Relationship between the accuracy of classifier error estimation and complexity of decision boundary

Esmaeil Atashpaz-Gargari a, Chao Sima b, Ulisses M. Braga-Neto a,e, Edward R. Dougherty a,b,c

a Department of Electrical and Computer Engineering, Texas A&M University, United States
b Computational Biology Division, Translational Genomics Research Institute, Phoenix, AZ, United States
c Department of Bioinformatics and Computational Biology, University of Texas M.D. Anderson Cancer Center, Houston, TX, United States

A R T I C L E  I N F O

Article history:
Received 13 February 2012
Received in revised form 22 October 2012
Accepted 27 October 2012
Available online 6 November 2012

Keywords:
Error estimation
Distribution complexity
Small samples
Complexity of decision boundary

A B S T R A C T

Error estimation is a crucial part of classification methodology and it becomes problematic with small samples. We demonstrate here that the complexity of the decision boundary plays a key role on the performance of error estimation methods. First, a model is developed which quantifies the complexity of a classification problem purely in terms of the geometry of the decision boundary, without relying on the Bayes error. Then, this model is used in a simulation study to analyze the bias and root-mean-square (RMS) error of a few widely used error estimation methods relative to the complexity of the decision boundary: resubstitution, leave-one-out, 10-fold cross-validation with repetition, 0.632 bootstrap, and bolstered resubstitution, in two- and three-dimensional spaces. Each estimator is implemented with three classification rules: quadratic discriminant analysis (QDA), 3-nearest-neighbor (3NN) and two-layer neural network (NNet). The results show that all the estimation methods lose accuracy as complexity increases.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Since the scientific content of any model depends on its predictive capacity, the most important attribute of any classifier is its error, being the probability of misclassification. Since the feature-label distribution is generally unknown when designing a classifier, its error must be estimated by an error-estimation rule, so that the validity of the classifier model, consisting of both the classifier and its error, depends on the accuracy of the error-estimation rule. If the sample is large, one can split the data into training and test data, design the classifier on the training set, and estimate its error on the test set. In this case, there exists a distribution-free bound on the root mean square (RMS) error of the error estimator, namely, $\text{RMS} \leq 1/\sqrt{m}$, where $m$ is the number of data points in the test set [1]. When the sample is small, splitting the data results in poor classifier design, so that data-efficient error estimators must be used, i.e., error estimators that operate on the same data used for designing the classifier. Given an error-estimation rule, an obvious question concerns when it performs well, meaning that it has an acceptable RMS. This depends strongly on the error-estimation rule, the classification rule used to design the classifier, the feature-label distribution, and the sample size.

In the present paper, we quantify the effect of distribution complexity on the RMS, for several data-efficient popular error estimators, including resubstitution, leave-one-out, cross-validation, bootstrap, and bolstering. Several classification rules are considered: Quadratic Discriminant Analysis (QDA), 3-nearest-neighbor (3NN) and neural networks (NNet). The analysis in this paper extends to the error estimation problem some of the ideas in [2], where the true classification error was studied as a function of the complexity of the feature-label distribution and of the sample size.

The issue studied in this paper is critical to experimental design. If one makes no modeling assumptions with small-sample classifier design, then, virtually nothing can be said about the error of the designed classifier and hence nothing can be said about the scientific content of the classifier—it is epistemologically vacuous [3]. On the other hand, in defining distributional complexity we want to differentiate between complexity and separability of the classes. Specifically, we want a measure of complexity that will not be related to the Bayes error but will be related to the complexity of the Bayes decision boundary. The classes may be multimodal, with different “modes” being highly interwoven in Euclidean space, but without overlap among the class-conditional densities. In this case the Bayes error will be zero, but the Bayes classifier may possess a complex decision boundary. Despite the fact that, in principle, such a situation involves perfectly separable classes, it presents one, in practice, with a difficult problem for both classifier design and error estimation.
Our interest here is not with error-estimation RMS as a function of Bayes error, but as a function of distribution complexity, and thus complexity of the decision boundary, and our proposed definition of complexity reflects this fact.

This paper is organized as follows. Section 2 describes in detail the Beta Mixture Model (BMM) used in the paper. Section 3 presents the simulation study results. Section 4 summarizes the conclusions of the study. A method to compute the complexity of a distribution and a review of all the error estimators used in the paper are found in the Appendices.

