BIOINFORMATICS ORIGINAL PAPER Vol. 21 no. 7 2005, pages 1046–1054
doi:10.1093/bioinformatics/bti081

Gene expression

Superior feature-set ranking for small samples using bolstered error estimation

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

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

Received on July 22, 2004; revised on September 25, 2004; accepted on September 30, 2004
Advance Access publication October 28, 2004

ABSTRACT

Motivation: Ranking feature sets is a key issue for classification, for instance, phenotype classification based on gene expression. 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.

Results: This paper 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, three-nearest-neighbor classification and classification trees. 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 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 feasible for feature-set ranking when the number of feature sets is extremely large.

Availability: We provide a companion website, which contains the complete set of tables and plots regarding the simulation study, and a compilation of references on feature-set ranking with applications in Genomics. The companion website can be accessed at the URL http://ee.tamu.edu/ ∼edward/bolster_ranking

Contact: edward@ee.tamu.edu

1 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 (see the companion website for a list of papers on this topic).

A critical issue for classification via microarray data is the frequent presence of small samples and the consequences flowing therefrom (Dougherty, 2001). For instance, with small-sample classifier design, one is typically limited to small feature sets to avoid overfitting (Jain and Chandrasekaran, 1982; Raudys and Jain, 1991; Devroye *et al*., 1996). 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 aticulated 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' (Dougherty, 2001). 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 (Kim *et al*., 2002). This 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'

*To whom correspondence should be addressed.

P-value (Liu *et al*., 2002). For spotted cDNA microarrays, the DeArray software of the National Human Genome Research Institute calculates a multi-faceted quality metric for each spot (Chen *et al*., 2002). 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 (Cover and van Campenhout, 1977), or suboptimal. 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. 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 suboptimal method must be employed. 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 suboptimal selection process. We refer to the literature for issues concerning suboptimal feature selection, including small-sample considerations (Jain and Zongker, 1997; Kudo and Sklansky, 2000).

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 (Devroye *et al*., 1996). 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 (Braga-Neto and Dougherty, 2004b). 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 (Braga-Neto *et al*., 2004). 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, 0.632 bootstrap generally outperforms cross-validation methods (the performances of which vary widely), the exception being 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. In this paper we demonstrate that the recently proposed *bolstered error estimation* (Braga-Neto and Dougherty, 2004a) not only outperforms cross-validation for feature-set ranking, but also outperforms 0.632 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 (Jain and Zongker, 1997; Kudo and Sklansky, 2000). 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), three-nearest-neighbor (3NN) classification and classification trees. We present detailed analysis in the paper for one case from each study and provide the bulk of the results on the companion website.

2 ERROR ESTIMATION

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, classifiers are designed 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:

$$\epsilon_n[g|S_n] = P(g(S_n, X) \neq Y) = E_{\mathbf{F}}(|Y - g(S_n, X)|), \qquad (1)$$

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 (1). In practice, one must use an *error estimator*. Ideally, this estimate should be fast to compute and as close as possible to the true error, for the given training data.

2.1 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 (Vapnik, 1998).

Cross-validation removes optimism by using test points not used in classifier design. 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 (Braga-Neto and Dougherty, 2004b; Devroye *et al*., 1996). They can also be slow to compute when the number of folds or samples is large.

Bootstrap error estimation (Efron, 1979, 1983) is based on the notion of an 'empirical distribution' $\mathbf{F}^*$, which replaces the original unknown distribution $\mathbf{F}$. The empirical distribution puts mass $1/n$ 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* (Efron, 1983) is written in terms of the empirical distribution as $\hat{\epsilon}_0 = E_{\mathbf{F}^*}\left(|Y - g(S_n^*, X)| : (X, Y) \in S_n \setminus S_n^*\right)$. 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 $(1 - \mathrm{e}^{-1})n \approx 0.632n$. The *0.632 bootstrap estimator* (Efron, 1983), $\hat{\epsilon}_{\text{b632}} = (1 - 0.632)\,\hat{\epsilon}_{\text{resub}} + 0.632\,\hat{\epsilon}_0$, 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 (Braga-Neto and Dougherty, 2004b).

