BIOINFORMATICS

Vol. 20 no. 2 2004, pages 253–258
DOI: 10.1093/bioinformatics/btg399

Is cross-validation better than resubstitution for ranking genes?

Ulisses Braga-Neto[1,4], Ronaldo Hashimoto[3,4], Edward R. Dougherty[2,4,*], Danh V. Nguyen[5] and Raymond J. Carroll[5]

[1]Section of Clinical Cancer Genetics and [2]Department of Pathology, University of Texas M. D. Anderson Cancer Center, Houston, TX, USA, [3]Departamento de Ciencia de Computação, Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, Brazil, [4]Department of Electrical Engineering and [5]Department of Statistics, Texas A&M University, College Station, TX, USA

Received on January 24, 2003; revised on April 29, 2003; accepted on August 5, 2003

ABSTRACT

Motivation: Ranking gene feature sets is a key issue for both phenotype classification, for instance, tumor classification in a DNA microarray experiment, and prediction in the context of genetic regulatory networks. Two broad methods are available to estimate the error (misclassification rate) of a classifier. Resubstitution fits a single classifier to the data, and applies this classifier in turn to each data observation. Cross-validation (in leave-one-out form) removes each observation in turn, constructs the classifier, and then computes whether this leave-one-out classifier correctly classifies the deleted observation. Resubstitution typically underestimates classifier error, severely so in many cases. Cross-validation has the advantage of producing an effectively unbiased error estimate, but the estimate is highly variable. In many applications it is not the misclassification rate *per se* that is of interest, but rather the construction of gene sets that have the potential to classify or predict. Hence, one needs to rank feature sets based on their performance.

Results: A model-based approach is used to compare the ranking performances of resubstitution and cross-validation for classification based on real-valued feature sets and for prediction in the context of probabilistic Boolean networks (PBNs). For classification, a Gaussian model is considered, along with classification via linear discriminant analysis and the 3-nearest-neighbor classification rule. Prediction is examined in the steady-distribution of a PBN. Three metrics are proposed to compare feature-set ranking based on error estimation with ranking based on the true error, which is known owing to the model-based approach. In all cases, resubstitution is competitive with cross-validation relative to ranking accuracy. This is in addition to the enormous savings in computation time afforded by resubstitution.

Contact: edward@ee.tamu.edu

*To whom correspondence should be addressed.

1 INTRODUCTION

When choosing among a collection of potential feature sets for classification (prediction), estimating the errors of designed classifiers (predictors) is a key issue. In particular, we may wish to order the potential feature sets according to the misclassification rates of their corresponding classifiers (Xiong *et al*., 2001a,b; Kim *et al*., 2002). For this kind of ranking, which falls into the category of biomarker identification, the degree to which an error estimator preserves the true ordering is the most critical aspect of its performance. Two common methods for estimating misclassification rates are *resubstitution* and *cross-validation*. In resubstitution, one simply constructs the classifier based on all the data, and then applies this classifier in turn to each observation. In cross-validation, one successively holds out observations from the data set, constructs the classifier based on the reduced data, and then observes whether the held-out observations are correctly classified. Here, we consider the leave-one-out form of cross-validation.

For most feature-label distributions and classification rules, resubstitution underestimates the rate of misclassification, and in some instances this bias can be severe, the most extreme case being single-nearest-neighbor classification, where the resubstitution estimate is always zero. On the other hand, cross-validation is close-to-unbiased in the following sense: if the procedure is repeated for different samples drawn from a population, then the average error estimate will approximate the expected error of the designed classifiers across all possible same-size samples. Owing to bias considerations, cross-validation is commonly preferred over resubstitution. However, for small samples, estimator variance needs to be considered, and here resubstitution is superior to cross-validation (Devroye *et al*., 1996). This small-sample issue is especially relevant when working with cDNA microarrays.

In the context of microarray experiments, error estimation *per se* is perhaps not the major issue; indeed, the generation of potential feature sets to be investigated in confirmatory experiments is at least as crucial as error estimation. This paper compares resubstitution and cross-validation relative to the degree to which performance ranking based on them is in accord with the true feature-set ranking. Two kinds of ranking problems are considered, both relating to estimating a discrete random variable on the basis of other observed random variables. In statistical terminology, both estimations are referred to as either classification or prediction; however, it is not uncommon to differentiate them based on their settings, and we will adhere to this dichotomy.

