Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 487893, 13 pages
doi:10.1155/2010/487893

Research Article

Exact Performance of CoD Estimators in Discrete Prediction

Ting Chen and Ulisses Braga-Neto

Department of Electrical Engineering, Texas A&M University, College Station, TX 77843, USA

Correspondence should be addressed to Ulisses Braga-Neto, ulisses@ece.tamu.edu

Received 1 April 2010; Accepted 9 July 2010

Academic Editor: Haris Vikalo

The coefficient of determination (CoD) has significant applications in genomics, for example, in the inference of gene regulatory networks. We study several CoD estimators, based upon the resubstitution, leave-one-out, cross-validation, and bootstrap error estimators. We present an exact formulation of performance metrics for the resubstitution and leave-one-out CoD estimators, assuming the discrete histogram rule. Numerical experiments are carried out using a parametric Zipf model, where we compute exact performance metrics of resubstitution and leave-one-out CoD estimators using the previously derived equations, for varying actual CoD, sample size, and bin size. These results are compared to approximate performance metrics of 10-repeated 2-fold cross-validation and 0.632 bootstrap CoD estimators, computed via Monte Carlo sampling. The numerical results lead to a perhaps surprising conclusion: under the Zipf model under consideration, and for moderate and large values of the actual CoD, the resubstitution CoD estimator is the least biased and least variable among all CoD estimators, especially at small number of predictors. We also observed that the leave-one-out and cross-validation CoD estimators tend to perform the worst, whereas the performance of the bootstrap CoD estimator is intermediary, despite its high computational complexity.

1. Introduction

The coefficient of determination (CoD) has significant applications in genomics, for example, in the inference of gene regulatory networks. We study several CoD estimators, based upon theresubstitution, leave-one-out, cross-validation, and bootstrap error estimators. We present an exact formulation of performance metrics for the resubstitution and leave-one-out CoD estimators, assuming the discrete histogram rule. Numerical experiments are carried out using aparametric Zipf model, where we compute exact performance metrics of resubstitution and leave-one-out CoD estimators using the previously derived equations, for varying actual CoD, sample size, and bin size. These results are compared to approximate performance metrics of10-repeated 2-fold cross-validation and 0.632 bootstrap CoD estimators, computed via Monte Carlo sampling. The numerical results lead to a perhaps surprising conclusion: under the Zipf model under consideration, and for moderate and large values of the actual CoD,the resubstitution CoD estimator is the least biased and least variable among all CoD estimators, especially at small number of predictors. We also observed that the leave-one-out andcross-validation CoD estimators tend to perform the worst whereas the performance of the bootstrap CoD estimator is intermediary, despite its high computational complexity.

In classical regression analysis, the nonlinear 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. Applying this to pattern prediction, Dougherty and collaborators [1] introduced a very similar concept, that of CoD for binary random variables, which measures the predictive power of a set of predictor variables $\mathbf{X} = \{X_1, X_2, \ldots, X_n\} \in \{0,1\}^n$ with respect to a target variable $Y \in \{0,1\}$, as given by

$$\text{CoD} = \frac{\varepsilon_0 - \varepsilon}{\varepsilon_0}, \tag{1}$$

where ε_0 is the error of the best predictor of Y in the absence of other observations and ε is the error of the best predictor of Y based on the observation of $\mathbf{X}$. The binary CoD measures the relative decrease in prediction error when using predictor variables to estimate the target variable, as opposed to using no predictor variables. The closer it is to one, the tighter the regulation of the target variable by the predictor variables is, whereas the closer it is to zero, the looser the regulation is. The CoD will correctly produce low values in cases where the no-predictor error is already small, or when adding predictors does not contribute to a significant decrease in error. The CoD is a function only of the joint distribution between predictors and target, thus it characterizes the regulatory relationship among them.

The concept of CoD has far-reaching applications in Genomics. The 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 [1–6]. In [2, 4, 6], the CoD is applied to the prediction problem dealing with gene expressions quantized into discrete levels in discrete prediction. In [3, 5], the CoD has its application in the reconstruction or inference of gene regulatory networks. 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, for example, the associations between gene expression patterns in practical applications. The CoD permits biologists to focus on particular connections in the genome, and the estimated coefficients provide a practical criterion for selecting among potential predictor sets [1].

The error of the best predictor corresponds to the optimal prediction error, also known as Bayes error, which depends only on the underlying probability model [7]. However, in practical real-world problems, the underlying probability model is unknown, and thus we arrive at the fundamental issue of how to find a good prediction error estimator in small-sample settings [8, 9]. An error estimator may be a deterministic function of the sample data, in which case it is called a nonrandomized error estimator; such popular error estimators as resubstitution and leave-one-out are examples. These error estimators are random only through the random sample data. Closed-form analytical expressions for performance metrics such as bias, deviation variance, and RMS of resubstitution and leave-one-out error estimators have been given in [9, 10]. By contrast, randomized error estimators, like cross-validation and bootstrap, have "internal" random factors that affect their outcome, and thus approximate approaches, usually via Monte Carlo sampling, are typically used to analyze their performance.

Likewise, the CoD must in practice be estimated from sample data. A CoD estimator is obtained from (1) by using one of the usual error estimators for the prediction error with variables ε, and the empirical frequency estimator for the prediction error with no variables ε_0; we may speak thus of non-randomized CoD estimators, including the resubstitution and leave-one-out CoD estimators, and randomized CoD estimators, including bootstrap and cross-validation CoD estimators. The CoD with the true values of ε and ε_0 in (1) will be called in this paper the "actual CoD." We will employ the discrete histogram rule [7, 8], the most widely used and intuitive rule for discrete prediction problems, in order to estimate prediction errors and CoDs from the sample data.

This paper presents, for the first time, an exact formulation for performance metrics of the resubstitution and leave-one-out CoD estimators, for the discrete histogram rule. Numerical experiments are carried out using a parametric Zipf model, where we compute the exact performance of resubstitution and leave-one-out CoD estimators using the previously derived equations, for varying actual CoD, sample size, and bin size. We compare these results to approximate performance metrics of randomized CoD estimators (bootstrap and cross-validation), computed via Monte Carlo sampling. The numerical results indicate that, under the Zipf model under consideration, and for moderate and large values of the actual CoD, the resubstitution CoD estimator is the least biased and least variable among all CoD estimators, especially at small number of predictors. In fact, with two predictors, the resubstitution CoD nearly dominates uniformly over all other estimators across all values of actual CoD. The leave-one-out and cross-validation CoD estimator tend to perform the worst whereas the performance of the bootstrap CoD estimator is intermediary, despite its high computational complexity. This indicates that provided one has evidence of moderate to tight regulation between the genes, and the number of predictors is not too large, one should use the CoD estimator based on resubstitution.

This paper is organized as follows. In section 2, the probability model used in discrete prediction is introduced. In section 3, the discrete histogram rule is recalled, and formal definitions are given for the actual CoD and several CoD estimators, including two non-randomized CoD estimators (i.e., resubstitution and leave-one-out) and two randomized CoD estimators (i.e., .632 bootstrap and 10-repeated 2-fold cross-validation). Section 4 introduces performance metrics (i.e., bias, deviation variance, RMS) of a CoD estimator, and Section 5 presents an analytical formulation of exact performance metrics of the resubstitution and leave-one-out CoD estimators. In Section 6, we present numerical results, based on a parametric Zipf model, that compare the performance metrics of all the CoD estimators considered in this paper. Finally, Section 7 presents concluding remarks.

2. Discrete Prediction

Let $X_1, X_2, \ldots, X_p$ be p predictor random variables, such that each X_i take on a finite number b_i of values, and $Y \in \{0, 1\}$ be the target random variable, for the discrete prediction problem. The predictors as a group can take on values in a finite space with $b = \prod_{i=1}^{p} b_i$ possible states. For analysis purposes, we establish a bijection between this finite state space and a single predictor variable X taking values in the set $X \in \{1, 2, \ldots, b\}$. The variable X has a one-to-one relationship with the finite space state coded by $X_1, X_2, \ldots, X_p$: one specific value of X represents a specific combination of the values of the original predictors, that is, a "bin" into which the data is categorized. The value b is the number of bins, which provides a direct measure of predictor complexity.

