BIOINFORMATICS

Vol. 20 no. 3 2004, pages 374–380
DOI: 10.1093/bioinformatics/btg419

Is cross-validation valid for small-sample microarray classification?

Ulisses M. Braga-Neto[1,3] and Edward R. Dougherty[2,3,*]

[1]Section of Clinical Cancer Genetics and [2]Department of Pathology, University of Texas MD Anderson Cancer Center, Houston, TX, USA and [3]Department of Electrical Engineering, Texas A&M University, College Station, TX, USA

Received on March 18, 2003; revised on July 3, 2003; accepted on August 7, 2003

ABSTRACT

Motivation: Microarray classification typically possesses two striking attributes: (1) classifier design and error estimation are based on remarkably small samples and (2) cross-validation error estimation is employed in the majority of the papers. Thus, it is necessary to have a quantifiable understanding of the behavior of cross-validation in the context of very small samples.

Results: An extensive simulation study has been performed comparing cross-validation, resubstitution and bootstrap estimation for three popular classification rules—linear discriminant analysis, 3-nearest-neighbor and decision trees (CART)—using both synthetic and real breast-cancer patient data. Comparison is via the distribution of differences between the estimated and true errors. Various statistics for the deviation distribution have been computed: mean (for estimator bias), variance (for estimator precision), root-mean square error (for composition of bias and variance) and quartile ranges, including outlier behavior. In general, while cross-validation error estimation is much less biased than resubstitution, it displays excessive variance, which makes individual estimates unreliable for small samples. Bootstrap methods provide improved performance relative to variance, but at a high computational cost and often with increased bias (albeit, much less than with resubstitution).

Availability and Supplementary information: A companion web site can be accessed at the URL http://ee.tamu.edu/∼edward/cv_ paper. The companion web site contains: (1) the complete set of tables and plots regarding the simulation study; (2) additional figures; (3) a compilation of references for microarray classification studies and (4) the source code used, with full documentation and examples.

Contact: edward@ee.tamu.edu

1 INTRODUCTION

A major interest in the application of expression microarrays is to perform classification via different expression patterns—for instance, cancer classification (see the companion web site for a compilation of references on microarray-based cancer classification). This requires assessing expression levels from RNA obtained from different tissues with microarrays, determining genes whose expression levels can be used as classifier variables, applying a rule to design the classifier from the sample data, and then applying an error estimation procedure.

Three critical issues arise. First, given a large set of variables (expression levels), how does one select a feature set? Second, given a feature set, how does one design a classifier from the sample data that provides good classification over the population? Third, how does one estimate the error of a designed classifier? Error estimation permeates the entire process because it is often a required step in both feature and model selection. A key point for microarray classification is that error estimation is greatly impacted by small samples (Dougherty, 2001). An estimator may be unbiased but have a large variance, and therefore often be low or high.

Cross-validation error estimation has been quite popular for microarray classification. Perhaps this is due to the fact that, on average, cross-validation error estimates nearly agree with the true errors. But is this important? Our concern is with error estimation obtained from the particular data set we have. This paper discusses the degree to which cross-validation procedures can be expected to estimate the true classification error from individual samples in a small-sample setting. Critical scientific issues are raised by using imprecise error estimation. Low estimation can lead to inferring a relation where there is none, or a strong relation when it is weak. High estimation can lead to inferring there is either no relation, or at best one beset by uncertainty, when there is a tight relation. In the first instance, one can use the microarray data in the context of gene discovery and attempt to validate the inference; in the second, there is no recovery. The problem can be so severe that perfectly consistent data (indicative of deterministic regulation) can yield error rates that make the

[*]To whom correspondence should be addressed at 214 Zachry Engineering Center, Department of Electrical Engineering, Texas A&M University, College Station, TX 77840, USA.

data appear substantially inconsistent (indicative of stochastic regulation).

