Journal of Machine Learning Research 23 (2022) 1-30 Submitted 7/21; Revised 5/22; Published 9/22

Generalized Resubstitution for Classification Error Estimation

Parisa Ghane PGHANE@TAMU.EDU
Ulisses Braga-Neto ULISSES@TAMU.EDU
Department of Electrical and Computer Engineering
Texas A&M University
College Station, TX 77843 USA

Editor: Gabor Lugosi

Abstract

We propose the family of generalized resubstitution classifier error estimators based on arbitrary empirical probability measures. These error estimators are computationally efficient and do not require retraining of classifiers. The plain resubstitution error estimator corresponds to choosing the standard empirical probability measure. Other choices of empirical probability measure lead to bolstered, posterior-probability, Gaussian-process, and Bayesian error estimators; in addition, we propose here bolstered posterior-probability error estimators, as a new family of generalized resubstitution estimators. In the two-class case, we show that a generalized resubstitution estimator is consistent and asymptotically unbiased, regardless of the distribution of the features and label, if the corresponding empirical probability measure converges uniformly to the standard empirical probability measure and the classification rule has finite VC dimension. A generalized resubstitution estimator typically has hyperparameters that can be tuned to control its bias and variance, which adds flexibility. We conducted extensive numerical experiments with various classification rules trained on synthetic data, which indicate that the new family of error estimators proposed here produces the best results overall, except in the case of very complex, overfitting classifiers, in which semi-bolstered resubstitution should be used instead. In addition, results of an image classification experiment using the LeNet-5 convolutional neural network and the MNIST data set show that naive-Bayes bolstered resubstitution with a simple data-driven calibration procedure produces excellent results, demonstrating the potential of this class of error estimators in deep learning for computer vision.

Keywords: Classification; Error Estimation; Resubstitution; Empirical Probability Measure; Convolutional Neural Networks.

1. Introduction

Given enough training data, good classification algorithms produce classifiers with small error rate on future data, which is also known in machine learning as the *generalization error*. But a classifier is useful only if its generalization error can be stated with confidence. Hence, at a fundamental level, one can only speak of the goodness of a classification algorithm together with an error estimation procedure that produces an accurate assessment of the true generalization error of the resulting classifier. Error estimation for classification has a long history and many different error estimation procedures have been proposed (Toussaint, 1974; Hand, 1986; McLachlan, 1987; Schiavo and Hand, 2000; Braga-Neto and Dougherty, 2015b). The subject has recently become a topic of concern in the deep learning community (Jiang et al., 2019). Error estimators based on resampling, such as cross-validation (Lachenbruch and Mickey, 1968; Cover, 1969; Toussaint and Donaldson, 1970; Stone, 1974), and bootstrap (Efron, 1979, 1983; Efron and Tibshirani, 1997), have long been popular choices of error estimation procedures.

However, in contemporary classification applications, particularly in the case of deep learning, training can be time and resource intensive (Simonyan and Zisserman, 2014). As a result, error estimators based on resampling are no longer a viable choice, since they require training tens or hundreds of classifiers on resampled versions of the training data. It has become instead the norm to use the test-set error, i.e., the error rate on data not used in training, to benchmark classifiers (Russakovsky et al., 2015). The test-set error estimator is an unbiased, consistent estimator of the generalization error regardless of the sample size or distribution of the problem (Braga-Neto and Dougherty, 2015b). However, this is only true if the test data is truly independent of training, and is not reused in any way (Yousefi et al., 2011). It has been recognized recently that this has not been always the case in image classification using popular benchmarks, where the same public test sets are heavily re-used to measure classification improvement, creating a situation known as "training to the test data" (Recht et al., 2019). Strictly speaking, true independent test sets are one-way: they can only be used once. In addition, if training and testing sample sizes are small, the test-set error estimator can display large variance, and become unreliable. All of this means that accurate test-set error estimation requires cheap access to plentiful labeled data.

The alternative to resampling and test-set error estimation is testing on the training data. The error rate on the training data is known as the *resubstitution* error estimator (Smith, 1947). This does not require retraining the classifier and is as fast as using a test-set error estimator, but does not assume any separate independent test data. The resubstitution estimator is however usually optimistically biased, the more so the more the classification algorithm overfits the training data. Optimistic bias implies that the difference between resubstitution estimate and the true error, which has been called the "generalization gap" (Keskar et al., 2016), is negative with a high probability. It is key therefore to investigate mechanisms to reduce the bias.

In this paper, we propose and study the family of generalized resubstitution error estimators, which are defined in terms of arbitrary empirical probability measures. In addition to plain resubstitution, this family includes well-known error estimators, such as posterior-probability (Lugosi and Pawlak, 1994), Gaussian-process (Hefny and Atiya, 2010), bolstered resubstitution (Braga-Neto and Dougherty, 2004c), and Bayesian (Dalton and Dougherty, 2011a,b) error estimators. The empirical probability measures used in generalized

resubstitution generally contain hyperparameters that can be tuned to reduce the bias and variance of the estimator with respect to plain resubstitution. We summarize our contributions as follows:

- We propose a completely general and unifying framework to study resubstitution-like classification error estimators in terms of arbitrary empirical probability measures.
- We provide a criterion for consistency and asymptotic unbiasedness and convergence of generalized resubstitution estimators in the case of two-class, finite-VC dimension classification rules (Thm. 1). This criterion is used to establish the consistency and asymptotic unbiasedness of several generalized resubstitution estimators, including bolstered resubstitution estimators, for which no such result existed previously (Thm. 2).
- We propose a new family of error estimators, called bolstered posterior-probability error estimators, which combine bolstered and posterior-probability estimators, and produced superior results in our experiments with synthetic data.
- We provide multi-class formulations of existing error estimators, namely, the bolstered and posterior-probability resubstitution error estimators.
- We show empirically that naive-Bayes resubstitution with a simple data-drive calibration procedure produces excellent results in image classification by convolutional neural networks on the MNIST data set, which indicates the potential of this class of error estimators in the computer vision area.

The paper is organized as follows. In Section 2, the necessary concepts about classification error estimation are reviewed. Section 3 introduces generalized resubstitution error estimators and provides a criterion for their consistency and asymptotic unbiasedness. Section 4 discusses the important special case of generalized resubstitution estimators based on smoothing the standard empirical probability measure; we show that the existing bolstering and posterior probability error estimators are a member of this family, and propose a new family of error estimators that combines them. Section 5 covers the additional examples of Bayesian and Gaussian-Process generalized resubstitution estimators, and discusses briefly the extension of the proposed framework to cross-validation and test-set error estimators. Section 6 contains the results of an extensive numerical study based on synthetic and real imaging data to examine the accuracy of several representative generalized resubstitution error estimators using a variety of classification rules. Finally, Section 7 presents concluding remarks.

2. Background on Classification Error Estimation

The subject of classification error estimation has a long history and has produced a large body of literature; four main review papers summarize major advances in the field up to 2000 (Toussaint, 1974; Hand, 1986; McLachlan, 1987; Schiavo and Hand, 2000); recent advances in error estimation since 2000 include work on model selection (Bartlett et al., 2002), bolstered error estimation (Braga-Neto and Dougherty, 2004c; Sima et al., 2005b), feature selection (Sima et al., 2005a; Zhou and Mao, 2006; Xiao et al., 2007; Hanczar et al., 2007), confidence intervals (Kaariainen and Langford, 2005; Kaariainen, 2005; Xu et al., 2006), model-based second-order properties (Zollanvari et al., 2011, 2012), and Bayesian error estimators (Dalton

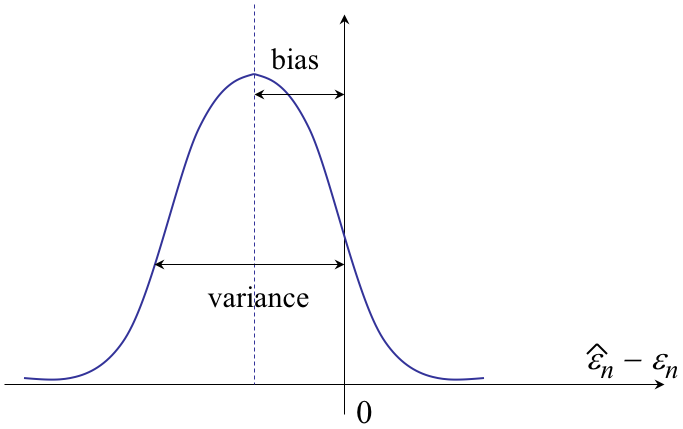


Figure 1: Deviation distribution showing bias and variance. The error estimator in this example is optimistically biased.

and Dougherty, 2011a,b). In this section, we provide a brief review of the basic concepts related to error estimation. A booklength treatment of the topic is provided in (Braga-Neto and Dougherty, 2015a); see also (McLachlan, 1992; Devroye et al., 1996).

Let the feature vector $\mathbf{X} \in R^d$ and the label $Y \in R$ be jointly distributed with corresponding probability measure ν, such that $\nu(R^d \times \{0, 1, \ldots, c-1\}) = 1$. An event is a Borel set $A \subseteq R^d \times \{0, 1, \ldots, c-1\}$. A *classifier* ψ is a Borel-measurable function from R^d to $\{0, 1, \ldots, c-1\}$. In practice, one collects i.i.d. *training data* $S_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$, where each pair $(\mathbf{X}_i, Y_i)$ is distributed as $(\mathbf{X}, Y)$, and designs a classifier $\psi_n = \Psi_n(S_n)$, by means of a classification rule Ψ_n. The quantity of interest is the classification error probability:

$$\varepsilon_n = \nu(\{(\mathbf{x}, y) : \psi_n(\mathbf{x}) \neq y\}), \tag{1}$$

where the probability measure ν, which is supported on $R^d \times \{0, 1, \ldots, c-1\}$ is the distribution of $(\mathbf{X}, Y)$.

If one knew the distribution of the features and label, then one could in principle compute the classification error ε_n by evaluating (1). In practice, such knowledge is rarely available, so one employs an *error estimation rule* Ξ_n in order to obtain a classification error estimator $\hat{\varepsilon}_n = \Xi_n(\Psi_n, S_n, \xi)$, where ξ denote *internal random factors* (if any) that represent randomness that is not introduced by the training data S_n; if there are no such internal random factors, the error estimator is said to be nonrandomized, in which case the error estimate is entirely determined by the training data, otherwise, it is said to be *randomized*. For example, the *resubstitution* error estimator is the proportion of errors committed on the training data, and is therefore nonrandomized, but the *cross-validation* error estimator randomly partitions the data into training and testing folds, trains classifiers on the training subsets, and evaluates then on the testing subset, which yields a randomized error estimator.

The performance of an error estimator can be assessed by the distribution of $\hat{\varepsilon}_n - \varepsilon_n$, called the *deviation distribution* (Braga-Neto and Dougherty, 2004b). For good performance, this distribution should be peaked (low-variance) and centered near zero (low-bias). See Figure 1 for an illustration.

The *bias* is defined as the first moment of the deviation distribution:

$$\mathrm{Bias}(\hat{\varepsilon}_n) \;=\; E[\hat{\varepsilon}_n - \varepsilon_n] \;=\; E[\hat{\varepsilon}_n] - E[\varepsilon_n]\,. \tag{2}$$

The error estimator $\hat{\varepsilon}_n$ is said to be *optimistically biased* if $\mathrm{Bias}(\hat{\varepsilon}_n) < 0$ and *pessimistically biased* if $\mathrm{Bias}(\hat{\varepsilon}_n) > 0$. It is *unbiased* if $\mathrm{Bias}(\hat{\varepsilon}_n) = 0$. The resubstitution error estimator is usually optimistically biased.

The *deviation variance* is the variance of the deviation distribution:

$$\mathrm{Var}_{\mathrm{dev}}(\hat{\varepsilon}_n) \;=\; \mathrm{Var}(\hat{\varepsilon}_n - \varepsilon_n) \;=\; \mathrm{Var}(\hat{\varepsilon}_n) + \mathrm{Var}(\varepsilon_n) - 2\mathrm{Cov}(\varepsilon_n, \hat{\varepsilon}_n)\,. \tag{3}$$

Unlike in classical statistics, where estimators for fixed parameters are sought, here the quantity being estimated, namely ε_n, is random and thus a "moving target." This is why it is appropriate to consider the variance of the difference, $\mathrm{Var}(\hat{\varepsilon}_n - \varepsilon_n)$. However, if the classification rule is not overfitting, then $\mathrm{Var}(\varepsilon_n) \approx 0$ — in fact, overfitting could be defined as present if $\mathrm{Var}(\varepsilon_n)$ is large, since in that case the classification rule is learning the changing data and not the fixed underlying feature-label distribution. It follows, from the Cauchy-Schwartz Inequality that $\mathrm{Cov}(\varepsilon_n, \hat{\varepsilon}_n) \le \sqrt{\mathrm{Var}(\varepsilon_n)\mathrm{Var}(\hat{\varepsilon}_n)} \approx 0$, and thus, from (3), $\mathrm{Var}(\hat{\varepsilon}_n - \varepsilon_n) \approx \mathrm{Var}(\hat{\varepsilon}_n)$. If an estimator is randomized, then it has additional *internal variance* $V_{\mathrm{int}} = \mathrm{Var}(\hat{\varepsilon}_n | S_n)$, which measures the variability due only to the internal random factors, while the full variance $\mathrm{Var}(\hat{\varepsilon}_n)$ measures the variability due to both the sample S_n and the internal random factors ξ. The following formula can be easily shown using the Conditional Variance Formula of probability theory:

$$\mathrm{Var}(\hat{\varepsilon}_n) \;=\; E[V_{\mathrm{int}}] + \mathrm{Var}(E[\hat{\varepsilon}_n | S_n])\,. \tag{4}$$

The first term on the right-hand side contains the contribution of the internal variance to the total variance. For nonrandomized $\hat{\varepsilon}_n$, $V_{\mathrm{int}} = 0$; for randomized $\hat{\varepsilon}_n$, $E[V_{\mathrm{int}}] > 0$.

