

# Classification and Error Estimation for Discrete Data\*

Ulisses M. Braga-Neto  
Department of Electrical and Computer Engineering  
Texas A&M University  
College Station, TX 77845

July 17, 2009

## Abstract

Discrete classification is common in Genomic Signal Processing applications, in particular in classification of discretized gene expression data, and in discrete gene expression prediction and the inference of boolean genomic regulatory networks. Once a discrete classifier is obtained from sample data, its performance must be evaluated through its classification error. In practice, error estimation methods must then be employed to obtain reliable estimates of the classification error based on the available data. Both classifier design and error estimation are complicated, in the case of Genomics, by the prevalence of small-sample data sets in such applications. This paper presents a broad review of the methodology of classification and error estimation for discrete data, in the context of Genomics, focusing on the study of performance in small sample scenarios, as well as asymptotic behavior.

**Keywords:** Genomics, Classification, Error Estimation, Discrete Histogram Rule, Sampling Distribution, Resubstitution, Leave-one-out, Ensemble Methods, Coefficient of Determination.

## 1 Introduction

In high-throughput Genomics applications, the objective is often to classify different phenotypes based on a panel of gene expression biomarkers, or to infer underlying gene regulatory networks from gene expression data. It is often advantageous to discretize gene expression data, for data efficiency and classification accuracy reasons. Classification of discrete data is a subject with a long history in Pattern Recognition [11, 22, 23, 25, 26, 29, 30]. In Genomics applications, this methodology has been applied both in classification of discretized gene expression data [38, 46], and in discrete gene expression prediction and the inference of boolean genomic regulatory networks, via the *binary coefficient of determination* (CoD) [16, 36, 37].

The most often employed discrete classification rule is the *discrete histogram rule* [11, 13, 22, 26, 29]. This classification rule has many desirable properties. For example, it can be shown that it is strongly universally consistent, meaning that, regardless of the particular distribution of the data, this rule can eventually learn the optimal classifier from the data, as the sample size increases, with probability one. In addition, the discrete histogram rule is simple enough to allow the exact analytical study of many of its properties.

---

\*To Appear in *Current Genomics*, 2009.

Once a classifier is obtained from sample data, its performance must be evaluated. The most important criterion for performance is the *classification error*, which is the probability of making an erroneous classification on a future sample. The classification error can be computed exactly only if the underlying distribution of the data is known, which is almost never the case in practice. Robust *error estimation* methods must then be employed to obtain reliable estimates of the classification error based on the available data. An error estimator is a sample-based statistic, the bias and variance (and thus the root mean square error, RMS) properties of which determine how consistently the error estimator is near the true classification error, considering all possible sample training data sets from a given population. More generally, all statistical questions regarding the accuracy of the error estimator can be answered through the joint sampling distribution of the error estimator and true probability of error [45]. From an epistemological perspective, error estimation has to do with the fundamental question of the validity of scientific knowledge [14]. The quality of the error estimate determines the accuracy of the predictions that can be performed by the inferred model, and thus its scientific content.

Both classifier design and error estimation are complicated, in the case of Genomics, by the prevalence of *small-sample* data sets in such applications. With a small training sample set, the designed classifier will be, on average, more dissimilar to the optimal classifier, and thus have a larger classification error. In addition to that, in a small-sample setting, one must use the same data to both design the classifier and assess its validity, which requires data-efficient error estimators, and this in turn calls for careful study of performance.

It is the goal of the present paper to present a broad review of the methodology of classification and error estimation for discrete data, in the context of Genomics. The paper is organized as follows. Section 2 illustrates the application of discrete classification in Genomics through a pair of simple examples. Section 3 formalizes the problem, with particular emphasis on the discrete histogram classification rule. Section 4 reviews the most common error estimators used in discrete classification, commenting briefly on their properties. Sections 5 through 7 contain the bulk of the literature review on the subject. Section 5 reviews results on the small-sample performance of discrete classification; these are analyses that must hold for a given finite number of samples. This section reviews exact and approximate expressions for performance metrics of the actual and estimated errors for the discrete histogram rule; complete enumeration methods that can deal with intractable cases such as conditional performance metrics; distribution-free results on small-sample performance, with emphasis on the pioneering work of G.F. Hughes; as well as recent analytical results that indicate that ensemble classification methods may be largely ineffective in discrete classification. Section 6, by contrast, focuses on the large-sample performance of discrete classification; this is a more technical section, which reviews asymptotic results on whether optimal performance is reached, and how fast, as the sample size increases. Finally, Section 7 reviews the binary coefficient of determination (CoD).

## 2 Discrete Classification in Genomics

The objective of *classification* is to employ a set of *training data*, consisting of independently observed known cases, and obtain a fixed rule to classify, as accurately as possible, unknown cases from the same population. The training data consists of carefully measured values of predictive variables and a response variable for each case. The response variable in classification is always *discrete*, i.e., it assumes a finite number of values; in fact, it is often binary, indicating one of two states, such as distinct cell phenotypes, disease severity, and so on.

If the predictor variables are also discrete, then one is in the context of *discrete classification*, also known as *multinomial classification* [13] and *discrete discriminant analysis* [23]. Additionally, in Statistics, the term *categorical data analysis* is often employed to refer to the statistical analysis of discrete data [3]. In

Genomics, the predictor variables often correspond to the expression of a set of suitably selected genes; for discrete classification, gene expression must first be discretized into a finite number of intervals — methods to accomplish this are described in [38, 46]. Note that the finite values taken on by the discrete predictors could be numeric (e.g., the mid-point value of an expression range), or purely categorical, as is often done by alluding to “up-regulated” and “down-regulated” genes. This distinction is immaterial in the case of the most commonly used discrete classification rule, known as the *discrete histogram rule* [11, 13, 22, 26, 29]. The discrete histogram rule simply assigns to each combination of possible values of the predictor variables a response value that is decided by majority voting among the observed response values. As will be seen in this paper, this simple rule has many desirable and interesting properties.

Figure 1 depicts an example of how the discrete histogram rule would function in the case of cell phenotype classification based on the discretized expression values of two genes. Classification is between the phenotypes of “treated” and “untreated” cells (e.g., presence or absence of some drug in the culture, presence of enough nutrients vs. starvation, normal vs. abnormal cells, and a host of other possible conditions), and gene expression is discretized in ternary values, corresponding to down-regulated, basal, and up-regulated values. There are therefore  $3^2 = 9$  possible combinations of observable values, or “bins,” which can be organized in this case into a  $3 \times 3$  matrix. In this example, the observed training data set contains a total of 40 cases, with an equal number of cases in each of the “treated” and “untreated” categories (sometimes called a “balanced experimental design” in Statistics). The *counts* of observed response values over each of the bins are shown in the figure. The majority class is underlined in each case, and this would be the class assigned to a future case by the discrete histogram rule if the observed gene expression values fall into that particular bin. Note that there are two particular cases that require attention: there could be a tie between the counts over a bin, and no values might be observed in the training data over a bin (missing data). These cases can be resolved by randomly picking one of the classes or, if one wants to avoid random classifiers, one can break ties, in a fixed manner, in favor of one of the classes; for example, one might classify such cases as “untreated.” Based on the resulting discrete classifier in this particular example, one might posit that up-regulation of both genes is associated with treatment of the cells.

		Gene A		
		down-regulated	basal	up-regulated
Gene B	down-regulated		<u>untreated = 6</u> treated = 3	
	basal	<u>untreated = 5</u>	<u>untreated = 3</u> treated = 2	untreated = 1 <u>treated = 5</u>
	up-regulated	<u>untreated = 3</u> treated = 2	untreated = 2 treated = 2	<u>treated = 6</u>

Figure 1: Phenotype classification based on discrete gene expression.

Figure 2 depicts another example, which illustrates how the discrete histogram rule would be applied in the case of discrete gene expression prediction; this constitutes the basic building block for the inference of gene regulatory networks [16, 36]. Gene expression values have been discretized into binary values, indicating activation or not of the particular gene, and the expression of three genes (the predictor variables) is used

to predict the expression of a fourth gene (the response, or *target*, variable). Note that the number of bins in this case is  $2^3 = 8$ . The bins are represented side by side in Figure 2, rather than organized into a matrix as in Figure 1. This clearly makes no difference to the discrete histogram rule (an important point we will return to in the next Section). Note that the values of all variables (predictors and target) are coded into 0 and 1, and that ties in this example are broken in favor of the class 1, that is, high-expression. As can be seen, prediction is based on a total of 40 cases, i.e., 40 instances of the 4-tuple consisting of the three predictor genes and the target gene. Note that the values 0 and 1 for the target gene are not represented equally, so the design is “unbalanced.” It will be rarely the case in gene prediction that the design is balanced, since here one cannot possibly or meaningfully specify in advance the target classes for the observations; this is a very important difference with respect to the previous case of phenotype classification, where it is often possible and meaningful to do so.

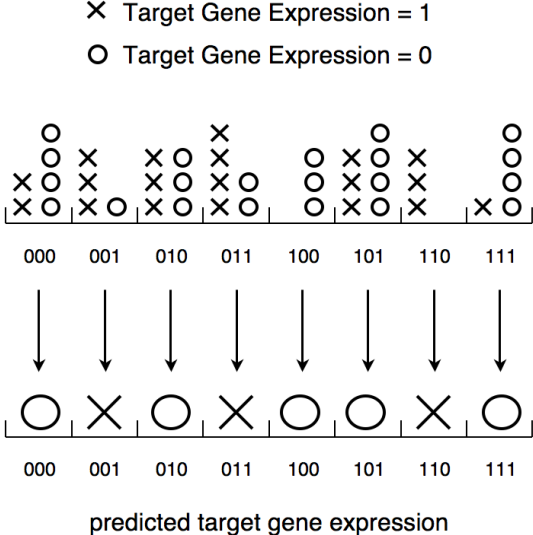


Figure 2: Prediction of discrete gene expression.

The validity of any scientific conclusions made based on the previous classification models depends, naturally, on the accuracy of the obtained classifiers. In addition, it critically depends on the reliable *estimation* of said accuracy, based on the available data. These issues will be approached in the sequel.

### 3 Discrete Classification Rules

Examples of discrete classification rules include the discrete histogram rule, mentioned in the previous section, as well as the maximum-mean-accuracy rule of [30], and many other examples of discrete rules used in Data Mining [43]. Among these, the discrete histogram rule is by far the most used one in practice. The discrete histogram rule is “natural” for categorical problems, not only due to its simplicity as majority voting over the bins, but also because it corresponds to the plug-in rule for approximating the optimal Bayes classifier, as we discuss below. In this section, we formalize the problem of discrete classification, which allows us to

examine the properties of the discrete histogram rule, including classification accuracy and its estimation from data.

Let the predictor variables be denoted by  $X_1, X_2, \dots, X_d$ , and the response variable be denoted by  $Y$ . We assume here, for simplicity, that  $Y \in \{0, 1\}$  is a binary response variable. In discrete classification, each  $X_i$  is allowed to take on a finite number  $b_i$  of values. The *feature space* is thus finite, consisting of  $b = \prod_{i=1}^d b_i$  possible states (see the matrix in Figure 1). As remarked in connection with Figure 2, for the discrete histogram rule, the space can be reorganized in any way one likes. Therefore, we adopt a (bijective) mapping between the original feature space and the sequence of integers  $1, \dots, b$ , and may equivalently assume, without loss of generality, a single predictor variable  $X$  taking on values in the set  $\{1, \dots, b\}$ . The value  $b$  is the total number of bins into which the data are categorized — this parameter provides a direct measure of the complexity of discrete classification.

