frac{BSS(j)}{WSS(j)} = \frac{\sum_i \sum_k I(y_i = k)(\bar{x}_{kj} - \bar{x}_{.j})^2}{\sum_i \sum_k I(y_i = k)(\bar{x}_{ij} - \bar{x}_{kj})^2}$$

where $\bar{x}_{.j}$ is the average expression level of variable *j* across all sample and $\bar{x}_{kj}$ is that of variable *j* across samples belonging to class *k*. An optimal number of variables (*p*) and the value of *k* were searched in the learning phase using nested CV. For *p*, we started with 10 and increased the number by 10 while *p*≤100. The increment was gradually increased when *p*>100. For *k*, we tried the value of 1 to 15. Pairs of (*p*, *k*) with the highest accuracy was chosen in each learning set.

4.5 Shrunken Centroids (SC)

SC (Tibshirani et al., 2002) classifies the observations using shrunken class centroids. The standard class centroids are calculated by the average value of each predictor in each class divided by within-class standard deviation for that predictor. A new observation is classified based on a squared distance of the centroids of each class. However, SC shrinks these standard class centroids using threshold and moves them toward zero. This is a more reliable estimation of centroids because there may be noise in gene expression data.

We used R package (*pamr*) of SC with a soft thresholding option. As described in Section 4.4, we performed variable selection using BW ratio in the learning phase for each data.

4.6 LDA: Linear Dicriminant Analysis

LDA is used to find the linear combination of features which best separate samples of distinct groups. It maximizes the ratio of between-class variance to within-class variance. There are several different algorithms for LDA. Diagonal Linear Discriminant Analysis (DLDA) employs a simpler rule than Fisher's Linear Discriminant Analysis (FLDA) in classification which uses a linear discriminant function. DLDA assumes that the class densities are assumed to have the same diagonal matrix. The class of a new observation is determined by the distance of mean vector of each group using Mahalanobis distance. DLDA ignores correlation among predictors and uses a simple weighted sum of the predictors. According to Dudoit et al. (2002), it is a consistent classifier and shows better generalization performance than other LDA algorithms such as DQDA (nonlinear discriminant analysis) and FLDA.

Since DLDA often performs well when the dimension is reduced by a variable selection, we selected variables using the BW ratio as in kNN. The stat.diag.da and “MASS” packages in R were used for DLDA and FLDA, respectively, in this study. The optimal number of predictors was selected as the same way as in Section 4.4.

4.7 Single Optimal Trees

CART (Breiman et al., 1984) and QUEST (Loh and Shih, 1997) are based on binary splits. The results are provided by the optimal trees from these algorithms.

CART searches the split exhaustively. A single CART tree is constructed and pruned by the 1-SE rule discussed in Section 3.1. We used the *rpart* package which is an R-version of CART.

QUEST uses linear combination splits for improving the prediction accuracy over univariate split used in CART. To conduct cross-validation, we wrote a shell program using the executable files downloaded from the authors’ homepage (http://www.stat.wisc.edu/~loh/quest.html). We used the linear combination option for QUEST.

Chapter 5

Data Sets

5.1 Classification of Chemicals for Estrogen Activity

A number of environmental chemicals known as endocrine-disrupting chemicals (EDCs) are suspected of disrupting endocrine functions by mimicking or antagonizing natural hormones in animals and humans (Hileman, 1997). The estrogen activity database (Blair et al., 2000) contains a large and diverse estrogen data set. This data set contains 232 samples (chemicals) with 312 predictors. Out of these 232 structurally diverse chemicals, 131 chemicals exhibit estrogen receptor binding activity, while 101 are inactive, meaning that no activity was detectable in the assay. This structurally diverse data set has 312 predictors generated using the Molconn-Z software 4.07. The data set is given at http://www.ams.sunysb.edu/~hahn/research/CERP/estrogen.txt.

5.2 Classification of Gene Imprinting Data

Genomic imprinting, defined as gene expression dependent on the parent of origin, gives rise to numerous human diseases (Reik and Walter, 2001). Greally (2002) described the first characteristic sequence parameter that discriminates imprinted regions-a paucity of short interspersed transposable elements (SINEs). This finding has subsequently been confirmed by other groups.

The genomic data collected to study imprinted genes were from the UCSC Genome Browser (http://genome.ucsc.edu/). Annotation data were downloaded for the human genome (hg16, July 2003 freeze). The data contain 131 samples and 1446 predictors. Among the 131 samples, 43 are imprinted and 88 are control genes (non-imprinted). The current data set has been made available by John Greally at http://greallylab.aecom.yu.edu/~greally/imprinting_data.txt.

The sequence features of interest were repetitive elements (chrN-rmsk files), CpG island (cpgIsland file), transcription start sites of other genes and the exon count of each gene (refFene file). Each feature was examined for varying window sizes around the transcription start and end site.

5.3 Classification of Colon Tissue Sample

In cancer research, DNA microarray technology makes it possible to classify the tissue sample based on gene expression data, without prior and often subjective biological knowledge (Golub et al., 1999; Dudoit et al., 2002). Gene expression in 40 colon adenocarcinoma tissue samples and 22 normal colon tissue samples were analyzed with an Affymetrix oligonucleotide array complementary to more than 6,500 human genes (Alon et al., 1999). We used the data with 2,000 genes of the highest minimal intensity across the 62 tissue samples. It is available at http://microarray.princeton.edu/oncology.affydata/index.html.

5.4 Classification of Leukemia Subtypes

The Golub leukemia data set is introduced in one of the seminal papers applying statistical classification techniques to microarray data. Golub et al. (1999) classified acute myeoloid leukemia (AML) and acute lymphoblastic leukemia (ALL) subtypes using a variant of linear discriminant analysis based on gene expression profiling. We included the performance using state-of-the-art classifiers such as those presented by Dudoit et al. (2002) for comparing CERP with other classification methods. They used 38 samples of AML and ALL as the learning set and 34 samples as test set. Dudoit et al. (2002) combined these learning and test set for the analysis. Therefore, the data contain 47 ALL and 25 AML. We obtained the data from the website http://www.broad.mit.edu/cancer/software/genepattern/datasets/ and pre-processed the data as described in Golub et al. so that 3,571 genes remain in the data.

5.5 Classification of Breast Cancer

van't Veer et al. (2002) used gene expression data to identify patients who would benefit from adjuvant chemotherapy according to classification of prognostication with distant metastases. The data contain 78 primary breast cancers: 34 from patients in poor prognosis and 44 from patients who continue to be disease-free (good prognosis) after a period of at least 5 years. These samples have been selected from patients who were lymph node negative and under 55 years of age at diagnosis. Out of approximately 25,000 gene expression levels, about 5,000 significantly regulated genes (at least a two-fold difference and a *p*-value of less than 0.01) in more than 3 tumors out of 78 were selected (Dudoit et al., 2002). The data can be downloaded from http://www.rii.com/publications/2002/vantveer.htm.

Chapter 6

Classification of Real Data Sets

We evaluated the prediction accuracy of LR-T CERP and LR CERP along with the balance between sensitivity and specificity using real data sets. Before applying the methods, we removed the predictors that had identical values for more than 98% of the samples in order to reduce the possibility that a predictor in a learning set would not have distinct values in the CV for building a base classifier. For the estrogen data, 250 out of 312 predictors were selected using this criterion, and for the gene imprinting data, 1248 out of 1446 predictors were selected for the analysis. For the other data sets, all the predictors were included by this criterion. We perform 20 repetitions of 10-fold CV for CERP and other methods and take the average of the results in order to obtain a stable result. Twenty CVs should be sufficient according to Molinaro et al. (2005) who recommended at least 10 CVs in order to have low MSE and bias.

We conducted a variable selection for kNN, SC and LDA. We used the BW ratio criteria in the learning phase as discussed in Section 4.4. In SC and LDA, only p is searched using the same method as in kNN. An average and standard deviation of the number of selected variable were included in result tables.

We found that the prediction accuracy slightly increases by having multiple ensembles, and 11 ensembles were enough to achieve an improvement in this study.

We compared the performance of LR CERP and LR-T CERP with other widely used methods. Tables 2 through 6 show the accuracy, sensitivity and specificity of each method. We also provide the accuracy graph of each classification model with a 1-sd bar in Figures 2 through 6.

In the comparison of the methods, CERP does not require any fine tuning of parameters because they are determined in the training phase inside the program. For the most relevant comparison, we provide the best result we obtained for each data set for the other methods and specify the parameters used in the footnote. For the methods requiring variable pre-selection, an optimal number of variables is searched and the variables are selected by the BW ratio in the training phase. For kNN, the optimal value of *k* is also obtained in nested CV.

DLDA did not perform well even with a variable selection for the estrogen data (see Table 2), because it assumes that features are not correlated. FLDA performed better than DLDA with variable selection, but the accuracy was still lower than most of the other methods. Because the data are reasonably balanced (proportion of the positive responses in the data is .56), the balance of sensitivity and specificity was good in most of the methods. In SVM, RBF performed better than linear kernel unlike in the other data sets. It appears to be due to a nonlinear relationship that was not captured using a linear function of the predictors. The performance of kNN and SC was not comparable to many other methods even

Table 2: Accuracy (standard deviation in parentheses) of classification methods for the **estrogen data** with 131 cases and 101 controls.

