Machine Learning Assessment: Implications to Cybersecurity

Waleed A. Yousef

Abstract After discussing the construction of machine learning (ML) algorithms in the previous chapter, this chapter is dedicated to their assessment and performance estimation (with an emphasis on classification assessment), a topic that is equally important specially in the context of cyberphysical security design. The literature is full of nonparametric methods to estimate a statistic from just one available dataset through resampling techniques, e.g., jackknife, bootstrap and cross validation (CV). Special statistics of great interest are the error rate and the area under the ROC curve (AUC) of a classification rule. The importance of these resampling methods stems from the fact that they require no knowledge about the probability distribution of the data or the construction details of the ML algorithm. This chapter provides a concise review of this literature to establish a coherent theoretical framework for these methods that can estimate both the error rate (a one-sample statistic) and the AUC (a two-sample statistic). The resampling methods are usually computationally expensive, because they rely on repeating the training and testing of a ML algorithm after each resampling iteration. Therefore, the practical applicability of some of these methods may be limited to the traditional ML algorithms rather than the very computationally demanding approaches of the recent deep neural networks (DNN). In the field of cyberphysical security, many applications generate structured (tabular) data, which can be fed to all traditional ML approaches. This is in contrast to the DNN approaches, which favor unstructured data, e.g., images, text, voice, etc.; hence, the relevance of this chapter to this field.

Keywords Assessment · Performance estimation · Resampling techniques · ROC curve · Area under the ROC curve · Classification · Machine learning · Deep neural network · Unstructured data · Sample · Bootstrap · Nonparametric · Estimators

W. A. Yousef (✉)
CS Department, HCILAB, Faculty of Computers and Artificial Intelligence, Helwan University, Helwan, Egypt
e-mail: wyousef@fci.helwan.edu.eg

I. Traore et al. (eds.), *Artificial Intelligence for Cyber-Physical Systems Hardening*,
Engineering Cyber-Physical Systems and Critical Infrastructures 2,
https://doi.org/10.1007/978-3-031-16237-4_3

1 Introduction

1.1 Motivation

Consider a ML problem, where some models have been trained on a given dataset. It is then required to know their performances, in terms of any performance measure, on the population of testers. This is not only for the sake of assessing each of them, but also to be able to select the best model among them. These different models could even represent different instances of the same ML algorithm, with different values of parameters (e.g., a KNN with different values of K), and it is required to choose the best value for the current problem. The performance on the population of testers is called the true performance, because this is the performance on the whole population, not on a subset of it.

If the underlying probability distribution of the testers is known, e.g., from a priori knowledge about the nature of the problem, the true performance can be calculated mathematically. One of the first attempts in this direction was Fukunaga [14], where he assumed the data follows a multinormal distribution, to find a closed-form expression of the error rate of a binary classification rule. An alternative to mathematical calculations is simulating a very large dataset, from the assumed distribution, from which a very accurate estimation of the true performance can be obtained.

The early work of Fukunuga was inspiring, from the theoretical point of view, for the early community of pattern recognition and machine learning to understand important theoretical properties and concepts. However, for real-life applications it is very unusual that the assumption of multinormality, or any other assumption, hold. In these situations, which are called nonparametric, or distribution-free, it is impossible to derive either the true performance in closed form, or estimate it using a very large simulated dataset. In such situations, the true performance must be estimated from a single testing dataset (testers). The way we obtain such a testing dataset defines two major paradigms, discussed next.

In Paradigm I, we only have one dataset $\mathbf{t}$, usually called the design or construction dataset, from which we have to make up a training dataset $\mathbf{tr}$ and a testing dataset $\mathbf{ts}$, such that $\mathbf{t} = \mathbf{tr} \cup \mathbf{ts}$. Otherwise, training and testing on the same dataset $\mathbf{t}$ would provide a very optimistic estimate of the performance measure. This splitting is performed iteratively using one of the resampling techniques, e.g., jackknife, bootstrap, or cross validation. In each resampling iteration we get a different pair of training and testing datasets, on which the algorithm will be trained and tested, respectively. The results from these different iterations will be compiled together, as defined by the resampling method, to provide a single estimate of the performance measure. It is obvious that the performance estimation obtained from any of these methods will vary with varying the design dataset $\mathbf{t}$. This chapter is dedicated to reviewing this paradigm, its different estimators, and the variance estimation of these estimators.

It is worth mentioning that fatal fallacies are committed by practitioners when using this paradigm. For example, a very common mistake is using the whole dataset $\mathbf{t}$ to learn some statistical properties of the different classes of the classification problem, mistakenly naming this a data preprocessing step, using these properties to construct a classifier, then excluding this step from the resampling mechanism afterwards. Although the correct way of performing preprocessing is explained in textbooks (see, e.g.,Hastie et al. [20], Sect. 7.10.2), we still see this mistake in several occasions in both academia and industry.

In Paradigm II, it is required, or even mandated (e.g., in several public-policy-making or regulatory settings), to maintain what might be called the traditional data hygiene of two independent datasets: the design dataset **t**, and a final testing dataset **TS**, which is a sequestered testing dataset that has never been available to the design procedure, but for just onetime final testing. Assessing a ML algorithm from independent testing dataset is as simple as applying the estimators of the performance measure of interest (Sect. 1.2) on the testing dataset. However, the estimator will then have two sources of variability, the design and the testing datasets. The mathematical details of this paradigm and the estimation of this variance are discussed in Yousef et al. [34], Chen et al. [4], and not reviewed in our present chapter.

Although it may seem very safe to use this testing paradigm, some practitioners abuse it as well. One possible common mistake is that they test several models on this sequestered testing set, then they analyze the relative estimated performances. Accordingly, these models are redesigned to improve their performance on the testing set! Worse than this is keeping iterating this processes several times, which indeed turns the independent sequestered testing dataset to be part of the training dataset, indirectly through this human mental parsing of the results, which acts as a feedback that guides the redesign process.

Nowadays, it is almost the default in the field of ML to leverage both paradigms in the task of model assessment and selection. The available dataset is initially split into two datasets:

1. the design dataset **t**, from which the ML algorithm is designed. This is conducted via one of the resampling methods of paradigm I explained above. Usually, several algorithms are used, and several parameters' values are examined for each algorithm. Then, the model with the best performance is chosen.
2. the sequestered testing dataset **TS**, on which the final chosen model from paradigm I is assessed once and only once, without redesign. This is the final estimation of the performance measure that should be reported, along with the estimation of its variance.

It is worth mentioning that, there is a convention in the field to call the dataset **ts** that is split from the design dataset **t** during the resampling process, a validation dataset rather than a testing dataset, to reserve the word testing to the final testing datset **TS** of paradigm II. However, in some applications, the converse is adopted; i.e., **ts** is called the a testing dataset and **TS** is called a validation dataset. To avoid ambiguity, any notation and expression should be defined clearly within any context.

What is introduced above is valid for any ML problem, whether it is regression or classification, and for any performance measure, whether it is the error rate Err, AUC, or any other. However, we emphasize below two very important issues.

(1) The true performance, which we discussed its estimation in this introduction so far, is itself a random variable whose randomness arises from the randomness of

the training dataset, as was explained in the previous chapter. Have we changed the training dataset, the true performance would change. For example, and without loss of generality (WLOG) but for the sake of illustration, suppose the whole design dataset **t** is used as a training dataset **tr** and we are interested in the AUC as a performance measure. Then, as was explained in the previous chapter, we should be interested in the following:

1. $\mathrm{AUC}_{\mathbf{t}}$: the true performance conditional on a particular training dataset **t** of a specified size n.
2. $\mathrm{E}_{\mathbf{t}}\mathrm{AUC}_{\mathbf{t}}$: the expectation of true performance over the population of training datasets of the same size n.
3. $\mathrm{Var}_{\mathbf{t}}\mathrm{AUC}_{\mathbf{t}}$: the variance of the true performance over the population of training datasets of the same size n.

(2) Regarding the meaning and utility of the performance measure, we emphasize the importance of the ROC curve and its AUC as a summary measure [2, 18, 19], where the former is a manifestation of the trade-off between the two types of error of any binary classification rule. We always advocate for the use of the ROC or its AUC since they are prevalence independent; i.e., they do not depend on a particular chosen threshold, class prior probability, or misclassification costs. Adopting a performance measure that is prevalence dependent, e.g., the overall accuracy or its many different versions, can provide a misleading measure of the classification power of the classification algorithm, especially in classification problems that involve, for instance, unbalanced data (different class size). Therefore, the present chapter assumes familiarity with the ROC and its AUC, at the level provided in the previous chapter. However, for the sake of completeness, all notations are tersely summarized in the following subsection.

1.2 Notation

Consider the binary classification problem, where a classification rule η gives a score of $h(x)$ for the predictor x, and classifies it to one of the two classes by comparing this score $h(x)$ to a chosen threshold th. The observation x belongs to one of the two classes with distributions F_i, $i = 1, 2$. The two error components of this rule (e_1, or the false negative fraction (FNF), and e_2 or the false positive fraction (FPF)), along with the risk, are given as follows:

$$\mathrm{FNF} = e_1 = \int_{-\infty}^{th} f_h\left(h(x)|\omega_1\right) dh(x), \tag{1a}$$

$$\mathrm{FPF} = e_2 = \int_{th}^{\infty} f_h\left(h(x)|\omega_2\right) dh(x), \tag{1b}$$

$$\mathrm{R} = c_{12} P_1 e_1 + c_{21} P_2 e_2. \tag{1c}$$

The cost $c_{ij},\ i, j = 1, 2$ is the cost of classifying an observation as belonging to class j whereas it belongs to class i; $c_{ii} = 0$, which means there is no cost for correct classification; and P_i is the prior probability of each class, $i = 1, 2$. The risk (1c) is called the "error rate" Err, or probability of misclassification (PMC), when putting $c_{12} = c_{21} = 1$, which is denoted by the 0-1 cost, or loss.

The receiver operating characteristics (ROC) curve is a plot of the true positive fraction (TPF), which is $1 - \mathrm{FNF}$, versus the FPF. Then the area under the curve (AUC) is given by:

$$\mathrm{AUC} = \int_0^1 \mathrm{TPF}\, d(\mathrm{FPF}). \tag{2a}$$

$$= \Pr\big[h(x)|\omega_2 < h(x)|\omega_1\big], \tag{2b}$$

which expresses how the classifier scores for class ω_1 are stochastically larger than those of class ω_2.

If the distributions F_1 and F_2 are not known, a setup that is called nonparametric or distribution-free, any performance measure can be estimated only numerically from a given dataset, called the testing dataset. This is regardless of the testing paradigm, i.e., whether this testing dataset is obtained by simulation, resampling, or sequestering. This is done by assigning equal probability mass for each observation:

$$\hat{F} : \text{mass } \frac{1}{n} \text{ on } t_i,\ i = 1, \ldots, n, \tag{3}$$

where n is the size of the testing dataset. Lemma 1 shows that this is the maximum likelihood estimator (MLE) of the distribution F.

In this case the performance measures (1) can be obtained as follows.

$$\widehat{\mathrm{FNF}} = \widehat{e_1} = \frac{1}{n} \sum_{i=1}^{n} I_{h(x_i|\omega_1)<th} \tag{4a}$$

$$\widehat{\mathrm{FPF}} = \widehat{e_2} = \frac{1}{n} \sum_{i=1}^{n} I_{h(x_i|\omega_2)>th} \tag{4b}$$

$$\widehat{\mathrm{R}(\eta)} = \frac{1}{n} \left(c_{12}\, \widehat{e_1}\, n_1 + c_{21}\, \widehat{e_2}\, n_2\right). \tag{4c}$$

The indicator function I_{cond} equals 1 or 0 when the Boolean expression *cond* is true or false, respectively. The values n_1 and n_2 are the number of observations in the two classes respectively, and $\widehat{P_1}$ and $\widehat{P_2}$ are the estimated a priori probabilities for each class.