The properties of the discrete classification problem are completely determined by the (discrete) joint probability distribution between the predictor  $X$  and the target  $Y$ :

$$P(X = i, Y = j), \text{ for } i = 1, \dots, b \text{ and } j = 0, 1.$$

Given the identity  $P(X = i, Y = j) = P(X = i|Y = j)P(Y = j)$ , it becomes clear that the discrete classification problem is determined by  $2b + 2$  positive parameters  $c_0 = P(Y = 0)$ ,  $c_1 = P(Y = 1)$ , and  $p_i = P(X = i|Y = 0)$ ,  $q_i = P(X = i|Y = 1)$ , for  $i = 1, \dots, b$ . Note that the parameters are not independent, since one must have  $c_0 + c_1 = 1$ ,  $\sum p_i = 1$ , and  $\sum q_i = 1$ .

Through Bayes' theorem, these model parameters determine the posterior probabilities  $P(Y = j|X = i)$  for the classification problem,

$$P(Y = 0|X = i) = \frac{P(Y = 0, X = i)}{P(X = i)} = \frac{c_0 p_i}{c_0 p_i + c_1 q_i}$$

with  $P(Y = 1|X = i) = 1 - P(Y = 0|X = i)$ . Therefore, the classifier  $\psi^*$  that achieves the minimum *probability of misclassification*  $P(Y \neq \psi(X))$ , known as the *Bayes classifier* [13], is given by

$$\psi^*(X = i) = \begin{cases} 1, & P(Y = 0|X = i) < P(Y = 1|X = i) \\ 0, & P(Y = 0|X = i) \geq P(Y = 1|X = i) \end{cases} = \begin{cases} 1, & c_0 p_i < c_1 q_i \\ 0, & c_0 p_i \geq c_1 q_i \end{cases} \quad (1)$$

It can be shown that if there are two or more discrete features in the original feature space (such as in Figure 1), and these features are independent conditionally to  $Y$ , i.e., within each class, then the Bayes classifier  $\psi^*$  is a linear function of those features [13, p. 466].

The minimum probability of misclassification, or *Bayes error*, achieved by the Bayes classifier, can be written as

$$\begin{aligned} \epsilon^* &= \sum_{i=1}^b P(X = i, Y = 1 - \psi^*(X = i)) = \sum_{i=1}^b [c_0 p_i I_{c_1 q_i > c_0 p_i} + c_1 q_i I_{c_0 p_i \geq c_1 q_i}] \\ &= \sum_{i=1}^b \min\{c_0 p_i, c_1 q_i\}. \end{aligned} \quad (2)$$

Here,  $I_A$  is an *indicator variable*, which is 1 when condition  $A$  happens, and 0, otherwise. Since  $\sum \min\{a_i, b_i\} \leq \min\{\sum a_i, \sum b_i\}$ , it follows that  $0 \leq \epsilon^* \leq \min\{c_0, c_1\}$ . The upper bound is reached if (though not only if)

$p_i = q_i$ , for all  $i = 1, \dots, b$ . The largest Bayes error possible is 0.5, which is achieved if and only if  $c_0 = c_1 = 0.5$  and  $p_i = q_i$ , for all  $i = 1, \dots, b$  (total confusion between the classes).

The Bayes error is a measure of distance between the classes, and it provides a lower bound on classification performance. For discrete histogram classification, the predictor variables in the original feature space should be chosen so that the Bayes error is as small as possible.

In practice, one almost never knows the model parameters completely, and therefore one does not know the Bayes classifier. One must rely instead on designing a classifier from sample *training data*; one hopes that such a sample-based classifier is close in some sense to the Bayes classifier. The classifier produced by the discrete histogram rule becomes indeed very close to the Bayes classifier, as sample size increases, in a few important senses; this will be discussed in Section 6.

Given sample data  $S_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$  containing  $n$  independent and identically distributed (i.i.d.) samples, one defines the *bin counts*  $U_i, V_i$  as the observed number of points with  $X = i$  for class 0 and 1, respectively, for  $i = 1, \dots, b$ . For example, in Figure 2, the bin counts  $U_i$  are 4,1,3,3,2,4,0,4, while the bin counts  $V_i$  are 2,3,3,4,0,3,3,1. The *discrete histogram classification rule* is given by majority voting between the bin counts over each bin:

$$\Psi_n(S_n, X = i) = I_{U_i < V_i} = \begin{cases} 1, & U_i < V_i \\ 0, & U_i \geq V_i \end{cases}, \quad i = 1, \dots, b. \quad (3)$$

When a specific training sample  $S_n = s_n$  is given, then the values of the bin counts  $U_i$  and  $V_i$  become fixed, leading to a fixed *designed* discrete histogram classifier  $\psi_n(\cdot) = \Psi_n(S_n = s_n, \cdot)$ . For an example, see Figure 2. Note that, in the above definition, as in Figure 2, we choose to break ties in favor of class 0.

Ordinarily, the samples in  $S_n$  are drawn from a mixture of the two class populations, and therefore the numbers  $N_0 = \sum U_i$  and  $N_1 = \sum V_i$  of samples in classes 0 and 1, respectively, are random variables. In this *full sampling* case,  $N_0$  and  $N_1$  are binomially distributed:  $N_0 \sim \text{Binomial}(n, c_0)$  and  $N_1 \sim \text{Binomial}(n, c_1)$ , with  $N_0 + N_1 = n$ . In addition, the vector of bin counts  $(U_1, \dots, U_b, V_1, \dots, V_b)$  is jointly multinomially distributed, with parameters  $(n, c_0 p_1, \dots, c_0 p_b, c_1 q_1, \dots, c_1 q_b)$ ; it follows that the individual bin counts are binomially distributed:  $U_i \sim \text{Binomial}(n, c_0 p_i)$  and  $V_i \sim \text{Binomial}(n, c_1 q_i)$ , for  $i = 1, \dots, b$ . On the other hand, one may design the experiment in a such a way that the number of samples  $N_0 = n_0$  and  $N_1 = n_1$  are fixed in advance, with  $n_0 + n_1 = n$ , and the class populations are sampled separately. To avoid bias, the values of  $n_0$  and  $n_1$  should be chosen to reflect the a priori probabilities  $c_0$  and  $c_1$  of each class:  $n_0 = \lceil c_0 n \rceil$  and  $n_1 = \lceil c_1 n \rceil$ , where  $\lceil x \rceil$  denotes the nearest integer to  $x$ . In this *stratified sampling* case, the vector of bin counts  $(U_1, \dots, U_b)$  is multinomially distributed with parameters  $(n_0, p_1, \dots, p_b)$ , and is independent from the vector of bin counts  $(V_1, \dots, V_b)$ , which is multinomially distributed with parameters  $(n_1, q_1, \dots, q_b)$ ; the individual bin counts are still binomially distributed:  $U_i \sim \text{Binomial}(n_0, p_i)$  and  $V_i \sim \text{Binomial}(n_1, q_i)$ , for  $i = 1, \dots, b$ .

The discrete histogram rule is the “plug-in” rule for discrete classification, that is, if one plugs the standard maximum-likelihood (ML) estimators of the unknown model parameters  $c_0, c_1$  and  $\{p_i\}, \{q_i\}$ ,

$$\hat{c}_0 = \frac{\sum_i U_i}{n}, \quad \hat{c}_1 = \frac{\sum_i V_i}{n} \quad \text{and} \quad \hat{p}_i = \frac{U_i}{\sum_i U_i}, \quad \hat{q}_i = \frac{V_i}{\sum_i V_i}, \quad \text{for } i = 1, \dots, b, \quad (4)$$

in the expression for the Bayes classifier in (1), one obtains precisely the histogram classifier in (3). Since the standard ML estimators in (4) are consistent, meaning that they converge to the true values of the parameters as the sample size increases, one would expect the discrete histogram classifier to approach the optimal Bayes classifier as more samples are acquired, which is indeed the case; we come back to this issue in Section 6.

The most important performance criterion for the designed classifier  $\psi_n$  is its accuracy on independent (e.g., future) data, which are assumed to come from the same population as the given training data. This accuracy is measured by the probability of misclassification  $\epsilon_n = P(Y \neq \psi_n(X))$ , where  $(X, Y)$  is i.i.d. with all  $(X_i, Y_i)$  in  $S_n$ . This is known as the *classification error*. It is clear that

$$\epsilon_n = \sum_{i=1}^b P(X = i, Y = 1 - \psi_n(X = i)) = \sum_{i=1}^b [c_0 p_i I_{V_i > U_i} + c_1 q_i I_{U_i \geq V_i}]. \quad (5)$$

Being a function of the random variables  $U_i$  and  $V_i$ ,  $\epsilon_n$  is a random variable ( $\epsilon_n$  ceases to be random, becoming fixed, when a fixed training data set  $S_n$ , and thus fixed values of  $U_i$  and  $V_i$ , are specified). The expected value of  $E[\epsilon_n]$  over the training data  $S_n$  has an important meaning in the context of classification rules. It does not depend on a particular set of training samples, but it gives the average classification error over all possible training data; therefore it is an intrinsic performance measure of the classification rule for the particular problem (i.e., joint distribution of  $X$  and  $Y$ ) and sample size  $n$ .

## 4 Error Estimation for Discrete Classification

In practice, the underlying probability model is unknown, and the classification error  $\epsilon_n$  has to be estimated from the sample data using an *error estimator*  $\hat{\epsilon}_n$ . An error estimator is a function of the classification rule  $\Psi_n$  and the sample data  $S_n$ . Therefore, it is a random variable through dependency on the random training data  $S_n$ . If the error estimator depends on any additional random factors, sometimes called *internal factors*, it is called *randomized*, otherwise it is said to be *nonrandomized*. Examples of the latter include the *apparent error* or *resubstitution* [39], and *leave-one-out* [33] error estimators, whereas popular examples of randomized error estimators include cross-validation [12, 33, 40, 41] and all bootstrap-based error estimators [17–19].

As the classification error  $\epsilon_n$  itself, a nonrandomized error estimator  $\hat{\epsilon}_n$  produces a fixed value once the training data set  $S_n$  is specified (“running the estimator again” on the data never alters the result), which is not the case for a randomized error estimator. Internal random factors introduce *internal variance* that adds to the *total variance* of an error estimator, which measures how dispersed its estimates can be for varying training data from the same population. Note that the internal variance is zero for nonrandomized estimators. Randomized estimators typically reduce the unwanted extra internal variance through averaging based on intensive computation. See [5, 6] for a detailed discussion of issues regarding randomized and non-randomized error estimators, and internal and full variance.

The variance of the error estimator, by itself, does not address its relationship to the quantity to be estimated, namely, the actual classification error. Relevant performance metrics that do so are discussed next. The *bias*  $E[\hat{\epsilon}_n - \epsilon_n]$  of an error estimator measures whether, on average, it overestimates the true error, or underestimates it, whereas the *deviation variance*  $\text{Var}(\hat{\epsilon}_n - \epsilon_n)$  measures the spread of the deviation between estimated and actual errors; it can in fact be written as

$$\text{Var}(\hat{\epsilon}_n - \epsilon_n) = \text{Var}(\hat{\epsilon}_n) + \text{Var}(\epsilon_n) - 2\rho(\hat{\epsilon}_n, \epsilon_n)\sqrt{\text{Var}(\hat{\epsilon}_n)\text{Var}(\epsilon_n)} \quad (6)$$

a remarkable formula that combines the variances of the actual error and error estimator, and their *correlation*  $\rho(\hat{\epsilon}_n, \epsilon_n)$ . Small bias is of small significance if the deviation variance is large; this would mean that on average the error estimator is close to the true error, but that in fact the estimate for any particular sample set is likely to be far away from the true error. The *root mean-square error* (RMS)

$$\text{RMS}(\hat{\epsilon}_n) = \sqrt{E[(\hat{\epsilon}_n - \epsilon_n)^2]} = \sqrt{E[\hat{\epsilon}_n - \epsilon_n]^2 + \text{Var}(\hat{\epsilon}_n - \epsilon_n)} \quad (7)$$

conveniently combines both the bias and the deviation variance into a single measure, and is widely adopted for comparison of error estimator performance. Additional performance measures include the *tail probabilities*  $P(|\hat{\epsilon}_n - \epsilon_n| > \tau)$ , for  $\tau > 0$ , which concern the likelihood of outliers, as well as the consistency of the error estimator; the *conditional bias*  $E[\hat{\epsilon}_n - \epsilon_n | \hat{\epsilon}_n] = \hat{\epsilon}_n - E[\epsilon_n | \hat{\epsilon}_n]$  (resp. conditional deviation variance and RMS); and *confidence intervals*  $[a, b]$  such that  $P(a \leq \epsilon_n \leq b | \hat{\epsilon}_n) > 1 - \alpha$ , for  $0 \leq \alpha \leq 1$ , which give bounds on the true error corresponding to a given precision  $\alpha$ , the observed error estimate, and the sample size. Confidence intervals express statistical power in error estimation — more powerful methods will produce shorter confidence intervals for the true error at the same sample size. A very important fact is that all of the aforementioned performance metrics, and in fact any others, can be determined if one has knowledge of the *joint sampling distribution* of the vector of actual and estimated errors  $(\epsilon_n, \hat{\epsilon}_n)$ . Section 5 reviews exact analytical methods to compute these performance metrics, as well as complete enumeration methods that allow the computation of the joint sampling distribution of actual and estimated errors.