Method	Approach	#predictors	Overall	Sensitivity	Specificity
CERP[a]	LR	all	.81 (.02)	.86 (.02)	.75 (.02)
	LR-T	all	.85 (.01)	.89 (.01)	.79 (.02)
RF[b]		all	.84 (.01)	.88 (.01)	.79 (.02)
SVM	Lin. Kernel	all	.79 (.01)	.83 (.02)	.74 (.03)
	RBF[c]	all	.83 (.01)	.89 (.02)	.75 (.01)
Boosting	LogitBoost	all	.82 (.02)	.85 (.03)	.77 (.03)
kNN[d]		63 (63)[e]	.74 (.03)	.83 (.03)	.62 (.05)
SC		51 (51)[e]	.70 (.02)	.75 (.03)	.63 (.02)
LDA	FLDA	60 (25)[e]	.78 (.02)	.87 (.02)	.66 (.03)
	DLDA	49 (33)[e]	.73 (.01)	.76 (.02)	.69 (.02)
Single	CART	all	.77 (.01)	.88 (.02)	.62 (.02)
tree	QUEST	all	.66 (.03)	.72 (.04)	.59 (.07)

[a] average partition size: 6.9, 11.1

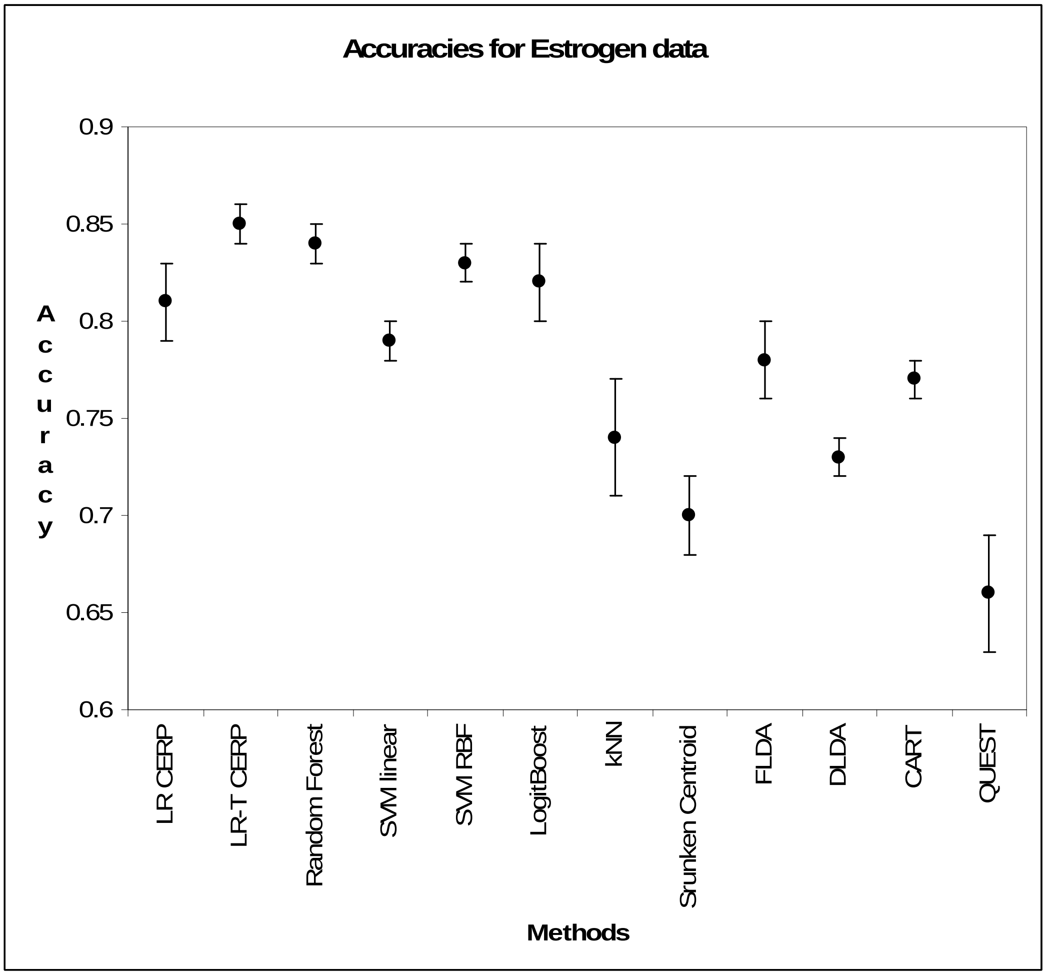
[b] average number of trees: 300; number of predictors: default (floor[$m^{1/2}$])

[c] radial basis function (default option for the SVM function in the R package E1071)

[d] average (sd in parantheses) of k obtained in the training phase: 3.1 (2.2)

[e] average (sd in parantheses) number of predictors selected in the training phase

Figure 2: Comparison of accuracies (with 1-sd bars) of classification methods for the **estrogen data**.



after a variable selection.

Table 3 shows that the sensitivity and specificity were 74% and 95%, respectively, by LR CERP, while they were 66% and 99%, respectively, by RF for the gene imprinting data. According to paired t-test, sensitivity of RF is significantly lower than that of LR CERP (t=5.34, p<0.0001*), while their accuracies are not significantly different. These results support the criticism about RF on the imbalance by Dudoit and Fridlyand (2003). For RF, we tried various choices of number of variables to be selected in each node of a tree, and the number of trees in the forest including the default option. Furthermore, we tested various thresholds in RF package of R. However, the results did not substantially differ beyond the random error. CERP gave high accuracy and good balance between sensitivity and specificity compared to the other methods. For DLDA, the accuracy reached the highest when a large number of variables were pre-selected. The single trees show severe imbalance between sensitivity and specificity as well as poor accuracy.

For the colon data set (see Table 4), LR CERP gave 85% of accuracy and perfect balance of sensitivity and specificity. The LogitBoost failed to give comparable prediction accuracy for this particular data set, although it performed well on the other data sets we examined. RF and SVM with RBF kernel showed poor balance of sensitivity and specificity compared to CERP. The difference of accuracy (t=3.21, p=0.0046*) and specificity (t=10.16, p<0.0001*) between RF

Table 3: Accuracy (standard deviation in parentheses) of classification methods for the **imprinting data** with 43 cases and 88 controls.

Method	Approach	#predictors	Overall	Sensitivity	Specificity
CERP[a]	LR	all	.88 (.02)	.74 (.03)	.95 (.02)
	LR-T	all	.89 (.01)	.72 (.04)	.97 (.02)
RF[b]		all	.88 (.01)	.66 (.03)	.99 (.01)
SVM	Lin. Kernel	all	.84 (.02)	.70 (.04)	.92 (.02)
	RBF[c]	all	.79 (.02)	.46 (.04)	.95 (.02)
Boosting	LogitBoost	all	.84 (.02)	.73 (.04)	.89 (.03)
kNN[d]		261 (322)[e]	.78 (.02)	.67 (.05)	.84 (.03)
SC		40 (11)[e]	.83 (.02)	.70 (.03)	.90 (.02)
LDA	FLDA	391 (107)[e]	.79 (.02)	.63 (.06)	.88 (.03)
	DLDA	667 (241)[e]	.86 (.02)	.63 (.04)	.97 (.01)
Single	CART	all	.73 (.03)	.40 (.10)	.89 (.04)
tree	QUEST	all	.67 (.04)	.31 (.09)	.85 (.05)

[a] average partition size: 61.1, 71.3

[b] average number of trees: 400; number of predictors: default (floor[$m^{1/2}$])

[c] radial basis function (default option for the SVM function in the R package E1071)

[d] average (sd in parantheses) of k obtained in the training phase: 2.7 (1.7)

[e] average (sd in parantheses) number of predictors selected in the training phase

Figure 3: Comparison of accuracies (with 1-sd bars) of classification methods for the **imprinting data**.

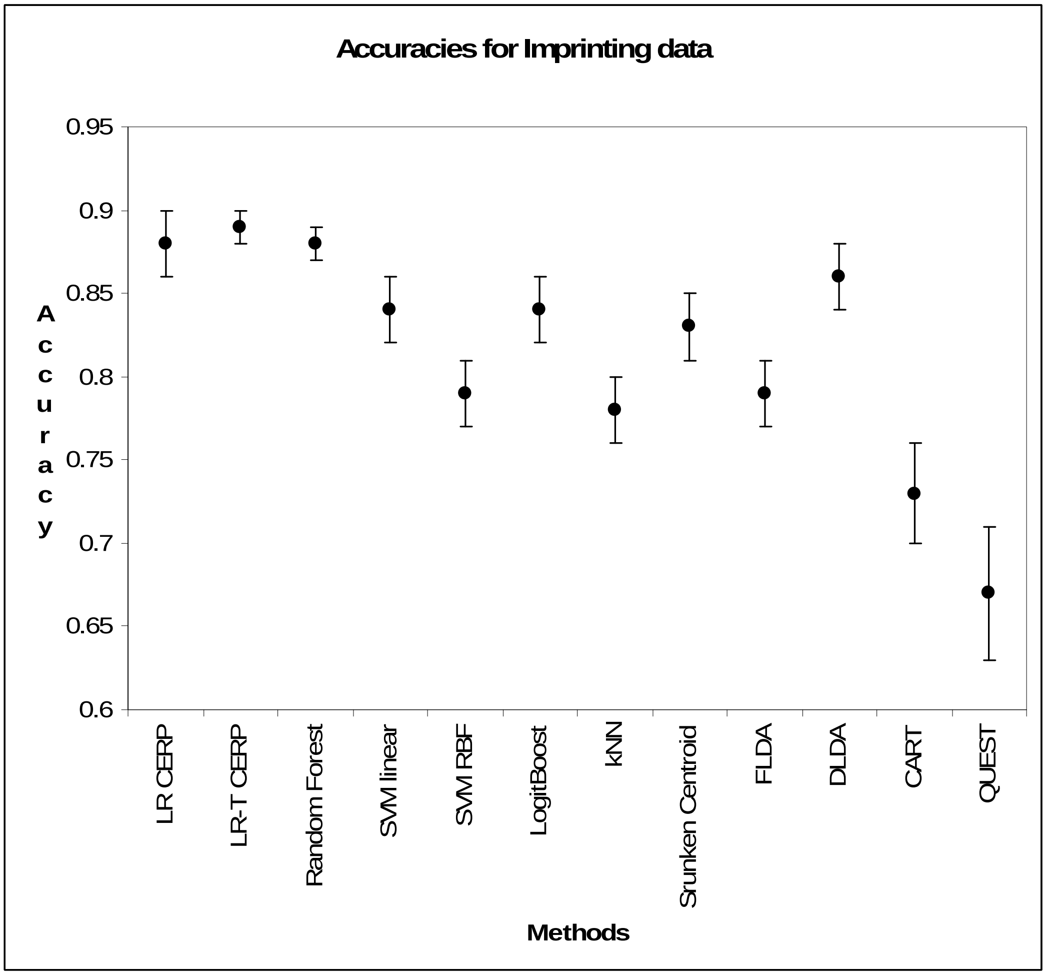


Table 4: Accuracy (standard deviation in parentheses) of classification methods for the **colon data** with 22 cases and 40 controls.