2 ERROR ESTIMATION

In statistical pattern recognition, there is a *feature vector* $X \in \mathbb{R}^d$ and a *label* $Y \in \mathbb{R}$, which takes on numerical values representing the different classes; here, we assume a two-class problem, $Y = \{0, 1\}$. A *classifier* is a function $g: \mathbb{R}^d \rightarrow \{0, 1\}$. The *error rate* of g is $\epsilon[g] = P[g(X) \neq Y] = E(|Y - g(X)|)$, which depends on the feature-label distribution $\mathbf{F}$. The *Bayes classifier* is given by $g_{\text{BAY}}(x) = 1$ if $P(Y = 1 \mid X = x) > 1/2$ and $g_{\text{BAY}}(x) = 0$ otherwise. For any classifier g, $\epsilon[g_{\text{BAY}}] \leq \epsilon[g]$, so that g_{BAY} is the optimal classifier.

In practice, $\mathbf{F}$ is unknown. Hence, one must design a classifier 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 $g: \{\mathbb{R}^d \times \{0, 1\}\}^n \times \mathbb{R}^d \rightarrow \{0, 1\}$ (we use the same letter g to denote both a classifier and a classification rule, the distinction being clear from the context). A classification rule maps the training data S_n into the *designed classifier* $g(S_n, \cdot)$. The *true error* of a designed classifier is its error rate given a fixed training data set:

$$\epsilon_n = \epsilon[g(S_n, \cdot)] = E_{\mathbf{F}}(|Y - g(S_n, X)|), \tag{1}$$

where the notation $E_{\mathbf{F}}$ indicates expectation with respect to $\mathbf{F}$. The expected error rate over the data is given by $E[\epsilon_n] = E_{\mathbf{F}_n} E_{\mathbf{F}}[|Y - g(S_n, X)|]$, where $\mathbf{F}_n$ is the joint distribution of the data S_n. This is sometimes called the *unconditional error* of the classification rule.

Were the underlying feature-label distribution $\mathbf{F}$ known, the true error could be computed exactly, via (1). In practice, one is limited to using an *error estimator*, that should be as 'close' as possible to the true error. Most error estimators used in practice implement some form of sample-mean-like approximation to the expectation in (1). This approximation is unbiased if the test points come from independent samples, not used to design the classifier.

For large samples, one can randomly choose a subset $S_{n_t} \subset S_n$ for test data, design the classifier on $S_n \setminus S_{n_t}$, and then estimate its error by applying it to S_{n_t}. This *holdout estimator* approximates (1), with S_n replaced by $S_n \setminus S_{n_t}$, and is an unbiased estimator of $E[\epsilon_{n-n_t}]$, with respect to expectation over S_n (in this paper, the bias of error estimation always refers to expectation over S_n). Holdout estimation is impractical with small samples.

The *resubstitution estimator*, $\hat{\epsilon}_{\text{resub}}$, estimates the error by directly computing the error on the training data:

$$\hat{\epsilon}_{\text{resub}} = \frac{1}{n} \sum_{i=1}^{n} |y_i - g(S_n, x_i)|. \tag{2}$$

It is usually low-biased as an estimator of $E[\epsilon_n]$—and can be severely low-biased. Typically, bias is worse for more complex classifiers (Vapnik, 1998).

In *k-fold cross-validation*, 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 testing set, and the estimate is the overall proportion of error committed on all folds:

$$\hat{\epsilon}_{\text{cvk}} = \frac{1}{n} \sum_{i=1}^{k} \sum_{j=1}^{n/k} |y_j^{(i)} - g(S_n \setminus S_{(i)}, x_j^{(i)})|, \tag{3}$$

where $(x_j^{(i)}, y_j^{(i)})$ is a sample in the i-th fold. The process may be repeated, where several cross-validated estimates are computed, using different partitions of the data into folds, and the results averaged. In *stratified* cross-validation, the classes are represented in each fold in the same proportion as in the original data—there is evidence that this improves the estimator (Witten and Frank, 2000). Clearly, a k-fold cross-validation estimator is unbiased as an estimator of $E[\epsilon_{n-n/k}]$. In *leave-one-out estimation*, a single observation is left out each time, which corresponds to n-fold cross-validation. The leave-one-out estimator is nearly unbiased as an estimator of $E[\epsilon_n]$.