The first problem involves estimating class labels. Here, there are classes of observations and a classifier is designed to estimate the class to which an observation belongs. For cancer classification, the observations are gene-expression levels and the classes are cancer types. We will refer to label estimation as *classification* and will confine ourselves to two classes labeled 0 and 1, the ideas generalizing easily to multi-class classification. The observations upon which the classifier operates compose the feature set for the classifier. The input observations are analog and the output is binary. Owing to the fact that we need ground truth with which to compare resubstitution and cross-validation error estimation, we will postulate a feature-label distribution from which to draw random samples.

The second problem involves estimating a random quantity based on similar quantities and is called *prediction*. Our interest here is predicting the expression level of a target gene via the expression levels of related genes. The sets of predictor genes constitute the feature sets to be ranked. We will work in the context of probabilistic Boolean networks (PBNs) (Shmulevich *et al*., 2002). PBNs have been used to model genetic regulatory networks in a binary manner (0 down-regulated, 1 up-regulated). The intent here is to generate synthetic data that is 'regulatory' in nature, so that its distributional properties reflect to some extent expression data derived from genetic regulatory networks. Recalling the generation of random Boolean networks (Kaufmann, 1993), we consider random PBNs. For each randomly chosen PBN, we obtain the steady-state distribution of its corresponding Markov chain by running the network a large number of times, and consider multivariate gene prediction in the steady-state distribution. Since we have the steady-state distribution, we can compute actual prediction errors. We can then take random samples from the steady-state distribution, form error estimates from these samples via both resubstitution and cross-validation, and then compare rankings for the true and estimated errors.

To express matters more precisely, prediction in the Boolean context means deciding whether or not a particular target gene is expressed, an event denoted by a binary variable Y. Available are the binary gene expressions, $X_1, X_2, \ldots, X_n$, of n other genes. Owing to the small sample sizes in microarray experiments, classifiers based on feature sets of three or fewer genes are of particular interest. Even for a small regulatory network, there are many such feature sets. Our interest is in describing the best feature sets, i.e. the ones that give the lowest prediction errors. In practice, the best feature sets are the ones whose variables are chosen as the input variables for the functions determining the network transitions in a PBN—hence, the importance of ranking error estimates. Keeping with our interest in multivariate gene interaction and regulatory networks, rather than work with prediction error directly, we will do the analysis using the coefficient of determination (CoD). The CoD gives the relative increase in prediction accuracy for Y resulting from the observations $X_1, X_2, \ldots, X_n$ in comparison with the best predictor of Y in the absence of observations. The CoD has been used to quantify gene interaction, with particular interest being in ranking (Kim *et al*., 2000), and it is the measure actually used in constructing PBNs. Rankings based on the CoD are equivalent to rankings based on misclassification error.

The results of our model-based simulations will show that resubstitution is competitive with cross-validation in generating feature sets. Since cross-validation requires redesign of the classifier many times and thus requires much more computation time than resubstitution, this gives a powerful argument for the use of the latter.

2 BASIC ERROR THEORY

This section sets out the basic notions regarding error rates (and the CoD) and discusses their estimation. Although we will consider both classification and prediction, to avoid redundancy we will describe matters in the context of prediction, from which application to classification is straightforward. More complete descriptions can be found in the literature: theory (Devroye *et al*., 1996) and application (Duda *et al*., 2001).

Given there are n genes to predict the expression level of the target gene, we denote the variables corresponding to the expression levels of the predictors by $X_1, X_2, \ldots, X_n$ and that of the target-gene variable by Y. Assuming there are P samples, we let $Y_1, Y_2, \ldots, Y_P$ denote the binary outcome variables indicating the expression of the target gene in sample $p = 1, 2, \ldots, P$, respectively. For the p-th sample, in addition to Y_p there is an n-vector $\mathbf{X}_p = (X_{p1}, \ldots, X_{pn})$ corresponding to the outcomes of $X_1, X_2, \ldots, X_n$, where $X_{pi} = 1$ means that the i-th gene is expressed, and $X_{pi} = 0$ means that it is not expressed in the p-th sample.