2. Model for distributional complexity

For clarity, we describe the model in two dimensions. The generalization of the model to three or more dimensions is straightforward. To begin, we define the configuration matrix

\[
H = \begin{bmatrix}
    h_{11} & h_{12} & \cdots & h_{1m} \\
    h_{21} & h_{22} & \cdots & h_{2m} \\
    \vdots & \vdots & \ddots & \vdots \\
    h_{m1} & h_{m2} & \cdots & h_{mm}
\end{bmatrix},
\]

where \( h_{ij} \in (0,1), \) and \( i,j \in \{1,2,\ldots,m\}. \) Next, we consider the set of two-dimensional square cells of side \( \ell > 0, \)

\[
D_{ij} = [(i-1)\ell, i\ell] \times [(j-1)\ell, j\ell], \quad i,j \in \{1,2,\ldots,m\}.
\]

On each cell \( D_{ij}, \) we consider the joint density \( f_{ij} \) of two independent, identically distributed Beta random variables with parameters \( a > 0, b > 0 \)

\[
f_{ij}(x_{1}, x_{2}) = \frac{1}{C} (x_{1} - (i-1)\ell)^{a-1} (x_{2} - (j-1)\ell)^{b-1},
\]

for \( (x_{1}, x_{2}) \in D_{ij}, \) with \( f_{ij}(x_{1}, x_{2}) = 0 \) for \( (x_{1}, x_{2}) \notin D_{ij}, \) where \( C > 0 \) is a normalization constant to make the density integrate to 1. Now, for any given matrix \( H, \) let \( R_{k} = \{(i,j) | h_{ij} = k\} \) for \( k = 0,1. \) We define the class conditional density \( f_{ij}(x_{1}, x_{2} | k) \) for our model as the mixture

\[
f_{ij}(x_{1}, x_{2} | k) = \frac{1}{|R_{k}|} \sum_{(i,j) \in R_{k}} f_{ij}(x_{1}, x_{2}), \quad k = 0,1.
\]

The feature-label distribution is given by the combination of the two class-conditional densities: \( f_{ij}(x_{1}, x_{2}) = c_{0} f_{ij}(x_{1}, x_{2} | 0) + (1 - c_{0}) f_{ij}(x_{1}, x_{2} | 1), \) where \( c_{0} \) is the prior probability of class 0. We refer to this model as a Beta Mixture Model (BMM). An example is shown in Fig. 1.

For any given matrix \( H, \) and corresponding feature-label distribution \( f_{ij}(x_{1}, x_{2}), \) the Bayes error is zero, and the data produced by the model are perfectly separable. This is crucial to our approach, as we want the complexity of the distribution to arise exclusively from the complexity of the Bayes decision boundary, which for this model consists of a combination of line segments and rays. See Fig. 2 for an illustration. We define the distributional complexity \( \chi(H) \) to be the total number of the line segments and rays in the Bayes decision boundary. For example, in Fig. 2, the distributional complexity is \( \chi(H) = 5. \) A simple method to compute the complexity for any given configuration is presented in Appendix A.

The collection of all configurations of a given complexity \( \chi \) will be denoted by \( M_{\chi}. \) For example, with \( \chi = 1, \) i.e., a Bayes decision boundary consisting of a single line, we have

\[
M_{1} = \begin{Bmatrix}
    \begin{bmatrix}
        1 & 1 & 1 \\
        0 & 0 & 0
    \end{bmatrix}, \begin{bmatrix}
        1 & 0 & 0 \\
        0 & 0 & 1
    \end{bmatrix}, \begin{bmatrix}
        0 & 0 & 0 \\
        1 & 0 & 0
    \end{bmatrix}, \begin{bmatrix}
        0 & 0 & 1 \\
        1 & 0 & 0
    \end{bmatrix}, \begin{bmatrix}
        0 & 1 & 1 \\
        0 & 1 & 1
    \end{bmatrix}, \begin{bmatrix}
        0 & 1 & 0 \\
        1 & 1 & 0
    \end{bmatrix}, \begin{bmatrix}
        1 & 1 & 1 \\
        0 & 1 & 0
    \end{bmatrix}, \begin{bmatrix}
        1 & 1 & 0 \\
        0 & 1 & 0
    \end{bmatrix}, \begin{bmatrix}
        0 & 1 & 1 \\
        1 & 0 & 0
    \end{bmatrix}
\end{Bmatrix}
\]