The resubstitution error estimator  $\hat{\epsilon}_n^r$  [39] is the simplest data-efficient alternative; it is simply the apparent error, or the proportion of errors the designed classifier makes on the training data itself. Clearly,

$$\hat{\epsilon}_n^r = \frac{1}{n} \sum_{i=1}^b \min\{U_i, V_i\} = \frac{1}{n} \sum_{i=1}^b [U_i I_{V_i > U_i} + V_i I_{U_i \geq V_i}]. \quad (8)$$

For example, in Figure 2, the resubstitution estimate for the classification error is  $12/40 = 0.3$ . It is easy to see that plugging the ML estimators of the model parameters in (4) into the formula for the Bayes error (2), results in expression (8). Therefore, resubstitution for the discrete histogram rule is the plug-in estimator of the Bayes error in discrete classification. The resubstitution estimator is clearly nonrandomized, and it is very fast to compute. This estimator is however always optimistically biased in the case of the discrete histogram rule, in the sense that  $E[\hat{\epsilon}_n^r] \leq E[\epsilon_n]$ , for any sample size and distribution of the data. In fact, it can be shown that

$$E[\hat{\epsilon}_n^r] \leq \epsilon^* \leq E[\epsilon_n] \quad (9)$$

so that the average resubstitution estimate bounds from below even the Bayes error; this fact seems to have been demonstrated for the first time in [11] (see also [25]). Observe though that this is not guaranteed to apply to any individual training data and classifier, but only to the average over all possible training data. The optimistic bias of resubstitution tends to be larger when the number of bins is large compared to the sample size; in other words, there is more overfitting of the classifier to the training data in such cases. On the other hand, resubstitution tends to have small variance. In cases where the bias is not too large, this makes resubstitution a very competitive option as an error estimator. In fact, the next Section contains results that show that resubstitution can have smaller RMS than even complex error estimators such as the bootstrap, provided that sample size is large compared to number of bins. In addition, it can be shown that as the sample size increases, both the bias and variance of resubstitution vanish (see Section 6). Finally, it is important to emphasize that these observations hold for the discrete histogram rule; for example, the resubstitution estimator is not necessarily optimistically-biased for other (continuous or discrete) classification rules.

The leave-one-out error estimator  $\hat{\epsilon}_n^l$  [33] removes the optimistic bias from resubstitution by counting errors committed by  $n$  classifiers, each designed on  $n - 1$  points and tested on the remaining left-out point, and dividing the total count by  $n$ . A little reflection shows that

$$\hat{\epsilon}_n^l = \frac{1}{n} \sum_{i=1}^b [U_i I_{V_i \geq U_i} + V_i I_{U_i \geq V_{i-1}}]. \quad (10)$$

For example, in Figure 2, the leave-one-out estimate for the classification error is  $15/40 = 0.375$ . This is higher than the resubstitution estimate of 0.3. In fact, by comparing (8) and (10), one can see that, in all cases, it is true that  $\hat{\epsilon}_n^l \geq \hat{\epsilon}_n^r$ . In particular,  $E[\hat{\epsilon}_n^l] \geq E[\hat{\epsilon}_n^r]$ , showing that the leave-one-out estimator must be necessarily less optimistic than the resubstitution estimator. In fact, it is a general result (not restricted to discrete histogram classification), that  $E[\hat{\epsilon}_n^l] = E[\epsilon_{n-1}]$ , making this estimator almost unbiased. As it turns out, this bias reduction is accomplished at the expense of an increase in variance [5]. The leave-one-out estimator is however nonrandomized.

A randomized estimator is obtained by selecting randomly  $k$  folds of size  $n - n/k$ , counting the errors committed by  $k$  classifiers, each designed on one of the folds and tested on the remaining points not in the fold, and dividing the total count by  $n$ . This yields the well-known  $k$ -fold *cross-validation* estimator [12, 33, 40, 41]. The process can be repeated several times and the results averaged, in order to reduce the internal variance associated with the random choice of folds. The leave-one-out estimator is a cross-validation estimator with  $k = n$ ; therefore, cross-validation is not randomized in this special case (it is also nonrandomized for other choices of  $k$  if one considers all possible folds of size  $n - n/k$ , which can be a very large number if  $n$  is large or  $k$  is small). It is a general result (not restricted to discrete histogram classification) that the  $k$ -fold cross-validation estimator  $\hat{\epsilon}_n^{cvk}$  is such that  $E[\hat{\epsilon}_n^{cvk}] = E[\epsilon_{n-n/k}]$ .

Another class of popular randomized error estimators are based on the the idea of bootstrap [17–19]. A “bootstrap sample” consists of  $n$  equally-likely draws with replacement from the original training data. The basic bootstrap estimator  $\hat{\epsilon}_0$  is similar to cross-validation, in that it counts the errors committed by  $B$  classifiers, each designed on a bootstrap sample and tested on training points not in the bootstrap sample, and divides the count by the total number of test points (which is variable). The number  $B$  of bootstrap samples must be made large to reduce the internal variance associated with bootstrap sampling (the ideal case  $B = \infty$  leading to a nonrandomized estimator; in practice, this would be achieved by a very large, but finite,  $B$ , which is equal to the number of all possible draws of  $n$  indices with replacement from the index set  $1, \dots, n$ ). The estimator  $\hat{\epsilon}_0$  tends to be pessimistically biased, and therefore a convex combination with resubstitution, which is optimistically biased (in the case of discrete histogram classification), was proposed in [18]:

$$\hat{\epsilon}_n^{b632} = (1 - 0.632) \hat{\epsilon}_r + 0.632 \hat{\epsilon}_0 . \quad (11)$$

This is known as the 0.632 *bootstrap* error estimator, and is quite popular in Machine Learning applications [43]. It has small variance, but can be very slow to compute. In addition, it will fail when the resubstitution estimator is too optimistic. A variant called the 0.632+ *bootstrap* error estimator was introduced in [19], in an attempt to correct this problem. All cross-validation and bootstrap error estimators tend to be computationally intensive, due to the large number of classifier design steps involved and the need to reduce internal variance by averaging over a large number of iterations.

## 5 Small-Sample Performance of Discrete Classification

The fact that the distribution of the vectors of bin counts  $(U_1, \dots, U_b)$  and  $(V_1, \dots, V_b)$  is multinomial (see Section 3), and thus easily computable, along with the simplicity and parallel among equations (2), (5), (8), and (10), for the Bayes error, actual error, resubstitution error, and leave-one-out error, respectively, allow the detailed analytical study of the small-sample performance of the discrete histogram classification rule and the associated resubstitution and leave-one-out error estimators.

## 5.1 Analytical Study of Actual Classification Error

From (5) it follows that the expected error over the sample is given by

$$\begin{aligned}
E[\epsilon_n] &= \sum_{i=1}^b [c_0 p_i E[I_{V_i > U_i}] + c_1 q_i E[I_{U_i \geq V_i}]] \\
&= \sum_{i=1}^b [c_0 p_i P(V_i > U_i) + c_1 q_i P(U_i \geq V_i)] \\
&= c_1 + \sum_{i=1}^b (c_0 p_i - c_1 q_i) P(V_i > U_i).
\end{aligned} \tag{12}$$

The computation of the probability  $P(V_i > U_i)$  depends on whether full or stratified sampling is used. In the full sampling case, the pair  $(U_i, V_i)$  has a *trinomial* joint distribution with parameters  $(n, c_0 p_i, c_1 q_i)$ , so that

$$P(V_i > U_i) = \sum_{l>k} \binom{n}{k, l, n-k-l} (c_0 p_i)^k (c_1 q_i)^l (1 - c_0 p_i - c_1 q_i)^{n-k-l}, \tag{13}$$

whereas in the stratified sampling case,  $U_i$  is independent of  $V_i$ , and each is binomially distributed with parameters  $(n_0, p_i)$  and  $(n_1, q_i)$ , respectively, so that

$$P(V_i > U_i) = \sum_{l>k} \binom{n_0}{k} p_i^k (1 - p_i)^{n_0-k} \binom{n_1}{l} q_i^l (1 - q_i)^{n_1-l}. \tag{14}$$

To obtain the variance  $\text{Var}[\epsilon_n] = E[\epsilon_n^2] - (E[\epsilon_n])^2$  one needs the second moment  $E[\epsilon_n^2]$ :

$$\begin{aligned}
E[\epsilon_n^2] &= \sum_{i=1}^b \{ c_0^2 p_i^2 P(V_i > U_i) + c_1^2 q_i^2 P(U_i \geq V_i) \} + \\
&\quad \sum_{\substack{i,j=1 \\ i \neq j}}^b \{ c_0^2 p_i p_j P(V_i > U_i, V_j > U_j) + \\
&\quad c_0 c_1 [p_i q_j P(V_i > U_i, U_j \geq V_j) + \\
&\quad p_j q_i P(U_i \geq V_i, V_j > U_j)] + \\
&\quad c_1^2 q_i q_j P(U_i \geq V_i, U_j \geq V_j) \}
\end{aligned} \tag{15}$$

This expression involves second-order bin probabilities, e.g.,  $P(V_i > U_i, V_j > U_j)$ , which can be computed in a similar fashion to the first-order bin probability in (13) and (14), by using the fact that, in the full sampling case, the vector  $(U_i, V_i, U_j, V_j)$  has a multinomial joint distribution with parameters  $(n, c_0 p_i, c_1 q_i, c_0 p_j, c_1 q_j)$ , whereas in the stratified sampling case, the vector  $(U_i, U_j)$  is independent of the vector  $(V_i, V_j)$ , and each is trinomially distributed with parameters  $(n_0, p_i, p_j)$  and  $(n_1, q_i, q_j)$ , respectively.

However, computation of the expression in (15) becomes difficult when  $n$  or  $b$  are large. But if one can assume that the event  $\{V_i > U_i\}$  is approximately independent of the event  $\{V_j > U_j\}$ , then it can be shown after some algebraic manipulation that cancellations occur in the expression (15), leading to a very simple expression for the variance, which involves only first-order bin probabilities:

$$\text{Var}[\epsilon_n] = \sum_{i=1}^b (c_0 p_i + c_1 q_i)^2 P(V_i > U_i) (1 - P(V_i > U_i)) \tag{16}$$

It is proved in [4] that, under a mild distributional assumption, the expression in (16) is asymptotically exact as the number of bins grows to infinity, for fixed sample size.