As the the two components TPF and FPF defined a single operating point on the ROC, the two components $\widehat{\text{TPF}}(= 1 - \widehat{\text{FNF}})$ and $\widehat{\text{FPF}}$ give one point on the empirical (estimated) ROC curve. To draw the complete curve in the nonparametric situation, the classifier's sore is calculated for each point of the available dataset. Then all possible thresholds are considered in turn, i.e., the threshold values between every two successive scores. At each threshold value a point on the ROC curve is calculated. Then the AUC (2a) can be estimated from the empirical ROC curve using the trapezoidal rule:

$$\widehat{\text{AUC}} = \frac{1}{2}\sum_{i=2}^{n_{th}} \left(\text{FNF}_i - \text{FNF}_{i-1}\right)\left(\text{TPF}_i + \text{TPF}_{i-1}\right), \tag{5}$$

where n_{th} is the number of threshold values taken over the dataset. By plotting the empirical ROC curve, it is easy to see that (5) is the same as the Mann-Whitney statistic—which is another form of the Wilcoxon rank-sum test [15, Chap. 4]—defined by:

$$\widehat{\text{AUC}} = \frac{1}{n_1 n_2}\sum_{j=1}^{n_2}\sum_{i=1}^{n_1} \psi\left(h\left(x_i|\omega_1\right), h\left(x_j|\omega_2\right)\right), \tag{6a}$$

$$\psi(a, b) = \begin{cases} 1 & a > b \\ 1/2 & a = b \\ 0 & a < b \end{cases}. \tag{6b}$$

It is interesting, as well, to know from the theory of U-statistics [25] that the estimator (6) is the uniform minimum variance unbiased estimator (UMVUE) for the probability (2b) under the distribution (3).

All the estimators given above have the nice property of converging to their corresponding population definitions, (1) and (2), as the size of the testing set goes to infinity. It is worth mentioning that each of the error estimators $\hat{e}_1$ and $\hat{e}_2$ in (4) is called a one-sample statistic, because its kernel $I_{(\cdot)}$ requires only one observation from either distributions. However, the AUC estimator in (6) is a two-sample statistic since its kernel $\psi(\cdot, \cdot)$ requires two observations, one from each distribution. This is a fundamental difference between both estimators (statistics) which will be treated and explained carefully in the present chapter.

1.3 Roadmap

The rest of this chapter is organized as follows. Section 2 paves the road to the chapter by reviewing the nonparametric estimators for estimating the mean and variance of one-sample statistics, including the preliminaries of bootstraps and influence function. This section is a very concise review mainly of the work done in Hampel [16], Efron and Tibshirani [11], and Huber [21]. Section 3 switches gears and reviews the nonparametric estimators that estimate the mean and variance of a special kind of statistics, i.e., the error rate of classification rules. This section is a concise review of the work done mainly in Efron [8], and Efron and Tibshirani [13]. Section 4 explains how the nonparametric estimators that estimate the error rate, a one-sample statistic, can be extended to estimate the AUC, a two-sample statistic. It does so by providing theoretical parallelism between the two sets of estimators and showing that the extension is rigorous and not just an ad hoc application. Section 6 concludes the chapter and provides a discussion and an advice for practitioners.

2 Nonparametric Methods for Estimating the Bias and the Variance of a Statistic

Consider a statistic s that is a function of a dataset $\mathbf{x} : \{x_i,\ i = 1, \ldots, n\}$, where $x_i \overset{i.i.d}{\sim} F$. The statistic s is now a random variable and its variability comes from the variability of x_i. Suppose that this statistic is used to estimate a real-valued parameter $\theta = f\,(F)$. Then $\hat{\theta} = s\,(\mathbf{x})$ has expected value E $s(\mathbf{x})$ and variance Var $s(\mathbf{x})$. The mean squared error (MSE) of the estimator $\hat{\theta}$ is defined as:

$$\mathrm{MSE}(\hat{\theta}) = \mathrm{E}\left[\hat{\theta} - \theta\right]^2. \tag{7}$$

The root of the mean squared error (RMS) has the same units and is on the same scale of the original variable θ, and hence has more intuitive value. The bias of the estimator $\hat{\theta} = s\,(\mathbf{x})$ is defined by the difference between the true value of the parameter and the expectation of the estimator, i.e.,

$$\mathrm{bias}_F\left(\hat{\theta}\right) = \mathrm{E}_F s\,(\mathbf{x}) - \theta. \tag{8}$$

Then, it is known that, the MSE in (7) can be decomposed to:

$$\mathrm{MSE}(\hat{\theta}) = \mathrm{bias}_F^2\left(\hat{\theta}\right) + \mathrm{Var}_F\hat{\theta}. \tag{9}$$

A critical question is whether the bias and variance of the statistic s in (9) may be estimated from the available dataset?

2.1 *Bootstrap Estimate*

The bootstrap was introduced by Efron [5] to estimate the standard error of a statistic. The bootstrap mechanism is implemented by treating the current dataset $\mathbf{x}$ as a

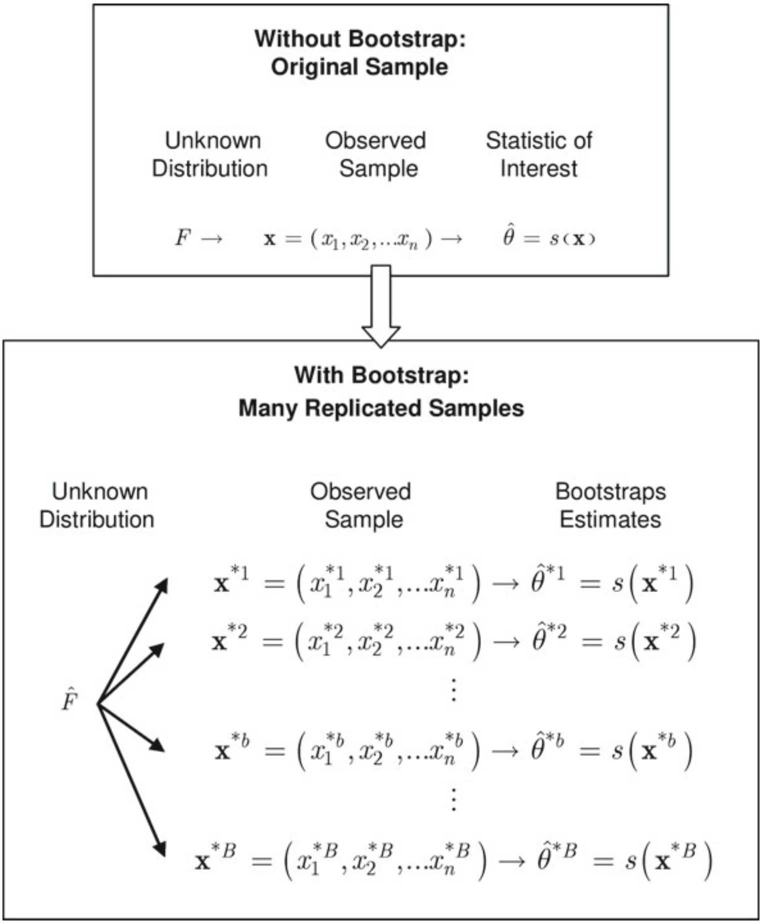
Without Bootstrap:
Original Sample
Unknown Distribution
Observed Sample
Statistic of Interest
$F \rightarrow$
$\mathbf{x} = (x_1, x_2, \ldots x_n) \rightarrow$
$\hat{\theta} = s(\mathbf{x})$
With Bootstrap:
Many Replicated Samples
Unknown Distribution
Observed Sample
Bootstraps Estimates
$\hat{F}$
$\mathbf{x}^{*1} = \left(x_1^{*1}, x_2^{*1}, \ldots x_n^{*1}\right) \rightarrow \hat{\theta}^{*1} = s\left(\mathbf{x}^{*1}\right)$
$\mathbf{x}^{*2} = \left(x_1^{*2}, x_2^{*2}, \ldots x_n^{*2}\right) \rightarrow \hat{\theta}^{*2} = s\left(\mathbf{x}^{*2}\right)$
$\vdots$
$\mathbf{x}^{*b} = \left(x_1^{*b}, x_2^{*b}, \ldots x_n^{*b}\right) \rightarrow \hat{\theta}^{*b} = s\left(\mathbf{x}^{*b}\right)$
$\vdots$
$\mathbf{x}^{*B} = \left(x_1^{*B}, x_2^{*B}, \ldots x_n^{*B}\right) \rightarrow \hat{\theta}^{*B} = s\left(\mathbf{x}^{*B}\right)$

Fig. 1 Bootstrap mechanism: B bootstrap replicates are withdrawn (by sampling and replacement) from the original sample. From each replicate the statistic is calculated. (The idea behind this figure first appeared in [11, Fig. 6.1, pp. 48])

representation for the population distribution F; i.e., approximating the distribution F by the MLE defined in (3). Then B bootstrap samples are drawn from that empirical distribution. Each bootstrap replicate is of size n, the same size as $\mathbf{x}$, and is obtained by sampling with replacement. Then in a bootstrap replicate some case x_i, in general, will appear more than once at the expense of another x_j that will not appear. The original dataset will be treated now as the population, and the replicates will be treated as samples from the population. This situation is illustrated in Fig. 1. Each of these bootstrap replicates is denoted by $\mathbf{x}^{*b}$, $b = 1, \ldots, B$, and the corresponding bootstrap replications of the statistics $\hat{\theta} = s(\mathbf{x})$ itself are given by:

$$\hat{\theta}^{*b} = s(\mathbf{x}^{*b}), \quad b = 1, \ldots, B, \tag{10}$$

The bootstrap estimate of bias and standard error are defined by:

$$\text{bias}_B(\hat{\theta}) = \hat{\theta}^{*} - \hat{\theta}, \tag{11}$$

$$\widehat{\text{SE}}_B = \left[\frac{1}{(B-1)} \sum_{b=1}^{B} \left[\hat{\theta}^{*b} - \hat{\theta}^{*} \right]^2 \right]^{1/2}, \tag{12}$$

$$\hat{\theta}^{*} = \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}^{*b}. \tag{13}$$

Either in estimating the bias or the standard error, the larger the number of bootstraps B the closer the estimate to the asymptotic value, i.e.,

$$\lim_{B \to \infty} \widehat{\text{SE}}_B(\hat{\theta}^{*}) = \text{SE}_{\hat{F}}(\hat{\theta}^{*}). \tag{14}$$

For more details and some examples the reader is referred to [11, Chap. 6, 7, and 10].

2.2 Jackknife Estimate

Instead of replicating from the original dataset, a new set $\mathbf{x}_{(i)}$ is created by removing the case x_i from the dataset. Then the jackknife samples are defined by:

$$\mathbf{x}_{(i)} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n), \quad i = 1, \ldots, n, \tag{15}$$

and the n-jackknife replications of the statistic $\hat{\theta}$ are:

$$\hat{\theta}_{(i)} = s(\mathbf{x}_{(i)}), \quad i = 1, \ldots, n. \tag{16}$$

The jackknife estimates of bias and standard error are defined by:

$$\widehat{\text{bias}}_J = (n-1)(\hat{\theta}^{J} - \hat{\theta}), \tag{17}$$

$$\widehat{\text{SE}}_J = \left[\frac{n-1}{n} \sum_{i=1}^{n} (\hat{\theta}_{(i)} - \hat{\theta}^{J})^2 \right]^{1/2}, \tag{18}$$

$$\hat{\theta}^{J} = \frac{1}{n} \sum_{i=1}^{n} \hat{\theta}_{(i)}. \tag{19}$$

For motivation behind the factors $(n-1)$ and $(n-1)/n$ in (17) see [11, Chap. 11]. The jackknife estimate of variance is discussed in detail in Efron [6] and Efron and Stein [10].

2.3 Bootstrap Versus Jackknife

Usually, it requires up to 200 bootstraps to yield acceptable bootstrap estimates; (in special situations like estimating the uncertainty in classifier performance it may take up to thousands of bootstraps). Hence, this requires calculating the statistic $\hat{\theta}$ the same number of times B, as well. In the case of the jackknife, it requires only n calculations as shown in (16). If the sample size is smaller than the required number of bootstraps, the jackknife is more economical in terms of computational cost.

In terms of accuracy, the jackknife can be seen to be an approximation to the bootstrap when estimating the standard error of a statistic [11, Chap. 20]. Thus, if the statistic is linear they almost give the same result; (the bootstrap gives the jackknife estimate multiplied by $[(n-1)/n]^{1/2}$). A statistic $s(\mathbf{x})$ is said to be linear if:

$$s(\mathbf{x}) = \mu + \frac{1}{n}\sum_{i=1}^{n} \alpha(x_i), \tag{20}$$

where μ is a constant and $\alpha(\cdot)$ is a function. This also can be viewed as having one data point at a time in the argument of the function α. Similarly, the jackknife can be seen as an approximation to the bootstrap when estimating the bias. If the statistic is quadratic, they almost agree except in a normalizing factor . A statistic $s(\mathbf{x})$ is quadratic if:

$$s(\mathbf{x}) = \mu + \frac{1}{n}\sum_{1\le i\le n} \alpha(x_i) + \frac{1}{n^2}\sum_{1\le i<j\le n} \beta(x_i, x_j). \tag{21}$$

An in-depth treatment of the bootstrap and jackknife, and their relation to each other, in mathematical detail is provided by Efron [7, Chaps. 1–5].