The probability model for the pair (X, Y) is specified by class prior probabilities: $c_0 = P(Y = 0)$, $c_1 = P(Y = 1)$, and class-conditional probabilities: $p_i = P(X = i \mid Y = 0)$ and $q_i = P(X = i \mid Y = 1)$, for $i = 1, \ldots, b$, where we have the identities

$$c_0 + c_1 = 1, \qquad \sum_{i=1}^{b} p_i = 1, \qquad \sum_{i=1}^{b} q_i = 1. \tag{2}$$

Given a specific probability model, the optimal predictor for the problem is given by

$$\psi(X = i) = \begin{cases} 1, & c_1 q_i > c_0 p_i, \\ 0, & \text{o.w} \end{cases} \tag{3}$$

with optimal error rate, also called the Bayes error [7], determined by

$$\varepsilon = \sum_{i=1}^{b} \min\{c_0 p_i, c_1 q_i\}. \tag{4}$$

If no features are provided, the optimal error rate becomes

$$\varepsilon_0 = \min\{c_0, c_1\}. \tag{5}$$

By using the simple inequality $\sum \min\{a_i, b_i\} \le \min\{\sum a_i, \sum b_i\}$, one concludes that $\varepsilon \le \varepsilon_0$ in all cases.

The coefficient of determination [1] is defined as (assuming that $\varepsilon_0 \neq 0$)

$$\text{CoD} = \frac{\varepsilon_0 - \varepsilon}{\varepsilon_0} = 1 - \frac{\varepsilon}{\varepsilon_0} = 1 - \frac{\sum_{i=1}^{b} \min\{c_0 p_i, c_1 q_i\}}{\min\{c_0, c_1\}}. \tag{6}$$

Since $0 \le \varepsilon \le \varepsilon_0$, we have that $0 \le \text{CoD} \le 1$. We have $\text{CoD} = 1$ if and only if $\varepsilon = 0$, that is, there is perfect regulation between predictors and target. On the other hand, $\text{CoD} = 0$ if and only if $\varepsilon = \varepsilon_0$, that is, the predictors exert no regulation on the target.

3. CoD Estimation

In practice, the underlying probability model is unknown, and thus the CoD is not known. The need arises thus to find estimators of the CoD from i.i.d. sample data $S_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ drawn from the unknown probability model distribution. All CoD estimators considered here will be of the form

$$\widehat{\text{CoD}} = \frac{\hat{\varepsilon}_0 - \hat{\varepsilon}}{\hat{\varepsilon}_0} = 1 - \frac{\hat{\varepsilon}}{\hat{\varepsilon}_0}, \tag{7}$$

where $\hat{\varepsilon}$ is one of the usual error estimators for a selected discrete prediction rule, and $\hat{\varepsilon}_0$ is the empirical frequency estimator for the prediction error with no variables

$$\hat{\varepsilon}_0 = \min\left\{\frac{N_0}{n}, \frac{N_1}{n}\right\}, \tag{8}$$

where N_0 and N_1 are random variables corresponding to the number of sample points belonging to classes $Y = 0$ and $Y = 1$, respectively. We assume throughout that $\hat{\varepsilon}_0 \neq 0$, that is, each class is represented by at least one sample. Note that $\hat{\varepsilon}_0$ has the desirable property of being a universally consistent estimator of ε_0 in (5), that is, $\hat{\varepsilon}_0 \to \varepsilon_0$ in probability (in fact, almost surely) as $n \to \infty$, regardless of the probability model.

The discrete prediction rule to be used with the error estimator $\hat{\varepsilon}$ is the discrete histogram rule, which is the "plug-in" rule for approximating the minimum-error Bayes predictor [9]. Even though we make this choice, we remark that the methods described here can be applied to any discrete prediction rule. Given the sample data S_n, the discrete histogram classifier is given by

$$\psi_n(X = i) = I_{V_i > U_i} = \begin{cases} 1, & V_i > U_i, \\ 0, & U_i \ge V_i, \end{cases} \quad i = 1, 2, \ldots, b, \tag{9}$$

where U_i is the number of samples with $Y = 0$ in bin $X = i$, and V_i is the number of samples with $Y = 1$ in bin $X = i$, for $i = 1, \ldots, b$.

We review next some facts about the distribution of the random vectors $\mathbf{U} = \{U_1, \ldots, U_b\}$ and $\mathbf{V} = \{V_1, \ldots, V_b\}$, which will be needed in the sequel. The variables $N_0 = \sum_{i=1}^{b} U_i$, $N_1 = \sum_{i=1}^{b} V_i$, U_i, and V_i, for $i = 1, \ldots, b$, are random variables due to the randomness of the sample data S_n (this is the case referred to as "full sampling" in [9]). More specifically, N_i is a random variable binomially distributed with parameters (n, c_i), that is, $N_i \sim B(n, c_i)$, for $i = 0, 1$, while the vector-valued random variable (U_i, V_i) is trinomially distributed with the parameter set $(n, c_0 p_i, c_1 q_i)$, that is,

$$P(U_i = k, V_i = l) = \binom{n}{k, l, n-k-l} (c_0 p_i)^k (c_1 q_i)^l \times (1 - c_0 p_i - c_1 q_i)^{n-k-l}, \tag{10}$$

for $i = 1, \ldots, b$. In addition, the vector $\{U_1, \ldots, U_b, V_1, \ldots, V_b\}$ follows a multinomial distribution with parameters $(n, c_0 p_1, \ldots, c_0 p_b, c_1 q_1, \ldots, c_1 q_b)$, so that

$$P(U_1 = u_1, \ldots, U_b = u_b, V_1 = v_1, \ldots, V_b = v_b) = \binom{n}{u_1, \ldots, u_b, v_1, \ldots, v_b} \times (c_0 p_1)^{u_1} \ldots (c_0 p_b)^{u_b} (c_1 q_1)^{v_1} \ldots (c_1 q_b)^{v_b}. \tag{11}$$

We introduce next each of the CoD estimators considered in this paper.

3.1. Resubstitution CoD Estimator. This corresponds to the choice of resubstitution [11] as the prediction error estimator

$$\widehat{\text{CoD}}_r = 1 - \frac{\hat{\varepsilon}_r}{\hat{\varepsilon}_0}, \tag{12}$$

where, for the discrete histogram predictor,

$$\hat{\varepsilon}_r = \frac{1}{n}\sum_{i=1}^{b}\left[U_i I_{V_i > U_i} + V_i I_{U_i \geq V_i}\right]. \tag{13}$$

The resubstitution CoD can be written equivalently as

$$\widehat{\mathrm{CoD}}_r = 1 - \frac{\sum_{i=1}^{b} \min\{(N_0/n)(U_i/N_0), (N_1/n)(V_i/N_1)\}}{\min\{(N_0/n), (N_1/n)\}}, \tag{14}$$

which reveals that $\widehat{\mathrm{CoD}}_r$ has the desirable property of being a universally consistent estimator of CoD in (6), that is, $\widehat{\mathrm{CoD}}_r \rightarrow$ CoD in probability (in fact, almost surely) as $n \rightarrow \infty$, regardless of the probability model.

3.2. Leave-One-Out CoD Estimator. This corresponds to the choice of the leave-one-out error estimator [12] as the prediction error estimator

$$\widehat{\mathrm{CoD}}_l = 1 - \frac{\hat{\varepsilon}_l}{\hat{\varepsilon}_0}, \tag{15}$$

where, for the discrete histogram predictor (as can be readily checked)

$$\hat{\varepsilon}_l = \frac{1}{n}\sum_{i=1}^{b}\left[U_i I_{V_i \geq U_i} + V_i I_{U_i \geq V_i - 1}\right]. \tag{16}$$

The leave-one-out CoD estimator provides an opportunity to reflect on the uniform choice of the empirical frequency estimator $\hat{\varepsilon}_0$ in (8) as an estimator of ε_0, including here. Clearly, the empirical frequency corresponds to the resubstitution estimator of ε_0. The question arises as to whether, for the leave-one-out CoD estimator, the leave-one-out error estimator of ε_0 should be used instead. For $N_0 = N_1 = n/2$, we get $\hat{\varepsilon}_0 = 1/2$ with the choice of the resubstitution estimator (empirical frequency), but $\hat{\varepsilon}_0 = 1$ with the choice of leave-one-out estimator, which is a useless result. Similar problems beset other estimators of ε_0. Hence, the empirical frequency estimator is employed here as the estimator of ε_0 for all CoD estimators.