The *root mean-square error* is the square root of the second moment of the deviation distribution:

$$\mathrm{RMS}(\hat{\varepsilon}_n) \;=\; \sqrt{E[(\hat{\varepsilon}_n - \varepsilon_n)^2]} \;=\; \sqrt{\mathrm{Bias}(\hat{\varepsilon}_n)^2 + \mathrm{Var}_{\mathrm{dev}}(\hat{\varepsilon}_n)} \tag{5}$$

The RMS is generally considered the most important error estimation performance metric. The other performance metrics appear within the computation of the RMS; indeed, all of the five basic moments — the expectations $E[\varepsilon_n]$ and $E[\hat{\varepsilon}_n]$, the variances $\mathrm{Var}(\varepsilon_n)$ and $\mathrm{Var}(\hat{\varepsilon}_n)$, and the covariance $\mathrm{Cov}(\varepsilon_n, \hat{\varepsilon}_n)$ — appear within the RMS.

Finally, an error estimator is said to be *consistent* if $\hat{\varepsilon}_n \to \varepsilon_n$ in probability as $n \to \infty$, and *strongly consistent* for almost sure (a.s.) convergence. By an application of Markov's Inequality, we have

$$P(|\hat{\varepsilon}_n - \varepsilon_n| \ge \tau) = P(|\hat{\varepsilon}_n - \varepsilon_n|^2 \ge \tau^2) \;\le\; \frac{E[|\hat{\varepsilon}_n - \varepsilon_n|^2]}{\tau^2} \;=\; \left(\frac{\mathrm{RMS}(\hat{\varepsilon}_n)}{\tau}\right)^2, \quad \text{for } \tau > 0\,. \tag{6}$$

Hence, if $\mathrm{RMS}(\hat{\varepsilon}_n) \to 0$, then $P(|\hat{\varepsilon}_n - \varepsilon_n| \ge \tau) \to 0$, for any $\tau > 0$, i.e. the error estimator is consistent.

Good error estimation performance requires that the bias, deviation variance, and RMS be as close as possible to zero.

3. Generalized Resubstitution Error Estimators

In this section, we introduce a general framework for resubstitution-like error estimators based on arbitrary empirical probability measures, which we call *generalized resubstitution error estimators*. These error estimators share with plain resubstitution the fact that training additional classifiers is not required. Like resubstitution, they are generally fast and low-variance. They are also nonrandomized, unless Monte-Carlo approximations are required to compute the estimator.

Definition 1. A generalized resubstitution error estimator $\hat{\varepsilon}_n$ can be written as:

$$\hat{\varepsilon}_n \,=\, \hat{\nu}_n(\{(\mathbf{x}, y) : \psi_n(\mathbf{x}) \neq y\})\,, \tag{7}$$

where $\hat{\nu}_n$ is an *empirical probability measure*, i.e., a random probability measure supported on $R^d \times \{0, 1, \ldots, c-1\}$ that is a function of the sample data.

If $\hat{\nu}_n$ is sufficiently close to ν, in a suitable sense, then $\hat{\varepsilon}_n$ is a good estimator of ε_n. However, it is important to note that the criterion of performance is not whether $\hat{\nu}_n$ is close to ν, but it is whether $\hat{\varepsilon}_n$ is close to ε_n, i.e., obtaining accurate error estimators is the goal, rather than performing accurate distribution estimation (which is generally a more difficult problem).

An important example of empirical probability measure is given next.

Definition 2. The standard empirical probability measure ν_n is given by:

$$\nu_n \,=\, \frac{1}{n}\sum_{i=1}^{n} \delta_{\mathbf{X}_i, Y_i}\,, \tag{8}$$

where $\delta_{\mathbf{X}_i, Y_i}$ is the (random) point measure located at $(\mathbf{X}_i, Y_i)$,

$$\delta_{\mathbf{X}_i, Y_i}(A) \,=\, I((\mathbf{X}_i, Y_i) \in A)\,,$$

for each event A.

Hence, $\nu_n(A)$ puts discrete mass $1/n$ on each data point, and thus yields the fraction of points in S_n that are contained in A.

The basic example of generalized resubstitution error estimator is produced by the standard empirical probability measure.

Definition 3. The standard resubstitution error estimator $\hat{\varepsilon}_n$ is a generalized resubstitution error estimator with the standard empirical probability measure ν_n,

$$\hat{\varepsilon}_n^r \,=\, \nu_n(\{(\mathbf{x}, y) : \psi_n(\mathbf{x}) \neq y\}) \,=\, \frac{1}{n}\sum_{i=1}^{n} I(\psi_n(X_i) \neq Y_i)\,. \tag{9}$$

Notice that ν_n has no hyperparameters that allow estimator bias and variance to be tuned. We will consider below further examples of generalized resubstitution error estimators that do allow bias and variance to be tuned. However, the standard resubstitution is important as a baseline for performance and for proving theoretical results.

Next, we consider the natural large-sample question of whether a generalized resubstitution error estimator approaches the true classification error as the training sample size increases to infinity. In particular, we are interested in the questions of consistency, i.e., whether $\hat{\varepsilon}_n \to \varepsilon_n$ a.s. as well as asymptotic unbiasedness, i.e., whether $E[\hat{\varepsilon}_n] \to E[\varepsilon_n]$, as $n \to \infty$.

First, note that for any fixed event $A \subseteq R^d \times \{0, 1, \ldots, c-1\}$, the standard empirical probability measure satisfies

$$\nu_n(A) \;=\; \frac{1}{n}\sum_{i=1}^{n} I((\mathbf{X}_i, Y_i) \in A) \;\to\; \nu(A) \;\; \text{a.s.},$$

by the Strong Law of Large Numbers (SLLN). Hence, for a fixed classifier ψ, we can plug in the fixed set $A = \{(\mathbf{x}, y) : \psi(\mathbf{x}) \neq y\}$ in the previous equation and conclude that the empirical classification error converges to the true error a.s. But this is not enough to obtain results concerning classifiers ψ_n designed from the data S_n, since these concern events $A_n = \{(\mathbf{x}, y) : \psi_n(\mathbf{x}) \neq y\}$, which are not fixed. What is needed instead is a *uniform* SLLN:

$$\sup_{A \in \mathcal{A}} |\nu_n(A) - \nu(A)| \;\to\; 0 \;\; \text{a.s.}, \tag{10}$$

where $\mathcal{A}$ is a family of sets that must contain all events $A_n = \{(\mathbf{x}, y) : \psi_n(\mathbf{x}) \neq y\}$ that can be produced by the classification rule. In the two-class case, the Vapnik-Chervonenkis (VC) Theorem (Vapnik and Chervonenkis, 1971; Devroye et al., 1996) guarantees that (10) holds if $\mathcal{A}$ is small enough, in the sense that its *VC dimension* $V_{\mathcal{A}}$ is finite. The VC dimension is a nonnegative integer that measures the size of $\mathcal{A}$; a smaller VC dimension implies that the classification rule is more constrained and less sensitive to the data S_n, i.e., it is less prone to overfitting at a fixed sample size. For example, a linear classification rule in R^d has VC dimension $d + 1$, which is finite, and small in low-dimensional spaces. If $V_{\mathcal{A}} < \infty$, then, according to the VC Theorem,

$$P\left(\sup_{A \in \mathcal{A}} |\nu_n(A) - \nu(A)| > \tau\right) \leq 8(n+1)^{V_{\mathcal{A}}} e^{-n\tau^2/32}, \quad \text{for all } \tau > 0\,.$$

The term $e^{-n\tau^2/32}$ dominates, and the bound decreases exponentially fast as $n \to \infty$. It then follows from the First Borel-Cantelli Lemma that $\sup_{A \in \mathcal{A}} |\nu_n(A) - \nu(A)| \to 0$ a.s. (Braga-Neto, 2020, Thm A.8). (Strictly speaking, it is necessary to assume that events of the kind $\sup_{A \in \mathcal{A}} |\nu_n(A) - \nu(A)| > \tau$ are measurable. General conditions to ensure that are discussed in (Pollard, 1984); such conditions are tacitly assumed throughout in this paper.) These facts are used in the proof of the following result.

Theorem 1 *In the two-class case, if the family $\mathcal{A}$ of all events $A_n = \{(\mathbf{x}, y) : \psi_n(\mathbf{x}) \neq y\}$ that can be produced by a classification rule has finite VC dimension, and the generalized empirical probability measure converges uniformly to the standard empirical probability measure as sample size increases, i.e.,*

$$\sup_{A \in \mathcal{A}} |\hat{\nu}_n(A) - \nu_n(A)| \;\to\; 0 \;\; \textit{a.s.}, \tag{11}$$

then the generalized resubstitution error estimator is consistent, $\hat{\varepsilon}_n \to \varepsilon_n$ a.s., as well as asymptotically unbiased, $E[\hat{\varepsilon}_n] \to E[\varepsilon_n]$, as $n \to \infty$, regardless of the feature-label distribution.

Proof. From

$$\begin{aligned}|\hat{\nu}_n(A) - \nu(A)| &= |\hat{\nu}_n(A) - \nu_n(A) + \nu_n(A) - \nu(A)| \\ &\leq |\hat{\nu}_n(A) - \nu_n(A)| + |\nu_n(A) - \nu(A)|\,,\end{aligned}$$

it follows that

$$\sup_{A\in\mathcal{A}} |\hat{\nu}_n(A) - \nu_n(A)| \leq \sup_{A\in\mathcal{A}} |\hat{\nu}_n(A) - \nu_n(A)| + \sup_{A\in\mathcal{A}} |\nu_n(A) - \nu(A)|\,.$$

The first term on the right converges to zero a.s. by hypothesis, while the second term does so by virtue of the VC Theorem. Hence, the left-hand side must also converge to zero a.s.,

$$\sup_{A\in\mathcal{A}} |\hat{\nu}_n(A) - \nu(A)| \to 0 \quad \text{a.s.}$$

Since

$$|\hat{\varepsilon}_n - \varepsilon_n| = |\hat{\nu}_n(A_n) - \nu(A_n)| \leq \sup_{A\in\mathcal{A}} |\hat{\nu}_n(A) - \nu(A)|,$$

it follows that $|\hat{\varepsilon}_n - \varepsilon_n| \to 0$ a.s. and the generalized resubstitution estimator is consistent.

Furthermore, since all random variables are uniformly bounded, the Dominated Convergence Theorem implies (Braga-Neto, 2020, Thm A.7) that

$$|E[\hat{\varepsilon}_n - \varepsilon_n]| \leq E[|\hat{\varepsilon}_n - \varepsilon_n|] \to 0\,,$$

i.e., the generalized resubstitution error estimator is asymptotically unbiased. All of these results are distribution-free, holding for any feature-label distribution ν. ■

The previous result applies trivially to the plain resubstitution estimator, a fact that has been long known, by virtue of the VC Theorem (Devroye et al., 1996). For other generalized resubstitution estimators, condition (11) needs to be checked. The point of (11) is that the generalized resubstitution should "look like" more and more like plain resubstitution as sample size increases.

4. Generalized Resubstitution based on Smoothing the Error Count

In this Section, we consider various classes of generalized resubstitution error estimators, which are all based on the family of empirical probability measures of the form:

$$\hat{\nu}_n = \frac{1}{n}\sum_{i=1}^{n} \beta_{n,\mathbf{X}_i,Y_i}\,, \tag{12}$$

where $\beta_{n,\mathbf{X}_i,Y_i}$ is a random probability measure depending on the training point $\mathbf{X}_i, Y_i$. Comparing this to (8), we realize that the empirical probability measure in (12) can be seen as smoothed version of the standard empirical probability measure, where $\beta_{n,\mathbf{X}_i,Y_i}$ provides a smoothed version of the point measure $\delta_{\mathbf{X}_i,Y_i}$. This is not the only case of useful empirical probability measure for generalized resubstitution error estimation; in Section 5.1, we give examples that are not of the form in (12).