If the statistic is not smooth the jackknife will fail. Informally speaking, a statistic is said to be smooth if a small change in the data leads to a small change in the statistic. An example of a non-smooth statistic is the median. If the sample cases are ranked and the median is calculated, it will not change when a sample case changes unless this sample case bypasses the median value. Using the same argument, we can see that an example of a smooth statistic is the sample mean.

2.4 Influence Function, Infinitesimal Jackknife, and Estimate of Variance

The infinitesimal jackknife was introduced by Jaeckel [22]. The concept of the influence curve was introduced later by Hampel [16]. In the present context and for pedagogical purposes, the influence curve will be explained before the infinitesimal jackknife, since the former can be understood as the basis for the latter.

Following Hampel [16], let $\mathfrak{R}$ be the real line and s be a real-valued functional defined on the distribution F, which is defined on $\mathfrak{R}$. The distribution F can be perturbed by adding some probability measure (mass) on a point x. This should be balanced by a decrement in F elsewhere, resulting in a new probability distribution $G_{\varepsilon,x}$ defined by:

$$G_{\varepsilon,x} = (1-\varepsilon)F + \varepsilon\delta_x, \ \ x \in \mathfrak{R}. \tag{22}$$

Then, the influence curve $IC_{s,F}(\cdot)$ is defined by:

$$IC_{s,F}(x) = \lim_{\varepsilon \to 0^+} \frac{s\left(\left(1-\varepsilon\right)F + \varepsilon\delta_x\right) - s\left(F\right)}{\varepsilon}. \tag{23}$$

It should be noted that F does not have to be a discrete distribution. A simple example of applying the influence curve concept is to consider the expectation $s = \int x \, dF(x) = \mu$. Substituting back in (23) gives:

$$IC_{s,F}(x) = x - \mu. \tag{24}$$

The meaning of this formula is the following: the rate of change of the functional s with the probability measure at a point x is $x - \mu$. This is how the point x influences the functional s. The influence curve can be used to linearly approximate a functional s, along with its variance, which is similar to taking up to only the first-order term in a Taylor series expansion (Appendix 7.2).

It is important to state here that s should be a functional in $\hat{F}$ that is an approximation to F, as was initially assumed in (23). If for example the value of the statistic s changes if every sample case x_i is duplicated, i.e., repeated twice, this is not a functional statistic. An example of a functional statistic is the biased version of the variance estimate $\Sigma_i(x_i - \bar{x}_i)^2/n$, while the unbiased version $\Sigma_i(x_i - \bar{x}_i)^2/(n-1)$ is not a functional statistic. Generally, any approximation $s(\hat{F})$ to the functional $s(F)$, by approximating F by the MLE $\hat{F}$, obviously will be functional. In such a case the statistic $s(\hat{F})$ is called the plug-in estimate of the functional $s(F)$. Moreover, the influence function (IF) method for variance estimation is applicable only to those functional statistics whose derivative (73) exists. If that derivative exists, the statistic is called a smooth statistic; i.e., a small change in the dataset leads a small change in the statistic. For instance, although the median is a functional statistic in the sense that duplicating any sample case will result in the same value of the median, it is not smooth as described at the end of Sect. 2.3. A key reference for the IF is Hampel [17]. Appendix 7.2 shows an interesting connection to the jackknife estimate.

3 Nonparametric Methods for Estimating the Error Rate of a Classification Rule

The review provided in this section is a terse summary of the main work of Efron [8, 11, 13]. In the previous section the statistic, or generally speaking the functional, was a function of just one dataset. For a non-fixed design, i.e., when the predictors of the testing set do not have to be the same as the predictors of the training dataset, a slight clarification for the previous notations is needed. The classification rule trained on the training dataset **t** will be denoted as $\eta_{\mathbf{t}}$. Any new observation that does not belong to **t** will be denoted by $t_0 = (x_0, y_0)$. Therefore, the classification loss is given by $L(y_0, \eta_{\mathbf{t}}(x_0))$. Any performance measure conditional on that training dataset will be similarly subscripted. Thus, all the performance measures should be subscripted **t**; and hence the risk and the error rate (1) should be denoted by $\mathrm{R}_{\mathbf{t}}$ and $\mathrm{Err}_{\mathbf{t}}$, respectively. In the sequel, for simplicity and WLOG, the 0-1 loss function will be used. In such a case the conditional error rate will be given by:

$$\mathrm{Err}_{\mathbf{t}} = \mathrm{E}_{0F} L\left(y_0, \eta_{\mathbf{t}}\left(x_0\right)\right), \quad (x_0, y_0) \sim F. \tag{25}$$

The expectation E_{0F} is subscripted so to emphasize that it is taken over the observations $t_0 \notin \mathbf{t}$. If the performance is measured in terms of the error rate and we are interested in the mean performance, not the conditional one, then it is given by:

$$\mathrm{Err} = \mathrm{E}_{\mathbf{t}} \mathrm{Err}_{\mathbf{t}}. \tag{26}$$

The expectation $\mathrm{E}_{\mathbf{t}}$ is the expectation over the training dataset **t**, which would be the same if we had written E_F; for notation clarity the former is chosen.

Consider a classification rule $\eta_{\mathbf{t}}$ already trained on a training dataset **t**. A natural next question is, given that there is just a single dataset available, how to use this dataset in assessing the classifier performance as well? Said differently, how should one estimate, using only the available dataset, the true classification performance of a classification rule in predicting new observations; these observations are different from those on which the rule was trained. In this section, we will review the principal methods in the literature for estimating both the true error rate (25) and its mean (26) of a classification rule.

3.1 Apparent Error

The apparent error is the error of the fitted model when it is tested on the same training data. Of course it is downward biased with respect to the true error rate since it results from testing on the same information used in training [9]. The apparent error is defined by:

$$\overline{\mathrm{Err}}_{\mathbf{t}} = \mathrm{E}_{\hat{F}} L(y, \eta_{\mathbf{t}}(x)), \quad (x, y) \in \mathbf{t} \tag{27a}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \left[I_{\hat{h}_{\mathbf{t}}(x_i|\omega_1)<th} + I_{\hat{h}_{\mathbf{t}}(x_i|\omega_2)>th} \right]. \tag{27b}$$

Overfitting a classifier to minimize the apparent error is not the goal. The goal is to minimize the true error rate (25) or its mean (26).

3.2 Cross Validation (CV)

The basic concept of CV, as a resampling approach, has been proposed in different articles since the mid-1930s. The concept simply leans on splitting the data into two parts; the first part is used in design (or training) without any involvement of the second part. Then the second part is used to test the designed procedure; this is to test how the designed procedure will behave for new datasets. Stone [28] is a key reference for CV that proposes different criteria for optimization.

CV can be used to assess the prediction error of a model or in model selection. The true error rate in (25) is the expected error rate for a classification rule if tested on the population, conditional on a particular training dataset $\mathbf{t}$. This performance measure can be approximated by the leave-one-out CV (LOOCV) by:

$$\widehat{\mathrm{Err}}_{\mathbf{t}}^{cv1} = \frac{1}{n} \sum_{i=1}^{n} L\left(y_i, \eta_{\mathbf{t}^{(i)}}(x_i)\right), \quad (x_i, y_i) \in \mathbf{t}. \tag{28}$$

This is done by training the classification rule on the dataset $\mathbf{t}^{(i)}$ that does not include the case t_i; then testing the trained rule on that omitted case. This proceeds in "round-robin" fashion until all cases have contributed one at a time to the error rate. There is a hidden assumption in this mechanism: the training dataset $\mathbf{t}$ will not change very much by omitting a single case. Therefore, testing on the omitted observation one at a time accounts for testing approximately the same trained rule on n new cases, all different from each other and different from those the classifier has been trained on. Besides this LOOCV, there are other versions named K-fold (or leave-n/K-out). In such versions the whole dataset is split into K roughly equal-sized subsets, each of which contains approximately n/K observations. The classifier is trained on $K - 1$ subsets and tested on the left-out one; hence we have K iterations. It is clear that the LOOCV is a special case of the K-fold CV, where $K = n$.

It is of interest to assess this estimator to see whether it estimates the conditional true error $\mathrm{E}\left[\widehat{\mathrm{Err}}_{\mathbf{t}}^{cv1} - \mathrm{Err}_{\mathbf{t}}\right]^2$, with small MSE, as was designed or not. Many simulation results, e.g., Efron [8], show that there is only a very weak correlation between the CV estimator $\widehat{\mathrm{Err}}_{\mathbf{t}}^{cv1}$ and the conditional true error rate $\mathrm{Err}_{\mathbf{t}}$. This issue is discussed in mathematical detail in the excellent paper by Zhang [35]. Those other estimators that are based on resampling as well, and will be reviewed below, are shown to have this same attribute. This very interesting (and perhaps surprising) result means the following: whether the estimator is designed to estimate the conditional performance or the mean performance it indeed estimates the latter because of the weak correlation with the former.

3.3 Bootstrap Methods for Error Rate Estimation

The prediction error in (25) is a function of the training dataset $\mathbf{t}$ and the testing population F. Bootstrap estimation can be implemented here by treating the empirical distribution $\hat{F}$ as an approximation to the actual population distribution F. By replicating from that distribution one can simulate many training datasets $\mathbf{t}^{*b},\ b = 1, \ldots, B$. For every replicated training dataset the classifier will be trained and then tested on the original dataset $\mathbf{t}$. This is the simple bootstrap (SB) estimator approach [11, Sect. 17.6] that was defined formally by:

$$\widehat{\mathrm{Err}}_{\mathbf{t}}^{SB} = \mathrm{E}_* \sum_{i=1}^{n} L(y_i, \eta_{\mathbf{t}^*}(x_i))/n, \quad \hat{F} \to \mathbf{t}^*. \tag{29}$$

It should be noted that this estimator no longer estimates the true error rate (25) because the expectation taken over the bootstraps mimics an expectation taken over the population of trainers, i.e., it is not conditional on a particular training dataset. Rather, the estimator (29) estimates the expected performance of the classifier $\mathrm{E}_F \mathrm{Err}_{\mathbf{t}}$. For a finite number of bootstraps, the expectation (29) can be approximated by:

$$\widehat{\mathrm{Err}}_{\mathbf{t}}^{SB} = \frac{1}{B} \sum_{b=1}^{B} \sum_{i=1}^{n} L\left(y_i, \eta_{\mathbf{t}^{*b}}(x_i)\right)/n. \tag{30}$$

3.3.1 Leave-One-Out Bootstrap (LOOB)

The previous estimator is obviously biased since the original dataset $\mathbf{t}$ used for testing includes part of the training data in every bootstrap replicate. Efron [8] proposed that, after training the classifier on every bootstrap replicate, it is tested on those cases in the set $\mathbf{t}$ that are not included in the training; this concept can be developed as follows. Equation (30) can be rewritten by interchanging the order of the double summation to give:

$$\widehat{\mathrm{Err}}_{\mathbf{t}}^{SB} = \frac{1}{n} \sum_{i=1}^{n} \sum_{b=1}^{B} L\left(y_i, \eta_{\mathbf{t}^{*b}}(x_i)\right) \Big/ B. \tag{31}$$

This equation is formally identical to (30) but it expresses a different mechanism for evaluating the same quantity. It says that, for a given point, the average performance over the bootstrap replicates is calculated; then this performance is averaged over all the n cases. Now, if every case t_i is tested only from those bootstraps that did not include it in the training, a slight modification of the previous expression yields the leave-one-out bootstrap (LOOB) estimator:

$$\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)} = \frac{1}{n} \sum_{i=1}^{n} \left[\sum_{b=1}^{B} I_i^b L\left(y_i, \eta_{\mathbf{t}^{*b}}(x_i)\right) \Big/ \sum_{b'=1}^{B} I_i^{b'} \right], \tag{32}$$

where the indicator function I_i^b equals one when the case t_i is not included in the training replicate b, and zero otherwise. Efron and Tibshirani [13] emphasized a critical point about the difference between this bootstrap estimator and the LOOCV. The CV tests on a given sample case t_i, having been trained just once on the remaining dataset. By contrast, the LOOB tests on a given sample case t_i using a large number of classifiers that result from a large number of bootstrap replicates that do not contain that sample. This results in a smoothed cross-validation-like estimator. We explained and elaborated on this smoothness property in Yousef [30].