3.3. Cross-Validation CoD Estimator. This corresponds to the choice of the cross-validation error estimator [12, 13] as the prediction error estimator. In k-fold cross-validation, sample data S_n is partitioned into k folds S_i, for $i = 1, \ldots, k$. For simplicity, we assume that k can divide n. A classifier ψ_i is designed on the training set $S_n \setminus S_i$, and tested on S_i, for $i = 1, \ldots, k$. Since there are different partitions of the data into k folds, one can repeat the k-fold cross-validation r times and then average the results. Such a process leads to the r-repeated k-fold cross-validation error estimator $\hat{\varepsilon}_{cv}$, given by

$$\hat{\varepsilon}_{cv} = \frac{1}{nr}\sum_{m=1}^{r}\sum_{i=1}^{k}\sum_{j=1}^{n/k}\left| Y_j^{i,m} - \psi_{i,m}\left(X_j^{i,m}\right)\right|, \tag{17}$$

where $(X_j^{i,m}, Y_j^{i,m})$ represents the jth sample point in the ith fold for the m-th repetition of the cross-validation, for $i = 1, \ldots, k$, $m = 1, \ldots, r$ and $j = 1, \ldots, n/k$.

Based upon (17), the r-repeated k-fold cross-validation CoD estimator is defined by

$$\widehat{\mathrm{CoD}}_{cv} = 1 - \frac{\hat{\varepsilon}_{cv}}{\hat{\varepsilon}_0}. \tag{18}$$

In order to get reasonable variance properties, a large number of repetitions may be required, which can make the cross-validation CoD estimator slow to compute.

3.4. Bootstrap CoD Estimator. This corresponds to the use of the bootstrap [14, 15] for the prediction error estimator. A bootstrap sample $S_n^* = \{(X_1^*, Y_1^*), \ldots, (X_n^*, Y_n^*)\}$ consists of n equally-likely draws with replacement from the original data S_n. Some sample points from the original data may appear multiple times in the bootstrap sample whereas other sample points may not appear at all. The actual proportion of times a sample point (X_i, Y_i) appears in S_n^* can be written as $P_i^* = (1/n)\sum_{j=1}^{n} I_{(X_j^*, Y_j^*) = (X_i, Y_i)}$, for $i = 1, \ldots, n$. A predictor ψ_t may be designed on a bootstrap sample S_n^{*t}, and tested on $S_n \setminus S_n^{*t}$, for $t = 1, \ldots, T$, where T is a sufficiently large number of repetitions (in this paper, $T = 100$). Then, the basic bootstrap zero estimator is given by

$$\hat{\varepsilon}_{\mathrm{ZERO}} = \frac{\sum_{t=1}^{T}\sum_{i=1}^{n}\left| Y_i - \psi_t(X_i)\right| I_{P_i^{*t}=0}}{\sum_{t=1}^{T}\sum_{i=1}^{n} I_{P_i^{*t}=0}}. \tag{19}$$

The .632 bootstrap estimator then performs a weighted average of the bootstrap zero and resubstitution estimators

$$\hat{\varepsilon}_{b632} = (1 - 0.632)\hat{\varepsilon}_r + 0.632\ \hat{\varepsilon}_{\mathrm{ZERO}}. \tag{20}$$

Based on (19) and (20), the 0.632 bootstrap CoD estimator is then defined as

$$\widehat{\mathrm{CoD}}_{b632} = 1 - \frac{\hat{\varepsilon}_{b632}}{\hat{\varepsilon}_0}. \tag{21}$$

The bootstrap CoD estimator can be very slow to compute due to the complexity of $\hat{\varepsilon}_{\mathrm{ZERO}}$.

4. Performance Metrics of CoD Estimators

In analogous fashion to the performance metrics of prediction error estimators [8], the key performance metrics for an CoD estimator $\widehat{\mathrm{CoD}}$ are its bias

$$\mathrm{Bias}\left[\widehat{\mathrm{CoD}}\right] = E\left[\widehat{\mathrm{CoD}} - \mathrm{CoD}\right] = E\left[\widehat{\mathrm{CoD}}\right] - \mathrm{CoD}, \tag{22}$$

the deviation variance (which in the present case is equal simply to its variance)

$$\mathrm{Var_d}\left[\widehat{\mathrm{CoD}}\right] = \mathrm{Var}\left(\widehat{\mathrm{CoD}} - \mathrm{CoD}\right) = \mathrm{Var}\left(\widehat{\mathrm{CoD}}\right), \quad (23)$$

and the root mean-square (RMS) error

$$\begin{aligned}\mathrm{RMS}\left[\widehat{\mathrm{CoD}}\right] &= \sqrt{E\left[\left(\widehat{\mathrm{CoD}} - \mathrm{CoD}\right)^2\right]} \\ &= \sqrt{\mathrm{Var}\left[\widehat{\mathrm{CoD}}\right] + \mathrm{B}ias\left[\widehat{\mathrm{CoD}}\right]^2}.\end{aligned} \quad (24)$$

For a given probability model, all the performance metrics are thus obtained as a function of the expectation $E[\widehat{\mathrm{CoD}}]$ and variance $\mathrm{Var}(\widehat{\mathrm{CoD}})$.

Working further, we obtain

$$E\left[\widehat{\mathrm{CoD}}\right] = 1 - E\left[\frac{\hat{\varepsilon}}{\hat{\varepsilon}_0}\right], \quad (25)$$

$$\begin{aligned}\mathrm{Var}\left[\widehat{\mathrm{CoD}}\right] &= E\left[\left(\widehat{\mathrm{CoD}}\right)^2\right] - \left(E\left[\widehat{\mathrm{CoD}}\right]\right)^2 \\ &= E\left[\frac{\hat{\varepsilon}^2}{\hat{\varepsilon}_0^2}\right] - \left(E\left[\frac{\hat{\varepsilon}}{\hat{\varepsilon}_0}\right]\right)^2,\end{aligned} \quad (26)$$

as can be easily checked. We conclude that all the key performance metrics for CoD estimators can be obtained from the first and second moments of $\hat{\varepsilon}/\hat{\varepsilon}_0$.

5. Exact Moments of Nonrandomized CoD Estimators

As mentioned in the Introduction, we can categorize CoD estimators into non-randomized and randomized, depending on whether the prediction error estimator $\hat{\varepsilon}$ is non-randomized or randomized. Non-randomized CoD estimators, such as the resubstitution and leave-one-out CoD estimators, are deterministic functions of the sample data, which makes it possible an analytical formulation of their performance metrics. On the other hand, the performance of randomized CoD estimators, such as the cross-validation and bootstrap CoD estimators, is very difficult to study analytically and is typically investigated via Monte Carlo sampling (which is done in Section 6).

In this section, we will present exact expressions for the computation of the first moment $E[\hat{\varepsilon}/\hat{\varepsilon}_0]$ and the second moment $E[\hat{\varepsilon}^2/\hat{\varepsilon}_0^2]$ for the case of resubstitution and leave-one-out error estimators, which suffices to compute the bias, variance, and RMS of the corresponding CoD estimator, as discussed in the previous section. These expressions are functions only of sample size, number of bins (complexity), and the probability model. We will assume throughout, for definiteness, that the sample size n is even. The case where n is odd is in fact slightly simpler and can be readily obtained in analogous fashion to the derivations presented below.

5.1. Resubstitution

5.1. Resubstitution. The first moment of $\hat{\varepsilon}_r/\hat{\varepsilon}_0$ is given by

$$\begin{aligned}E\left[\frac{\hat{\varepsilon}_r}{\hat{\varepsilon}_0}\right] &= E\left[E\left[\frac{\hat{\varepsilon}_r}{\hat{\varepsilon}_0} \mid \hat{\varepsilon}_0\right]\right] \\ &= \sum_{m=1}^{n/2} E\left[\frac{\hat{\varepsilon}_r}{m/n} \mid M = m\right] P(M = m),\end{aligned} \quad (27)$$

where $M = n\hat{\varepsilon}_0$. Since $\hat{\varepsilon}_0 = (1/n)\min(N_0, N_1)$, we have $M = \min(N_0, n - N_0)$. It follows that the event $[M = m]$ is equal to the union of the disjoint events $[N_0 = m]$ and $[N_0 = n - m]$, for $m = 1, \ldots, n/2 - 1$ whereas $[M = n/2] = [N_0 = n/2]$. By using Proposition 1 in the appendix, we can write both cases in a single expression as follows:

$$\begin{aligned}&E\left[\frac{\hat{\varepsilon}_r}{m/n} \mid M = m\right] \\ &\quad = \frac{P(N_0 = m)}{P(N_0 = m) + P(N_0 = n - m)} \\ &\qquad \times E\left[\frac{\hat{\varepsilon}_r}{m/n} \mid N_0 = m\right] I_{1 \le m < n/2} \\ &\qquad + \frac{P(N_0 = n - m)}{P(N_0 = m) + P(N_0 = n - m)} \\ &\qquad \times E\left[\frac{\hat{\varepsilon}_r}{m/n} \mid N_0 = n - m\right] I_{1 \le m \le n/2}, \\ &\qquad\qquad m = 1, \ldots, n/2.\end{aligned} \quad (28)$$