The *bootstrap* methodology is a general resampling strategy that can be applied to error estimation (Efron, 1979). It is based on the notion of an 'empirical distribution' $\mathbf{F}^*$, which puts mass $1/n$ on each of the n data points. A 'bootstrap sample' S_n^* from $\mathbf{F}^*$ consists of n equally-likely draws with replacement from the original data S_n. Hence, some of the samples will appear multiple times, whereas others will not appear at all. The probability that any given data point will not appear in S_n^* is $(1 - 1/n)^n \approx \mathrm{e}^{-1}$. It follows that a bootstrap sample of size n contains on average $(1 - \mathrm{e}^{-1})n \approx 0.632n$ of the original data points. The actual proportion of times a data point (x_i, y_i) appears in S_n^* can be written as $P_i^* = (1/n) \sum_{j=1}^{n} I_{(x_j^*, y_j^*) = (x_i, y_i)}$, where $I_S = 1$ if the statement S is true, zero otherwise. The *bootstrap zero estimator* (Efron, 1983), $\hat{\epsilon}_0$, mimics (1) with respect to the empirical distribution (note that S_n is fixed here): $\hat{\epsilon}_0 = E_{\mathbf{F}^*}(\,|Y - g(S_n^*, X)| : (X, Y) \in S_n \setminus S_n^*)$. The classifier is designed on the bootstrap sample and tested on the left-out data points. In practice, the expectation $E_{\mathbf{F}^*}$ has to be approximated by a sample mean based on independent replicates S_n^{*b}, for $b = 1, \ldots, B$, where B is recommended to be between 25 and 200 in Efron (1983):

$$\hat{\epsilon}_0 = \frac{\sum_{b=1}^{B} \sum_{i=1}^{n} |y_i - g(S_n^{*b}, x_i)| \, I_{P_i^{*b}=0}}{\sum_{b=1}^{B} \sum_{i=1}^{n} I_{P_i^{*b}=0}}. \tag{4}$$

A variance-reducing technique often employed is the *balanced* bootstrap resampling (Chernick, 1999), where each sample is made to appear exactly B times in the computation. The bootstrap zero estimator tends to be a high-biased estimator of $E[\epsilon_n]$, as the number of data points available for design is on average only $0.632n$. The *0.632 bootstrap estimator* (Efron, 1983),

$$\hat{\epsilon}_{\text{b632}} = (1 - 0.632)\,\hat{\epsilon}_{\text{resub}} + 0.632\,\hat{\epsilon}_0, \tag{5}$$

tries to correct this bias via a weighted average of the zero and resubstitution estimators.

For the sake of completeness, we also consider the *bias-corrected bootstrap estimator*:

$$\hat{\epsilon}_{bbc} = \hat{\epsilon}_{\text{resub}} + \frac{1}{B}\sum_{b=1}^{B}\sum_{i=1}^{n}\left(\frac{1}{n} - P_i^{*b}\right)|y_i - g(S_n^{*b}, x_i)|. \tag{6}$$

This estimator tries to correct directly the bias of resubstitution, by adding to $\hat{\epsilon}_{\text{resub}}$ its bootstrap estimation of bias (Efron, 1983).

3 PERFORMANCE OF ERROR ESTIMATORS

In our view, the salient issue regarding performance of error estimators in small-sample problems is variability. It is preferable to have some (low) degree of bias and small variance than to have unbiasedness and large variance. Unbiasedness is of limited use if the estimate corresponding to a given sample can be often far from the actual error value, due to high variability. Cross-validation estimators are especially problematic in small-sample settings, typically having higher variance than that of resubstitution or bootstrap estimators. The variance problem of cross-validation makes its use questionable for the kinds of very small samples used in microarray analysis. Let us quote Devroye, Gyorfi and Lugosi (Devroye *et al.*, 1996) on leave-one-out estimation: 'One of the drawbacks of the deleted estimate is that it requires much more computation than the resubstitution estimate. Another, and probably more serious, disadvantage of the deleted estimate is its large variance.'