Method	Approach	#predictors	Overall	Sensitivity	Specificity
CERP[a]	LR	all	.85 (.02)	.85 (.01)	.85 (.01)
	LR-T	all	.85 (.02)	.87 (.01)	.83 (.05)
RF[b]		all	.81 (.04)	.88 (.01)	.68 (.10)
SVM	Lin. Kernel	all	.85 (.03)	.88 (.02)	.79 (.06)
	RBF[c]	all	.81 (.02)	.94 (.02)	.56 (.05)
Boosting	LogitBoost	all	.73 (.03)	.82 (.03)	.58 (.07)
kNN[d]		303 (396)[e]	.84 (.04)	.88 (.02)	.75 (.08)
SC		452 (374)[e]	.85 (.02)	.87 (.02)	.81 (.07)
LDA	FLDA	32 (25)[e]	.87 (.03)	.88 (.03)	.84 (.05)
	DLDA	46 (117)[e]	.85 (.02)	.86 (.02)	.83 (.05)
Single	CART	all	.71 (.04)	.87 (.04)	.41 (.10)
tree	QUEST	all	.82 (.02)	.86 (.03)	.75 (.04)

[a] average partition size: 138.1, 172.3

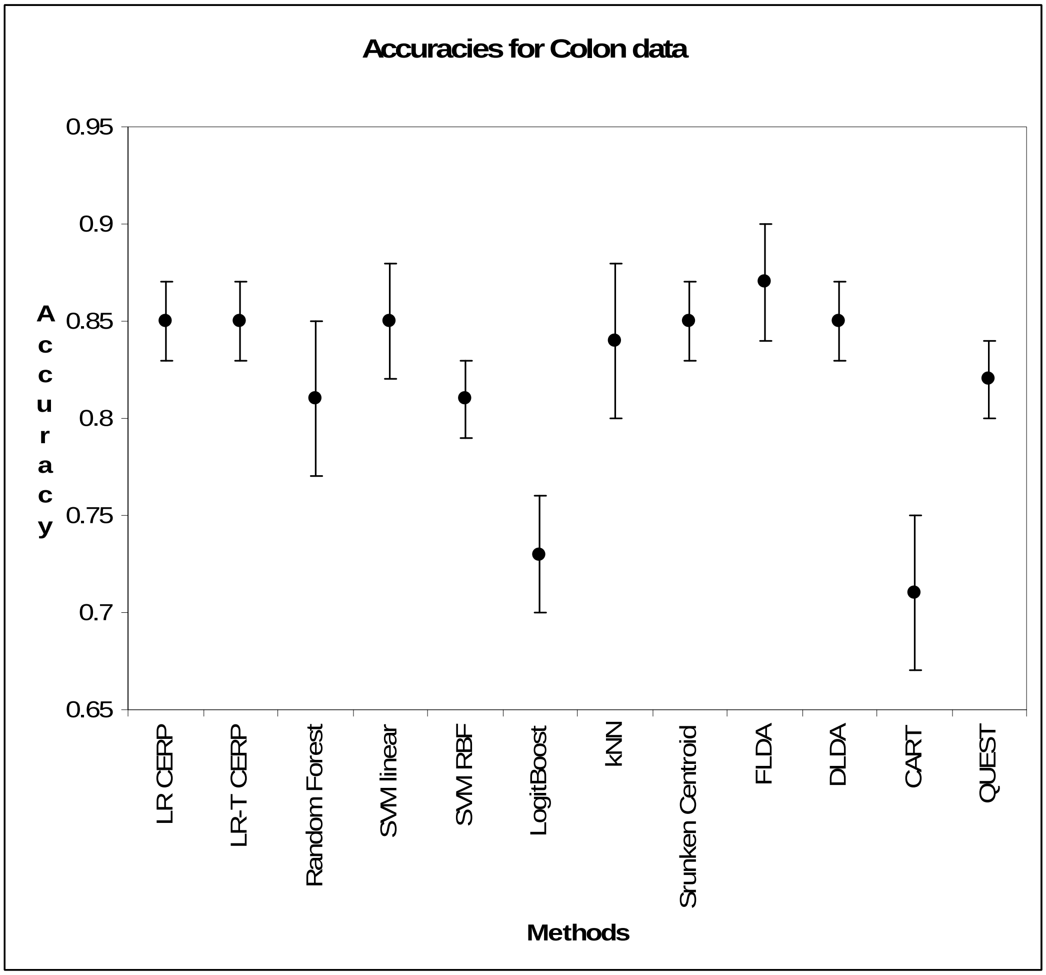
[b] average number of trees: 200; number of predictors: default (floor[$m^{1/2}$])

[c] radial basis function (default option for the SVM function in the R package E1071)

[d] average (sd in parantheses) of k obtained in the training phase: 5.1 (2.3)

[e] average (sd in parantheses) number of predictors selected in the training phase

Figure 4: Comparison of accuracies (with 1-sd bars) of classification methods for the **colon data**.



and LR CERP are significant based on paired t-test. The results shown in this table agree with the results by other researchers. Ambroise and McLachlan (2002) studied the difference of prediction error between internal and external cross-validation. They used SVM with a combination of linear kernel and a backward selection procedure. They stated that the error rate as estimated was above 15% in external CV. Tsai et al. (2004) reported that their prediction accuracy of kNN (k=1) and SVM was 84%. kNN and DLDA gave high accuracy with pre-selected predictors. FLDA showed the best performance with 87% accuracy, but this is obtained after a variable selection was done for FLDA. SVM performed better when the linear kernel was used instead of the radial based function (RBF). It is notable that QUEST is comparable to RF for the colon and leukemia data sets, while it performed considerably worse than RF for the other data sets. For this data set, RF required fewer trees (*ntree*=200) for the optimal performance compared to other data sets.

For the leukemia data (see Table 5), all the methods gave high prediction accuracy ranging from 95% to 98% except for CART. This tendency of high accuracy is also shown in Dudoit et al. (2002). This data set also shows heavy imbalance (proportion of the positive responses in the data is .35) of the frequencies of the two classes, but the balance between sensitivity and specificity is not a concern because of the high prediction accuracy.

For the breast cancer data (see Table 6), the prediction accuracies of all the

Table 5: Accuracy (standard deviation in parentheses) of classification methods for the **leukemia data** with 47 ALL and 25 AML samples.

Method	Approach	#predictors	Overall	Sensitivity	Specificity
CERP[a]	LR	all	.98 (.01)	.96 (.00)	.99 (.01)
	LR-T	all	.98 (.01)	.96 (.01)	.99 (.01)
RF[b]		all	.98 (.01)	.96 (.01)	1.00 (.01)
SVM	Lin. Kernel	all	.98 (.01)	.96 (.01)	.99 (.01)
	RBF[c]	all	.98 (.01)	.93 (.03)	1.00 (.00)
Boosting	LogitBoost	all	.96 (.01)	.95 (.02)	.96 (.01)
kNN[d]		198 (507)[e]	.97 (.02)	.93 (.04)	.98 (.02)
SC		23 (16)[e]	.96 (.01)	.92 (.02)	.98 (.01)
LDA	FLDA	178 (151)[e]	.95 (.02)	.92 (.04)	.96 (.02)
	DLDA	140 (367)[e]	.97 (.01)	.94 (.02)	.99 (.01)
Single	CART	all	.81 (.03)	.77 (.06)	.83 (.04)
tree	QUEST	all	.96 (.03)	.96 (.01)	.95 (.04)

[a] average partition size: 245.2, 238.7

[b] average number of trees: 200; number of predictors: default (floor[$m^{1/2}$])

[c] radial basis function (default option for the SVM function in the R package E1071)

[d] average (sd in parantheses) of k obtained in the training phase: 3.1 (3.3)

[e] average (sd in parantheses) number of predictors selected in the training phase

Figure 5: Comparison of accuracies (with 1-sd bars) of classification methods for the **leukemia data**.

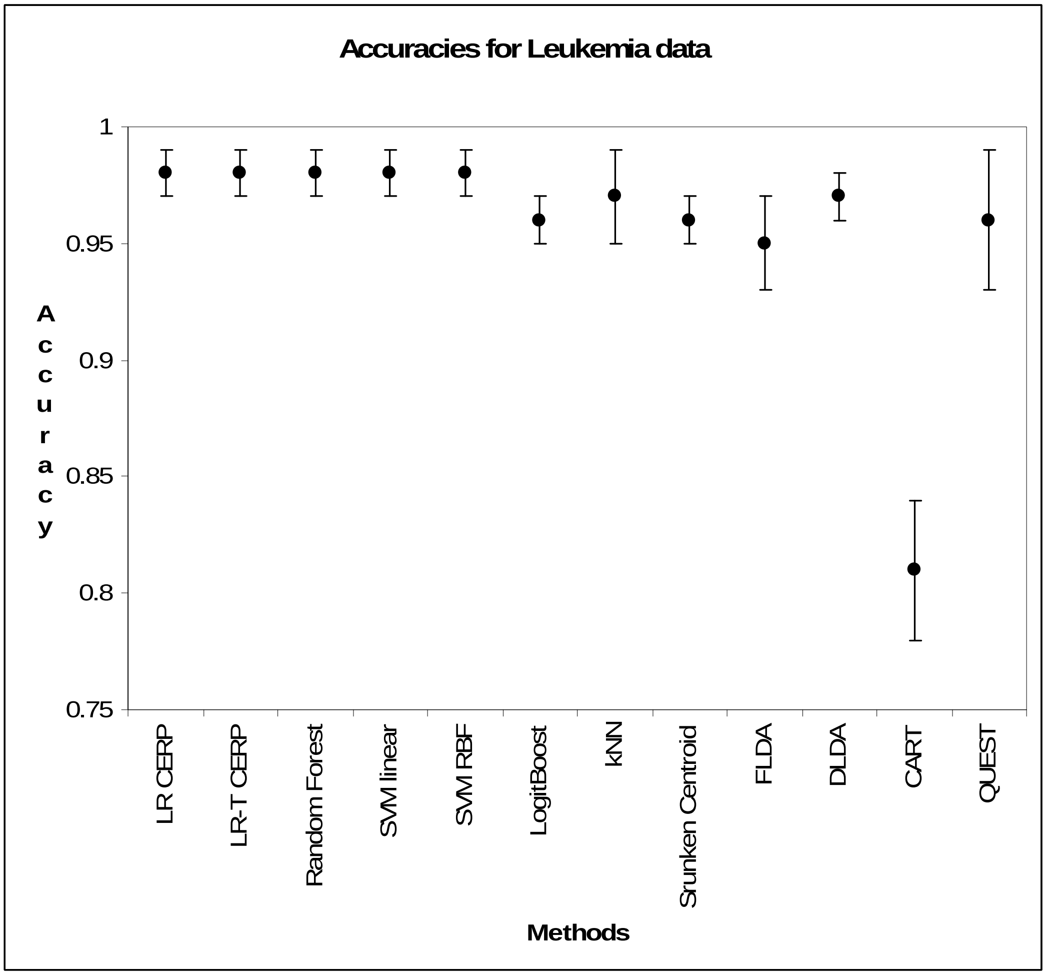


Table 6: Accuracy (standard deviation in parentheses) of classification methods for the **breast cancer data** with 34 cases and 44 controls.

