SMALL SAMPLE FEATURE SELECTION

A Dissertation

by

CHAO SIMA

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2006

Major Subject: Electrical Engineering

SMALL SAMPLE FEATURE SELECTION

A Dissertation

by

CHAO SIMA

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Edward R. Dougherty
Committee Members,	Aniruddha Datta
	Don R. Halverson
	Raymond J. Carroll
Head of Department,	Costas N. Georghiades

May 2006

Major Subject: Electrical Engineering

ABSTRACT

Small Sample Feature Selection. (May 2006)
Chao Sima, B.Eng., Xi'an Jiaotong University
Chair of Advisory Committee: Dr. Edward R. Dougherty

High-throughput technologies for rapid measurement of vast numbers of biological variables offer the potential for highly discriminatory diagnosis and prognosis; however, high dimensionality together with small samples creates the need for feature selection, while at the same time making feature-selection algorithms less reliable. Feature selection is required to avoid overfitting, and the combinatorial nature of the problem demands a suboptimal feature-selection algorithm.

In this dissertation, we have found that feature selection is problematic in small-sample settings via three different approaches. First we examined the feature-ranking performance of several kinds of error estimators for different classification rules, by considering all feature subsets and using 2 measures of performance. The results show that their ranking is strongly affected by inaccurate error estimation. Secondly, since enumerating all feature subsets is computationally impossible in practice, a suboptimal feature-selection algorithm is often employed to find from a large set of potential features a small subset with which to classify the samples. If error estimation is required for a feature-selection algorithm, then the impact of error estimation can be greater than the choice of algorithm. Lastly, we took a regression approach by comparing the classification errors for the optimal feature sets and the errors for the feature sets found by feature-selection algorithms. Our study shows that it is unlikely that feature selection will yield a feature set whose error is close to that of the optimal feature set, and the inability to find a good feature set should not lead

to the conclusion that good feature sets do not exist.

To my beloved grandma

ACKNOWLEDGMENTS

I'd like to thank my advisor, Dr. Edward Dougherty, for his kind guidance for all these years. Especially I feel grateful for his patience and encouragement during the time I was struggling. I couldn't have gone this far without his vision and advice.

At the same time I'd like thank Dr. Aniruddha Datta, Dr. Don Halverson and Dr. Raymond J. Carroll: not only because they are taking their valuable time and serving on my committee, but also their willingness to help. They have been, and will continue to be, an inspiration for me.

Last but not least, I want to thank my parents and my fiancée for standing firmly behind me during the years past, and in the years ahead.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

cDNA microarray technology has made it possible for researchers to investigate behaviors of thousands of features (genes) all at once [1][2][3]; on the other hand, the number of samples (tissues) is relatively very small, especially when dealing with human subjects, in which cases the number can be in the range from 20 to 50. The following sample sizes for cancer studies are indicative of the commonplace paucity of data points: cutaneous melanoma, 31 [4]; leukemia, 37 [5]; acute leukemia, 38 [6]; breast cancer, 38 [7], follicular lymphoma, 24 [8]; uveal melanoma, 20 [9], glioma, 50 (but only 21 classic tumors used for class prediction) [10]; ovarian carcinoma, 44 [11]; lymphoma, 47 [12]; and glioma, 25 [13]. More than often, we are interested in finding a gene set with which we can classify the samples into, for example, normal and cancer tissues, or tissues with different stages of cancers, with minimum errors. The imbalance between the number of potential features and sample size poses a problem for this classification, known as *small sample issue* [14]. For instance, with small-sample classifier design, one is limited to small feature sets to avoid overfitting [15][16][17]. Therefore, classification in small sample setting consists of three major parts: feature selection, classifier design and error estimation.

It is important to notice that the three stages are not independent of each other: for example, feature selection is part of the classification rule, and error estimation is inside feature selection algorithms for evaluating criterion functions. As a result, even our main interest is in feature selection, we will have to study these interacting factors to gain a thorough understanding of the problem in small sample settings.

The journal model is *IEEE Transactions on Automatic Control.*

The dissertation is organized as following: we first devote Chapter II to reviewing some of the classical and recently proposed error estimation methods; then we will study the impact of error estimation on feature ranking (Chapter III) and feature selection (Chpater IV). Next in Chapter V we study the general problem of feature selection via a regression approach. Lastly in Chapter VI we draw some concluding remarks.

CHAPTER II

CLASSIFIER ERROR ESTIMATION

Error estimation plays an important role in feature selection, the details of which will be studied in later chapters. In this chapter, we will introduce some classical and recently proposed error estimation techniques.

A. Classical Error Estimation Problem

In two-group statistical pattern recognition, there is a *feature vector* $X \in \mathbb{R}^p$ and a *label* $Y \in \{0, 1\}$. The pair (X, Y) has a joint probability distribution $\mathbf{F}$, which is unknown in practice. Hence, one has to resort to designing classifiers from *training data*, which consists of a set of n independent observations, $S_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, drawn from $\mathbf{F}$. A *classification rule* is a mapping $g : \{\mathbb{R}^p \times \{0, 1\}\}^n \times \mathbb{R}^p \to \{0, 1\}$. It maps S_n into the *designed classifier* $g(S_n, \cdot) : \mathbb{R}^p \to \{0, 1\}$. In fact, a classification rule is actually a collection of mappings, one for each n; however, we follow the usual practice of using a single operator notation g to represent all of the individual mappings. The *true error* of a designed classifier is its error rate given the training data set:

$$\epsilon_n[g|S_n] = P(g(S_n, X) \neq Y) = E_{\mathbf{F}}(|Y - g(S_n, X)|), \tag{2.1}$$

where $E_{\mathbf{F}}$ denotes expectation with respect to $\mathbf{F}$. The expected error rate over the data is given by $\epsilon_n[g] = E_{\mathbf{F}_n} E_{\mathbf{F}}(|Y - g(S_n, X)|)$, where $\mathbf{F}_n$ is the joint distribution of the training data S_n. Were the underlying feature-label distribution $\mathbf{F}$ known, the true error could be computed exactly via (2.1). In practice, one must use an *error estimator*. Ideally, this estimate should be be fast to compute and as close as possible to the true error, for the given training data.

B. Classical Error Estimation

The simplest way to estimate the error of a designed classifier in the absence of independent test data is to compute its error directly on the sample data itself. This *resubstitution estimator*, $\hat{\epsilon}_{\text{resub}}$, is very fast, but is usually optimistic (i.e., biased low) as an estimator of $\epsilon_n[g]$. For some classification rules, resubstitution can be severely low-biased, an extreme case being one-nearest-neighbor classification, in which the resubstitution estimator is identically zero. Typically, the more complex the classifier is, the more optimistic resubstitution is, since complex classifiers tend to overfit the data, especially with small samples [18].

Cross-validation removes the optimism from resubstitution by employing test points not used in the design of the classifier. In *k-fold cross-validation*, the data set S_n is partitioned into k *folds* $S_{(i)}$, for $i = 1, \ldots, k$ (for simplicity, we assume that k divides n). Each fold is left out of the design process and used as a test set, and the estimate, $\hat{\epsilon}_{\text{cvk}}$, is the overall proportion of error on all folds. The process may be repeated: several cross-validation estimates are computed using different partitions of the data into folds, and the results are averaged. A k-fold cross-validation estimator is unbiased as an estimator of $\epsilon_{n-n/k}[g]$. The *leave-one-out estimator*, $\hat{\epsilon}_{\text{loo}}$, in which a single observation is left out each time, corresponds to n-fold cross-validation. It is unbiased as an estimator of $\epsilon_{n-1}[g]$. Cross-validation estimators are often pessimistic, since they use smaller training sets to design the classifier. Their main drawback is their variance [19][17]. They can also be quite slow to compute when the number of folds or samples is large.

The bootstrap error estimation technique [20][21] is based on the notion of an "empirical distribution" $\mathbf{F}^*$, which serves as a replacement to the original unknown distribution $\mathbf{F}$. The empirical distribution puts mass $\frac{1}{n}$ on each of the n available data

points. A "bootstrap sample" S_n^* from $\mathbf{F}^*$ consists of n equally-likely draws with replacement from the original data S_n. The basic *bootstrap zero estimator*[21] is written in terms of the empirical distribution as $\hat{\epsilon}_0 = E_{\mathbf{F}^*}\left(\,|Y - g(S_n^*, X)|\colon (X,Y) \in S_n \setminus S_n^*\,\right)$. In practice, the expectation $E_{\mathbf{F}^*}$ has to be approximated by a Monte-Carlo estimate based on independent replicates S_n^{*b}, for $b = 1, \ldots, B$. The bootstrap zero estimator works like cross-validation: the classifier is designed on the bootstrap sample and tested on the original data points that are left out. It tends to be high-biased as an estimator of $\epsilon_n[g]$, since the amount of samples available for designing the classifier is on average only $(1 - e^{-1})n \approx 0.632n$. The *.632 bootstrap estimator* [21], $\hat{\epsilon}_{\text{b632}} = (1 - 0.632)\,\hat{\epsilon}_{\text{resub}} + 0.632\,\hat{\epsilon}_0$, tries to correct this bias by doing a weighted average of the bootstrap zero and resubstitution estimators. It has low variance, but can be extremely slow to compute. In addition, it can fail when resubstitution is too low-biased [19].

C. Bolstered Error Estimation

A relatively new error estimation technique is introduced in [22], which is proved especially effective in small sample classification problems. We shall briefly explain it as following.

The resubstitution estimator is defined in terms of the empirical feature-label distribution F^* by $\hat{\varepsilon}_n^R = E_{F^*}[|Y - g(S_n, \mathbf{X})|]$. Relative to F^*, no distinction is made between points near or far from the decision boundary. If one spreads the probability mass at each point of the empirical distribution, then variation is reduced because points near the decision boundary will have more mass on the other side of the boundary than will points far from the decision boundary. To take advantage of this observation, consider a probability density function $f_i^{\diamondsuit}$, for $i = 1, \ldots, n$, called a

bolstering kernel, and define the *bolstered empirical distribution* $F^{\diamondsuit}$, with probability density function given by $f^{\diamondsuit}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i^{\diamondsuit}(\mathbf{x} - \mathbf{x}_i)$. The *bolstered resubstitution* estimator [22] is obtained by replacing F^* by $F^{\diamondsuit}$ in the definition of $\hat{\varepsilon}_n^R$ to obtain

$$\hat{\varepsilon}_n^{\diamondsuit R} = E_{F^{\diamondsuit}}[|Y - g(S_n, \mathbf{X})|]. \tag{2.2}$$

A computational expression for the bolstered resubstitution estimator is given by

$$\hat{\varepsilon}_n^{\diamondsuit R} = \frac{1}{n}\sum_{i=1}^{n}\left(I_{y_i=0}\int_{A_1} f_i^{\diamondsuit}(x - x_i)\,dx + I_{y_i=1}\int_{A_0} f_i^{\diamondsuit}(x - x_i)\,dx\right), \tag{2.3}$$

where $A_j = \{x \mid g(S_n, x) = j\}$. The integrals are the error contributions made by the data points, according to whether $y_i = 0$ or $y_i = 1$. The bolstered resubstitution error estimate is equal to the sum of all error contributions divided by the number of points. If the classifier is linear, then the decision boundary is a hyperplane and it is usually possible to find analytical expressions for the integrals; otherwise, Monte-Carlo integration can be employed:

$$\hat{\varepsilon}_n^{\diamondsuit R} \approx \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{M} I_{x_{ij}\in A_1} I_{y_i=0} + \sum_{j=1}^{M} I_{x_{ij}\in A_0} I_{y_i=1}\right), \tag{2.4}$$

where $\{x_{ij}\}_{j=1,\ldots,M}$ are samples drawn from the distribution $f_i^{\diamondsuit}$. The experiments in [22] indicate that a small number M of Monte-Carlo samples is needed (in our simulations, a value $M = 10$ was adequate, and increasing M beyond that did not substantially reduce the variance of the estimator). Fig. 1 illustrates the situation where the bolstering kernels are given by uniform circular distributions and the classifier is linear. In this case, no Monte-Carlo computation is needed; the bolstered resubstitution error estimate is given in terms of the areas of the shaded regions.

When resubstitution is strongly low-biased, it may not be good to spread incor-

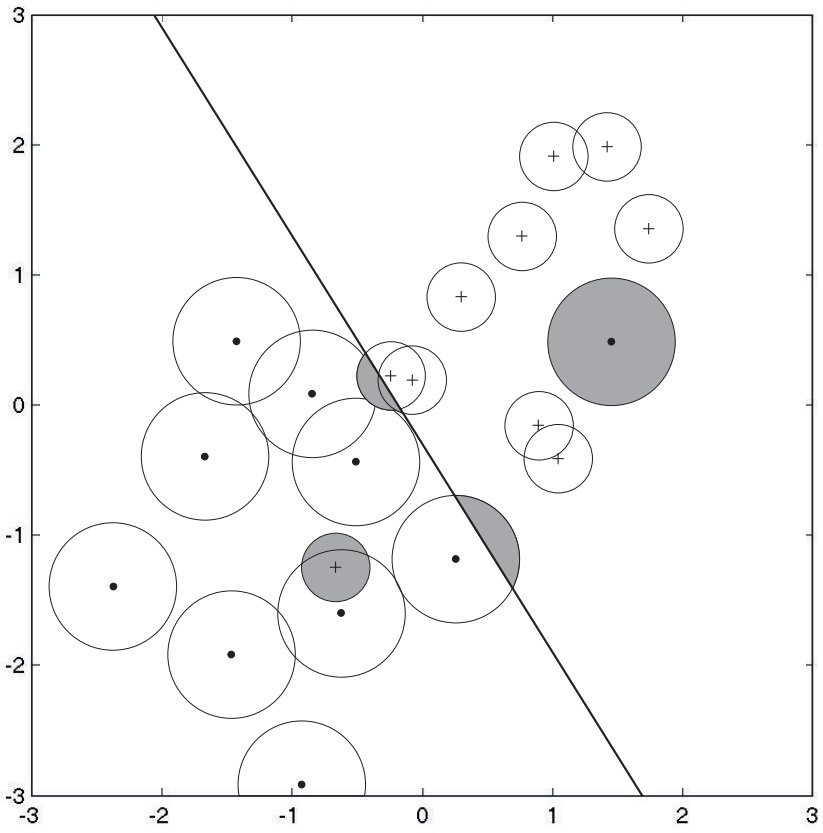


Fig. 1. Bolstered resubstitution for a linear classifier, assuming uniform circular bolstering kernels. The area of each shaded region divided by the area of the associated circle is the error contribution made by a point. The bolstered resubstitution error is the sum of all contributions divided by the number of points.

rectly classified data points, as that increases optimism of the error estimator. Bias is reduced by using no bolstering for incorrectly classified points. The result is the *semi-bolstered resubstitution* estimator [22].

Bolstering can be applied to any error-counting error estimation method. For the leave-one-out estimation, let S_{n-1}^i denote the data set resulting from deleting data point i from the original data set S_n and $A_j^i = \{x \mid g(S_{n-1}^i, x) = j\}$, for $j = 0, 1$, be the the decision region for the classifier designed from S_{n-1}^i. The *bolstered leave-one-out* estimator [22] can be computed via

$$\hat{\epsilon}_{\text{loo}}^{\diamond} = \frac{1}{n} \sum_{i=1}^{n} \left(I_{y_i=0} \int_{A_1^i} f_i^{\diamond}(x - x_i)\, dx + I_{y_i=1} \int_{A_0^i} f_i^{\diamond}(x - x_i)\, dx \right). \tag{2.5}$$

When the integrals cannot be computed exactly, a Monte-Carlo expression like (2.4) can be used.

1. Choosing Bolstering Kernel

Although more general bolstering kernels may be considered, in keeping with the principle of not making complicated inferences from a limited amount of data, we only consider zero-mean, spherical bolstering kernels $f_i^{\diamond}$, with covariance matrices of the form $\sigma_i^2 I_p$. In each case there is a family of bolstered estimators, corresponding to the choices of the standard deviations $\sigma_1, \ldots, \sigma_n$. The choice of these parameters determines the variance and bias properties of the corresponding bolstered estimator. If $\sigma_i = 0$, for $i = 1, \ldots, n$, then there is no bolstering and the bolstered estimator reduces to the original estimator. As a general rule, larger σ_i's, i.e., "wider" bolstering kernels, lead to lower-variance estimators, but after a certain point this advantage becomes offset by increasing bias.

The choice of the standard deviations is a critical issue. A non-parametric sample-based method to choose these parameters that is applicable in small-sample

settings has been proposed [22]. The method, together with details about bolstering using Gaussian kernels (the kind used in this paper), is described below.

2. Choosing the Amount of Bolstering

When bolstering resubstitution, the aim is to select the parameters so that the bolstered resubstitution estimator is nearly unbiased. One can think of (X, Y) in (2.1) as a random test point. Given that $Y = y$, this test point is at a "true mean distance" $\delta(y)$ from the data points belonging to class y. This distance is determined by the underlying class-conditional distribution $F(X|Y = y)$. One reason why plain resubstitution is optimistically biased is that the test points are all at distance zero from the training data. Since bolstered estimators spread the test points, the task is to find the amount of spreading that makes the test points to be as close as possible to the true mean distance to the training data points. The true mean distance can be estimated by its sample-based estimate:

$$\hat{d}(y) = \frac{\sum_{i=1}^{n} \min_{j \neq i} \{||x_i - x_j||\} : I_{y_i = y}}{\sum_{i=1}^{n} I_{y_i = y}}. \tag{2.6}$$

The estimate $\hat{d}(y)$ is the mean minimum distance between points belonging to class y.

Let $f_i^{\diamond,1}$ be a unit-variance bolstering kernel, and let D_i be the random variable equal to the distance of a point randomly selected from $f_i^{\diamond,1}$ to the origin. Let $F_{D_i}(x)$ be the cdf of D_i. In the case of the bolstering kernel $f_i^{\diamond}$ with variance $\sigma_i^2 I_p$, all distances get multiplied by σ_i. We find the value of σ_y for class y such that the median distance of a test point to the origin is equal to the estimated true mean distance $\hat{d}(y)$, so that half of the test points will be farther from the center than $\hat{d}(y)$, and the other half will be nearer. Hence, σ_y is the solution of the equation $\sigma_y F_{D_i}^{-1}(1/2) = \hat{d}(y)$. Note

that

$$\alpha_{p,i} = F_{D_i}^{-1}(1/2) \tag{2.7}$$

can be viewed as a constant "correction" factor, which can be computed and stored off-line. The subscript p indicates explicitly that the correction factor is a function of the dimensionality. The estimated standard deviations for the bolstering kernels are thus given by:

$$\sigma_i = \frac{\hat{d}(y_i)}{\alpha_{p,i}}, \quad \text{for } i = 1, \ldots, n. \tag{2.8}$$

Clearly, as the number of samples in the training data increases, the standard deviations σ_i decrease, and there is less bias correction introduced by the bolstered resubstitution. This is in accordance with the fact that resubstitution tends to be less optimistically-biased as the sample size increases.