2.2 Bolstered error estimation

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 $f^{\diamondsuit}(\mathbf{x}) = (1/n)\sum_{i=1}^n f_i^{\diamondsuit}(\mathbf{x} - \mathbf{x}_i)$. The *bolstered resubstitution* estimator (Braga-Neto and Dougherty, 2004a) is obtained by replacing F^* by $F^{\diamondsuit}$ in the definition of $\hat{\varepsilon}_n^R$ to obtain

$$\hat{\varepsilon}_n^{\diamondsuit R} = E_{F^{\diamondsuit}}[|Y - g(S_n, \mathbf{X})|]. \tag{2}$$

Whereas (Braga-Neto and Dougherty, 2004a) treated the definitions, properties and comparisons of bolstering with classical error estimation relative to variance and deviation from the true error, this paper considers the ability of bolstering to provide accurate feature-set ranking.

A computational expression for the bolstered resubstitution estimator is given by

$$\hat{\varepsilon}_n^{\diamondsuit R} = \frac{1}{n}\sum_{i=1}^n \left(I_{y_i=0} \int_{A_1} f_i^{\diamondsuit}(x - x_i)\,\mathrm{d}x + I_{y_i=1} \times \int_{A_0} f_i^{\diamondsuit}(x - x_i)\,\mathrm{d}x \right). \tag{3}$$

where $A_j = \{x | g(S_n, x) = j\}$. 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:

$$\hat{\varepsilon}_n^{\diamondsuit R} \approx \frac{1}{n}\sum_{i=1}^n \left(\sum_{j=1}^M I_{x_{ij} \in A_1} I_{y_i=0} + \sum_{j=1}^M I_{x_{ij} \in A_0} I_{y_i=1} \right), \tag{4}$$

where $\{x_{ij}\}_{j=1,\ldots,M}$ are samples drawn from the distribution $f_i^{\diamondsuit}$. The experiments in Braga-Neto and Dougherty (2004a) 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). Figure 1 illustrates the situation where the bolstering kernels are given by uniform circular distributions and the classifier is linear. In this case, 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 incorrectly classified data points, as that increases optimism. Bias is reduced by using no bolstering for incorrectly classified points. The result is the *semi-bolstered resubstitution* estimator (Braga-Neto and Dougherty, 2004a).

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 $A_j^i = \{x | g(S_{n-1}^i, x) = j\}$, for $j = 0, 1$, be the decision region for the classifier designed from S_{n-1}^i. The *bolstered leave-one-out* estimator (Braga-Neto and Dougherty, 2004a) can be

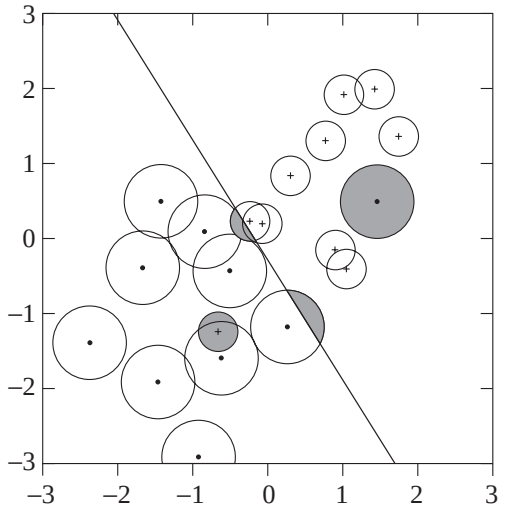

Fig. 1. Bolstered resubstitution for a linear classifier, assuming uniform circular bolstering kernels. The area of each shaded region divided by the area of the associated circle is the error contribution made by a point. The bolstered resubstitution error is the sum of all contributions divided by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

computed via

$$\hat{\epsilon}^{\diamond}_{\text{loo}} = \frac{1}{n}\sum_{i=1}^{n}\left(I_{y_i=0}\int_{A_1^i} f_i^{\diamond}(x-x_i)\,\mathrm{d}x \; + I_{y_i=1} \times \int_{A_0^i} f_i^{\diamond}(x-x_i)\,\mathrm{d}x \right). \tag{5}$$

When the integrals cannot be computed exactly, a Monte-Carlo expression like (4) can be used.