Each configuration in \( M_{1} \) leads to distinct classification problem and decision boundary; however, from the perspective of the difficulty of classifier design or error estimation, complementation of the labels 0 and 1 and rotation produce equivalent configurations. This defines an equivalence relation, and our concern is with the equivalence classes \( \hat{M}_{\chi}. \) For example, taking the standard approach of listing a single representative of each equivalence class, for \( \chi = 1, 2, 3 \) we have

\[
\hat{M}_{1} = \begin{Bmatrix}
    \begin{bmatrix}
        1 & 1 & 1 \\
        0 & 0 & 0
    \end{bmatrix}, \begin{bmatrix}
        0 & 0 & 0 \\
        1 & 1 & 1
    \end{bmatrix}, \begin{bmatrix}
        0 & 0 & 1 \\
        0 & 1 & 1
    \end{bmatrix}, \begin{bmatrix}
        0 & 1 & 1 \\
        1 & 0 & 1
    \end{bmatrix}, \begin{bmatrix}
        1 & 0 & 0 \\
        1 & 1 & 0
    \end{bmatrix}, \begin{bmatrix}
        1 & 0 & 1 \\
        1 & 1 & 0
    \end{bmatrix}, \begin{bmatrix}
        0 & 1 & 0 \\
        1 & 1 & 0
    \end{bmatrix}, \begin{bmatrix}
        0 & 1 & 1 \\
        1 & 1 & 0
    \end{bmatrix}
\end{Bmatrix}
\]

\[
\hat{M}_{2} = \begin{Bmatrix}
    \begin{bmatrix}
        0 & 0 & 0 \\
        0 & 0 & 0
    \end{bmatrix}, \begin{bmatrix}
        0 & 0 & 1 \\
        0 & 1 & 1
    \end{bmatrix}, \begin{bmatrix}
        0 & 1 & 0 \\
        1 & 1 & 1
    \end{bmatrix}, \begin{bmatrix}
        0 & 1 & 1 \\
        1 & 1 & 1
    \end{bmatrix}, \begin{bmatrix}
        1 & 0 & 0 \\
        1 & 1 & 0
    \end{bmatrix}, \begin{bmatrix}
        1 & 0 & 1 \\
        1 & 1 & 0
    \end{bmatrix}, \begin{bmatrix}
        1 & 1 & 0 \\
        1 & 1 & 1
    \end{bmatrix}, \begin{bmatrix}
        1 & 1 & 1 \\
        1 & 1 & 1
    \end{bmatrix}
\end{Bmatrix}
\]

\[
\hat{M}_{3} = \begin{Bmatrix}
    \begin{bmatrix}
        0 & 0 & 0 \\
        0 & 0 & 0
    \end{bmatrix}, \begin{bmatrix}
        0 & 0 & 0 \\
        0 & 0 & 0
    \end{bmatrix}, \begin{bmatrix}
        0 & 0 & 0 \\
        0 & 0 & 0
    \end{bmatrix}, \begin{bmatrix}
        0 & 0 & 0 \\
        0 & 0 & 0
    \end{bmatrix}, \begin{bmatrix}
        0 & 0 & 0 \\
        0 & 0 & 0
    \end{bmatrix}, \begin{bmatrix}
        0 & 0 & 0 \\
        0 & 0 & 0
    \end{bmatrix}, \begin{bmatrix}
        0 & 0 & 0 \\
        0 & 0 & 0
    \end{bmatrix}, \begin{bmatrix}
        0 & 0 & 0 \\
        0 & 0 & 0
    \end{bmatrix}
\end{Bmatrix}
\]