Figure 3 illustrates the application of the formulas above in an example where stratified sampling is assumed, with  $c_0 = c_1 = 0.5$  (so that, in particular,  $n_0 = n_1 = n/2$ ), and probabilities  $p_i$  and  $q_i$  given by a parametric Zipf (power-law) model:  $p_i = Ki^{-\alpha}$  and  $q_i = p_{b-i+1}$ , for  $i = 1, \dots, b$ . Here,  $K$  is a normalizing constant given by  $K = [\sum_{i=1}^b i^{-\alpha}]^{-1}$ . The parameter  $\alpha$  controls the Bayes error of the model, and is set in all cases to  $\alpha = \sqrt{2}$ . We can see in Figure 3 that the expected classification error decreases with increasing sample size as expected. The expected classification error also decreases with increasing bin size, but it starts to increase again after  $b > 16$  for  $n = 20$ . This is an example of the “peaking phenomenon” that affects the expected classification error (see Section 5.4). As for the variance, one can see that it also decreases with increasing sample size, as expected. Except for the anomalous case  $b = 2$ , the variance seems to be insensitive to bin size. One can also appreciate that the approximation to the variance given by (16) is quite accurate, particularly at larger sample sizes. The good accuracy of the approximation is obtained at a huge savings in computation time. As an example, for  $b = 16$  and  $n = 60$ , it takes more than 30 minutes and less than 1 second to compute the exact and approximate expressions for the variance, respectively, using state-of-the-art computing technology.

## 5.2 Analytical Study of Error Estimators

Similar exact expressions can be derived for the expectation and variance of the resubstitution and leave-out-error estimators, as well as their correlation with the actual error; see [7, 8]. These exact expressions allow one to compute exactly the bias, deviation variance, and RMS of both resubstitution and leave-one-out. This is illustrated in Figure 4, where results for resubstitution (resub), leave-one-out (loo), 10-repeated 4-fold cross-validation (cv), and the .632 bootstrap (b632) are displayed. In this figure, “standard deviation” refers to the square root of the deviation variance. For the 0.632 error estimator,  $B = 100$  bootstrap samples are employed. Performance measures for resubstitution and leave-one-out are exact; they are computed using the exact expressions mentioned previously. For the other error estimators, performance measures are derived from a Monte-Carlo computation using 20,000 samples from each probability model. The model is the Zipf parametric model mentioned previously, with  $c_0 = c_1 = 0.5$ , and parameter  $\alpha$  adjusted to yield  $E[\epsilon_n] \approx 0.20$ , which corresponds to intermediate classification difficulty. Stratified sampling is assumed, with  $n = 20$  (so that  $n_0 = n_1 = 10$ ). The value of  $n$  was chosen to emphasize small-sample effects. The results show that resubstitution is the most optimistically biased estimator, with bias that increases with complexity, but it is also much less variable than all other estimators, including the bootstrap ones. The cross-validation estimators are the most variable, but are nearly unbiased. The bootstrap estimator is affected by the bias of resubstitution when complexity is high, since it incorporates the resubstitution estimate in its computation, but it is clearly superior to the cross-validation estimators in RMS. Perhaps the most remarkable observation is that, for very low complexity classifiers (around  $b=4$ ), the simple resubstitution estimator becomes more accurate than the cross-validation error estimators, and as accurate as the 0.632 bootstrap error estimator, according to RMS, despite the fact that resubstitution is typically much faster to compute than those other error estimators (in some cases considered in [5], hundreds of times faster). In our experiments, we observed that this is true for small sample sizes ( $n < 30$ ), low complexity, and low to moderate expected classification errors. This has an important consequence for the inference of genomic boolean regulatory networks: if the number of boolean predictors for a particular gene is small (on the order of 2 or 3), then it is more advantageous to use resubstitution to estimate prediction accuracy than more complicated error estimation schemes.

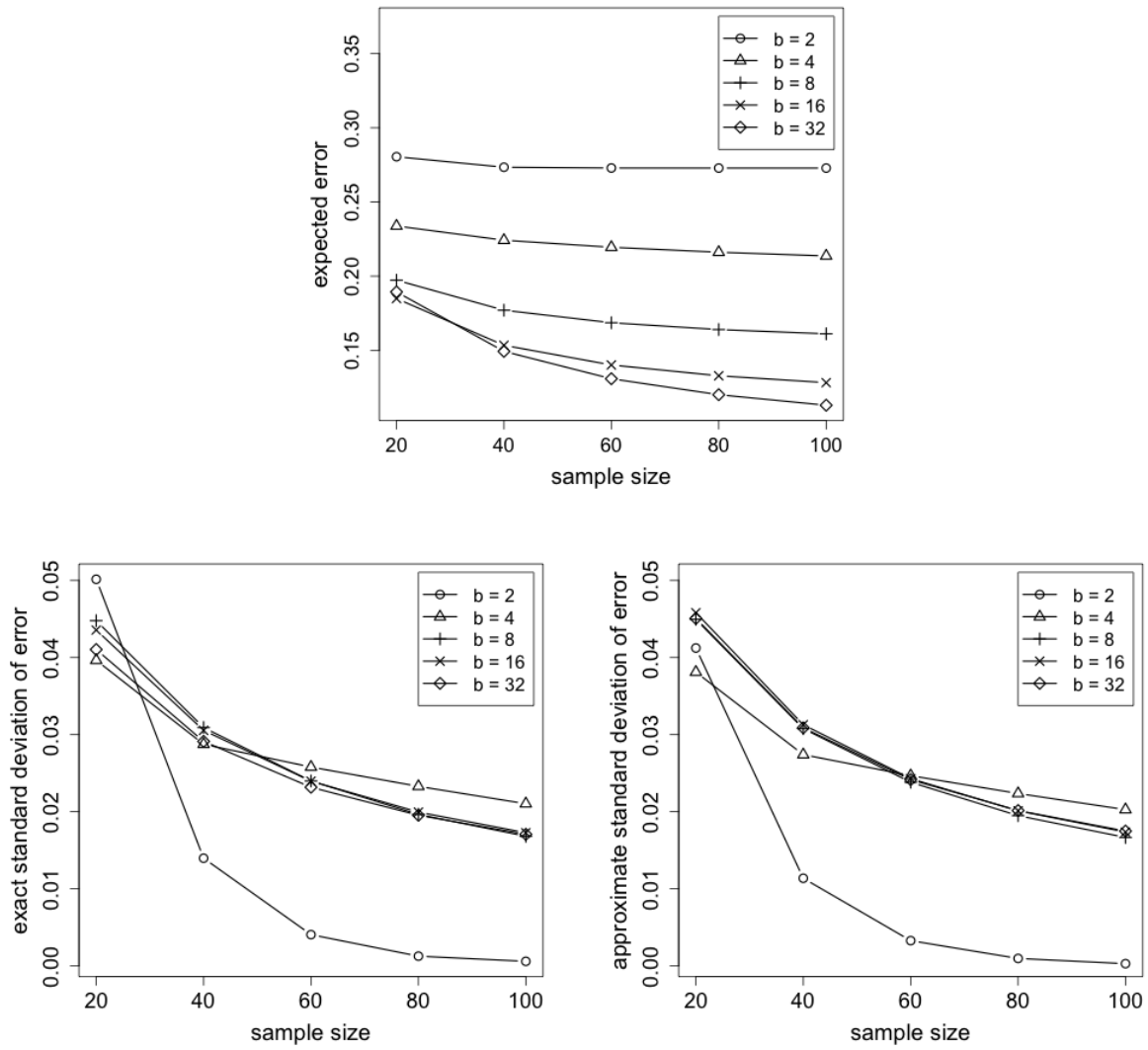


Figure 3: Performance of the discrete histogram classification rule.

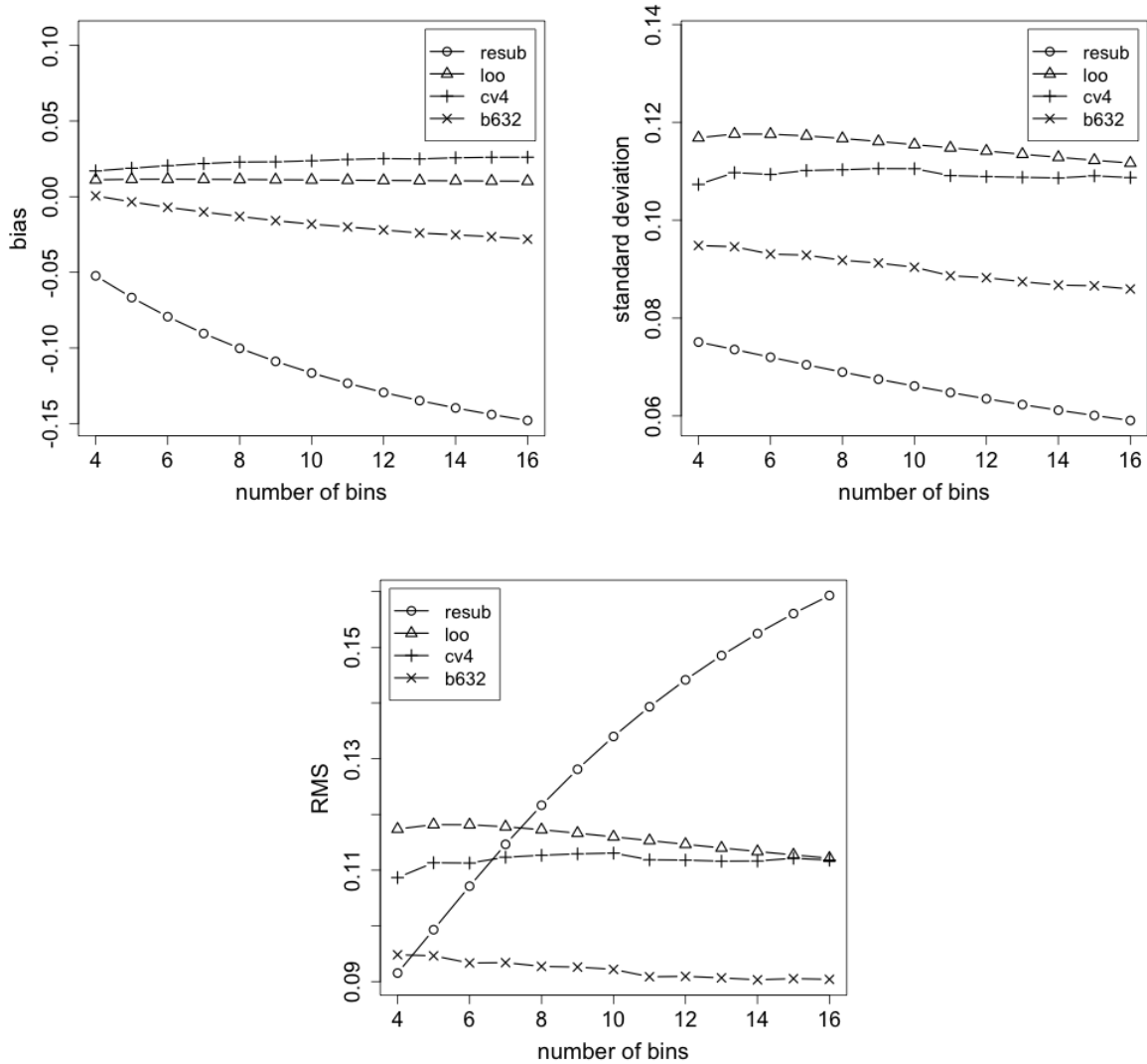


Figure 4: Performance of error estimators, for  $n = 20$  and  $E[\epsilon_n] = 0.2$ . The values for resubstitution and leave-one-out are exact; the values for the other error estimators are approximations based on Monte-Carlo computation.