3.3.2 The Refined Bootstrap (RB)

The SB and the LOOB, from their definitions, look like designed to estimate the mean true error rate (26) of a classifier. For estimating the true conditional error rate of a classifier, conditional on a particular training dataset, Efron [8] proposed to correct for the downward biased estimator $\overline{\mathrm{Err}}_{\mathbf{t}}$. Since the true error rate $\mathrm{Err}_{\mathbf{t}}$ can be written as $\overline{\mathrm{Err}}_{\mathbf{t}} + (\mathrm{Err}_{\mathbf{t}} - \overline{\mathrm{Err}}_{\mathbf{t}})$, then it can be approximated by $\overline{\mathrm{Err}}_{\mathbf{t}} + \mathrm{E}_F(\mathrm{Err}_{\mathbf{t}} - \overline{\mathrm{Err}}_{\mathbf{t}})$. The term $(\mathrm{Err}_{\mathbf{t}} - \overline{\mathrm{Err}}_{\mathbf{t}})$ is called the optimism. The expectation of the optimism can be approximated over the bootstrap population. Finally the refined bootstrap approach, as named in Efron and Tibshirani [11, Sect. 17.6], gives the estimator:

$$\widehat{\mathrm{Err}}_{\mathbf{t}}^{RB} = \overline{\mathrm{Err}}_{\mathbf{t}} + \mathrm{E}_*(\mathrm{Err}_{\mathbf{t}*}(\hat{F}) - \overline{\mathrm{Err}}_{\mathbf{t}*}), \tag{33}$$

where $\mathrm{Err}_{\mathbf{t}*}(\hat{F})$ represents the error rate obtained from training the classifier on all bootstrap replicates $\mathbf{t}^*$ and testing on the empirical distribution $\hat{F}$. This can be approximated for a limited number of bootstraps by:

$$\widehat{\mathrm{Err}}_{\mathbf{t}}^{RB} = \overline{\mathrm{Err}}_{\mathbf{t}} + \frac{1}{B} \sum_{b=1}^{B} \left[\sum_{i=1}^{n} L\left(y_i, \eta_{\mathbf{t}^{*b}}(x_i)\right)/n - \sum_{i=1}^{n} L\left(y_{ib}^*, \eta_{\mathbf{t}^{*b}}(x_{ib}^*)\right)/n \right]. \tag{34}$$

3.3.3 The 0.632 Bootstrap

If the concept used in developing the LOOB estimator, i.e., testing on cases not included in training, is used again in estimating the optimism described above, this gives the 0.632 bootstrap estimator. Since the probability of including a case t_i in the bootstrap $\mathbf{t}^{*b}$ is given by:

$$\Pr(t_i \in \mathbf{t}^{*b}) = 1 - (1 - 1/n)^n \approx 1 - e^{-1} = 0.632, \tag{35}$$

the effective number of sample cases contributing to a bootstrap replicate is approximately 0.632 of the size of the training dataset. Efron [8] introduced the concept of a *distance* between a point and a sample in terms of a probability. Having trained on a bootstrap replicate, testing on those cases in the original dataset not included in the bootstrap replicate accounts for testing on a set far from the training one, i.e., the bootstrap replicate. This is because every sample case in the testing set has zero probability of belonging to the training dataset, i.e., very distant from the training dataset. This is a reason for why the LOOB is an upwardly biased estimator. Efron [8] showed roughly that:

$$\mathrm{E}_F \left[\mathrm{Err}_{\mathbf{t}} - \overline{\mathrm{Err}}_{\mathbf{t}} \right] \approx 0.632 \, \mathrm{E}_F \left[\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)} - \overline{\mathrm{Err}}_{\mathbf{t}} \right]. \tag{36}$$

Substituting back in (33) gives the 0.632 estimator:

$$\widehat{\mathrm{Err}}_{\mathbf{t}}^{(0.632)} = 0.368 \, \overline{\mathrm{Err}}_{\mathbf{t}} + 0.632 \, \widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)}. \tag{37}$$

The proof of the above results can be found in Efron [8] and Efron and Tibshirani [11, Sect. 6].

The motivation behind this estimator as stated earlier is to correct for the downward biased apparent error by adding a piece of the upward biased LOOB estimator. But an increase in variance should be expected as a result of adding this piece of the relatively variable apparent error. Moreover, this new estimator is no longer smooth since the apparent error itself is unsmooth.

3.3.4 The 0.632+ Bootstrap Estimator

The 0.632 estimator reduces the bias of the apparent error. But for over-trained classifiers, i.e., those whose apparent error tends to be zero, the 0.632 estimator is still downward biased. Breiman et al. [3] provided the example of an overfitted rule, like 1NN where the apparent error is zero. If, however, the class labels are assigned randomly to the predictors the true error rate will obviously be 0.5. But substituting in (37) gives an estimate of 0.632 × 0.5 = 0.316. To account for this bias for such over-fitted classifiers, Efron and Tibshirani [13] defined the *no-information error rate* γ by:

$$\gamma = \mathrm{E}_{0F_{ind}} L\left(y_0, \eta_{\mathbf{t}}(x_0)\right), \tag{38}$$

where F_{ind} means that x_0 and y_0 are distributed marginally as F but they are independent. Or said differently, the label is assigned randomly to the predictor. Then for a training sample $\mathbf{t}$, γ can be estimated by:

$$\hat{\gamma} = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} L\left(y_i, \eta_{\mathbf{t}}(x_j)\right). \tag{39}$$

This means that the n predictors have been permuted with the n responses to produce n^2 non-informative cases. In the special case of binary classification, let $\hat{p}_1$ be the proportion of the response classified as belonging to class 1. Also, let $\hat{q}_1$ be the proportion of the responses classified as belonging to class 1. Then (39) reduces to:

$$\hat{\gamma} = \hat{p}_1(1 - \hat{q}_1) + (1 - \hat{p}_1)\hat{q}_1. \tag{40}$$

Also define the *relative overfitting rate*:

$$\hat{R} = \frac{\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)} - \overline{\mathrm{Err}}_{\mathbf{t}}}{\hat{\gamma} - \overline{\mathrm{Err}}_{\mathbf{t}}}. \tag{41}$$

Efron and Tibshirani [13] showed that the bias of the 0.632 estimator for the case of over-fitted classifiers is alleviated by using a renormalized version of that estimator:

$$\widehat{\mathrm{Err}}_{\mathbf{t}}^{(0.632+)} = (1 - \hat{w})\overline{\mathrm{Err}}_{\mathbf{t}} + \hat{w}\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)}, \tag{42a}$$

$$\hat{w} = \frac{0.632}{1 - 0.368\hat{R}}. \tag{42b}$$

It is useful to express the 0.632+ estimator in terms of its predecessor, the 0.632 estimator. Combining (37), (40), and (41) then substituting in (42a) yields:

$$\widehat{\mathrm{Err}}_{\mathbf{t}}^{(0.632+)} = \widehat{\mathrm{Err}}_{\mathbf{t}}^{(0.632)} + (\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)} - \overline{\mathrm{Err}}_{\mathbf{t}})\frac{0.368 \cdot 0.632 \cdot \hat{R}}{1 - 0.368\hat{R}}. \tag{43}$$

Efron and Tibshirani [13] consider the possibility that $\hat{R}$ lies out of the region [0, 1]. This leads to their proposal of defining:

$$\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)\prime} = \min(\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)}, \hat{\gamma}), \tag{44}$$

$$\hat{R}' = \begin{cases} (\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)} - \overline{\mathrm{Err}}_{\mathbf{t}})/(\hat{\gamma} - \overline{\mathrm{Err}}_{\mathbf{t}}) & \overline{\mathrm{Err}}_{\mathbf{t}} < \widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)} < \gamma \\ 0 & \text{otherwise} \end{cases}, \tag{45}$$

to obtain a modification to (43) that finally becomes:

$$\widehat{\mathrm{Err}}_{\mathbf{t}}^{(0.632+)} = \widehat{\mathrm{Err}}_{\mathbf{t}}^{(0.632)} + (\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)\prime} - \overline{\mathrm{Err}}_{\mathbf{t}})\frac{0.368 \cdot 0.632 \cdot \hat{R}'}{1 - 0.368\hat{R}'}. \tag{46}$$

3.4 Estimating the Standard Error of Error Rate Estimators

What have been reviewed above are several resampling methods: the CV, 0.632, and 0.632+ estimate the conditional error rate of a classification rule, conditional on that training dataset; and the LOOB estimates the mean error rate, where the expectation is taken over the population of training datasets. Regardless of what the estimator is designed to estimate, it is still a function of the current dataset $\mathbf{t}$, i.e., it is a random variable. If, e.g., the LOOB estimator $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)}$ is considered, it estimates a constant real-valued parameter $\mathrm{E}_{0F}\mathrm{E}_F L(y_0, \eta_{\mathbf{t}}(x_0))$ with expectation taken over all the trainers and then over all the testers, respectively; this is the overall mean error rate. Yet, $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)}$ is a random variable whose variability comes from the finite size of the available dataset. If the classifier is trained and tested on a very large number of observations, this would approximate training and testing on the entire population, and the variability would shrink to zero. This also applies for any performance measure other than the error rate. So, we are interested now in estimating $\mathrm{Var}_{\mathbf{t}}\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)}$, the variance of the estimator, not estimating $\mathrm{Var}_{\mathbf{t}}\mathrm{Err}_{\mathbf{t}}$, the variance of the true performance.

The next question then is, having estimated the mean performance of a classifier: what is the associated uncertainty of this estimate. Said differently: an estimate of the variance of this estimator be obtained from the same training dataset? Efron and Tibshirani [13] proposed the use of the IF method (Sect. 2.4), to estimate the uncertainty (variability) in $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)}$. The reader is alerted that estimators that incorporate a piece of the apparent error are not suitable for the IF method. Such estimators are not smooth because the apparent error itself is not smooth.

By recalling the definitions of Sect. 2.4, $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)}$ is now the statistic $s(\hat{F})$. To simplify notation, the error $L(y_i, \eta_{\mathbf{t}^{*b}}(x_i))$ may be denoted by L_i^b, and define the following notation:

$$l_{\cdot}^{b} = \frac{1}{n}\sum_{i=1}^{n} I_i^b L_i^b, \tag{47}$$

Also, define N_i^b to be the number of times the case t_i is included in the bootstrap b. Then, it has been proven in Efron and Tibshirani [12] that the IF of such an estimator is given by:

$$\left.\frac{\partial s(\hat{F}_{\varepsilon,i})}{\partial \varepsilon}\right|_{\varepsilon=0} = (2 + \frac{1}{n-1})(\hat{E}_i - \widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)}) + \frac{n\sum_{b=1}^{B}(N_i^b - \bar{N}_i)I_i^b}{\sum_{b=1}^{B} I_i^b}. \tag{48}$$

Combining (78) and (48) gives an estimation to the uncertainty in $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)}$.

4 Nonparametric Methods for Estimating the AUC of a Classification Rule

In the present section, we extend the study carried out in Efron [8], Efron and Tibshirani [13], and summarized in Sect. 3, to construct nonparametric estimators for the AUC (a two-sample statistic) analogue to those of the error rate (a one-sample statistic). Although some previous experimental comparative studies [26, 27, 32] were conducted to compare some of these resampling-based AUC estimators, in particular the 0.632 versions, there was no theoretical justification of using these estimators for the AUC. We provide here a full account of the different versions of bootstrap estimators reviewed in Sect. 3 and show how they can be formally extended to estimate the AUC.

4.1 Construction of Nonparametric Estimators for AUC

Before switching to the AUC, some more elaboration on Sect. 3 is needed. The SB estimator (29) can be rewritten as:

$$\widehat{\mathrm{Err}}_{\mathbf{t}}^{SB} = \mathrm{E}_{*}\mathrm{E}_{\widehat{F}}\left[L(\eta_{\mathbf{t}^{*}}(x), y)|\mathbf{t}^{*}\right]. \tag{49}$$

Since there would be some observation overlap between $\mathbf{t}$ and $\mathbf{t}^{*}$, this approach suffers an obvious bias as was introduced in that section. This was the motivation behind interchanging the expectations and defining the LOOB (Sect. 3.3.1). Alternatively, we could have left the order of the expectation but with testing on only those observations in $\mathbf{t}$ that do not appear in the bootstrap replication $\mathbf{t}^{*}$, i.e., the distribution $\hat{F}^{(*)}$. The parenthesis notation $(*)$ refers to excluding from $\widehat{F}$, in the testing stage, the training cases $\mathbf{t}^{*}$ that were generated from the bootstrap replication. We call the resulting estimator $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(*)}$, which we define formally by:

$$\widehat{\mathrm{Err}}_{\mathbf{t}}^{(*)} = \mathrm{E}_{*}\mathrm{E}_{\hat{F}^{(*)}}\left[L(\eta_{\mathbf{t}^{*}}(x), y)|\mathbf{t}^{*}\right] \tag{50}$$

We can give the inner expectation the notation $\mathrm{Err}_{\mathbf{t}^{*b}}(\widehat{F}^{(*)})$, and rewrite the estimator as:

$$\widehat{\mathrm{Err}}_{\mathbf{t}}^{(*)} = \mathrm{E}_{*}\mathrm{Err}_{\mathbf{t}^{*b}}(\widehat{F}^{(*)}) \tag{51a}$$

$$= \frac{1}{B}\sum_{b=1}^{B}\left[\sum_{i=1}^{N} I_i^b L(\eta_{\mathbf{t}^{*b}}(x_i), y_i)\Big/\sum_{i'=1}^{N} I_{i'}^b\right], \tag{51b}$$

where the indicator I_i^b equals one if the observation t_i is excluded from the bootstrap replication $\mathbf{t}^{*b}$, and equals zero otherwise. The inner expectation in (50) is taken over those observations not included in the bootstrap replication $\mathbf{t}^{*}$, whereas the outer expectation is taken over all the bootstrap replications.