Notice that a sufficient condition for (11) in Theorem 1 is the uniform convergence of the smoothed measure $\beta_{n,\mathbf{X}_i,Y_i}$ to the point measure $\delta_{\mathbf{X}_i,Y_i}$ for any training point $(\mathbf{X}_i, Y_i)$:

$$\sup_{A\in\mathcal{A}} |\beta_{n,\mathbf{X}_i,Y_i}(A) - \delta_{\mathbf{X}_i,Y_i}(A)| \to 0 \quad \text{a.s.} \tag{13}$$

4.1 Bolstered Resubstitution

Given an event $A \subseteq R^d \times \{0, 1, \ldots, c-1\}$, define its slices by

$$A_y \;=\; \{\mathbf{x} \in R^d \mid (\mathbf{x}, y) \in A\}\,, \quad y = 0, 1, \ldots, c-1\,. \tag{14}$$

It is clear that A_y is an event (i.e., a Borel set) in R^d for each y. Note that $\delta_{\mathbf{X}_i,Y_i}(A) = \delta_{\mathbf{X}_i}(A_{Y_i})$, where $\delta_{\mathbf{X}_i}$ is a point measure in R^d. Similarly, let $\beta_{n,\mathbf{X}_i,Y_i}(A) = \mu_{n,\mathbf{X}_i,Y_i}(A_{Y_i})$, where $\mu_{n,\mathbf{X}_i,Y_i}$ is an empirical probability measure on R^d. Though discrete bolstering is possible, in practice the *bolstered probability measure* $\mu_{n,\mathbf{X}_i,Y_i}$ is assumed to be absolutely continuous, with density function $p_{n,\mathbf{X}_i,Y_i}(\mathbf{x})$, so that

$$\beta_{n,\mathbf{X}_i,Y_i}(A) \;=\; \int_{A_{Y_i}} p_{n,\mathbf{X}_i,Y_i}(\mathbf{x})\, d\mathbf{x}\,.$$

The probability densities $p_{n,\mathbf{X}_i,Y_i}$ are called *bolstering kernels*. Plugging $\beta_{n,\mathbf{X}_i,Y_i}(A)$ in (12), and then in (7), yields the *bolstered resubstitution error estimator* proposed in (Braga-Neto and Dougherty, 2004b) (here extended to the multi-class case). Note that the misclassification event $\{(\mathbf{x}, y) : \psi_n(\mathbf{x}) \neq y\}$ has slices $A_y = \{\mathbf{x} : \psi_n(\mathbf{x}) \neq y\}$. The bolstered resubstitution error estimator can be thus written as:

$$\hat{\varepsilon}_n^{br} \;=\; \frac{1}{n}\sum_{i=1}^{n} \int_{\{\mathbf{x}:\psi_n(\mathbf{x})\neq Y_i\}} p_{n,\mathbf{X}_i,Y_i}(\mathbf{x})\, d\mathbf{x}\,. \tag{15}$$

The integral in (15) gives the error contribution made by training point $(\mathbf{X}_i, Y_i)$; these are real-valued numbers between 0 and 1, unlike plain resubstitution, in which contributions are 0 or 1. See Figure 2 for an illustration, where the bolstering kernels are uniform distributions over disks centered at each of the points $\mathbf{X}_i$, with radii that depend on Y_i. Notice that this allows counting partial errors, including errors for correctly classified points that are near the decision boundary.

In some cases, it is possible to solve the integrals in (15) analytically, and the estimator is fast and low-variance. Otherwise, one has to apply approximations. For example, simple Monte-Carlo integration yields:

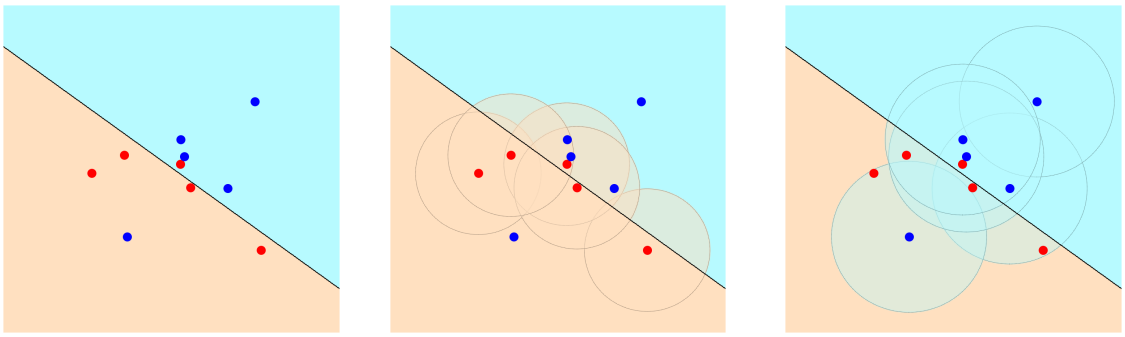
$$\hat{\varepsilon}_n^{br} \;\approx\; \frac{1}{nM}\sum_{i=1}^{n}\sum_{j=1}^{M} I(\psi_n(\mathbf{X}_{ij}^{\mathrm{MC}}) \neq Y_i)\,, \tag{16}$$

where $\{\mathbf{X}_{ij}^{\mathrm{MC}}; j = 1, \ldots, M\}$ are random points drawn from the density $p_{n,i}$, for $i = 1, \ldots, n$. In this case, the estimation procedure is randomized due to MC sampling.

The radii of the disks in Figure 2 are hyperparameters that can be adjusted to control the bias of the resulting generalized resubstitution estimator. If the radii are too small, the estimator is close to plain resubstitution and could be optimistically biased; if they are too large, the estimator tends to be pessimistically biased.

The most common choice for bolstering kernels are multivariate Gaussian densities with mean $\mathbf{X}_i$ and covariance matrix $K_{n,\mathbf{X}_i,Y_i}$:

$$p_{n,\mathbf{X}_i,Y_i}(\mathbf{x}) \;=\; \frac{1}{\sqrt{(2\pi)^d\, \det(K_{n,\mathbf{X}_i,Y_i})}} \exp\left(-\frac{1}{2}\,(\mathbf{x}-\mathbf{X}_i)^T K_{n,\mathbf{X}_i,Y_i}^{-1}(\mathbf{x}-\mathbf{X}_i)\right)\,.$$



Original classifier Bolstering kernels (class 0) Bolstering kernels (class 1)

Figure 2: Bolstered resubstitution for a linear classifier with uniform circular bolstering kernels. The error contribution made by each point is the area of the disk segment extending across the decision boundary (if any) divided by the area of the entire disk. The bolstered resubstitution error is the sum of all contributions divided by the number of points. Reproduced from (Braga-Neto, 2020).

If the matrices $K_{n,\mathbf{X}_i,Y_i}$ are diagonal, with freely adjustable diagonal elements, then the procedure is known as *Naive-Bayes bolstering* (Jiang and Braga-Neto, 2014).

We consider next in detail the case $K_{n,\mathbf{X}_i,Y_i} = \sigma^2_{n,Y_i} I_d$. The hyperparameters here are the c standard deviations $\sigma_{n,j}$ for each class (here, kernel variance is not a function of $\mathbf{X}_i$). This demands much less effort than the Naive-Bayes case, which requires in general nd hyperparameters. (Nevertheless, the analysis below could be extended to the Naive-Bayes case with more effort.) The hyperparameters $\sigma^{n,j}$ can be estimated by making the median distance of a point sampled from the corresponding kernel to the origin match the mean minimum distance $\hat{d}_{n,j}$ among training points in class j:

$$\hat{d}_{n,j} = \frac{1}{n_j} \sum_{i=1}^{n_j} ||\mathbf{X}_{ij} - \mathbf{X}'_{ij}||\,, \quad j = 0, 1, \ldots, c-1\,, \tag{17}$$

where n_j is the number of points from class j ($n_j \geq 2$ is assumed), $\mathbf{X}_{ij}$ is a point in class j, and $\mathbf{X}'_{ij}$ is its nearest neighbor in class j (Braga-Neto and Dougherty, 2004c).

Now, let R be the random variable corresponding to the distance to the origin of a point randomly selected from a unit-variance spherically-symmetric density with cumulative distribution function $F_R(x)$. The median distance of such a point to the origin is $\alpha_d = F_R^{-1}(1/2)$, where the subscript d indicates explicitly that α_d depends on the dimensionality. If the density has variance σ^2, all distances get multiplied by σ. Hence, $\sigma_{n,j}$ is the solution of the equation $\sigma_{n,j}\alpha_d = \hat{d}_{n,j}$, i.e.,

$$\sigma_{n,j} = \frac{\hat{d}_{n,j}}{\alpha_d}\,, \quad j = 0, 1, \ldots, c-1 \tag{18}$$

The constant α_d can be interpreted as a "dimensionality correction," which adjusts the value of the estimated mean distance to account for the feature space dimensionality. Indeed, this approach to selecting the hyperparameters is applicable to any spherically-symmetric kernel, such as the uniform disks of Figure 2. In the case of spherical Gaussian densities, R is

distributed as a *chi* random variable with d degrees of freedom, and the median $\alpha_d = F_R^{-1}(1/2)$ can be easily computed numerically. For example, the values up to five dimensions are $\alpha_1 = 0.674$, $\alpha_2 = 1.177$, $\alpha_3 = 1.538$, $\alpha_4 = 1.832$, $\alpha_5 = 2.086$.

Next, we consider the asymptotic properties of the bolstered resubstitution estimator with spherical Gaussian kernels in the case $c = 2$. First, we define a classification rule to be *regular* if it produces "thin" decision boundaries. In the general case $c \geq 2$, the decision boundary D of classifier ψ_n is

$$D = \bigcup_{y=0}^{c-1} \partial A_y$$

where $A_y = \{\mathbf{x} : \psi_n(\mathbf{x}) \neq y\}$ are the misclassification event slices, as defined previously, and a point is in ∂A_y if it does not belong to the interior of either A_y or A_y^c. A classification rule Ψ_n is regular if D has Lebesgue measure zero for all its classifiers ψ_n. If the distribution of $\mathbf{X}$ is absolutely continuous (with respect to the Lebesgue measure), i.e., if $\mathbf{X}$ is a continuous feature vector in the usual sense, then the probability that a training point $\mathbf{X}_i$ sits on the decision boundary is zero. The vast majority, if not all, classification rules encountered in practice are regular.

Theorem 2 *In the two-class case, if Ψ_n is a regular classification rule with finite VC dimension and the distribution of $\mathbf{X}$ is absolutely continuous, then the bolstered resubstitution estimator with spherical Gaussian kernels, with hyperparameters $\sigma_{n,j}$ selected as in* (18)*, is consistent and asymptotically unbiased.*

Proof. By virtue of Theorem 1 and (12), it suffices to show that (13) holds, which in the present case reduces to proving that

$$\sup_{A \in \mathcal{A}} |\mu_{n,\mathbf{X}_i,Y_i}(A_{Y_i}) - \delta_{\mathbf{X}_i}(A_{Y_i})| \to 0 \ \text{ a.s.},$$

where $\mathcal{A}$ s the family of all events $\{(\mathbf{x}, y) : \psi_n(\mathbf{x}) \neq y\}$ that can be produced by the classification rule. Notice that, for any given $\tau > 0$, whenever $\sup_{A \in \mathcal{A}} |\mu_{n,\mathbf{X}_i,Y_i}(A_{Y_i}) - \delta_{\mathbf{X}_i}(A_{Y_i})| > \tau$, there is an $A^* \in \mathcal{A}$, which is a function of the data, such that $|\mu_{n,\mathbf{X}_i,Y_i}(A^*_{Y_i}) - \delta_{\mathbf{X}_i}(A^*_{Y_i})| > \tau$, with probability 1. In other words,

$$P\left(|\mu_{n,\mathbf{X}_i,Y_i}(A^*_{Y_i}) - \delta_{\mathbf{X}_i}(A^*_{Y_i})| > \tau \;\middle|\; \sup_{A \in \mathcal{A}} |\mu_{n,\mathbf{X}_i,Y_i}(A_{Y_i}) - \delta_{\mathbf{X}_i}(A^*_{Y_i})| > \tau\right) = 1,$$

which in turn implies that

$$P\left(\sup_{A \in \mathcal{A}} |\mu_{n,\mathbf{X}_i}(A_{Y_i}) - \delta_{\mathbf{X}_i}(A_{Y_i})| > \tau\right) \leq P\left(|\mu_{n,\mathbf{X}_i}(A^*_{Y_i}) - \delta_{\mathbf{X}_i}(A^*_{Y_i})| > \tau\right). \tag{19}$$

By regularity of the classification rule, $\mathbf{X}_i$ belongs to the interior of $A^*_{Y_i}$ or $(A^*_{Y_i})^c$ with probability 1. Hence, we can find an open ball $B(\mathbf{X}_i, \rho)$ centered on $\mathbf{X}_i$ that is entirely contained in A_{Y_i} or $A^c_{Y_i}$. If the variance $\sigma^2_{n,i}$ tends to zero as at least $O(n)$, the Gaussian measure will concentrate exponentially fast inside such a ball, such that $P\left(|\mu_{n,\mathbf{X}_i}(A^*_{Y_i}) - \delta_{\mathbf{X}_i}(A^*_{Y_i})| > \tau\right) \to 0$ exponentially fast, for any $\tau > 0$, and the Theorem is proved, via (19) and the First Borel-Cantelli Lemma.