3. Simulation study

Based on the Beta Mixture Model and definition of complexity proposed in the previous section, we carry out a detailed simulation study to evaluate the impact of distributional complexity on the performance of different error estimation methods. We consider dimensionality \( p = 2, 3. \) In the case \( p = 2 \) we obtain...
results for each of the $2^8-2$ possible $3 \times 3$ configurations (leaving out complements and trivial cases), whereas in the case $p=3$, results are obtained for a random sampling of the possible $3 \times 3 \times 3$ configurations at each level of complexity. In all cases, the cells have size $\ell = 1$, and the beta parameters are set to $\alpha = \beta = 1.5$. Results for other choices of parameters are similar, and can be found on the companion website at http://gsp.tamu.edu/Publications/supplementary/atashpaz12a. Three classification methods are employed: Quadratic Discriminant Analysis (QDA), 3-Nearest-Neighbors (3NN) and two-layer Neural Networks (NNet). The error estimators considered in this study are resubstitution (resub), 10-fold cross-validation with repetition (cv10r), leave-one-out (loo), bootstrap 0.632 (boot), and bolstered resubstitution (bolstrd). Following the recommendation of [4], we utilize in the 3NN case a variation of bolstering, called semi-bolstered resubstitution, whereas for QDA and NNet, the standard bolstered resubstitution estimator is employed. A description of all error estimators used in this study is given in Appendix B.

The resubstitution estimator is only plotted in the case $p=2$, as its bias becomes disproportionate in the case $p=3$, making the differences among the other estimators difficult to visualize if they are plotted together. We present here results for sample size $n=60$; in addition, the companion website contains results for $n=120$, which were observed to be similar, with the exception that true classification errors are smaller, as expected. The number of Monte-Carlo runs is set to 10,000 in each experiment. Based on this large number of repetitions, the expected true error of each classification rule is computed along with the bias and RMS of the several error estimators; these are used to analyze their performance as a function of distributional complexity.

### 3.1. Expected true error

Before analyzing the performance of the error estimators, it is worth considering briefly the behavior of the true classification error. Fig. 3 displays the true expected error of the various classification rules as a function of distributional complexity. As we see in the plots, the expected true error monotonically increases as the complexity increases. All the classification rules show good performance for low-complexity models. Interestingly, QDA outperforms 3NN and NNet for low-complexity models, but its performance quickly degrades as complexity increases. This occurs because the simple structure of QDA makes it unsuitable for application to complex models. Across moderate and large complexities, 3NN is the best classification rule. This occurs because, compared to QDA, 3NN is more flexible and can create decision boundaries of different complexities. On the other hand, although NNet is capable of constructing complex decision boundaries, it requires large sample sizes for training, and this requirement can be seen to increase sharply with larger distributional complexity.

### 3.2. Performance of error estimators

Here we analyze the performance of the different error estimators, in terms of bias and RMS, as a function of distributional complexity, for the 3NN, QDA, and NNet classification rules.

#### 3.2.1. 3NN

Fig. 4 displays the bias and RMS of the different error estimators for the 3NN classification rule. As mentioned previously, resubstitution is not plotted in the case $p=3$ due to its disproportionate bias. We can see that performance degrades as distributional complexity increases. The RMS plot shows that semi-bolstered resubstitution outperforms the other error estimation methods across the entire complexity range, followed closely by bootstrap. The cross-validation estimators cv10r and loo show acceptable performance for lower complexities. The worst error estimation method is resub, due to its bias, which rapidly increases as the complexity increases. Nevertheless, for low complexities, where the decision boundary is simple, the RMS of resub and the cross-validation estimators are essentially equal.

#### 3.2.2. QDA

Fig. 5 displays the bias and RMS of the different error estimation methods for the QDA classification rule. Once again, we observe that the performance of all error estimators become worse as distributional complexity increases. The bolstered resubstitution error estimator outperforms the others, followed closely by bootstrap, with cross-validation estimators lagging behind, even though the latter have the least bias. Resubstitution is extremely optimistically biased, which makes its RMS largest.

#### 3.2.3. NNet

Fig. 6 displays the bias and RMS of the different error estimation methods for the NNet classification rule. The general trends described earlier hold. Of note however is the very large bias displayed by resubstitution, which occurs due to the fact that NNet is more prone to overfitting than 3NN and QDA.