Analogously to Sect. 3, and to what has been introduced above, we can define several bootstrap estimators for the AUC. The start is the SB estimate, which can be defined as:

$$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{SB} = \mathrm{E}_*\mathrm{AUC}_{\mathbf{t}^*}(\widehat{F}), \quad \widehat{F} \to \mathbf{t}^* \tag{52a}$$

$$= \mathrm{E}_*\left[\frac{1}{n_1 n_2}\sum_{j=1}^{n_2}\sum_{i=1}^{n_1}\psi(\hat{h}_{\mathbf{t}^*}(x_i), \hat{h}_{\mathbf{t}^*}(x_j))\right], \quad x_i \in \omega_1, x_j \in \omega_2. \tag{52b}$$

This averages the Mann-Whitney statistic over the bootstraps, where $\mathrm{AUC}_{\mathbf{t}^*}(\widehat{F})$ refers to the AUC obtained from training the classifier on the bootstrap replicate $\mathbf{t}^*$ and testing it on the empirical distribution $\widehat{F}$. In the approach used here, the bootstrap replicate $\mathbf{t}^*$ preserves the ratio between n_1 and n_2, which is called stratification. That is, the training sample $\mathbf{t}$ is treated as $\mathbf{t} = \mathbf{t}_1 \cup \mathbf{t}_2,\ \mathbf{t}_1 \in \omega_1,\ \mathbf{t}_2 \in \omega_2$; then n_1 cases are replicated from the first-class sample and n_2 cases are replicated from the second-class sample to produce $\mathbf{t}_1^*$ and $\mathbf{t}_2^*$ respectively, where $\mathbf{t}^* = \mathbf{t}_1^* \cup \mathbf{t}_2^*$. This was not needed when the performance measure was the error rate since it is a statistic that does not operate simultaneously on two different sets of observations as the Mann-Whitney statistic does (in U-statistic theory [25], error rate and Mann-Whitney are called one-sample and two-sample statistics respectively). The expectation (52a) is approximated by averaging over a finite number of bootstrap:

$$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{SB} = \frac{1}{B}\sum_{b=1}^{B}\mathrm{AUC}_{\mathbf{t}^{*b}}(\widehat{F}), \tag{53}$$

The same motivation behind the estimator (32) can be applied here, i.e., testing only on those cases in $\mathbf{t}$ that are not included in the training dataset $\mathbf{t}^{*b}$, in order to reduce the bias. This can be carried out in (53) without interchanging the summation order. The new estimator is named $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}$, where the parenthesis notation $(*)$ refers to the exclusion, in the testing stage, of the training cases that were generated from the bootstrap replication. Formally, we define this as:

$$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)} = \mathrm{E}_*\mathrm{AUC}_{\mathbf{t}^{*b}}(\widehat{F}^{(*)}) \tag{54a}$$

$$= \frac{1}{B}\sum_{b=1}^{B}\left[\sum_{j=1}^{n_2}\sum_{i=1}^{n_1}\psi(\hat{h}_{\mathbf{t}^*}(x_i), \hat{h}_{\mathbf{t}^*}(x_j))I_i^b I_j^b \Big/ \sum_{i'=1}^{n_1} I_{i'}^b \sum_{j'=1}^{n_2} I_{j'}^b\right]. \tag{54b}$$

The RB and 0.632 estimators can be introduced here in the same way it was used for the true error rate (Sect. 3.3.3) as:

$$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{RB} = \overline{\mathrm{AUC}}_{\mathbf{t}} + \mathrm{E}_*\left[\mathrm{AUC}_{\mathbf{t}*}(\widehat{F}) - \overline{\mathrm{AUC}}_{\mathbf{t}*}\right]. \tag{55}$$

Then, if testing is carried out on cases excluded from the bootstraps, analogously to the 0.632 estimator of the error rate, this gives rise to the 0.632 estimator of the AUC:

$$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632)} = 0.368\,\overline{\mathrm{AUC}_{\mathbf{t}}} + 0.632\,\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}. \tag{56}$$

It should be noted that this estimator is designed to estimate the true AUC for a classifier trained on the dataset **t** (the classifier performance conditional on the training dataset **t**). This is on contrary to the estimator (54) that estimates the mean performance of the classifier (this is the expectation over the training dataset population for the conditional performance).

The 0.632+ estimator $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632+)}$ develops from $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632)}$ in the same way as $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(0.632+)}$ developed from $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(0.632)}$ in Sect. 3.3.4. There are two modifications to the details. The first regards the *no-information error rate* γ; it can be proven that the *no-information* AUC is given by $\gamma_{\mathrm{AUC}} = 0.5$ (Lemma 2). The second regards the definitions (44), which should be modified to accommodate for the AUC. The new definitions are given by:

$$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632+)} = \widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632)} + (\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)\prime} - \overline{\mathrm{AUC}_{\mathbf{t}}})\frac{0.368 \cdot 0.632 \cdot \hat{R}'}{1 - 0.368\hat{R}'}, \tag{57a}$$

$$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)\prime} = \max\bigl(\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}, \gamma_{\mathrm{AUC}}\bigr), \tag{57b}$$

$$\hat{R}' = \begin{cases} \frac{(\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)} - \overline{\mathrm{AUC}_{\mathbf{t}}})}{(\gamma_{\mathrm{AUC}} - \overline{\mathrm{AUC}_{\mathbf{t}}})} & \text{if } \overline{\mathrm{AUC}_{\mathbf{t}}} > \widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)} > \gamma_{\mathrm{AUC}} \\ 0 & \text{otherwise} \end{cases}. \tag{57c}$$

To this end, we have constructed the AUC nonparametric estimators analogue to those of the error rate. Some of them, mainly the 0.632+ estimator, will have the least bias [13]. However, all of these estimators are not "smooth" and not eligible for the variance estimation via, e.g., the IF method (Sects. 2.4 and 3.4). The only estimator that may seem smooth, is the star versions $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(*)}$and $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}$. However, the inner components $\mathrm{Err}_{\mathbf{t}^{*b}}(\widehat{F}^{(*)})$ and $\mathrm{AUC}_{\mathbf{t}^{*b}}(\widehat{F}^{(*)})$ are unsmooth themselves, because the classifier is trained on just one dataset. Applying the influence function enforces distributing the differential operator $\partial/\partial\varepsilon$, of the IF, over the summation to be encountered by these unsmooth components.

4.2 The Leave-Pair-Out Boostrap (LPOB) $\widehat{\mathbf{AUC}}^{(1,1)}$, Its Smoothness and Variance Estimation

The above discussion suggests introducing an analogue to $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)}$ for measuring the performance in AUC. This estimator is motivated from (52a) the same way the estimator $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)}$ was motivated from (31). The SB estimator (52a) can be rewritten as:

$$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{SB} = \frac{1}{n_1 n_2} \sum_{j=1}^{n_2} \sum_{i=1}^{n_1} \mathrm{E}_{*} \psi\left(\hat{h}_{\mathbf{t}^{*}}(x_i), \hat{h}_{\mathbf{t}^{*}}(x_j)\right) \tag{58}$$

$$= \frac{1}{n_1 n_2} \sum_{j=1}^{n_2} \sum_{i=1}^{n_1} \sum_{b=1}^{B} \left[\psi\left(\hat{h}_{\mathbf{t}^{*b}}(x_i), \hat{h}_{\mathbf{t}^{*b}}(x_j)\right) \Big/ B \right]. \tag{59}$$

In words, the procedure is to select a pair (one observation from each class) and calculate for that pair the mean—over many bootstrap replications and training—of the Mann-Whitney kernel. Then, average over all possible pairs. This procedure will be optimistically biased because sometimes the testers will be the same as the trainers. To eliminate that bias, the inner bootstrap expectation should be taken only over those bootstrap replications that do not include the pair (t_i, t_j) in the training. Under that constraint, the estimator (58) becomes the leave-pair-out bootstrap (LPOB) estimator:

$$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(1,1)} = \frac{1}{n_1 n_2} \sum_{j=1}^{n_2} \sum_{i=1}^{n_1} \widehat{\mathrm{AUC}}_{i,j}, \tag{60a}$$

$$\widehat{\mathrm{AUC}}_{i,j} = \sum_{b=1}^{B} I_j^b I_i^b \psi\left(\hat{h}_{\mathbf{t}^{*b}}(x_i), \hat{h}_{\mathbf{t}^{*b}}(x_j)\right) \Big/ \sum_{b'=1}^{B} I_j^{b'} I_i^{b'}. \tag{60b}$$

The two estimators $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}$ and $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(1,1)}$ produce very similar results; this is expected since they both estimate the same thing, i.e., the mean AUC. However, the inner component $\widehat{\mathrm{AUC}}_{i,j}$ of the estimator $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(1,1)}$ also enjoys the smoothness property of $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)}$.

4.3 *Estimating the Standard Error of AUC Estimators*

The only smooth nonparametric estimator for the AUC so far is the LPOB estimator (60). Yousef et al. [33] discusses how to extend the approach of estimating the uncertainty in the error rate estimator using the IF method (Sect. 3.4) to estimate the uncertainty of this estimator, where interested readers may be referred to for all mathematical details and experimental results that show that the IF method provides almost unbiased estimation for the standard error of the LPOB estimator.

5 Illustrative Numerical Examples

5.1 *Error Rate Estimation*

Efron [8] and Efron and Tibshirani [13] provide comparisons of their proposed estimators (discussed in Sect. 3). They ran many simulations considering a variety of

Table 1 Average of RMS error of each estimator over 24 experiments run by Efron and Tibshirani [13]. The estimator $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)}$ is the next to the estimator $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(0.632+)}$ with only 2.5% increase in RMS

Estimator	Average RMS
$\mathrm{Err}_{\mathbf{t}}$	0
$\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)}$	0.083
$\widehat{\mathrm{Err}}_{\mathbf{t}}^{(0.632)}$	0.101
$\widehat{\mathrm{Err}}_{\mathbf{t}}^{(0.632+)}$	0.081
$\overline{\mathrm{Err}}_{\mathbf{t}}$	0.224

classifiers and data distributions, as well as real datasets. They assessed the estimators in terms of the RMS, the root of the experimental MSE:

$$\mathrm{MSE} = \mathrm{E}_{MC}(\widehat{\mathrm{Err}}_{\mathbf{t}} - \mathrm{Err}_{\mathbf{t}})^2 \tag{61a}$$

$$= \frac{1}{G}\sum_{g=1}^{G}(\widehat{\mathrm{Err}}_{\mathbf{t}_g} - \mathrm{Err}_{\mathbf{t}_g})^2, \tag{61b}$$

where $\widehat{\mathrm{Err}}_{\mathbf{t}_g}$ is the estimator (any estimator) conditional on a training dataset $\mathbf{t}_g$, and $\mathrm{Err}_{\mathbf{t}_g}$ is the true prediction error conditional on the same training dataset. The number of MC trials G in their experiments was 200. The following statement is quoted from Efron and Tibshirani [13]:

> The results vary considerably from experiment to experiment, but in terms of RMS error the 0.632+ rule is an overall winner.

This conclusion was without stating the criterion for deciding the *overall winner*. It was apparent from their results that the 0.632+ rule is the winner in terms of the bias—as was designed for. We calculated the average of the RMS of every estimator across all the 24 experiments they ran; Table 1 displays these averages. The estimators $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)}$ and $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(0.632+)}$ are quite comparable to each other with only 2.5% increase in the average RMS of the former. We will show below in Sect. 5.2 that the AUC estimators exhibit the same behavior but with magnified difference between the two estimators.