By definition, a feature set is a subset of gene-expression levels used for prediction. Let S be the number of feature sets and $\mathbf{Z}_s = (Z_{s1}, Z_{s2}, \ldots, Z_{sq})$ denote the s-th feature set, where q is the number of genes whose expression levels are being used as predictors. The expression levels constituting a feature set form a subset of the n expression levels for the full gene set: $\{Z_{s1}, Z_{s2}, \ldots, Z_{sq}\} \subset \{X_1, X_2, \ldots, X_n\}$. A predictor is a binary-valued function, Ψ, of the features, with the error of Ψ being given by the probability of erroneous prediction, $\epsilon[\Psi] = \mathrm{pr}(\Psi(\mathbf{Z}_s) \neq Y)$. The optimal predictor for the s-th feature set is given by $\Psi_{\text{opt}}(\mathbf{Z}_s) = 1$ if $\text{pr}(Y = 1|\mathbf{Z}_s) > \text{pr}(Y = 0|\mathbf{Z}_s)$ and $\Psi_{\text{opt}}(\mathbf{Z}_s) = 0$ otherwise. Owing to optimality, $\epsilon[\Psi_{\text{opt}}] \leq \epsilon[\Psi]$ for any predictor Ψ.

A predictor $\widehat{\Psi}_s$ is designed using the sample data $\{(\mathbf{z}_{s1}, y_1), (\mathbf{z}_{s2}, y_2,), \ldots, (\mathbf{z}_{sP}, y_P)\}$ corresponding to the s-th feature set and its error is given by $\epsilon_s = \epsilon[\widehat{\Psi}_s] = \text{pr}(\widehat{\Psi}_s(\mathbf{Z}_s) \neq Y)$. The feature sets can then be ranked by the values of ϵ_s, $s = 1, 2, \ldots, S$. For predictor design (training), we use the plug-in rule, which has been used in a number of circumstances for gene-expression prediction. For this rule, the designed predictor is defined in the same manner as the Bayes (optimal) predictor from the distribution except that $\text{pr}(Y = 1|\mathbf{Z}_s)$ and $\text{pr}(Y = 0|\mathbf{Z}_s)$ are replaced by their frequency-based estimates (missing data and ties are resolved by using the majority label in the sample). In our simulations we can directly compute ϵ_s because we have the joint distribution of $\mathbf{Z}_s$ and Y.

In practice, the joint distribution of $\mathbf{Z}_s$ and Y is unknown and we resort to error estimation. The resubstitution error estimate, ϵ_{sR}, for the s-th feature set is found by designing the predictor $\widehat{\Psi}_s$ and taking the fraction of errors made by applying $\widehat{\Psi}_s$ on the sample data, an error occurring when $\widehat{\Psi}_s(\mathbf{z}_{sp}) \neq y_p$, for $p = 1, 2, \ldots, P$. The cross-validation (leave-one-out) error estimate for the s-th feature set is found by the following procedure. Design the predictor $\widehat{\Psi}_{s1}$ on the data set $\{(\mathbf{z}_{s2}, y_2), (\mathbf{z}_{s3}, y_3,), \ldots, (\mathbf{z}_{sP}, y_P)\}$ formed by leaving out the data point $(\mathbf{z}_{s1}, y_1)$ and define an error to occur if $\widehat{\Psi}_{s1}(\mathbf{z}_{s1}) \neq y_1$; design the predictor $\widehat{\Psi}_{s2}$ on the data set $\{(\mathbf{z}_{s1}, y_1), (\mathbf{z}_{s3}, y_3,), \ldots, (\mathbf{z}_{sP}, y_P)\}$ formed by leaving out the data point $(\mathbf{z}_{s2}, y_2)$ and define an error to occur if $\widehat{\Psi}_{s2}(\mathbf{z}_{s2}) \neq y_2$; and continue doing this for all P data points. The cross-validation error, ϵ_{sC}, is the fraction of errors made in this procedure.