Let us consider now the leave-one-out estimator. In this case, no bias-correction is necessary or desired; the aim is solely reducing the variance of the estimator. Considering the distance argument, we see that each point left out in the the design of the classifier g is an independent sample and is already at the right distance to the design data set (this is the reason for the unbiasedness of leave-one-out as as estimator of $\epsilon_{n-1}[g]$). Therefore, we propose to use the minimum distance $d(x_i, S_{n-1}^i)$ of each point to the rest of the data set as the basis for selecting the standard deviation of the corresponding bolstering kernel $f_i^{\diamond}$. As before, we want half of the test points to be farther from the center than $d(x_i, S_{n-1}^i)$, and the other half to be nearer. Therefore, the standard deviations are distinct for each data point, and given by

$$\sigma_i = \frac{d(x_i, S_{n-1}^i)}{\alpha_{p,i}}, \quad \text{for } i = 1, \ldots, n, \tag{2.9}$$

where $\alpha_{p,i}$ is the correction factor in (2.7).

3. Gaussian-Bolstered Error Estimation

An important case of bolstering, which is the one assumed in this paper, is the choice of Gaussian kernels:

$$f_i^{\diamond}(x) = \frac{1}{(2\pi)^{p/2}\sigma_i^p} \exp\left(-\frac{||x||^2}{2\sigma_i^2}\right). \tag{2.10}$$

For a general classifier, the integrals in (2.3) and (2.5) have to be computed by Monte-Carlo sampling. For a linear classifier, however, analytical expressions are possible. For example, for LDA (Linear Discriminant Analysis), the Gaussian-bolstered resubstitution error estimate is given by (see [22] for a proof):

$$\hat{\epsilon}_{\text{resub}}^{\diamond} = \frac{1}{n}\sum_{i=1}^{n} \left(\Phi_{\sigma_i}(W_a(x_i))I_{y_i=0} + \Phi_{\sigma_i}(-W_a(x_i))I_{y_i=1}\right), \tag{2.11}$$

where Φ_{σ_i} is the cumulative distribution function of a zero-mean Gaussian random variable with variance σ_i^2, and W_a is the normalized W statistic, given by $W_a(x) = (a^T x + m)/||a||$, with

$$\begin{aligned} a &= \Sigma^{-1}(\mu_1 - \mu_0) \\ m &= \frac{1}{2}(\mu_0 + \mu_1)^T \Sigma^{-1}(\mu_0 - \mu_1). \end{aligned}$$

Here, $\Sigma = \frac{1}{2}(\Sigma_0 + \Sigma_1)$ is the pooled covariance matrix, with μ_i and Σ_i denoting the mean and covariance matrix for class i, respectively, which are obtained via their usual maximum-likelihood estimates. The parameters a and m specify the separating hyperplane produced by LDA: a is a vector normal to the hyperplane, and $m/||a||$ is its distance to the origin.

A similar expression to (2.11) applies to the Gaussian-bolstered leave-one-out.

Note that $\Phi_\sigma(0) = 1/2$, which corresponds to the error contribution of a point on the decision boundary. As $\sigma_i \to 0$, for $i = 1, \ldots, n$, then all functions Φ_{σ_i} collapse to indicator step functions and the Gaussian-bolstered error estimator reduces to the

original estimator. On the other hand, if $\sigma_i \to \infty$, for $i = 1, \ldots, n$, then the functions Φ_{σ_i} become constant and equal to $\frac{1}{2}$, so that the bolstered estimator is identically equal to $\frac{1}{2}$, regardless of the data. This estimator has zero variance, but is of course not useful.

The actual values of σ_i in a practical situation are computed according to the distance-based scheme outlined in the previous subsection. In the present Gaussian case, the distance variables D_i are distributed as a *chi* random variable D with p degrees of freedom. The density function of D is given by [23]:

$$f_D(x) = \frac{2^{1-p/2} x^{p-1} e^{-x^2/2}}{\Gamma(\frac{p}{2})}, \tag{2.12}$$

where Γ is the gamma function. For $p = 2$, this becomes the well-known Rayleigh density. The cdf F_D can be computed by numerical integration of (2.12), and the inverse at point 1/2 can be found by a simple binary search procedure (using the fact that F_D is monotonically increasing), which yields the correction factor α_p. For instance, the values of the correction factor up to five dimensions are: $\alpha_1 = 0.674$, $\alpha_2 = 1.177$, $\alpha_3 = 1.538$, $\alpha_4 = 1.832$, $\alpha_5 = 2.086$.

CHAPTER III

IMPACT OF ERROR ESTIMATION ON THE PERFORMANCES OF FEATURE RANKING *

Ranking feature sets is a key issue for classification, for instance, phenotype classification based on gene expression. Often we are interested in selecting a list of potential genes with which to classify tissues, and the top ranked genes become a natural choice.

Since ranking is often based on error estimation, and error estimators suffer to differing degrees of imprecision in small-sample settings, it is important to choose a computationally feasible error estimator that yields good feature-set ranking.

This chapter examines the feature-ranking performance of several kinds of error estimators: resubstitution, cross-validation, bootstrap, and bolstered error estimation. It does so for three classification rules: linear discriminant analysis (LDA), 3-nearest-neighbor classification (3NN), and classification trees (CART). Two measures of performance are considered. One counts the number of the truly best feature sets appearing among the best feature sets discovered by the error estimator and the other computes the mean absolute error between the top ranks of the truly best feature sets and their ranks as given by the error estimator. Our results indicate that performances using different error estimation techniques vary and generally suffer from lack of data; specifically, bolstering is superior to bootstrap, and bootstrap is better than cross-validation, for discovering top-performing feature sets for classification when using small samples. A key issue is that bolstered error estimation is tens of times faster than bootstrap, and faster than cross-validation, and is therefore

*Reprinted with permission from "Superior Feature-Set Ranking For Small Samples Using Bolstered Error Estimation" by C. Sima, U. Braga-Neto and E.R. Dougherty, 2005, *Bioinformatics*, Vol. 21, No. 7, pp1046-1054. Copyright 2005 by Oxford University Press.

feasible for feature-set ranking when the number of feature sets is extremely large.

A. Introduction

When choosing among a collection of potential feature sets for classification, estimating the errors of designed classifiers is a key issue; indeed, it is natural to order the potential feature sets according to the misclassification rates of their corresponding classifiers. Hence, it is important to apply error estimators that provide rankings that better correspond to rankings produced by the true errors. For phenotype classification based on gene expression, feature selection can be viewed as gene selection: find sets of genes whose expressions can be used for phenotypic discrimination. In recent years, gene selection has been heavily investigated.

A critical issue for classification via microarray data is the frequent presence of small samples and the consequences flowing therefrom [14]. For instance, with small-sample classifier design, one is limited to small feature sets to avoid overfitting [15][16][17]. While this may be an impediment, small gene sets are advantageous relative to the very expensive and time-consuming analysis required to determine if they could serve as useful targets for therapy. In any event, since all feature-selection algorithms are subject to significant errors when samples are small, in the context of microarray experiments, it is prudent to approach feature selection as finding a list of potential feature sets, and not as trying to find a best feature set. Indeed, the entire matter of feature selection and classification in the context of small samples can be conservatively viewed as an exploratory methodology. This conservative position has been articulated in the following manner: "Most likely, it will not be possible to design a classifier from a single set of microarray experiments. Separation of the sample data by designed classifiers will likely have to be taken as evidence that the corresponding

gene sets are potential variable sets for classification. Their effectiveness will have to be checked by large-replicate experiments designed to estimate their classification error, perhaps in conjunction with biological input or phenotype evidence. There may, in fact, be many gene sets that provide accurate classification of a given pathology. Of these, some sets may provide mechanistic insights into the molecular etiology of the disease, while other sets may be indecipherable"[14]. For instance, this approach has been explicitly taken in the case of discovering markers for different types of glioma, where the number of available tissue samples is severely limited [13]. That study states, "We have identified robust classifier gene sets containing one to three genes that distinguish each type of glioma from the other three. This provides guidance for the development of pathological assays using a reasonable number of markers for clinical use."

The raw data associated with microarray experiments usually contain an extraordinarily large number of gene expression measurements, in the order of tens of thousands. On any given microarray, many of these measurements fall below an acceptable quality level. In the case of the software provided with the Affymetrix platform, an unacceptable signal-to-noise ratio is quantified by a bad "detection" p-value [24]. For spotted cDNA microarrays, the DeArray software of the National Human Genome Research Institute calculates a multi-faceted quality metric for each spot [25]. This quality problem is a result of imperfections in RNA preparation, hybridization to the arrays, scanning, and also intrinsic factors, such as low expressed genes. Genes whose expressions fail to be effectively detected on a large number of microarrays are rejected from further consideration. Furthermore, many of the reliably-detected genes possess expression values that do not change appreciably across the microarrays in the experiment – for instance, "house-keeping" genes. These genes can also be removed from consideration, by means of a simple variance filter, since they clearly cannot

contribute to discrimination. This *pre-filtering* process usually reduces the number of variables by an order of magnitude. One then proceeds to apply a feature selection algorithm to obtain small feature sets (combinations of genes). Feature selection can be either optimal, which requires that *all* possible feature sets of a given size are examined [26], or sub-optimal. If pre-filtering reduces the number of potential features to around a thousand, it becomes computationally possible to employ optimal feature selection and examine all possible two- and three-gene feature sets. Larger numbers of potential features or larger feature sets are possible in an appropriate supercomputer environment [27]. If the initial number of genes to be considered, after pre-filtering, is too large, or if the size of the feature sets is large, then a sub-optimal method must be employed (and here we include the branch-and-bound algorithm [28] as suboptimal because monotonicity of the error measure can fail significantly for small samples). It is not uncommon to apply a second filtering (say, by standard t-tests) to further reduce the number of features, and then follow this by an optimal or sub-optimal selection process.

A natural way to measure the performance of an error estimator relative to feature-set ranking is to measure the degree to which application of the estimator yields a ranking that reflects the ranking based on the true errors of the classifiers designed for the feature sets. Here we will consider two performance measures. The first counts the number of top feature sets based on the true error that are rated as top feature sets based on the estimated error. For feature (gene) discovery, this performance measure is critical because the features discovered based on the data will be the ones listed best based on error estimation, and we would like that list to contain a good supply of truly good feature sets. A second measure computes the mean deviation between the rankings of the top feature sets (based on true error) and their corresponding rankings based on error estimation.

A perusal of the literature shows that cross-validation methods (especially leave-one-out estimation) are often used for error estimation during feature selection; however, cross-validation estimators display high variance [17]. This variance results in a widely dispersed deviation distribution (deviation between the true and estimated errors of a classifier), thereby making cross-validation unreliable for small samples [19]. In a previous paper, it has been demonstrated that, for small samples, leave-one-out cross-validation-based feature ranking does not outperform resubstitution-based feature ranking on the best feature sets, these being the ones whose designed classifiers possess the smallest errors [29]. Owing to typical experimental methodology, the conclusions of that paper are too narrow. While it is theoretically revealing to know that a popular cross-validation procedure does not outperform resubstitution on the best feature sets, in practice we do not know the best feature sets and must draw our conclusions from feature sets ranked according to an error estimator. Thus, we are presented with a list of feature sets whose errors are estimated, and further investigation – for instance, laboratory analysis to determine the biological basis of discrimination – will proceed based on the list. Owing to imprecision in error estimation, an experimentally derived list is likely to contain among its best feature sets some that are not truly the best. Hence, in evaluating error estimators we cannot limit our view to the best feature sets; otherwise, we will not take into account the confusion created by mediocre (or even poor) feature sets appearing at the top of an experimentally derived list.

Going further, we do not want to limit ourselves to leave-one-out cross validation and resubstitution. Admittedly, these are computationally efficient compared to replicated cross-validation and bootstrap, but as we will see, they are among the worst performers relative to ranking. Indeed, .632 bootstrap generally outperforms cross-validation methods (the performances of which vary widely), the exception be-

ing for the best feature sets, where the performances of all the tested estimators do not differ greatly. Owing to its high computational complexity, bootstrap is not feasible for ranking very large collections of feature sets; nonetheless, owing to its generally superior performance to cross-validation, it can serve as a benchmark. The recently proposed *bolstered error estimation* [22] not only outperforms cross-validation for feature-set ranking, but also outperforms .632 bootstrap, even though the bootstrap takes tens of times longer to compute than the bolstered estimators.

We use simulation studies to analyze feature-set ranking for a number of cross-validation, bootstrap, and bolstered error estimators. The use of simulation studies is commonplace for feature-selection analysis [30][31]. We conduct two large studies, one based on a Gaussian mixture model that allows us to vary a number of parameters, and the other based on patient data from a large microarray breast cancer study. In both studies we consider linear discriminant analysis (LDA), 3-nearest-neighbor classification (3NN), and classification trees (CART). We will present detailed analysis for one case from each study.

B. Ranking Feature Sets

We consider two performance measures concerning how well feature ranking using the error estimators agrees with feature ranking based on the true errors. Since our main interest is in finding good feature sets, say the best K feature sets, we wish to compare the rankings of the K best estimate-based feature sets with those of the K best based on the true errors. Moreover, in a similar vein to [29], we want to make this comparison for feature sets whose true performances attain certain levels. For $t > 0$, let $\mathcal{G}_t^K$ be the collection of all feature sets of a given size whose true errors are less than t, where $\mathcal{G}_t^K$ is defined only if there exists at least K feature sets with true

error less than t. Rank the best K feature sets according to their true errors and rank all features sets in $\mathcal{G}_t^K$ according to their estimated errors, with rank 1 corresponding the lowest error. We then have two ranks for each of the K best feature sets: k (true) and k^* (estimated) for all feature sets in $\mathcal{G}_t^K$. In case of ties, the rank is equal to the mean of the ranks. It should be noted that the selection of feature sets for inclusion in $\mathcal{G}_t^K$ is based on the true error and is therefore not subject to the kind of selection bias discussed in [32]. Moreover, any selection bias that might occur in the ranking based on error estimation is part of the estimation-based ranking process and its effect is *ipso facto* incorporated into the ranking analysis.

If our interest is in feature discovery, then a key interest is whether truly important features appear in the list of important feature sets based on error estimation. This is the list we obtain from data analysis, and good classification depends on discovering truly good classifying feature sets. Moreover, in gene discovery, the ultimate analysis is not that based on the classification data, but is instead the laboratory analysis of genes discovered via classification, and therefore we would like the classification methodology to produce key genes. The first performance statistic counts the number of feature sets among the top K feature sets that also appear in the top K using the error estimator,

$$R_1^K(t) = \sum_{k=1}^{K} I_{k^* \leq K}. \tag{3.1}$$

where I_A denotes the indicator function. For this measure, higher scores are better. Since k^* is the estimate-based rank of the k^{th} true-ranked feature set among the feature sets in $\mathcal{G}_t^K$ and since we only consider feature sets in $\mathcal{G}_t^K$, the larger t, the larger the collection of ranks k^* and the greater possibility that erroneous feature sets appear among the top K, thereby resulting in a smaller value of $R_1^K(t)$. As will be seen in the experimental results, the curve of $R_1^K(t)$ will flatten out for increasing t, which

is reflective of the fact that, as we consider ever poorer feature sets, their effect on the top ranks becomes negligible owing the fact that inaccuracy in the measurement of their errors is not sufficient to make them confuse the ranking of the best feature sets.

The second performance metric measures the mean absolute deviation in the ranks for the K best features sets,

$$R_2^K(t) = \frac{1}{K}\sum_{k=1}^{K}|k-k^*|. \tag{3.2}$$

For this measure, lower scores are better. In analogy to $R_1^K(t)$, the larger t, the larger the collection of ranks k^* and the greater possible deviation between k and k^*. When t is small, rank comparison is only being made between (truly) good feature sets, which was the interest in [29]. Our interest here is broader. Not only are we interested in a wide variety of error estimators, but we are concerned with the pragmatic issue of having to rank feature sets based on error estimates without necessarily having any *a priori* restriction on the goodness of the feature sets being considered. Hence, we are interested in large t, and in analogy to $R_1^K(t)$ the curve for $R_2^K(t)$ will flatten out as t increases.

C. Experimental Results

We consider two basic sets of experiments, one using synthetic data generated from a model based on Gaussian class conditional distributions, and another using microarray data categorizing a breast-cancer patient prognosis. In both cases we consider three classification rules: linear discriminant analysis (LDA), 3-nearest neighbor (3NN), and classification and regression trees (CART). In all cases we consider a sample size of 30, do the analysis for 2 and 3 features, and consider top lists of sizes $K = 20$

and $K = 40$. We shall provide detailed analysis for one Gaussian case and one from the breast-cancer data.

1. Synthetic Data

The synthetic data used in our experiments is based on a Gaussian model, under which the classes are equally likely and the class-conditional densities are spherical unit-variance Gaussians. The class means are located at δa and $-\delta a$, where $\delta > 0$ is a separation parameter and $a = (a_1, a_2..., a_n)$ is a parameter vector with $||a|| = 1$. The Bayes classifier is a hyperplane perpendicular to the axis joining the means, with Bayes error $\epsilon_{BAYES} = 1 - \Phi(\delta)$, where Φ is the standard normal cumulative distribution function. Since $\delta = \Phi^{-1}(1 - \epsilon_{BAYES})$, one can find δ for a prescribed Bayes error. If a subset L of the original variables is selected, then again one has a standard Gaussian model, but now the separation between the classes is a function of which variables are selected. The Bayes error is a function of both the separation and the model parameters, specifically, $\epsilon^L_{BAYES} = 1 - \Phi\left(\delta \sqrt{\sum_{k \in L} a_k^2}\right)$. To minimize ϵ^L_{BAYES} for a given number of selected variables, one should pick the variables corresponding to the largest parameters.

For the simulation, we let the total number of variables in the Gaussian model be 20 and consider feature sets of sizes 2 and 3. The separation parameter δ is chosen so that the Bayes error in the space of dimension corresponding to the feature-set sizes of 2 and 3 is 0.05 or 0.10, respectively. We consider equal or unequal (1 and 1.5) class-conditional standard deviations. The parameter vector $a = (a_1, a_2..., a_n)$ is picked from a sigmoidal distribution in order to favor a few of the feature sets and make the rest unimportant. We generate 200 independent samples of size 30. For each, we apply the three classification rules, LDA, 3NN, and CART, with all possible feature sets, and apply the different error estimators to compute the statistics $R_1^K(t)$

and $R_2^K(t)$. The number of feature sets for which each statistic is computed depends on the maximum true error threshold t. For a given feature set size, classification rule, and error estimator, we can average $R_1^K(t)$ and $R_2^K(t)$. There is a proviso here: for small t there may not be K feature sets satisfying the threshold for all samples of size 30, and therefore we only consider those samples for which there are K sets satisfying the threshold.