We make a distinction regarding variability affecting error estimation. An error estimator $\hat{\epsilon}$ is a function of the random training data S_n. In some cases, it is also a function of other random factors, such as the random fold partitions used in cross-validation. Therefore, $\hat{\epsilon}$ is a random variable. The variance $\text{Var}[\hat{\epsilon} \mid S_n]$, i.e. the variance due to random factors other than the data sample S_n, is sometimes called the *internal variance* of the estimator (Efron, 1983). For example, in bootstrap estimation, the internal variance is associated with the variance of the sample mean used for approximating the expectation over the bootstrap sample. Resubstitution and leave-one-out have no internal variance. Of greater concern is the full variance of the estimator, which takes into account the variability due to the random sample S_n. This variance is typically much larger than the internal variance. This is an important point, because internal variance has been used previously to compare error estimators (Azuaje, 2003).

A well-known factor in the variability of the error estimators considered here is that they are *error-counting* estimates. Resubstitution and cross-validation count errors committed in n tries and divide the result by n to get the error estimate. Thus, these estimates can only change by $1/n$ increments. In small-sample settings, $1/n$ can be quite large, which creates an irreducible element of variability. Repeated cross-validation and bootstrap are less affected by this problem, because they average over more than n tries (e.g. for bootstrap, nB tries).

Another variability issue affecting cross-validation is the manner in which expression (3) serves as a sample-mean approximation to the unconditional error rate. The difficulty is that the sets $S_n \backslash S_{(i)}$ are not independent samples from $\boldsymbol{F}_{n-n/k}$, and this adds variance to the estimate (Hastie *et al.*, 2001). The problem is more serious for larger k.

So far, we have mostly discussed the variability of error estimators as estimators of the unconditional error rate. This assesses the *global* performance of an error estimator. In practice, the *conditional* error rate (for the given data set) ϵ_n in (1) is of greatest concern. Taking into consideration how far the error estimator is from the conditional error rate assesses the *local* performance of the estimator. Of course, the local and global performance of an error estimator are related. Viewed as estimators of the conditional error rate, resampling estimators, such as cross-validation and bootstrap, have the following issue: performance of the originally designed classifier is assessed in terms of 'surrogate' classifiers, designed by the classification rule applied on reduced data from which samples were left out. If these surrogate classifiers are too different from the original classifier too often, then the estimate may be far from ϵ_n—a similar observation was made in Kohavi (1995), in connection with the stability of classification rules. We call this the 'surrogate problem'. This problem is severely aggravated in small-sample settings; the designed surrogate classifiers may look nothing like the original one. The more sample points left out, and the more complex the classification rule (in particular, the larger the number of bins into which it divides the feature space), the worse is the surrogate problem. Please see the companion web site for plots of three cases corresponding to popular classification rules of increasing complexity, which illustrate the surrogate problem.

In light of the preceding considerations, we propose to study the performance of an error estimator $\hat{\epsilon}$, particularly in small-sample settings, via the distribution of the error $\epsilon_n - \hat{\epsilon}$, where ϵ_n is the true error rate of the designed classifier from each given sample S_n, as in (1). We call this the *deviation distribution* of the error estimator. It is the distribution of the random variable $\epsilon_n - \hat{\epsilon}$, which measures how far an error estimator is from the true error. Note that $\hat{\epsilon}$ is unbiased if and only if the mean on the deviation distribution is zero. Unbiasedness will be of limited utility, however, if the deviation distribution is highly variable, since then there will be a high probability of the estimator being far from the true error for any given sample S_n.

Several statistics of the deviation distribution provide useful error-estimation properties. Estimator bias is reflected by the mean deviation, $E[\epsilon_n - \hat{\epsilon}]$. $\mathrm{Var}[\epsilon_n - \hat{\epsilon}]$ reflects the confidence we can have in our estimates from actual samples. The root-mean square (RMS) error, $\sqrt{E[(\epsilon_n - \hat{\epsilon})^2]}$, combines the effects of both bias and variance: $E[(\epsilon_n - \hat{\epsilon})^2] = E[(\epsilon_n - \hat{\epsilon})]^2 + \mathrm{Var}[\epsilon_n - \hat{\epsilon}]$. Lastly, we consider the quartiles of the deviation distribution, which are less affected by outliers than the mean (these being visualized in our simulations via box plots).