Method	Approach	#predictors	Overall	Sensitivity	Specificity
CERP[a]	LR	all	.61 (.03)	.53 (.06)	.66 (.03)
	LR-T	all	.61 (.03)	.55 (.06)	.65 (.04)
RF[b]		all	.64 (.03)	.49 (.04)	.75 (.03)
SVM	Lin. Kernel	all	.61 (.02)	.56 (.05)	.65 (.02)
	RBF[c]	all	.57 (.04)	.39 (.07)	.70 (.04)
Boosting	LogitBoost	all	.72 (.04)	.64 (.07)	.78 (.04)
kNN[d]		707 (1040)[e]	.63 (.05)	.53 (.09)	.72 (.05)
SC		314 (622)[e]	.60 (.03)	.50 (.05)	.68 (.04)
LDA	FLDA	368 (601)[e]	.60 (.04)	.55 (.06)	.64 (.05)
	DLDA	314 (685)[e]	.61 (.02)	.54 (.03)	.67 (.02)
Single	CART	all	.54 (.03)	.17 (.09)	.83 (.08)
tree	QUEST	all	.52 (.03)	.14 (.06)	.82 (.06)

[a] average partition size: 302.6, 343

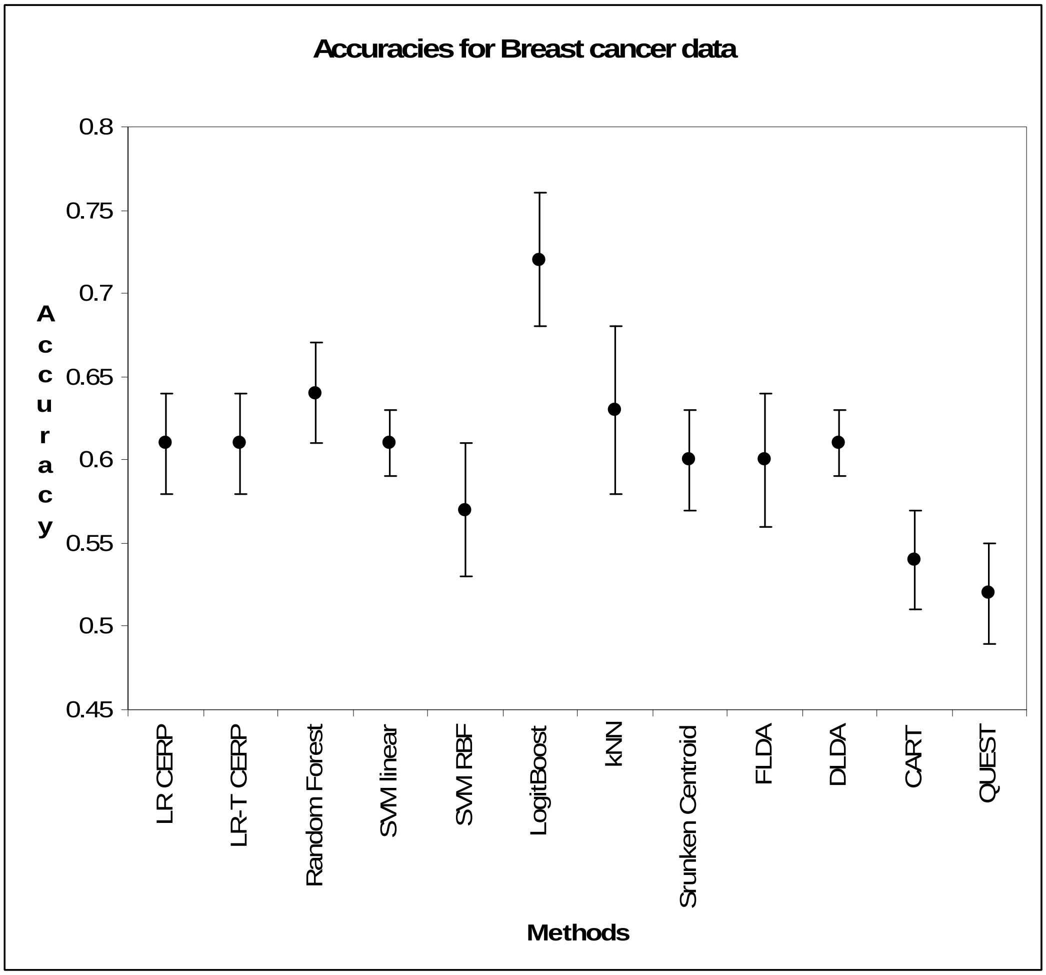
[b] average number of trees: 200; number of predictors: default (floor[$m^{1/2}$])

[c] radial basis function (default option for the SVM function in the R package E1071)

[d] average (sd in parantheses) of k obtained in the training phase: 4.3 (2.6)

[e] average (sd in parantheses) number of predictors selected in the training phase

Figure 6: Comparison of accuracies (with 1-sd bars) of classification methods for the **breast cancer data**.



methods were lower than 65% except for LogitBoost. This low accuracy is in line with the results published by Moon et al. (2007). LogitBoost was superior to other methods in accuracy for this data. Although RF gave the second best accuracy, the balance between sensitivity and specificity was worse than those of CERP, kNN and SVM with linear kernel. Not only the performance of single tree methods, but also the balance between sensitivity and specificity were poor.

Table 7 provides the performance ranking of the classification methods for the five real data sets based on the overall accuracy. This table shows that both LR and LR-T CERP are ranked the highest among all the methods compared in this study. They give high prediction accuracy and the ranks are in the top half for all five real data sets.

Table 7: Performance ranking of the classification method for each data set.

		Data set					
Method	Approach	Estr	Impr	Colon	Leuk	Breast	Avg. rank
CERP[a]	LR	5	2	2	1	4	2.8
	LR-T	1	1	2	1	4	1.8
RF[b]		2	2	9	1	2	3.2
SVM	Lin. Kernel	6	5	2	1	4	3.6
	RBF[c]	3	8	9	1	10	6.2
Boosting	LogitBoost	4	5	11	8	1	5.8
kNN[d]		9	10	7	6	3	7.0
SC		11	7	2	8	8	7.2
LDA	FLDA	7	8	1	11	8	7.0
	DLDA	10	4	2	6	7	5.8
Single	CART	8	11	12	12	11	10.8
tree	QUEST	12	12	8	8	12	10.4

