RESEARCH METHODS

Neyman-Pearson classification algorithms and NP receiver operating characteristics

Xin Tong,[1]*† Yang Feng,[2]† Jingyi Jessica Li[3]*

In many binary classification applications, such as disease diagnosis and spam detection, practitioners commonly face the need to limit type I error (that is, the conditional probability of misclassifying a class 0 observation as class 1) so that it remains below a desired threshold. To address this need, the Neyman-Pearson (NP) classification paradigm is a natural choice; it minimizes type II error (that is, the conditional probability of misclassifying a class 1 observation as class 0) while enforcing an upper bound, α, on the type I error. Despite its century-long history in hypothesis testing, the NP paradigm has not been well recognized and implemented in classification schemes. Common practices that directly limit the empirical type I error to no more than α do not satisfy the type I error control objective because the resulting classifiers are likely to have type I errors much larger than α, and the NP paradigm has not been properly implemented in practice. We develop the first umbrella algorithm that implements the NP paradigm for all scoring-type classification methods, such as logistic regression, support vector machines, and random forests. Powered by this algorithm, we propose a novel graphical tool for NP classification methods: NP receiver operating characteristic (NP-ROC) bands motivated by the popular ROC curves. NP-ROC bands will help choose α in a data-adaptive way and compare different NP classifiers. We demonstrate the use and properties of the NP umbrella algorithm and NP-ROC bands, available in the R package nproc, through simulation and real data studies.

INTRODUCTION

In statistics and machine learning, the purpose of classification is to automatically predict discrete outcomes (that is, class labels) for new observations on the basis of labeled training data. The development of classification theory, methods, and applications has been a dynamic area of research for more than half a century (*1*). Well-known examples include disease diagnosis, email spam filters, and image classification. Binary classification, in which the class labels are 0 and 1, is the most common type. Most binary classifiers are constructed to minimize the expected classification error (that is, risk), which is a weighted sum of type I and II errors. We refer to this paradigm as the classical classification paradigm in this paper. Type I error is defined as the conditional probability of misclassifying a class 0 observation as a class 1 observation; it is also called the false-positive rate (that is, 1 − specificity). Type II error is the conditional probability of misclassifying a class 1 observation as class 0; it is also called the false-negative rate (that is, 1 − sensitivity). Note that in some specific scientific or medical contexts, positiveness and negativeness have strict definitions, and their definition of "false-positive rate" may be different from the probability of classifying an observation whose true label is 0 as class 1. Because the binary class labels can be arbitrarily defined and switched, in the following text, we refer to the prioritized type of error as the type I error.

In the risk, the weights of the type I and II errors are the marginal probabilities of classes 0 and 1, respectively. In real-world applications, however, users' priorities for type I and II errors may differ from these weights. For example, in cancer diagnosis, a type I error (that is, misdiagnosing a cancer patient as healthy) has more severe consequences than a type II error (that is, misdiagnosing a healthy patient with cancer); the latter may lead to extra medical costs and patient anxiety but will not result in tragic loss of life (*2*–*5*). For these applications, a prioritized control of asymmetric classification errors is sorely needed. The Neyman-Pearson (NP) classification paradigm was developed for this purpose (*6*–*9*); it seeks a classifier that minimizes the type II error while maintaining the type I error below a user-specified level α, usually a small value (for example, 5%). In statistical learning theory, this target classifier is called the oracle NP classifier; it achieves the minimum type II error given an upper bound α on the type I error. It is different from the cost-sensitive learning (*10*, *11*) and the asymmetric support vector machines (*12*), which also address asymmetric classification errors but provide no probabilistic control on the errors. Previous studies addressed NP classification using both empirical risk minimization (*6*–*8*, *13*–*15*) and plug-in approaches (*9*, *16*). For a review of the current status of NP classification, we refer the readers to the study of Tong *et al.* (*17*). A main advantage of the NP classification is that it is a general framework that allows users to find classifiers with (population) type I errors under α with high probability. In many biomedical, engineering, and social applications, users often have prespecified α values to reflect their tolerance on the type I errors and use diverse classification algorithms. Example applications include diagnosis of coronary artery disease (*18*), cancer early warning system (*19*), network security control (*20*, *21*), Environmental Protection Agency water security research (*22*), prediction of regional and international conflicts (*23*, *24*), and the Early Warning Project to identify countries at risk of new mass atrocities (*25*). However, existing ad hoc use of classification algorithms cannot control type I errors under α with high probability. Although the NP paradigm can address the needs, how to implement it with diverse classification algorithms remains a challenge.

Here, we address two important but unanswered questions regarding the practicality of the NP classification paradigm and the evaluation of NP classification methods. The first question is how to adapt popular classification methods [for example, logistic regression (LR) (*26*), support vector machines (SVMs) (*27*), AdaBoost (*28*), and random forests (RFs) (*29*)] to construct NP classifiers. We address this question by proposing an umbrella algorithm to implement a broad class of classification methods under the NP paradigm. The second question is how to evaluate and compare the performance of different NP classification

[1]Department of Data Sciences and Operations, Marshall School of Business, University of Southern California, Los Angeles, CA 90089, USA. [2]Department of Statistics, Columbia University, New York, NY 10027–5927, USA. [3]Department of Statistics, University of California, Los Angeles, CA 90095–1554, USA.
†These authors contributed equally to this work.
*Corresponding author. Email: xint@marshall.usc.edu (X.T.); jli@stat.ucla.edu (J.J.L.)

methods. We propose NP receiver operating characteristic (NP-ROC) bands, a variant of ROC, as a new visualization tool for NP classification. Possible uses of NP-ROC bands for practitioners include but are not limited to (i) choosing α in a data-adaptive way and (ii) comparing different NP classifiers. The NP umbrella algorithm and NP-ROC bands together provide a flexible pipeline to implement binary classification in broad applications where controlling the prioritized type of error is needed, such as disease diagnosis in biomedical applications (*18*, *19*) and loan screening in financial applications (*30*). Other potential applications are described in detail in the Discussion section.