5.2 AUC Estimation

We carried out different experiments to compare the three bootstrap-based estimators $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}$, $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(.632)}$, and $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(.632+)}$ considering different dimensionalities, different parameter values, and training set sizes. All experiments provided consistent and similar results. Here, in this section, we illustrate the results when the dimensionality $p = 5$, for multinormal 2-class data, with $\Sigma_1 = \Sigma_2 = \mathbf{I}$, $\mu_1 = \mathbf{0}$, $\mu_2 = c\mathbf{1}$, and c is an adjusting parameter to adjust the Mahalanobis distance

Table 2 Comparison of the different bootstrap-based estimators of the AUC. They are comparable to each other in the RMS sense, $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(.632+)}$ is almost unbiased, and all are weakly correlated with the true conditional performance $\mathrm{AUC}_{\mathbf{t}}$

Estimator	Mean	SD	RMS	RMS_{AM}	ρ	Size
$\mathrm{AUC}_{\mathbf{t}}$	0.6181	0.0434	0	0.0434	1.0000	20
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}$	0.5914	0.0947	0.0973	0.0984	0.2553	
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632)}$	0.7012	0.0749	0.1128	0.1119	0.2559	
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632+)}$	0.6431	0.0858	0.0906	0.0894	0.2218	
$\overline{\mathrm{AUC}}_{\mathbf{t}}$	0.8897	0.0475	0.2774	0.2757	0.2231	
$\mathrm{AUC}_{\mathbf{t}}$	0.6231	0.0410	0	0.0410	1.0000	22
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}$	0.5945	0.0947	0.0956	0.0990	0.2993	
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632)}$	0.6991	0.0763	0.1066	0.1077	0.3070	
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632+)}$	0.6459	0.0846	0.0863	0.0876	0.2726	
$\overline{\mathrm{AUC}}_{\mathbf{t}}$	0.8788	0.0499	0.2615	0.2606	0.2991	
$\mathrm{AUC}_{\mathbf{t}}$	0.6308	0.0400	0	0.0400	1.0000	25
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}$	0.5991	0.0865	0.0897	0.0922	0.2946	
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632)}$	0.6971	0.0701	0.0961	0.0965	0.2997	
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632+)}$	0.6442	0.0817	0.0815	0.0828	0.2758	
$\overline{\mathrm{AUC}}_{\mathbf{t}}$	0.8656	0.0471	0.2406	0.2395	0.2833	
$\mathrm{AUC}_{\mathbf{t}}$	0.6359	0.0358	0	0.0358	1.0000	28
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}$	0.6035	0.0840	0.0874	0.0901	0.2904	
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632)}$	0.6962	0.0688	0.0906	0.0915	0.2934	
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632+)}$	0.6479	0.0792	0.0785	0.0802	0.2719	
$\overline{\mathrm{AUC}}_{\mathbf{t}}$	0.8554	0.0472	0.2253	0.2246	0.2747	
$\mathrm{AUC}_{\mathbf{t}}$	0.6469	0.0343	0	0.0343	1.0000	33
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}$	0.6170	0.0750	0.0792	0.0807	0.2746	
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632)}$	0.6997	0.0623	0.0818	0.0817	0.2722	
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632+)}$	0.6553	0.0761	0.0752	0.0766	0.2656	
$\overline{\mathrm{AUC}}_{\mathbf{t}}$	0.8419	0.0439	0.2010	0.1999	0.2434	
$\mathrm{AUC}_{\mathbf{t}}$	0.6571	0.0308	0	0.0308	1.0000	40
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}$	0.6244	0.0711	0.0753	0.0783	0.3185	
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(.632)}$	0.6981	0.0598	0.0710	0.0725	0.3167	
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(.632+)}$	0.6595	0.0739	0.0707	0.0739	0.3092	
$\overline{\mathrm{AUC}}_{\mathbf{t}}$	0.8246	0.0431	0.1735	0.1730	0.2923	
$\mathrm{AUC}_{\mathbf{t}}$	0.6674	0.0271	0	0.0271	1.0000	50
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}$	0.6357	0.0654	0.0690	0.0727	0.3534	
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(.632)}$	0.6995	0.0556	0.0615	0.0642	0.3570	
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(.632+)}$	0.6685	0.0690	0.0646	0.0690	0.3522	
$\overline{\mathrm{AUC}}_{\mathbf{t}}$	0.8091	0.0406	0.1473	0.1474	0.3517	
$\mathrm{AUC}_{\mathbf{t}}$	0.6808	0.0217	0	0.0217	1.0000	66
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}$	0.6533	0.0546	0.0602	0.0611	0.2451	
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(.632)}$	0.7053	0.0471	0.0527	0.0531	0.2488	
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(.632+)}$	0.6840	0.0568	0.0556	0.0569	0.2477	
$\overline{\mathrm{AUC}}_{\mathbf{t}}$	0.7946	0.0355	0.1195	0.1192	0.2499	

(continued)

Table 2 (continued)

Estimator	Mean	SD	RMS	RMS_{AM}	ρ	Size
$\text{AUC}_{\mathbf{t}}$	0.6965	0.0158	0	0.0158	1.0000	100
$\widehat{\text{AUC}}_{\mathbf{t}}^{(*)}$	0.6738	0.0454	0.0483	0.0507	0.3422	
$\widehat{\text{AUC}}_{\mathbf{t}}^{(.632)}$	0.7119	0.0399	0.0405	0.0428	0.3492	
$\widehat{\text{AUC}}_{\mathbf{t}}^{(.632+)}$	0.7004	0.0452	0.0426	0.0453	0.3448	
$\overline{\text{AUC}}_{\mathbf{t}}$	0.7772	0.0312	0.0860	0.0866	0.3596	
$\text{AUC}_{\mathbf{t}}$	0.7141	0.0090	0	0.0090	1.0000	200
$\widehat{\text{AUC}}_{\mathbf{t}}^{(*)}$	0.6991	0.0298	0.0327	0.0334	0.2288	
$\widehat{\text{AUC}}_{\mathbf{t}}^{(.632)}$	0.7205	0.0272	0.0273	0.0279	0.2291	
$\widehat{\text{AUC}}_{\mathbf{t}}^{(.632+)}$	0.7170	0.0285	0.0279	0.0286	0.2294	
$\overline{\text{AUC}}_{\mathbf{t}}$	0.7573	0.0228	0.0487	0.0489	0.2277	

$\Delta = \left[(\mu_1 - \mu_2)'\Sigma^{-1}(\mu_1 - \mu_2)\right]^{1/2} = c^2 p$. We adjust c to keep a reasonable inter-class separation of $\Delta = 0.8$. When the classifier is trained, it will be tested on a pseudo-infinite test set, here 1000 cases per class, to obtain a very good approximation to the true AUC for the classifier trained on this very training dataset; this is called a single realization or a Monte-Carlo (MC) trial. Many realizations of the training datasets with same n are generated over MC simulation to study the mean and variance of the AUC for the Bayes classifier under this training set size. The number of MC trials is 1000 and the number of bootstraps is 100. It is apparent from Fig. 2 that the $\widehat{\text{AUC}}_{\mathbf{t}}^{(*)}$ is downward biased. This is a natural opposite of the upward bias observed in Efron and Tibshirani [13] when the performance measure was the true error rate as a measure of incorrectness, by contrast with the true AUC

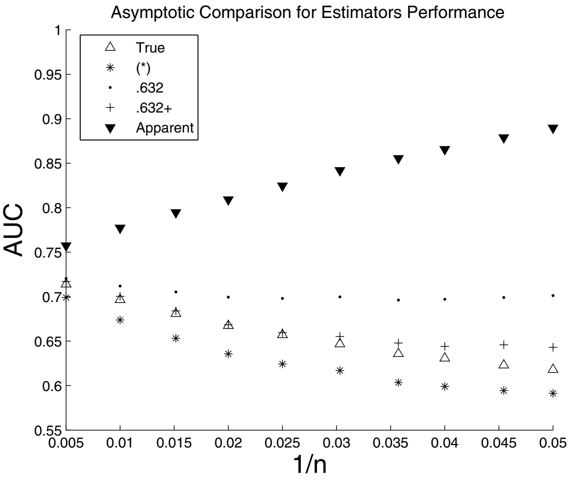

Fig. 2 Comparison of the three bootstrap estimators, $\widehat{\text{AUC}}_{\mathbf{t}}^{(*)}$, $\widehat{\text{AUC}}_{\mathbf{t}}^{(0.632)}$, and $\widehat{\text{AUC}}_{\mathbf{t}}^{(0.632+)}$ for 5-feature predictor. The $\widehat{\text{AUC}}_{\mathbf{t}}^{(*)}$ is downward biased, while the $\widehat{\text{AUC}}_{\mathbf{t}}^{(0.632)}$ is an over correction for that bias. $\widehat{\text{AUC}}_{\mathbf{t}}^{(0.632+)}$ is almost the unbiased version of the $\widehat{\text{AUC}}_{\mathbf{t}}^{(0.632)}$. The figure first appeared in Yousef et al. [32]

Table 3 Average of RMS error of each estimator over the 10 experiments displayed in Table 2. The estimator $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}$ is the next to $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632+)}$ with only 9% increase in RMS

Estimator	Average RMS
$\mathrm{AUC}_{\mathbf{t}}$	0
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}$	0.07347
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632)}$	0.07409
$\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632+)}$	0.06735
$\overline{\mathrm{AUC}}_{\mathbf{t}}$	0.17808

as a measure of correctness. The $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(.632)}$ is designed as a correction for $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}$; it appears in the figure to correct for that but with an over-shoot. The correct adjustment for the remaining bias is almost achieved by the estimator $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(.632+)}$. The $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(.632)}$ estimator can be seen as an attempt to balance between the two extreme biased estimators, $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}$ and $\overline{\mathrm{AUC}}_{\mathbf{t}}$. However, it is expected that the component of $\overline{\mathrm{AUC}}_{\mathbf{t}}$ that is inherent in both $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632+)}$ and $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(0.632)}$ increases the variance of these two estimators that my compensate for the decrease in the bias. Therefore, we assess all estimators in terms of the RMS, the root of the MSE defined in (61), and report the results in Table 2. In addition, we average the RMS of these estimators over the 10 experiments of Table 2 and list the average in Table 3. It is evident that the 0.632+ is slightly the overall winner with only 9% decrease in RMS if compared to the $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}$ estimator. This almost agrees with the same result obtained for the error rate estimators and reported in Table 1.

In addition to the RMS, Table 2 compares the estimators in terms of the RMS around mean (RMS_{AM}): the root of the mean squared difference between an estimate and the mean performance (the mean over all possible training sets), instead of the conditional performance (conditional on a particular training set). The motivation behind that is explained next. The estimators $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(*)}$ and $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(1,1)}$ seem, at least from their formalization, to estimate the mean AUC of the classifier (this is the analogue of $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(*)}$ and $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(1)}$). However, the basic motivation for the $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(.632)}$ and $\widehat{\mathrm{AUC}}_{\mathbf{t}}^{(.632+)}$ is to estimate the AUC conditional on the given dataset $\mathbf{t}$ (this is the analogue of $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(.632)}$ and $\widehat{\mathrm{Err}}_{\mathbf{t}}^{(.632+)}$). Nevertheless, as mentioned in Efron and Tibshirani [13] and detailed in Zhang [35] the CV, the basic ingredient of the bootstrap based estimators, is weakly correlated with the true performance on a sample by sample basis. This means that no estimator has a preference in estimating the conditional performance. Section 5.3 elaborates more on this phenomenon.