### 4. Conclusion

In this paper, we develop a model for distributional complexity and study the behavior of different error estimation methods as a function of complexity for three different classification rules using the proposed model. The model is based on a mixture of beta distributions with a Bayes error of zero, so that only the complexity of the decision boundaries comes into play. Simulation results show that the
increase of distributional complexity leads to increasing degradation in error estimation performance. The best error estimator according to RMS is observed to be bolstered resubstitution, across the entire range of complexities considered, followed closely by bootstrap, with cross-validation estimators lagging behind. Resubstitution tends to present a dramatic increase in bias with increasing complexity, making this estimator unsuitable for complex distributions.

It has been observed and analytically demonstrated that cross-validation error estimation performance degrades with increasing Bayes error, in particular, for Gaussian class-conditional densities [5,6]. The difference between degradation owing to increasing Bayes error and increasing distributional complexity can be clearly seen by considering optimal classification in the Gaussian model where both classes share the identity covariance matrix. In such a case, the Bayes decision boundary is a hyperplane equidistant between the means of the class-conditional densities. Moving the means closer at the same rate for both (the limit being Bayes error 0.5), the Bayes hyperplane remains fixed but the performance of cross-validation degrades. This
corresponds to decreasing estimation performance while keeping the distributional complexity fixed at 1. In our proposed model, the Bayes error remains constant at 0, but the Bayes decision boundary increases in complexity. Clearly, the decreasing estimation precision is a consequence of very different factors in the two scenarios.

Acknowledgments

The authors acknowledge the support of the National Science Foundation, through NSF awards CCF-0845407 (Braga-Neto) and CCF-0634794 (Dougherty), as well as Texas A&M AgriLife Research.

Appendix A. Complexity computation

As mentioned in Section 2, in the $p=2$ case, the complexity of the distribution is defined to be the total number of lines, line segments and rays in the Bayes optimal piecewise linear classifier. In the $p=3$ case, the Bayes piecewise linear classifier will be a combination of planes, half-planes, quarter planes and plane segments.

This can be extended to obtain the definition of complexity for general dimensionality $p \geq 2$ as follows. Let $H$ be a $p$-dimensional configuration matrix of size $m \times m \times \cdots \times m$. Let $L^i_j$ be a $(p-1)$-dimensional matrix obtained from $H$ by fixing the $i$-th coordinate at value $j$, where $i \in \{1, \ldots, p\}$ and $j \in \{1, \ldots, m\}$. Define $(p-1)$-dimensional matrices $D^{i}_{j,j+1}$ by

$$D^{i}_{j,j+1} = L^i_j - L^i_{j+1},$$

where $i = 1, \ldots, p$ and $j = 1, \ldots, m-1$. The complexity $\chi(H)$ is the total number of connected components of nonzero terms in the $p \times (m-1)$ matrices $D^{i}_{j,j+1}$, where two cells are defined to be connected if they share a “face” in $(p-1)$-dimensional space.

These definitions can be best understood by means of examples in the $p=2$ and $p=3$ cases. We will assume in both cases that $m=3$, as in the examples given in Section 2.

![Fig. A1](image-url)  
Fig. A1. The process of finding the complexity $\chi(H)$ in the case $p=2$. Here, $\chi(H)=5$.

A.1. 2D case

In the $p=2$ case, given a configuration matrix

$$H = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,1} & h_{2,2} & h_{2,3} \\ h_{3,1} & h_{3,2} & h_{3,3} \end{bmatrix},$$

the matrices $D^{1}_{1,2}, D^{1}_{2,3}, D^{2}_{1,2}$ and $D^{2}_{2,3}$ are given by

$$D^{1}_{1,2} = [h_{1,1} - h_{2,1}, h_{1,2} - h_{2,2}, h_{1,3} - h_{2,3}],$$

$$D^{2}_{1,2} = [h_{2,1} - h_{3,1}, h_{2,2} - h_{3,2}, h_{2,3} - h_{3,3}],$$

$$D^{1}_{2,3} = \begin{bmatrix} h_{1,1} - h_{1,2} \\ h_{2,1} - h_{2,2} \\ h_{3,1} - h_{3,2} \end{bmatrix},$$

$$D^{2}_{2,3} = \begin{bmatrix} h_{1,2} - h_{1,3} \\ h_{2,2} - h_{2,3} \\ h_{3,2} - h_{3,3} \end{bmatrix}.$$
in Fig. A2, with a matrix $H = [h_{ij}]$, the matrices $D_{1,2}, D_{2,3}, D_{1,2}, D_{2,3}, D_{1,2}, D_{2,3}$ and $D_{2,3}$ are given by

$$D_{1,2} = [h_{1,2} - h_{1,3}], \quad D_{2,3} = [h_{2,2} - h_{2,3}], \quad D_{1,2} = [h_{1,1} - h_{1,2}], \quad D_{2,3} = [h_{2,1} - h_{2,3}], \quad D_{1,2} = [h_{1,1} - h_{1,2}], \quad D_{2,3} = [h_{2,1} - h_{2,3}].$$