RESULTS

Mathematical formulation

To facilitate our discussion of technical details, we introduce the following mathematical notation and review our previous theoretic formulation (*8*, *9*, *16*) to further explain the classical and NP classification paradigms. Let (X, Y) be random variables where $X \in \mathcal{X} \subset \mathbb{R}^d$ is a vector of d features, and $Y \in \{0, 1\}$ represents a binary class label. A data set that contains independent observations $\{(x_i, y_i)\}_{i=1}^n$ sampled from the joint distribution of (X, Y) is often divided into training data and test data. On the basis of training data, a classifier $\phi(\cdot)$ is a mapping $\phi : \mathcal{X} \to \{0, 1\}$ that returns the predicted class label given X. Classification errors occur when $\phi(X) \neq Y$ and the binary loss is defined as $I(\phi(X) \neq Y)$, where $I(\cdot)$ denotes the indicator function. The risk is defined as $R(\phi) = \mathbb{E}[I(\phi(X) \neq Y)] = \mathbb{P}(\phi(X) \neq Y)$, which can be expressed as a weighted sum of type I and II errors: $R(\phi) = \mathbb{P}(Y = 0)R_0(\phi) + \mathbb{P}(Y = 1)R_1(\phi)$, where $R_0(\phi) = \mathbb{P}(\phi(X) \neq Y | Y = 0)$ denotes the (population) type I error, and $R_1(\phi) = \mathbb{P}(\phi(X) \neq Y | Y = 1)$ denotes the (population) type II error. The classical classification paradigm aims to mimic the classical oracle classifier ϕ^* that minimizes the risk

$$\phi^* = \operatorname{argmin}_\phi R(\phi).$$

In contrast, the NP classification paradigm aims to mimic the NP oracle classifier ϕ_α^* with respect to a prespecified upper bound on the type I error, α

$$\phi_\alpha^* = \operatorname{argmin}_{\phi: R_0(\alpha) \leq \alpha} R_1(\phi),$$

where α reflects users' conservative attitude (that is, priority) toward type I error. Figure 1 shows a toy example that demonstrates the difference between a classical oracle classifier that minimizes the risk and an NP oracle classifier that minimizes the type II error, given that type I error ≤ α = 0.05.

In practice, $R(\cdot)$, $R_0(\cdot)$, and $R_1(\cdot)$ are unobservable because they depend on the unknown joint distribution of (X, Y). Instead, their estimates based on data (that is, the empirical risk, empirical type I error, and empirical type II error) are often used in practice. Here, we denote the empirical risk and type I and II errors based on training data as $\hat{R}(\cdot)$, $\hat{R}_0(\cdot)$, and $\hat{R}_1(\cdot)$ and the empirical risk and type I and II errors based on test data as $\tilde{R}(\cdot)$, $\tilde{R}_0(\cdot)$, and $\tilde{R}_1(\cdot)$.

Because of the wide applications of classification in real-world problems, a vast array of classification methods have been developed to construct "good" binary classifiers. Here, we focus on the scoring type of classification methods, which first train a scoring function $f: \mathcal{X} \to \mathbb{R}$ using the training data. The scoring function $f(\cdot)$ assigns a classification score $f(x)$ to an observation $x \in \mathbb{R}^d$. By setting a threshold $c \in \mathbb{R}$ on the classification scores, a classifier can be obtained. In other words, we consider classifiers with the form $\phi^c(\cdot) = I(f(\cdot) > c)$. Most popular classification methods are of this type (*31*). For example, LR, SVMs, naïve Bayes (NB), and neural networks all output a numeric value (that is, a classification score) to represent the degree to which a test data point belongs to class 1. The classification scores can be strict probabilities or uncalibrated numeric values as long as a higher score indicates a higher probability of an observation belonging to class 1. Another type of classification algorithms (for example, RFs) uses bagging to generate an ensemble of classifiers, each of which predicts a class label for a test data point; in these scenarios, the proportion of predicted labels being 1 serves as a classification score.

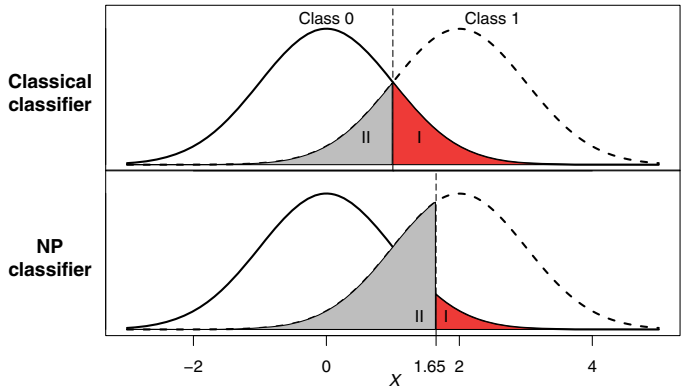

Fig. 1. Classical versus NP oracle classifiers in a binary classification example. In this toy example, the two classes have equal marginal probabilities, that is, $\mathbb{P}(Y = 0) = \mathbb{P}(Y = 1) = 0.5$. Suppose that a user prefers a type I error of ≤0.05. The classical classifier $I(X > 1)$ that minimizes the risk would result in a type I error of 0.16. On the other hand, the NP classifier $I(X > 1.65)$ that minimizes the type II error under the type I error constraint (α = 0.05) delivers a desirable type I error. This figure is adapted from Tong *et al.* (*17*) and Li and Tong (*52*), with permissions.