The CoD measures the relative increase in prediction accuracy from using the feature set in comparison with the estimate of the target in the absence of predictors. Without predictors, Y is predicted to be the majority label m among $Y_1, Y_2, \ldots, Y_P$. The true error of this predictor is given by $\epsilon_0 = \text{pr}(Y \neq m) = \text{pr}(Y = 1 - m)$. The true CoD for the s-th feature set is defined as

$$\text{CoD}_s = \frac{\epsilon_0 - \epsilon_s}{\epsilon_0}.$$

For estimating the CoD in a practical situation, where the true underlying probabilities are not known, both ϵ_0 and ϵ_s have to be estimated using only the data. If $\widehat{\pi}$ is the fraction of the sample for which $Y = 1$, then the empirical error is $\hat{\epsilon}_0 = \min(\widehat{\pi}, 1 - \widehat{\pi})$. The empirical CoDs for resubstitution and cross-validation using the s-th feature set are defined similarly to CoD_s, with $\hat{\epsilon}_0$ in place of ϵ_0, and with ϵ_{sR} and ϵ_{sC} in place of ϵ_s, respectively. They are denoted by CoD_{sR} and CoD_{sC}, respectively. Feature sets with low error rates are those with high CoDs. Owing to the fact that estimates are being used, it is possible to have $\hat{\epsilon}_0 < \epsilon_{sR}$, in which case we define $\text{CoD}_{sR} = 0$. In addition, if $\hat{\epsilon}_0 = 0$, then we set $\text{CoD}_{sR} = 1$ if $\epsilon_{sR} = 0$, and $\text{CoD}_{sR} = 0$ otherwise (analogous remarks apply to CoD_{sC}).

3 MODELS

This section describes the models we will use for studying ranking based on resubstitution and cross-validation for both classification and prediction, as described in the Introduction section.

3.1 Gaussian model

We consider the standard Gaussian model, where 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 $a = (a_1, a_2, \ldots, a_n)$ is a parameter vector. Without loss of generality, we assume that $\|a\| = 1$. It is well known that the Bayes classifier is a hyperplane perpendicular to the axis joining the means, with Bayes error $\epsilon_{\text{BAYES}} = 1 - \Phi(\delta)$, where Φ is the standard normal cumulative distribution function. In particular, it follows that $\delta = \Phi^{-1}(1 - \epsilon_{\text{BAYES}})$, which allows one to find δ for a prescribed Bayes error.

The Bayes error increases with decreased class separation, and vice versa. If $\delta = 0$, then $\epsilon_{\text{BAYES}} = \frac{1}{2}$, which corresponds to total overlap between classes. On the other hand, if $\delta = \infty$, then $\epsilon_{\text{BAYES}} = 0$, which corresponds to complete separation between classes. A plot of ϵ_{BAYES} versus δ is shown in Figure 1.

If a subset L of the original variables is selected, then again one has a standard Gaussian model as before, but now the separation between the classes is a function of which variables are selected. More specifically, one has

$$\epsilon^L_{\text{BAYES}} = 1 - \Phi\left(\delta\sqrt{\sum_{k\in L} a_k^2}\right).$$

The Bayes error is a function of both the separation and the model parameters. If L contains all the original variables, then $\epsilon^L_{\text{BAYES}}$ reduces to ϵ_{BAYES}. To minimize $\epsilon^L_{\text{BAYES}}$ for a given number of selected variables, one should pick the variables corresponding to the largest parameters.

3.2 Probabilistic Boolean networks

A PBN is defined by a set of binary-valued nodes $\{X_1, X_2, \ldots, X_n\}$ and a list $F = \{F_1, F_2, \ldots, F_n\}$ of sets $F_i = \{f^{(i)}_1, f^{(i)}_2, \ldots, f^{(i)}_{l(i)}\}$ of Boolean functions. Each node $X_i \in \{0, 1\}$ represents the state (expression) of gene i, where $X_i = 1$ means that gene i is expressed and $X_i = 0$ means that it is not expressed. The set F_i contains the possible rules of regulatory interactions for gene i. For $j = 1, 2, \ldots, l(i)$, $f^{(i)}_j$ is a possible Boolean function determining the value of X_i in terms of some other gene states. The functions are called *predictors*. All genes (nodes) are updated synchronously and repeatedly in accordance with the functions assigned to them.