Fig. 2 provides the $R_1^{40}(t)$ and $R_2^{40}(t)$ curves for the synthetic data, in the equal-variance case and with feature sets of size 3 , for resubstitution (resub), leave-one-out cross-validation (loo), 10-fold cross-validation with replications (cv10r), 0.632 bootstrap (b632), bolstered resubstitution (bresub), semi-bolstered resubstitution (sresub), and bolstered leave-one-out (bloo), for the three assumed classification rules. Each plot in Fig. 2 assumes a range of maximum true error threshold $t = 0.25$ through $t = 0.50$. Table I shows two statistics for a few values of t: s_1 is the average error for all feature sets having error less than t, which is the average error among those feature sets for which the performance statistics have been computed, while s_2 is the average number of feature sets having error less than t.

For LDA, Fig. 2 shows that bolstered resubstitution performs best over the entire range of t, with the other bolstered estimators also performing better than the .632 bootstrap. Both cross-validation estimators, loo and cv10r, perform about the same as resubstitution, with the latter three all performing much worse than the .632 bootstrap. In our experiments it is seen that cv10 is by far the poorest among all estimators considered. The quantitative interpretation of the difference in performance is that, on average, bolstered resubstitution will correctly discover two more feature sets among the top 40 than will .632 bootstrap, and the latter will discover two more than loo or the heavily computational cv10r, neither of which perform substantially better than resubstitution. Fig. 2 shows that the pattern shown

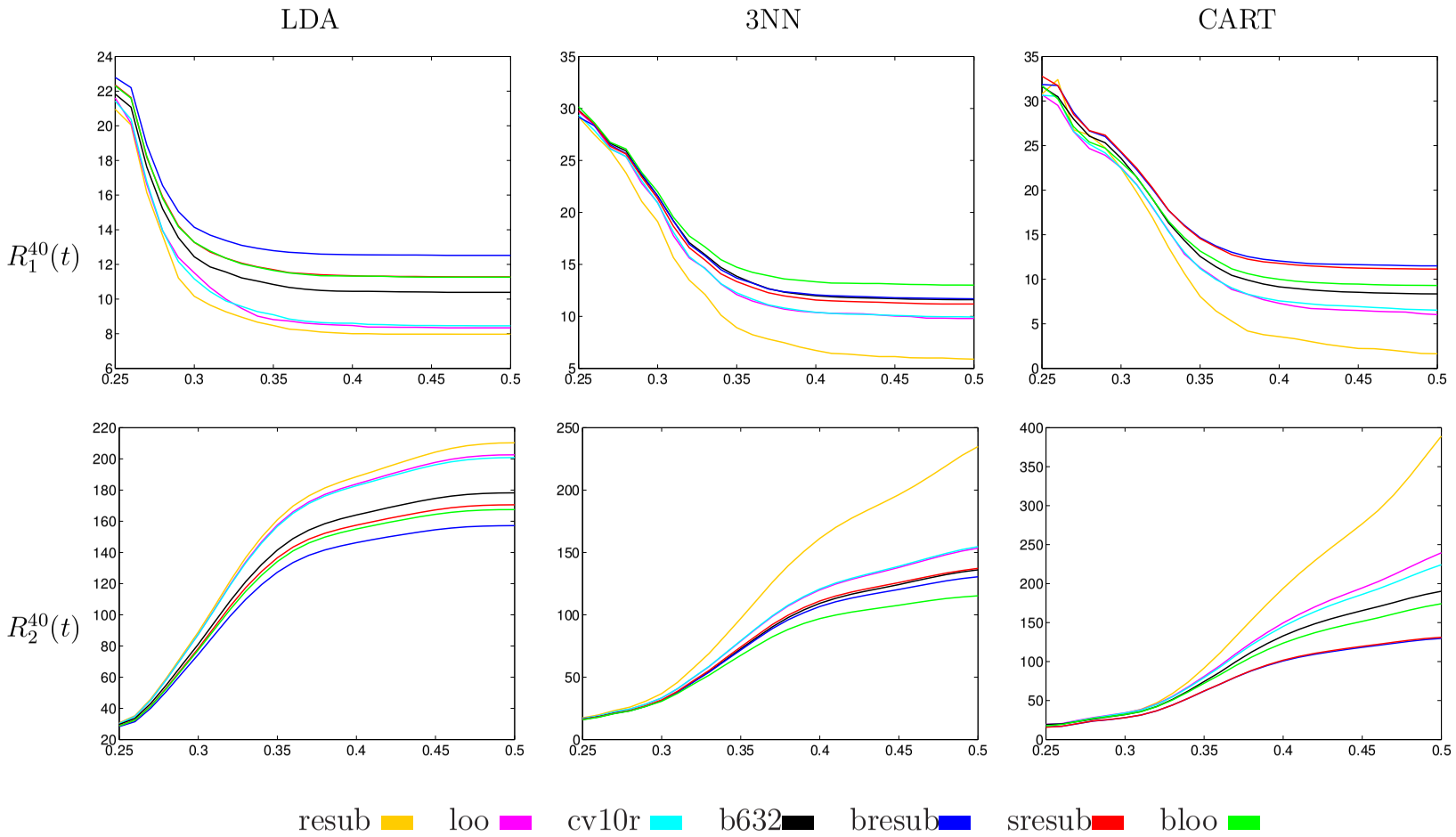


Fig. 2. Mean ranking statistics versus maximum true error threshold, for the synthetic data, in the equal-variance case and feature sets of size 3.

Table I. Two statistics for a few values of the maximum true error threshold t in Fig. 2, for the synthetic data, in the equal-variance case and feature sets of size 3: s_1 is the average true error for all feature sets having error less than t, while s_2 is the average number of feature sets having true error less than t.

	LDA		3NN		CART	
t	s_1	s_2	s_1	s_2	s_1	s_2
0.25	0.227	102.65	0.223	63.05	0.226	57.71
0.27	0.245	155.19	0.244	81.60	0.248	73.53
0.30	0.265	304.30	0.273	125.40	0.277	100.78
0.32	0.278	433.89	0.288	190.41	0.293	135.08
0.35	0.293	623.28	0.308	322.71	0.314	241.22
0.37	0.300	706.18	0.320	430.29	0.326	335.76
0.40	0.311	804.06	0.334	573.68	0.343	480.03
0.42	0.321	883.44	0.342	648.93	0.352	566.84
0.45	0.337	1026.25	0.356	756.25	0.367	690.53
0.47	0.345	1098.47	0.369	858.75	0.380	798.94
0.50	0.350	1140.00	0.391	1056.07	0.404	1023.02

by LDA with respect to $R_1^{40}(t)$ also holds for the ranking-comparison statistic $R_2^{40}(t)$. We remark that not only does bolstered resubstitution outperform .632 bootstrap in terms of feature discovery, it does so with much less computation time. Table II(a) provides typical computation times for the error estimators in this experiment.

For 3NN, Fig. 2 shows the very bad performance of resubstitution as measured by $R_1^{40}(t)$. This results from the extreme low bias of resubstitution for the 3NN classification rule (indeed, for the 1NN rule resubstitution always yields zero error). Nonetheless, bolstered resubstitution still performs as well as .632 bootstrap, which also suffers on account of the low bias of resubstitution, and outperforms all cross-validation estimators. The best performance is exhibited by bolstered leave-one-out, which is consistent with the comments of [29] regarding bolstering in the case of 3NN classification. Similar comments apply to $R_2^{40}(t)$, the only difference being that bolstered resubstitution slightly outperforms .632 bootstrap.

For CART, Fig. 2 shows that bolstered and semi-bolstered resubstitution significantly outperform .632 bootstrap, with bolstered-leave-one-out slightly outperforming .632 bootstrap, which itself outperforms the cross-validation estimators to about the same extent. Compared to the commonly employed cross-validation estimators, bolstered resubstitution finds on average five more top-40 feature sets among the top 40 based on error estimation, which means the discovery of substantially more features. Analogous relations among the estimators are found for $R_2^{40}(t)$.

2. Patient Data

We have conducted experiments based on patient data from a microarray-based classification study [33] that analyzes microarrays prepared with RNA from breast tumor samples from 295 patients. Using a previously established 70-gene prognosis profile [34], a prognosis signature based on gene-expression is proposed in [33] that correlates

Table II. Computation times for (a) the synthetic data, in the equal-variance case and feature sets of size 3, and (b) the patient data, for feature sets of size 3. The values are relative to the resubstitution timing.

	loo	cv10r	b632	bresub	sresub	bloo
LDA	90.30	306.27	465.44	7.40	6.30	97.15
3NN	0.94	7	48.09	12.27	10.39	12.08
CART	1224.50	3895.47	1931.31	103.93	97.85	1527.95

(a)

	loo	cv10r	b632	bresub	sresub	bloo
LDA	128.37	460.29	611.40	12.29	10.91	130.14
3NN	1	8.38	100.39	11.59	10.88	11.54
CART	1441.87	4584.47	4758.40	96.00	84.87	1512.67

(b)

well with patient survival data and other clinical measures. Of the 295 microarrays, 115 belong to the "good-prognosis" class and 180 belong to the "poor-prognosis" class.

Our experiments are set up in the following way. We use log-ratio gene expression values associated with the top 20 genes ranked according to [34]. The true error for each sample of size $n = 30$ is approximated by a holdout estimator, whereby the 265 sample points not drawn are used as the test set (a very good approximation to the true error, given the large test sample). It should be noted that the samples are not fully independent on account of overlap resulting from choosing the 30 samples from among the same 295 sample points; however, as discussed in [19], the samples are only weakly dependent.

The results corresponding to Fig. 2 are shown in Fig. 3 for the patient data experiments, with feature sets of size 3. The associated sample information and computation times are given in Tables III and II(b), respectively.

The trends regarding bolstering, bootstrap, and cross-validation observed in the Gaussian model are closely reflected in the patient data. We note that the performance measures are weaker in the patient data. This is because we are choosing feature sets from among the best correlated 20 genes, so that there are many good feature sets, and it is difficult to distinguish among them. Our goal was to see if bolstering would still prove superior to bootstrap and cross-validation in such a difficult scenario, and our results indicate it does so.

D. Conclusion

The results demonstrate, for the three classification rules and the datasets considered, that bolstering is superior to bootstrap, and bootstrap is better than cross-

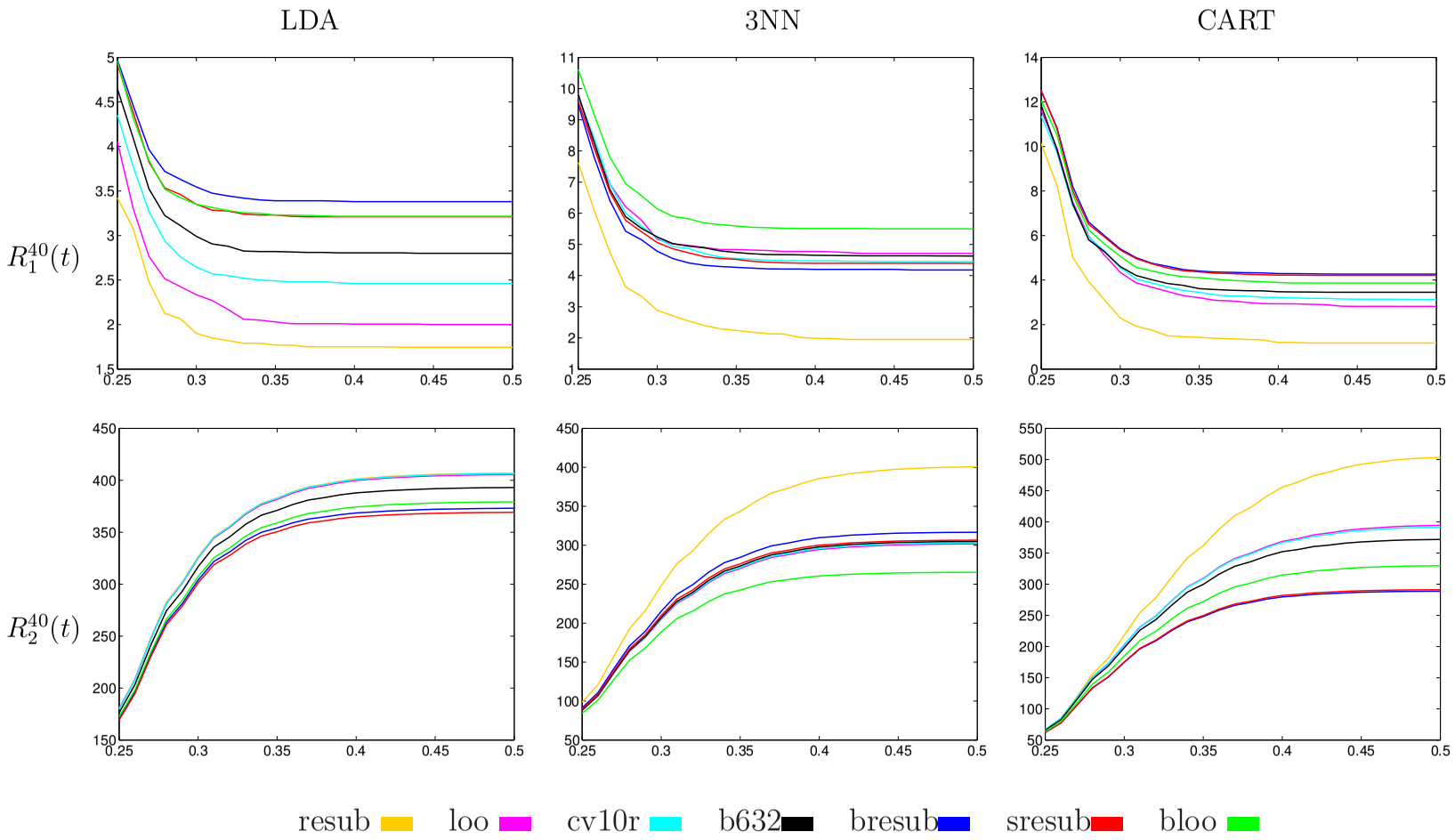


Fig. 3. Mean ranking statistics versus maximum true error threshold, for the patient data, for feature sets of size 3.

Table III. Two statistics for a few values of the maximum true error threshold t in Fig. 3, for the patient data, for feature sets of size 3: s_1 is the average true error for all feature sets having error less than t, while s_2 is the average number of feature sets having true error less than t.

	LDA		3NN		CART	
t	s_1	s_2	s_1	s_2	s_1	s_2
0.25	0.224	445.20	0.228	256.35	0.231	171.27
0.27	0.234	617.20	0.240	401.78	0.244	290.15
0.30	0.247	830.74	0.257	642.42	0.262	502.55
0.32	0.253	921.19	0.266	768.61	0.272	629.30
0.35	0.260	1019.06	0.277	923.54	0.286	810.46
0.37	0.265	1064.17	0.283	1003.04	0.294	917.19
0.40	0.269	1100.21	0.290	1071.38	0.303	1020.35
0.42	0.270	1113.42	0.293	1097.96	0.307	1063.04
0.45	0.272	1126.73	0.296	1123.37	0.312	1105.74
0.47	0.273	1131.89	0.297	1131.44	0.314	1120.66
0.50	0.274	1136.39	0.298	1137.56	0.316	1132.37

validation. Superior performance has been demonstrated with respect to two measures, one counting the number of the truly best feature sets appearing among the best feature sets discovered by the error estimator and the other computing the mean absolute error between the top ranks of the truly best feature sets and their ranks as given by the error estimator. A key issue is that bolstered error estimation is generally much faster than bootstrap and is therefore feasible for feature-set ranking when the number of feature sets is extremely large.

It should be recognized that the ranking results presented herein apply directly to only the specific classification rules and datasets presented and that more work is needed to determine the extent of the superiority of bolstering with regard to ranking. More importantly, it should be noticed that ranking performances using all of these error estimation techniques suffer greatly from the lack of data points. We shall further investigate the impact of error estimation on feature selection in next chapter.

CHAPTER IV

IMPACT OF ERROR ESTIMATION ON FEATURE SELECTION ALGORITHMS *

Given a large set of potential features, it is usually necessary to find a small subset with which to classify. The task of finding an optimal feature set is inherently combinatoric and therefore suboptimal algorithms are typically used to find feature sets. If feature selection is based directly on classification error, then a feature-selection algorithm must base its decision on error estimates. In this chapter we shall address the impact of error estimation on feature selection using two performance measures: comparison of the true error of the optimal feature set with the true error of the feature set found by a feature-selection algorithm, and the number of features among the truly optimal feature set that appear in the feature set found by the algorithm. The study considers seven error estimators applied to three standard suboptimal feature-selection algorithms and exhaustive search, and it considers three different feature-label model distributions. It draws two conclusions for the cases considered: (1) depending on the sample size and the classification rule, feature-selection algorithms can produce feature sets whose corresponding classifiers possess errors far in excess of the classifier corresponding to the optimal feature set; and (2) for small samples, differences in performances among the feature-selection algorithms are less significant than performance differences among the error estimators used to implement the algorithms. Moreover, keeping in mind that results depend on the particular classifier-distribution pair, for the error estimators considered in this study, bootstrap

*Reprinted with permission from "Impact of Error Estimation on Feature-selection Algorithms" by C. Sima, S. Attoor, U. Braga-Neto, J. Lowey, E. Suh and E.R. Dougherty, 2005, *Pattern Recognition*, Vol. 38, No. 12, pp2472-2482. Copyright 2005 by Elsevier.

and bolstered resubstitution usually outperform cross-validation, and bolstered resubstitution usually performs as well as or better than bootstrap.

A. Introduction

Given a large set of potential features for classification, it is necessary to find a small subset with which to classify. The problem is statistically inherent in classification because typically (but not universally), the true error of a designed classifier will fall with use of more features and, after some optimal number of features for a given sample size, begin to rise. For small samples the optimal number can be very small. The task of finding an optimal feature set is inherently combinatoric. According to a classical theorem, to be assured of finding the optimal feature set of a given size, all feature subsets of that size must be checked unless there is distributional knowledge that mitigates the search requirement, a mitigating condition not occurring in practice [26]. There are various methods of choosing feature sets, the intent being to choose a set of features that provides good classification. When there is a large number of potential features for classification, feature selection is problematic and the best method to use depends on the circumstances. Evaluation of methods is generally comparative and based on simulations [30][31].

If feature selection is based directly on classification error, and not on some auxilliary measure such as correlation, then an algorithm searching for a good feature set must base its decision on estimates of the error. If there is a large data set, then one can obtain good error estimates; however, if the sample is small, then error estimation is problematic and the performance of the feature-selection algorithm will be impacted by the performance of the error estimator. As will be demonstrated in this paper, the lack of optimality with feature selection can be impacted to a

greater extent by error estimation than by the choice of feature-selection algorithm, and performance of a particular feature-selection algorithm is affected by the choice of error-estimation rule.