An umbrella algorithm for NP classification

Recent studies describe several NP classifiers that respect the type I error bound with high probability. They are built upon plug-in estimators of density ratios (that is, class 1 density/class 0 density) (*9*, *16*). However, these classifiers are only applicable under a number of restricted scenarios, such as low feature dimension (*9*) and feature independence (*16*). Many other statistical and machine learning algorithms have been shown to be effective classification methods but not yet implemented under the NP paradigm; these include but are not limited to (penalized) LR, SVMs, AdaBoost, and RFs. To develop NP classifiers that are diverse and adaptable to various practical scenarios and avoid duplicating efforts, we choose to implement these popular classification algorithms under the NP paradigm rather than construct numerous new NP classifiers based on complex models for density ratios. Moreover, classifiers in the study of Tong (*9*) use the Vapnik-Chervonenkis theory to guide an upper bound for empirical type I error and require a sample size larger than available in many modern applications, whereas classifiers in the study of Zhao *et al.* (*16*) resort to concentration inequalities to get an explicit order at the expense of possible overly conservatism in type I errors. Here, we develop an alternative approach by calculating exact probabilities (under mild continuity assumptions) based on order statistic distributions to find thresholds on classification scores under the NP paradigm.

The first main contribution of this paper is our proposed umbrella algorithm that adapts popular classification methods to the NP

paradigm. These methods include LR and penalized LR (penLR), SVMs, linear discriminant analysis (LDA), NB, AdaBoost, classification trees, and RFs. Specifically, we seek an efficient way to choose a threshold for the classification scores predicted by each algorithm so that the threshold leads to classifiers with type I errors below the user-specified upper bound α with high probability. This algorithm is needed because the naïve approach, which simply picks a threshold by setting the empirical type I error to no more than α, fails to satisfy the type I error constraint, as demonstrated in the simulation study described below.

Simulation 1

Data are generated from two Gaussian distributions: $(X|Y=0) \sim N(0,1)$ and $(X|Y=1) \sim N(2,1)$, with $\mathbb{P}(Y=0)=\mathbb{P}(Y=1)=0.5$. We denote a data set from this distribution as $\{(x_i, y_i)\}_{i=1}^{N}$, where $N = 1000$. The classifiers we consider are $I(X > c)$, where $c \in \mathbb{R}$. Here, the classification scoring function $f(x) = x$, the identity function, because in this one-dimensional case, x_i's naturally serve as good classification scores. Hence, no training for f is needed, and we only rely on this data set to find c such that the corresponding classifier has the population type I error below α with high probability. Figure 2 illustrates this problem. The black curves denote the oracle ROC curves, which trace the population type I error and (1 − population type II error) of these classifiers as c varies. To find a value of c such that the corresponding classifier has type I error ≤ α = 0.05, a common and intuitive practice is to choose the smallest c such that the empirical type I error is no greater than α, resulting in a classifier $\bar{\phi}_\alpha(\cdot) = I(\cdot > \bar{c}_\alpha)$, where $\bar{c}_\alpha = \inf\left\{c : \frac{\sum_{i=1}^{N} I(x_i > c, y_i = 0)}{\sum_{i=1}^{N} I(y_i = 0)} \le \alpha\right\}$. In our simulation, we generate $D = 1000$ data sets; this procedure results in D classifiers, $\bar{\phi}_\alpha^{(1)}, \cdots, \bar{\phi}_\alpha^{(D)}$, shown as red "×" on the oracle ROC curve (Fig. 2A). However, only approximately half of these classifiers have type I errors below α, which is far from achieving the users' goal. Therefore, this commonly used method does not work well in this case. We also use another approach based on the fivefold cross-validation (CV), although in this case, no training is needed to estimate the type I error of the classifier $I(X > c)$ for every $c \in \mathbb{R}$. Concretely, we randomly split the class 0 data into fivefolds. On each fold, we calculate the empirical type I error of the classifier, and we take the average of the five empirical type I errors as the CV type I error, denoted by $\hat{R}_0^{\mathrm{CV}}(c)$. On the basis of these CV type I errors of various c, we can construct a classifier $\tilde{\phi}_\alpha(\cdot) = I(\cdot > \tilde{c}_\alpha)$, where $\tilde{c}_\alpha = \inf\left\{c : \hat{R}_0^{\mathrm{CV}}(c) \le \alpha\right\}$. Applying this procedure to the $D = 1000$ data sets, we obtain D classifiers, $\tilde{\phi}_\alpha(1), \cdots, \tilde{\phi}_\alpha(D)$, shown as cyan "×" on the oracle ROC curve (Fig. 2B). However, still only about half of these classifiers have type I errors below α. Therefore, this CV-based approach still does not work well in this case.

In view of this failure, we propose an umbrella algorithm to implement the NP paradigm, summarized as pseudocodes in Algorithm 1.

The essential idea is to choose the smallest threshold on the classification scores such that the violation rate (that is, the probability that the population type I error exceeds α) is controlled under some prespecified tolerance parameter δ, that is, $\mathbb{P}[R_0(\phi^c) > \alpha] \le \delta$. The threshold c is to be chosen from an order statistic of classification scores of a left-out class 0 sample, which is not used to train base algorithms (for example, LR, SVMs, and RFs) to obtain f. Because we do not impose any assumptions on the underlying data-generating process, it is not feasible to establish oracle-type theoretical properties regarding type II errors under this umbrella algorithm. However, because users may favor different base algorithms, this umbrella algorithm is generally applicable in light of the general preference toward conservatism with regard to type I errors. The theoretical foundation of the umbrella algorithm is explained by the following proposition.