Analytical exact expressions for the correlation between actual and estimated errors can also be derived [8]. This is illustrated in Figure 5, where the correlation for resubstitution and leave-one-out error estimators is plotted versus sample size, for different bin sizes. In this example, we assume full sampling and the Zipf

parametric model mentioned previously, with  $c_0 = c_1 = 0.5$  and parameter  $\alpha$  adjusted to yield two cases: easy (Bayes error = 0.1) and difficult (Bayes error = 0.4) classification.

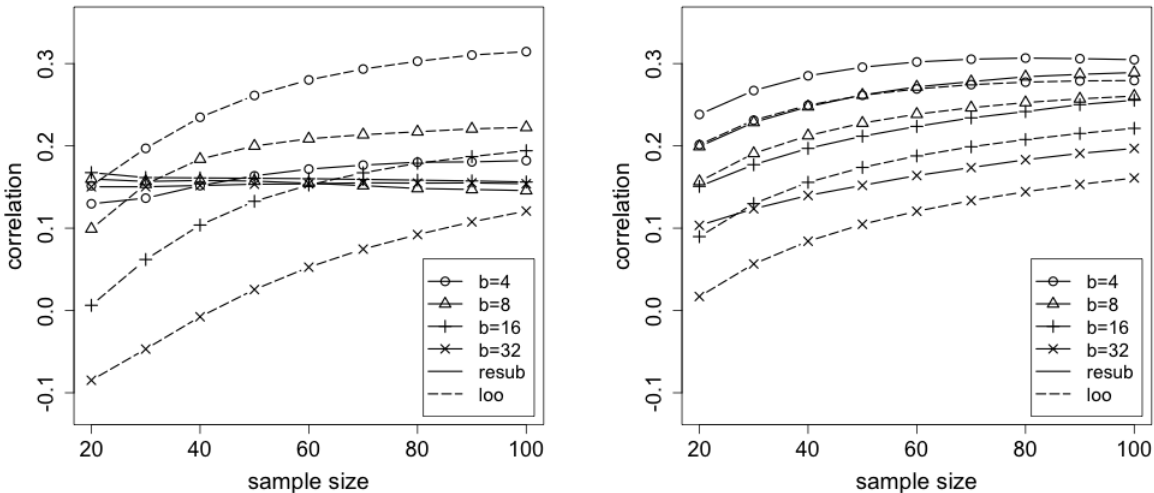


Figure 5: Exact correlation between the actual error and the resubstitution and leave-one-out cross-validation error estimators for probability models of distinct difficulty, as determined by the Bayes error. Left plot: Bayes error = 0.10. Right plot: Bayes error = 0.40.

We can observe that the correlation is generally low (below 0.3). We can also observe that at small sample sizes, correlation for resubstitution is larger than for leave-one-out cross-validation, and, with a larger difficulty of classification, this is true even at moderate sample sizes. Correlation generally decreases with increasing bin size; in one striking case, the correlation for leave-one-out becomes negative, at the critical small-sample situation of  $n = 20$  and  $b = 32$ . This behavior of the correlation for leave-one-out mirrors the behavior of deviation variance of this error estimator, which is known to be large under complex models and small sample sizes [5, 13, 24], and is in accord with (6).

### 5.3 Complete Enumeration Methods

As mentioned previously, all the performance metrics of interest for the actual error  $\epsilon_n$  and any given error estimator  $\hat{\epsilon}_n$  can be derived from joint sampling distribution of the pair of random variables  $(\epsilon_n, \hat{\epsilon}_n)$ . These include conditional metrics, such as the conditional expected actual error given the estimated error and confidence bounds on the actual error conditioned on the estimated error, which are very difficult to study via the analytical approach used in the previous subsections, due to the complexity of the expressions involved.

However, due to the finiteness of the discrete problem, it turns out that the joint sampling distribution of actual and estimated errors in the discrete case can be computed exactly by means *complete enumeration*. Such methods have been extensively used in categorical data analysis [2, 3, 27, 32, 42]; complete enumeration has been particularly useful in the computation of exact distributions and critical regions for tests based on

contingency tables, as in the case of the well-known Fisher exact test and the chi-square approximate test [2, 42].

Basically, complete enumeration relies on intensive computational power to list all possible configurations of the discrete data and their probabilities, and from this exact statistical properties of the methods of interest are obtained. The availability of efficient algorithms to enumerate all possible cases on fast computers has made possible the use of complete enumeration in an increasingly wider variety of settings.

In the case of discrete classification, recall that the random sample is specified by the vector of bin counts  $W_n = (U_1, \dots, U_b, V_1, \dots, V_b)$  defined previously, so that we can write  $\epsilon_n = \epsilon_n(W_n)$  and  $\hat{\epsilon}_n = \hat{\epsilon}_n(W_n)$ . The random vector  $W_n$  is discrete, and so the random variables  $\epsilon_n$  and  $\hat{\epsilon}_n$  are also discrete, and so is the configuration space  $D_n$  of all possible distinct sample values  $w_n = (u_1, \dots, u_b, v_1, \dots, v_b)$  that can be taken on by  $W_n$ . The discrete joint probability distribution of  $(\epsilon_n, \hat{\epsilon}_n)$  is given by:

$$P(\epsilon_n = k, \hat{\epsilon}_n = l) = \sum_{w_n \in D_n} I_{\{\epsilon_n(w_n)=k, \hat{\epsilon}_n(w_n)=l\}} P(W_n = w_n), \quad (17)$$

where  $P(W_n = w_n)$ , is a multinomial probability that is computed according to the parameters  $(n, c_0 p_1, \dots, c_0 p_b, c_1 q_1, \dots, c_1 q_b)$  as

$$P(W_n = w_n) = \binom{n}{u_1, \dots, u_b, v_1, \dots, v_b} c_0^{\sum_i u_i} c_1^{\sum_i v_i} \prod_{i=1}^b p_i^{u_i} q_i^{v_i} \quad (18)$$

Even though the configuration space  $D_n$  is finite, it quickly becomes huge with increasing sample size  $n$  and bin size  $b$ . In [7] an algorithm is given to traverse  $D_n$  efficiently, which leads to reasonable computational times to evaluate the joint sampling distribution when  $n$  and  $b$  are not too large. Figure 6 displays the joint distribution  $P(\epsilon_n = k, \hat{\epsilon}_n = l)$  for the resubstitution and leave-one-out cross-validation error estimators, for a small-sample case,  $n = 20$  and  $b = 8$ , and a Zipf probability model of intermediate difficulty (Bayes error = 0.2). One can observe that the joint distribution for resubstitution is much more compact than for leave-one-out cross-validation, which explains in part its larger correlation in small-sample cases.

This approach can be easily modified to compute the conditional sampling distribution  $P(\epsilon_n = k | \hat{\epsilon}_n = l)$ . This was done in [45] in order to find exact conditional metrics of performance for resubstitution and leave-one-out error estimators. Those included the conditional expectation  $E[\epsilon_n | \hat{\epsilon}_n]$  and conditional variance  $\text{Var}(\epsilon_n | \hat{\epsilon}_n)$  of the actual error given the estimated error, as well as the 100(1 -  $\alpha$ )% upper confidence bound  $\gamma_\alpha$ , such that  $P(\epsilon_n < \gamma_\alpha | \hat{\epsilon}_n) = 1 - \alpha$ .

This is illustrated in Figure 5, where the aforementioned conditional metrics of performance for resubstitution and leave-one-out are plotted versus conditioning estimated errors, for different bin sizes. In this example, we assume stratified sampling and the Zipf parametric model mentioned previously, with  $c_0 = c_1 = 0.5$  and parameter  $\alpha$  adjusted to yield  $E[\epsilon_n] \approx 0.25$ , which corresponds to intermediate classification difficulty. Sample size is fixed at  $n = 20$  to emphasize small-sample effects and two bin sizes are considered,  $b = 4, 8$ . The curves for the conditional expectation rise with the estimated error; they also exhibit the property that the conditional expected actual error is larger than the estimated error for small estimated errors and smaller than the estimated error for large estimated errors. A point to be noted is the flatness of the leave-one-out curves. This reflects the high variance of the leave-one-out estimator. Note that the 95% upper confidence bounds are nondecreasing with respect to increasing estimated error, as expected. The flat spots observed in the bounds result from the discreteness of the estimation rule (this phenomenon is more pronounced when the number of bins is smaller).

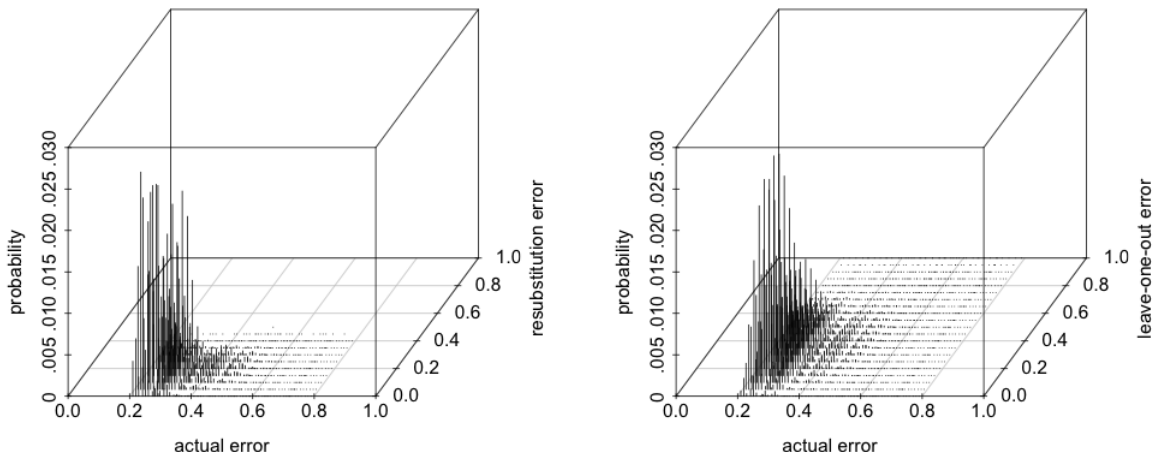


Figure 6: Exact joint distribution between the actual error and the resubstitution and leave-one-out cross-validation error estimators, for  $n = 20$  and  $b = 8$ , and a Zipf probability model of intermediate difficulty (Bayes error = 0.2).

## 5.4 Distribution-Free Analysis of Performance

Note that the model parameters  $p_i$  and  $q_i$  must be nonnegative and satisfy the constraints  $\sum_{i=1}^{b-1} p_i \leq 1$  and  $\sum_{i=1}^{b-1} q_i \leq 1$ . Each of these equations determines a *simplex*  $S_{b-1}$  in  $(b-1)$ -dimensional Euclidean space. Therefore, given the value of  $c_0 = P(Y = 0)$  (so that  $c_1 = 1 - c_0$  is also known), the discrete classification problem is completely determined by a vector of  $2(b-1)$  values, which must be a point in the *model space*  $\Pi(c_0) = S_{b-1} \times S_{b-1}$ .

In [29], G.F. Hughes provided exact expressions that allow the computation of the average Bayes error  $\bar{\epsilon}^*(b, c_0)$  and the average expected actual error  $\overline{E[\epsilon]}(n, b, c_0)$  for the discrete histogram rule, both averaged over the model space  $\Pi(c_0)$ , by assuming that all models in  $\Pi(c_0)$  are equally-likely to occur. This provides a distribution-free analysis of performance, some of the qualitative features of which are still valid in particular distributional settings. For example, one of the famous conclusions derived in [29] is that, with  $n$  and  $c_0$  fixed, the curve of the expected actual *accuracy*  $1 - \overline{E[\epsilon]}(n, b, c_0)$  as a function of number of bins  $b$  *peaks* around an optimal value  $b^*$ , which increases with increasing sample size  $n$ . Even though this result was derived in terms of the average accuracy over the model space, and for the discrete histogram rule, this “peaking phenomenon” is in fact observed for the majority of individual distributions, and indeed for the majority of classification rules, both discrete and continuous [28].