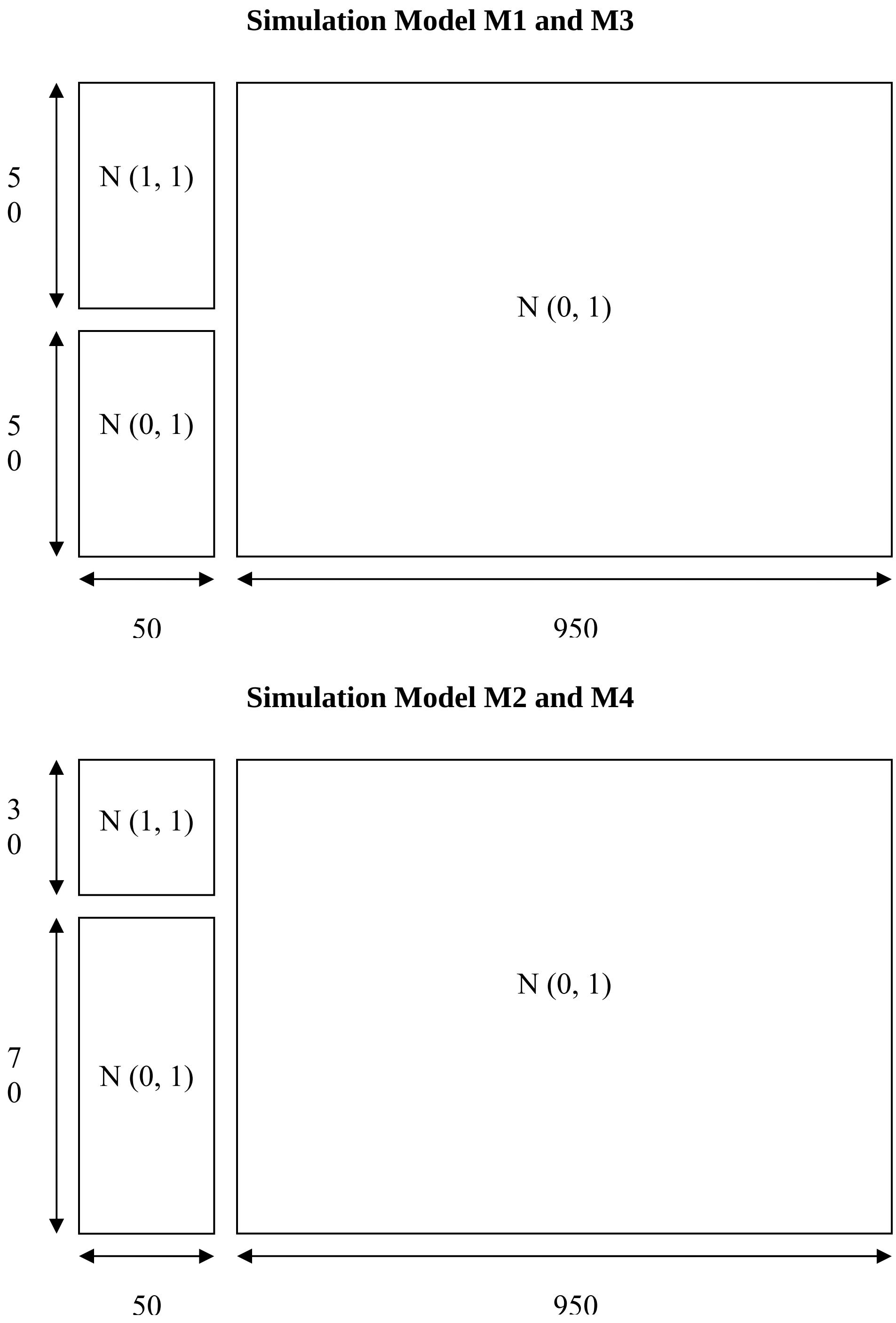
Chapter 7

Simulation Study

7.1 Models

We performed a simulation study to evaluate the proposed LR CERP and LR-T CERP and compared their performance with other well known classification methods. This simulation consists of two parts. In the first part, predictors were independently generated, and in the second part, predictors were generated to have correlation. We generated each simulation data set with 100 subjects and 1000 predictors. Fifty of these predictors were generated from two different normal distributions, and the remaining 950 predictors were generated from one normal distribution and served as noise. In each of the simulation data sets, the first fifty variables were generated from N(1, 1) for cases and N(0, 1) for controls. Four different models were considered in this simulation study. For models M1 and M2, these variables were generated independently. For models M3 and M4, correlation was given to each pair of 50 variables. The upper-diagonal elements of the correlation matrix were generated randomly from the Uniform(0, 0.8) distribution. This correlation structure was generated once before generating simulation data, and used for generating all the simulation data sets from each of M3 and M4. The average pairwise correlation obtained in this study was 0.4255 for M3 and 0.4053 for M4. The remaining 950 variables were independently

Figure 7: The structure of 4 simulation models.



generated from N (0, 1). The case-control ratio was given as 50:50 for M1 and M3, and 30:70 for M2 and M4.

7.2 Results

One hundred data sets were generated from each of the four models. For each simulation replication, 10-fold cross validation was performed for evaluating the performance of each classification method. The average of the accuracies from the 100 simulation data sets from each classification method are reported in Tables 8 through 11, and Figures 8 through 11.

As seen in the classification of the leukemia data set, simulation from M1 (see Table 8) showed high accuracies in all methods except for LogitBoost and CART. The balance between sensitivity and specificity was also good for all methods because the positive rate of this model is exactly .5.

For M2 (see Table 8), RF, SVM with RBF kernel, LogitBoost and CART showed low accuracy compared to other methods. This result supports our observation that RF does not perform well for unbalanced data. These methods tended to classify most of the cases into majority class. CERP performed well for both of M1 and M2.

The accuracies of the classification methods were lower for the models with correlated data (M3 and M4). Although there was no clear distinction in accuracy among the methods for these models, CART consistently performed worse than the other methods poorly for these models. LogitBoost, SVM RBF, CART and RF showed a severe imbalance between sensitivity and specificity for both models with unbalanced data (M2 and M4).

Table 8: Accuracy (standard deviation in parentheses) of classification methods for the simulation data M1.

Method	Approach	#predictors	Overall	Sensitivity	Specificity
CERP[a]	LR	all	.97 (.02)	.97 (.02)	.97 (.02)
	LR-T	all	.94 (.03)	.94 (.03)	.94 (.04)
RF[b]		all	.98 (.01)	.98 (.02)	.98 (.02)
SVM	Lin. Kernel	all	.99 (.01)	.99 (.02)	.99 (.02)
	RBF[c]	all	.98 (.02)	.98 (.02)	.98 (.03)
Boosting	LogitBoost	all	.69 (.08)	.69 (.08)	.69 (.10)
kNN[d]		104 (189)[e]	.96 (.02)	.96 (.03)	.96 (.03)
SC		32 (35)[e]	.99 (.01)	.99 (.01)	.99 (.01)
LDA	FLDA	32 (22)[e]	.98 (.02)	.98 (.03)	.98 (.02)
	DLDA	34 (43)[e]	.99 (.01)	.99 (.01)	.99 (.01)
Single	CART	all	.69 (.07)	.69 (.07)	.70 (.09)
tree	QUEST	all	.99 (.01)	.99 (.01)	.99 (.03)

[a] average partition size: 73.2, 82.2

[b] average number of trees: 305; number of predictors: default (floor[$m^{1/2}$])

[c] radial basis function (default option for the SVM function in the R package E1071)

[d] average (sd in parantheses) of k obtained in the training phase: 11.9 (2.6)

[e] average (sd in parantheses) number of predictors selected in the training phase

Figure 8: Comparison of accuracies (with 1-sd bars) of classification methods for simulation model M1.

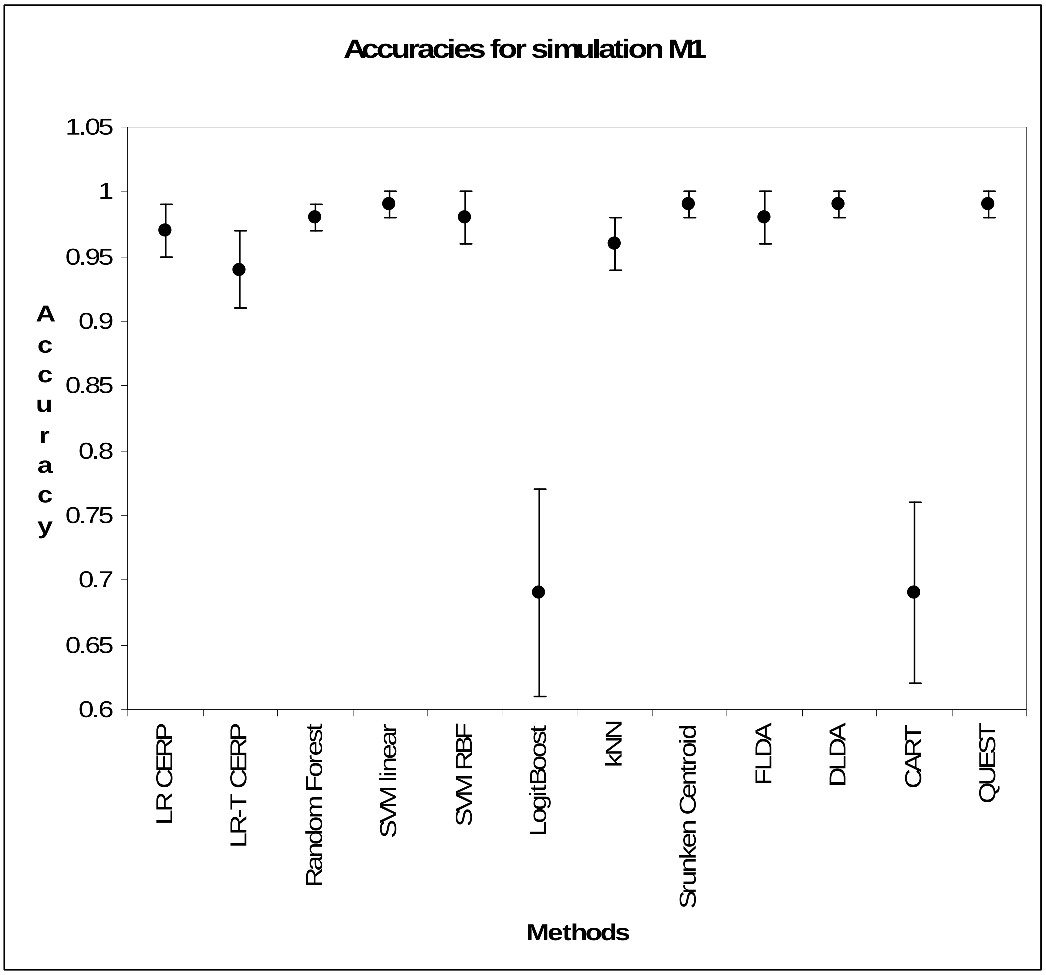


Table 9: Accuracy (standard deviation in parentheses) of classification methods for the simulation data M2.