Proposition 1. *Suppose that we divide the training data into two parts, one with data from both classes 0 and 1 for training a base algorithm (for example, LR) to obtain f and the other as a left-out class 0 sample for choosing c. Applying the resulting f to the left-out class 0 sample of size n, we denote the resulting classification scores as $T_1, \cdots, T_n$, which are real-valued random variables. Then, we denote by $T_{(k)}$ the kth order statistic (that is, $T_{(1)} \le \cdots \le T_{(n)}$). For a new observation, if we denote its classification score based on f as T, then we can construct a classifier $\hat{\phi}_k = (T > T_{(k)})$. Then, the population type I error of $\hat{\phi}_k$,*

Algorithm 1. An NP umbrella algorithm

Input:
Training data: A mixed i.i.d. sample $\mathcal{S} = \mathcal{S}^0 \cup \mathcal{S}^1$, where $\mathcal{S}^0$ and $\mathcal{S}^1$ are class 0 and 1 samples, respectively
α: Type I error upper bound, $0 \le \alpha \le 1$; (default $\alpha = 0.05$)
δ: A small tolerance level, $0 < \delta < 1$; (default $\delta = 0.05$)
M: Number of random splits on $\mathcal{S}^0$; (default $M = 1$)

Function RankThreshold(n, α, δ)

For k in {1, ⋯, n} **do** — ◂ For each rank threshold candidate k

$v(k) \leftarrow \sum_{j=k}^{n} \binom{n}{j} (1-\alpha)^j \alpha^{n-j}$ — ◂ Calculate the violation rate upper bound

$k^* \leftarrow \min\{k \in \{1, \cdots, n\} : v(k) \le \delta\}$ — ◂ Pick the rank threshold

Return k^*

Procedure NPClassifier($\mathcal{S}$, α, δ, M)

$n = |\mathcal{S}^0|/2$ — ◂ Denote half of the size of $|\mathcal{S}^0|$ as n

$k^* \leftarrow$ RankThreshold(n, α, δ) — ◂ Find the rank threshold

For i in {1, ⋯, M} **do** — ◂ Randomly split $\mathcal{S}^0$ for M times

$\mathcal{S}_{i,1}^0, \mathcal{S}_{i,2}^0 \leftarrow$ random split on $\mathcal{S}^0$ — ◂ Each time randomly split $\mathcal{S}^0$ into two halves with equal sizes

$\mathcal{S}_i \leftarrow \mathcal{S}_{i,1}^0 \cup \mathcal{S}^1$ — ◂ Combine $\mathcal{S}_{i,1}^0$ and $\mathcal{S}^1$

$\mathcal{S}_{i,2}^0 = \{x_1, \cdots, x_n\}$ — ◂ Write $\mathcal{S}_{i,2}^0$ as a set of n data points

$f_i \leftarrow$ ClassificationAlgorithm($\mathcal{S}_i$) — ◂ Train a scoring function f_i on $\mathcal{S}_i$

$\mathcal{T}_i = \{t_{i,1}, \cdots, t_{i,n}\} \leftarrow \{f_i(x_1), \cdots, f_i(x_n)\}$ — ◂ Apply the scoring function f_i to $\mathcal{S}_{i,2}^0$ to obtain a set of score threshold candidates

$\{t_{i,(1)}, \cdots, t_{i,(n)}\} \leftarrow$ sort($\mathcal{T}_i$) — ◂ Sort elements of $\mathcal{T}_i$ in an increasing order

$t_i^* \leftarrow t_{i,(k^*)}$ — ◂ Find the score threshold corresponding to the rank threshold k^*

$\phi_i(X) = I(f_i(X) > t_i^*)$ — ◂ Construct an NP classifier based on the scoring function f_i and the threshold t_i^*

Output: an ensemble NP classifier $\hat{\phi}_\alpha(X) = I\left(\frac{1}{M}\sum_{i=1}^{M} \phi_i(X) \ge \frac{1}{2}\right)$ — ◂ By majority vote

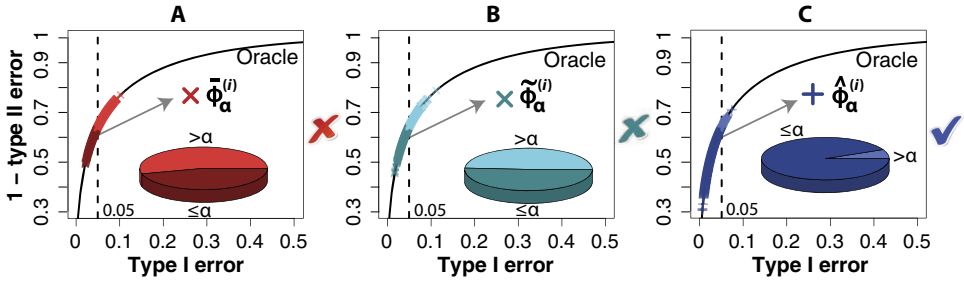

Fig. 2. Choose a threshold such that the type I error is below α with high probability. (**A**) The naïve approach. (**B**) The fivefold CV–based approach. (**C**) The NP approach. *D* NP classifies, based on *D* = 1000 data sets, shown as overlapping red and teal "x" (A and B, respectively) and blue "+" (C) on the curves.

denoted by $R_0(\hat{\phi}_k)$*, is a function of* $T_{(k)}$ *and hence a random variable. Assuming that the data used to train the base algorithm and the left-out class 0 data are independent, we have*

$$\mathbb{P}[R_0(\hat{\phi}_k) > \alpha] \le \sum_{j=k}^{n} \binom{n}{j} (1-\alpha)^j \alpha^{n-j}. \quad (1)$$

That is, the probability that the type I error of $\hat{\phi}_k$ *exceeds* α *is under a constant that only depends on k and* α*. We call this probability the "violation rate" of* $\hat{\phi}_k$ *and denote its upper bound by* $v(k) = \sum_{j=k}^{n} \binom{n}{j} (1-\alpha)^j \alpha^{n-j}$*. When* T_i*'s are continuous, this bound is tight.*

For the proof of Proposition 1, refer to the Supplementary Materials. Note that Proposition 1 is general because it does not rely on any distributional assumptions or on base algorithm characteristics.

Note that $v(k)$ decreases as k increases. If we would like to construct an NP classifier based on an order statistic of the classification scores of the left-out class 0 sample, the right order should be

$$k^* = \min\{k \in \{1, \cdots, n\} : v(k) \le \delta\}. \quad (2)$$

To control the violation rate under δ at least in the extreme case when $k = n$, we need to have $v(n) = (1-\alpha)^n \le \delta$. If the nth order statistic cannot guarantee this violation rate control, then other order statistics certainly cannot. Therefore, given α and δ, we need to have the minimum sample size requirement $n \ge \log \delta / \log(1-\alpha)$ for type I error violation rate control; otherwise, the control cannot be achieved, at least by this order statistic approach.