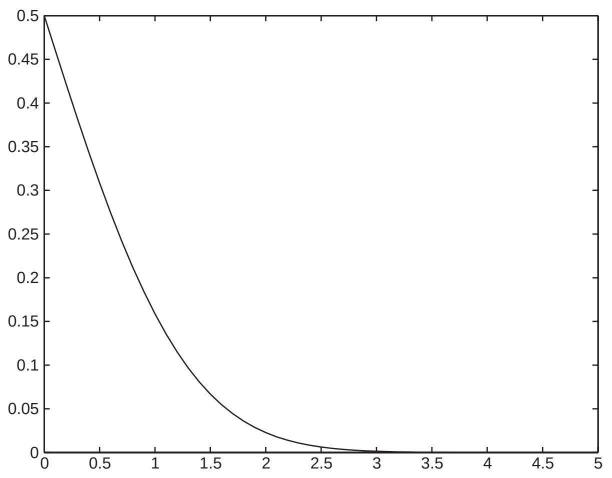

Fig. 1. Bayes error versus class separation in the Gaussian model.

A realization of a PBN at a given time is determined by a vector $\mathbf{f}$ of Boolean functions. If there are N possible realizations, then there are N vector functions $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N$ of the form $\mathbf{f}_k = (f^{(1)}_{k_1}, f^{(2)}_{k_2}, \ldots, f^{(n)}_{k_n})$, for $k = 1, 2, \ldots, N$, $1 \le k_i \le l(i)$, and $f^{(i)}_{k_i} \in F_i$ $(i = 1, 2, \ldots, n)$. The *network function* $\mathbf{f}_k$ acts as a transition mapping representing a possible realization of the entire PBN. Each predictor function usually has many fictitious variables, which means there are only a few input genes that actually regulate X_i at any given time. At each time point, the expression vector $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ is called a *gene activity pattern*.

Not only must the PBN transition between gene activity patterns, it must transition between network functions. A binary random variable Γ, with $\mathrm{pr}(\Gamma = 1) = \alpha$, governs whether or not there is a change of network function at each time instance. There is a network-function change if and only if $\Gamma = 1$. Γ is independent of the state of the network. Given a network change $(\Gamma = 1)$, there are selection probabilities $c_1, c_2, \ldots, c_N$ determining which of the network functions $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N$ will govern the network until the next switch.

The PBN model also allows for random perturbations. For each gene, there is a small probability β that it will flip its value, from 0 to 1 or from 1 to 0. Hence, there is a binary random variable Θ, independent of the network state and Γ, with $\mathrm{pr}(\Theta = 0) = (1-\beta)^n$, such that when $\Theta = 0$ the transition from one state to another occurs as usual via a network function, and when $\Theta = 1$ the state will change due to random bit permutation.

A PBN induces a homogeneous Markov chain whose states are pairs $(\mathbf{X}, \mathbf{f})$. The chain is ergodic and possesses a steady-state distribution. The steady-state distribution for the expression values is the marginal distribution of $\mathbf{X}$ (of the genes) relative to the steady-state distribution for the Markov chain.

4 RANKING OF GENE FEATURES

Ranking feature sets by estimations of their errors or CoDs should agree as much as possible to ranking by the true errors or true CoDs. Our question is whether resubstitution is as efficient in ranking the top feature sets as is cross-validation. We emphasize that our only interest is in ranking the best feature sets for classification; we have no interest in distinguishing among features sets that yield poor classifiers. In describing our ranking metrics, we stay with the convention we have adopted previously and describe matters in terms of CoDs, with the same definitions applying to error ranking, except with reverse ordering.

We consider three metrics for success. Suppose there are K feature sets that exceed a given threshold. Rank these feature sets according to their true CoDs, so that the feature sets are called $\mathbf{Z}_{s(1)} \ge \mathbf{Z}_{s(2)} \ge \cdots \ge \mathbf{Z}_{s(K)}$. Thus the feature set $\mathbf{Z}_{s(1)}$ has the highest true CoD, while $\mathbf{Z}_{s(K)}$ has the lowest true CoD among all feature sets whose true CoD exceeds the threshold. If there are no ties, the rank of $\mathbf{Z}_{s(1)} = 1$ and the rank of $\mathbf{Z}_{s(K)} = K$. In case of ties, the rank is equal to the mean of the ranks. Ranking based on the estimates is done analogously. We then have three ranks for a feature set: k (true), k^*_R (resubstitution) and k^*_C (cross-validation).