Method	Approach	#predictors	Overall	Sensitivity	Specificity
CERP[a]	LR	all	.97 (.02)	.91 (.06)	1.00 (.01)
	LR-T	all	.96 (.02)	.87 (.07)	1.00 (.01)
RF[b]		all	.77 (.03)	.22 (.08)	1.00 .(00)
SVM	Lin. Kernel	all	.95 (.02)	.85 (.07)	1.00 (.00)
	RBF[c]	all	.70 (.00)	.00 (.00)	1.00 (.00)
Boosting	LogitBoost	all	.70 (.07)	.68 (.12)	.71 (.07)
kNN[d]		161 (238)[e]	.95 (.02)	.84 (.07)	.99 (.01)
SC		30 (13)[e]	.99 (.01)	.96 (.03)	1.00 (.00)
LDA	FLDA	29 (11)[e]	.98 (.01)	.97 (.03)	.99 (.01)
	DLDA	30 (11)[e]	.99 (.01)	.99 (.02)	1.00 (.01)
Single	CART	all	.72 (.05)	.30 (.17)	.90 (.05)
tree	QUEST	all	.96 (.03)	.90 (.05)	.99 (.03)

[a] average partition size: 79.1 80.7

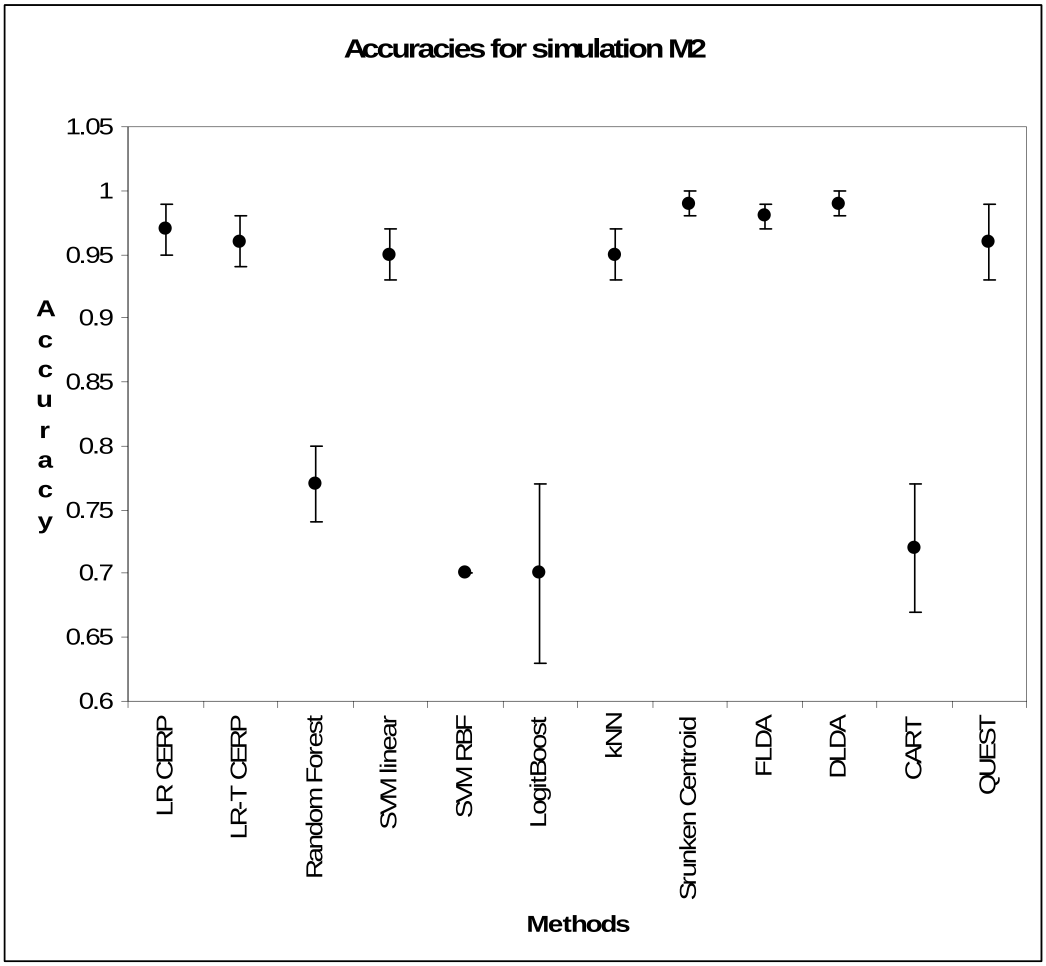
[b] average number of trees: 229; number of predictors: default (floor[$m^{1/2}$])

[c] radial basis function (default option for the SVM function in the R package E1071)

[d] average (sd in parantheses) of k obtained in the training phase: 6.0 (3.0)

[e] average (sd in parantheses) number of predictors selected in the training phase

Figure 9: Comparison of accuracies (with 1-sd bars) of classification methods for simulation model M2.



DLDA and SC showed high generalization accuracy for all four simulation models. Furthermore, they achieved a good balance between sensitivity and specificity even for M4 although all other methods showed poor balance for it. Note that these two methods used the variable selection prior to classification.

Between the single optimal trees, QUEST performed better than CART for all four models. Because the simulation is based on normal distribution, linear combination splits may have more strength than univariate splits.

This simulation study shows how the correlation affects the accuracy of ensemble when we compare the accuracies between M1 and M3, and between M2 and M4. Two sample t-test shows the significant difference in accuracies of LR CERP between M1 and M3 (t=33.42, p=0.005), and between M2 and M4 (t=35.78, p=0.004). Correlation among the predictors makes the base classifiers dependent to each other and causes a slow improvement of the ensemble accuracy as demonstrated in Chapter 2. It leads to the difference of the overall accuracies among the simulation models. Because the partition size is limited, ensemble accuracy of CERP reaches their bound which is lower than 1 as discussed in Chapter 2.

Table 10: Accuracy (standard deviation in parentheses) of classification methods for the simulation data M3.

Method	Approach	#predictors	Overall	Sensitivity	Specificity
CERP[a]	LR	all	.79 (.05)	.79 (.05)	.79 (.06)
	LR-T	all	.78 (.05)	.78 (.06)	.77 (.06)
RF[b]		all	.79 (.04)	.80 (.05)	.78 (.06)
SVM	Lin. Kernel	all	.77 (.05)	.77 (.06)	.78 (.06)
	RBF[c]	all	.78 (.05)	.79 (.07)	.78 (.07)
Boosting	LogitBoost	all	.74 (.06)	.73 (.07)	.74 (.07)
kNN[d]		178 (250)[e]	.78 (.05)	.79 (.07)	.77 (.07)
SC		175 (227)[e]	.81 (.04)	.82 (.05)	.81 (.05)
LDA	FLDA	169 (219)[e]	.74 (.07)	.74 (.08)	.74 (.08)
	DLDA	171 (225)[e]	.81 (.04)	.81 (.05)	.81 (.05)
Single	CART	all	.68 (.08)	.69 (.10)	.68 (.11)
tree	QUEST	all	.77 (.06)	.78 (.06)	.76 (.07)

[a] average partition size: 71.3, 79.5

[b] average number of trees: 262; number of predictors: default (floor[$m^{1/2}$])

[c] radial basis function (default option for the SVM function in the R package E1071)

[d] average (sd in parantheses) of k obtained in the training phase: 10.8 (3.3)

[e] average (sd in parantheses) number of predictors selected in the training phase

Figure 10: Comparison of accuracies (with 1-sd bars) of classification methods for simulation model M3.

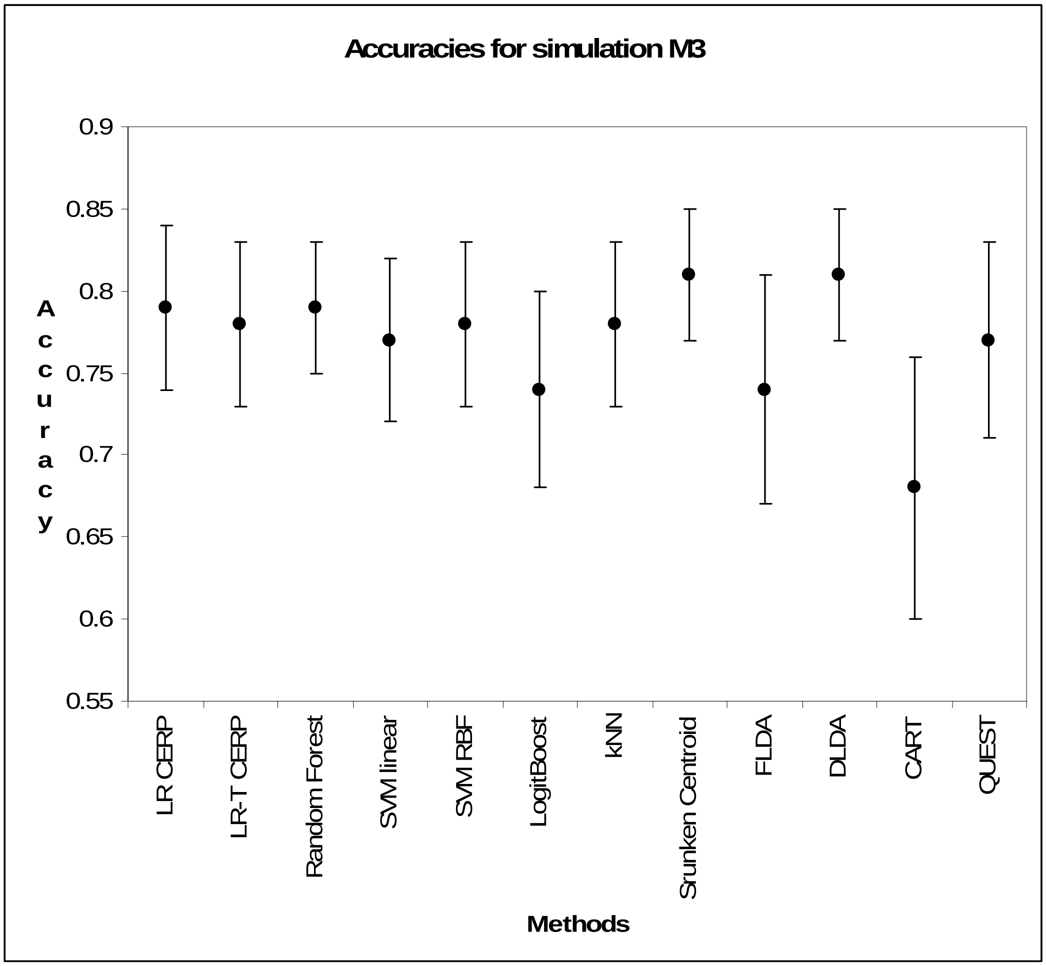


Table 11: Accuracy (standard deviation in parentheses) of classification methods for the simulation data M4.