From (17) and (18), it suffices to show that the nearest neighbor $\mathbf{X}'_{ij}$ to $\mathbf{X}_{ij}$ converges to $\mathbf{X}_{ij}$ exponentially fast as $n \to \infty$. Note that, for any $\tau > 0$,[1]

$$P(||\mathbf{X}'_{ij} - \mathbf{X}_{ij}|| > \tau) = P(||\mathbf{X}_{kj} - \mathbf{X}_{ij}|| > \tau; \text{ for all } k \neq i) = (1 - P(||\mathbf{X}_{lj} - \mathbf{X}_{ij}|| < \tau))^{n_j}, \tag{20}$$

for some $l \neq i$. Notice that, since $P(Y = j) > 0$, $n_j \to \infty$ as $O(n)$ a.s. as $n \to \infty$. If we can show that $P(||\mathbf{X}_{lj} - \mathbf{X}_{ij}|| < \tau) > 0$, then it follows from (20) that $P(||\mathbf{X}_{lj} - \mathbf{X}|| > \tau) \to 0$ exponentially fast a.s. and the claim is proved. To ease notation, let $\mathbf{Z}' = \mathbf{X}'_{ij}$ and $\mathbf{Z} = \mathbf{X}_{ij}$. Since $\mathbf{Z}'$ and $\mathbf{Z}$ are independent and identically distributed with density $p_{\mathbf{X}}$, $\mathbf{Z}' - \mathbf{Z}$ has a density $p_{\mathbf{Z}'-\mathbf{Z}}$, given by the classical convolution formula:

$$p_{\mathbf{Z}'-\mathbf{Z}}(\mathbf{w}) \;=\; \int p_{\mathbf{X}}(\mathbf{w}+\mathbf{u})\, p_{\mathbf{X}}(\mathbf{w})\, d\mathbf{u}\,.$$

From this, we have $p_{\mathbf{Z}'-\mathbf{Z}}(\mathbf{0}) = \int p^2_{\mathbf{X}}(\mathbf{u})\, d\mathbf{u} > 0$. It follows, by continuity of the integral, that $p_{\mathbf{Z}'-\mathbf{Z}}$ must be nonzero in a neighborhood of $\mathbf{0}$, i.e., $P(||\mathbf{Z}' - \mathbf{Z}|| < \tau) > 0$, as was to be shown. ■

4.2 Posterior-Probability Generalized Resubstitution

The bolstered empirical probability measure relies on measures $\mu_{n,\mathbf{X}_i}$ on R^d, which provide smoothing in the $\mathbf{X}$ direction. If one performs smoothing in the Y direction, the so-called posterior-probability empirical probability measure results.

Given an event $A \subseteq R^d \times \{0, 1, \ldots, c-1\}$, define the slices

$$A_{\mathbf{x}} \;=\; \{y \in \{0, 1, \ldots, c-1\} \mid (\mathbf{x}, y) \in A\}\,, \quad \mathbf{x} \in R^d\,. \tag{21}$$

(Compare to the slices in (14).) Note that $\delta_{\mathbf{X}_i,Y_i}(A) = \delta_{Y_i}(A_{\mathbf{X}_i})$, where δ_{Y_i} is a point measure on $\{0, 1, \ldots, c-1\}$. Similarly, let $\beta_{n,\mathbf{X}_i,Y_i}(A) = \eta_{n,\mathbf{X}_i,Y_i}(A_{\mathbf{X}_i})$, where $\eta_{n,\mathbf{X}_i,Y_i}$ is an empirical probability measure on $\{0, 1 \ldots, c-1\}$. This is called a *posterior-probability measure* as $\eta_{n,\mathbf{X}_i,Y_i}(A_{\mathbf{X}_i})$ is to be interpreted as a "posterior-probability" estimate $\widehat{P}_n(Y_i \in A_{\mathbf{X}_i} \mid \mathbf{X} = \mathbf{X}_i)$. Plugging $\beta_{n,\mathbf{X}_i,Y_i}(A)$ in (12), and then in (7), yields the *posterior-probability resubstitution error estimator* (e.g., see (Lugosi and Pawlak, 1994), here extended to the multi-class case).

If $A = \{(\mathbf{x}, y) : \psi_n(\mathbf{x}) \neq y\}$ is the misclassification event, then $A_{\mathbf{x}} = \{\psi_n(\mathbf{x})\}^c$. Using the $\widehat{P}_n$ notation, it is easy to see that the posterior-probability resubstitution error estimator can be written as:

$$\hat{\varepsilon}_n^{\text{ppr}} \;=\; \frac{1}{n}\sum_{i=1}^{n} \widehat{P}_n(\psi_n(\mathbf{X}_i) \neq Y_i \mid \mathbf{X} = \mathbf{X}_i)\,. \tag{22}$$

Here, $\widehat{P}_n(\psi_n(\mathbf{X}_i) \neq Y_i \mid \mathbf{X} = \mathbf{X}_i)$ is the error contribution made by training point $(\mathbf{X}_i, Y_i)$, rather than 0 or 1 as in plain resubstitution. The idea is that if one is more confident that the classifier disagrees with the training label, this error should count more, and the reverse is true if one is not. This smooths the error count of plain resubstitution and reduces variance.

The simplest concrete example is afforded by k-nearest neighbor (kNN) posterior probability estimation. Let $\{y^1(\mathbf{x}), \ldots, y^k(\mathbf{x})\}$ denote the labels of the k nearest training points to

1. Equation (20) appears in a similar context in the proof of the Cover-Hart Theorem for nearest-neighbor classification (Cover and Hart, 1967). The rest of the argument is distinct.

$\mathbf{x}$, for $k = 1, \ldots, n$. The k-nearest-neighbor (kNN) posterior probability measure is defined by

$$\widehat{P}_n(Y = y \mid \mathbf{X} = \mathbf{x}) \;=\; \frac{1}{k} \sum_{j=1}^{k} I(y^j(\mathbf{x}) = y)\,, \tag{23}$$

for $\mathbf{x} \in R^d$. This makes sense since the more labels y there are in the neighborhood of $\mathbf{x}$, the more likely it should be that its label is y. Plugging (23) into (22) leads to the kNN posterior-probability error estimator:

$$\hat{\varepsilon}_n^{\mathrm{kNN}} \;=\; \frac{1}{nk} \sum_{i=1}^{n} \sum_{j=1}^{k} I(\psi_n(\mathbf{X}_i) \neq y^j(\mathbf{X}_i))\,, \tag{24}$$

Clearly, the case $k = 1$ reduces to plain resubstitution. It is clear that if $k = k_n$ is a function of n, and $k_n \to 1$ as $n \to \infty$, (13) is satisfied and the kNN posterior probability estimator is consistent and asymptotically unbiased, under the conditions of Theorem 1.

4.3 Bolstered Posterior-Probability Generalized Resubstitution

A novel class of generalized resubstitution estimator results if one performs smoothing in both the $\mathbf{X}$ and Y directions. Notice that $\delta_{\mathbf{X}_i,Y_i}(A) = \delta_{\mathbf{X}_i}(A_{Y_i})\delta_{Y_i}(A_{\mathbf{X}_i})$, where the slices A_y and $A_{\mathbf{x}}$ are defined in (14) and (21), respectively. Let $\beta_{n,\mathbf{X}_i,Y_i}(A) = \mu_{n,\mathbf{X}_i,Y_i}(A_{Y_i})\eta_{n,\mathbf{X}_i,Y_i}(A_{\mathbf{X}_i})$, where $\mu_{n,\mathbf{X}_i,Y_i}$ and $\eta_{n,\mathbf{X}_i,Y_i}$ are respectively the bolstered and posterior-probability empirical probability measures defined previously. Plugging $\beta_{n,\mathbf{X}_i,Y_i}(A)$ in (12), and then in (7), yields the *bolstered posterior-probability resubstitution error estimator*, a new estimator that combines features of bolstered and posterior-probability resubstitution. Using the $\widehat{P}_n$ notation, it is easy to see that the bolstered posterior-probability resubstitution error estimator can be written as:

$$\hat{\varepsilon}_n^{\mathrm{bppr}} \;=\; \frac{1}{n} \sum_{i=1}^{n} \left(\int_{\{\mathbf{x}:\psi_n(\mathbf{x}) \neq Y_i\}} p_{n,\mathbf{X}_i,Y_i}(\mathbf{x})\, d\mathbf{x} \right) \widehat{P}_n(\psi_n(\mathbf{X}_i) \neq Y_i \mid \mathbf{X} = \mathbf{X}_i)\,. \tag{25}$$

This estimator seeks to combine the bias-reducing properties of the bolstered estimator with the variance-reducing properties of the posterior-probability estimator.

In the two-class case, with the Gaussian bolstered and k-nearest neighbor empirical probability measures, it can be shown that the bolstered posterior-probability error estimator for a linear classifier $\psi_n(\mathbf{x}) = I(\mathbf{a}_n^T\mathbf{x} + b_n > 0)$ can be computed analytically as

$$\hat{\varepsilon}_n^{\mathrm{GS-kNN}} = \frac{1}{nk} \sum_{i=1}^{n} \left[\Phi\left(\frac{(-1)^{1-Y_i}(\mathbf{a}_n^T\mathbf{X}_i + b_n)}{\sqrt{\mathbf{a}_n^T K_{n,\mathbf{X}_i,Y_i}\mathbf{a}_n}} \right) \left(\sum_{j=1}^{k} I((-1)^{y^j(\mathbf{X}_i)}(\mathbf{a}_n^T\mathbf{x} + b_n) > 0) \right) \right]. \tag{26}$$

This estimator is consistent and asymptotically unbiased under the conditions of Theorem 2 and $k_n \to 1$ as $n \to \infty$.

5. Extensions

In this section we give additional examples and discuss the extension of the framework to cross-validation and test-set error estimators.

5.1 Bayesian Generalized Resubstitution

All previous examples of generalized resubstitution were based on smoothing the error count. In this section, we given an example that shows that the family of generalized resubstitution estimators is more general than that.

Let the unknown probability measure ν belongs to a parametric family of probability measures $\{\nu_\theta; \theta \in \Theta\}$. Assume a prior distribution $p(\theta)$ for the parameter, and let $p(\theta \mid S_n)$ be its posterior distribution. We define the *Bayesian empirical probability measure* as:

$$\hat{\nu}_n^{\text{bay}}(A) \;=\; \int_\Theta \nu_\theta(A)\, p(\theta \mid S_n)\, d\theta\,,$$

for each event $A \subseteq R^d \times \{0, 1, \ldots, c-1\}$. Plugging this for $\hat{\nu}_n$ in (7) yields the *Bayesian generalized resubstitution estimator*. This family of Bayesian error estimators was proposed in (Dalton and Dougherty, 2011a), and later studied by the same authors in a series of papers (Dalton and Dougherty, 2011b,c, 2012a,b). In (Dalton and Dougherty, 2011a,b), analytical expressions for the Bayesian resubstitution error estimator are given in a few cases. In more general cases, the required integrals must be computed by numerical methods, such as Markov-Chain Monte-Carlo, making the error estimator randomized.

5.2 Gaussian-Process Generalized Resubstitution

Gaussian-process classification and regression (Rasmussen and Williams, 2006) have become very popular recently. Using Gaussian process regression to estimate posterior probabilities in (22) leads to a *Gaussian-process generalized resubstitution estimator*. The idea of using Gaussian processes in a posterior-probability error estimator was previously suggested in (Hefny and Atiya, 2010).