Algorithm 1 describes our umbrella NP algorithm, which supports popular classification methods such as LR and SVM. Proposition 1 provides the theoretical guarantee on type I error violation rate control for one random split ($M = 1$ in the algorithm). In multiple ($M > 1$) random splits, with each split dividing class 0 training data into two halves, an ensemble classifier by majority voting is demonstrated to maintain the type I error control and reduce the expectation and variance of the type II error in numerical experiments (see Simulation S2 and tables S1 to S6 in the Supplementary Materials).

Applying this algorithm to the example in Simulation 1, with α = 0.05, δ = 0.05, and the number of random splits $M = 1$, we construct D NP classifiers $\hat{\phi}_\alpha^{(1)}, \ldots, \hat{\phi}_\alpha^{(D)}$ based on the $D = 1000$ data sets. We mark these D NP classifiers on the oracle ROC curve (shown as blue "+" in Fig. 2C). Unlike the classifiers constructed by the naïve approach (red "×" in Fig. 2A) or the fivefold CV (cyan "×" in Fig. 2B), we see that these D NP classifiers have type I errors below α with high (at least 1 − δ) probability.

Empirical ROC curves

A popular tool to evaluate binary classification methods is the ROC curve, which provides graphical illustration of the overall performance of a classification method, by showing its all possible type I and II errors. ROC curves have numerous applications in signal detection theory, diagnostic systems and medical decision-making, among other fields (*32*–*34*). For the scoring type of binary classification methods that we focus on in this paper, ROC curves illustrate the overall performance at all possible values of the threshold c on the output classification scores. An ROC space is defined as a two dimensional $[0, 1] \times [0, 1]$ space whose horizontal and vertical axes correspond to "type I error" (or "false-positive rate") and "1 − type II error" (or "true-positive rate"), respectively. For a binary classification method, its scoring function $f(\cdot)$ estimated from the training data corresponds to an ROC curve, with every point on the curve having the coordinates (type I error, 1 − type II error) for a given threshold c. The area under the ROC curve is a widely used metric to evaluate a classification method and compare different methods. In practice, the typical construction of empirical ROC curves includes the following three approaches: by varying the threshold value c, points on empirical ROC curves have horizontal and vertical coordinates, respectively, (approach 1) defined as $\hat{R}_0(\phi^c)$ and $1 - \hat{R}_1(\phi^c)$ on the training data; (approach 2) defined as $\tilde{R}_0(\phi^c)$ and $1 - \tilde{R}_1(\phi^c)$ on the test data; and (approach 3) empirical type I error and (1 − empirical type II error) are estimated by CV on the training data.

Because an empirical ROC curve constructed by any of the above three approaches is an estimate of the unobserved oracle ROC curve, there is literature on the construction of confidence bands of the oracle ROC curve. For pointwise confidence intervals of points on the oracle ROC curve, typical construction methods use a binomial distribution (*35*) or binormal distribution (*36*) to model the distribution of the given point's corresponding points on empirical ROC curves. With regard to constructing simultaneous confidence intervals to form a confidence band of the oracle ROC curve, there are methods based on bootstrapping (*37*) or the Working-Hotelling method (*36*), among others. For a review of existing methods and an empirical study that compares them, see the study of Macskassy *et al.* (*38*). Although these construction methods of empirical ROC curves and confidence bands have many useful applications, none of them is appropriate for evaluating NP classification methods because the empirical ROC curves and confidence bands do not provide users with direct information to find the classifiers that satisfy their prespecified type I error upper bound α with high probability. Simulation S1 in the Supplementary Materials and Fig. 3 demonstrate this issue.

NP-ROC bands

Motivated by the NP umbrella algorithm, we propose that the NP-ROC bands, the second main contribution of this paper, serve as a new visualization tool for classification methods under the NP paradigm. In the NP-ROC space, the horizontal axis is defined as the type I error upper bound (with high probability), and the vertical axis represents (1 − conditional type II error), where we define the conditional type II error of a classifier as its type II error conditioning on training data. An NP classifier corresponds to a vertical line segment (that is, a blue dashed line segment in Fig. 4A) in the NP-ROC space. The horizontal coordinate of a line segment represents the type I error upper bound of that classifier. The vertical coordinates of the upper and lower ends of the segment represent the upper and lower high-probability bounds of (1 − conditional type II error) of that classifier.

To create NP-ROC bands, the sample splitting scheme is slightly different from that of the umbrella algorithm. We still follow the umbrella algorithm to divide the class 0 data into two halves, using the first half to train the scoring function and the second half (size n) to estimate the score threshold. However, to calculate the high-probability (1 − conditional type II error) bounds, we also need to divide the class 1 data into two halves, using the first half to train the scoring function and the second half to calculate the bounds. Therefore, in the construction of NP-ROC bands, we refer to the class 0 data and the first half of the class 1 data as the training data. After we train a scoring function f and apply it to the left-out class 0 data, we obtain n score thresholds and sort them in an increasing order. For the classifier corresponding to the kth-ordered score threshold, that is, $\hat{\phi}_k$, given a predefined tolerance level δ, we find the (1 − δ) probability upper bound of $R_0(\hat{\phi}_k)$ as

$$\alpha(\hat{\phi}_k) = \inf\left\{\alpha \in [0,1] : \sum_{j=k}^{n} \binom{n}{j} (1-\alpha)^j \alpha^{n-j} \le \delta\right\}, \qquad (3)$$