4 SIMULATION STUDY

In this section, we report the results obtained from a large simulation study based on synthetic and real patient data, which measured the performance of resubstitution, several cross-validation estimators, and a pair of bootstrap estimators. The simulations were performed on a 2.5 GHz Pentium 4 computer, running Windows 2000 and Cygwin (a UNIX environment for Windows). The C code that was developed to implement the various error estimators, with full documentation and examples, can be downloaded from the companion web site.

4.1 Experimental setup

We consider in our experiments three classification rules: linear discriminant analysis (LDA), 3-nearest-neighbor (3NN) and decision trees (CART). To improve performance and minimize overfitting in CART, the tree is not fully grown, but splitting stops when there are six points or fewer in a node. The error estimators studied are resubstitution (resub); four variations of cross-validation: leave-one-out (loo), 5-fold cross validation (cv5), 10-fold cross validation (cv10) and repeated cross-validation that averages 10 runs of 10-fold cross validation, while picking the folds randomly each time (cv10r); and two bootstrap estimators, the 0.632 bootstrap (b632) and the bias-corrected bootstrap (bbc). For the computation of cv5, cv10 and cv10r, we use stratified cross-validation, and for computation of b632 and bbc, we use balanced bootstrap samples, with $B = 100$ replicates, which makes the number of designed classifiers be the same as for cv10r.