We consider three summary measures.

(1) For any given target gene, suppose that there are $K \ge 1$ feature sets whose CoD exceeds a threshold t such that $0 < t < 1$, the threshold being required so that only well-performing feature sets are considered. Then for that target gene the first summary statistics using resubstitution and cross-validation are, respectively,

$$R_1 = \frac{\sum_{k=1}^{K} |k - k^*_R|}{K}; \qquad C_1 = \frac{\sum_{k=1}^{K} |k - k^*_C|}{K}. \tag{1}$$

(2) We consider the top feature sets as measured by the true CoD. To do this, we consider only those target genes that have $K \ge 20$ features sets with true CoD exceeding the threshold t, and the summary statistics measure how far off the rankings are for just the top 20 feature sets: for resubstitution and cross-validation, respectively,

$$R_2 = \frac{\sum_{k=1}^{20} |k - k^*_R|}{20}; \qquad C_2 = \frac{\sum_{k=1}^{20} |k - k^*_C|}{20}. \tag{2}$$

(3) Lastly, we compare the top 20 lists irrespective of order. We again consider only those target genes that have $K \ge 20$ features sets with true CoD exceeding the threshold t. The summary statistics are the number of feature sets among the top 20 feature sets

that also appear in the top 20 using resubstitution and cross-validation, respectively,

$$R_3 = \sum_{k=1}^{20} I(k_R^* \leq 20); \qquad C_3 = \sum_{k=1}^{20} I(k_C^* \leq 20). \tag{3}$$

where I is the indicator function [i.e. $I(v) = 1$ if v is true, otherwise $I(v) = 0$]. Note that for this measure, higher scores are better.

5 EXPERIMENTAL RESULTS

Let us begin by discussing a few details about the simulation. For the Gaussian model, we let the total number of variables be 20 and consider feature sets of size 3. We choose the separation parameter δ so that the Bayes error in the original space is 0.1, and pick the parameter vector $a = (a_1, a_2, \ldots, a_n)$ 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 from the Gaussian model. In each of the 200 classification experiments, there are $C_3^{20} = 1140$ feature sets, which are used with two classification rules, namely, linear discriminant analysis (LDA) and 3-nearest-neighbor (3NN) classification. For each error threshold, and for each of the 200 experiments, the 1140 error triples $(\epsilon_s, \epsilon_{sR}, \epsilon_{sC})$ are ranked and the ranking statistics computed, resulting in a total of 200 ranking statistics pairs (R_1, C_1), (R_2, C_2) and (R_3, C_3).

For the PBN model, generation of random PBNs involves random generation of its constituent Boolean networks, which means random generation of the network functions. Although one can randomize the number of network functions and the number of essential variables for each component of a network function, we will fix the number of network functions at N and the number of essential variables at T. Each network function is of the form $\mathbf{f}_k = (f_{k_1}^{(1)}, f_{k_2}^{(2)}, \ldots, f_{k_n}^{(n)})$, for $k = 1, 2, \ldots, N$, and for $i = 1, 2, \ldots, n$, where n is the total number of genes. The component function $f_{k_i}^{(i)}$ is generated in two steps: (1) randomly select T genes from among $\{X_1, X_2, \ldots, X_n\}$ to be the variables for $f_{k_i}^{(i)}$; and (2) using these variables as entries in a truth table, uniformly randomly assign the 2^T values of the truth table. In our simulations, we set $n = 20$, $N = 5$ and $T = 3$, the latter being typical of the PBNs that have been generated in practice. We let the network functions have equal probability, $c_k = 0.2$, and we set $\alpha = 0.001$ and $\beta = 0.00001$.

There are 100 PBNs in our simulation, the large number being used to get a good sampling of PBNs. For each PBN, there is a total of 200 prediction experiments (20 genes $\times$ 10 samples). For each experiment, there are $C_3^{19} = 969$ feature sets. For each CoD threshold, and for each of the 20 000 experiments, the 969 CoD triples $(\mathrm{CoD}_s, \mathrm{CoD}_{sR}, \mathrm{CoD}_{sC})$ are ranked and the ranking statistics computed, resulting in a total of 20 000 ranking statistics pairs (R_1, C_1), (R_2, C_2) and (R_3, C_3).