The role of error estimation in the choice of feature sets for small samples has been addressed relative to the absolute ranking of feature sets in previous chapter and in [29]. In these studies, based on an exhaustive search, the classifiers corresponding to all feature sets of a given size were found, their true errors and their estimated errors based on various estimation rules were calculated, and the feature sets were ranked based on their true and estimated errors. The key issue was ranking order. It was seen that certain error-estimation rules gave better feature-set ranking, depending on the class-conditional distributions, classification rule, and sample size.

This chapter concerns the performance of feature-selection algorithms relative to their purpose of finding good feature sets – in particular, the impact of error estimation in this regard. Thus, we employ two measures of merit: (1) we will compare the true error of the optimal feature set with the true error of the feature set found by a feature-selection algorithm; and (2) we will see how many of the features among the truly optimal feature set appear in the feature set found by the algorithm. In all cases we will average the results over a large collection of samples, and we will categorize the results by feature-selection algorithm, error-estimation rule, classification rule, class-conditional distributions, and sample size. Owing to the large number of simulations and computations, the project has been carried out on a massively parallel Beowulf cluster.

To a great extent, this study has been motivated by the large number of papers in recent years dealing with phenotype classification based on expression microarrays. Perhaps the most salient characteristic of expression-based phenotype classification using microarray data is the vast number of potential features (genes) in comparison

to the small number of data points (microarrays), and the effect this disparity has on classifier design, error estimation, and feature selection [14]. Whereas there are typically thousands of genes on a microarray, laboratory costs and availability of patient tissue stringently limits the number of microarrays. Even though sample sizes are slowly growing as costs decline, availability of tissue will continue to limit sample sizes. Our simulation analyses reflect this limitation by considering sample sizes of 30, 50, 70, and 90.

B. Experimental Set-up

For simulation studies, we consider 3 models. **Model 1** is the same as the synthetic model in Chapter III, Section C.1: it is a 2-class Gaussian model, with the classes equally likely and the class-conditional densities being spherical unit variance Gaussians. The class means are located at δa and $-\delta a$, where $\delta > 0$ is a separation parameter and $a = (a_1, a_2, \ldots, a_n)$ is a parameter vector with $||a|| = 1$. It is well-known that the Bayes classifier is a hyperplane perpendicular to the axis joining the means, with Bayes error $\epsilon_{BAYES} = 1 - \Phi(\delta)$, where Φ is the standard normal cumulative distribution function. Since $\delta = \Phi^{-1}(1 - \epsilon_{BAYES})$, one can find δ for a prescribed Bayes error. In our experiments, we choose δ so that the Bayes error is 0.1. The parameter vector $a = (a_1, a_2, \ldots, a_n)$ is picked from a sigmoidal distribution in order to favor a few of the feature sets.

Model 2 is similar to Model 1, but instead of both covariance matrices for the class-conditional densities being $\boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix, we let them be $\sigma_1 \boldsymbol{I}$ and $\sigma_2 \boldsymbol{I}$ for class 1 and class 2, respectively, with $\sigma_1 \neq \sigma_2$. Since there is no closed-form formula for Bayes error in this model, we resort to Monte Carlo methods for computing the separation parameter δ for the desired Bayes errors. We let $\sigma_1 = 1$

and $\sigma_2 = 1.5$ in our experiments, and choose δ so that Bayes error using all the features equals 0.03 or 0.04.

Model 3 is also an equally likely 2-class Gaussian model, with means located at $(\delta, \delta, \ldots, \delta)$ and $(-\delta, -\delta, \ldots, -\delta)$. Here we employ a block-like covariance matrix structure

$$\mathbf{\Sigma} = \sigma^2 \begin{bmatrix} \mathbf{1} & \boldsymbol{\rho} & & & & & & & \\ \boldsymbol{\rho} & \mathbf{1} & & & & & & & \\ & & \mathbf{1} & & & & & & \\ & & & \mathbf{1} & & & & & \\ & & & & \mathbf{1} & \boldsymbol{\rho} & & & \\ & & & & \boldsymbol{\rho} & \mathbf{1} & & & \\ & & & & & & \mathbf{1} & & \\ & & & & & & & \mathbf{1} & \\ & & & & & & & & \mathbf{1} \end{bmatrix},$$

where all features are equally divided into G groups. The features from different groups are uncorrelated, and the features from the same group possess the same correlation ρ among each other. In our studies, we let both covariance matrices be $\mathbf{\Sigma}$ with $G = 5$, $\sigma = 1$ and $\rho = 0.5$, and choose δ so that the Bayes error is 0.05 (again by Monte Carlo methods). Notice if G equals the total number of features, then all features are uncorrelated and this model is similar to Model 1.

The experiments are set up in the following manner: 200 independent samples of size S with N features are generated, and we select the K features, using exhaustive search (exhst), sequential forwarding search (SFS) [35], sequential forwarding floating search (SFFS) [35], and the improved Branch-and-Bound search method (enhBB) [36].

The criterion function is the correct recognition rate, defined as 1 minus the estimated error. We consider three classification rules: linear discriminate analysis (LDA), 3-nearest neighbor (3NN), and classification and regression trees (CART). We apply 7 error estimation methods: true error (true), resubstitution (resub), leave-one-out (loo), 5-fold cross validation (cv5), .632 bootstrap (bstrap) , bolstered resubstitution (blstr) and semi-bolstered resubstitution (semib). By "true error" we mean the computed error for the designed classifier using the known underlying distribution of synthetic data, not the Bayes error, which the "true error" can achieve only when the designed classifier is optimal.

K features are found at the end of the feature search for each sample, and two performance measures, $T1$ and $T2$, are computed. $T1$ is the average true error over the 200 samples. Except in the case of Model 3, $T2$ is defined as the average, over the samples, of the number of common features when we compare the K features found by the feature selection to the K features found by exhaustive search and true error estimation using the same classifier.

The second measure for Model 3, denoted by $\widehat{T2}$, is computed differently. Here features within the same group are equivalent in the sense that, with all other features fixed, choosing any feature in the group should give the same classifying power. Furthermore, the groups are equivalent between each other in the sense that, choosing a feature from group i gives the same classifying power as choosing one from another group j, given the other features are fixed and not coming from group i or j. Thus, the key issue is the number of distinct groups represented by the K features found by feature selection. $\widehat{T2}$ is the average, over the 200 samples, of the number of represented groups.

We consider three (total features, selected features) pairs:$(N, K) = (20, 4), (20, 5)$, and $(25, 4)$. For each pair, we repeat the experiment for $S = 30, 50, 70$ and 90, and

Table IV. Experiments setup for impact of error estimation on feature selection study.

	Exp 1	**Exp 2**	**Exp 3**	**Exp 4**
Model	Model 1	Model 2	Model 2	Model 3
Bayes Error	0.10	0.03	0.04	0.05
Classification Rule	LDA, 3NN and CART			
Feature Selection Algorithm	exhst, SFS, SFFS and enhBB			
Error Estimation Method	true, resub, loo, cv5, bstrap, blstr, and semib			
Sample Size	30 , 50 , 70 and 90			
(N, K) pair	$(20, 4)$, $(20, 5)$ and $(25, 4)$			
Performance Measure	$T1$ and $T2$			$T1$ and $\widehat{T2}$

the performance measures $T1$ and $T2$ (or $\widehat{T2}$, in the case of Model 3) are computed. The experiments are summarized in Table IV .

C. Experimental Results

Selected results from the experiments are shown in Tables V, VI, VII, VIII, IX, and X. To use the tables, suppose we are interested in the performance for the Branch-and-Bound method for selecting 4 features out of 20, when the sample size is 30, under the LDA rule, in **Exp 2**. Then we look at Table VII and VIII, under column LDA/enhBB and row "size 30". There we find the results for each of the seven error estimation methods.

Table V. Selected performance measures results for **Exp 1**: T1 for N =20, K = 4, S = 30,50,70

T1 for N =20, K = 4, S = 30,50,70													
		LDA				**3NN**				**CART**			
		exhst	SFS	SFFS	enhBB	exhst	SFS	SFFS	enhBB	exhst	SFS	SFFS	enhBB
size 30	true	0.1667	0.1757	0.1737	0.2272	0.1867	0.1889	0.1882	0.2014	0.2421	0.2515	0.2503	0.2922
	resub	0.2494	0.2594	0.2551	0.2492	0.3094	0.3015	0.3018	0.2993	0.3741	0.3658	0.3664	0.3468
	loo	0.2452	0.2670	0.2569	0.2794	0.2770	0.2753	0.2755	0.2768	0.3404	0.3353	0.3359	0.3384
	cv5	0.2490	0.2636	0.2558	0.2933	0.2682	0.2754	0.2851	0.2841	0.3275	0.3354	0.3289	0.3541
	bstrap	0.2418	0.2517	0.2347	0.2935	0.2608	0.2599	0.2613	0.2586	0.3174	0.3218	0.3214	0.3195
	blstr	0.2161	0.2325	0.2184	0.2488	0.2570	0.2595	0.2672	0.2696	0.3023	0.3042	0.3036	0.3408
	semib	0.2214	0.2367	0.2243	0.2421	0.2635	0.2668	0.2654	0.2822	0.2958	0.3016	0.3013	0.3344
size 50	true	0.1683	0.1706	0.1699	0.1826	0.1954	0.1981	0.1972	0.2091	0.2390	0.2485	0.2466	0.2828
	resub	0.2261	0.2440	0.2339	0.2211	0.2969	0.2951	0.2949	0.2877	0.3746	0.3674	0.3668	0.3245
	loo	0.2241	0.2431	0.2320	0.2397	0.2645	0.2614	0.2641	0.2669	0.3222	0.3320	0.3315	0.3286
	cv5	0.2240	0.2459	0.2365	0.2588	0.2585	0.2732	0.2652	0.2754	0.3160	0.3218	0.3186	0.3453
	bstrap	0.2164	0.2292	0.2178	0.2267	0.2511	0.2532	0.2573	0.2527	0.3085	0.3059	0.3055	0.3108
	blstr	0.1956	0.2172	0.1957	0.1981	0.2532	0.2531	0.2500	0.2533	0.2802	0.2893	0.2888	0.3329
	semib	0.2019	0.2199	0.2038	0.2031	0.2522	0.2565	0.2587	0.2678	0.2836	0.2891	0.2890	0.3258
size 70	true	0.1654	0.1667	0.1660	0.1704	0.1929	0.1955	0.1945	0.2039	0.2321	0.2407	0.2394	0.2705
	resub	0.2033	0.2251	0.2127	0.1975	0.2756	0.2743	0.2738	0.2764	0.3564	0.3481	0.3490	0.3122
	loo	0.2018	0.2263	0.2122	0.2103	0.2447	0.2487	0.2500	0.2552	0.3127	0.3261	0.3252	0.3120
	cv5	0.2040	0.2226	0.2099	0.2286	0.2467	0.2546	0.2519	0.2631	0.3037	0.3049	0.3087	0.3374
	bstrap	0.1941	0.2149	0.1963	0.1998	0.2379	0.2416	0.2415	0.2394	0.2872	0.2889	0.2905	0.2956
	blstr	0.1806	0.2081	0.1826	0.1795	0.2340	0.2326	0.2357	0.2403	0.2657	0.2724	0.2728	0.3232
	semib	0.1868	0.2115	0.1900	0.1866	0.2342	0.2350	0.2400	0.2543	0.2669	0.2707	0.2736	0.3150

Table VI. Selected performance measures results for **Exp 1**: T2 for N =20, K = 4, S = 30,50,70.

T2 for N =20, K = 4, S = 30,50,70													
		LDA				**3NN**				**CART**			
		exhst	SFS	SFFS	enhBB	exhst	SFS	SFFS	enhBB	exhst	SFS	SFFS	enhBB
size 30	true	4.0000	3.0900	3.2900	1.9550	4.0000	3.4650	3.5300	3.0800	4.0000	1.9400	1.9100	1.3700
	resub	1.5700	1.4400	1.4850	1.6700	1.4700	1.5500	1.5350	1.5950	0.9100	0.8500	0.8250	1.1300
	loo	1.6650	1.3650	1.4900	1.3050	1.8300	1.8400	1.8250	1.8450	0.9850	1.0350	1.0300	1.0200
	cv5	1.6000	1.3450	1.5100	1.0800	1.9050	1.8200	1.7450	1.7150	1.0650	1.1450	1.1100	0.8800
	bstrap	1.7850	1.5650	1.8650	1.1000	2.0700	2.0850	2.0500	2.0900	1.1550	1.1900	1.1450	1.1600
	blstr	2.1000	1.7850	2.0900	1.6750	2.1000	2.0500	2.0200	1.8950	1.2050	1.2700	1.2050	1.0950
	semib	2.0350	1.7250	1.9550	1.7400	2.0450	1.9600	1.9450	1.7700	1.3450	1.2500	1.2600	0.9600
size 50	true	4.0000	3.2250	3.3500	2.6250	4.0000	3.3850	3.4850	3.0150	4.0000	2.2650	2.2800	1.6150
	resub	1.7750	1.4700	1.6300	1.8450	1.6600	1.6800	1.6550	1.7000	0.9300	0.9500	0.9600	1.3950
	loo	1.7850	1.4650	1.6800	1.5950	1.9800	2.0700	2.0350	1.9750	1.2650	1.1550	1.1450	1.2500
	cv5	1.7950	1.3850	1.5950	1.2700	2.1800	1.9350	2.0150	1.9400	1.2750	1.1650	1.2550	1.0250
	bstrap	1.9200	1.6600	1.8900	1.7750	2.2900	2.2650	2.1400	2.2300	1.3800	1.4250	1.4100	1.3400
	blstr	2.3000	1.7950	2.2850	2.2300	2.2650	2.1850	2.2900	2.2200	1.6350	1.4950	1.4150	1.1650
	semib	2.1700	1.7700	2.0650	2.1850	2.2450	2.2000	2.1650	2.0450	1.6700	1.4750	1.5650	1.1500
size 70	true	4.0000	3.2800	3.4400	2.8450	4.0000	3.3250	3.3350	3.0350	4.0000	2.5950	2.6150	1.9500
	resub	2.0550	1.6350	1.8500	2.1400	1.8550	1.8050	1.8100	1.7750	1.1200	1.0950	1.1000	1.5350
	loo	2.0100	1.5750	1.8300	1.9350	2.2850	2.1100	2.1100	2.0450	1.4900	1.2150	1.2150	1.5250
	cv5	1.9450	1.5850	1.9400	1.6250	2.2250	2.0000	2.1000	1.9600	1.5000	1.5200	1.3700	1.2050
	bstrap	2.1750	1.7350	2.1400	2.1050	2.3900	2.3250	2.3450	2.3000	1.8650	1.7850	1.7200	1.6300
	blstr	2.4800	1.8550	2.4450	2.5400	2.4300	2.4300	2.3550	2.2700	1.9950	1.8300	1.8400	1.3150
	semib	2.3100	1.7800	2.2150	2.4200	2.4500	2.4100	2.2850	2.0500	2.0300	1.9200	1.8950	1.3850

Table VII. Selected performance measures results for **Exp 2**: T1 for N =20, K = 4, S = 30,50,70.

T1 for N =20, K = 4, S = 30,50,70													
		LDA				**3NN**				**CART**			
		exhst	SFS	SFFS	enhBB	exhst	SFS	SFFS	enhBB	exhst	SFS	SFFS	enhBB
size 30	true	0.1440	0.1508	0.1494	0.2054	0.1525	0.1559	0.1549	0.1759	0.1848	0.2089	0.2046	0.2518
	resub	0.2256	0.2387	0.2345	0.2262	0.2620	0.2667	0.2678	0.2543	0.3110	0.3101	0.3100	0.3047
	loo	0.2224	0.2403	0.2294	0.2534	0.2301	0.2351	0.2364	0.2444	0.2923	0.2888	0.2898	0.2908
	cv5	0.2289	0.2367	0.2304	0.2755	0.2298	0.2314	0.2375	0.2524	0.2823	0.2875	0.2846	0.2967
	bstrap	0.2190	0.2235	0.2129	0.2632	0.2216	0.2192	0.2201	0.2226	0.2731	0.2810	0.2779	0.2799
	blstr	0.1923	0.2053	0.1918	0.2236	0.2140	0.2241	0.2270	0.2347	0.2486	0.2600	0.2626	0.2857
	semib	0.1955	0.2151	0.2016	0.2210	0.2195	0.2228	0.2230	0.2451	0.2474	0.2658	0.2621	0.2920
size 50	true	0.1407	0.1431	0.1425	0.1540	0.1484	0.1515	0.1504	0.1660	0.1749	0.1904	0.1878	0.2289
	resub	0.1896	0.2069	0.1981	0.1849	0.2353	0.2326	0.2307	0.2367	0.3080	0.2966	0.2963	0.2648
	loo	0.1865	0.2083	0.1972	0.2034	0.2063	0.2147	0.2152	0.2222	0.2627	0.2712	0.2719	0.2672
	cv5	0.1907	0.2126	0.1991	0.2269	0.2026	0.2106	0.2124	0.2314	0.2555	0.2595	0.2632	0.2792
	bstrap	0.1802	0.1928	0.1825	0.1970	0.1956	0.1995	0.2026	0.2027	0.2434	0.2478	0.2500	0.2581
	blstr	0.1625	0.1830	0.1638	0.1663	0.1929	0.1970	0.1990	0.2103	0.2223	0.2295	0.2308	0.2722
	semib	0.1693	0.1893	0.1713	0.1707	0.1985	0.2024	0.2057	0.2193	0.2230	0.2296	0.2289	0.2699
size 70	true	0.1388	0.1404	0.1397	0.1437	0.1457	0.1482	0.1471	0.1590	0.1710	0.1832	0.1815	0.2156
	resub	0.1754	0.1992	0.1892	0.1752	0.2152	0.2210	0.2222	0.2140	0.2849	0.2792	0.2802	0.2494
	loo	0.1756	0.1951	0.1824	0.1847	0.1960	0.2007	0.2017	0.2052	0.2413	0.2463	0.2476	0.2511
	cv5	0.1750	0.1970	0.1881	0.1994	0.1961	0.1964	0.2001	0.2130	0.2333	0.2427	0.2444	0.2736
	bstrap	0.1680	0.1864	0.1692	0.1693	0.1847	0.1874	0.1891	0.1858	0.2271	0.2319	0.2310	0.2354
	blstr	0.1536	0.1769	0.1531	0.1495	0.1835	0.1854	0.1903	0.1947	0.2060	0.2161	0.2136	0.2632
	semib	0.1583	0.1812	0.1591	0.1549	0.1843	0.1880	0.1865	0.1999	0.2074	0.2151	0.2169	0.2573

Table VIII. Selected performance measures results for **Exp 2**: T2 for N =20, K = 4, S = 30,50,70.