because we have $\mathbb{P}[R_0(\hat{\phi}_k) \le \alpha(\hat{\phi}_k)] \ge 1 - \sum_{j=k}^{n} \binom{n}{j} (1-\alpha(\hat{\phi}_k))^j (\alpha(\hat{\phi}_k))^{n-j} \ge 1 - \delta$, where the first inequality follows from Eq. 1. We next derive the (1 − δ) high-probability lower and upper bounds of the conditional type II error, denoted by $R_1^c(\hat{\phi}_k)$, as $\beta_L(\hat{\phi}_k)$ and $\beta_U(\hat{\phi}_k)$ based on eqs. S9 and S10. For every rank $k \in \{1, \cdots, n\}$, we calculate $(\alpha(\hat{\phi}_k), 1 - \beta_U(\hat{\phi}_k))$ and $(\alpha(\hat{\phi}_k), 1 - \beta_L(\hat{\phi}_k))$ for the classifier $\hat{\phi}_k$ (shown as the lower and upper ends of a blue dashed vertical line segment in Fig. 4A). Varying k from 1 to n, we obtain n vertical line segments in the NP-ROC space. For a classifier $\hat{\phi}$ with a score threshold between two ranks of the left-out class 0 scores, say the $(k-1)$-th and the kth, we have $R_0(\hat{\phi}_k) \le R_0(\hat{\phi}) \le R_0(\hat{\phi}_{k-1})$ and $R_1^c(\hat{\phi}_{k-1}) \le R_1^c(\hat{\phi}) \le R_1^c(\hat{\phi}_k)$. Hence, we set $\beta_L(\hat{\phi}) = \beta_L(\hat{\phi}_{k-1})$ and $\beta_U(\hat{\phi}) = \beta_U(\hat{\phi}_k)$. Because k increases from 1 to n, the vertical line segment shifts from right to left, and we interpolate the n upper ends of these segments using right-continuous step functions and the n lower ends using left-continuous step functions. The band created after the interpolation is called an NP-ROC band (between the two black stepwise curves in Fig. 4A). This band has the interpretation that every type I error upper bound α corresponds to a vertical line segment and the achievable (1− conditional type II error) is sandwiched between the lower and upper ends of the line segment with a probability of at least 1 − 2δ. When we randomly split the training data for $M > 1$ times and repeat the above procedure, we obtain M NP-ROC bands. For the M upper curves and M lower curves, respectively, we calculate the average of the vertical values for every horizontal value to obtain an average upper curve and an average lower curve, which form an NP-ROC band for multiple random splits.

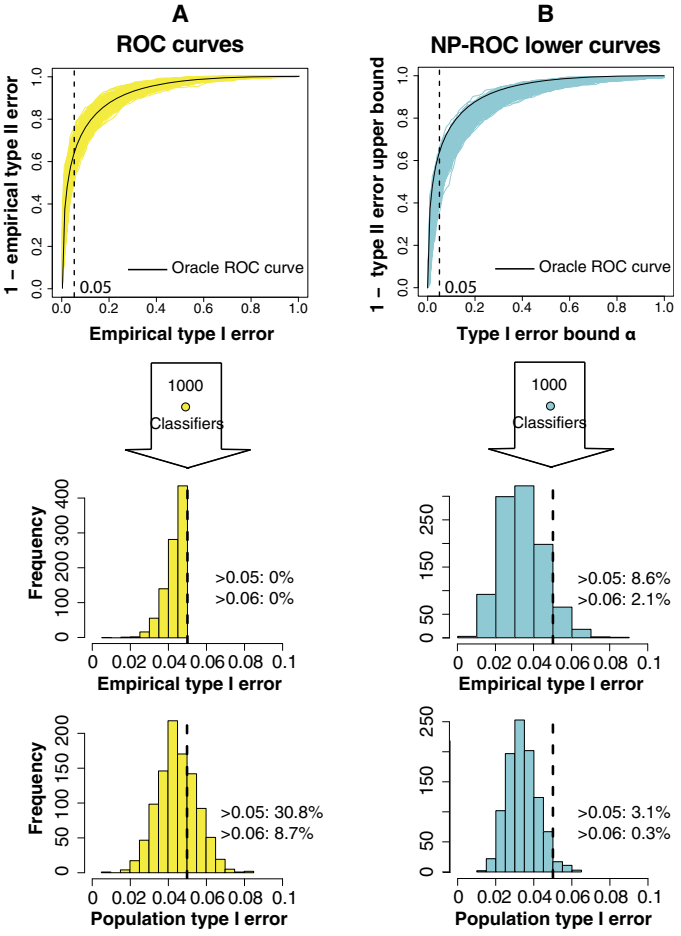

Fig. 3. Illustration of choosing NP classifiers from empirical ROC curves and NP-ROC lower curves in Simulation S1 (see the Supplementary Materials). (**A**) Distributions of empirical type I errors and population type I errors of 1000 classifiers, with each classifier chosen from one empirical ROC curve corresponding to the largest empirical type I error no greater than 0.05. (**B**) Distributions of empirical type I errors and population type I errors of 1000 classifiers, with each classifier chosen from one NP-ROC lower curve (δ = 0.05) corresponding to α = 0.05.

Applications of NP-ROC bands

By definition, the NP-ROC bands work naturally as a visualization tool for NP classification methods. In practice, two issues remain to be addressed in the implementation of classification methods under the NP paradigm: (i) how to choose a reasonable type I error upper bound α if users do not have a prespecified value in mind and (ii) how to compare two classification methods under the NP paradigm. Below, we use simulation and real data applications to demonstrate that the NP-ROC bands serve as an effective tool to address these two questions.