The results of the experiments are shown in Figure 2, with the ranking-measure means graphed as functions of the error (CoD) threshold for feature sets considered. For instance, the bottom row displays the curves for the means of the 20 000 values for R_1, C_1, R_2, C_2, R_3 and C_3 as functions of the CoD threshold. The solid and dashed lines give the resubstitution and cross-validation measures, respectively. For instance, for (R_2, C_2) in the second row, the value 0.25 on the horizontal axis means that we are considering the ranking measure for all feature sets having error less than or equal to 0.25, and for those features sets, $R_2 \approx C_2 \approx 13$. The error range for 3NN (middle row) is higher than for LDA (top row) because the small samples have resulted in poorer training for 3NN than for LDA. For PBN CoDs (bottom row), we only consider up to CoD $= 0.85$ because for higher CoDs there are too few feature sets satisfying the threshold requirement to obtain good estimates of the ranking-measure means (note that CoD $= 0.85$ is very high, in practice CoD $= 0.7$ indicating strong multivariate gene interaction).

Figure 2 shows that, in the case of LDA, ranking accuracy based on resubstitution and cross-validation is virtually equal for all three measures for feature sets with true error below 0.2. For 3NN, cross-validation outperforms resubstitution for higher error rates (which are not of interest) but performance is virtually identical once in the range of error 0.26, with resubstitution actually slightly outperforming cross-validation below 0.25. For prediction in the PBN model, $R_1 \approx C_1$ for CoDs above 0.5, with resubstitution outperforming cross-validation slightly for all CoD values. Similar comments apply to (R_2, C_2) and (R_3, C_3).

6 CONCLUDING REMARKS

In microarray experiments, estimation of misclassification rates is perhaps not the major issue; indeed, the generation of potential feature sets to be investigated in confirmatory experiments is at least as crucial as estimation of misclassification rates *per se*. While resubstitution may be an inappropriate method for estimating error rates, a priori it is not clear that it is less appropriate than cross-validation for the purpose of ranking feature sets.

We have tested this question by generating data via a classification model and a gene-regulatory network prediction model. Our results show that resubstitution performs as well as cross-validation in terms of the metrics we considered, namely various versions of the ranking of top feature sets. Because resubstitution requires far less computation, the results indicate a powerful reason to consider the use of resubstitution for generating potential feature sets to be investigated in confirmatory experiments.