By using (28) in (27) and considering that $P(M = m) = P(N_0 = m) + P(N_0 = n - m)$, we obtain

$$\begin{aligned}E\left[\frac{\hat{\varepsilon}_r}{\hat{\varepsilon}_0}\right] = \sum_{m=1}^{n/2} \Bigg\{ &E\left[\frac{\hat{\varepsilon}_r}{m/n} \mid N_0 = m\right] \\ &\times P(N_0 = m) I_{1 \le m < n/2} \\ &+ E\left[\frac{\hat{\varepsilon}_r}{m/n} \mid N_0 = n - m\right] \\ &\times P(N_0 = n - m) I_{1 \le m \le n/2} \Bigg\},\end{aligned} \quad (29)$$

where

$$\begin{aligned}&E\left[\frac{\hat{\varepsilon}_r}{m/n} \mid N_0 = t\right] \\ &\quad = \frac{1}{m} \sum_{i=1}^{b} \Bigg\{ \sum_{l > k} k P(U_i = k, V_i = l \mid N_0 = t) \\ &\qquad + \sum_{k \ge l} l P(U_i = k, V_i = l \mid N_0 = t) \Bigg\},\end{aligned} \quad (30)$$

with

$$\begin{aligned} P(U_i = k, V_i = l \mid N_0 = t) &\\ &= P(U_i = k \mid N_0 = t) P(V_i = l \mid N_1 = n - t) \\ &= \binom{t}{k} p_i^k (1 - p_i)^{(t-k)} \\ &\quad \times \binom{n-t}{l} q_i^l (1 - q_i)^{(n-t-l)}, \end{aligned} \tag{31}$$

for $t = m,\ n - m$.

The second moment of $\widehat{\varepsilon}_r/\widehat{\varepsilon}_0$ is given by

$$E\left[\frac{\widehat{\varepsilon}_r^2}{\widehat{\varepsilon}_0^2}\right] = \sum_{m=1}^{n/2} E\left[\frac{\widehat{\varepsilon}_r^2}{m^2/n^2} \mid M = m\right] P(M = m), \tag{32}$$

where $M = n\widehat{\varepsilon}_0$, as before. By using Proposition 1 in the appendix, and the same reasoning applied previously in the case of the first moment, we can write

$$\begin{aligned} E\left[\frac{\widehat{\varepsilon}_r^2}{m^2/n^2} \mid M = m\right] &\\ &= \frac{P(N_0 = m)}{P(N_0 = m) + P(N_0 = n - m)} \\ &\quad \times E\left[\frac{\widehat{\varepsilon}_r^2}{m^2/n^2} \mid N_0 = m\right] I_{1 \le m < n/2} \\ &\quad + \frac{P(N_0 = n - m)}{P(N_0 = m) + P(N_0 = n - m)} \\ &\quad \times E\left[\frac{\widehat{\varepsilon}_r^2}{m^2/n^2} \mid N_0 = n - m\right] I_{1 \le m \le n/2}, \\ &\qquad m = 1, \ldots, n/2. \end{aligned} \tag{33}$$

Combining (33) and (32) leads to

$$\begin{aligned} E\left[\frac{\widehat{\varepsilon}_r^2}{\widehat{\varepsilon}_0^2}\right] = \sum_{m=1}^{n/2} \Bigg\{ & E\left[\frac{\widehat{\varepsilon}_r^2}{m^2/n^2} \mid N_0 = m\right] \\ & \times P(N_0 = m) I_{1 \le m < n/2} \\ & + E\left[\frac{\widehat{\varepsilon}_r^2}{m^2/n^2} \mid N_0 = n - m\right], \\ & \times P(N_0 = n - m) I_{1 \le m \le n/2} \Bigg\}, \end{aligned} \tag{34}$$

where

$$\begin{aligned} E\left[\frac{\widehat{\varepsilon}_r^2}{m^2/n^2} \mid N_0 = t\right] &\\ = \frac{1}{m^2} \sum_{i=1}^{b} \Bigg\{ & \sum_{l > k} k^2 P(U_i = k, V_i = l \mid N_0 = t) \\ & + \sum_{k \ge l} l^2 P(U_i = k, V_i = l \mid N_0 = t) \Bigg\} + \frac{1}{m^2} \\ \times \sum_{\substack{i,j=1 \\ i \ne j}}^{b} \Bigg\{ & \sum_{l > k} \sum_{s > r} kr P\left(U_i = k, V_i = l, U_j = r, V_j = s \mid N_0 = t\right) \\ & + \sum_{l > k} \sum_{r \ge s} ks P\left(U_i = k, V_i = l, U_j = r, V_j = s \mid N_0 = t\right) \\ & + \sum_{k \ge l} \sum_{s > r} lr P\left(U_i = k, V_i = l, U_j = r, V_j = s \mid N_0 = t\right) \\ & + \sum_{k \ge l} \sum_{r \ge s} ls P\left(U_i = k, V_i = l, U_j = r, V_j = s \mid N_0 = t\right) \Bigg\}, \end{aligned} \tag{35}$$

with $P(U_i = k, V_i = l \mid N_0 = t)$ as in (31) and

$$\begin{aligned} P\left(U_i = k, V_i = l, U_j = r, V_j = s \mid N_0 = t\right) &\\ &= P\left(U_i = k, U_j = r \mid N_0 = t\right) \\ &\quad \times P\left(V_i = l, V_j = s \mid N_1 = n - t\right) \\ &= \binom{t}{k, r, t - k - r} p_i^k p_j^r \left(1 - p_i - p_j\right)^{t-k-r} \\ &\quad \times \binom{n-t}{l, s, n - t - l - s} q_i^l q_j^s \left(1 - q_i - q_j\right)^{n-t-l-s}, \end{aligned} \tag{36}$$

for $t = m,\ n - m$.

5.2. Leave-One-Out. To obtain the first moment of $\widehat{\varepsilon}_r/\widehat{\varepsilon}_0$, one can proceed exactly as in the resubstitution case to get

$$\begin{aligned} E\left[\frac{\widehat{\varepsilon}_l}{\widehat{\varepsilon}_0}\right] &\\ = \sum_{m=1}^{n/2} \Bigg\{ & E\left[\frac{\widehat{\varepsilon}_l}{m/n} \mid N_0 = m\right] P(N_0 = m) I_{1 \le m < n/2} \\ & + E\left[\frac{\widehat{\varepsilon}_l}{m/n} \mid N_0 = n - m\right] P(N_0 = n - m) I_{1 \le m \le n/2} \Bigg\}, \end{aligned} \tag{37}$$