Simulation 2

We consider two independent predictors X_1 and X_2 with $(X_1|Y=0) \sim N(0,1)$, $(X_1|Y=1) \sim N(1,1)$, $(X_2|Y=0) \sim N(0,1)$, $(X_2|Y=1) \sim N(1,6)$, and $\mathbb{P}(Y=0) = \mathbb{P}(Y=1) = 0.5$. We simulate a data set with sample size $N = 1000$ and set the number of random splits to $M = 11$ and the tolerance level to δ = 0.1. We use the LDA with only X_1 (referred

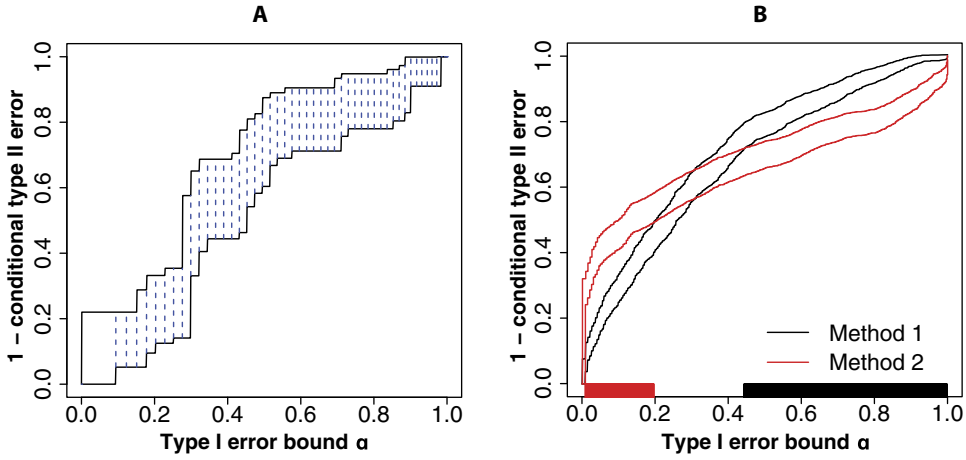

Fig. 4. Illustration of NP-ROC bands. (**A**) How to draw an NP-ROC band. Each blue dashed line represents one NP classifier, with horizontal coordinate α and vertical coordinates $1 - \beta_U$ (lower) and $1 - \beta_L$ (upper). Right-continuous and left-continuous step functions are used to interpolate points on the upper and lower ends, respectively. (**B**) Use of NP-ROC bands to compare the two LDA methods in Simulation 2.

to as method 1) and only X_2 (referred to as method 2). For each classification method, we generate its corresponding NP-ROC bands for comparison, as shown in Fig. 4B. At the horizontal axis, we mark in black the α values for which the lower curve of method 1 is higher than the upper curve of method 2, and similarly, we mark in red the α values for which the lower curve of method 2 is higher than the upper curve of method 1. Given a tolerance level δ, users can read from these colored regions and easily decide which method performs better at any α values under the NP paradigm. Specifically, if users prefer a small α under 0.05, then method 2 would be the choice between the two methods. In a different situation, suppose a user wants to choose α for method 1 and would like to have type II error no greater than 0.5; the NP-ROC band suggests that a reasonable 90% probability upper bound on the type I error should be greater than 0.2.

Real data application 1

We apply NP classification algorithms to a data set from the Early Warning Project. Detailed information of this data set is described in Materials and Methods. We formulated a binary classification problem with a binary response variable, for which a value 0 indicates mass killing and a value 1 indicates otherwise, and 32 predictors. Our goal is to control the type I error (the conditional probability of misclassifying a future mass killing) while optimizing the type II error. After data processing and filtering, there are 6365 observations, among which only 60 have responses as 0's. This is a scenario with extremely imbalanced classes, where the more important class 0 is the rare class. If we use the classical classification paradigm to minimize the risk because of the dominating marginal probability of class 1, then the resulting classifier would prioritize the type II error and possibly result in a large type I error, which is unacceptable in this application. On the other hand, the NP classification paradigm is specifically designed to control the type I error under a prespecified level with high probability. We apply three NP classification methods (RF, penLR, and SVM) to this data and summarize the resulting NP-ROC bands in Fig. 5A. As a comparison, we also use fivefold CV to calculate empirical ROC curves of each method and plot the vertical average curve (for each horizontal value, we calculate the average of the five vertical values in the five empirical ROC curves), which we denote by ROC-CV, in Fig. 5B. In this application, both NP-ROC bands and ROC-CV curves indicate that RF is the best among the three methods. For direct visual comparison, we plot the NP-ROC band and the ROC-CV curve in one panel for RF (Fig. 5C) and SVM (Fig. 5D), respectively. Here, we emphasize again that NP-ROC bands and ROC-CV curves have different horizontal and vertical axes. From Fig. 5 (C and D), it is obvious that ROC-CV curves do not provide guidance on how to determine α in a data-adaptive way because the horizontal axis of ROC-CV represents the empirical type I error, whereas NP-ROC bands have α as the horizontal axis and allow users to decide a reasonable α based on the corresponding (1 – δ) high-probability lower and upper bounds of the conditional type II errors. After RF is chosen as the classification method, the NP-ROC band in Fig. 5C suggests that α = 0.21 (black dashed vertical line) might be a reasonable 90% probability upper bound on the type I error if political scientists desire to have the conditional type II error no greater than 0.4. We note that, although the ROC-CV curve in Fig. 5C suggests that the classifier with the empirical type I error of 0.2 might be a reasonable choice because the point on the ROC-CV curve corresponding to that classifier has the smallest distance to the point (0,1), this conclusion is made from a perspective different from the type I error control, and the chosen classifier is also different from the one corresponding to α = 0.2. Another note is that, if users do not have a clear threshold on the conditional type II error to choose α, then another possible way to choose α in a data-adaptive way from the NP-ROC lower curve is to use the idea of the Youden index (*39*): find α to maximize (1 – conditional type II error – α), that is, the vertical distance from the lower curve to the diagonal line. We find α = 0.249 by this criterion (green dashed vertical line in Fig. 5C).