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 $\sigma_1, \ldots, \sigma_n$. These parameters determine the variance and bias properties of the 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_is, 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 critical. A non-parametric sample-based method to choose these parameters that is applicable in small-sample settings has been proposed (Braga-Neto and Dougherty, 2004a). The method is described in the Appendix, which also describes in detail bolstering using Gaussian kernels, the kind used in this paper.

3 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 (Braga-Neto *et al*., 2004), 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 (Ambroise and McLachlan, 2002). 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,

$$R_1^K(t) = \sum_{k=1}^{K} I_{k^* \le K}. \tag{6}$$

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 to 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 feature sets,

$$R_2^K(t) = \frac{1}{K}\sum_{k=1}^{K} |k - k^*|. \tag{7}$$

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 Braga-Neto *et al*. (2004). Our interest here is broader. We interested in a wide variety of error estimators and 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.

4 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 breast-cancer patient prognosis. In both cases we consider three classification rules: LDA, 3NN, and classification and regression trees (CART). In all cases we consider a sample size of 30, do the analysis for two and three features, and consider top lists of sizes $K = 20$ and 40. We provide detailed analysis in the paper for one Gaussian case and one from the breast-cancer data. The results for all other cases appear on the companion website.

4.1 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 $a = (a_1, a_2, \ldots, a_n)$ is a parameter vector with $||a|| = 1$. The Bayes classifier is a hyperplane perpendicular to the axis joining the means, with Bayes error $\epsilon_{\text{BAYES}} = 1 - \Phi(\delta)$, where Φ is the standard normal cumulative distribution function. Since $\delta = \Phi^{-1}\left(1 - \epsilon_{\text{BAYES}}\right)$, 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, $\epsilon^L_{\text{BAYES}} = 1 - \Phi(\delta \sqrt{\sum_{k \in L} a_k^2})$. To minimize $\epsilon^L_{\text{BAYES}}$ 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 $a = (a_1, a_2, \ldots, a_n)$ 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 sample sets 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 compute the 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.

Figure 2a provides the mean $R_1^{40}(t)$ 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. The companion website contains the complete set of plots for all cases. In addition, the plots on the companion website include curves for 5-fold cross-validation, 10-fold cross-validation and the bias-corrected bootstrap (bbc) (Efron, 1983). These have been left out in the paper for clarity of the graphical results (generally, cv10r outperforms cv5 and cv10, sometimes significantly, cv5 and cv10 outperforms loo, and the performances of b632 and bbc are often comparable, with b632 usually providing slightly better performance). Each plot in Figure 2 assumes a range of maximum true error threshold $t = 0.25$ through $t = 0.50$. Table 1 shows two statistics for a few values of t: s_1 is the average error for all feature sets having error $< t$, which is the average error among those feature sets for which the performance statistics have been computed, and s_2 is the average number of feature sets having error $< t$. A third statistic s_3 (not shown in the table) gives the number of sample sets for which there are at least K feature sets having error $< t$, which is the number of sample sets over which the performance statistics have been averaged. For this statistic, please see the companion web site.

For LDA, Figure 2a shows that bolstered resubstitution performs best over the entire range of t, with the other bolstered estimators also performing better than the 0.632 bootstrap. Both cross-validation estimators, loo and cv10r, perform about the same as resubstitution, with the latter three all performing much worse than the 0.632 bootstrap. On the companion website 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 0.632 bootstrap, and the latter will discover two more than loo or the heavily computational cv10r, neither of which perform substantially better than resubstitution. Figure 2a shows that the pattern shown 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 0.632 bootstrap in terms of feature discovery, it does so with much less computation time. Table 2a provides computation times for the error estimators in this experiment, with all values normalized with respect to the resubstitution timing (meaning that, relative to the table, the resubstitution timing is 1 in each case). Tables for all experiments are given on the companion website.