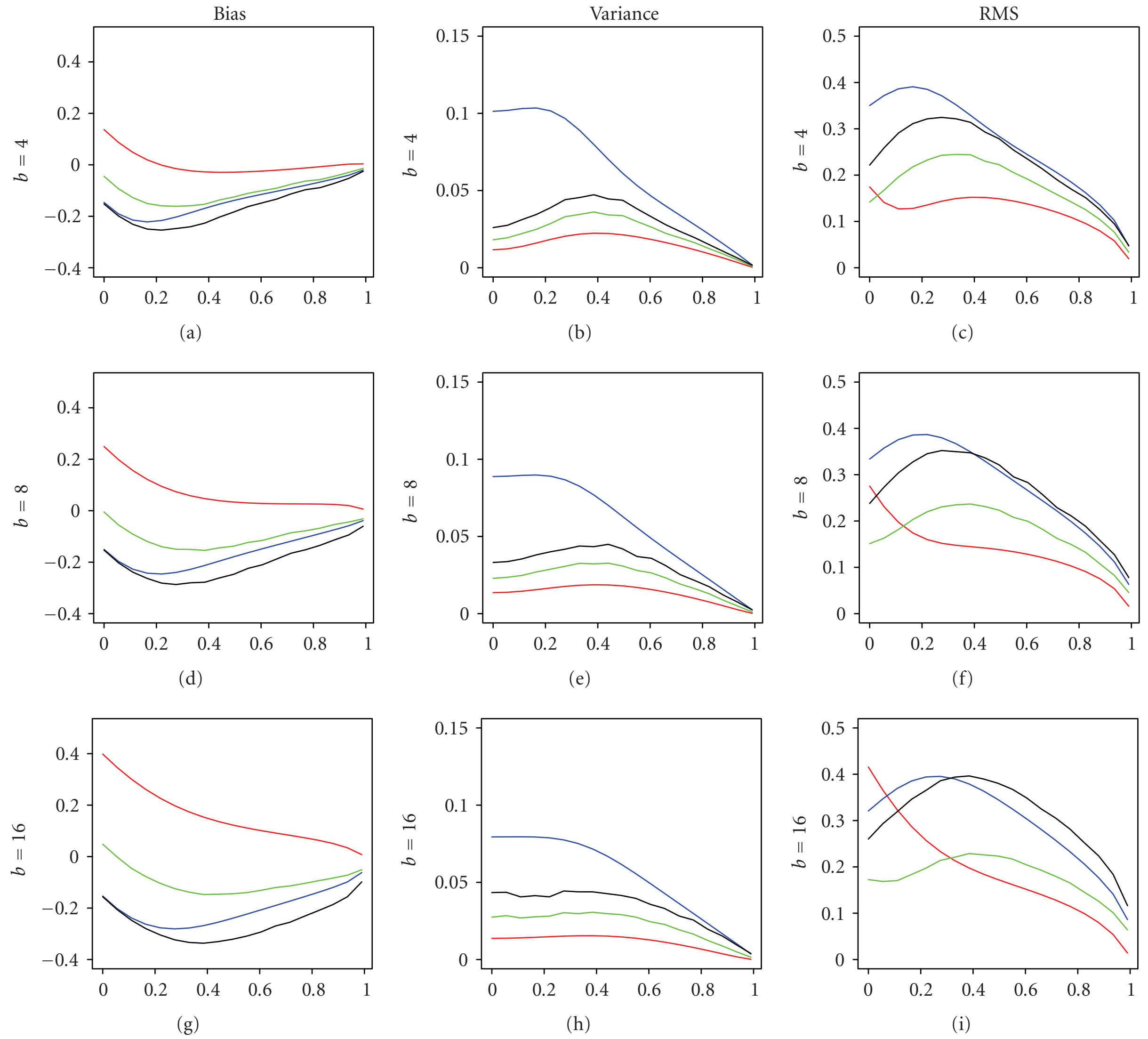


Figure 1: Bias, variance, and RMS for several CoD estimators versus actual CoD under a Zipf model with $c_0 = 1/2$, for $n = 40$ and varying number of bins. Plot key: resubstitution (red), leave-one-out (blue), 0.632 bootstrap (green), 10-repeated 2-fold cross-validation (black). The curves for resubstitution and leave-one-out are exact; the curves for the other CoD estimators are approximations based on Monte Carlo sampling.

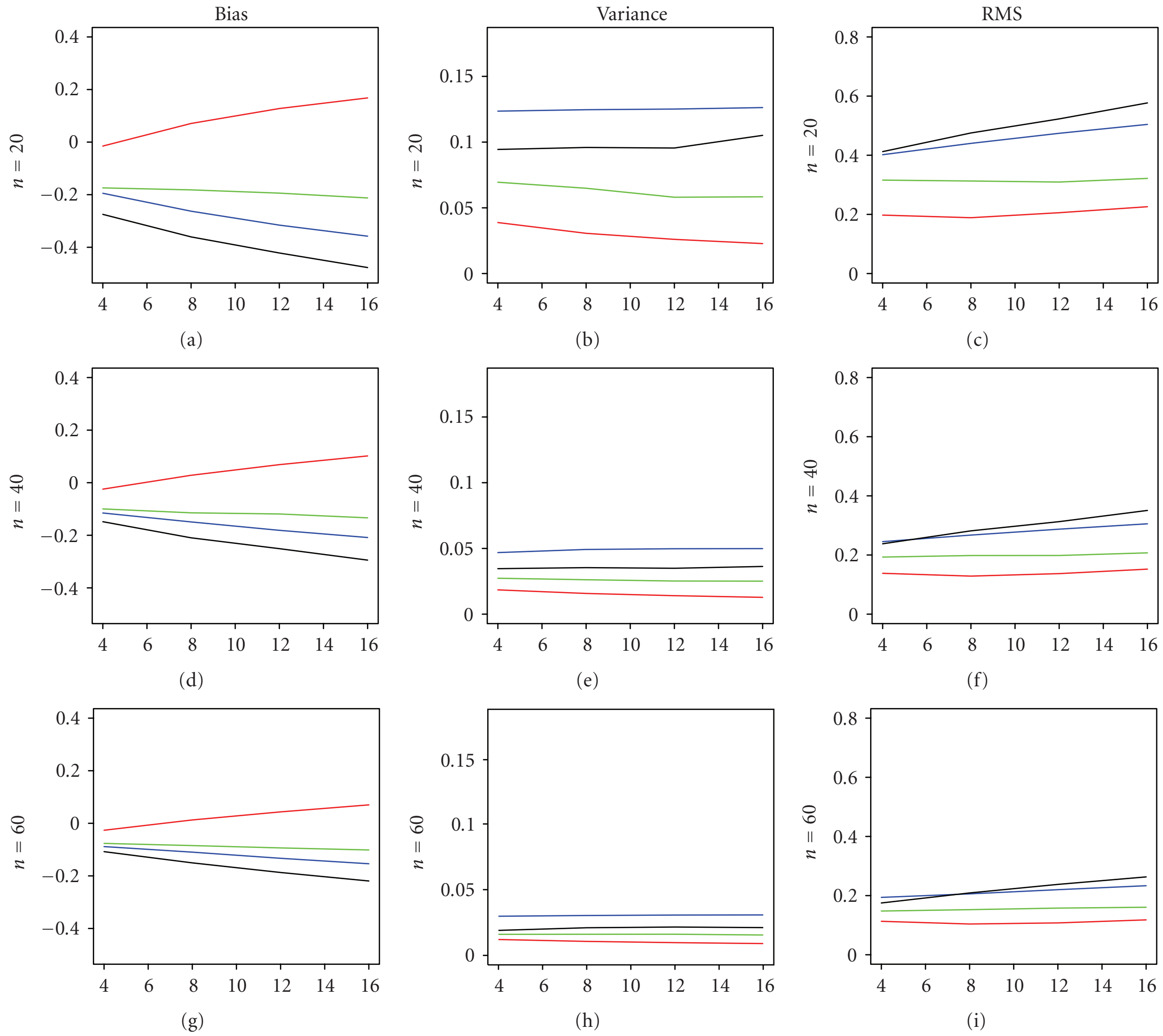
where now

$$E\left[\frac{\hat{\varepsilon}_l}{m/n} \mid N_0 = t\right] = \frac{1}{m}\sum_{i=1}^{b}\left\{\sum_{l\geq k} kP(U_i = k, V_i = l \mid N_0 = t) + \sum_{k\geq l-1} lP(U_i = k, V_i = l \mid N_0 = t)\right\}, \tag{38}$$

with $P(U_i = k, V_i = l \mid N_0 = t)$ as in (31), for $t = m,\ n - m$.