Using the expressions in Hughes’ paper, we plotted the average Bayes accuracy  $1 - \bar{\epsilon}^*(b, c_0)$  and average expected actual accuracy  $1 - \overline{E[\epsilon]}(n, b, c_0)$ , both as a function of  $b$ , for various values of  $n$ , assuming the balanced case  $c_0 = 0.5$ ; the results are displayed in Figure 8. The curves are plotted as a function of  $p = \log_2 b$ . This corresponds to the case where  $p$  binary predictors are used in the original feature space; for example, the point  $p = 5$  in the plot corresponds to 5 binary features, with  $b = 2^5 = 32$ . One can easily

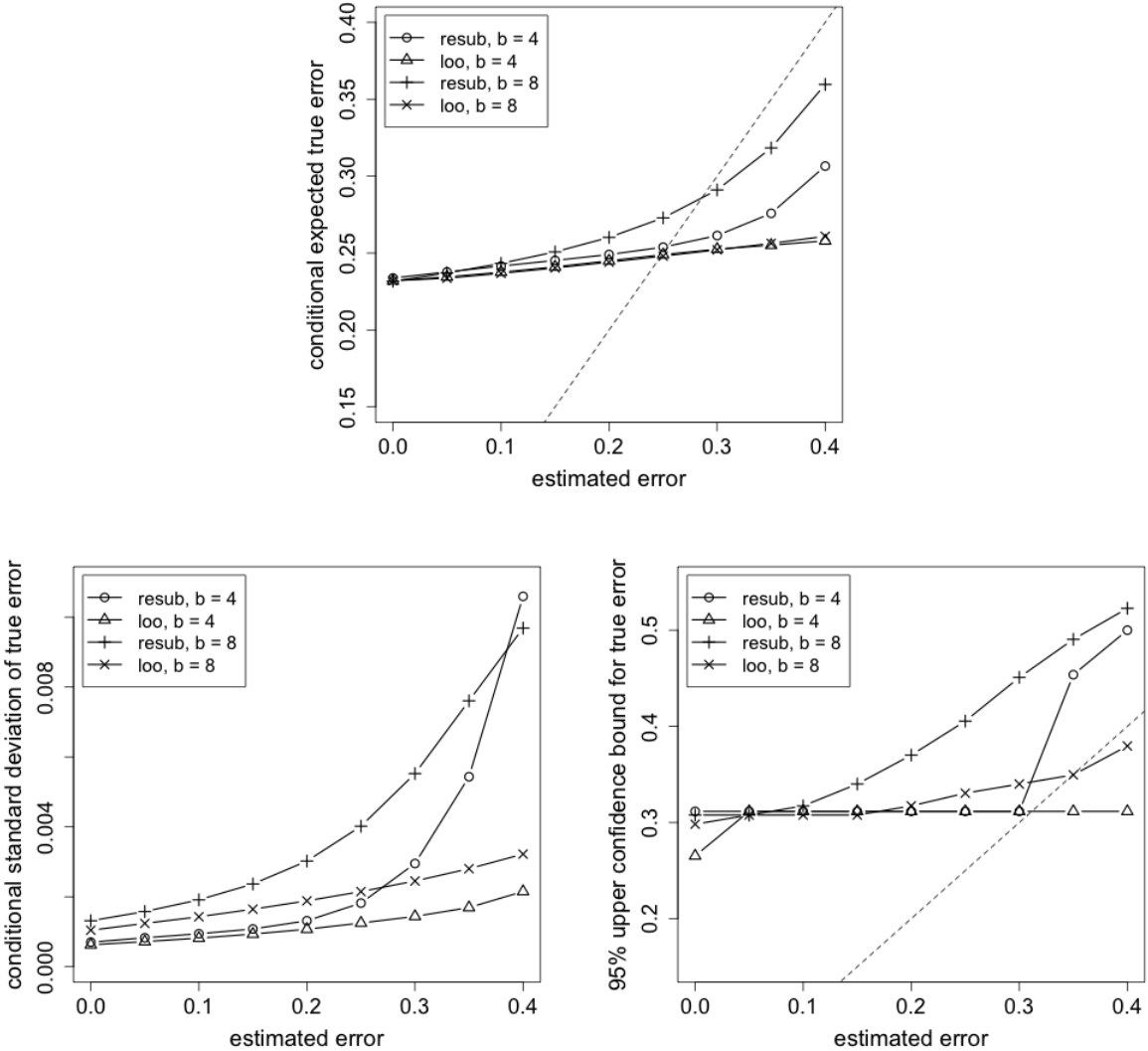


Figure 7: Exact conditional metrics of performance for resubstitution and leave-one-out error estimators. The dashed line indicates the  $y = x$  line.

observe the “peaking phenomenon” in this plot. The optimal number of features moves to the right with increasing sample size  $n$ , and, regardless of the value of  $n$ , accuracy tends to the no-information value of 0.5 as the number of predictors increases. Sample size computations can be performed based on the curves of Figure 8; for example, if one has  $p = 3$  binary predictors, so that  $b = 8$ , then sample size should be equal to  $n = 60$  at a minimum, according to this analysis. The expressions for these curves are quite complicated

and computationally intensive for large  $n$ ; however for small  $n$ , the expressions become quite simple. For example, with  $n = 2$ ,

$$1 - \overline{E[\epsilon]}(2, b, 0.5) = \frac{1}{2} + \frac{1}{2} \frac{b-1}{b(b+1)}$$

so that the accuracy margin over the no-information value of 0.5 vanishes as  $1/b$ . This implies that the decrease is exponential in  $p = \log_2(b)$ , as can be gleaned from Figure 8.

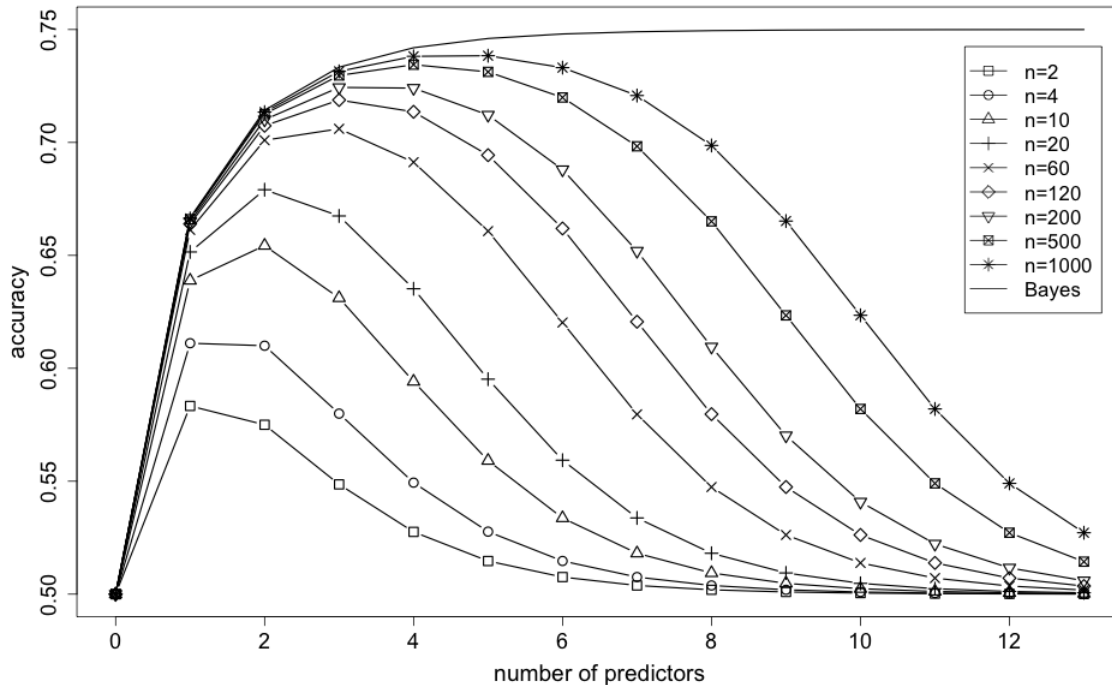


Figure 8: Average Bayes accuracy and average expected actual accuracy plotted as a function of number of binary predictors  $p = \log_2(b)$ .

Note that peaking ceases to occur as  $n \rightarrow \infty$ , which corresponds to the Bayes accuracy (see the next Section). This must be the case, since the Bayes accuracy is known to be nondecreasing in the number of features. The expression for the average Bayes accuracy in the case  $c_0 = 0.5$  is simple; as shown in [29], this is given by

$$1 - \overline{\epsilon^*}(b, 0.5) = \frac{3b-2}{4b-2}$$

with an asymptotic value (as  $b \rightarrow \infty$ ) of 0.75 (it is shown in [29] that for general  $c_0$ , this asymptotic value is equal to  $1 - c_0(1 - c_0)$ ). This relatively small value highlights the conservative character of Hughes' distribution-free approach; for example, in practice, where one deals with a fixed distribution of the data,

the optimal number of features would typically be larger than the ones observed in Figure 8, so that sample size recommendations based on this analysis tend to be pessimistic — a fact that was pointed out in [1]. Nevertheless, the qualitative behavior of the analysis is entirely correct. Finally, we remark that Figure 8 closely matches Figure 3 in [29], but larger values of  $b$  are shown here, which possibly were difficult to compute in 1968.

## 5.5 Performance of Ensemble Methods in Discrete Classification

In [9], Braga-Neto and Dougherty carried out an analysis of the performance of ensemble classification methods [10, 44] when applied to the discrete histogram rule, which provided evidence that such ensemble methods may be largely ineffective in discrete classification. Part of the analysis is similar to the work of Hughes', discussed in the previous subsection, in the sense that it examines the average performance over the model space, assuming equally-likely models. Two methods were considered, namely, the *jackknife* and *bagging* ensemble classification rules obtained from the discrete histogram rule. Briefly, ensemble methods are based on perturbing the training data, designing an ensemble of classifiers based on the perturbed data sets using a given base classification rule (in this case, the discrete histogram rule), and aggregating the individual decisions to obtain the final classifier. Data perturbation is often accomplished by resampling methods such as the jackknife [35] and bootstrap [17] — the latter case being known as “bagging” [10] — whereas aggregation is done by means of majority voting among the individual classifier decisions. For the jackknife majority-vote classification rule, it was shown in [9] that, under full sampling and equally-likely classes, the best gain in performance (i.e., decrease in expected classification error) over all models in the model space  $\Pi(c_0)$  is smaller than the worst deficit (i.e., increase in expected classification error). Any discrepancy in performance however disappears as sample size increases; in particular the following bound is shown to hold:

$$|E[\epsilon_n^J] - E[\epsilon_n]| \leq \frac{1}{ne} \sqrt{\frac{2}{\pi(n+1)}} \quad (19)$$

where  $E[\epsilon_n^J]$  and  $E[\epsilon_n]$  are the expected classification errors of the jackknife and base classification rules, respectively. In addition, an exact expression is given for the average  $\overline{E[\epsilon_n^J] - E[\epsilon_n]}$  over the model space  $\Pi(c_0)$ , assuming equally-likely distributions as in the work of Hughes. In the case of equally-likely classes ( $c_0 = 0.5$ ), the result simplifies to show that the average difference is positive, that is, there is an average deficit (which in fact is shown to still hold if the classes are only approximately equally likely, in a precise sense). The left plot in Figure 9 displays these quantities plotted as a function of sample size, for  $p = 2$  ( $b = 4$  discrete cells), and for the balanced case,  $c_0 = 0.5$ . We can observe in the plot that the best gain (inf) is smaller than the worst deficit (sup) and that there is an average deficit (positive average deviation). The values of inf and sup are actually independent of  $b$ .