Briefly, given the data $S_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$, consider a Gaussian vector of *latent values* :

$$\mathbf{f} \;=\; (f_0(\mathbf{X}_1), \ldots, f_0(\mathbf{X}_n), \ldots, f_{c-1}(\mathbf{X}_1), \ldots, f_{c-1}(\mathbf{X}_n)\,,$$

corresponding to samples of a vector-valued Gaussian process. The vectors $\mathbf{f}_y = (f_y(\mathbf{X}_1), \ldots, f_y(\mathbf{X}_n))$ are assumed to be uncorrelated with each other, with distribution $\mathcal{N}(\mathbf{0}, K_y)$, where the covariance matrix has elements:

$$k_y(\mathbf{x}, \mathbf{x}') \;=\; C_y \exp\left(-\frac{1}{2\sigma_y^2}(\mathbf{x} - \mathbf{x}')^2\right)\,,$$

and the constant $C_y > 0$ ensures that $\int k_y(\mathbf{x}, \mathbf{x}')d\mathbf{x} = 1$. Therefore $\mathbf{f}$ is zero-mean and has a block-structured covariance matrix with the K_y matrices along the diagonal. The posterior distribution of vector $\mathbf{f}^* = (f_0^*(\mathbf{x}), \ldots, f_{c-1}^*(\mathbf{x}))$ at each test point $\mathbf{x} \in R^d$ is given by Rasmussen and Williams (2006):

$$p(\mathbf{f}^* \mid S_n) \;=\; \int p(\mathbf{f}^* \mid S_n, \mathbf{f}) p(\mathbf{f} \mid S_n)\, d\mathbf{f}$$

If the value $f_y^*(\mathbf{x})$ is large compared to the other values $f_{y'}^*(\mathbf{x})$, for $y' \neq y$, then $\mathbf{x}$ is likely to be from class y, for $y = 0, \ldots, c-1$. Accordingly, we define a vector $(\xi_0(\mathbf{f}^*), \ldots, \xi_{c-1}(\mathbf{f}^*))$ as the output of a *softmax* function on $\mathbf{f}^*$ and define the posterior probability function estimator

$$\widehat{P}_n(Y = y \mid \mathbf{X} = \mathbf{x}) \;=\; \int \xi_y(\mathbf{f}^*) p(\mathbf{f}^* \mid S_n)\, d\mathbf{f}^*\,.$$

Plugging this in (22) yields the Gaussian-process error estimator. The hyperparameter σ_y correspond to the length-scale of process f_y, for $y = 0, \ldots, c-1$. These hyperparameters are critical to the bias properties of the error estimator; in the literature of Gaussian processes, they are typically chosen by maximum-likelihood methods (Rasmussen and Williams, 2006).

5.3 Generalized Cross-Validation and Test-Set Error Estimators

The *generalized cross-validation* error estimation procedure results from a random *resampling* process that produces R subsets $S^i_{n_i} = \{(\mathbf{X}^i_1, Y^i_1) \ldots, (\mathbf{X}^i_{n_i}, Y^i_{n_i})\}$, where $1 \leq n_i \leq n$. A classification rule Ψ_{n_i} is applied to $S^i_{n_i}$ to obtain a classifier $\psi^i_{n_i}$, for $i = 1, \ldots, R$. It is also assumed that the resampling process produces R empirical probability measures $\nu_{n,1}, \ldots, \nu_{n,R}$. The generalized cross-validation error estimator is

$$\hat{\varepsilon}^{\text{cvk}}_n = \frac{1}{R} \sum_{i=1}^{R} \nu_{n,i}(\{(\mathbf{x}, y) : \psi^{(i)}_{n_i}(\mathbf{x}) \neq y\}) .$$

The *generalized leave-one-out* error estimator corresponds to the special case $R = n$, $n_i = n-1$, and S^i_{n-1} equal to the original data with the point $(\mathbf{X}_i, Y_i)$ deleted, for $i = 1, \ldots, n$.

The *generalized test-set* error estimator is based on an empirical probability measure ν^t_m, which is a function of independent test data S^t_m, to produce the error estimator

$$\hat{\varepsilon}^{\text{t}}_{n,m} = \nu^t_m(\{(\mathbf{x}, y) : \psi_n(\mathbf{x}) \neq y\}) .$$

6. Experimental Results

In this section, the performance of several of the generalized resubstitution error estimators discussed in this paper is evaluated empirically, by means of classification experiments with a variety of linear and nonlinear classification rules. Using synthetic data, we compare the performance of generalized resubstitution error estimators against each other and against representative cross-validation and bootstrap error estimators, both in terms of accuracy and computation speed. We also report the results of an experiment on the applicability of generalized resubstitution in image classification by convolutional neural networks (CNN), using the LeNet-5 CNN architecture and the MNIST image data set. In the latter case, due to the high-dimensionality of the feature space, we employ Naive-Bayes bolstered resubstitution, as well as a simple data-driven calibration procedure to further reduce the bias.

6.1 Synthetic Data Experiment

In this section we employ synthetic data to investigate the performance of the plain resubstitution ("resub"), bolstered resubstitution with spherical Gaussian kernels, with hyperparameter estimated as in (18) ("bolster"), a variant of bolstering that applies the kernels only to correctly-classified training points ("semi-bolster"), the k-NN posterior-probability estimator, with $k = 3$ ("3NNpp"), and the bolstered k-NN posterior probability estimator with spherical Gaussian kernels, with hyperparameters as in the previous two cases ("bolster 3NNpp"). For comparison with resampling error estimators, we also include the 10-fold cross-validation estimator ("cross valid") (Braga-Neto and Dougherty, 2015b) and the zero bootstrap estimator ("boot") (Efron, 1983) in the experiments.

Sample Size	Linear SVM	RBF SVM	CART	3NN	5NN	7NN
$n = 20$	0.311	0.311	0.377	0.329	0.315	0.315
$n = 40$	0.268	0.255	0.350	0.297	0.286	0.273
$n = 60$	0.244	0.234	0.341	0.288	0.267	0.263
$n = 80$	0.233	0.232	0.332	0.283	0.264	0.251
$n = 100$	0.224	0.225	0.330	0.277	0.261	0.249

Table 1: Classification errors in the synthetic data experiment.

The generative model consists of multivariate Gaussian distributions for each of two classes, containing d_n noisy features and $d - d_n$ informative features, for each sample size. The values of the noisy features are sampled independently from a zero-mean, unit-variance Gaussian distribution across both classes. For the informative features, the class mean vectors are $(-\delta, \ldots, -\delta)$ and $(\delta, \ldots, \delta)$, where the parameter $\delta > 0$ is adjusted to obtain a desired level of classification difficulty. The covariance matrices for both classes are block matrices

$$\Sigma_{d\times d} = \sigma^2 \times \begin{bmatrix} \Sigma_{l_1\times l_1} & & & 0 \\ & \Sigma_{l_2\times l_2} & & \\ & & \ddots & \\ 0 & & & I_{d_n\times d_n} \end{bmatrix},$$

where σ^2 is a variance parameter and Σ_{l_i} is an $l_i \times l_i$ matrix,

$$\Sigma_{l_i\times l_i} = \begin{bmatrix} 1 & \rho & \cdots & \rho \\ \rho & 1 & \cdots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \cdots & 1 \end{bmatrix},$$

representing l_i correlated features, with correlation coefficient $-1 < \rho < 1$, such that $\sum_i l_i = d - d_n$.

In the experiments below, we considered $d = 10$ features, consisting of $d_n = 4$ noisy and $d - d_n = 6$ informative features. The latter are correlated in pairs, i.e., $l_1 = l_2 = l_3 = 2$, with correlation coefficient $\rho = 0.2$. Four classification rules were considered: linear support vector machine (SVM), nonlinear SVM with radial basis kernel function (RBF-SVM), classification tree (CART) with stopped splitting at 5 points per leaf node, and k-nearest neighbors (k-NN), with $k = 3, 5, 7$. We adjusted δ to produce moderate classification difficulty over a range of sample sizes $n = 20, 40, 60, 80$, and 100; the corresponding average classification errors and 100; see Table 1.

The bias, variance, and RMS of each error estimator were estimated as sample-average approximations of (2), (3) and (5), respectively, by training the classifier 200 times using independently generated data sets and approximating the true classification error using a large test data set of size 5000. The bolstered resubstitution and bolstered posterior-probability error estimators for nonlinear classifiers used $M = 100$ Monte-Carlo points in their computation (the estimators are computed exactly for the linear SVM). The results are displayed in Tables 2, 3, and 4; boldface indicates the best error estimator for each

combination of classification rule and sample size. We can see in Table 2 that in almost all cases, the best bias value is obtained by one of the three bolstered error estimators. For a very small sample size $n = 20$, bootstrap is the best estimator in four out of the six classification rules; however, as will be seen later, bootstrap is much more computationally intensive than all other error estimators. Plain resubstitution is heavily negatively-biased at small sample sizes, as expected. All three bolstered estimators are able to significantly reduce this bias, except in the case of nearest neighbor classification, with small numbers of neighbors. What happens in this case is that the classification boundary is very complex, which affects negatively the bias-reducing properties of bolstering (incidentally, nearest-neighbor classification rules have infinite VC dimension, and Theorem 2 does not apply). As already indicated in Braga-Neto and Dougherty (2004a), this can be corrected by using the semi-bolstered resubstitution error estimator, which only applies bolstering kernels to correctly-classified points. Indeed, as can be seen in Table 2, semi-bolstered resubstitution produces the best bias values in nearly all cases in the 3NN and 5NN experiments. One can also see in the table that, as the number of neighbors increases from 3 to 7, the original bolstered error estimator becomes less biased; this occurs because the classification boundary becomes less rough. At $k = 7$ neighbors, the best results are obtained by the bolstered-3NNpp estimator, a member of the family of bolstered posterior probability resubstitution estimators proposed in this work. The plain 3NNpp estimator has good bias properties only in the linear SVM case (indeed, Lugosi and Pawlak (1994) warned that the resubstitution-like posterior-probability estimator was expected to be significantly biased). The cross-validation and bootstrap estimators are positively biased, which is expected since both estimators employ classifiers trained on data sets of smaller effective sample size than the original one (Braga-Neto and Dougherty, 2015b).

On the other hand, Table 3 shows that the bolstered estimators, but not semi-bolstering, achieve the smallest variance in nearly all cases. Indeed, semi-bolstering trades off less bias for more variance, since its bolstered empirical measure is more sparse than in full bolstering, as was also noted in Braga-Neto and Dougherty (2004a). Notice that both cross-validation and bootstrap display large variable at small sample sizes, which is expected, since they are resampling estimators; resubtitution-like estimators avoid resampling and should be less variable (Braga-Neto and Dougherty, 2004b).

Finally, in Table 4, we can see that the RMS (which combines bias and variance in a single metric) reveals a clear superiority of the bolstered estimators, and in particular the new bolstered-3NNpp estimator, over plain resubstitution, cross-validation, and bootstrap, except in the case of nearest-neighbor classification with a small number of neighbors, due to the aforementioned bias issue. In this case, semi-bolstering produces the best compromise between bias and variance. Nevertheless, with $k = 7$, we can see that the new bolstered-3NNpp estimator achieves the best RMS values, except at the very small sample size $n = 20$.

In order to examine the results further, Figures 3 and 4 display the box plots. These confirm the observations made previously about the bias and variance of the different error estimators. We can see, additionally, that at small sample size $n = 20$, all error error estimators, except for cross-validation, tend to be skewed towards optimistic biases, an effect that which disappears as sample size increases. Cross-validation is always skewed towards pessimistic bias, at all sample sizes.

In addition to the statistical issues discussed above, a very important issue is the computational complexity of the various error estimators, particularly in cases where thousands (or more) error estimates must be computed, as in wrapper feature selection. Table 5 displays

Classification Rule	Sample Size	resub	bolster	semi-bolster	3NNpp	bolster 3NNpp	cross valid	boot
Linear SVM	$n = 20$	-0.292	-0.076	-0.066	-0.070	-0.025	0.011	**0.010**
	$n = 40$	-0.169	-0.030	0.008	-0.039	**0.002**	0.021	0.027
	$n = 60$	-0.107	**-0.004**	0.043	-0.009	0.021	0.028	0.029
	$n = 80$	-0.079	**-0.001**	0.048	-0.004	0.021	0.029	0.025
	$n = 100$	-0.063	**0.008**	0.058	0.006	0.028	0.030	0.019
RBF SVM	$n = 20$	-0.277	-0.107	-0.100	-0.086	-0.066	0.070	**0.042**
	$n = 40$	-0.204	-0.085	-0.069	-0.035	**-0.018**	0.067	0.033
	$n = 60$	-0.168	-0.078	-0.058	-0.032	**-0.016**	0.072	0.036
	$n = 80$	-0.157	-0.056	-0.033	-0.015	**0.001**	0.053	0.023
	$n = 100$	-0.143	-0.059	-0.033	-0.021	**-0.006**	0.054	0.023
CART	$n = 20$	-0.354	-0.060	-0.054	-0.124	-0.026	0.017	**0.014**
	$n = 40$	-0.323	-0.040	-0.033	-0.119	**-0.013**	0.034	0.017
	$n = 60$	-0.314	-0.028	-0.020	-0.116	**-0.004**	0.031	0.013
	$n = 80$	-0.305	-0.022	-0.013	-0.111	**0.001**	0.036	0.014
	$n = 100$	-0.304	-0.016	-0.008	-0.108	**0.005**	0.030	0.011
3NN	$n = 20$	-0.175	-0.112	-0.034	-0.132	-0.124	0.031	**0.032**
	$n = 40$	-0.150	-0.105	**-0.028**	-0.122	-0.114	0.048	0.040
	$n = 60$	-0.144	-0.097	**-0.023**	-0.112	-0.105	0.046	0.037
	$n = 80$	-0.143	-0.096	**-0.021**	-0.113	-0.106	0.046	0.035
	$n = 100$	-0.136	-0.096	**-0.022**	-0.114	-0.107	0.049	0.036
5NN	$n = 20$	-0.124	-0.091	**0.023**	-0.086	-0.076	0.047	0.045
	$n = 40$	-0.110	-0.074	**0.018**	-0.071	-0.062	0.051	0.040
	$n = 60$	-0.099	-0.059	**0.023**	-0.061	-0.05	0.057	0.044
	$n = 80$	-0.095	-0.057	**0.023**	-0.060	-0.051	0.056	0.041
	$n = 100$	-0.091	-0.055	**0.022**	-0.059	-0.050	0.057	0.040
7NN	$n = 20$	-0.107	-0.073	**0.044**	-0.066	-0.056	0.055	0.045
	$n = 40$	-0.083	-0.047	0.046	-0.044	**-0.033**	0.062	0.044
	$n = 60$	-0.081	-0.042	0.043	-0.042	**-0.030**	0.059	0.038
	$n = 80$	-0.070	-0.032	0.048	-0.034	**-0.022**	0.064	0.043
	$n = 100$	-0.072	-0.034	0.044	-0.035	**-0.022**	0.062	0.039