4.2 Simulation based on synthetic data

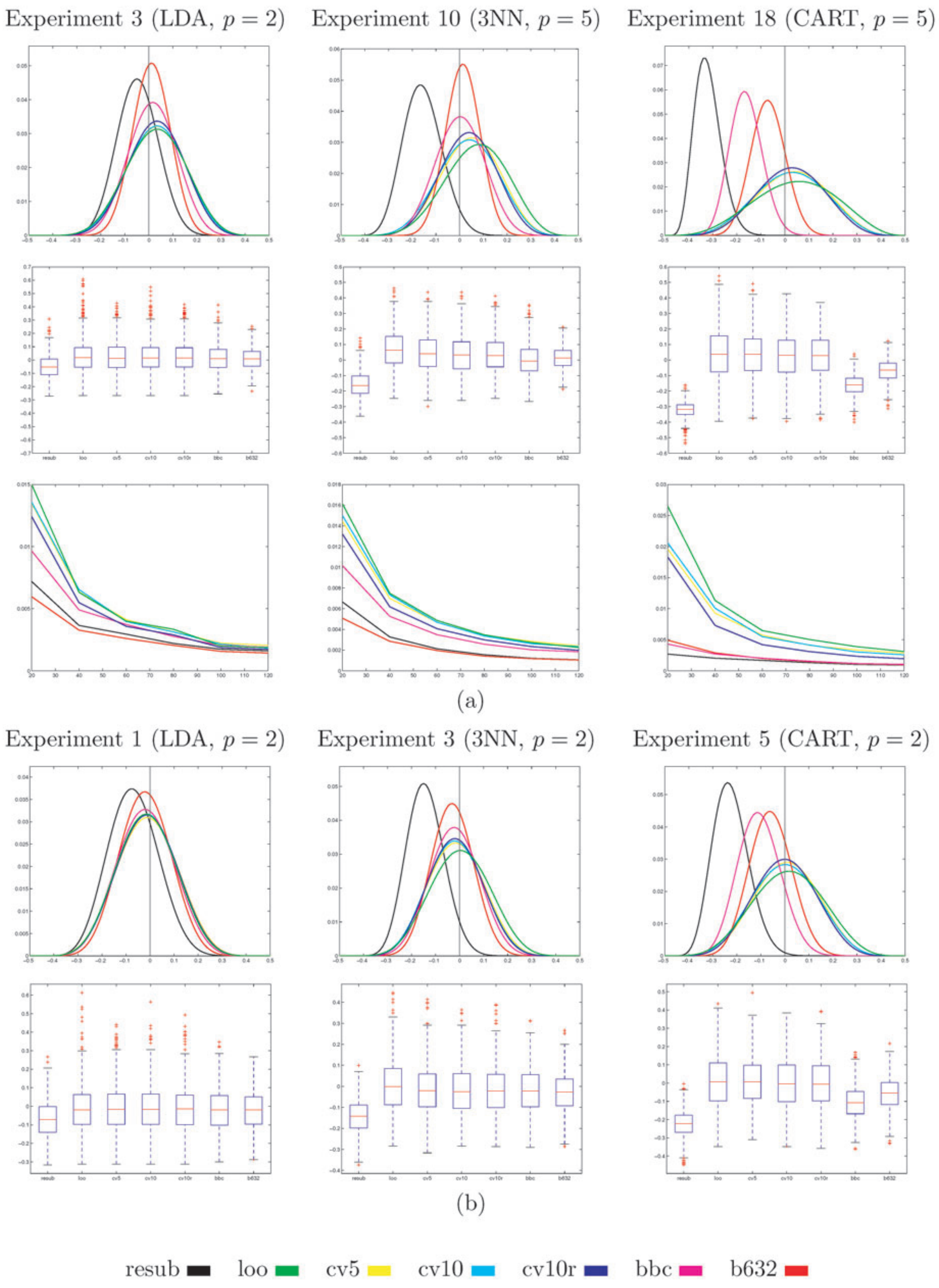
Our catalog of simulations using synthetic data consists of a total of 108 experimental conditions, each involving a thousand replications using different sample data drawn from an underlying model. The model for LDA consists of Gaussian class-conditional densities, with spherical covariances and means located at $(\delta, \ldots, \delta)$ and $(-\delta, \ldots, -\delta)$, where $\delta > 0$ is a separation parameter. The model for 3NN and CART corresponds to class-conditional densities given by a mixture of Gaussians, with spherical covariances and means at opposing vertices of a hypercube centered at the origin and side $2\,\delta$; e.g. in five dimensions, the class-conditional density for class 1 has means at $(\delta, \delta, \delta, \delta, \delta)$ and $(-\delta, -\delta, -\delta, -\delta, -\delta)$, whereas the class-conditional density for class 2 has means at $(\delta, -\delta, \delta, -\delta, \delta)$ and $(-\delta, \delta, -\delta, \delta, -\delta)$. In all cases, we assume equal prior probabilities for each class.

We consider 18 experiments and six sample sizes, varying from 20 to 120 in increments of 20, which make up the total of 108 experimental conditions. The experiments cover the three classification rules, under low ($p = 2$) or moderate ($p = 5$) dimensionality. In each of these six cases, three different choices of class separation and variance are employed, corresponding to a Bayes error of approximately 0.1, 0.15 or 0.2. The parameters for the 18 experiments are summarized in a table that can be accessed on the companion web site.

For each experiment and sample size, we computed the empirical deviation distribution, derived from the 1000 independent draws of the observations. The true error for each observation was computed exactly for LDA, and by Monte-Carlo computation for 3NN and CART. Due to space constraints, we discuss three representative experiments, one for each of the classifiers considered. In addition, we focus for the most part on sample size $n = 20$. The full results for the complete set of experiments can be found on the companion web site. These include tables with the mean, variance and RMS of the deviation distribution, along with the mean and variance of the true error in each case.