Regarding the bagging case, it is shown in [9] that, given the training data, and for any sample size, number of cells, or distribution of the data, the random bagging classifier converges to the original discrete histogram classifier with probability 1 as the number of classifiers in the ensemble  $m$  increases, and, furthermore, it also gives the following exponential bound on the absolute difference  $|\epsilon_{n,m}^B - \epsilon_n|$  between the generalization errors of the bagging and the base classifiers,

$$|\epsilon_{n,m}^B - \epsilon_n| \leq e^{-2mc^2} \quad (20)$$

where the constant  $c > 0$  does not depend on  $m$ , but depends in a simple way on the distribution of the data. The difference therefore converges exponentially fast to zero as the number of classifiers in the ensemble

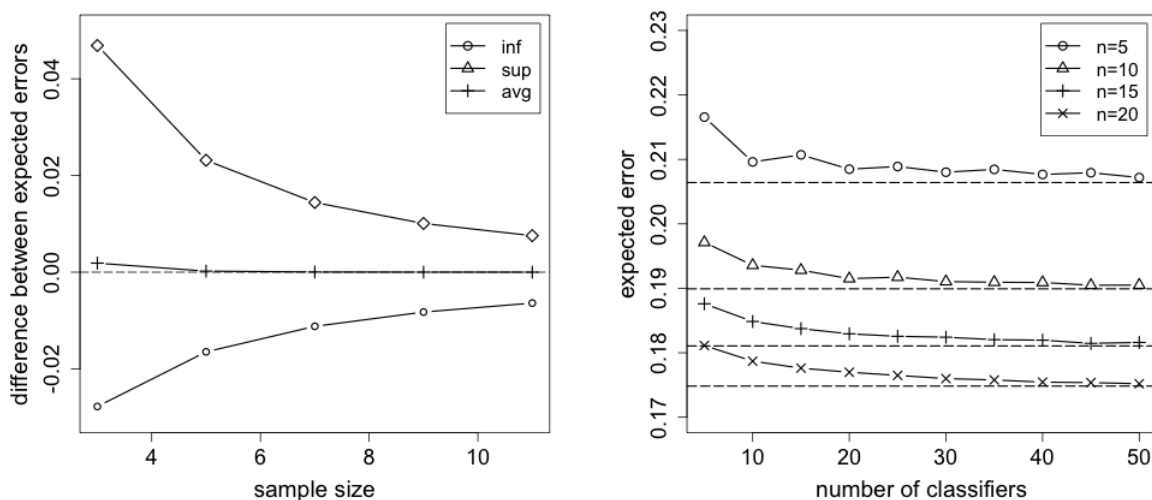


Figure 9: Performance of ensemble discrete classification rules, for  $p = 2$ . Left plot: bounds and average difference between expected errors of the jackknife and discrete histogram classification rules, as a function of sample size, for  $c_0 = 0.5$ . Right plot: expected error for the bagged discrete histogram classification rule, found by Monte-Carlo computation, as a function of number of classifiers in the ensemble, for model parameters derived from an actual data set. Also indicated are the exact expected errors of the base discrete histogram classification rule, by means of dashed horizontal lines, to which the expected error of the bagged classification rule in each case is clearly converging, as expected.

increases (for fixed  $n$ ). From this it follows that the difference between expected errors over all training data also converges exponentially fast to zero (the constant  $c$  is larger, guaranteeing faster convergence, if the classes are more separated, in a precise sense). The right plot in Figure 9 displays the expected error for the bagged discrete histogram classification rule as a function of number of classifiers in the ensemble, for model parameters derived from an actual data set, corresponding to  $p = 2$  binary features picked from the SPECT Heart data set of the UCI Machine Learning Repository. The expected classification error for the bagging classifier is found by means of a Monte-Carlo computation using 100,000 simulated training sets, assuming full sampling. The Monte-Carlo computation introduces the wobble visible in the plots (even at this very large number of simulated training sets). Also indicated are the exact expected errors of the base discrete histogram classification rule, by means of dashed horizontal lines. We can see that in all cases bagging leads to a larger expected classification error than the base classification rule, although the deviation quickly converges to zero in each case, in agreement with equation (20) above.

## 6 Large-Sample Performance of Discrete Classification

Large-sample analysis of performance has to do with behavior of classification error and error estimators as sample size increases without bound, i.e., as  $n \rightarrow \infty$ . From a practical perspective, one expects performance to improve, and eventually reach an optimum, as more time and cost is devoted to obtaining an increasingly large number of samples. It turns out that not only this is true for the discrete histogram rule, but also it is possible in several cases to obtain fast (exponential) rates of convergence. Critical results in this area are due to Cochran and Hopkins [11], Glick [21, 22], and Devroye, Györfi and Lugosi [13]. We will review briefly these results in this Section.

Recall the bin counts  $U_i$  and  $V_i$  introduced in Section 3. By a straightforward application of the Strong Law of Large Numbers (SLLN) [20], we obtain that  $U_i/n \rightarrow c_0 p_i$  and  $V_i/n \rightarrow c_1 q_i$  as  $n \rightarrow \infty$ , with probability 1. From this and eqs. (1) and (3), it follows immediately that

$$\lim_{n \rightarrow \infty} \psi_n(X = i) = \lim_{n \rightarrow \infty} I_{U_i < V_i} = I_{c_0 p_i < c_1 q_i} = \psi^*(X = i) \text{ with probability 1.} \quad (21)$$

that is, the discrete histogram classifier designed from sample data converges to the optimal classifier over each bin, with probability 1. This is a distribution-free result, so it is true regardless of the joint distribution of predictors  $X$  and target  $Y$ , as the SLLN itself is distribution-free. One says then that the discrete histogram rule is *universally strongly consistent* [13].

The exact same argument, in connection with eqs. (2), (5) and (8), shows that

$$\lim_{n \rightarrow \infty} \epsilon_n = \lim_{n \rightarrow \infty} \hat{\epsilon}_n^r = \epsilon^* \text{ with probability 1.} \quad (22)$$

so that the classification error, and also the apparent error, converge to the optimal Bayes error as sample size increases. From the previous equation it also follows that

$$\lim_{n \rightarrow \infty} E[\epsilon_n] = \lim_{n \rightarrow \infty} E[\hat{\epsilon}_n^r] = \epsilon^*, \quad (23)$$

In particular,  $\lim_{n \rightarrow \infty} E[\hat{\epsilon}_n^r - \epsilon_n] = 0$  and the bias of resubstitution vanishes with increasing sample size. Recalling (9), one always has  $E[\hat{\epsilon}_n^r] \leq \epsilon^* \leq E[\epsilon_n]$ , so that (23) in fact implies that  $E[\epsilon_n] \downarrow \epsilon^*$ , while  $E[\hat{\epsilon}_n^r] \uparrow \epsilon^*$ , as  $n \rightarrow \infty$ .

These results are all based on the SLLN (and are thus distribution-free). The question arises as to the speed with which the limits are attained, as the SLLN can yield notoriously slow rates of convergence. This

is not only a theoretical question, as the usefulness in practice of such results may depend on how large a sample size needs to be to guarantee that the discrete classifier or error estimator is close enough to optimality. The answer is that exponential rates of convergence can be obtained, if one is willing to drop the distribution-free requirement. Otherwise, polynomial rates of convergence can be established. These results are briefly reviewed below.

Regarding the discrete histogram rule, with a proviso that ties in bin counts are assigned a class randomly (with equal probability), it is shown in [22, Theorem A], that the following exponential bound on the convergence of  $E[\epsilon_n]$  to  $\epsilon^*$  applies

$$E[\epsilon_n] - \epsilon^* \leq \left(\frac{1}{2} - \epsilon^*\right) e^{-cn}, \quad (24)$$

where the constant  $c > 0$  is distribution-dependent:

$$c = \log \frac{1}{1 - \min_{\{i: c_0 p_i \neq c_1 q_i\}} |\sqrt{c_0 p_i} - \sqrt{c_1 q_i}|^2}$$

Interestingly, the number of bins does not figure in this bound. The speed of convergence of the bound is determined by the minimum (nonzero) difference between the probabilities  $c_0 p_i$  and  $c_1 q_i$  over any one cell. The larger this difference is, the larger  $c$  is, and the faster convergence is. Conversely, the presence of a single cell where these probabilities are close slows down convergence of the bound.

On the other hand, a distribution-free bound is provided by [13, Theorem 27.1]:

$$E[\epsilon_n] - \epsilon^* \leq \sqrt{\frac{b}{2(n+1)}} + \frac{b}{en} \leq 1.075 \sqrt{\frac{b}{n}} \quad (25)$$

This polynomial  $O(n^{-1/2})$  bound is inferior to the exponential bound in (24), but it does guarantee a fixed rate of convergence that is independent of the distribution.

Regarding convergence of  $E[\hat{\epsilon}_n^r]$  to  $\epsilon^*$ , and again assuming random tie-breaking over cells, it is shown in [22, Theorem B], that the following exponential bound applies

$$\epsilon^* - E[\hat{\epsilon}_n^r] \leq \frac{1}{2} b n^{-1/2} e^{-cn}, \quad (26)$$

*provided that* there is no cell over which  $c_0 p_i = c_1 q_i$ . Here, the constant  $c > 0$  is the same as in (24). The presence of a cell where  $c_0 p_i = c_1 q_i$  invalidates the bound in (26) and slows down convergence; in fact, it is shown in [22] that in such a case  $\epsilon^* - E[\hat{\epsilon}_n^r]$  has both upper and lower bounds that are  $O(n^{-1/2})$ , so that convergence *cannot* be exponential. Finally, observe that the bounds in (24) and (26) can be combined to bound the *bias* of resubstitution  $E[\hat{\epsilon}_n^r - \epsilon_n]$ . We can conclude, for example, that in case there are no cells over which  $c_0 p_i = c_1 q_i$ , convergence of the bias to zero is exponentially fast.

The previous results on the discrete histogram rule concern expectation and bias. In [13], (distribution-free) results on variance and RMS are also given, both for resubstitution and leave-one-out (here, the convention we have adopted of breaking ties in the direction of class 0 is again in effect). For the resubstitution error estimator, one has the following bounds [13, Theorem 23.3]:

$$\text{Var}[\hat{\epsilon}_n^r] \leq \frac{1}{n} \quad (27)$$

and

$$\text{RMS}(\hat{\epsilon}_n^r) \leq \sqrt{\frac{6b}{n}} \quad (28)$$

In particular, both quantities converge to zero as sample size increases. For the leave-one-out error estimator, one has the following bound [13, Theorem 24.7]:

$$\text{RMS}(\hat{\epsilon}_n^l) \leq \sqrt{\frac{1 + 6/e}{n} + \frac{6}{\sqrt{\pi(n-1)}}} \quad (29)$$

This guarantees, in particular, convergence to zero as sample size increases.

An important factor in the comparison of the resubstitution and leave-one-out error estimators for discrete histogram classification resides in the different speeds of convergence of the RMS to zero. Convergence of the RMS bound for the resubstitution estimator is  $O(n^{-1/2})$  (for fixed  $b$ ), whereas convergence of the RMS bound for the leave-one-out estimator is  $O(n^{-1/4})$ , thus slower. Now, as remarked in [13, p.419], it can be shown that for some distributions there is also a *lower bound* of kind  $O(n^{-1/4})$  on the RMS of leave-one-out. Therefore, in the worst case, the RMS of leave-one-out to zero is guaranteed to decrease as  $n^{-1/4}$ , and therefore is certain to decrease slower than the RMS of resubstitution. Note that the bad RMS of leave-one-out is due almost entirely to its large variance, typical of the cross-validation approach, since this estimator is essentially unbiased.