T2 for N =20, K = 4, S = 30,50,70													
		LDA				**3NN**				**CART**			
		exhst	SFS	SFFS	enhBB	exhst	SFS	SFFS	enhBB	exhst	SFS	SFFS	enhBB
size 30	true	4.0000	3.1500	3.2650	1.9550	4.0000	3.3300	3.4400	2.7800	4.0000	1.7050	1.8350	1.1750
	resub	1.5300	1.3600	1.4800	1.5650	1.4550	1.3400	1.3500	1.5450	0.8500	0.8800	0.9000	1.0050
	loo	1.6150	1.3000	1.4700	1.2400	1.7600	1.7000	1.6850	1.6200	0.9000	0.8750	0.8800	0.9650
	cv5	1.5100	1.3750	1.5100	0.9700	1.7700	1.7100	1.6600	1.5100	0.9350	0.8800	0.8950	0.9350
	bstrap	1.7150	1.5500	1.7650	1.1500	1.9350	1.9100	1.9700	1.9800	1.0200	0.8850	0.9200	0.9000
	blstr	2.1550	1.7900	2.1250	1.7300	1.9350	1.7950	1.7900	1.7150	1.2650	1.0200	1.0400	1.0300
	semib	2.0550	1.6250	1.9650	1.7250	1.9300	1.8400	1.8550	1.6650	1.2400	0.9950	1.0350	0.9750
size 50	true	4.0000	3.2350	3.3200	2.7300	4.0000	3.3000	3.3600	2.8600	4.0000	2.1950	2.2850	1.6050
	resub	1.9500	1.6100	1.7700	2.0250	1.6650	1.6550	1.6650	1.6050	1.0150	1.0000	1.0050	1.3450
	loo	1.9950	1.5500	1.7750	1.7700	1.9900	1.7750	1.7650	1.7000	1.2600	1.0800	1.0750	1.2300
	cv5	1.9350	1.5350	1.7450	1.3600	2.0700	1.9300	1.8850	1.6500	1.3100	1.2000	1.2150	1.0100
	bstrap	2.1050	1.7750	1.9850	1.8900	2.1550	2.1150	2.1050	2.0500	1.3900	1.3750	1.2800	1.2400
	blstr	2.3950	1.9650	2.3500	2.4350	2.1500	2.2150	2.1500	2.0000	1.5250	1.5250	1.5000	1.2300
	semib	2.2700	1.8450	2.2500	2.3150	2.1700	2.0150	1.9150	1.8350	1.5300	1.4300	1.4100	1.1600
size 70	true	4.0000	3.3150	3.4350	2.9450	4.0000	3.2500	3.3550	2.8150	4.0000	2.3850	2.4500	1.7350
	resub	2.0850	1.6900	1.8650	2.0650	1.7200	1.7050	1.6700	1.8350	1.0350	1.1450	1.1200	1.4000
	loo	2.0700	1.7750	1.9900	1.9650	2.0500	2.0400	2.0150	2.0250	1.4350	1.3700	1.3400	1.4200
	cv5	2.0800	1.6850	1.8800	1.6500	2.0650	2.0400	2.0200	1.9100	1.4700	1.4050	1.3850	1.0600
	bstrap	2.2250	1.8700	2.1900	2.2400	2.2600	2.2550	2.2500	2.2800	1.5400	1.5650	1.4650	1.5650
	blstr	2.5400	2.0300	2.5800	2.7000	2.3250	2.2900	2.1800	2.1800	1.9250	1.7300	1.7100	1.2700
	semib	2.4350	2.0300	2.4600	2.5150	2.3400	2.2200	2.2400	2.0100	1.8050	1.7200	1.7300	1.3250

Table IX. Selected performance measures results for **Exp 4**: T1 for N =20, K = 4, S = 30,50,70.

T1 for N =20, K = 4, S = 30,50,70													
		LDA				**3NN**				**CART**			
		exhst	SFS	SFFS	enhBB	exhst	SFS	SFFS	enhBB	exhst	SFS	SFFS	enhBB
size 30	true	0.1211	0.1289	0.1266	0.1590	0.1322	0.1384	0.1376	0.1377	0.1987	0.2190	0.2160	0.2339
	resub	0.1740	0.1741	0.1729	0.1648	0.2009	0.1900	0.1897	0.1850	0.2638	0.2695	0.2692	0.2594
	loo	0.1611	0.1731	0.1722	0.1825	0.1719	0.1735	0.1728	0.1781	0.2473	0.2583	0.2583	0.2574
	cv5	0.1643	0.1708	0.1685	0.1717	0.1731	0.1742	0.1776	0.1782	0.2487	0.2551	0.2551	0.2542
	bstrap	0.1513	0.1656	0.1508	0.1723	0.1621	0.1671	0.1685	0.1727	0.2470	0.2525	0.2516	0.2552
	blstr	0.1458	0.1634	0.1483	0.1691	0.1669	0.1737	0.1721	0.1732	0.2390	0.2465	0.2461	0.2542
	semib	0.1470	0.1628	0.1518	0.1656	0.1667	0.1737	0.1727	0.1712	0.2393	0.2470	0.2462	0.2515
size 50	true	0.1205	0.1260	0.1249	0.1297	0.1335	0.1397	0.1381	0.1357	0.1872	0.2061	0.2044	0.2191
	resub	0.1430	0.1553	0.1514	0.1439	0.1764	0.1786	0.1778	0.1767	0.2528	0.2503	0.2511	0.2364
	loo	0.1404	0.1560	0.1522	0.1559	0.1632	0.1680	0.1674	0.1681	0.2314	0.2392	0.2394	0.2388
	cv5	0.1409	0.1549	0.1493	0.1604	0.1630	0.1680	0.1654	0.1709	0.2329	0.2352	0.2371	0.2433
	bstrap	0.1339	0.1491	0.1399	0.1555	0.1574	0.1616	0.1618	0.1650	0.2292	0.2333	0.2340	0.2403
	blstr	0.1391	0.1492	0.1404	0.1456	0.1611	0.1649	0.1636	0.1638	0.2227	0.2258	0.2271	0.2390
	semib	0.1386	0.1521	0.1416	0.1453	0.1622	0.1647	0.1650	0.1687	0.2223	0.2281	0.2262	0.2382
size 70	true	0.1199	0.1240	0.1236	0.1219	0.1340	0.1407	0.1395	0.1358	0.1809	0.1976	0.1966	0.2114
	resub	0.1370	0.1446	0.1440	0.1355	0.1691	0.1755	0.1752	0.1741	0.2415	0.2402	0.2408	0.2298
	loo	0.1347	0.1454	0.1413	0.1442	0.1589	0.1680	0.1668	0.1651	0.2228	0.2302	0.2301	0.2357
	cv5	0.1381	0.1454	0.1431	0.1476	0.1593	0.1680	0.1644	0.1655	0.2240	0.2312	0.2299	0.2376
	bstrap	0.1346	0.1434	0.1353	0.1455	0.1552	0.1607	0.1610	0.1632	0.2192	0.2227	0.2229	0.2280
	blstr	0.1325	0.1432	0.1354	0.1407	0.1563	0.1603	0.1609	0.1604	0.2115	0.2159	0.2160	0.2346
	semib	0.1354	0.1431	0.1389	0.1414	0.1590	0.1619	0.1625	0.1652	0.2101	0.2179	0.2152	0.2355

Table X. Selected performance measures results for **Exp 4**: $\widehat{T2}$ for N =20, K = 4, S = 30,50,70.

$\widehat{T2}$ **for N =20, K = 4, S = 30,50,70**													
		LDA				**3NN**				**CART**			
		exhst	SFS	SFFS	enhBB	exhst	SFS	SFFS	enhBB	exhst	SFS	SFFS	enhBB
size 30	true	4.0000	3.9950	3.9900	3.3200	4.0000	3.9950	4.0000	3.9550	3.7350	3.3700	3.4300	3.1750
	resub	2.9500	3.2250	2.9950	3.2100	2.7000	2.9800	2.9900	3.0400	2.1100	2.4600	2.4400	2.6150
	loo	3.3500	3.3200	3.1600	2.9550	3.2850	3.1750	3.1750	3.1200	2.8700	2.6300	2.6250	2.4450
	cv5	3.2650	3.3350	3.2400	3.1650	3.3000	3.2050	3.1450	3.1650	3.1050	2.9950	3.0000	3.1300
	bstrap	3.6600	3.5550	3.6200	3.1350	3.4700	3.3350	3.3050	3.1850	3.1100	3.1650	3.1150	3.1850
	blstr	3.4250	3.4550	3.3950	3.2250	3.3350	3.1600	3.2450	3.2250	3.1300	3.0900	3.0500	3.1500
	semib	3.4300	3.4300	3.2950	3.2850	3.4000	3.1500	3.2200	3.2750	3.1900	3.0250	3.0000	3.0900
size 50	true	4.0000	4.0000	4.0000	3.8800	4.0000	4.0000	4.0000	3.9800	3.9150	3.5650	3.6350	3.3100
	resub	3.5950	3.4450	3.4200	3.6000	3.3250	3.2100	3.2300	3.2100	2.5850	2.8550	2.8300	3.1100
	loo	3.6750	3.5150	3.4300	3.2800	3.5100	3.3400	3.3350	3.3050	3.3000	3.0350	3.0250	2.8450
	cv5	3.6250	3.5100	3.4750	3.2000	3.5150	3.3500	3.4250	3.2950	3.2950	3.1900	3.1850	3.1300
	bstrap	3.8350	3.6550	3.7000	3.2800	3.6450	3.5000	3.4900	3.3500	3.3350	3.3050	3.2700	3.1450
	blstr	3.5700	3.6550	3.5550	3.4050	3.4700	3.3950	3.4650	3.4300	3.1900	3.2200	3.2000	3.1700
	semib	3.6150	3.5350	3.5400	3.4450	3.4900	3.4300	3.3900	3.2700	3.2100	3.1050	3.1700	3.1700
size 70	true	4.0000	4.0000	4.0000	3.9700	4.0000	4.0000	4.0000	3.9900	3.9650	3.7650	3.7250	3.4800
	resub	3.7250	3.6250	3.5000	3.7550	3.5650	3.2900	3.3150	3.3250	2.8900	3.0300	3.0300	3.1650
	loo	3.7800	3.6100	3.6100	3.5100	3.6150	3.3300	3.3850	3.3800	3.3500	3.1800	3.1800	3.0700
	cv5	3.7350	3.5800	3.5150	3.4050	3.6700	3.3850	3.4700	3.4000	3.4450	3.2900	3.2800	3.1400
	bstrap	3.8450	3.7150	3.7700	3.4700	3.7500	3.5800	3.6150	3.4150	3.4500	3.3800	3.3600	3.2750
	blstr	3.7350	3.7200	3.6850	3.4550	3.6800	3.5700	3.6050	3.5250	3.3650	3.2100	3.2600	3.1050
	semib	3.7150	3.6800	3.5650	3.4350	3.5550	3.5200	3.5450	3.3850	3.4000	3.2500	3.2700	3.1750

1. Significance of Error Estimation Relative to Feature Selection

The most important conclusion we draw from the experiments is that, for small samples, differences in performances among the feature selection algorithms are much less significant than the effects of error estimation. Except for several cases in which branch and bound performs very badly (see Section C.4 in this chapter), performances across different feature-selection algorithms are mostly comparable, including exhaustive search. We note three points in this regard.

SFFS generally outperforms SFS, which outperforms enhBB when doing feature selection using the true error, but this is not necessarily the case when using error estimation. For instance, when using 3NN, SFFS outperforms enhBB when true error is used; however, if resubstitution is used, enhBB outperforms SFFS, and if cross-validation or bootstrap are used, the SFFS and enhBB perform essentially the same.

For LDA, SFFS and SFS perform almost equivalently to exhaustive search when the true error is used, but they degrade relative to exhaustive search when error estimation is employed, SFS doing worse than SFFS, and the latter degrading little in relation to exhaustive search when using bootstrap or bolstering.

The choice of error estimator for feature selection can make more of a difference than choice of feature selection algorithm in terms of the true error of the designed classifier. Consider the following observations. Referring to Table V and VI (Exp 1), for LDA and $S = 50$, if leave-one-out is used along with a full search, then the error of the designed classifier is 0.2241, but if bolstered resubstitution is used, then the worst result occurs with SFS, and this classifier has error 0.2172, better than an exhaustive search with leave-one-out (and better than an exhaustive search with 5-fold cross-validation). Similar phenomena occur throughout the results. In

particular, there are many cases where bolstered resubstitution and bootstrap yield better feature sets using SFFS than the feature sets obtained by cross-validation (both loo and cv5) using an exhaustive search. For instance, for all cases in Tables V, VI, VII and VIII, bolstered resubstitution and bootstrap yield better $T1$ values using SFFS than cross-validation using an exhaustive search, with bolstered resubstitution outperforming bootstrap for LDA and CART in all cases in both tables. Moreover, bolstered resubstitution yields better $T2$ values using SFFS than cross-validation using an exhaustive search for all cases in Tables V, VI, VII and VIII.

2. Some General Trends

Besides observations regarding the prominence of error-estimation choices relative to feature-selection choices, some general trends can be discerned. As would be expected, throughout the experimental results larger samples yield better performances of $T1$ and $T2$ ($\widehat{T2}$ in Exp 4). No matter which error estimation procedure is adopted, the results are much worse than using the true error for all feature selection methods, both for $T1$ and $T2$ ($\widehat{T2}$). The feature-selection algorithms perform better for the blocked covariance structure of Model 3 (Exp 4) than for Models 1 and 2. All feature-selection algorithms perform the worst for CART, and this is especially true fore small sample size ($S = 30$), no matter the error estimation method, including using the true underlying distribution. This suggests that one should avoid feature selection for complicated classification rules when only small samples are available.

3. Comparison of Error Estimation Methods

Consistent with the results reported in straight feature ranking (in Chapter III), for feature selection, bootstrap and bolstered resubstitution usually outperform cross-validation, with bolstering usually performing as well as or better than bootstrap;

however, we must take care and consider individual results, because, specific results, and sometimes even trends in the results, must be examined for each particular classification-distribution combination. Considering Exp 1 in some detail, we note several phenomena.

For LDA and $S = 50$, with exhaustive search, SFS, or SFFS, resubstitution and cross-validation estimators perform about the same with respect to error. Bootstrap does better and bolstering does even better. However, for branch and bound, resubstitution and bootstrap both outperform cross-validation and are comparable. Moreover, the advantage of bolstering over bootstrap is even greater. Most of these observations are mirrored in the T2 statistic.

Now look at LDA and $S = 30$. The overall situation is different. For exhaustive search, resubstitution, cross-validation, and bootstrap all perform about the same, with bolstering substantially better. For SFS, there is a slight ordering, cross-validation being the worst, resubstitution being slightly better, bootstrap being still slightly better, and bolstering having a more substantial advantage over bootstrap. For SFFS, the results are similar to $S = 50$. For enhBB, there is generally worse performance, especially for the computationally intensive cv5 and bootstrap, with loo being slightly better. The striking difference is that resubstitution and bolstering perform about the same, with both being much better than bootstrap.

For 3NN and all sample sizes, there appears to be a more consistent trend based on both $T1$ and $T2$ than for LDA: bootstrap and bolstering perform about the same and are better than cross-validation, and resubstitution is by far the worst.

For CART and all sample sizes, we again witness the main trend from best to worst: bolstering, bootstrap, cross-validation, and finally resubstitution, which is far worse than any of the others.

4. Remarks on the Performance of Branch-and-Bound

We have seen that the branch-and-bound algorithm can perform much worse than SFS and SFFS for LDA with very small samples. To appreciate the source of this problem, we refer to a typical branch-and-bound search in Fig. 4. The $N = 20$ features are labeled $0, 1, ..., 19$. Marked at each node explored is the label number of the feature discarded at that point, along with the criterion function value evaluated [28][36]. Notice that the criterion function value at node 17 is higher than that at node 4. Thus, the search stops after merely one branch exploration. This gives us the best features as $0, 1, 4$, and 7, whereas the best features found by exhaustive search are $0, 1, 3$, and 15. The monotonicity assumption for branch and bound is severely violated here. The poor performance of enhBB is largely due to designing a classifier on a very small sample. At level 1 in Fig. 4, a 19-dimensional LDA classifier must be designed with only 30 data points, and the designed LDA classifier is likely to possess a large error.

D. Conclusion

Feature selection is unavoidable when there is a large number of features from which to choose. Our experiments indicate SFS and SFFS (and even branch and bound) can perform close to optimal (full search with true error) when the true error is employed in feature selection, but in practice knowledge of the true error is impossible. With large samples, most error estimation procedures work quite well so that one has good estimates of the true error; however, this is not the case with small samples, as are common in situations where data are expensive or difficult to obtain owing to a limitation on their availability, as is often the case with patient samples. Depending on the sample size and the classification rule, in particular its complexity,

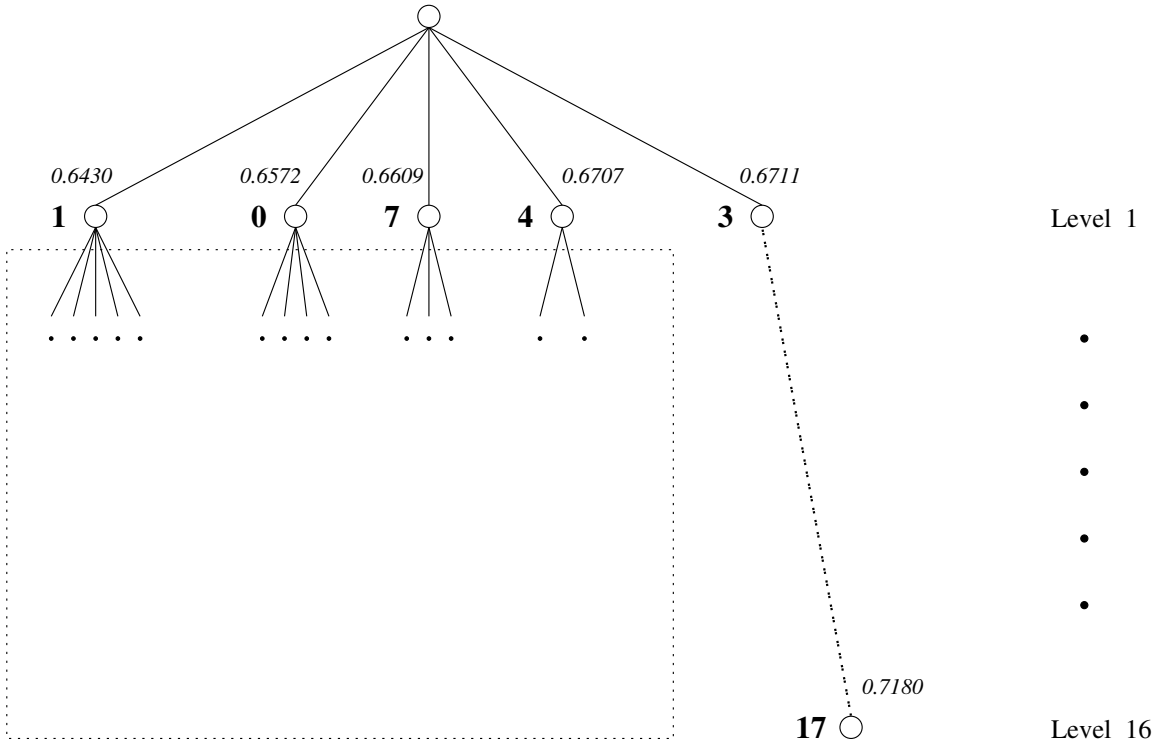


Fig. 4. A typical Branch-and-Bound tree searching path for $N = 20$, $K = 4$, $S = 30$ using "true error" estimation for LDA rule. (Taken from a simulation from Exp 1)

feature-selection algorithms can produce feature sets whose corresponding classifiers possess errors far in excess of the classifier corresponding to the optimal feature set. Moreover, and most importantly in application since one may have no alternative to a small sample, our experiments show that, for small samples, differences in performances among the feature selection algorithms are less significant than performance differences among the error estimators used to implement the feature-selection algorithm. Keeping in mind that specific results, and sometimes even trends in the results, depend on the particular classifier-distribution pair, for the error estimators considered in this study, bootstrap and bolstered resubstitution usually outperform cross-validation. Moreover, bolstered resubstitution usually performs as well as or better than bootstrap, and with much less computation time.