The top two rows of Figure 1a display beta-distribution fits and box plots of the empirical deviation distribution for the three selected representative experiments (the full set being on the companion web site). It is seen that all cross-validation estimators perform similarly. As expected, they are slightly high-biased. Their main drawback is high variability—their distributions tend to be rather flat. Thus, they have low probability of being close to the actual classification error. They also tend to produce large outliers, which can lead to severely misleading conclusions. The disparity in performance between cross-validation and the other estimators increases as one goes from LDA to CART; i.e. as the complexity of the classification rule increases. Resubstitution is low-biased, as expected, but shows smaller variance than cross-validation. The overall performance of bias-corrected bootstrapping is slightly better than cross-validation, but in a few experiments cross-validation is better. The 0.632 bootstrap proved to be the best overall estimator in our simulations; however, in the case of CART, where resubstitution is badly low-biased, both bootstrap estimators are low-biased (since the resubstitution estimate is used in their computation).

The best performing cross-validation estimator, by a small margin, is cv10r, but this comes at a steep computational price. In the same vein, the bootstrap estimators show good performance, but their computational cost is very high. The companion web site includes average computation time tables for all experiments and sample sizes. The speed of loo is acceptable for small sample sizes, but it quickly slows down as the sample size increases. The timings of cv5 and cv10 equal that of resubstitution only in the 3NN case, since this is a ‘lazy’ classifier; i.e. no explicit design of the surrogate classifiers



(a)

(b)

Fig. 1. Empirical deviation distribution for selected simulations. (**a**) Synthetic data. Top row: beta fits, $n = 20$. Middle row: box plots, $n = 20$. Bottom row: variance as a function of sample size. (**b**) Patient data. Top row: beta fits, $n = 20$. Bottom row: box plots, $n = 20$.

has to take place. The bootstrap estimators benefit far less from this, because in their case, resampling produces no data reduction. As expected, cv10r takes about 10 times longer to compute than cv10, in all cases. Except in the case of CART, with small sample sizes, the bootstrap estimators take longer to compute than cv10r. Overall, cv10r, bbc and b632 are very slow compared to the other estimators. In a large experiment with thousands of genes, where hundreds of thousands of gene subsets have to be considered for feature extraction, repeated cross-validation and bootstrap estimation, as well as leave-one-out if many samples are considered at a time, can be impractical.

The bottom row of Figure 1a displays the variance of the empirical deviation distribution, plotted as a function of the sample size (analogous mean and RMS curves can be accessed on the companion web site). These curves confirm the facts that resubstitution and the bootstrap estimators have smaller variances than cross-validation estimators, and the discrepancy increases as the sample size becomes smaller. Note that for all cross-validation estimators, the slope of the curves becomes very steep for small sample sizes, indicating rapidly degenerating performance in that case. By analyzing the slopes, we can also determine the approximate sample size beyond which there is no considerable additional improvement in variance. This happens for cross-validation around $n = 100$, whereas for resubstitution and the bootstrap estimators, this value is smaller in the case of 3NN, and much smaller in the case of CART. Note, however, that bias is not taken into consideration in this analysis, which in practice will increase the number of samples needed by resubstitution and the bootstrap estimators. Note also that LDA is the case in which the variance curves are most similar; i.e. LDA is the classification rule that is the most insensitive to the choice of error estimator.

4.3 Simulation based on patient data

To support the preceding findings, we have conducted simulations based on real patient data. These data come from a recently published microarray-based cancer classification study (van de Vijver *et al*., 2002), which analyzes a large number of microarrays, prepared with RNA from breast tumor samples from 295 patients. Using a previously established 70-gene prognosis profile (van't Veer *et al.*, 2002), a prognosis signature based on gene-expression is proposed in van de Vijver *et al*. (2002), which correlates well with patient survival data and other existing clinical measures. Of the 295 microarrays, 115 belong to the 'good-prognosis' class, whereas the remaining 180 belong to the 'poor-prognosis' class.

Our simulation was set up in the following way. We used log-ratio gene expression values associated with the top $p = 2$ and top $p = 5$ genes, as ranked by a correlation-based measure, described in van't Veer *et al.* (2002). In each case, 1000 observations of size $n = 20$ and 40 were drawn independently from the pool of 295 microarrays. Sampling was stratified in the sense that half of the sample points were drawn from each of the two prognosis classes. The true error for each observation of size n was approximated by a holdout estimator, whereby the $295 - n$ sample points not drawn are used as the test set (a very good approximation to the true error, given the large test sample). This allowed us to compute the empirical deviation distribution for each error estimator, using the three classification rules (LDA, 3NN and CART), with either $p = 2$ or 5 genes, which led to a total of six experiments (per sample size). These experiments are summarized in a table on the companion web site.