## 7 Binary Coefficient of Determination (CoD)

In classical regression analysis, the *coefficient of determination* (CoD) gives the relative decrease in unexplained variability when entering a variable  $X$  into the regression of the dependent variable  $Y$ , in comparison with the total unexplained variability when entering no variables:

$$\text{CoD}(X, Y) = \frac{SS_Y - SS_{XY}}{SS_Y} \quad (30)$$

where  $SS_Y$  and  $SS_{XY}$  are the sums of squared errors associated with entering no variables and entering variable  $X$  to predict  $Y$ , respectively. The term  $SS_Y$  is proportional to the total variance  $\sigma_Y^2$ , which is the error around the mean  $\mu_Y$  (so that entering no variables in the regression corresponds to using the mean as the predictor).

In classification, a very similar concept was introduced in [16]:

$$\text{CoD}(X, Y) = \frac{\epsilon_Y^* - \epsilon_{XY}^*}{\epsilon_Y^*}, \quad (31)$$

where  $\epsilon_Y^* = \min\{P(Y=0), P(Y=1)\}$  is the Bayes error in the absence of any features, and  $\epsilon_{XY}^*$  is the Bayes error when using feature vector  $X$  to predict  $Y$ . By convention, one assumes  $0/0 = 1$  in the above definition. This *binary coefficient of determination* measures the relative decrease in prediction error of a target variable when using predictor variables, relative to using no predictor variables; notice the remarkable similarity between (30) and (31).

The binary CoD was perhaps the first predictive paradigm utilized in the context of microarray data, the goal being to provide a measure of nonlinear interaction among genes [16]. Even though the binary CoD, as defined in (31), has general application in classification, it has been extensively used in the case of discrete classification and prediction, particularly in problems dealing with gene expression quantized into discrete levels [31, 46] — see the examples given in Section 2 — and its use in the inference of gene regulatory networks [36, 37]. As its classic counterpart, the binary CoD is a goodness-of-fit statistic that can be used

to assess the relationship between predictor and target variables (e.g., how tight the association between a set of predictor genes and a target gene is).

Even though the definition above employs Bayes errors, the CoD can be likewise defined in terms of the classification error of predictors designed from sample data, using for example the discrete histogram rule. In addition, the actual classification errors will typically need to be computed through error estimation techniques; e.g., one may speak of resubstitution and leave-one-out CoD estimates. All the issues discussed in previous sections regarding classification and error estimation for discrete data generally apply here.

A recent paper [34] defined and studied the concept of *intrinsically multivariate predictive* (IMP) genes using the binary CoD. Briefly, IMP genes are those the expression of which cannot be predicted well by any subset of binary predicting gene expressions, but is predicted very well by the entire set. In [34], the properties of IMP genes were characterized analytically, and it was shown that high-predictive power, small covariance among predictors, a large entropy of the joint probability distribution of predictors, and certain logics, such as XOR in the 2-predictor case, are factors that favor the appearance of IMP. In addition, quantized gene-expression microarray data were employed to show that the gene DUSP1, which exhibits control over a central, process-integrating signaling pathway, exhibits IMP behavior, thereby providing preliminary evidence that IMP can be used as a criterion for discovery of *canalizing* genes, i.e., master genes that constrain (“canalize”) large gene-expression pathways [15].

## 8 Conclusion

The importance of discrete classification in Genomics lies in its broad application in problems of phenotype classification based on panels of gene-expression biomarkers and inference of gene regulatory networks from gene-expression data, where data discretization is often employed for data efficiency and classification accuracy reasons. This paper presented a broad review of methods of classification and error estimation for discrete data, focusing for the most part on the discrete histogram rule, which is the classification rule most employed in practice for discrete data, due to its excellent properties, such as low complexity and small data requirement (under small number of cells), and universal consistency. The most important criterion for performance is the classification error, which can be computed exactly only if the underlying distribution of the data is known. In practice, robust error estimation methods must be employed to obtain reliable estimates of the classification error based on available sample data. This paper reviewed analytical and empirical results concerning the performance of discrete classifiers (in terms of the classification error) as well as of error estimators for discrete classification. Those results were categorized into small-sample results — small-sample data being prevalent in Genomics applications — and large-sample (i.e., asymptotic) results. The binary Coefficient of Determination was also reviewed briefly; it provides a measure of nonlinear interaction among genes and is therefore very useful in the inference of gene regulatory networks. Progress in classification and error estimation for discrete data, particularly the analysis of performance in small-sample cases, has a clear potential to lead to genuine advances in Genomics and Medicine, and therefore the study of such methods is a topic of considerable research interest at present.

## Acknowledgements

This work was supported by the National Science Foundation, through NSF award CCF-0845407.

## References

- [1] K. Abend, Jr. T.J. Harley, B. Chandrasekaran, Jr. T.J. Harley, and G.F. Hughes. Comments “on the mean accuracy of statistical pattern recognizers”. *IEEE Transactions on Information Theory*, IT-15(3):420–423, 1969.
- [2] A. Agresti. A survey of exact inference for contingency tables. *Statistical Science*, 7(1):131–153, 1992.
- [3] A. Agresti. *Categorical Data Analysis*. Wiley, New York, NY, 2nd edition, 2002.
- [4] U.M. Braga-Neto. An asymptotically-exact expression for the variance of classification error for the discrete histogram rule, 2008. GENSIPS’2008 - IEEE International Workshop on Genomic Signal Processing and Statistics, Phoenix, AZ.
- [5] U.M. Braga-Neto and E.R. Dougherty. Is cross-validation valid for microarray classification? *Bioinformatics*, 20(3):374–380, 2004.
- [6] U.M. Braga-Neto and E.R. Dougherty. Classification. In E. Dougherty, I. Shmulevich, J. Chen, and Z. J. Wang, editors, *Genomic Signal Processing and Statistics*, EURASIP Book Series on Signal Processing and Communication. Hindawi Publishing Corporation, 2005.
- [7] U.M. Braga-Neto and E.R. Dougherty. Exact performance of error estimators for discrete classifiers. *Pattern Recognition*, 38(11):1799–1814, 2005.
- [8] U.M. Braga-Neto and E.R. Dougherty. Exact correlation between actual and estimated errors in discrete classification, 2009. Submitted.
- [9] U.M. Braga-Neto and E.R. Dougherty. Performance of ensemble methods in discrete classification, 2009. Submitted.
- [10] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
- [11] W.G. Cochran and C.E. Hopkins. Some classification problems with multivariate qualitative data. *Biometrics*, 17(1):10–32, 1961.
- [12] T. Cover. Learning in pattern recognition. In S. Watanabe, editor, *Methodologies of Pattern Recognition*, pages 111–132. Academic Press, New York, NY, 1969.
- [13] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, New York, 1996.
- [14] E.R. Dougherty and U.M. Braga-Neto. Epistemology of computational biology: Mathematical models and experimental prediction as the basis of their validity. *Journal of Biological Systems*, 14(1):65–90, 2006.
- [15] E.R. Dougherty, M. Brun, J.M. Trent, and M.L. Bittner. A conditioning-based model of contextual regulation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2008. Preprint available on line.
- [16] E.R. Dougherty, S. Kim, and Y. Chen. Coefficient of determination in nonlinear signal processing. *EURASIP Journal on Signal Processing*, 80(10):2219–2235, 2000.

- [17] B. Efron. Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 7:1–26, 1979.
- [18] B. Efron. Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, 78(382):316–331, 1983.
- [19] B. Efron and R. Tibshirani. Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560, 1997.
- [20] W. Feller. *An Introduction to Probability Theory and Its Applications. Vol. 1*. Wiley, New York, NY, 1968.
- [21] N. Glick. Sample-based classification procedures derived from density estimators. *Journal of the American Statistical Association*, 67(337):116–122, 1972.
- [22] N. Glick. Sample-based multinomial classification. *Biometrics*, 29(2):241–256, 1973.
- [23] M. Goldstein and W.R. Dillon. *Discrete Discriminant Analysis*. Wiley, New York, 1978.
- [24] B. Hanczar, J. Hua, and E.R. Dougherty. Decorrelation of the true and estimated classifier errors in high-dimensional settings. *EURASIP Journal on Bioinformatics and Systems Biology*, 2007, 2007. Article ID 38473, 12 pages.
- [25] M. Hills. Allocation rules and their error rates. *Journal of the Royal Statistical Society. Series B (Methodological)*, 28(1):1–31, 1966.
- [26] M. Hills. Discrimination and allocation with discrete data. *Applied Statistics*, 16(3):237–250, 1967.
- [27] K.F. Hirji, C.R. Mehta, and N.R. Patel. Computing distributions for exact logistic regression. *Journal of the American Statistical Association*, 82(400):1110–1117, 1987.
- [28] J. Hua, Z. Xiong, J. Lowey, E. Suh, and E.R. Dougherty. Optimal number of features as a function of sample size for various classification rules. *Bioinformatics*, 21(8):1509–1515, 2005.
- [29] G.F. Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, IT-14(1):55–63, 1968.
- [30] G.F. Hughes. Number of pattern classifier design samples per class. *IEEE Transactions on Information Theory*, IT-15(5):615–618, 1969.
- [31] S. Kim, E.R. Dougherty, Y. Chen, K. Sivakumar, P. Meltzer, J. Trent, and M.L. Bittner. Multivariate measurement of gene expression relationships. *Genomics*, 67(2):201–209, 2000.
- [32] J.H. Klotz. The wilcoxon, ties, and the computer. *Journal of the American Statistical Association*, 61(315):772–787, 1966.
- [33] P.A. Lachenbruch and M.R. Mickey. Estimation of error rates in discriminant analysis. *Technometrics*, 10:1–11, 1968.
- [34] D. Martins, U.M. Braga-Neto, R. Hashimoto, M.L. Bittner, and E.R. Dougherty. Intrinsically multivariate predictive genes. *IEEE Journal of Selected Topics in Signal Processing*, 2(3):424–439, 2008.

- [35] M.H. Quenouille. Approximate tests of correlation in time series. *Journal of the Royal Statistical Society. Series B (Methodological)*, 11:68–84, 1949.
- [36] I. Schmulevich, E.R. Dougherty, S. Kim, and W. Zhang. Probabilistic Boolean Networks: a rule-based uncertainty model for gene-regulatory networks. *Bioinformatics*, 18:261–274, 2002.
- [37] I. Schmulevich, E.R. Dougherty, and W. Zhang. From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proceedings of the IEEE*, 90:1778–1792, 2002.
- [38] I. Shmulevich and W. Zhang. Binary analysis and optimization-based normalization of gene expression data. *Bioinformatics*, 18(4):555–565, 2002.
- [39] C.A.B. Smith. Some examples of discrimination. *Annals of Eugenics*, 18:272–282, 1947.
- [40] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36:111–147, 1974.
- [41] G.T. Toussaint and R. Donaldson. Algorithms for recognizing contour-traced hand-printed characters. *IEEE Transactions on Computers*, 19:541–546, 1970.
- [42] A. Verbeek. A survey of algorithms for exact distributions of test statistics in rxc contingency tables with fixed margins. *Computational Statistics and Data Analysis*, 3:159–185, 1985.
- [43] I.H. Witten and E. Frank. *Data Mining*. Academic Press, San Diego, CA, 2000.
- [44] L. Xu, A. Krzyzak, and C.Y. Suen. Methods for combining multiple classifiers and their applications in handwritten character recognition. *IEEE Trans. Systems, Man, and Cybernetics*, 22:418–435, 1992.
- [45] Q. Xu, J. Hua, U.M. Braga-Neto, Z. Xiong, E. Suh, and E.R. Dougherty. Confidence intervals for the true classification error conditioned on the estimated error. *Technology in Cancer Research and Treatment*, 5(6):579–590, 2006.
- [46] X. Zhou, X. Wang, and E.R. Dougherty. Binarization of microarray data based on a mixture model. *Molecular Cancer Therapeutics*, 2(7):679–684, 2003.