CHAPTER V

FEATURE SELECTION: A REGRESSION STUDY

High-throughput technologies for rapid measurement of vast numbers of biological variables offer the potential for highly discriminatory diagnosis and prognosis; however, high dimensionality together with small samples creates the need for feature selection, while at the same time making feature-selection algorithms less reliable. In previous chapters, we have shown error estimation could be problematic for feature ranking and feature selection in such small sample settings; combined with the suboptimality of the feature selection algorithm itself, one is naturally faced with the following twin questions. (1) Is it likely that feature selection will yield a feature set whose error is close to that of the optimal feature set? (2) If one cannot find a good feature set, should it be concluded that good feature sets do not exist?

We take a regression approach to the two questions. The first question is addressed via regression of the selected-feature-set error on the optimal-feature-set error: does the error of the optimal feature set predict the error of the selected feature set? The second question is addressed via regression of the optimal-feature-set error on the selected-feature-set error: does the error of the selected feature set predict the error of the optimal feature set? Three classification rules (linear discriminant analysis, linear support vector machine, and 3-nearest-neighbor classification) are considered in conjunction with two feature-label models and patient data from a study concerning survival prognosis for breast cancer. With respect to the two focus questions, there is similarity across all experiments: (1) it is unlikely that feature selection will yield a feature set whose error is close to that of the optimal feature set; and (2) the inability to find a good feature set should not lead to the conclusion that good feature sets do not exist. The lack of regression of the best-feature-set error with respect to

the selected-feature-set error for the patient data is striking, with the regression line being almost horizontal.

A. Introduction

High-throughput technologies for rapid measurement of vast numbers of biological variables offer the potential to model complex biological processes. In translational genomics, phenotype classification via gene expression portends highly discriminatory molecular-based diagnosis and prognosis. Yet one must recognize the obstacles inherent in dealing with extremely large numbers of interacting variables. So long as sample sizes remain small, large data sets may have the perverse effect of limiting the ability to design good classifiers and to obtain satisfactory error estimates [14]. In particular, this dimensionality problem creates the need for feature selection, while at the same time making feature-selection algorithms less reliable. One is faced with twin questions. (1) Is it likely that feature selection will yield a feature set whose error is close to that of the optimal feature set? (2) If one cannot find a good feature set, should it be concluded that good feature sets do not exist?

Feature selection is required when the number of features is large with respect to the sample size because the use of a large number of features can result in overfitting the data: the designed classifier performs well on the sample data but not on the feature-label distribution from which the data have been drawn. A feature-selection algorithm is part of the classification rule. This is why feature selection must be included when using cross-validation error estimation. Feature selection yields classifier constraint, not a reduction in the dimensionality of the feature space relative to design. For instance, if there are D features available for linear discriminant analysis (LDA), when used directly, then the classifier family consists of all hyperplanes

in D-dimensional space, but if a feature-selection algorithm reduces the number of variables to $d < D$ prior to application of LDA, then the classifier family consists of all hyperplanes in D-dimensional space confined to d-dimensional subspaces. The dimensionality of the classification rule has not been reduced, but the new classification rule (feature selection plus LDA) is constrained. The issue is whether it is sufficiently constrained. Given 20,000 gene-expression levels as features, the new rule has significant potential for overfitting.

B. Systems and Methods

We take a regression approach to the two questions posed at the outset. The question of whether feature selection yields a close-to-optimal feature set is addressed by regressing the error of the selected feature set on the error of the optimal feature set – does the error of the optimal feature set predict the error of the selected feature set? The question of whether the inability to find a good feature set implies that good feature sets do not exist is addressed by regressing the error of the optimal feature set on the error of the selected feature set – does the error of the selected feature set predict the error of the optimal feature set?

The regression analysis, and therefore the answers to the questions posed, will depend on the population from which the data are drawn, the classification rule, and the feature selection algorithm employed. We consider three classification rules: linear discriminant analysis (LDA), linear support vector machine (SVM), and 3-nearest-neighbor classification (3NN). We employ the sequential floating forward search (SFFS) feature-selection algorithm [35], whose performance has been extensively studied and shown to provide good results in relation to competing algorithms [31, 30]. Error estimation is critical within the SFFS algorithm and since, depending

on the classification rule, bolstered and semi-bolstered resubstitution [22] have been shown to perform well within the algorithm [37], we employ these with SFFS. Finally, there is the issue of modelling the feature-label distribution. Here we are motivated by two factors. First, since in general one must consider all feature sets of a given size to determine the best feature set of that size, we choose models in which the optimal feature sets are known from the model. Second, since in applications we do not know the feature-label distribution, we will assume a random feature-label distribution governed by a random parameter. The exact way in which we carry out the experiments with synthetic data, as well as how we utilize real patient data, will be discussed subsequently.

Before describing the details of our experiments, we explain how, given the feature-label distribution, we rank feature sets for a classification rule. Given a set G of features corresponding to the feature-label distribution $\mathbf{F}_G$, and the desire to select the best feature set of size d, then an obvious choice is that subset $H \subset G$ such that H possesses d features and the Bayes classifier for the distribution $\mathbf{F}_H$, the marginal distribution of $\mathbf{F}_G$ corresponding to H, has minimal error among all Bayes classifiers corresponding to subsets of G possessing d features. In this case we say that H has minimal Bayes error. This approach is reasonable because we are interested in finding good feature sets, regardless of the classification rule employed.

One might argue that there is a problem with this approach if the classification rule is not consistent, meaning that, given a feature set H and a sample of size n, the expected error of the designed classifier does not converge in mean to the Bayes error for the feature set as $n \to \infty$. To illustrate the issue, suppose the class conditional distributions are Gaussian with equal covariance matrices. Then the Bayes classifier is a hyperplane and LDA provides a consistent rule. If a feature set has an error not close to the Bayes error, then this difference is clearly a consequence of the feature-

selection algorithm (and its application to a sample of the given size). But what of the situation when there are two class conditional Gaussian distributions with unequal covariance matrices and the Bayes classifier is determined by a quadratic surface? In this case, if LDA is the classification rule, then there is an inherent lower bound for the difference between the error of a designed classifier and that of the Bayes classifier, this bound determining the cost, where in the present case it arises from the fact that the LDA rule cannot achieve a better result than the optimal-hyperplane decision boundary. Rather than compare the error of the selected feature set using LDA to the Bayes error, would it not be better to compare it to the error of the optimal hyperplane decision boundary, which in this case exceeds the Bayes error? Certainly this is an option, but this would require that we know a classifier to whose error the errors of the designed classifiers converge in mean. Although there are some situations in which such a classifier is known, such cases are not common.

Aside from this practical reason for comparing the classifier error for a selected feature set to the minimal Bayes error among feature sets is that, were the size of the sample not restricted, one would not be using a constrained classification rule like LDA but would instead be estimating the Bayes classifier from an estimate of the feature-label distribution. Indeed, were it known that the class conditional distributions were Gaussian with unequal covariance matrices, were it not for insufficient data, we would be using quadratic discriminant analysis (QDA) as the classification rule. Using LDA instead of QDA is a form of regularization to offset the effects of too little data, so that, ipso facto, LDA is being used as the way to better design an approximation to the Bayes classifier. Hence, like the choice of feature-selection algorithm, the choice of LDA represents our effort to discover good features.

While taking as optimal the feature set with the minimal Bayes error is a suitable approach for model-based analysis in which a feature-label distribution is assumed, it

is not appropriate when considering patient data because we lack the Bayes classifier. For the patient data, the best feature sets are taken from a feature-selection test bed that utilizes the classification rule being applied [38].

C. Implementation

We employ two models in the model-based analysis. The optimal feature sets are known for both of these models. Some typical results will be displayed in here. The real-data results will use patient data from a study of survivability of cancer patients.

1. Model-based Study

The first model we use to generate synthetic sample points is called the *linear model* and is a two-class Gaussian model with the classes equally likely and the class-conditional densities being spherical Gaussians possessing common variance σ^2, the common covariance matrix being $\sigma^2\mathbf{I}$. One class mean is located at the origin $\vec{0}$ and the other at $\vec{A}$, where $\vec{A} = [a_1\ a_2\ \ldots\ a_D]$. The Bayes classifier is a hyperplane perpendicular to the axis joining the means. The second model is called the *quadratic model* and is similar to the first model, but instead of there being equal covariance matrices for the class-conditional densities, the covariance matrices are $\sigma_0^2\mathbf{I}$ and $\sigma_1^2\mathbf{I}$, for class 0 and class 1, respectively, with $\sigma_0 \neq \sigma_1$.

With variances fixed, the Bayes error is solely determined by the distance, $\|\vec{A}\|$, between the means of the classes. Moreover, since all features are independent, the set of the d best features is $A_{best} = \{a_{k1},\ a_{k2},\ \ldots,\ a_{kd}\}$, where $a_{k1}^2 + a_{k2}^2 + \cdots + a_{kd}^2$ is maximum for $1 \leqslant k_1, k_2, \cdots, k_d \leqslant D$. In the simulations, each a_i composing $\vec{A}$ is independently drawn from a beta distribution, $\mathbf{F}(\alpha, \beta)$. We further let α be fixed and let β follow a uniform distribution, $\mathbf{U}(\beta_1, \beta_2)$. To generate sample points, first we

draw randomly from $\mathbf{U}(\beta_1, \beta_2)$ to get β, and then from $\mathbf{F}(\alpha, \beta)$ to get $\vec{A}$. A sample set of size n is generated for each model. We repeat the procedure N times for a total of N random samples. Aside from A_{best}, which is determined before the sample points are generated, we find a feature set, $A_{fs(\mathcal{R})}$, using SFFS feature selection.

Overall, for the model-based study, the simulation utilizes the following protocol:

1. Choose a model by randomly selecting β from $\mathbf{U}(\beta_1, \beta_2)$ and then $a_1,\ a_2,\ \ldots,\ a_D$ from $\mathbf{F}(\alpha, \beta)$ to get $\vec{A}$.
2. Obtain $A_{best} = \{a_{k1},\ a_{k2},\ \ldots,\ a_{kd}\}$ from the model (A_{best} relative to the Bayes classifier).
3. Generate an n-point sample S from the model.
4. Design a classifier ψ_{best} for the feature set A_{best} according to the classification rule $\mathcal{R}$ from S.
5. Compute the error ε_{best} for ψ_{best} using the underlying distribution of the model.
6. Apply SFFS using the classification rule $\mathcal{R}$ on S to find a feature set $A_{fs(\mathcal{R})}$.
7. Design a classifier $\psi_{fs(\mathcal{R})}$ for the feature set $A_{fs(\mathcal{R})}$ according to the rule $\mathcal{R}$ from S.
8. Compute the error $\varepsilon_{fs(\mathcal{R})}$ for $\psi_{fs(\mathcal{R})}$ using the underlying distribution of the model.
9. Repeat steps 1 through 8 N times to form N error pairs $(\varepsilon^i_{best}, \varepsilon^i_{fs(\mathcal{R})})$, $i = 1, 2, \ldots, N$.

Since we have the underlying distribution, in steps 4 and 7 we can find the Bayes classifiers for A_{best} and $A_{fs(\mathcal{R})}$ instead of using the sample data, thereby leading to N Bayes-error pairs $(\xi^i_{best}, \xi^i_{fs(\mathcal{R})})$, $i = 1, 2, \ldots, N$.

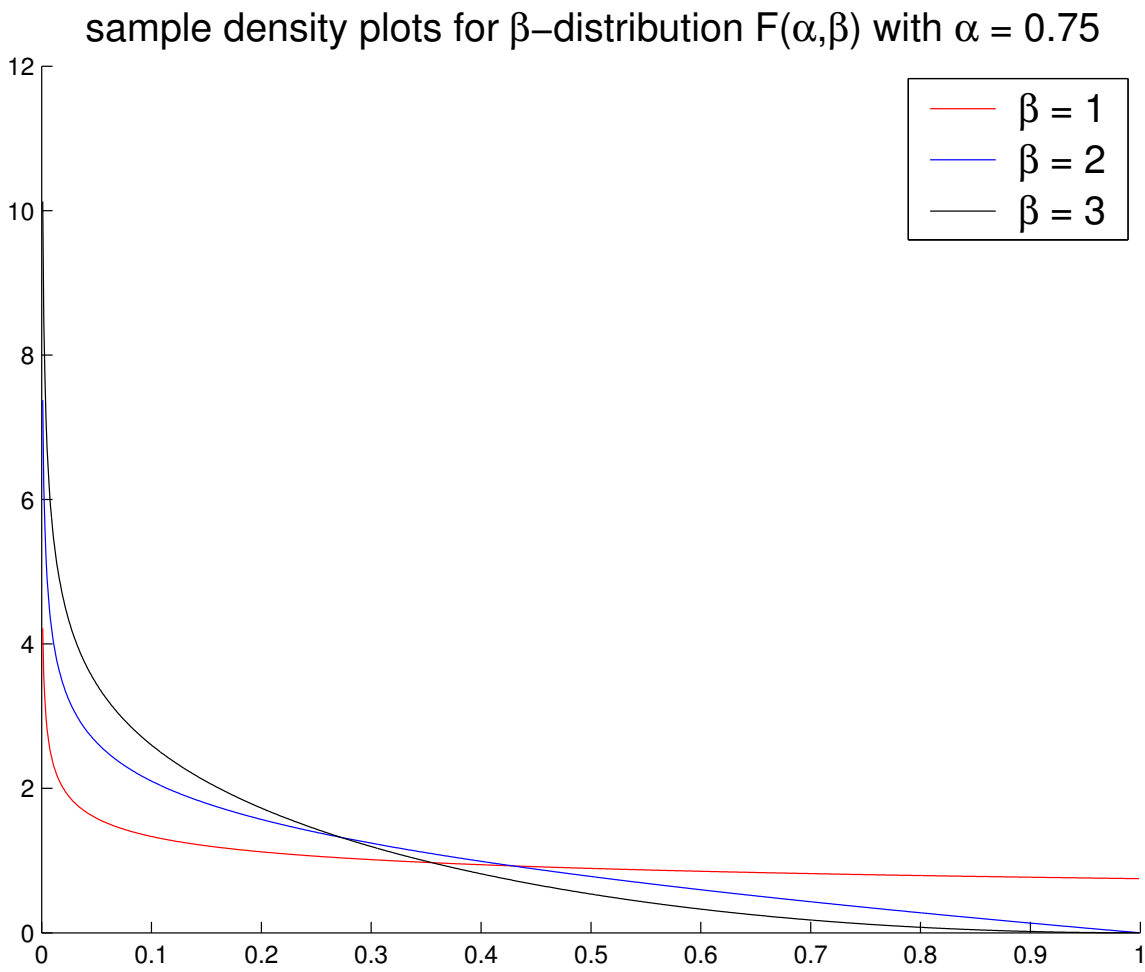


Fig. 5. Sample density plots for β-distribution $\mathbf{F}(\alpha, \beta)$ with $\alpha = 0.75$

A summary of the experiments with different parameters is provided in Table XI. In all experiments, $D = 500$ and $d = 5$ or $d = 10$. Sample plots for $\mathbf{F}(\alpha, \beta)$ are shown in Fig. 5.

We have two interests. First, given the error for the best feature set A_{best}, what error is expected for the SFFS feature set $A_{fs(\mathcal{R})}$? Second, given the error for $A_{fs(\mathcal{R})}$, what error is expected for A_{best}? We denote the results for the two scenarios by $A_{fs(\mathcal{R})}|A_{best}$ and $A_{best}|A_{fs(\mathcal{R})}$, respectively. Three figures are plotted for each combination of $\mathcal{R}$ and d in every experiment. For the first scenario the three figures are: (1)

Table XI. Experiments setup for model-based regression studies.

Exp	Model	σ	n	N	$\mathcal{R}$	D	d	β_1	β_2	α
Exp 1	linear model quadratic model	$\sigma_1 = \sigma_2 = 0.75$ $\sigma_1 = 0.6, \sigma_2 = 0.9$	50	12000	LDA, SVM and 3NN	500	$5, 10$	1	3	0.75
Exp 2	quadratic model	$\sigma_1 = 0.6, \sigma_2 = 0.9$	100	12000	LDA, 3NN					
Exp 3	quadratic model	$\sigma_1 = 0.8, \sigma_2 = 1.2$	50	6000						
Exp 4	quadratic model	$\sigma_1 = 0.8, \sigma_2 = 1.2$	100	6000						

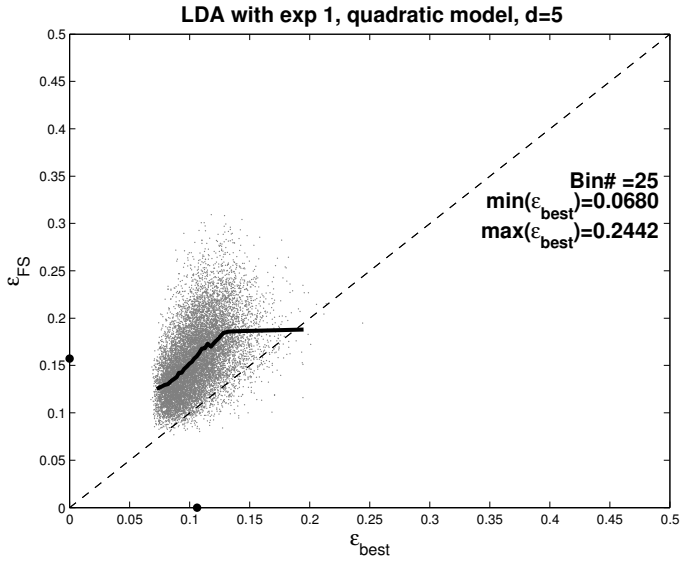
a scatter plot for $(\varepsilon_{best}, \varepsilon_{fs(\mathcal{R})})$, with the average errors marked with bold dots on their respective axes; (2) a curve of the conditional expectation $E[\varepsilon_{fs(\mathcal{R})}|\varepsilon_{best}]$, estimated by dividing all points into bins based on ε_{best}, with each bin containing the same number of points, and averaging the corresponding values of $\varepsilon_{fs(\mathcal{R})}$ in each bin; and (3) the scatter plot superimposed with the expectation curve. For the second scenario, the figures are the same but with the roles of ε_{best} and $\varepsilon_{fs(\mathcal{R})}$ reversed. In all three figure types, the 45° line is shown, along with the number of bins, and maximum and minimum values. Figs. 6 to 9 show examples of the scatter plots with superimposed expectation curves for the quadratic model in experiment 1.