For 3NN, Figure 2a 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 0.632 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 Braga-Neto and Dougherty (2004) 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 0.632 bootstrap.

For CART, Figure 2a shows that bolstered and semi-bolstered resubstitution significantly outperform 0.632 bootstrap, with bolstered-leave-one-out slightly outperforming 0.632 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 $R_2^{40}(t)$.

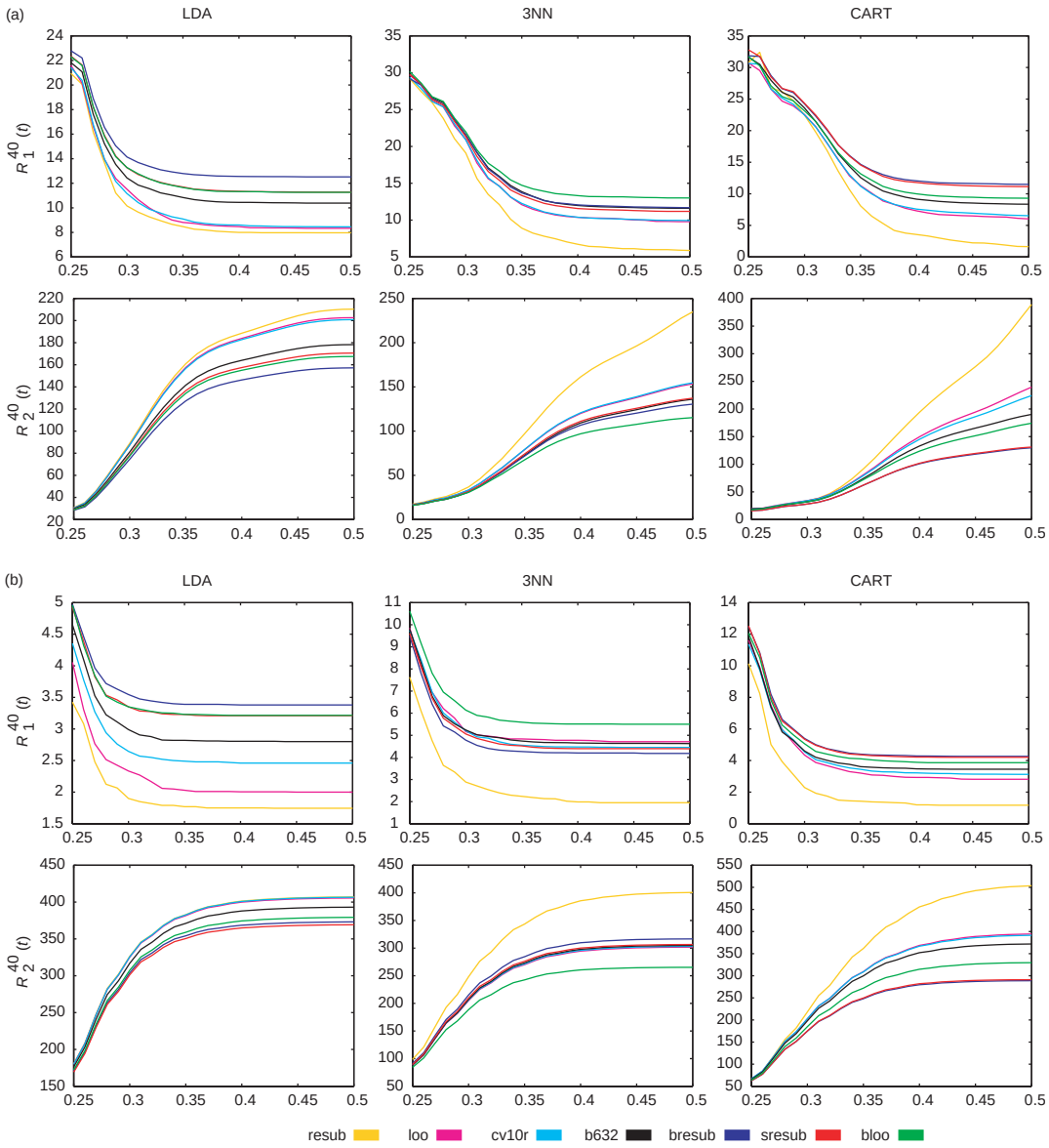

Fig. 2. Mean ranking statistics versus maximum true error threshold, for (**a**) the synthetic data, in the equal-variance case and feature sets of size 3, and (**b**) the patient data, for feature sets of size 3.

4.2 Patient data

We have conducted experiments based on patient data from a microarray-based classification study (van de Vijver *et al*., 2002) that analyzes 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) that correlates 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 van't Veer *et al*. (2002). 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 Braga-Neto and Dougherty (2004a), the samples are only weakly dependent.

The results corresponding to Figure 2a are shown in Figure 2b for the patient data experiments, with feature sets of size 3. The associated sample information and computation times are given in Tables 1b and 2b, respectively. The companion website contains the other case (feature sets of size 2), as well as curves for 5-fold cross-validation, 10-fold cross-validation and the bias-corrected bootstrap.

Table 1. Two statistics for a few values of the maximum true error threshold t in Figure 2a, 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: s_1 is the average true error for all feature sets having error $< t$, while s_2 is the average number of feature sets having true error $< t$

	LDA		3NN		CART	
t	s_1	s_2	s_1	s_2	s_1	s_2
(a)						
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
(b)						
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

Table 2. 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

	loo	cv10r	b632	bresub	sresub	bloo
(a)						
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
(b)						
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

The values are relative to the resubstitution timing.

The trends regarding bolstering, bootstrap and cross-validation observed in the Gaussian model are closely reflected in the patient data. 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 is to see if bolstering would still prove superior to bootstrap and cross-validation in such a difficult scenario. Our results indicate it does so.