To obtain the second moment of $\hat{\varepsilon}_r/\hat{\varepsilon}_0$, one can again proceed as in the resubstitution case to get

$$E\left[\frac{\hat{\varepsilon}_l^2}{\hat{\varepsilon}_0^2}\right] = \sum_{m=1}^{n/2}\left\{E\left[\frac{\hat{\varepsilon}_l^2}{m^2/n^2} \mid N_0 = m\right]P(N_0 = m)I_{1\leq m<n/2} + E\left[\frac{\hat{\varepsilon}_l^2}{m^2/n^2} \mid N_0 = n-m\right]P(N_0 = n-m)I_{1\leq m\leq n/2}\right\}, \tag{39}$$



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 2: Bias, variance, and RMS for several CoD estimators versus number of bins ($b = 4, 8, 12$, and 16) under a Zipf model with $c_0 = 1/2$, for actual CoD $= 0.6$ and varying sample size. Plot key: resubstitution (red), leave-one-out (blue), 0.632 bootstrap (green), 10-repeated 2-fold cross-validation (black). The curves for resubstitution and leave-one-out are exact; the curves for the other CoD estimators are approximations based on Monte Carlo sampling.

where now

$$
\begin{aligned}
&E\left[\frac{\hat{\varepsilon}_l^2}{m^2/n^2} \mid M = t\right] \\
&\quad = \frac{1}{m^2}\sum_{i=1}^{b}\left\{\sum_{l\geq k} k^2 P(U_i = k, V_i = l \mid N_0 = t)\right. \\
&\qquad\quad + \sum_{k\geq l-1} l^2 P(U_i = k, V_i = l \mid N_0 = t) \\
&\qquad\quad \left. + \sum_{l-1\leq k\leq l} 2klP(U_i = k, V_i = l \mid N_0 = t)\right\} + \frac{1}{m^2} \\
&\qquad \times \sum_{\substack{i,j=1\\ i\neq j}}^{b}\left\{\sum_{l\geq k}\sum_{s\geq r} krP\Big(U_i = k, V_i = l, U_j = r, V_j = s \mid N_0 = t\Big)\right. \\
&\qquad\quad + \sum_{l\geq k}\sum_{r\geq s-1} ksP\Big(U_i{=}k, V_i{=}l, U_j{=}r, V_j{=}s|N_0{=}t\Big) \\
&\qquad\quad + \sum_{k\geq l-1}\sum_{s\geq r} lrP\Big(U_i{=}k, V_i{=}l, U_j{=}r, V_j{=}s|\, N_0{=}t\Big) \\
&\qquad\quad \left. + \sum_{k\geq l-1}\sum_{r\geq s-1} lsP\Big(U_i{=}k, V_i{=}l, U_j{=}r, V_j{=}s\, |N_0{=}t\Big)\right\},
\end{aligned}
\tag{40}
$$

with $P(U_i = k, V_i = l \mid N_0 = t)$ as in (31) and $P(U_i = k, V_i = l, U_j = r, V_j = s \mid N_0 = t)$ as in (36), for $t = m,\ n - m$.

6. Numerical Experiments

Assuming a parametric probability model in this section, we plot the exact performance metrics of the resubstitution

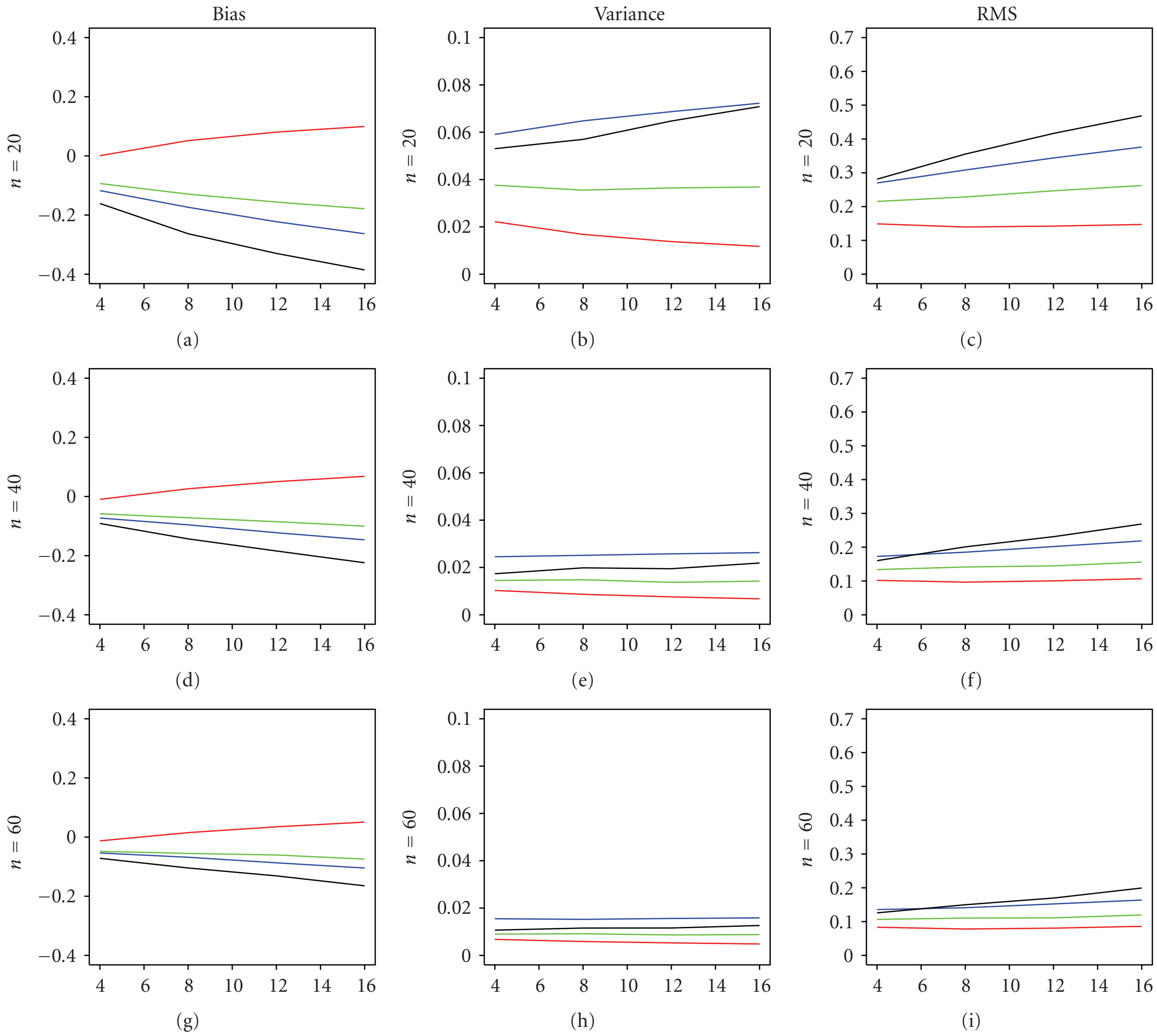


FIGURE 3: Bias, variance, and RMS for several CoD estimators versus number of bins ($b = 4, 8, 12$, and 16) under a Zipf model with $c_0 = 1/2$, for actual CoD $= 0.8$ and varying sample size. Plot key: resubstitution (red), leave-one-out (blue), 0.632 bootstrap (green), 10-repeated 2-fold cross-validation (black). The curves for resubstitution and leave-one-out are exact; the curves for the other CoD estimators are approximations based on Monte Carlo sampling.

and leave-one-out CoD estimators, by using the analytical expressions obtained in Sections 4 and 5, under varying actual CoD, sample size, and predictor complexity (number of bins). We also compare these exact performance metrics with the approximate performance metrics for cross-validation and bootstrap CoD estimators computed via Monte Carlo sampling. The Monte Carlo computation was carried out by drawing $M = 5000$ simulated training data sets of the required sample size from the probability model in each case, and employing sample means and sample variances to approximate the performance metrics in Section 4.

The probability model used here is a parametric Zipf model [16]. The class-conditional probabilities under the parametric Zipf model are given by