Table 2: Bias results in the synthetic data experiment. The best bias value in each row is printed in bold.

the average computation time obtained by the error estimators in the experiment. The results confirm that plain resubstitution is lightning fast; its drawback is its large negative bias, as already mentioned. We can see that the plain posterior-probability error estimator, despite some bias issues, is also very fast. Combined with its small variance, this makes this estimator attractive for computationally-expensive classification tasks. Cross-validation (at 10 folds and no repetition) is the next fastest error estimator. Its poor variance properties under small sample sizes — and, in the case of wrapper feature selection, the issue of selection bias (Ambroise and McLachlan, 2002) — makes it unattractive. The bolstered resubstitution estimators are less fast but still much faster than the bootstrap. The latter is tens of times slower than the other estimators, in most cases, a fact that was already noted in (Braga-Neto and Dougherty, 2004b).

Classification Rule	Sample Size	resub	bolster	semi-bolster	3NNpp	bolster 3NNpp	cross valid	boot
Linear SVM	$n = 20$	0.0038	0.0035	0.0049	0.0056	**0.0033**	0.0168	0.0101
	$n = 40$	0.0053	**0.0016**	0.0033	0.0038	0.0018	0.0091	0.0059
	$n = 60$	0.0033	**0.0013**	0.0023	0.0027	**0.0013**	0.0053	0.0034
	$n = 80$	0.0018	**0.0010**	0.0016	0.0019	**0.0010**	0.0033	0.002
	$n = 100$	0.0017	**0.0008**	0.0013	0.0015	**0.0008**	0.0028	0.0017
RBF SVM	$n = 20$	**0.0043**	**0.0043**	0.0044	0.0092	0.0067	0.0262	0.0141
	$n = 40$	0.0017	**0.0013**	0.0016	0.0031	0.0018	0.0113	0.0055
	$n = 60$	0.0012	0.0008	0.0011	**0.002**	0.0011	0.007	0.0036
	$n = 80$	0.0008	0.0006	0.0009	0.0015	**0.0007**	0.0045	0.0023
	$n = 100$	0.0007	**0.0005**	0.0007	0.0011	0.0006	0.0035	0.0019
CART	$n = 20$	0.0028	**0.0030**	0.0032	0.0070	0.0036	0.0228	0.0082
	$n = 40$	0.0018	**0.0012**	0.0024	0.0032	0.0014	0.0108	0.0037
	$n = 60$	0.0012	**0.0008**	0.0009	0.0020	0.0009	0.0066	0.0022
	$n = 80$	0.0009	**0.0007**	0.008	0.0015	**0.0007**	0.0048	0.0017
	$n = 100$	0.0008	**0.0005**	0.006	0.0012	**0.0005**	0.0039	0.0014
3NN	$n = 20$	0.0064	**0.0022**	0.0059	0.0035	0.0027	0.0154	0.0081
	$n = 40$	0.0033	**0.0010**	0.0030	0.0015	**0.001**	0.0074	0.0036
	$n = 60$	0.0023	**0.0006**	0.0017	0.0009	**0.0006**	0.0051	0.0026
	$n = 80$	0.0017	0.0005	0.0013	0.0007	**0.0004**	0.0036	0.0017
	$n = 100$	0.0013	**0.0004**	0.0011	0.0005	0.0003	0.0030	0.0014
5NN	$n = 20$	0.0082	**0.0042**	0.0064	0.0068	0.0047	0.018	0.0098
	$n = 40$	0.0033	**0.0013**	0.0029	0.0022	**0.0013**	0.0075	0.0035
	$n = 60$	0.0023	0.0009	0.0020	0.0014	**0.0008**	0.0054	0.0026
	$n = 80$	0.0019	0.0007	0.0016	0.0011	**0.0006**	0.0039	0.0019
	$n = 100$	0.0014	0.0005	0.0013	0.0009	**0.0004**	0.0032	0.0015
7NN	$n = 20$	0.0091	**0.0052**	0.0066	0.0090	0.0064	0.0192	0.0113
	$n = 40$	0.0037	**0.0016**	0.0032	0.0030	**0.0016**	0.0076	0.0037
	$n = 60$	0.0025	**0.0010**	0.0021	0.0020	**0.0010**	0.0051	0.0026
	$n = 80$	0.0019	0.0007	0.0016	0.0012	**0.0006**	0.0039	0.0019
	$n = 100$	0.0014	**0.0005**	0.0012	0.0010	**0.0005**	0.0030	0.0015

Table 3: Variance results in the synthetic data experiment. The smallest variance in each row is printed in bold.

6.2 MNIST Data Experiment

In this section we present results of a simple experiment that indicate the potential of generalized resubstitution estimators in image classification with convolutional neural networks (CNN). The experiment uses the well-known MNIST data set and the LeNet-5 CNN architecture.

The MNIST training data set contains $60,000$ 28×28 grayscale images of handwritten digits between 0 and 9 (hence, 10 classes). It is well-known that LeNet-5 can achieve accuracies of upwards of 99% on this data set; e.g., see Tabik et al. (2017). Problems with small classification error tend to be easier in terms of error estimation performance (Braga-Neto and Dougherty, 2015b). To make the problem more challenging, we train the LeNet-5 classifier on random subsets of $n = 200, 400, 600$ and 800 images from the original data set. The remaining data are used to obtain accurate test-set estimates of the true classification error, in order to compute estimates of the bias, variance, and RMS of each

Classification Rule	Sample Size	resub	bolster	semi-bolster	3NNpp	bolster 3NNpp	cross valid	boot
Linear SVM	$n = 20$	0.2981	0.0963	0.0963	0.1029	**0.0626**	0.1302	0.1010
	$n = 40$	0.1836	0.0506	0.0579	0.073	**0.0422**	0.0976	0.0811
	$n = 60$	0.1213	**0.0362**	0.0650	0.0532	0.0421	0.0783	0.0651
	$n = 80$	0.0896	**0.0309**	0.0623	0.0440	0.0378	0.0641	0.0508
	$n = 100$	0.0751	**0.0289**	0.0680	0.0396	0.0391	0.0611	0.0453
RBF SVM	$n = 20$	0.2844	0.1258	0.1196	0.1289	**0.1050**	0.1765	0.1259
	$n = 40$	0.2086	0.0921	0.0798	0.0652	**0.0455**	0.1255	0.0816
	$n = 60$	0.1718	0.0831	0.0666	0.0545	**0.0364**	0.1101	0.0702
	$n = 80$	0.1596	0.0614	0.0441	0.0412	**0.0271**	0.0852	0.0535
	$n = 100$	0.1455	0.0632	0.0432	0.0388	**0.0242**	0.0806	0.0495
CART	$n = 20$	0.3581	0.0812	0.0785	0.1494	**0.0650**	0.1520	0.0914
	$n = 40$	0.326	0.053	0.0492	0.1314	**0.0401**	0.1093	0.0631
	$n = 60$	0.3164	0.0401	0.0363	0.1242	**0.0306**	0.0867	0.0489
	$n = 80$	0.3067	0.0334	0.0301	0.1175	**0.0265**	0.0783	0.0431
	$n = 100$	0.3049	0.0277	0.0251	0.1134	**0.0239**	0.0692	0.039
3NN	$n = 20$	0.1924	0.1227	**0.0869**	0.1450	0.1337	0.1239	0.0955
	$n = 40$	0.1616	0.1092	**0.0573**	0.1284	0.1179	0.1020	0.0721
	$n = 60$	0.1524	0.1015	**0.0461**	0.1159	0.1069	0.0838	0.0622
	$n = 80$	0.1485	0.0981	**0.0452**	0.1169	0.1079	0.0756	0.0532
	$n = 100$	0.1418	0.0981	**0.0378**	0.1157	0.1089	0.0775	0.0538
5NN	$n = 20$	0.1532	0.1090	**0.0832**	0.1175	0.1033	0.1382	0.1097
	$n = 40$	0.1253	0.0841	**0.0531**	0.0868	0.0738	0.1034	0.0721
	$n = 60$	0.1109	0.0662	**0.0550**	0.0729	0.0583	0.0903	0.0666
	$n = 80$	0.1031	0.0644	**0.0461**	0.0671	0.0548	0.0821	0.0573
	$n = 100$	0.0994	0.0585	**0.0477**	0.0662	0.0539	0.0828	0.0566
7NN	$n = 20$	0.1465	0.1011	**0.0913**	0.1116	0.0977	0.1504	0.1188
	$n = 40$	0.1024	0.0617	0.0756	0.0666	**0.0519**	0.1093	0.0744
	$n = 60$	0.0952	0.0516	0.0659	0.0580	**0.0424**	0.0915	0.0628
	$n = 80$	0.0806	0.0439	0.0625	0.0525	**0.0297**	0.0877	0.0587
	$n = 100$	0.0824	0.0394	0.0595	0.0461	**0.0312**	0.0796	0.0559

Table 4: RMS results in the synthetic data experiment. The best RMS value in each row is printed in bold.

error estimator, using 200 independently drawn training data sets for each sample size. The LeNet-5 network was trained using 200 epochs of stochastic gradient descent, with batch size 32, employing 10% of the training data in each case as a validation data set to stop training early if the validation loss was not reduced for 10 consecutive epochs.

We investigate the performance of bolstered resubstitution with diagonal Gaussian kernels, which leads to a "Naive-Bayes" bolstering resubstitution estimator, as explained in Section 4.1. This is done since spherical kernels tend to perform poorly in very high-dimensional spaces (Sima et al., 2014). Likewise, k-nearest neighbor posterior-probability estimators led to too much bias in this high-dimensional space, and are not considered further.

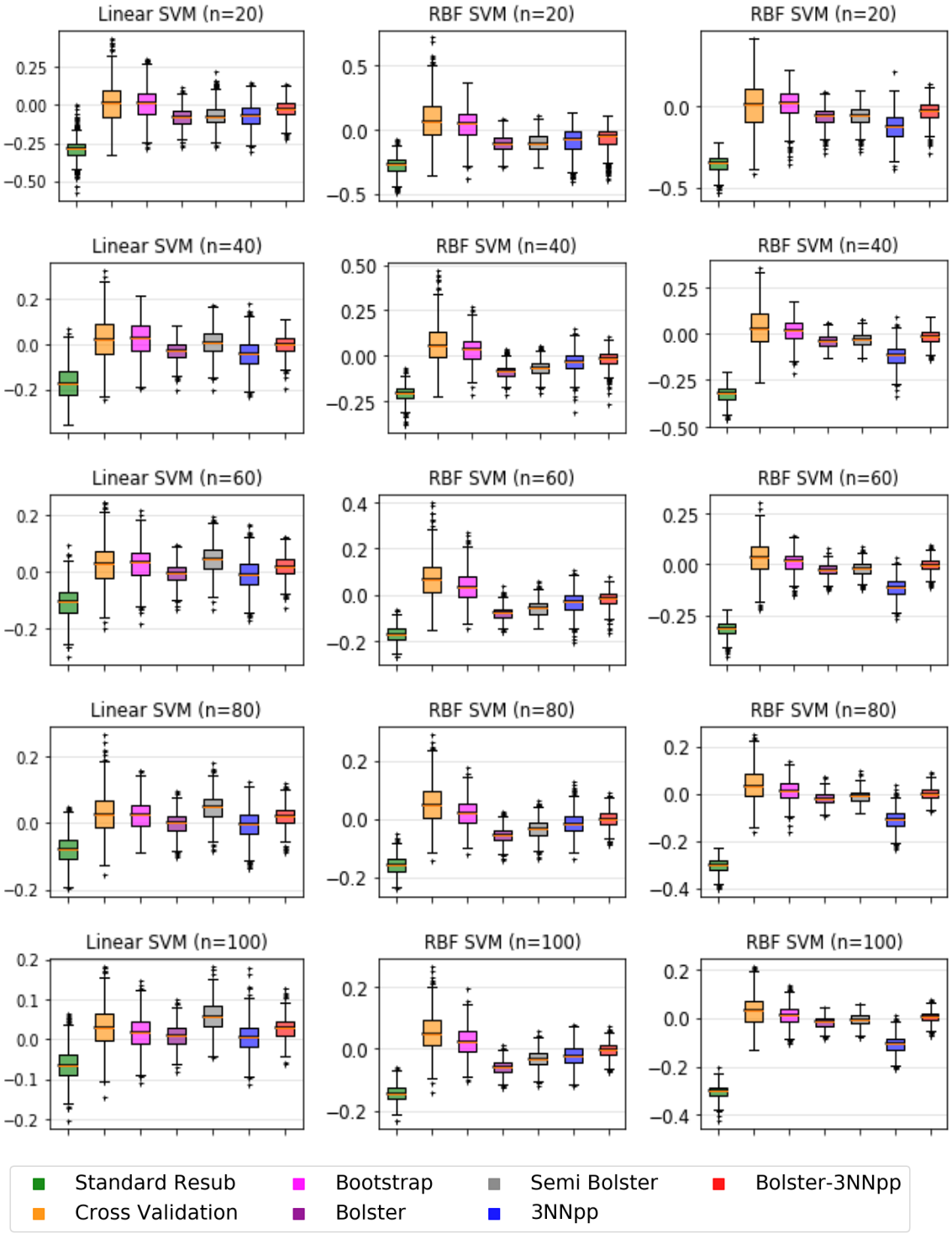


Figure 3: Boxplots for SVM and CART classification rules in the synthetic data experiment.