The complexity $\chi(H)$ is the total number of connected components of nonzero terms in the previous matrices. This is illustrated in Fig. A2, with a matrix $H$ for which $\chi(H) = 9$.

A.2. 3D case

In the $p=3$ case, given a $3 \times 3 \times 3$ configuration matrix $H = [h_{ij}]$, the matrices $D_{1,2}, D_{2,3}, D_{1,2}, D_{2,3}, D_{1,2}, D_{2,3}$ and $D_{2,3}$ are given by

$$D_{1,2} = [h_{1,2} - h_{1,3}], \quad D_{2,3} = [h_{2,2} - h_{2,3}], \quad D_{1,2} = [h_{1,1} - h_{1,2}], \quad D_{2,3} = [h_{2,1} - h_{2,3}], \quad D_{1,2} = [h_{1,1} - h_{1,2}], \quad D_{2,3} = [h_{2,1} - h_{2,3}].$$

The complexity $\chi(H)$ is the total number of connected components of nonzero terms in the previous matrices. This is illustrated in Fig. A2, with a matrix $H$ for which $\chi(H) = 9$.

A.3. Discussion

The idea of using the grid-based configuration for modeling the complexity of decision boundary is taken from [2], where a definition of complexity is proposed based on the Bayes tree classifier designed for each configuration. There, the distributional complexity of a feature-label distribution is defined to be “the minimal number of hyperplanes necessary to achieve the Bayes classifier if the Bayes classifier is achievable by a finite number of hyperplanes, and in infinity otherwise”. Whereas that former definition links the complexity to the characteristics of the Bayes tree classifier, our definition deals exactly with the final outcome of this classifier, which is the Bayes decision boundary.

Fig. A3 shows a configuration in the 2D case which has complexity 5 according to our new definition. Based on the previous definition, the complexity of this configuration is 4, as we need four hyperplanes to make the Bayes tree classifier. The measure defined in this paper focuses on the decision boundary, assuming completely separable classes. Fixing the value of Bayes error to zero, this definition enables one to directly study the effect of the geometric complexity of the decision boundary on the performance of error estimators.

$$\epsilon_n = E_{S_n} \left( \epsilon_n(\epsilon_n[S_n]) = E_{S_n} \left( \epsilon_n(Y - g(S_n,X)) \right) \right).$$

where the notation $\epsilon_n$ indicates that the expectation is taken with respect to $F$; in fact, one can think of $(X, Y)$ in the above equation as a random test point (this interpretation being useful in understanding error estimation). The expected error rate over the data is given by

$$\epsilon_n = E_{S_n} \left( \epsilon_n(\epsilon_n[S_n]) = E_{S_n} \left( \epsilon_n(Y - g(S_n,X)) \right) \right).$$

where $F_n$ is the joint distribution of the training data $S_n$. This is sometimes called the unconditional error of the classification rule, for sample size $n$.

Were the underlying feature-label distribution $F$ known, the true error could be computed exactly, via (B.1). In practice, one is limited to using 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. Most error estimators used in practice implement some form of sample-mean-like approximation using test points. The error estimator is unbiased, with respect to the unconditional error, if the test points come from independent samples not used to design the classifier.