Note that, as the observations are not independent, there is a degree of inaccuracy in the computation of the deviation distribution. However, for sample sizes $n = 20$ and 40 out of a pool of 295 sample points, the amount of overlap between samples will be small: as can be easily computed, for $n = 20$, the probability of overlap of 3 or fewer points between any two given observations is over 95%, with a mean overlap of 1.425 points; for $n = 40$, the situation degrades a little—the probability of overlap is over 96% for 9 or fewer sample points, with a mean overlap of 5.701 sample points. These numbers mean that, especially with $n = 20$, the observations are only weakly dependent, and the resulting empirical deviation distribution can be considered to be a good approximation to the true deviation distribution.

Statistics, beta fits, box plots and average timings were computed as previously (complete results being on the companion web site). We focus here on the case with the two top genes and $n = 20$. Figure 1b displays plots of the empirical deviation distribution, for these three experiments. It can be seen that the results obtained with the patient data confirm the general conclusions obtained with the synthetic data.

5 CONCLUSION

By considering the empirical deviation distributions computed in a large number of simulations using synthetic and real patient data, we have shown that all cross-validated estimators display undesirable features, such as high variance and large outliers. These undesirable features tend to worsen as the complexity of the classification rule increases. The large outliers produced by cross-validation estimators, especially the loo estimator, mean that severely inaccurate conclusions can be reached for a given data set. Even though cv5 and cv10 have been recommended in the literature as improvements over loo (Hastie *et al.*, 2001), in terms of decreased variance, whereas cv10r has been recommended as the overall estimator of choice (Kohavi, 1995), we have not been able to verify a substantial difference in performance among these estimators. Indeed, the best overall cross-validation estimator appears to be cv10r, but not by much. Moreover, improvement with cv10r comes at a steep computational price. We have found that the bootstrap estimators, in particular the 0.632 estimator, display the best overall performance; but again, a major issue with bootstrap estimators is their high computational cost. Resubstitution, on the other hand, tends to be low-biased, in some cases severely. It is, however, inexpensive, and generally displays lower variability than cross-validation. Any proposed error estimation method must take these points into consideration to be applicable to small-sample microarray classification.

ACKNOWLEDGEMENTS

U.M.B.-N. was supported by the National Cancer Institute, while E.R.D. was supported by the National Human Genome Research Institute and the Translational Genomics Research Institute.

REFERENCES

Azuaje,F. (2003) Genomic data sampling and its effect on classification performance assessment. *BMC Bioinformatics*, **4**, 5.

Chernick,M. (1999) *Bootstrap Methods: A Practitioner's Guide*. Wiley, New York, NY.

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

Dougherty,E.R. (2001) Small sample issues for microarray-based classification. *Comp. Funct. Genom.*, **2**, 28–34.

Efron,B. (1979) Bootstrap methods: another look at the jacknife. *Ann. Stat.*, **7**, 1–26.

Efron,B. (1983) Estimating the error rate of a prediction rule: improvement on cross-validation. *J. Am. Stat. Assoc.*, **78**, 316–331.

Hastie,T., Tibshirani,R. and Friedman,J. (2001) *The Elements of Statistical Learning*. Springer, New York, NY.

Kohavi,R. (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)* Montreal, CA, pp. 1137–1143.

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

van't Veer,L.J., Dai,H., van de Vijver,M.J., He,Y.D., Hart,A.A.M., Mao,M., Peterse,H.L., van der Kooy,K., Marton,M.J., Witteveen,A.T. *et al.* (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**, 530–536.

Vapnik,V.N. (1998) *Statistical Learning Theory*. Wiley, New York, NY.

Witten,I.H. and Frank,E. (2000) *Data Mining*. Academic Press, San Diego, CA.