5 CONCLUSION

The results demonstrate, for the three classification rules and the data sets considered, that bolstering is superior to bootstrap, and bootstrap is better than cross-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 data sets presented and that more work is needed to determine the extent of the superiority of bolstering with regard to ranking. We mention two potential directions of future work. First, one could try to discover theoretical results regarding the comparison of error estimators for ranking. Given the paucity of such results to date, and the highly specialized hypotheses that theoretical results would likely require, a theoretical approach might not lead to a wide understanding of applicability. A second approach would be to find counterexamples where the trends of the current paper do not hold and to try to explain the reasons behind the varying behavior. This could lead to a broader set of conclusions that would state exactly what kind of classification problems can be expected to behave as those studied in the current paper.

ACKNOWLEDGEMENTS

Braga-Neto's research was supported by a contract from the National Cancer Institute (N01–CN15102), and Dougherty's research was supported by the Translational Genomics Research Institute.

REFERENCES

Ambroise,C. and McLachlan,G.J. (2002) Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc. Natl Acad. Sci. USA*, **99**, 6562–6566.

Braga-Neto,U.M. and Dougherty,E.R. (2004a) Bolstered error estimation. *Pattern Recogn.*, **37**, 1267–1281.

Braga-Neto,U.M. and Dougherty,E.R. (2004b) Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, **20**, 374–380.

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

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

Cover,T. and van Campenhout,J. (1997) On the possible orderings in the measurement selection problem. *IEEE Trans. Systems Man Cybernet.*, **7**, 657–661.

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

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

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

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

Jain,A.K. and Chandrasekaran,B. (1982) Dimensionality and sample size considerations in pattern recognition practice. In Krishnaiah,P.R. and Kanal,L.N. (eds), *Handbook of Statistics*. North-Holland, Amsterdam, Vol. II, pp. 835–855.