B.1. Resubstitution

The simplest and fastest way to estimate the error of a designed classifier in the absence of test data is to compute its error directly on the sample data itself

$$\epsilon_{\text{resub}} = \frac{1}{n} \sum_{i=1}^{n} |y_i - g(S_{n},x_i)|.$$
B.2. Cross-validation

Cross-validation removes the optimism from resubstitution by employing test points not used in classifier design [9]. In k-fold cross-validation, the data set $S_n$ is partitioned into $k$ folds $S_{in}$, 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 is the overall proportion of error committed on all folds

$$\hat{e}_{cvk} = \frac{1}{k} \sum_{i=1}^{k} \sum_{j=1}^{n/k} |y_j^{i} - g(S_{n-S_{i}}, x_j^{i})|, \quad \text{(B.4)}$$

where $(x_j^{i}, y_j^{i})$ is a sample in the $i$-th fold. 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 $e_{n,k}[g]$. The most well-known cross-validation method, usually attributed to [10], is the leave-one-out estimator, whereby a single observation is left out each time

$$\hat{e}_{loo} = \frac{1}{n} \sum_{i=1}^{n} |y_i - g(S_{n-1}, x_i)|, \quad \text{(B.5)}$$

where $S_{n-1}$ is the data set resulting from deleting data point $i$ from the original data set $S_n$. This corresponds to $n$-fold cross-validation. The leave-one-out estimator is unbiased as an estimator of $e_{n,1}[g]$. Cross-validation estimators are often pessimistic, since they use smaller training sets to design the classifier. Their main drawback is their variance [11,1]. They can also be quite slow to compute when the number of folds or samples is large.

B.3. Bootstrap

The bootstrap error estimation technique [12,13] is based on the notion of an “empirical distribution” $F^*$, which serves as a replacement to the original unknown distribution $F$. The empirical distribution puts mass $1/n$ on each of the $n$ available data points. A “bootstrap sample” $S_n^b$ from $F^*$ consists of $n$ equally likely draws with replacement from the original data set $S_n$. Hence, some of the samples will appear multiple times, whereas others will not appear at all. The actual proportion of times a data point $(x_i,y_i)$ appears in $S_n^b$ can be written as $P_i^b = (1/n)^{N_i} \sum_{j=1}^{n} I(x_j^{b}=x_i, y_j^{b}=y_i)$, where $N_i$ is 1 if the statement $S$ is true, zero otherwise. The basic bootstrap zero estimator [14] is written in terms of the empirical distribution as $\hat{e}_b = E_{F^*}[Y - \psi(X)] \in S_n^b$. In practice, the expectation $E_{F^*}$ can be approximated by a Monte-Carlo estimate based on independent replicates $S_n^b$, for $b = 1, \ldots, B$ ($B$ between 25 and 200 being recommended [14])

$$\hat{e}_b = \frac{1}{B} \sum_{b=1}^{B} \sum_{i=1}^{n} |y_i - g(S_{n-S_i}, x_i)| / P_i^b = 0.$$  \quad \text{(B.6)}$$

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 $e_{n}[g]$, since the amount of samples available for designing the classifier is on average only $(1-1/n)^n \approx 0.632n$. The estimator

$$\hat{e}_{b,632} = (1-0.632)\hat{e}_{\text{resub}} + 0.632\hat{e}_b,$$  \quad \text{(B.7)}$$
tries to correct this bias by doing a weighted average of the bootstrap zero and resubstitution estimators. It is known as the 0.632 bootstrap estimator [14], and has been perhaps the most popular bootstrap estimator in data mining [15]. It has low variance, but can be extremely slow to compute. In addition, it can fail when resubstitution is too low-biased [11].

B.4. Bolstered resubstitution

As mentioned, the empirical feature-label distribution $F^*$ is a discrete distribution that puts mass $1/n$ on each of the $n$ available data points. The resubstitution estimator can be written in terms of the empirical feature-label distribution as

$$\hat{e}_{\text{resub}} = E_{F^*}[Y - \psi(X)].$$  \quad \text{(B.8)}$$

Relative to $F^*$, no distinction is made between points near or far from the decision boundary. If one spreads the probability mass of the empirical distribution at each point, 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. Consider a probability density function $f_i^g$, for $i = 1, \ldots, n$, called a bolstering kernel, and define the bolstered empirical distribution $F^g$, with probability density function given by

$$f^g(X) = \frac{1}{n} \sum_{i=1}^{n} f_i^g(X - X_i).$$  \quad \text{(B.9)}$$

The bolstered resubstitution estimator $\hat{e}^g$ is obtained by replacing $F^*$ by $F^g$ in Eq. (B.8) to obtain

$$\hat{e}_{\text{bolst}} = E_{F^g}[Y - \psi(X)].$$  \quad \text{(B.10)}$$

Bolstering can be applied to other error estimators; however, we only use bolstered resubstitution, the bolstering method used the most to date.