Method	Approach	#predictors	Overall	Sensitivity	Specificity
CERP[a]	LR	all	.81 (.04)	.52 (.12)	.94 (.03)
	LR-T	all	.81 (.04)	.51 (.13)	.94 (.03)
RF[b]		all	.77 (.04)	.27 (.13)	.98 (.02)
SVM	Lin. Kernel	all	.80 (.04)	.48 (.10)	.93 (.03)
	RBF[c]	all	.70 (.00)	.00 (.01)	1.00 (.00)
Boosting	LogitBoost	all	.77 (.05)	.49 (.11)	.89 (.04)
kNN[d]		162 (228)[e]	.80 (.05)	.51 (.12)	.92 (.04)
SC		285 (269)[e]	.82 (.04)	.75 (.09)	.85 (.03)
LDA	FLDA	305 (280)[e]	.76 (.05)	.50 (.11)	.87 (.06)
	DLDA	290 (270)[e]	.82 (.04)	.73 (.08)	.86 (.04)
Single	CART	all	.70 (.04)	.19 (.16)	.91 (.06)
tree	QUEST	all	.78 (.05)	.43 (.18)	.93 (.03)

[a] average partition size: 72.9, 74.4

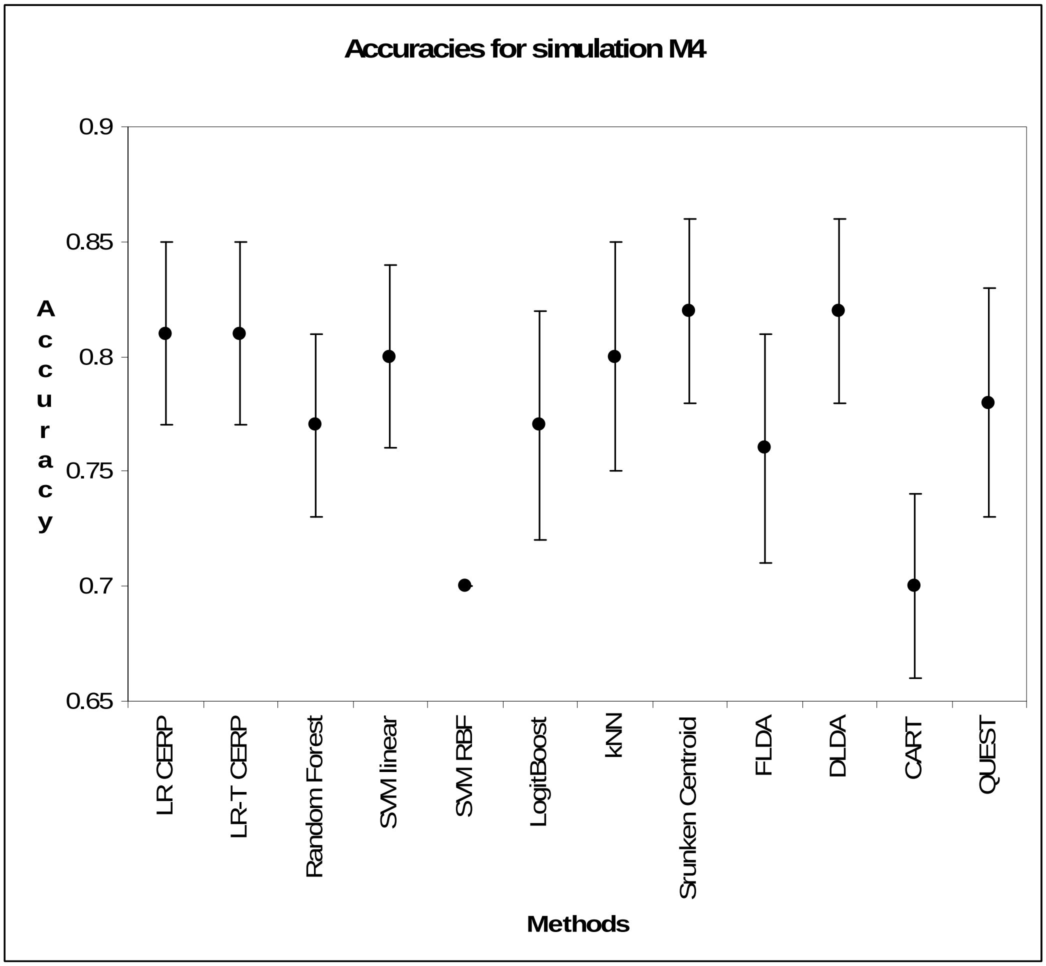
[b] average number of trees: 249; number of predictors: default (floor[$m^{1/2}$])

[c] radial basis function (default option for the SVM function in the R package E1071)

[d] average (sd in parantheses) of k obtained in the training phase: 8.4 (3.7)

[e] average (sd in parantheses) number of predictors selected in the training phase

Figure 11: Comparison of accuracies (with 1-sd bars) of classification methods for simulation model M4.



Chapter 8

Conclusions and Future Studies

We have developed an ensemble-based classifier with logistic regression models on each of the subsets in a random partition of the parameter space. The new methods have the following advantages compared to other well-known statistical classification methods.

- Logistic regression models can be used for a high-dimensional data with an improved performance without variable pre-selection. Computationally simple models such as LDA often require variable selection for an optimal performance. The variable selection can be computer-intensive for a high-dimensional data, and it does not guarantee an optimal subset.
- LR CERP is based on a widely used standard regression model for binary responses. Thus it is substantially less computer-intensive than other aggregation methods such as RF or SVM.
- The proposed methods take advantage of the CERP methods including low correlation among base classifiers by random partitioning of the feature space.

In a logistic regression model, the balance of sensitivity and specificity relies highly on the threshold of classification. We searched the optimal thresholds from cross validation based on learning sets. The balance is significantly improved compared to other aggregation methods such as RF and SVM for unbalanced data sets.

The main idea of CERP is that we partition the feature space to mutually exclusive subspaces so that we can bypass the difficulty of the high-dimensional data. This is a major difference among CERP and other ensemble-based methods such as RF which uses the same feature in each classifier. Since we select a different subset of predictors in each classifier, we can reduce the correlation between classifiers and gain an improvement discussed in Chapter 2. We have shown empirically that huge data sets need not be handled as a whole; the subspaces of the feature space created through partitioning may be treated independently and separately until after the classifiers are developed. This gives CERP a huge computational advantage to tackling the growing problem of dimensionality. Like RF, CERP does not require variable pre-selection, thus it is straightforward and easy to implement the algorithm.

We showed that LR CERP and LR-T CERP are comparable to other well-known classification methods. The classification methods we selected for comparison performed well, while no method consistently outperforms the others. Using a CV estimate, we demonstrated that LR CERP and LR-T CERP show consistently high accuracy. This high accuracy was achieved partly due to the diversity created among classifiers.

Although RF gives a high accuracy in general, it often fails to give a good balance between sensitivity and specificity for unbalanced data as criticized by Dudoit and Fridlyand (2003). According to Chen et al. (2004), the imbalance occurs in many classification methods because they tend to focus on improvement of accuracy. We also found in the analysis of gene imprinting data that RF gives a poor sensitivity, while it gives almost perfect specificity. This imbalance is not desirable because a goal here is to identify more positives (imprinted genes). The balance between sensitivity and specificity is improved by LR CERP and LR-T CERP in unbalanced data. In LR CERP and LR-T CERP, the optimal threshold choice helps improve the balance. We are mainly interested in showing the enhanced accuracy, but the better balance is a strong attribute of CERP as well. Further exploration of the performance of LR CERP and LR-T CERP with respect to imbalance will be done in the future work.

As shown in this study, some methods such as LDA, kNN and SC often show an improved performance when using the pre-selected variables. However, variable pre-selection based on the BW ratio does not always provide the best performance. Furthermore, some variable selection rules are often computer intensive for high-dimensional data.

Since all the parameters are determined in the training phase of the program, CERP does not require any fine tuning for specific data sets in the comparison. Thus it can be used for any type of high-dimensional data set. Although RF tends to perform well with the default parameter values, the performance may depend on the number of classifiers or number of randomly selected predictors in each node of a tree. When using the RBF kernel for SVM, a fine tuning of the relevant parameters such as kernel width is needed. Often the default parameter value does not work very well for large number of attributes.

When we performed CV, the run time of CERP was reasonable compared to that of other methods. For leukemia data, for example, it took approximately 20 minutes to finish a 10-fold CV for LR CERP with 11 ensembles, and approximately 4 minutes for RF on a Window XP 3.0GHz machine.

A drawback of LR CERP and LR-T CERP is that it cannot obtain the explicit model. This problem also appears in other ensemble-based methods. However, the main goal of microarray studies is to find an accurate classification model. Furthermore, we experienced a heavy memory consumption and long computation time due to the high dimensionality of data. This problem can be solved by using a parallel computing.

There are a few issues remaining to be investigated. The first one is a variable importance. We can easily extract this information from our CERP model. We may assign a variable importance ranking according to the frequencies of feature appearing in the LR-T CERP model and the variable selection in LR CERP.

We may encounter a problem to classify data whose response contains more than two classes. By modifying the base classifiers, LR CERP or LR-T CERP can be used to classify dichotomous data into multiple classes. A strength of the CERP methodology is that various types of classifiers can be used as a base classifier.

In LR CERP and LR-T CERP, the performance highly depends on the number of partitions and threshold for decision. We plan to study how these parameters affect the accuracy and balance between sensitivity and specificity. Furthermore, we can further improve the algorithm to find an optimal partition size and threshold.

The source codes of LR CERP and LR-T CERP algorithms are implemented using R. After debugging and configuring, an R package for CERP can be developed and contributed to the R library.

Bibliography

[1] Ahn, H., Moon, H., Fazzari, M. J., Lim, N., Chen, J. J., and Kodell, R. L. (2007), "Classification by Ensembles from Random Partitions of High-Dimensional Data," *Computational Statistics and Data Analysis*, **51**, 6166-6179.

[2] Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. (1999), "Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Array," *Proceedings of National Academy of Science*, **96**, 6745-6750.

[3] Ambroise, C., and McLachlan, G. J. (2002), "Selection Bias in Gene Extraction on the Basis of Microarray Gene-Expression Data," *Proceedings of National Academy of Science*, **99**, 6562-6566.

[4] Banerji, R. B. (1980), *Artificial Intelligence: A Theoretical Approach*. North Holland, New York.

[5] Bauer, E. and Kohavi, R. (1999), "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting and Variants," *Machine Learning*, **36**, 105-139.

[6] Blair, R., Fang, H., Branham, W. S., Hass, B., Dial, S. L., Moland, C. L., Tong, W., Shi, L., Perkins, R. and Sheehan, D. M. (2000), "Estrogen Receptor Relative Binding Affinities of 188 Natural and Xenochemicals: Structural

Diversity of Ligands," *Toxicological Sciences*, **54**, 138-153.