Jain,A.K. and Zongker,D. (1997) Feature selection—evaluation, application, and small sample performance. *IEEE Trans. Pattern Anal. Machine Intell.*, **19**, 153–158.

Kim,S., Dougherty,E.R., Shmulevich,I., Hess,K.R., Hamilton,S.R., Trent,J.M., Fuller,G.N. and Zhang,W. (2002) Identification of combination gene sets for glioma classification. *Mol. Cancer Therap.*, **1**, 1229–1236.

Kudo,M. and Sklansky,J. (2000) Comparison of algorithms that select features for pattern classifiers. *Pattern Recogn.*, **33**, 25–41.

Liu,W.-M., Mei,R., Di,X., Ryder,T.B., Hubbell,E., Dee,S., Webster,T.A., Harrington,C.A., Ho,M.-H., Baid,J. and Smeekens,S.P. (2002) Analysis of high density expression microarrays with signed-rank call algorithms. *Bioinformatics*, **18**, 1593–1599.

Raudys,S.J. and Jain,A.K. (1991) Small sample size effects in statistical pattern recognition: recommendations for practitioners. *IEEE Trans. Pattern Anal. Machine Intell.*, **13**, 252–262.

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. *N. Engl. 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.

6 APPENDIX

The appendix describes how to choose the amount of bolstering and it describes in detail the special situation of bolstering using Gaussian kernels, the kind used in this paper.

6.1 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 (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:

$$\hat{d}(y) = \frac{\sum_{i=1}^{n} \min_{j \neq i} \{||x_i - x_j||\} : I_{y_i = y}}{\sum_{i=1}^{n} I_{y_i = y}}. \tag{8}$$

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 $\alpha_{p,i} = F_{D_i}^{-1}(1/2)$ 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 $\sigma_i = \hat{d}(y_i)/\alpha_{p,i}$, for $i = 1, \ldots, n$. As the sample size 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 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 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

$$\sigma_i = \frac{d(x_i, S_{n-1}^i)}{\alpha_{p,i}}, \quad \text{for } i = 1, \ldots, n, \tag{9}$$

6.2 Gaussian-bolstered error estimation

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

$$f_i^{\diamond}(x) = \frac{1}{(2\pi)^{p/2}\sigma_i^p} \exp\left(-\frac{||x||^2}{2\sigma_i^2}\right). \tag{10}$$

For a general classifier, the integrals in (3) and (5) have to be computed by Monte-Carlo sampling. For a linear classifier, however, analytical expressions are possible. For example, for LDA, the Gaussian-bolstered resubstitution error estimate is given by (see Braga-Neto and Dougherty, 2004a for a proof):

$$\hat{\epsilon}_{\text{resub}}^{\diamond} = \frac{1}{n} \sum_{i=1}^{n} \left(\Phi_{\sigma_i}(W_a(x_i)) I_{y_i=0} + \Phi_{\sigma_i}(-W_a(x_i)) I_{y_i=1}\right), \tag{11}$$

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 $a = \Sigma^{-1}(\mu_1 - \mu_0)$ and

$$m = \tfrac{1}{2}(\mu_0 + \mu_1)^T \Sigma^{-1}(\mu_0 - \mu_1). \tag{12}$$

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 (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. If $\sigma_i \to \infty$, for $i = 1, \ldots, n$, then Φ_{σ_i} becomes 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 not useful.

The actual values of σ_i in a practical situation are computed according to the distance-based scheme outlined in the previous section. 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

$$f_D(x) = \frac{2^{1-p/2} x^{p-1} e^{-x^2/2}}{\Gamma(\frac{p}{2})}, \quad (13)$$

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 Equation (13), 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: $\alpha_1 = 0.674$, $\alpha_2 = 1.177$, $\alpha_3 = 1.538$, $\alpha_4 = 1.832$ and $\alpha_5 = 2.086$.