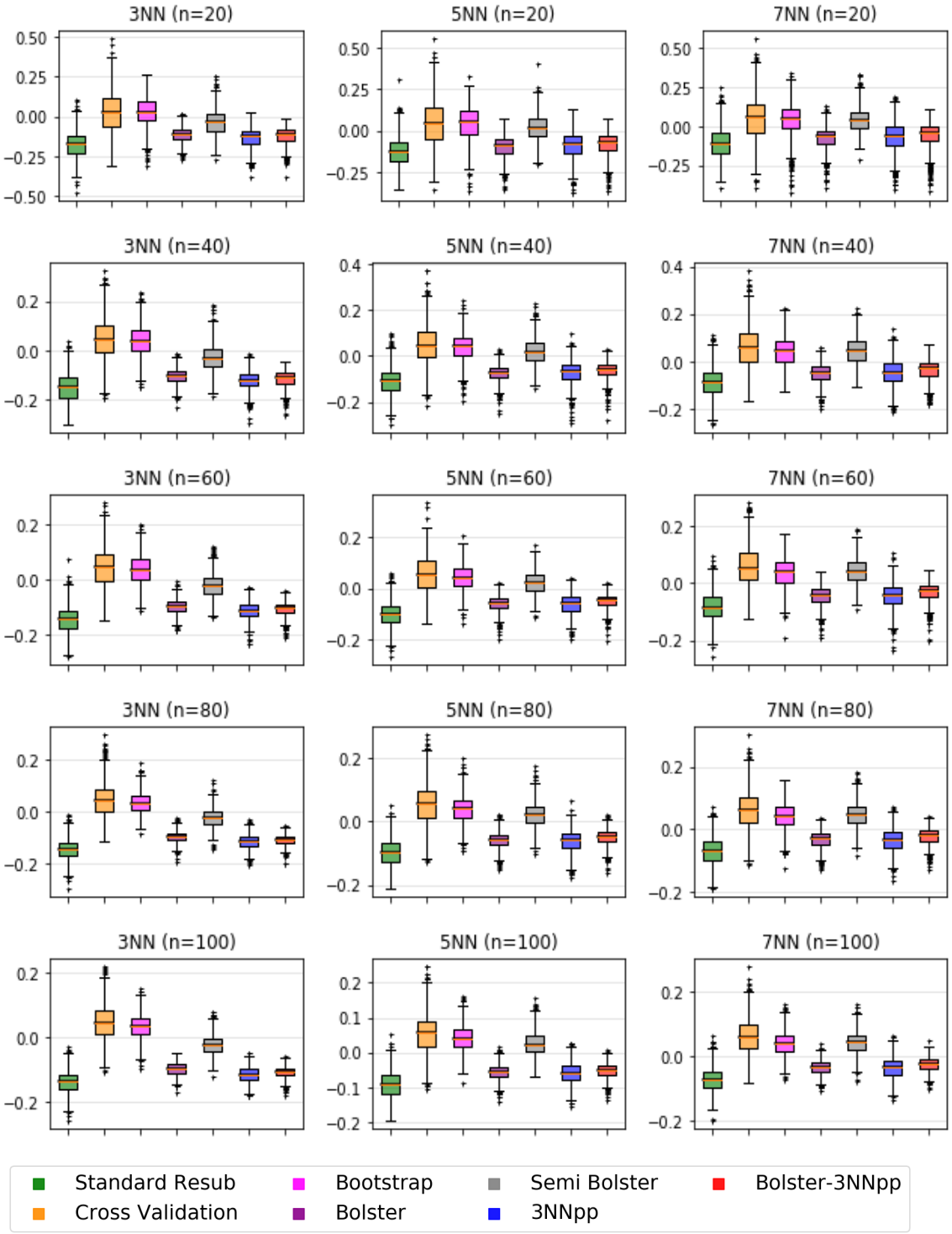


Figure 4: Boxplots for nearest-neighbor classification rules in the synthetic data experiment.

Classification Rule	Sample Size	resub	bolster	semi-bolster	3NNpp	bolster 3NNpp	cross valid	boot
Linear SVM	$n = 20$	0.00012	7.72	7.17	2.18	6.94	3.98	111.75
	$n = 40$	0.00013	13.23	13.01	1.84	13.28	5.17	111.83
	$n = 60$	0.00013	19.95	19.27	1.73	18.84	6.23	136.64
	$n = 80$	0.00013	28.06	26.59	1.76	28.15	9.41	194.78
	$n = 100$	0.00014	35.67	33.98	1.95	36.04	14.51	259.07
RBF SVM	$n = 20$	0.00021	9.37	7.81	1.98	9.34	5.76	141.81
	$n = 40$	0.00029	17.53	15.23	1.74	17.96	7.11	166.50
	$n = 60$	0.00040	27.99	28.09	1.65	30.94	9.90	222.74
	$n = 80$	0.00053	37.45	36.41	1.67	45.82	12.60	272.67
	$n = 100$	0.00070	53.53	50.92	1.87	68.38	18.73	367.78
CART	$n = 20$	0.00014	5.99	4.26	1.87	7.33	3.15	91.54
	$n = 40$	0.00014	12.15	11.37	1.97	16.45	4.79	99.37
	$n = 60$	0.00015	18.06	15.79	1.96	22.57	4.89	112.91
	$n = 80$	0.00015	24.08	21.96	1.51	27.54	4.94	113.20
	$n = 100$	0.00016	34.82	31.87	1.98	36.51	6.00	133.62
3NN	$n = 20$	0.0012	63.17	55.19	3.30	121.20	10.54	263.76
	$n = 40$	0.0017	125.55	120.16	3.91	239.93	11.28	300.8
	$n = 60$	0.0028	188.34	185.14	4.50	360.82	11.96	346.19
	$n = 80$	0.0029	254.18	252.59	5.21	397.00	12.82	390.31
	$n = 100$	0.0034	319.55	319.53	5.90	445.04	13.56	517.62
5NN	$n = 20$	0.0011	61.95	51.02	3.24	118.45	10.4	261.14
	$n = 40$	0.0016	123.18	113.18	3.84	235.02	11.12	297.05
	$n = 60$	0.0021	185.08	176.05	4.43	253.90	11.79	346.69
	$n = 80$	0.0027	249.10	238.91	5.11	391.04	12.61	417.84
	$n = 100$	0.0034	310.63	300.12	5.76	436.26	13.35	499.96
7NN	$n = 20$	0.0011	53.07	41.08	2.76	101.20	9.01	230.98
	$n = 40$	0.0016	116.56	104.69	3.67	221.75	10.65	287.32
	$n = 60$	0.0022	175.79	163.67	4.23	329.95	11.23	335.52
	$n = 80$	0.0027	237.74	224.36	4.89	377.69	12.03	416.78
	$n = 100$	0.0034	299.76	286.09	5.53	422.69	12.72	477.60

Table 5: Average computation time (in milliseconds) in the synthetic data experiment.

If X_{ijk} denotes pixel k in image i of class j, the mean minimum distance $d_{n,jk}$ among pixels k in class j is:

$$\hat{d}_{n,jk} \;=\; \frac{1}{n_j} \sum_{i=1}^{n_j} |X_{ijk} - X'_{ijk}|\,, \quad j = 0, 1, \ldots, c-1\,,\; k = 1, \ldots, 28 \times 28\,,$$

where n_j is the number of images in class j ($n_j \geq 2$ is assumed) and X'_{ijk} is the nearest pixel (in value) in position k to X_{ijk} among images in class j. The bolstering kernel standard deviations are then given by:

$$\sigma_{n,jk} \;=\; \frac{\hat{d}_{n,jk}}{\alpha_1}\,, \quad j = 0, 1, \ldots, c-1\,,\; k = 1, \ldots, 28 \times 28\,, \tag{27}$$

where $\alpha_1 = 0.674$, as seen in Section 4.1.

In order to further reduce the bias of the Naive-Bayes bolstered resubstitution estimator in this high-dimensional space, we employ a data-driven calibration procedure: we multiply

Sample Size	resub	nBbolster	calibrated nBbolster (r=1)	calibrated nBbolster (r=5)
$n = 200$	-0.131	-0.082	0.019	**0.014**
$n = 400$	-0.100	-0.064	**0.007**	0.012
$n = 600$	-0.080	-0.052	**0.005**	**0.005**
$n = 800$	-0.072	-0.051	0.003	**0.001**

Table 6: Bias results in the MNIST data experiment. The best bias value in each row is printed in bold.

the kernel standard deviation by a constant $\kappa > 0$, which is adjusted so as to minimize the estimated bias of the estimator. The bias is roughly estimated by training the classifier on a random sample of 80% of the images form the available training data, and testing it on remaining 20%. Depending on the computational cost of training the classifiers, this process can be repeated a number of times r and the results averaged. The point is that the bias does not need to be accurately estimated in order to find a useful value for κ. The calibration process consists of starting at $\kappa = 1$, computing the corrected Naive-Bayes bolstered resubstitution estimate, and increasing κ by a fixed step-size (here, 0.1) until the magnitude of the roughly estimated bias does not decrease for two consecutive iterations. A similar model-based calibration method was proposed in (Sima et al., 2014); however, the procedure proposed here is entirely data-driven and makes no modeling assumptions.

Finally, the naive-Bayes bolstered resubstitution estimator is computed by Monte-Carlo, as in (16), where $\{\mathbf{X}_{ij}^{\mathrm{MC}}; j = 1, \ldots, M\}$ are random images generated by drawing each pixel from a Gaussian distribution with mean equal to the original pixel value and standard deviation in (27). This generates "noisy images" where the intensity of the noise in each pixel is correlated with the variability of pixel values at that position across the training data (for that digit class). Here we employed $M = 100$ Monte-Carlo images for each training image. A few of these Monte-Carlo images can be seen in Figure 5.

The results of the experiment are displayed in Tables 6, 7 and 8. We can see that, while Naive-Bayes bolstered resubstitution is able to improve somewhat the optimistic bias of resubstitution, calibration succeeded into reducing the bias to nearly zero, especially as sample size increases. The generalized resubstitution estimators were not able to match the low variance of plain resubstitution. Calibration increased the variance of the plain Naive-Bayes estimator, as might be expected (though much less in the case $r = 5$ than in the case $r = 1$). When bias and variance are combined in the RMS metric, the clear winners are the calibrated estimators, particularly at $r = 5$. Even at $r = 1$, (i.e., just one additional step of classifier training), there is a substantial improvement. If more than $r = 5$ repetitions are used, it is expected that results will improve further, though at a higher computational cost. Notice that the bias, variance, and RMS of all estimators decrease monotonically with increasing sample size.

Finally, Table 9 displays the average computation time for the error estimators in the experiment. The results again confirm that plain resubstitution is very fast. The superior performance of the calibrated naive-Bayes bolstered resubstitution estimators come at a computational price; this is due to the fact that additional classifiers need to be trained in order to perform the calibration procedure.