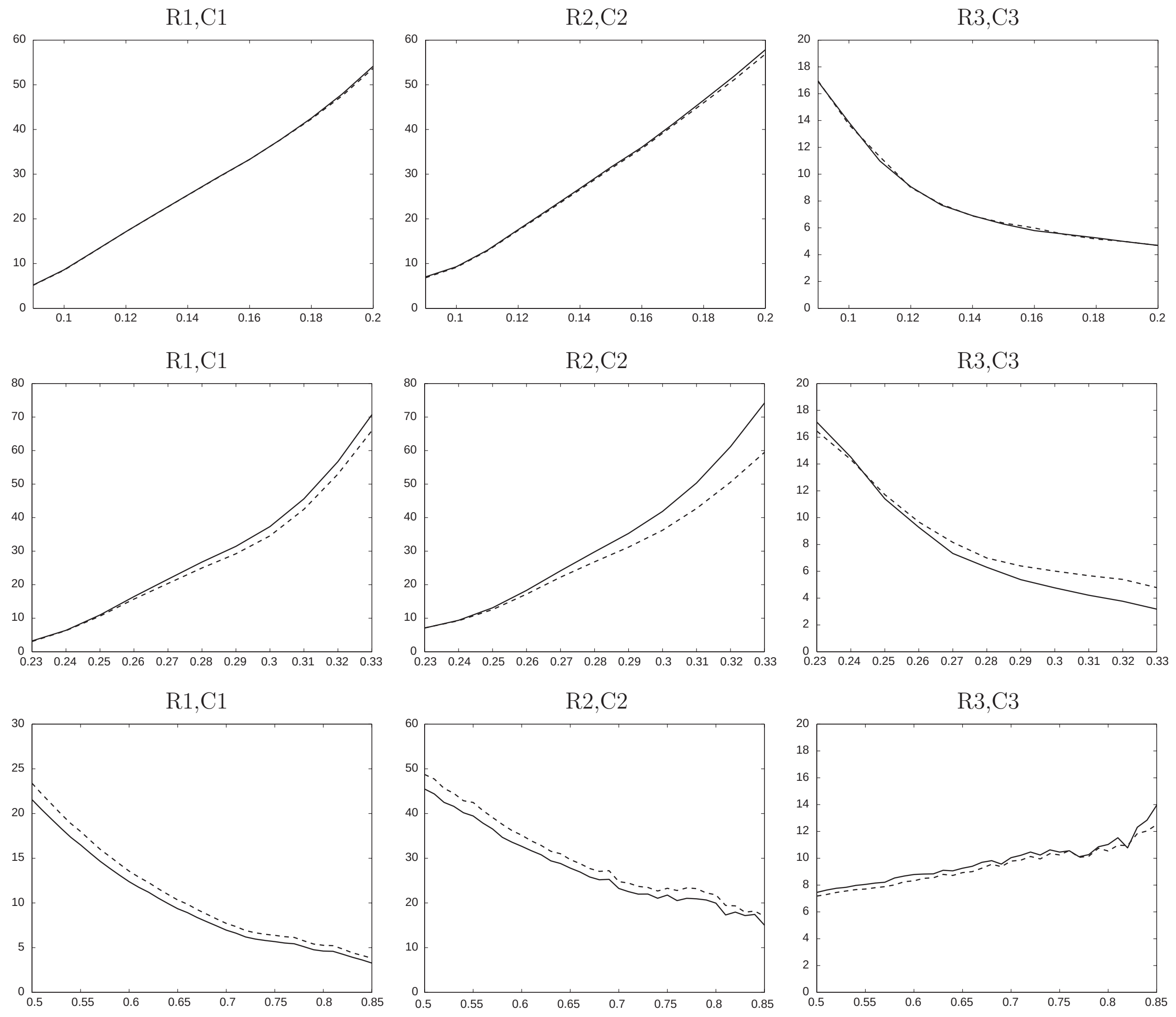

Fig. 2. Means of ranking statistics versus true error (resp. COD) threshold. Top row: Gaussian model, LDA. Middle row: Gaussian model, 3NN. Bottom row: PBN model. The solid line corresponds to resubstitution, while the dashed line corresponds to leave-one-out.

ACKNOWLEDGEMENTS

The research of R.H. and E.R.D. was supported by the National Human Genome Research Institute. U.B.-N. research was supported by a contract from the National Cancer Institute (N01-CN15102). D.V.N. research was supported by a grant from the National Cancer Institute (CA-90301). R.J.C. research was supported by a grant from the National Cancer Institute (CA-57030), and by the Texas A&M Center for Environmental and Rural Health via a grant from the National Institute of Environmental Health Sciences (P30-ES09106).

REFERENCES

Devroye,L., Gyorfi,L. and Lugosi,G. (1996) *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York.

Duda,R.O., Hart,P.E. and Stork,D.G. (2001) *Pattern Recognition*, 2nd edn. John Wiley, New York.

Kaufmann,S.A. (1993) *Origins of Order: Self-organization and Selection in Evolution*. Oxford University Press, New York.

Kim,S., Dougherty,E.R., Bittner,M.L., Chen,Y., Sivakumar,K., Meltzer,P. and Trent,J.M. (2000) A general framework for the analysis of multivariate gene interaction via expression arrays. *J. Biomed. Optics*, **4**, 411–424.

Kim,S., Dougherty,E.R., Barrera,J., Chen,Y., Bittner,M.L. and Trent,J.M. (2002) Strong feature sets from small samples. *J. Comput. Biol.*, **9**, 127–146.

Shmulevich,I., Dougherty,E.R., Kim,S. and Zhang,W. (2002) Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics,* **18**, 261–274.

Xiong,M., Li,W., Zhao,J., Jin,L. and Boerwinkle,E. (2001a) Feature (gene) selection in gene expression-based tumor classification. *Mol. Genet. Metab.*, **73**, 239–247.

Xiong,M., Fang,X. and Zhao,J. (2001b) Biomarker identification by feature wrappers. *Genome Res.*, **11**, 1878–1887.