[7] Breiman, L. (1996), "Bagging Predictors," *Machine Learning*, **24**, 123-140.

[8] Breiman, L. (1998), "Arcing Classifiers," *The Annals of Statistics*, **26**, 801-849.

[9] Breiman, L. (2001), "Random Forest," *Machine Learning*, **45**, 5-32.

[10] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984), *Classification and Regression Trees*, Belmont, CA: Wadsworth.

[11] Chen, C., Liaw, A., and Breiman, L. (2004), "Using Random Forest to Learn Imbalanced Data," *Technical Report #666*, Department of Statistics, University of California, Berkeley.

[12] Domingos, P. (1999), "Metacost: A General Method for Making classifiers Cost-Sensitive," *Proceedings of the 5th SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM Press, San Diego, CA, 155-164.

[13] Dudoit, S., and Fridlyand, J. (2003), "Classification in Microarray Experiments," In *Statistical Analysis of Gene Expression Microarray data*, T. P. Speed (ed.), Boca Raton, FL: Chapman and Hall/CRC Press, Chapter 3.

[14] Dudoit, S., Fridlyand, J., and Speed, T. P. (2002), "Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data," *Journal of the American Statistical Association*, **97**, 77-87.

[15] Fisher, R.A. (1936), "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics,* **7**, 179-188.

[16] Freund, Y., and Schapire, R. (1996), Experiments with a New Boosting Algorithm, In *Machine Learning: Proceedings of the Thirteenth International Conference,* pp. 148-156.

[17] Freund, Y., and Schapire, R. (1997), "A Decision-Theoretic Generalization of Online Learning and an Application to Boosting," *Journal of Computer and System Science*, **55**, 119-139.

[18] Friedman, J. (1997), "On Bias, Variance, 0/1-Loss, and the Curse-of-Dimensionality," *Data Mining and Knowledge Discovery*, **1**, 55-77.

[19] Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., and Lander, E.S. (1999), "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science*, **286(5439)**, 531-537.

[20] Greally, J. M., (2002), "Short Interspersed Transposable Elements (SINEs) Are Excluded from Imprinted Regions in the Human Genome," *Proceedings of National Academy of Science*, **99**, 327-332.

[21] Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction,* New York, NY: Springer Verlag.

[22] Hileman, B. (1997), "Hormone Disrupter Research Expands," *Chemical Engineering News*, **75**, 24-25.

[23] Kass, C. V. (1980), "An Exploratory Technique for Investigating Large Quantities of Categorical Data," *Journal of Applied Statistics*, **29(2)**, 119-127.

[24] Kuncheva, L. I., Whitaker, C. J., Shipp, C. A., and Duin, R. P. W. (2003), "Limits on the Majority Vote Accuracy in Classifier Fusion," *Pattern analysis and Applications*, **6**, 22-31.

[25] Lam, L., and Suen, C. Y. (1997), "Application of Majority Voting to Pattern Recognition: An Analysis of Its Behavior and Performance," *IEEE Transaction on Systems, man, and Cybernetics*, **27**, 553-568.

[26] Loh, W.-Y., and Shih, Y. S. (1997), "Split Selection Methods for Classification Trees," *Statistica Sinica*, **7**, 815-840.

[27] Molinaro, A. M., Simon, R., and Pfeiffer, R. M. (2005), "Prediction Error Estimation: A Comparison of Resampling Methods," *Bioinformatics,* **21**, 3301-3307.

[28] Moon, H., Ahn, H., Kodell, R. L., Baek, S., Lin, C., Lee, T. and Chen, J. J. (2007), "Ensemble Methods for Classification of Patients for Personal Medicine with High-Dimensional Data," *Artificial Intelligence in Medicine*, In press.

[29] Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T. and Brunk, C. (1994), "Reducing Misclassification Costs: Kowledge-Intensive Approaches to Learning from Noisy Data," *Proceedings of the 11th International Conference on Machine Learning*, New Brunswick, NJ, ML-94, pp217-225.

[30] Prentice, R. L. (1986), "Binary Regression Using an Extended Beta-Binomial Distribution, With Discussion of Correlation Induced by Covariate Measurement Errors," *Journal of the American Statistical Association*, **81**, 321-327.

[31] Quinlan, J. R. (1993), *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers.

[32] Quinlan, J. R. (1996), Bagging, Boosting, and C4.5. *Proceedings, Fourteenth National Conference on Artificial Intelligence,* 1-6.

[33] Reik, W. and Walter, J. (2001), "Genomic imprinting: Parental influence on the genome," *Nature Reviews: Genetics*, **2(1)**, 21-32.

[34] Schapire, R. E. (1990), "The Strength of Weak Learnability," *Machine Learning*, **5**, 197-227.

[35] Schapire, R. E. (2002), "The Boosting Approach to Machine Learning, an Overview," *MSRI Workshop on Nonlinear Estimation and Classification,* **1-23**.

[36] Schapire, R. E., Freund, Y., Bartlett, P. A., and Lee, W. S. (1998), "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods," *The Annals of Statistics*, **26** 1651-1686.

[37] Therneau, T. M., and Atkinson, E. J. (1997), "An Introduction to Recursive Partitioning Using the RPART Routines," *Technical Report*, Mayo Foundation.

[38] Tibshirani, R., Hastie, T., Narasimhan, B. and Chu, G. (2002), "Diagnosis

of Multiple Cancer Types by Shrunken Centroids of Gene Expression," *Proceedings of National Academy of Science*, **99**, 6567-6572.

[39] Tsai, C. A., Chen, C. H., Lee, T. C., Ho, I. C., Yang, U. C. and Chen, J. J. (2004), "Gene Selection for Sample Classifications in Microarray Experiments," *DNA and Cell Biology*, **23**, 607-614.

[40] van't Veer, L. J., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A. M., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Wittenveen, A. T., Schreiber, G. J., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R., and Friend, S. H. (2002), "Gene Expression Profiling Predicts Clinical Outcome of Breast Cancer," *Nature*, **415**, 530-536.

[41] Vapnik, V. (1995), *The Nature of Statistical Learning Theory*, New York, NY: Springer Verlag.

[42] Williams, D. A. (1975), "The Analysis of Binary Responses from Toxicological Experiments Involving Reproduction and Teratogenicity," *Biometrics*, **31**, 949-952

Appendix

A.1 Algorithm for the Cross-Validation of LR CERP

```
1) Separate data $\mathcal{L}$ into 10 subsets $T_1, \ldots, T_{10}$, with roughly equal size for CV.
2) Do for $i$=1 to 10.
    i) Take $T_i$ as test set and $\mathcal{L}_i = \mathcal{L} - T_i$ as learning set.
    ii) Search an optimal partition size ($p_i$) and threshold ($t_i$) for $\mathcal{L}_i$ using 3-fold CV.
    iii) Do for ensemble 1 to 11.
        (a) Randomly partition the predictors of $\mathcal{L}_i$ into $p_i$ subspaces.
        (b) Do for subset 1 to $p_i$.
            1. Fit a full logistic regression as a base classifier.
            2. Apply this model to test set $T_i$.
        (c) End the loop.
        (d) Take an average of fitted values for test set $T_i$.
        (e) Make a decision as 0/1 using threshold ($t_i$).
    iv) End the loop.
    v) Majority vote of the decision is made using 11 ensembles.
3) End the loop.
4) Gather classification results for all the samples.
```

A.2 Algorithm for the Cross-Validation of LR-T CERP

```
1) Separate data $\mathcal{L}$ into 10 subsets $T_1, \ldots, T_{10}$, with roughly equal size for CV.

2) Do for $i$=1 to 10.

    i) Take $T_i$ as test set and $\mathcal{L}_i = \mathcal{L} - T_i$ as learning set.

    ii) Search an optimal partition size ($p_i$) and threshold ($t_i$) for $\mathcal{L}_i$ using 3-fold CV.

    iii) Do for ensemble 1 to 11.

        (a) Randomly partition the predictors of $\mathcal{L}_i$ into $p_i$ subspaces.

        (b) Do for subset 1 to $p_i$.

            1. Build the fully grown tree.
            2. Prune the tree using 1-SE rule
            3. Trim the nodes containing only one class.
            4. Do for terminal node 1 to $k$.
                i. If the sample size is larger than the number of predictor, fit the full logistic regression model
                ii. Else, fit the univariate logistic regression models with each predictor including the intercept term, and the $n-2$ predictors with
```

```
                    smaller deviances plus the intercept term are
                    chosen to be included in the model.
                iii. Apply this model to test set $T_i$.
            5. End the loop.
            6. Take an average of fitted value for test samples $T_i$.
        (c) End the loop.
        (d) Make a decision as 0/1 using threshold ($t_i$).
    iv) End the loop.
    v) Majority vote of the decision is made using 11 ensembles.
3) End the loop.
4) Gather classification results for all the samples.
```

A.3 Algorithm for Searching an Optimal Partition Size and Threshold

```
1) Do for the partition size $p$ as each subspace has around $n/2$, $n/3$, $n/4$, ..., $n/10$
   and $n/12$.
    i) Fit the LR or LR-T CERP model using 3-fold CV.
    ii) Do for the thresholds $ts_j$ = 0.50, 0.52, …, $r$ (or $ts_j = r$, $r$ + 0.02, …, 0.48,
        0.50).
        (a) Apply $ts_j$ to the fitted model.
        (b) Evaluate the accuracy of 3-fold CV model.
        iii) End the loop.
2) End the loop.
3) Do for the thresholds $ts_j$ = 0.50, 0.52, …, $r$ (or $ts_j = r$, $r$ + 0.02, …, 0.48, 0.50).
        i) Choose $p$ with the highest prediction accuracy using $ts_j$.
        ii) Do the bisection method between $n/i$ and $n/(i-1)$ until there is no improvement of prediction accuracy using 3-fold CV and $ts_j$.
        iii) Do the bisection method between $n/i$ and $n/(i+1)$ until there is no improvement of prediction accuracy using 3-fold CV and $ts_j$.
        iv) Take the one with higher overall accuracy.
4) End the loop.
5) Take the pair of $(p,ts)$ with the highest overall accuracy.
```