The bolstered resubstitution estimator is given by

$$\hat{e}_{\text{bolst}} = 1/n \sum_{i=1}^{n} \left( I_{y_i=0} \int_{h_i} f_i^g(X-X_i) \, dx + I_{y_i=1} \int_{0}^{h_i} f_i^g(X-X_i) \, dx \right).$$  \quad \text{(B.11)}$$

where $A_i = \{X|\psi(X) = y_i\}$. The integrals are the error contributions made by the data points, according to whether $y_i = 0$ or $y_i = 1$. 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.

The amount of bolstering determines the variance and bias properties (hence, RMS also) of the bolstered estimator. As a general rule, wider bolstering kernels lead to lower-variance estimators, but after a certain point this advantage becomes offset by increasing bias. A zero-mean, spherical Gaussian bolstering kernel $f_i^g$ with covariance matrix of the form $\kappa_i I$, where $I$ is the identity matrix, has been proposed [4], and has been shown to work well in low-dimensional feature spaces. The standard deviation $\kappa_i$ is based on a non-parametric sample-based estimator, $\delta_p$, of the mean minimum distance between points belonging to class $y_i$. Specifically, $\kappa_i = \delta_p / 3p$, where $3p$ is a correction constant that depends only on the dimension $p$. We refer to [4] for details on computing $\delta_p$ and $3p$.

References

Esmaeil Atashpaz-Gargari received electrical engineering B.S. degree from Tabriz University, Iran, in 2004 and M.Sc. degree from University of Tehran, Iran, in 2007. He is currently a Ph.D. student in Genomic Signal Processing Laboratory at the Department of Electrical and Computer Engineering of Texas A&M University, College Station, TX. His current research includes the study of performance of error estimators in small-sample classification.

Chao Sima is a Research Assistant Professor in Computational Biology Division of the Translational Genomics Research Institute in Phoenix, AZ. He received his Ph.D. degree in electrical and computer engineering from Texas A&M University, College Station, TX in 2006, and was a post-doctoral associate in the Department of Statistics in Texas A&M University until February 2007. His research interests include genomic signal processing and statistical analysis, pattern recognition, and microarray classification. He is also interested in prior knowledge based engineering approach, including gene regulatory networks and time-series data inference, to model drug responses in cancer cells.

Ulisses M. Braga-Neto is an Assistant Professor and member of the Genomic Signal Processing Laboratory at the Department of Electrical and Computer Engineering of Texas A&M University, College Station, TX. He received the Ph.D. degree in electrical and computer engineering from The Johns Hopkins University, Baltimore, Maryland, in 2002. He held a post-doctoral fellowship in the section of clinical cancer genetics at the University of Texas M.D. Anderson Cancer Center, Houston, from 2002 to 2004. His current research interests include statistical pattern recognition and genomic and proteomic signal processing. He is a Senior Member of the IEEE. He received the NSF CAREER Award in 2008 for his work on small-sample error estimation.

Edward R. Dougherty is a Professor in the Department of Electrical and Computer Engineering at Texas A&M University in College Station, TX, where he holds the Robert M. Kennedy '26 Chair in Electrical Engineering and is Director of the Genomic Signal Processing Laboratory. He is also co-Director of the Computational Biology Division of the Translational Genomics Research Institute in Phoenix, AZ, and is an Adjunct Professor in the Department of Bioinformatics and Computational Biology at the University of Texas M.D. Anderson Cancer Center in Houston, TX. He holds a Ph.D. in mathematics from Rutgers University and an M.S. in Computer Science from Stevens Institute of Technology, and has been awarded the Doctor Honoris Causa by the Tampere University of Technology in Finland. He is a fellow of both IEEE and SPIE, has received the SPIE President’s Award, and served as the editor of the SPIE/IS&T Journal of Electronic Imaging. At Texas A&M University he has received the Association of Former Students Distinguished Achievement Award in Research, been named Fellow of the Texas Engineering Experiment Station, and named Halliburton Professor of the Dwight Look College of Engineering. He is author of 16 books, editor of 5 others, and author of 280 journal papers.