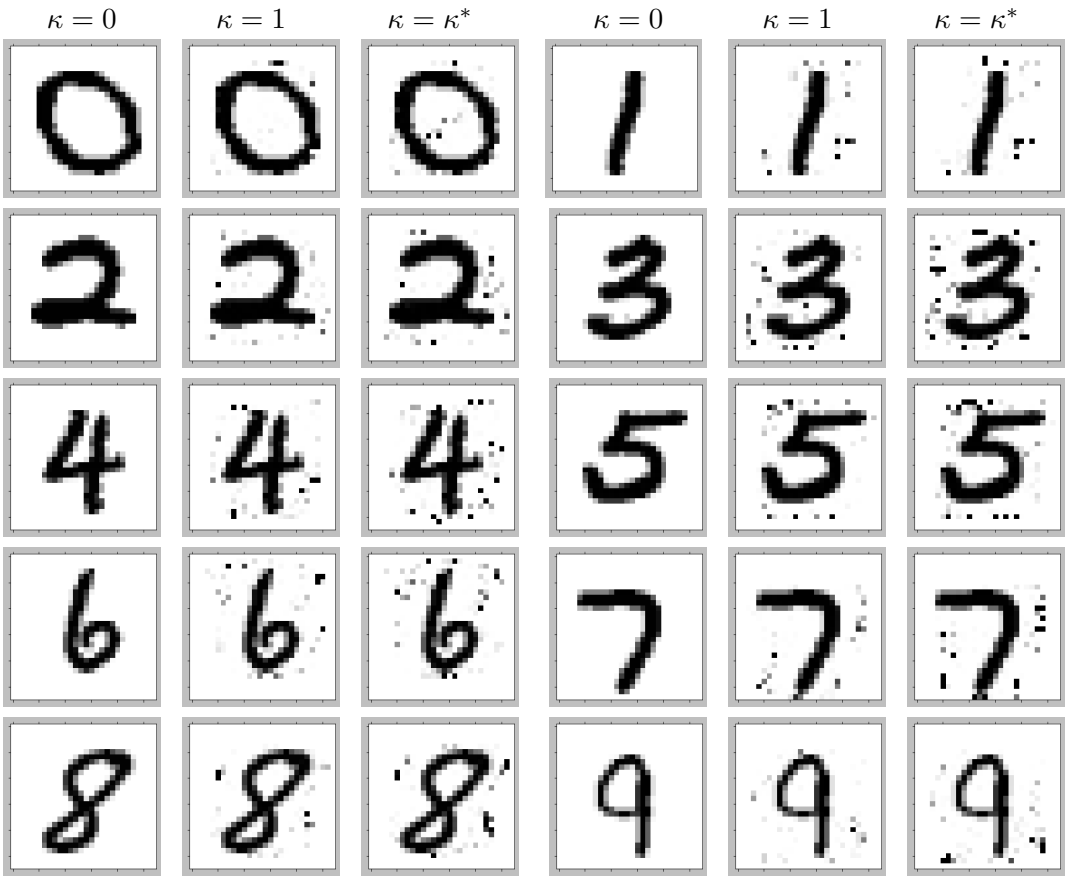


Figure 5: A few Monte-Carlo images used in the Naive-Bayes bolstered resubstitution error estimator, with $n = 600$. The parameter κ^* is the optimal correction factor for the calibrated naive Bayes bolstered error estimator (r=1). The cases $\kappa = 0$ and $\kappa = 1$ refer to the original image and the uncorrected Naive-Bayes bolstered image, respectively.

Sample Size	resub	nBbolster	calibrated nBbolster (r=1)	calibrated nBbolster (r=5)
$n = 200$	**0.0006**	0.0013	0.0064	0.0027
$n = 400$	**0.0002**	0.0008	0.0029	0.0013
$n = 600$	**0.0002**	0.0006	0.0018	0.0006
$n = 800$	**0.0001**	0.0003	0.0011	0.0006

Table 7: Variance results in the MNIST data experiment. The best variance value in each row is printed in bold.

Sample Size	resub	nBbolster	calibrated nBbolster (r=1)	calibrated nBbolster (r=5)
$n = 200$	0.1336	0.0897	0.0822	**0.0519**
$n = 400$	0.1010	0.0702	0.0505	**0.0418**
$n = 600$	0.0812	0.0576	0.0403	**0.3040**
$n = 800$	0.0728	0.0539	0.0301	**0.0200**

Table 8: RMS results in the MNIST data experiment. The best RMS value in each row is printed in bold.

Sample Size	resub	nBbolster	calibrated nBbolster (r=1)	alibrated nBbolster (r=5)
$n = 200$	0.00043	7.24	65.91	298.51
$n = 400$	0.00085	15.42	127.78	597.24
$n = 600$	0.00131	25.54	216.99	928.61
$n = 800$	0.00175	35.59	296.86	1342.19

Table 9: Average computation time (in seconds) in the MNIST data experiment.

7. Conclusions

We proposed in this paper a completely general and unifying framework to study resubstitution-like classification error estimators in terms of arbitrary empirical probability measures. This is a broad family of classification error estimators who can all be computed in the same way by using different empirical probability measures. They do not require resampling and retraining of classifiers (though in the MNIST classification example a data-driven calibration procedure to further reduce bias was used, which may employ resampling, though it is not necessary). We showed that various existing error estimators are special cases of generalized resubstitution estimators, and proposed bolstered posterior probability error estimators as a novel example in this class. Generalized resubstitution estimators are generally fast and thus can be used in settings where computational complexity issue is an issue, such as in wrapper feature selection for large data sets. In the two-class case, we showed that these estimators have good large-sample properties, provided that the classification rule has a a finite VC dimension and the corresponding empirical probability measure converges to the standard empirical probability measure, in a precise sense. The extension of these results to more than two classes is left to future research — the main obstacle being the absence of a multi-class version of the VC Theorem. In addition, we showed empirically, by means of numerical experiments, that generalized resubstitution error estimators also display excellent small-sample performance. In particular, we showed that bolstered-3NNpp,

an example of error estimator in the new family proposed here, is the error estimator of choice in traditional classification, except in the case of very complex, overfitting classifiers, such as those produced by nearest-neighbor rules with a small number of neighbors, in which semi-bolstered resubstitution should be used instead. For image classification by deep convolutional neural networks, we showed empirically that naive-Bayes bolstering with a simple data-driven calibration procedure produces excellent results. This indicates the potential of this approach in the area of computer vision, a topic that will be further explored in future work.

Acknowledgements

We would like to thank the anonymous reviewers of this work, whose detailed comments helped improve its clarity and presentation.

References

C. Ambroise and G. McLachlan. Selection bias in gene extraction on the basis of microarray gene expression data. *Proc. Natl. Acad. Sci.*, 99(10):6562–6566, 2002.

P.L. Bartlett, S. Boucheron, and G. Lugosi. Model selection and error estimation. *Machine Learning*, 48:85–113, 2002.

Ulisses Braga-Neto and Edward Dougherty. Bolstered error estimation. *Pattern Recognition*, 37(6):1267–1281, 2004a.

Ulisses M Braga-Neto. *Fundamentals of Pattern Recognition and Machine Learning*. Springer, 2020.

Ulisses M Braga-Neto and Edward R Dougherty. *Error estimation for pattern recognition*. John Wiley & Sons, 2015a.

U.M. Braga-Neto and E.R. Dougherty. Is cross-validation valid for microarray classification? *Bioinformatics*, 20(3):374–380, 2004b.

U.M. Braga-Neto and E.R. Dougherty. Bolstered error estimation. *Pattern Recognition*, 37 (6):1267–1281, 2004c.

U.M. Braga-Neto and E.R. Dougherty. *Error Estimation for Pattern Recognition*. Wiley, New York, 2015b.

T. Cover. Learning in pattern recognition. In S. Watanabe, editor, *Methodologies of Pattern Recognition*, pages 111–132. Academic Press, New York, NY, 1969.

T. Cover and P. Hart. Nearest-neighbor pattern classification. *IEEE Trans. on Information Theory*, 13:21–27, 1967.

L. Dalton and E.R. Dougherty. Bayesian minimum mean-square error estimation for classification error – part I: Definition and the bayesian mmse error estimator for discrete classification. *IEEE Transactions on Signal Processing*, 59(1):115–129, 2011a.

L. Dalton and E.R. Dougherty. Bayesian minimum mean-square error estimation for classification error – part II: Linear classification of gaussian models. *IEEE Transactions on Signal Processing*, 59(1):130–144, 2011b.

L. Dalton and E.R. Dougherty. Application of the bayesian mmse error estimator for classification error to gene-expression microarray data. *IEEE Transactions on Signal Processing*, 27(13):1822–1831, 2011c.

L. Dalton and E.R. Dougherty. Exact mse performance of the bayesian mmse estimator for classification error – part i: Representation. *IEEE Transactions on Signal Processing*, 60 (5):2575–2587, 2012a.

L. Dalton and E.R. Dougherty. Exact mse performance of the bayesian mmse estimator for classification error – part ii: Performance analysis and applications. *IEEE Transactions on Signal Processing*, 60(5):2588–2603, 2012b.

Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*. Springer, 1996.

B. Efron. Bootstrap methods: Another look at the jacknife. *Annals of Statistics*, 7:1–26, 1979.

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.

B. Efron and R. Tibshirani. Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560, 1997.

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.

D.J. Hand. Recent advances in error rate estimation. *Pattern Recognition Letters*, 4:335–346, 1986.

Ahmed Hefny and Amir F Atiya. A new monte carlo-based error rate estimator. In *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pages 37–47. Springer, 2010.

X. Jiang and U.M. Braga-Neto. A naive-bayes approach to bolstered error estimation in high-dimensional spaces, 2014. Proceedings of the IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS'2014), Atlanta, GA.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019.

M. Kaariainen. Generalization error bounds using unlabeled data. In *Proceedings of COLT'05*, 2005.

M. Kaariainen and J. Langford. A comparison of tight generalization bounds. In *Proceedings of the 22nd International Conference on Machine Learning*. Bonn, Germany, 2005.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

P.A. Lachenbruch and M.R. Mickey. Estimation of error rates in discriminant analysis. *Technometrics*, 10:1–11, 1968.

G. Lugosi and M. Pawlak. On the posterior-probability estimate of the error rate of nonparametric classification rules. *IEEE Transactions on Information Theory*, 40(2): 475–481, 1994.

G. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, New York, 1992.

G.J. McLachlan. Error rate estimation in discriminant analysis: recent advances. In A.K. Gupta, editor, *Advances in Multivariate Analysis*. D. Reidel, Dordrecht, 1987.

D. Pollard. *Convergence of Stochastic Processes*. Springer, New York, 1984.

Carl Edward Rasmussen and Christopher KI Williams. Gaussian processes for machine learning. isbn 026218253x, 2006.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *arXiv preprint arXiv:1902.10811*, 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

R.A. Schiavo and D.J. Hand. Ten more years of error rate research. *International Statistical Review*, 68(3):295–310, 2000.

C. Sima, S. Attoor, U.M. Braga-Neto, J. Lowey, E. Suh, and E.R. Dougherty. Impact of error estimation on feature-selection algorithms. *Pattern Recognition*, 38(12):2472–2482, 2005a.

C. Sima, U.M. Braga-Neto, and E.R. Dougherty. Bolstered error estimation provides superior feature-set ranking for small samples. *Bioinformatics*, 21(7):1046–1054, 2005b.

C. Sima, T. Vu, U.M. Braga-Neto, and E.R. Dougherty. High-dimensional bolstered error estimation. *Bioinformatics*, 27(21):3056–3064, 2014.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

C.A.B. Smith. Some examples of discrimination. *Annals of Eugenics*, 18:272–282, 1947.

M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36:111–147, 1974.

Siham Tabik, Daniel Peralta, Andres Herrera-Poyatos, and Francisco Herrera. A snapshot of image pre-processing for convolutional neural networks: case study of mnist. *International Journal of Computational Intelligence Systems*, 10(1):555–568, 2017.

G.T. Toussaint. Bibliography on estimation of misclassification. *IEEE Transactions on Information Theory*, IT-20(4):472–479, 1974.

G.T. Toussaint and R. Donaldson. Algorithms for recognizing contour-traced hand-printed characters. *IEEE Transactions on Computers*, 19:541–546, 1970.

V.N. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Probability and its Applications*, 16:264–280, 1971.

Y. Xiao, J. Hua, and E.R. Dougherthy. Quantification of the impact of feature selection on cross-validation error estimation precision. *EURASIP J. Bioinformatics and Systems Biology*, 2007.

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.

Mohammadmahdi R Yousefi, Jianping Hua, and Edward R Dougherty. Multiple-rule bias in the comparison of classification rules. *Bioinformatics*, 27(12):1675–1683, 2011.

X. Zhou and K.Z Mao. The ties problem resulting from counting-based error estimators and its impact on gene selection algorithms. *Bioinformatics*, 22:2507–2515, 2006.

A. Zollanvari, U.M. Braga-Neto, and E.R. Dougherty. Analytic study of performance of error estimators for linear discriminant analysis. *IEEE Transactions on Signal Processing*, 59(9): 1–18, 2011.

A. Zollanvari, U.M. Braga-Neto, and E.R. Dougherty. Exact representation of the second-order moments for resubstitution and leave-one-out error estimation for linear discriminant analysis in the univariate heteroskedastic gaussian model. *Pattern Recognition*, 45(2): 908–917, 2012.