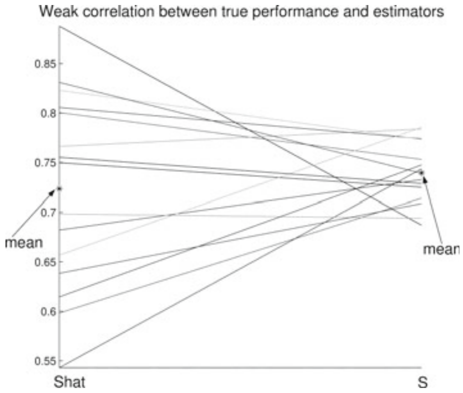

Fig. 3 The lack of correlation (or the weak correlation) between the bootstrap-based estimators and the true conditional performance. Every line connects the true performance of the classifier trained on a data set $\mathbf{t}_i$ and the estimated value. The figure represents 15 trials of the 1000 MC trials. Two nearby values of true performance may correspond to two widely separated estimates on different sides of the mean

5.3 Components of Variance and Weak Correlation

Many simulation results, e.g., Efron [8], Efron and Tibshirani [13], show that there is only a weak correlation between the CV estimator and the conditional true error rate $\text{Err}_{\mathbf{t}}$. This issue is discussed in mathematical detail in the excellent paper by Zhang [35], which therefore concludes that the CV estimator should not be used to estimate the true error rate of a classification rule conditional on a particular training data set. Other estimators discussed in the present article have this same attribute, since they have the same resampling ingredient of the CV estimator and "*we would guess, for any other estimate of conditional prediction error*" (Sect. 7.12, [20]). We provide our simple mathematical elaboration as follows. Denote the true performance of the classification rule conditional on the training set $\mathbf{t}$ (whether $\text{Err}_{\mathbf{t}}$, $\text{AUC}_{\mathbf{t}}$, or any other performance measure) by $S_{\mathbf{t}}$, the unconditional performance by $\text{E}_{\mathbf{t}} S_{\mathbf{t}}$, and an estimator of either of them by $\widehat{S}_{\mathbf{t}}$. For easier notation we can unambiguously drop the subscript $\mathbf{t}$ and decompose the MSE as

$$\text{MSE}(\widehat{S}, S) = \text{E}(\widehat{S} - S)^2 \tag{62a}$$

$$= \text{E}(\widehat{S} - \text{E}S)^2 + \text{Var}(S) - 2\text{Cov}(\widehat{S}, S). \tag{62b}$$

Then, by normalizing with the standard deviations we get:

$$\frac{\text{MSE}(\widehat{S}, S)}{\sigma_S \sigma_{\widehat{S}}} = \frac{\text{MSE}(\widehat{S}, \text{E}S)}{\sigma_S \sigma_{\widehat{S}}} + \frac{\sigma_S}{\sigma_{\widehat{S}}} - 2\rho_{\widehat{S}S}. \tag{63}$$

Table 4 Estimating the uncertainty in the estimator that estimates the difference in performance of two competing classifiers, the LDA and the QDA. The quantity M represents AUC_1 for LDA, AUC_2 for QDA, and $\varDelta$ for the difference

Metric M	LDA	QDA	$\varDelta$
E $M_{\mathbf{t}}$	0.7706	0.7163	0.0543
SD $M_{\mathbf{t}}$	0.0313	0.0442	0.0343
E $\widehat{M}^{(1,1)}$	0.7437	0.6679	0.0758
SD $\widehat{M}^{(1,1)}$	0.0879	0.0944	0.0533
E $\widehat{\mathrm{SD}}\ \widehat{M}^{(1,1)}$	0.0898	0.1003	0.0708
SD $\widehat{\mathrm{SD}}\ \widehat{M}^{(1,1)}$	0.0192	0.0163	0.0228

This equation relates four crucial components to each other:

- $\mathrm{MSE}(\widehat{S}, S)\big/\sigma_S\sigma_{\widehat{S}}$, the normalized MSE of $\widehat{S}$, if we see it as an estimator of the conditional performance S.
- $\mathrm{MSE}(\widehat{S}, \mathrm{E}S)\big/\sigma_S\sigma_{\widehat{S}}$, the normalized MSE of $\widehat{S}$, if we see it as an estimator of the expected performance ES (and therefore called MSE around the mean).
- $\sigma_S\big/\sigma_{\widehat{S}}$, the standard deviation ratio between S and $\widehat{S}$.
- $\rho_{\widehat{S}S}$, the correlation coefficient between S and $\widehat{S}$.

From (63), an estimator $\widehat{S}$ is a good candidate to estimate S than ES if its $\mathrm{MSE}(\widehat{S}, S)$ is less than its $\mathrm{MSE}(\widehat{S}, \mathrm{E}S)$. Then, it is the responsibility of the correlation coefficient $\rho_{\widehat{S}S}$ to be high enough to cancel $\sigma_S\big/\sigma_{\widehat{S}}$ and a portion of $\mathrm{MSE}(\widehat{S}, \mathrm{E}S)$. Unfortunately, this is not the case as we illustrate experimentally in Table 2, which provides all quantities of the decomposition (63). It is obvious from the values that $\mathrm{RMS}(\widehat{S}, S)$ and $\mathrm{RMS}(\widehat{S}, \mathrm{E}S)$ are very close to each other because the quantity $\sigma_S\big/\sigma_{\widehat{S}} - 2\rho_{\widehat{S}S} \simeq 0.413 - 2 \times 0.290 = -0.167$ (on average over the 10 experiments shown in the table). Moreover, in some cases, e.g., the first experiment, it goes as low as -0.052. The correlation between $\widehat{S}$ and S is weak to cast $\widehat{S}$ as an estimate to S, although it is designed to estimate it! For more illustration, Fig. 3 visualizes the components in Eq. (63) and the numbers in Table 2. This figure shows 15 realizations of the 1000 MC trials of the same experiment above. On the right, are the true values of S when trained on these different 15 training sets. On the left, are the corresponding 15 estimated values of $\widehat{S}$. The lines provide links between the true values and the corresponding estimates. This figure shows that two nearby true values of S are likely to have two widely separated estimated values $\widehat{S}$ on different sides of the mean. This visually illustrates the lack of correlation (or the weak correlation) between the estimators and the true conditional performance.

5.4 Two Competing Classifiers

If the assessment problem is how to compare two classifiers, rather than the individual performance, then the measure to be used is either the conditional difference

$$\varDelta_{\mathbf{t}} = \mathrm{AUC}_{1_{\mathrm{t}}} - \mathrm{AUC}_{2_{\mathrm{t}}}, \tag{64}$$

or the mean, unconditional, difference

$$\varDelta = \mathrm{E}\,\varDelta_{\mathbf{t}} = \mathrm{E}\left[\mathrm{AUC}_{1_{\mathrm{t}}} - \mathrm{AUC}_{2_{\mathrm{t}}}\right], \tag{65}$$

where, we defined them for the AUC just for illustration with immediate identical treatment for other measures. Then it is obvious that there is nothing new in the estimation task, i.e., it is merely the difference of the performance estimate of each classifier, i.e.,

$$\widehat{\varDelta} = \widehat{\mathrm{E\,AUC}}_{1_{\mathrm{t}}} - \widehat{\mathrm{E\,AUC}}_{2_{\mathrm{t}}}, \tag{66}$$

where each of the two estimators in (66) is obtained by any estimator. A natural candidate, from the point of view of the present chapter is the LPOB estimator $\widehat{\mathrm{AUC}}^{(1,1)}$—because of both the smoothness and weak correlation issues discussed so far.

Then, how to estimate the uncertainty (variance) of $\widehat{\varDelta}$. This is very similar to estimating the variance in $\widehat{\mathrm{E\,AUC}}_{\mathbf{t}}$. There is nothing new in estimating Var $\widehat{\varDelta}$. It is obtained by replacing $\widehat{\mathrm{AUC}}^{(1,1)}$, in Yousef et al. [33], by the statistic $\widehat{\varDelta}$ in (66). For demonstration, typical values are given in Table 4, for comparing the linear and quadratic discriminants, where the training set size per class is 20 and number of features is 4.

6 Discussion and Conclusion

In this chapter, the very important topic of the assessment of ML algorithms is reviewed, with an emphasis on the nonparametric assessment of classification rules. The topic is quite important to many fields and applications, in particular cyberphysical security, where ML algorithms are almost ubiquitous. We started with reviewing the basic nonparametric methods for estimating the bias and variance of a statistic. Then, we reviewed the basic resampling-based methods for estimating the error rate of a classification rule. Departing from that, we extended these estimators from estimating the error rate (a one-sample statistic) to estimating the AUC (a two-sample statistic). This extension is theoretically justified, and not just an ad hoc application. Among these estimators, we identified those that are smooth and eligible for estimating their standard error using the IF method.

It was interesting to see, through the whole chapter, the connection among different resampling-based estimators. It is worth mentioning that, in addition to the conventional K-fold CV, there are other versions and variants, which are usually used in an ad hoc way by many practitioners. The formalization of these versions and variants, and the mathematical connection among them, along with their connection to the bootstrap-based estimators, all can be established in the same spirit and approach followed in the present chapter. However, many of them are unsmooth except possibly the repeated CV, which is partially smooth and suitable for the IF method [30, 31].

With this rich variety of estimators, a practitioner may legitimately wonder about the "optimal" estimator (in terms of any optimality criterion) that should be systematically used. There are three aspects, on which we can base our comparison: accuracy, uncertainty estimation, and computational efficiency.

In terms of accuracy, it is surprising to know that, from the few number of comparative studies available in the literature, there is no overall winner among these estimators. All of them have comparable accuracy, measured in terms of RMS, with a little superiority of the 0.632+ bootstrap estimator. In addition, and most importantly, all estimators have a weak correlation with the true conditional performance (e.g., $\mathrm{Err}_{\mathbf{t}}$, the conditional error rate, or $\mathrm{AUC}_{\mathbf{t}}$, the conditional AUC), a phenomenon that allows them to be eligible only for estimating the mean true performance (e.g., $\mathrm{E}_{\mathbf{t}}\mathrm{Err}_{\mathbf{t}}$ or $\mathrm{E}_{\mathbf{t}}\mathrm{AUC}_{\mathbf{t}}$), where the mean is taken over the population of training datasets as explained through the chapter. Said differently, the performance estimation that a practitioner obtains using, e.g., the CV, is not an estimation of the performance of this very trained ML algorithm; rather, it is an estimation of the mean performance of this algorithm had we trained it on all possible training datasets of the same size! We quote from [20, Sect. 7.12]:

> *This phenomenon also occurs for bootstrap estimates of error, and we would guess, for any other estimate of conditional prediction error.*

In terms of the variance estimation of these estimators (not the estimation of the variance of the algorithm itself), only a few of them are smooth and candidates for a sophisticated method like the IF. The ordinary K-fold CV is not among those! Rather, only the computationally expensive version of it, the repeated CV, is partially smooth as mentioned above.

It terms of the computational aspects, the bootstrap-based estimators are computationally expensive. If compared to the conventional K-fold CV, which requires only K iterations of both training and testing, the former require hundreds of bootstrap replications. Because the majority of recent ML applications involve both massive datasets and complex algorithms, including DNN that is very computationally expensive, it is obvious that the CV may be more practical than the bootstrap-based estimators. However, for some other fields, e.g., cyberphysical security, many applications produce tabular (structured) data. Tabular data are more suitable for the traditional and less computationally expensive ML algorithms. Therefore, serious practitioners in these fields and applications may need to keep all of these estimators in their toolbox. Moreover, it is quite prudent to see a future benchmark that compiles these estimators, along with different datasets from a wide range of applications, in a single comprehensive comparative study.

Acknowledgements The author is grateful to the U.S. Food and Drug Administration (FDA) for funding a very early stage of this chapter, and to Dr. Kyle Myers for her support. In his memorial, special thanks and gratitude to Dr. Robert F. Wagner, the supervisor and the teacher, or Bob Wagner, the big brother and friend. He reviewed a very early version of this chapter before he passed away.

7 Appendix

7.1 Proofs

Lemma 1 *The maximum likelihood estimation (MLE) for the probability mass function under nonparametric distribution, given a sample of n observations, is given by:*

$$\hat{F} : \; mass \; \frac{1}{n} \; on \; t_i, \quad i = 1, \ldots, n. \tag{67}$$

Proof The proof is carried out by maximizing the likelihood function $l(f) = \prod_{i=1}^{n} p_i$, which can be rewritten under the constraint $\sum_i p_i = 1$, using a Lagrange's multiplier, as:

$$l(f) = \prod_{i=1}^{n} p_i + \lambda \left(\sum_{i=1}^{n} p_i - 1 \right). \tag{68}$$

The likelihood (68) is maximized by taking the first derivative and setting it to zero to obtain:

$$\frac{\partial l(f)}{\partial p_j} = \prod_{i \neq j} p_i + \lambda \overset{set}{=} 0, \quad j = 1, \ldots, n. \tag{69}$$

These n equations along with the constraint $\sum_i p_i = 1$ can be solved straightforwardly to give $\hat{p}_i = \frac{1}{n}, \; i = 1, \ldots, n$, which completes the proof. □

Lemma 2 *The no-information* AUC *is given by* $\gamma_{\mathrm{AUC}} = 0.5$.