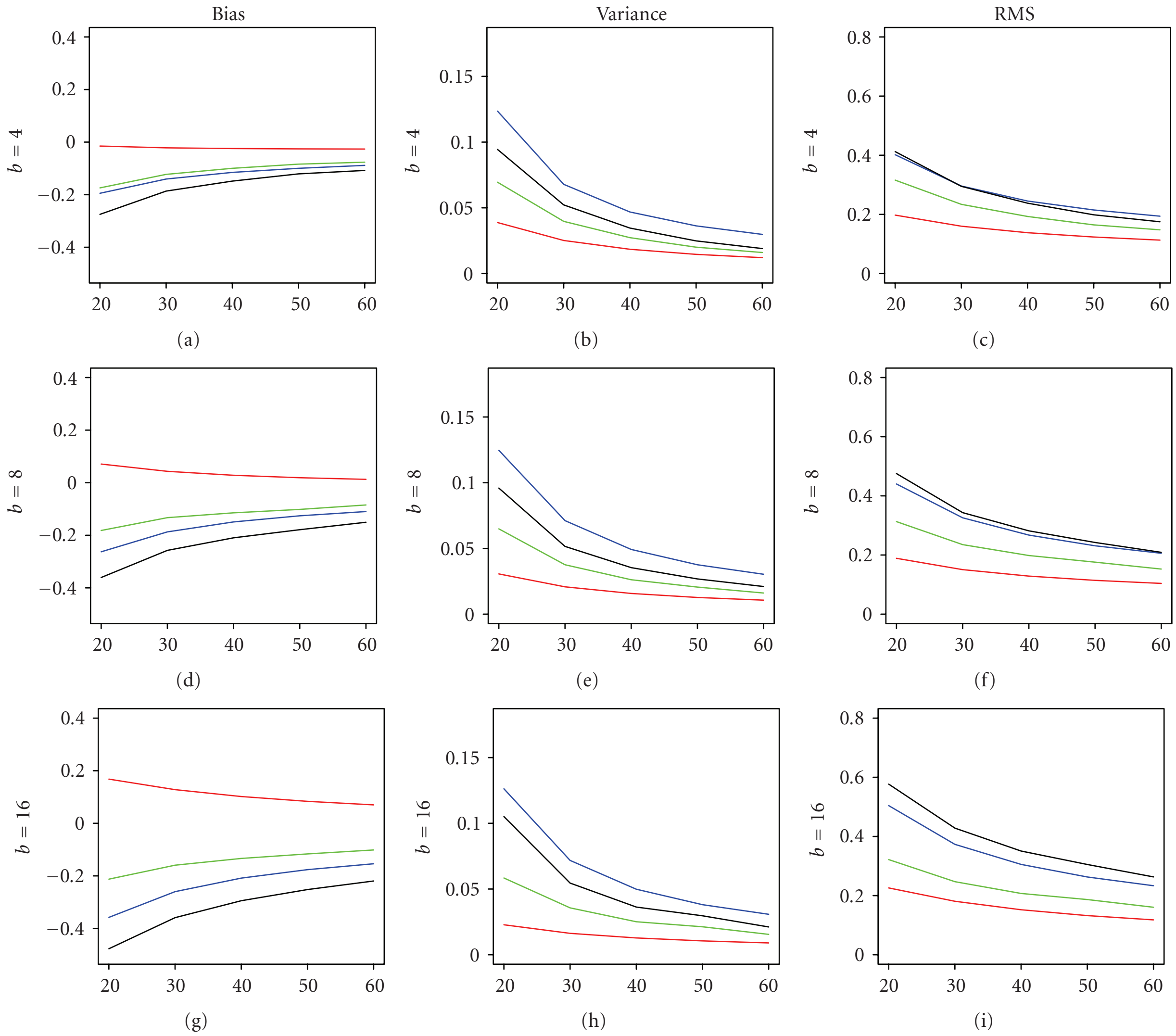
$$p_i = \frac{K}{i^\alpha}, \qquad q_i = p_{b-i+1}, \tag{41}$$

for $i = 1, \ldots, b$, and $\alpha > 0$. The normalizing constant K is given by

$$K = \left[\sum_{i=1}^{b} \frac{1}{i^\alpha}\right]^{-1}. \tag{42}$$

For simplicity, we assume that $c_0 = c_1 = 1/2$. It can be seen easily from (6) that the CoD increases monotonically with α, so that large α leads to tight regulation, that is, easy prediction, and vice versa. There are two extreme cases. When $\alpha = 0$, there is maximal confusion between the classes, and CoD $= 0$. When $\alpha \to \infty$, there is maximal discrimination between the classes, and CoD $= 1$. Thus, varying the parameter α can traverse the probability model space continuously from easy to difficult models.

We consider here the prediction setting where each predictor variable is binary. If we employ 2, 3, and 4 predictor



(a) (b) (c)

(d) (e) (f)

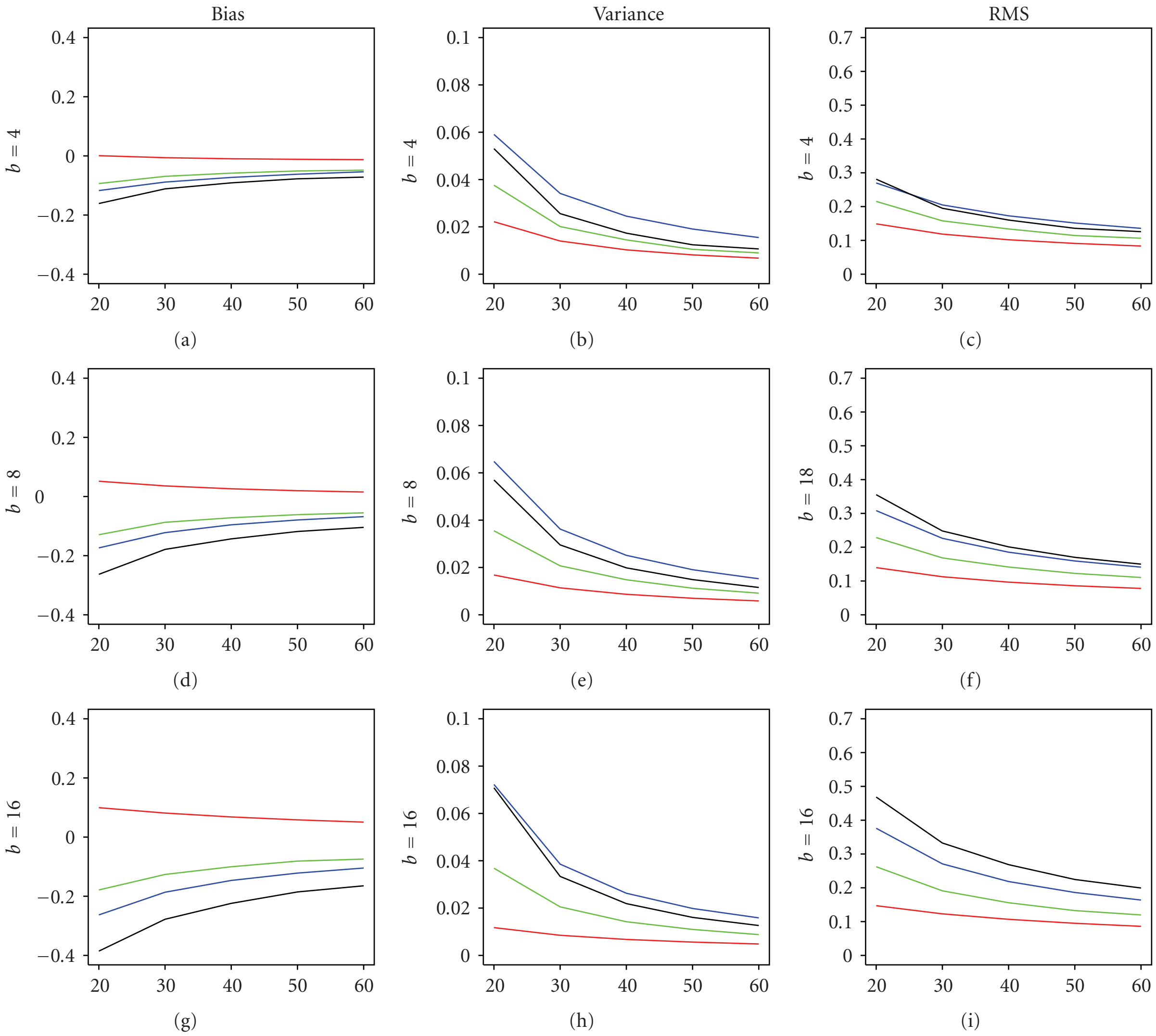
(g) (h) (i)

Figure 4: Bias, variance, and RMS for several CoD estimators versus sample size ($n = 20, 30, 40, 50$, and 60) under a Zipf model with $c_0 = 1/2$, for actual CoD = 0.6 and varying number of bins. Plot key: resubstitution (red), leave-one-out (blue), 0.632 bootstrap (green), 10-repeated 2-fold cross-validation (black). The curves for resubstitution and leave-one-out are exact; the curves for the other CoD estimators are approximations based on Monte Carlo sampling.

variables then this would correspond to bin sizes $b = 4, 8, 16$, respectively. In functional genomics applications, these cases correspond to the gene prediction problem by using 2, 3, and 4 genes, where the activity of each gene is represented by binary gene expressions, for example, the on-and-off switch effect of a promoter.

Figure 1 displays bias, variance, and RMS of the CoD estimators considered here, as a function of varying actual CoD (computed by suitable tuning the parameter α). We recall that, in the figure, tight regulation, that is, easy prediction, is located on the right of these plots whereas loose regulation, that is, difficult prediction, is located on the left.