2. Patient Study

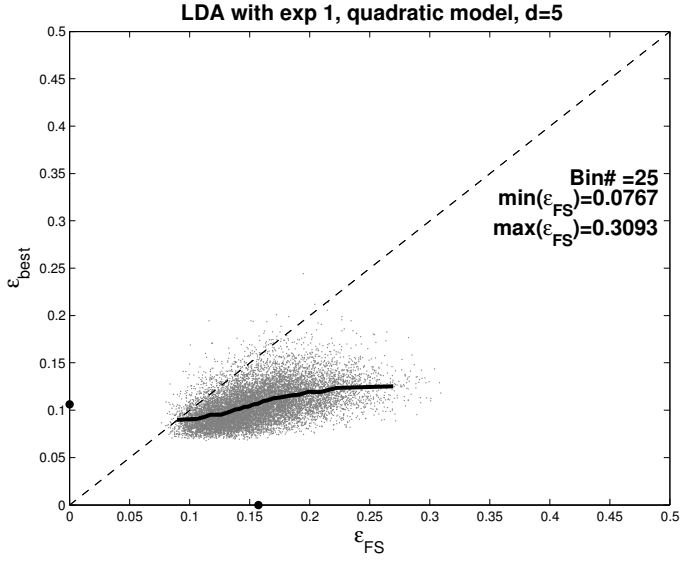
Similar experiments are conducted using the same patient data as in Chapter III, Section C.2: the microarrays prepared with RNA from breast tumor samples from 295 patients [33], 115 of which belong to the "good-prognosis" class and 180 belong to the "poor-prognosis" class.

Our experiment uses intensity gene-expression values associated with the $D = 70$ genes. The best feature sets of size $d = 5, 6$ and 7 are obtained from the test bed developed in [38]. This test bed utilizes high-performance computing to find optimal feature sets for empirical distributions via exhaustive searches. Because we lack the Bayes classifier in this empirical study, the best feature sets are taken from the test bed for the rule $\mathcal{R}$ being considered. The following protocol is utilized for the patient data:

1. Obtain $A_{best} = \{a_{k1}, a_{k2}, \ldots, a_{kd}\}$ from the genomic test bed for the rule $\mathcal{R}$.
2. Generate a 50-point sample S from the 295-point empirical distribution.
3. Design a classifier ψ_{best} for the feature set A_{best} according to the rule $\mathcal{R}$ from S.

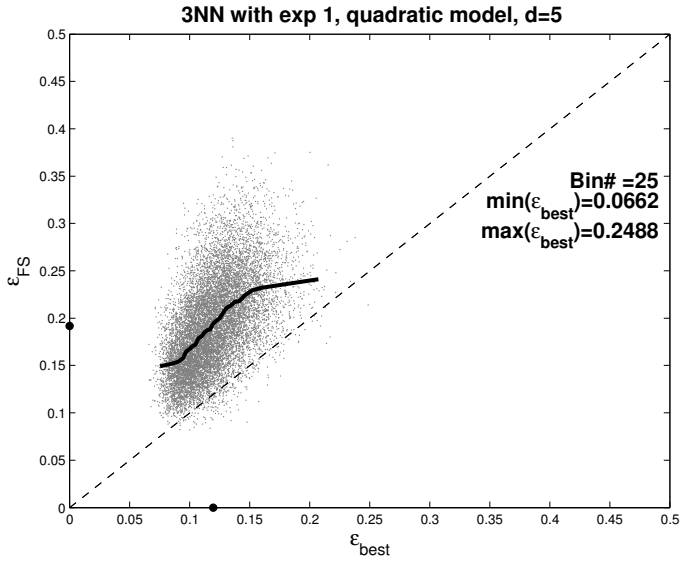


(a)

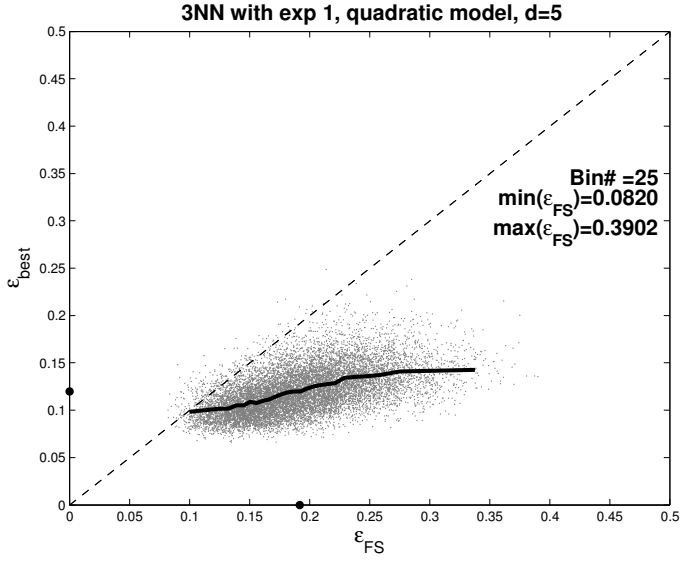


(b)

Fig. 6. Examples of the scatter plots with superimposed expectation curves for the quadratic model in Exp 1 for $d = 5$ and LDA: (a)$A_{fs(\mathcal{R})}|A_{best}$; (b) $A_{best}|A_{fs(\mathcal{R})}$.

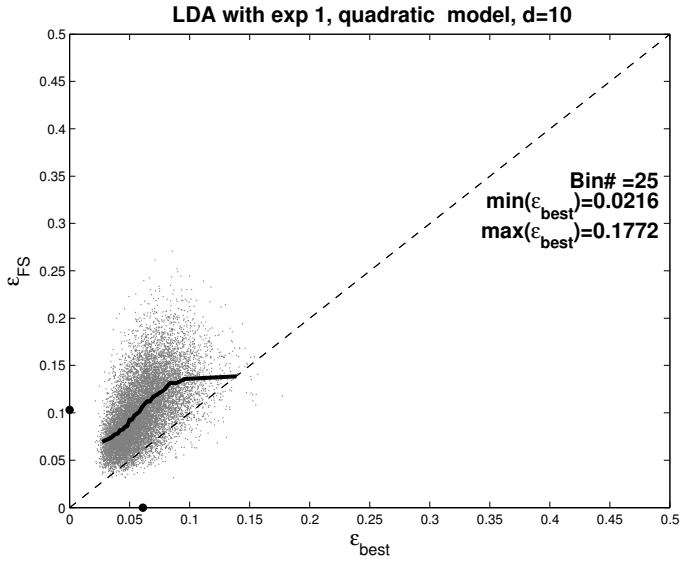


(a)

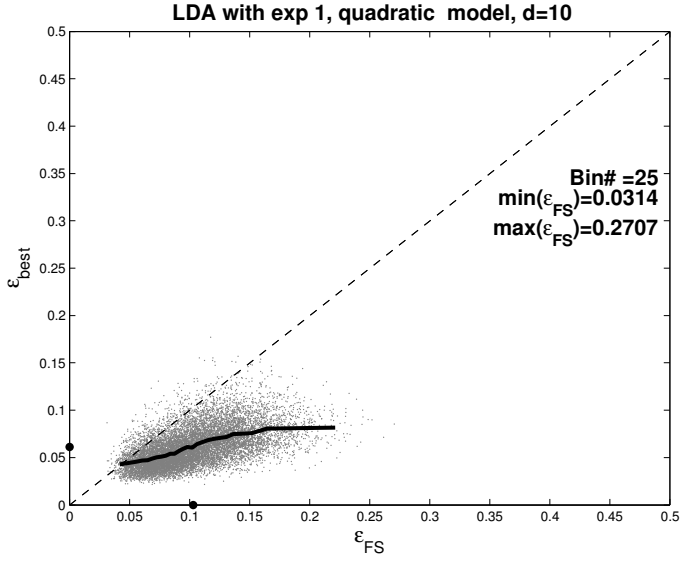


(b)

Fig. 7. Examples of the scatter plots with superimposed expectation curves for the quadratic model in Exp 1 for $d = 5$ and 3NN: (a)$A_{fs(\mathcal{R})}|A_{best}$; (b) $A_{best}|A_{fs(\mathcal{R})}$.

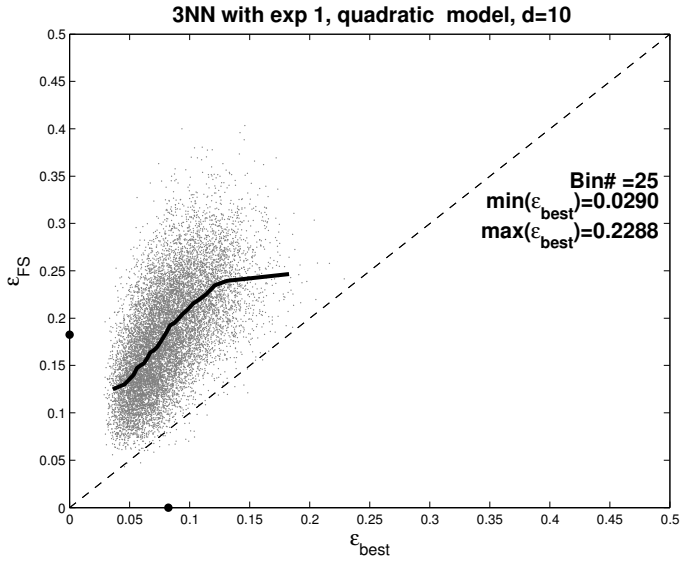


(a)

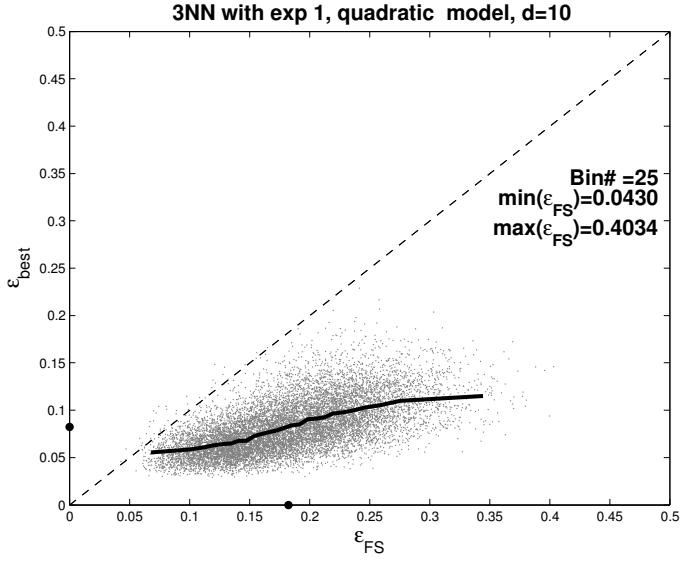


(b)

Fig. 8. Examples of the scatter plots with superimposed expectation curves for the quadratic model in Exp 1 for $d = 10$ and LDA: (a)$A_{fs(\mathcal{R})}|A_{best}$; (b) $A_{best}|A_{fs(\mathcal{R})}$.



(a)



(b)

Fig. 9. Examples of the scatter plots with superimposed expectation curves for the quadratic model in Exp 1 for $d = 10$ and 3NN: (a)$A_{fs(\mathcal{R})}|A_{best}$; (b) $A_{best}|A_{fs(\mathcal{R})}$.

4. Compute the error ε_{best} for ψ_{best} using hold-out on the 245 points not in S.

5. Apply SFFS using the rule $\mathcal{R}$ on S to find a feature set $A_{fs(\mathcal{R})}$.

6. Design a classifier $\psi_{fs(\mathcal{R})}$ for the feature set $A_{fs(\mathcal{R})}$ according to the rule $\mathcal{R}$ from S.

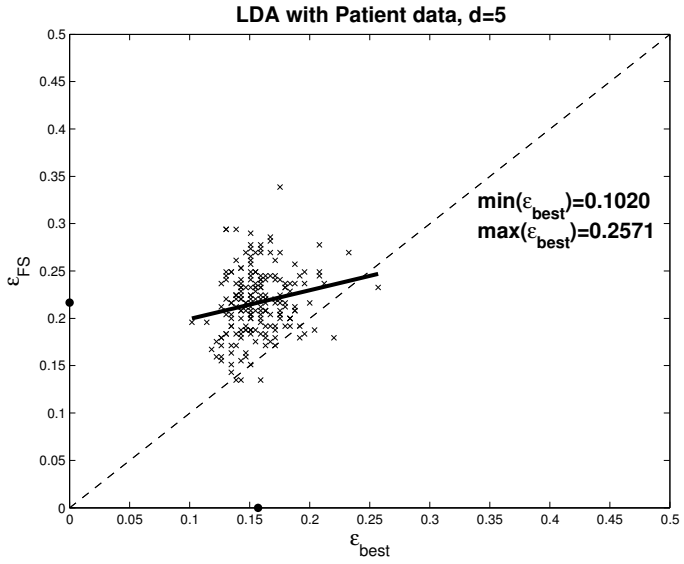
7. Compute the error $\varepsilon_{fs(\mathcal{R})}$ for $\psi_{fs(\mathcal{R})}$ using hold-out on the 245 points not in S.

8. Repeat steps 1 through 8 N times to form N error pairs $(\varepsilon^i_{best}, \varepsilon^i_{fs(\mathcal{R})})$, $i = 1, 2, \ldots, N$.

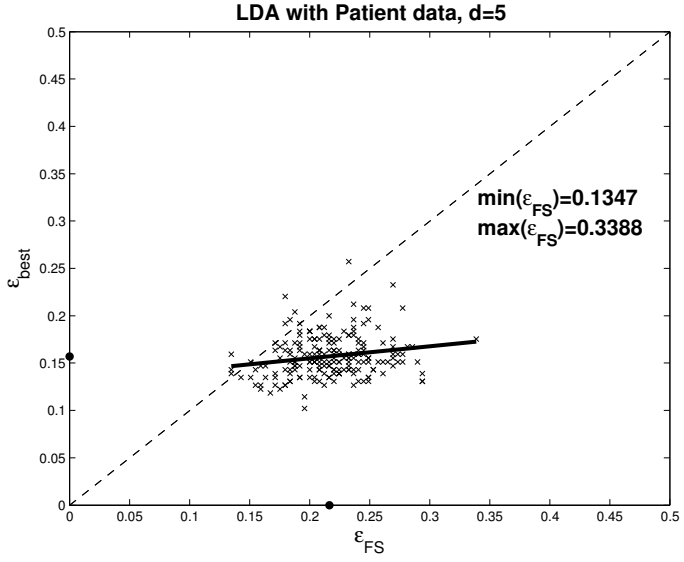
Once again it should be noted that the samples are not fully independent on account of overlap resulting from choosing the 50 sample points from among the same 295 sample points, but they are only weakly dependent. Owing to the dependency, we limit the total number of samples N to 200, which is sufficient for linear regression. For the patient data, corresponding to Figs. 6 to 9, Fig. 10 shows: (a) $A_{fs(\mathcal{R})}|A_{best}$ for LDA; (b) $A_{best}|A_{fs(\mathcal{R})}$ for LDA and Fig. 11 shows: (a) $A_{fs(\mathcal{R})}|A_{best}$ for 3NN; (b)$A_{best}|A_{fs(\mathcal{R})}$ for 3NN.

D. Discussion and Conclusion

In discussing the regression for the model-based study, we focus on the quadratic model of experiment 1. Similar observations apply to the other models. The regression $A_{fs(\mathcal{R})}|A_{best}$ concerns our first question, predicting the performance of a selected feature set based on the performance of the best feature set. Parts (a) of Figs. 6 and 7 provide the scatter plots and conditional expectations for LDA and 3NN for $d = 5$ (linear SVM being very similar to LDA). The regression for LDA is approximately parallel to the 45-degree line, with $E[\varepsilon_{fs(\mathcal{R})}|\varepsilon_{best}]$ exceeding ε_{best} by about

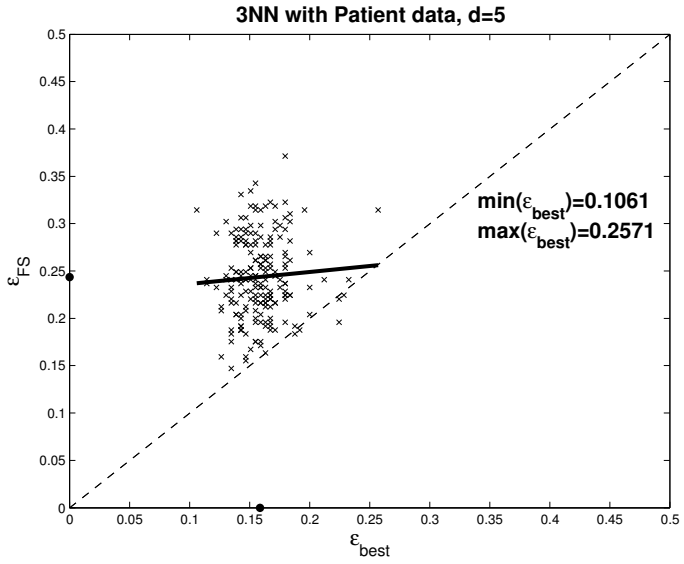


(a)

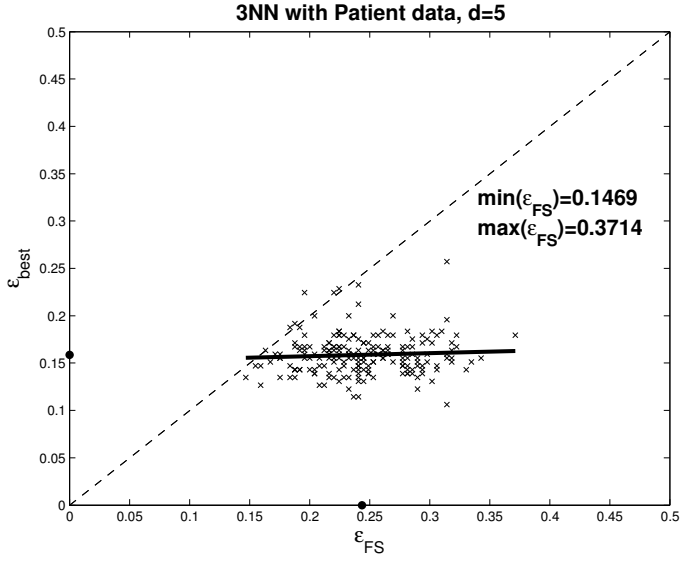


(b)

Fig. 10. Scatter plot and least squares linear regression line (bold line) for patient data, $d = 5$. Also marked are 45° line (dashed) and averages for ε_{best} and ε_{FS} (bold dots on axes). (a) $A_{fs(\mathcal{R})}|A_{best}$ for LDA; (b) $A_{best}|A_{fs(\mathcal{R})}$ for LDA.