Proof γ_{AUC}, an analogue to the *no-information error rate* γ, is given by (2a) but with TPF and FPF given under the *no-information* distribution E_{0F} (see Sect. 3.3.4). Therefore, assume that there are n_1 and n_2 observations from class ω_1 and ω_2, respectively. Assume also for a fixed threshold th the two quantities that define the error rate are TPF and FPF. Also, assume that the sample observations are tested by the classifier and each sample has been assigned a decision value (score). Under the *no-information* distribution, consider the following. For every decision value $h_{\mathbf{t}}(x_i)$ assigned for the observation $t_i = (x_i, y_i)$, create new $n_1 + n_2 - 1$ observations; all of them have the same decision value $h_{\mathbf{t}}(x_i)$, while their responses are equal to the responses of the rest $n_1 + n_2 - 1$ observations $t_j,\ j \neq i$. Under this new sample that consists of $(n_1 + n_2)^2$ observations, it is quite easy to see that the new TPF and FPF for the same threshold th are given by $\text{FPF}_{0\widehat{F},th} = \text{TPF}_{0\widehat{F},th} = (\text{TPF} \cdot n_1 + \text{FPF} \cdot n_2)/(n_1 + n_2)$. This means that the ROC curve under the *no-information* rate is a straight line with slope equal to one; this directly gives $\gamma_{\text{AUC}} = 0.5$.

7.2 More on Influence Function (IF)

Assume that there is a distribution G near to the distribution F; then under some regularity conditions(see, e.g., [21], Chap. 2) a functional s can be approximated as:

$$s(G) \approx s(F) + \int IC_{s,F}(x)\, dG(x). \tag{70}$$

The residual error can be neglected since it is of a small order in probability. Some properties of (70) are:

$$\int IC_{T,F}(x)\, dF(x) = 0, \tag{71}$$

and the asymptotic variance of $s(F)$ under F, following from (71), is given by:

$$\text{Var}_F s(F) \simeq \int \left[IC_{T,F}(x)\right]^2\, dF(x), \tag{72}$$

which can be considered as an approximation to the variance under a distribution G near to F. Now, assume that the functional s is a functional statistic in the dataset $\mathbf{x} = \{x_i : x_i \sim F,\ i = 1, \ldots, n\}$. In that case the influence curve (23) is defined for each sample case x_i, under the true distribution F as:

$$U_i(s, F) = \lim_{\varepsilon \to 0} \frac{s(F_{\varepsilon,i}) - s(F)}{\varepsilon} = \left.\frac{\partial s(F_{\varepsilon,i})}{\partial \varepsilon}\right|_{\varepsilon=0}, \tag{73}$$

where $F_{\varepsilon,i}$ is the distribution under the perturbation at observation x_i. Equation (73) is called the IF. If the distribution F is not known, the MLE $\hat{F}$ of the distribution F is given by (3), and as an approximation $\hat{F}$ may substitute for F in (73). The result may then be called the empirical IF [24], or infinitesimal jackknife [22]. In such an approximation, the perturbation defined in (22) can be rewritten as:

$$\hat{F}_{\varepsilon,i} = (1 - \varepsilon)\hat{F} + \varepsilon\delta_{x_i},\quad x_i \in \mathbf{x},\quad i = 1, \ldots, n. \tag{74}$$

This kind of perturbation is illustrated in Fig. 4.

It will often be useful to write the probability mass function of (74) as:

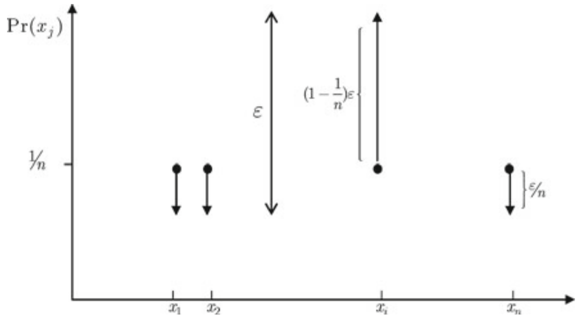

Fig. 4 The new probability masses for the dataset **x** under a perturbation at sample case x_i obtained by letting the new probability, at x_i exceed the new probability at any other case x_i by, ε

$$\hat{f}_{\varepsilon,i}(x_j) = \begin{cases} \frac{1-\varepsilon}{n} + \varepsilon & j = i \\ \frac{1-\varepsilon}{n} & j \neq i \end{cases}. \tag{75}$$

A very interesting case arises from (75) if $-1/(n+1)$ is substituted for ε. In this case the new probability mass assigned to the point $x_{j=i}$ in (75) will be zero. This value of ε simply generates the jackknife estimate discussed in Sect. 2.2, where the whole observation is removed from the dataset.

Substituting $\hat{F}$ for G in (70) and combining the result with (73) gives the IF approximation for any functional statistic under the empirical distribution $\hat{F}$. The result is:

$$s(\hat{F}) = s(F) + \frac{1}{n}\sum_{i=1}^{n} U_i(s, F) + O_p(n^{-1}) \tag{76a}$$

$$\approx s(F) + \frac{1}{n}\sum_{i=1}^{n} U_i(s, F). \tag{76b}$$

The term $O_p(n^{-1})$ reads "big-O of order $1/n$ in probability". In general, $U_n = O_p(d_n)$ if U_n/d_n is bounded in probability, i.e., $\Pr\{|U_n|/d_n < k_\varepsilon\} > 1 - \varepsilon \ \forall\ \varepsilon > 0$. This concept can be found in [1, Chap. 2]. Then the asymptotic variance expressed in (72) can be given for $s(F)$ by:

$$\mathrm{Var}_F s = \frac{1}{n}\mathrm{E}_F U^2(x_i, F), \tag{77}$$

which can be approximated under the empirical distribution $\hat{F}$ to give the nonparametric estimate of the variance for a statistic s by:

$$\widehat{\mathrm{Var}}_{\hat{F}} s = \frac{1}{n^2}\sum_{i=1}^{n} U_i^2(x_i, \hat{F}). \tag{78}$$

7.3 ML in Other Fields

In this section we provide very brief miscellanea from other fields for the reader to see a bigger picture of this chapter. As already was mentioned, ML is crucial to many applications. For example, in the medical imaging field, a tumor on a mammogram must be classified as malignant or benign. This is an example of prediction, regardless of whether it is done by a radiologist or by a computer aided detection (CAD) software. In either case, the prediction is done based on learning from previous mammograms. The features, i.e., predictors, in this case may be the size of the tumor, its density, various shape parameters, etc. The output, i.e., response, is categorical and belongs to the set: $\mathcal{G} = \{benign,\ malignant\}$. There are so many such examples in biology and medicine that it is almost a field unto itself, i.e., biostatistics. The task may be diagnostic as in the mammographic example, or prognostic where, for example, one estimates the probability of occurrence of a second heart attack for a particular patient who has had a previous one. All of these examples involve a prediction step based on previous learning. A wide range of commercial and military applications arises in the field of satellite imaging. Predictors in this case can be measures from the image spectrum, while the response can be the type of land, crop, or vegetation of which the image was taken.

Some expressions and terminology of ML belong to some fields and applications more than the others. E.g., it is conventional in medical imaging to refer to e_1 as the false negative fraction (FNF), and e_2 as the false positive fraction (FPF). This is because diseased patients typically have a higher output value for a test than non-diseased patients. For example, a patient belonging to class 1 whose test output value is less than the threshold setting for the test will be called “test negative”, while the patient is in fact in the diseased class. This is a false negative decision; hence the name FNF. The situation is reversed for the other error component.

The importance of the AUC is natural and unquestionable in some applications than others. The equivalence of the area under the empirical ROC and the Mann-Whitney-Wilcoxon statistic is the basis of its use in the assessment of diagnostic tests; see Hanley and McNeil [19]. Swets [29] has recommended it as a natural summary measure of detection accuracy on the basis of signal-detection theory. Applications of this measure are widespread in the literature on both human diagnosis and computer-aided diagnosis, in medical imaging [23]. In the field of machine learning, Bradley [2] has recommended it as the preferred summary measure of accuracy when a single number is desired. These references also provide general background and access to the large literature on the subject.

Even the mistakes committed by some practitioners are obvious in some fields more than others. E.g., in DNA microarrays, these mistakes are fatal and produce very fragile results. This is because of the very high dimensionality of the problem with respect to the amount of available dataset. A more elaborate assessment phase should follow the design and construction phase in such ill-posed applications.

References

1. Barndorff-Nielsen OE, Cox DR (1989) Asymptotic techniques for use in statistics. Chapman and Hall, New York
2. Bradley AP (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recogn 30(7):1145
3. Breiman L, Friedman J, Olshen R, Stone C (1984) Classification and regression trees. Wadsworth International Group, Belmont
4. Chen W, Gallas BD, Yousef WA (2012) Classifier variability: accounting for training and testing. Pattern Recogn 45(7):2661–2671
5. Efron B (1979) Bootstrap methods: another look at the Jackknife. Ann Stat 7(1):1–26
6. Efron B (1981) Nonparametric estimates of standard error: the Jackknife, the bootstrap and other methods. Biometrika 68(3):589–599
7. Efron B (1982) The Jackknife, the bootstrap, and other resampling plans. Society for Industrial and Applied Mathematics, Philadelphia
8. Efron B (1983) Estimating the error rate of a prediction rule: improvement on cross-validation. J Am Stat Assoc 78(382):316–331
9. Efron B (1986) How biased is the apparent error rate of a prediction rule? J Am Stat Assoc 81(394):461–470
10. Efron B, Stein C (1981) The Jackknife estimate of variance. Ann Stat 9(3):586–596
11. Efron B, Tibshirani R (1993) An introduction to the bootstrap. Chapman and Hall, New York
12. Efron B, Tibshirani R (1995) Cross validation and the bootstrap: estimating the error rate of a prediction rule. Technical report 176, Stanford University, Department of Statistics
13. Efron B, Tibshirani R (1997) Improvements on cross-validation: the .632+ Bootstrap method. J Am Stat Assoc 92(438):548–560
14. Fukunaga K (1990) Introduction to statistical pattern recognition, 2nd edn. Academic Press, Boston
15. Hájek J, Šidák Z, Sen PK (1999) Theory of rank tests, 2nd edn. Academic Press, San Diego
16. Hampel FR (1974) The influence curve and its role in robust estimation. J Am Stat Assoc 69(346):383–393
17. Hampel FR (1986) Robust statistics?: the approach based on influence functions. Wiley, New York
18. Hanley JA (1989) Receiver operating characteristic (ROC) methodology: the state of the art. Crit Rev Diagn Imaging 29(3):307–335
19. Hanley JA, McNeil BJ (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143(1):29–36
20. Hastie T, Tibshirani R, Friedman JH (2009) The elements of statistical learning: data mining, inference, and prediction, 2nd edn. Springer, New York
21. Huber PJ (1996) Robust statistical procedures, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia
22. Jaeckel L (1972) The infinitesimal jackknife. Memorandum, MM 72-1215-11, Bell Lab Murray Hill
23. Jiang Y, Nishikawa RM, Schmidt RA, Metz CE, Giger ML, Doi K (1999) Improving breast cancer diagnosis with computer-aided diagnosis. Acad Radiol 6(1):22–33
24. Mallows C (1974) On some topics in robustness. Memorandum, MM 72-1215-11, Bell Lab Murray Hill, NJ
25. Randles RH, Wolfe DA (1979) Introduction to the theory of nonparametric statistics. Wiley, New York
26. Sahiner B, Chan HP, Petrick N, Hadjiiski L, Paquerault S, Gurcan MN (2001) Resampling schemes for estimating the accuracy of a classifier designed with a limited data set. In: Medical image perception conference IX, airlie conference Center, Warrenton VA, 20–23
27. Sahiner B, Chan HP, Hadjiiski L (2008) Classifier performance prediction for computer-aided diagnosis using a limited dataset. Med Phys 35(4):1559

28. Stone M (1974) Cross-validatory choice and assessment of statistical predictions. J Roy Stat Soc: Ser B (Methodol) 36(2):111–147
29. Swets JA (1986) Indices of discrimination or diagnostic accuracy: their ROCs and implied models. Psychol Bull 99:100–117
30. Yousef WA (2019) A leisurely look at versions and variants of the cross validation estimator. arXiv preprint arXiv:1907.13413
31. Yousef WA (2021) Estimating the standard error of cross-validation-based estimators of classifier performance. Pattern Recogn Lett 146:115–145
32. Yousef WA, Wagner RF, Loew MH (2004) Comparison of non-parametric methods for assessing classifier performance in terms of ROC parameters. In: Proceedings of 33rd applied imagery pattern recognition workshop, 2004. IEEE Computer Society, pp 190–195
33. Yousef WA, Wagner RF, Loew MH (2005) Estimating the uncertainty in the estimated mean area under the ROC curve of a classifier. Pattern Recogn Lett 26(16):2600–2610
34. Yousef WA, Wagner RF, Loew MH (2006) Assessing classifiers from two independent data sets using ROC analysis: a nonparametric approach. IEEE Trans Pattern Anal Mach Intell 28(11):1809–1817
35. Zhang P (1995) Assessing prediction error in nonparametric regression. Scand J Stat 22(1):83–94