Figure 1 makes apparent several facts. The resubstitution CoD is often optimistically biased, except at moderate to large CoD with $b = 4$ (two binary predictors) whereas the other estimators are generally pessimistically biased. As the number of predictors increase, the bias (in magnitude) of the resubstitution CoD increases accordingly; however, its variance remains quite low in each case. The leave-one-out CoD is highly variable, in addition to being pessimistically biased. By observing the RMS, we conclude that the resubstitution CoD estimator is the best-performing estimator, except at small values of the actual CoD, beating all the other estimators, including the bootstrap. The leave-one-out CoD estimator is the worst-performing estimator for cases with small number of predictors ($b = 4$) whereas the cross-validation CoD estimator becomes the worst-performing estimator for large number of predictors and moderate actual CoD. As the number of predictors increases, the actual CoD cut off decreases accordingly at which the leave-one-out CoD estimator starts to outperform the cross-validation CoD estimator. It is also interesting to note that, for $b = 4$, only



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 5: Bias, variance, and RMS for several CoD estimators versus sample size ($n = 20, 30, 40, 50$, and 60) under a Zipf model with $c_0 = 1/2$, for actual CoD = 0.8 and varying number of bins. Plot key: resubstitution (red), leave-one-out (blue), 0.632 bootstrap (green), 10-repeated 2-fold cross-validation (black). The curves for resubstitution and leave-one-out are exact; the curves for the other CoD estimators are approximations based on Monte Carlo sampling.

the bootstrap beats resubstitution, and for very small actual CoD. For $b = 8$, both bootstrap and cross-validation perform better than the resubstitution, for small actual CoD. For $b = 16$, all the other CoD estimators outperform resubstitution for small actual CoD. As the number of predictors increases, the cut-off at which the resubstitution CoD estimator beats all other estimators increases.

In order to assess the performance of the resubstitution CoD estimator and the remaining CoD estimators with respect to the classifier complexity (number of predictors), we display the performance metrics as a function of varying number of bins in Figures 2 and 3, for sample size $n = 20, 40$, and 60, and moderate CoD = 0.6 and large CoD = 0.80. The bias column shows that, for CoD = 0.60, the resubstitution CoD is actually slightly pessimistically biased for $b = 4$ (a perhaps surprising fact, given the optimistic bias of resubstitution in discrete classification), but quickly becomes optimistically biased for larger bin sizes. In the RMS column, we can see that the resubstitution CoD always beats all other estimators, especially in the case of CoD = 0.80 (tight regulation), which is the more surprising when we consider that the other estimators are much more computation-intensive. It is interesting to see that the leave-one-out CoD estimator beats the more complex cross-validation CoD estimator for small number of bins and large sample size. The resubstitution CoD is the least biased and least variable among all CoD estimators, across the whole range of classifier complexity and sample size considered here, and thus it also displays the best RMS overall.

In Figures 4 and 5, we examine how these performance metrics behave with varying sample sizes for $b = 4, 8, 16$, and moderate CoD = 0.6 and large CoD = 0.80. As expected, bias (in magnitude), variance and RMS all decrease as sample size increases. We can see that the resubstitution CoD is the least biased and least variable among all estimators, and thus also displays the best RMS. The cross-validation CoD estimator is the most biased, and the leave-one-out CoD estimator is the most variable, among all CoD estimators. The bootstrap CoD estimator is less variable than the cross-validation CoD estimator.

7. Conclusion

This paper presented a comprehensive study of CoD estimators. We derived for the first time exact analytical expressions of performance metrics of the resubstitution and leave-one-out CoD estimators. Using a parametric Zipf model, we have compared the exact performance metrics of resubstitution and leave-one-out between each other and against approximate performance metrics of cross-validation and bootstrap CoD estimators. Our results lead to a perhaps surprising conclusion: under the Zipf model under consideration, the resubstitution CoD estimator is the best-performing estimator among all, for moderate to large actual CoD and not too large number of predictors. However, for small actual CoD values and high classifier complexity, the other three CoD estimators can outperform resubstitution. This indicates that provided one has evidence of moderate to tight regulation between the genes, and the number of predictors is not too large, one should use the CoD estimator based on resubstitution.

This work is intended to serve as foundation for a detailed study of the application of CoD estimation in Genomics and related fields. An obvious application is the inference of genomic regulatory networks from sample microarray data. In addition to that, there are several issues related to nonlinear prediction in the discrete domain, which can benefit from the work presented here.

Appendix

Proposition 1. *For a discrete random variable X and disjoint events A and B, one has*

$$\begin{aligned} E[X \mid A \cup B] = {} & \frac{P(A)}{P(A)+P(B)}\, E[X \mid A] \\ & + \frac{P(B)}{P(A)+P(B)}\, E[X \mid B]. \end{aligned} \tag{A.1}$$

Proof.

$$\begin{aligned} & E[X \mid A \cup B] \\ & \quad = \sum_x x\, P(X = x \mid A \cup B) \\ & \quad = \sum_x x\, \frac{P(A \cup B \mid X = x) P(X = x)}{P(A \cup B)} \\ & \quad = \sum_x x \frac{[P(A \mid X=x) + P(B \mid X=x)] P(X=x)}{P(A)+P(B)} \\ & \quad = \sum_x x \frac{P(X=x \mid A)P(A) + P(X=x \mid B)P(B)}{P(A)+P(B)} \\ & \quad = \frac{P(A)}{P(A)+P(B)} \sum_x x\, P(X = x \mid A) \\ & \qquad + \frac{P(B)}{P(A)+P(B)} \sum_x x\, P(X = x \mid B) \\ & \quad = \frac{P(A)}{P(A)+P(B)} E[X \mid A] \\ & \qquad + \frac{P(B)}{P(A)+P(B)} E[X \mid B]. \end{aligned} \tag{A.2}$$

□

Acknowledgment

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

References

[1] E. R. Dougherty, S. Kim, and Y. Chen, "Coefficient of determination in nonlinear signal processing," *Signal Processing*, vol. 80, no. 10, pp. 2219–2235, 2000.

[2] D. C. Martins Jr., U. M. Braga-Neto, R. F. Hashimoto, M. L. Bittner, and E. R. Dougherty, "Intrinsically multivariate predictive genes," *IEEE Journal on Selected Topics in Signal Processing*, vol. 2, no. 3, pp. 424–439, 2008.

[3] S. Kim, E. R. Dougherty, M. L. Bittner et al., "A general nonlinear framework for the analysis of gene interaction via multivariate expression arrays," *Journal of Biomedical Optics*, vol. 5, no. 4, pp. 411–424, 2000.

[4] S. Kim, E. R. Dougherty, Y. Chen et al., "Multivariate measurement of gene expression relationships," *Genomics*, vol. 67, no. 2, pp. 201–209, 2000.

[5] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.

[6] X. Zhou, X. Wang, and E. R. Dougherty, "Binarization of microarray data based on a mixture model," *Molecular Cancer Therapeutics*, vol. 2, no. 7, pp. 679–684, 2003.

[7] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, Springer, New York, NY, USA, 1996.

[8] U. M. Braga-Neto, "Classification and error estimation for discrete data," *Current Genomics*, vol. 10, no. 7, pp. 446–462, 2009.

[9] U. Braga-Neto and E. Dougherty, "Exact performance of error estimators for discrete classifiers," *Pattern Recognition*, vol. 38, no. 11, pp. 1799–1814, 2005.

[10] Q. Xu, J. Hua, U. 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*, vol. 5, no. 6, pp. 579–589, 2006.

[11] C. A. B. Smith, “Some examples of discrimination,” *Annals of Eugenics*, vol. 18, pp. 272–282, 1947.

[12] P. A. Lachenbruch and M. R. Mickey, “Estimation of error rates in discriminant analysis,” *Technometrics*, vol. 10, pp. 1–11, 1968.

[13] M. Stone, “Cross-validatory choice and assessment of statistical predictions,” *Journal of the Royal Statistical Society. Series B*, vol. 36, pp. 111–147, 1974.

[14] B. Efron, “Bootstrap methods: another look at the jackknife,” *Annals of Statistics*, vol. 7, pp. 1–26, 1969.

[15] B. Efron, “Estimating the error rate of a prediction rule: improvement on cross-validation,” *Journal of the American Statistical Association*, vol. 78, no. 382, pp. 316–331, 1983.

[16] G. K. Zipf, *Psycho-Biology of Languages*, Houghton-Mifflin, Boston, Mass, USA, 1935.