(a)



(b)

Fig. 11. Scatter plot and least squares linear regression line (bold line) for patient data, $d = 5$. Also marked are 45° line (dashed) and averages for ε_{best} and ε_{FS} (bold dots on axes). (a) $A_{fs(\mathcal{R})}|A_{best}$ for 3NN; (b)$A_{best}|A_{fs(\mathcal{R})}$ for 3NN.

0.05 for the bulk of the mass, including the mean of ε_{best}. The situation improves for $\varepsilon_{best} > 0.13$, but there is little mass there. The situation is worse for 3NN, with $E[\varepsilon_{fs(\mathcal{R})}|\varepsilon_{best}]$ exceeding ε_{best} by about 0.07 for most of the mass, with improvement only for $\varepsilon_{best} > 0.15$ and the improvement being less pronounced. The corresponding plots for $d = 10$ are in parts (a) of Figs. 8 and 9. For LDA, the regression curve is similar to that in the case of $d = 5$, except that the errors are smaller and $E[\varepsilon_{fs(\mathcal{R})}|\varepsilon_{best}]$ exceeds ε_{best} by less. For 3NN, the regression curve is also similar and the errors are smaller; however, the amount by which $E[\varepsilon_{fs(\mathcal{R})}|\varepsilon_{best}]$ exceeds ε_{best} is substantially more than for $d = 5$, indicating worse prediction. The salient point deduced from the $A_{fs(\mathcal{R})}|A_{best}$ regression curves is that one can expect the error of a selected feature set to be substantially worse than the error of the best feature set.

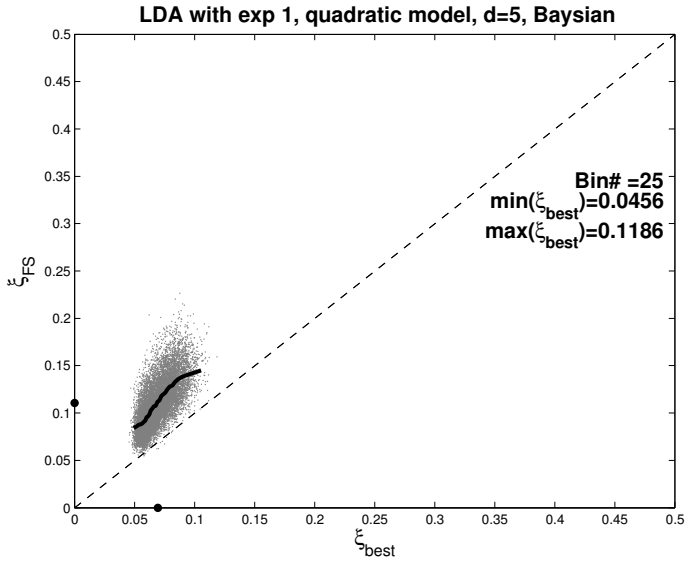
The regression $A_{best}|A_{fs(\mathcal{R})}$ concerns our second question, predicting the performance of the best feature set based on the performance of the selected feature set. Parts (b) of Figs. 6 and 7 provide the scatter plots and conditional expectations for LDA and 3NN for $d = 5$. In some sense, these are inverse to the $A_{fs(\mathcal{R})}|A_{best}$ plots, with $E[\varepsilon_{fs(\mathcal{R})}]$ exceeding $E[\varepsilon_{best}]$ exceeding by about 0.05 for LDA and 0.07 for 3NN. The difference is with the interpretation. Since the regression curves for $E[\varepsilon_{best}|\varepsilon_{fs(\mathcal{R})}]$ are close to being horizontal, and especially so for large feature-set errors, there is little relation between the errors of the classifiers designed from the selected and best feature sets. In particular, if feature selection results in a poor result, one should not conclude that there does not exist good feature sets. Indeed, if we look at Fig. 7(b) for 3NN, there is a substantial number of samples that yield $\varepsilon_{fs(\mathcal{R})} > 0.25$ and $\varepsilon_{best} < 0.13$, and many for which $\varepsilon_{fs(\mathcal{R})} > 0.30$ and $\varepsilon_{best} < 0.14$.

For the patient data, we focus on 3NN, referring to Fig. 11 (the results for LDA in Fig. 10 being very similar). In part (a), linear regression for the patient data yields a straight line that has important similarities with the curve for $A_{fs(\mathcal{R})}|A_{best}$

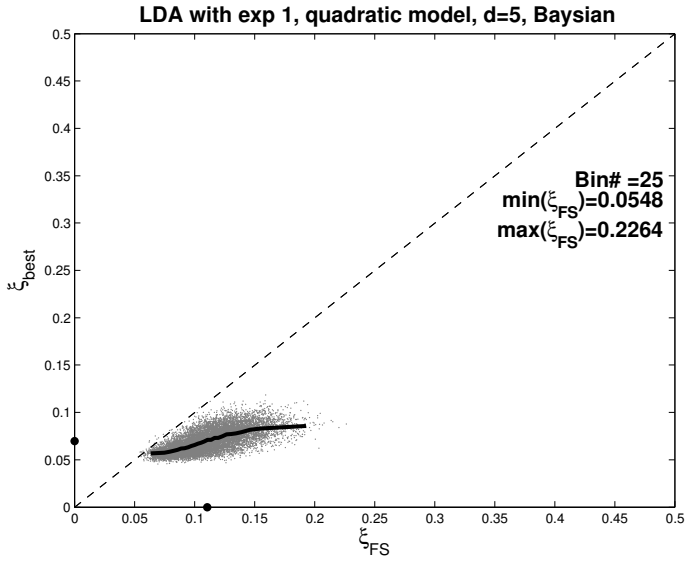
in the model-based study: (i) the line is increasing; (ii) the line lies almost entirely above the 45-degree line; (iii) for the bulk of the mass, $\varepsilon_{fs(\mathcal{R})}$ significantly exceeds ε_{best}, with $E[\varepsilon_{fs(\mathcal{R})}]$ exceeding $E[\varepsilon_{best}]$ by 0.08; and (iv) only for large values of ε_{best} can we expect the two errors to be close, and there is little mass in this region. As with the model-based analysis, one can expect the error of a selected feature set to be substantially worse than the error of the best feature set. Note, however, that with the patient data the regression line is more horizontal, indicating less predictability than for the synthetic data. The situation with $A_{best}|A_{fs(\mathcal{R})}$ for 3NN with the patient data is striking. The regression line is practically horizontal, and once again nothing can be concluded from a poor result when using feature selection. We note that using quadratic regression yields curves not significantly different than those for linear regression.

Although our main interest is with designed classifiers, in the model-based studies we have also considered the Bayes-error pairs $(\xi_{best}, \xi_{fs(\mathcal{R})})$, with one of the results shown in Fig. 12. While there are some differences in the curvatures of the regression curves for the Bayes-error pairs, these are not significant. The main difference in the Bayes-error scatter plots are that they are tighter and show smaller errors than for the designed classifiers (as is expected). In the model-based studies we always have $\xi_{best} < \xi_{fs(\mathcal{R})}$, indicating that $A_{fs(\mathcal{R})}$ is not optimal. On the other hand, while $\varepsilon_{best} < \varepsilon_{fs(\mathcal{R})}$ for most points, there are points with $\varepsilon_{best} > \varepsilon_{fs(\mathcal{R})}$. This occurs because the optimal feature set is defined over the whole distribution, whereas feature selection is carried out over a particular sample, thereby making it is possible that ψ_{best}, designed according to the sample, may be outperformed by $\psi_{fs(\mathcal{R})}$.

For some concluding observations, the lack of relation between the errors of the best and selected feature sets is observed throughout our experiments, including both models and patient data, different classification rules, and different feature-



(a)



(b)

Fig. 12. Examples of the Bayes-error plots (scatter plots with superimposed expectation curves) for the quadratic model in Exp 1 for $d = 5$ and LDA: (a)$A_{fs(\mathcal{R})}|A_{best}$; (b) $A_{best}|A_{fs(\mathcal{R})}$.

set sizes. It is generally more evident for higher variance cases (experiments 3 and 4) than lower variance cases (experiments 1 and 2), and for smaller sample sizes (experiments 1 and 3) than for larger sample sizes (experiments 2 and 4), reflecting the comparative difficulty of feature selection. Regarding the focus questions for the study, it is unlikely that feature selection will yield a feature set whose error is close to that of the optimal feature set, and the inability to find a good feature set should not lead to the conclusion that good feature sets do not exist.

CHAPTER VI

CONCLUSION

High-throughput technologies for rapid measurement of vast numbers of biological variables offer the potential for highly discriminatory diagnosis and prognosis; however, high dimensionality together with small samples creates the need for feature selection, while at the same time making feature-selection algorithms less reliable. Feature selection is required because the number of features is large with respect to the sample size because the use of a large number of features can result in overfitting the data: the designed classifier performs well on the sample data but not on the feature-label distribution from which the data have been drawn. However, a major impediment to feature selection is the combinatorial nature of the problem. To select a subset of d features from a set of D potential features and be assured that it provides an optimal classifier with minimum error among all optimal classifiers for subsets of size d, all d-element subsets must be checked unless there is distributional knowledge that mitigates the search requirement, a condition rarely satisfied in practice [26]. Thus a suboptimal feature-selection algorithm is required.

In this dissertation, we have studied that: for small-sample settings, feature selection is problematic. We haven shown that even if we were able to consider all feature subsets, in small-sample settings their ranking would be strongly affected by inaccurate error estimation. Also, if error estimation (or other parameter estimation) is required for a feature-selection algorithm, then the impact of error estimation can be greater than the choice of algorithm. Since in practice only suboptimal feature-selection algorithm could be used, the combined factors of inaccurate error estimation and suboptimal feature-selection algorithm make feature selection process in small-sample settings very unreliable and highly inaccurate. As in our regression study

shows, it is unlikely that feature selection will yield a feature set whose error is close to that of the optimal feature set, and the inability to find a good feature set should not lead to the conclusion that good feature sets do not exist.

REFERENCES

[1] J. L. DeRisi, V. R. Iyer, and P. O. Brown, "Exploring the metabolic and genetic control of gene expression on a genomic scale," *Science*, vol. 278, pp. 680–686, 1997.

[2] D. J. Duggan, M. L. Bittner, Y. Chen, P. S. Meltzer, and J. M. Trent, "Expression profiling using cDNA microarrays," *Nature Genetics*, vol. 21, pp. 10–14, 1999.

[3] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray," *Science*, vol. 270, pp. 467–470, 1995.

[4] M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, N. Sampas, E. Dougherty, E. Wang, F. Marincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpten, Leja D. Gillanders, E. and, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, V. Sondak, N. Hayward, and J. Trent, "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, no. 6795, pp. 536–540, 2000.

[5] S. A. Armstrong, J. E. Staunton, L. B. Silverman, R. Pieters, M. L. den Boer, M. D. Minden, S. E. Sallan, E. S. Lander, T. R. Golub, and S. J. Korsmeyer, "Mll tanslocations specify a distinct gene expression profile that distinguishes a unique leukemia," *Nature Genetics*, vol. 30, pp. 41–47, 2002.

[6] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, Loh M. L. Coller, H. and, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and

E. S. Lander, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[7] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. A. Olson, J. R. Jr. Marks, and J. R. Nevins, "Predicting the clinical status of human breast cancer by using gene expression profiles," *Proc Natl Acad Sci*, vol. 98, pp. 11462–11467, 2001.

[8] S. P. Bohen, O. G. Troyanskaya, O. Alter, R. Warnke, D. Botstein, P.O. Brown, and R. Levy, "Variation in gene expression patterns in follicular lymphoma and the response to rituximab," *Proc Natl Acad Sci*, vol. 100, no. 4, pp. 1926–30, 2003.

[9] F. Tschentscher, J. Hüsing, T. Hölter, E. Kruse, I. G. Dresen, K. H. Jöckel, G. Anastassiou, H. Schilling, N. Bornfeld, B. Horsthemke, D. R. Lohmann, and M. Zeschnigk, "Tumor classification based on gene expression profiling shows that uveal melanomas with and without monosomy 3 represent two distinct entities," *Cancer Research*, vol. 63, pp. 2578–2584, 2003.

[10] C. L. Nutt, D. R. Mani, R. A. Betensky, P. Tamayo, J. G. Cairncross, C. Ladd, U. Pohl, C. Hartmann, M. E. McLaughlin, T. T. Batchelor, P. M. Black, A. von Deimling, S. L. Pomeroy, T. R. Golub, and D. N. Louis, "Gene expression-based classification of malignant gliomas correlates better with survival than histological classification," *Cancer Research*, vol. 63, pp. 1602–1607, 2003.

[11] M. E. Schaner, "Gene expression patterns in ovarian carcinomas," *Mol Biol Cell*, vol. 14, no. 11, pp. 4376–4386, 2003.

[12] L. Li, C. R. Weinberg, T. A. Darden, and L. G. Pedersen, "Gene selection for sample classification based on gene expression data: Study of sensitivity to choice of parameters of the GA/KNN method," *Bioinformatics*, vol. 17, pp. 1131–1142, 2001.

[13] S. Kim, E. R. Dougherty, I. Shmulevich, K. R. Hess, S. R. Hamilton, J. M. Trent, G. N. Fuller, and W. Zhang, "Identification of combination gene sets for glioma classification," *Molecular Cancer Therapeutics*, vol. 1, no. 13, pp. 1229–1236, 2002.

[14] E. R. Dougherty, "Small sample issues for microarray-based classification," *Comparative and Functional Genomics*, vol. 2, pp. 28–34, 2001.

[15] A. K. Jain and B. Chandrasekaran, "Dimensionality and sample size considerations in pattern recognition practice", in *Handbook of Statistics*, vol. 2, P.R. Krishnaiah and L.N. Kanal, Eds., Amsterdam: North-Holland, 1982.

[16] S. J. Raudys and A. K. Jain, "Small sample size effects in statistical pattern recognition: Recommendations for practitioners," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 252–262, 1991.

[17] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, New York: Springer-Verlag, 1996.

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

[19] U. M. Braga-Neto and E. R. Dougherty, "Is cross-validation valid for small-sample microarray classification?," *Bioinformatics*, vol. 20, pp. 374–380, 2004.

[20] B. Efron, "Bootstrap methods: Another look at the jacknife," *Annals of Statistics*, vol. 7, pp. 1–26, 1979.

[21] B. Efron, "Estimating the error rate of a prediction rule: Improvement on cross-validation," *Journal of the American Statistical Society*, vol. 78, pp. 316–331, 1983.

[22] U. M. Braga-Neto and E. R. Dougherty, "Bolstered error estimation," *Pattern Recognition*, vol. 37, pp. 1267–1281, 2004.

[23] M. Evans, N. Hastings and B. Peacock, *Statistical Distributions*, 3rd edition, New York: John Wiley and Sons Inc., July 2000.

[24] W.M. Liu, R. Mei, X. Di, T.B. Ryder, E. Hubbell, S. Dee, T.A. Webster, C.A. Harrington, M.H. Ho, J. Baid, and S.P. Smeekens, "Analysis of high density expression microarrays with signed-rank call algorithms," *Bioinformatics*, vol. 18, no. 12, pp. 1593–1599, 2002.

[25] Y. Chen, V. Kamat, E. R. Dougherty, M. L. Bittner, P. S. Meltzer, and J. M. Trent, "Ratio statistics of gene expression levels and applications to microarray data analysis," *Bioinformatics*, vol. 18, pp. 1207–1215, 2002.

[26] T. M. Cover and J. Van Campenhout, "On the possible orderings in the measurement selection problem," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 7, no. 9, pp. 657–661, 1977.

[27] E.B. Suh, E.R. Dougherty, S. Kim, M.L. Bittner, D.E. Russ, Y. Chen, and R.R. Martino, "Parallel computation and visualization tools for codetermination analysis of multivariate gene-expression relations," in *Computational And Statistical Approaches To Genomics*, W. Zhang and I. Shmulevich, Eds. New York: Kluwer Academic Publishers, 2002.

[28] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Trans. Computers*, vol. 26, pp. 917–922, 1977.

[29] U. M. Braga-Neto, R. Hashimoto, E. R. Dougherty, D. V. Nguyen, and R. J. Carroll, "Is cross-validation better than resubstitution for ranking genes?," *Bioinformatics*, vol. 20, pp. 253–258, 2004.

[30] A. K. Jain and D. Zongker, "Feature selection – evaluation, application, and small sample performance," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 153–158, 1997.

[31] M. Kudo and J. Sklansky, "Comparison of algorithms that select features for pattern classifiers," *Pattern Recognition*, vol. 33, pp. 25–41, 2000.

[32] C. Ambroise and G. J. McLachlan, "Selection bias in gene extraction on the basis of microarray gene-expression data," *PNAS*, vol. 99, pp. 6562–6566, 2002.

[33] M. J. van de Vijver, Y. D. He, L. J. van't Veer, H. Dai, A. A. M. Hart, D. W. Voskuil, G. J. Schreiber, J. L. Peterse, C. Roberts, and M. J. Marton, "A gene-expression signature as a predictor of survival in breast cancer," *New Eng. J. Med.*, vol. 347, pp. 1999–2009, 2002.

[34] L. J. van't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. M. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, and A. T. Witteveen, "Gene expression profiling predicts clinical outcome of breast cancer," *Nature*, vol. 415, pp. 530–36, 2002.

[35] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, pp. 1119–1125, 1994.

[36] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd edition, Boston: Academic Press, 1990.

[37] C. Sima, S. Attoor, U. Braga-Neto, J. Lowey, E. Suh, and E. R. Dougherty, "Impact of error estimation on feature-selection algorithms," *Pattern Recognition*, vol. 38, no. 12, pp. 2472–2482, 2005.

[38] A. Choudhary, M. Brun, J. Hua, J. Lowey, E. Suh, and E.R. Dougherty, "Genetic test bed for feature selection," *Bioinformatics*, 2006, (in press).

VITA

Name:	Chao Sima
Address:	Department of Electrical and Computer Engineering, MS 3128 c/o Edward Dougherty Texas A&M University, College Station, TX 77843-3128
Email Address:	simachao@tamu.edu
Education:	Ph.D., Texas A&M University, USA, 2006 B.Eng., Xi'an Jiaotong University, China, 1995