Real data application 2

We also implement the NP umbrella algorithm and NP-ROC bands on a neuroblastoma data set containing 43,827 gene expression measurements of 498 neuroblastoma samples. Detailed information of this data set is described in Materials and Methods. These neuroblastoma samples fall into two categories: 176 high-risk (HR) samples or 322 non-HR samples. It is commonly understood that misclassifying an HR sample as non-HR will have more severe consequences than the other way around. Formulating this problem under the NP classification framework, we denote the HR samples as class 0 and the non-HR samples as class 1 and use the 43,827 gene expression measurements as features to classify the samples. Setting the tolerance level as δ = 0.1, we create NP-ROC bands for three classification methods: penLR, RF, and NB. In Fig. 6A, we compare penLR and RF. At every α value,

because neither band dominates the other, we cannot distinguish between these two methods with confidence across the whole domain. In Fig. 6B, we compare penLR and NB. The long black bar at the bottom of the plot indicates that penLR dominates NB for most of the type I upper bound α values. In this application, if we choose penLR or RF, then Fig. 6A shows that it is reasonable to set $\alpha = 0.1$. Given that $\alpha = 0.1$ and $\delta = 0.1$, after randomly splitting the data into training data with a size of 374 (three-fourths of the observations) and test data with a size of 124 (one-fourth of the observations) for 100 times, we also calculate the empirical type I and II errors on the test data (table S8). Although we can never observe the distribution of population type I and II errors, the results show that the NP approach in practice also effectively controls the empirical type I errors under α with high probability by paying the price of having larger empirical type II errors.

Besides the three examples noted above, in Simulation S1 in the Supplementary Materials, we further demonstrate that users cannot simply determine a classifier from an empirical ROC curve to control the type I error under α with high probability, whereas our proposed NP-ROC bands provide direct information for users to make such a decision. We also show in Simulation S1 and prove in the Supplementary Materials that the NP-ROC lower curve is a conservative pointwise estimate of the oracle ROC curve.

DISCUSSION

Here, we propose an NP umbrella algorithm to implement scoring-type classification methods under the NP paradigm. This algorithm guarantees the desired high-probability control of type I error, allowing us to construct NP classifiers in a wide range of application contexts. We also propose NP-ROC bands, a new variant of the ROC curves under the NP paradigm. NP-ROC bands provide direct information on the population type I error bounds, α, and a range of achievable type II errors for any given α.

The NP umbrella algorithm and NP-ROC bands have broad application potentials in fields where users often face asymmetric binary classification problems and have some prespecified tolerance levels on the prioritized type of error, which is regarded as the type I error under the NP paradigm. The NP-ROC lower curves have been successfully applied in a comparison of similarity measures for gene expression samples, where the prioritized type of error is mispredicting dissimilar samples as similar, because the

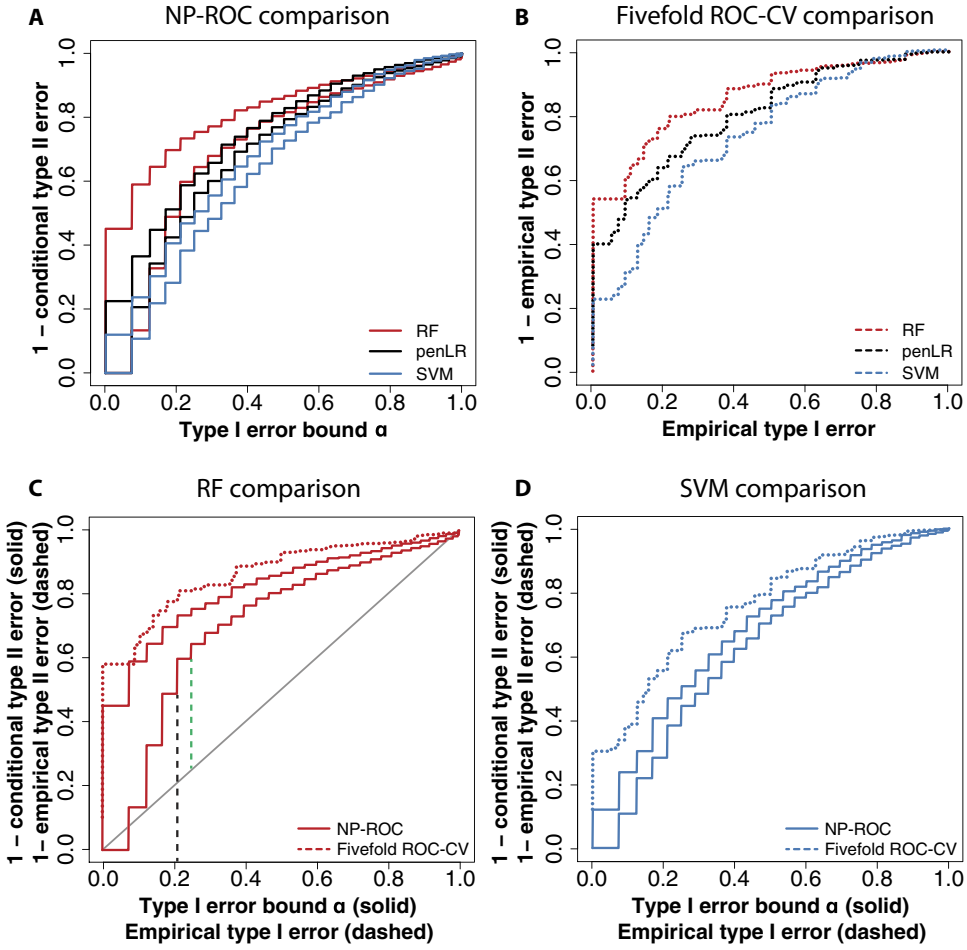

Fig. 5. NP-ROC bands and ROC-CV curves (generated by fivefold CV) of three classification methods in real data application 1. (**A**) NP-ROC bands of RFs versus penLR versus SVMs. RF dominates the other two methods for a wide range of α values. (**B**) ROC-CV curves of RF, penLR, and SVM. Among the three methods, RF has the largest area under the curve. (**C**) NP-ROC band and ROC-CV curve of RF. The black dashed vertical line marks $\alpha = 0.21$, the smallest α such that the conditional type II error is no greater than 0.4. The green dashed vertical line marks $\alpha = 0.249$, the value that maximizes the vertical distance from the lower curve to the diagonal line, a criterion motivated by the Youden index (*39*). (**D**) NP-ROC band and ROC-CV curve of SVM.

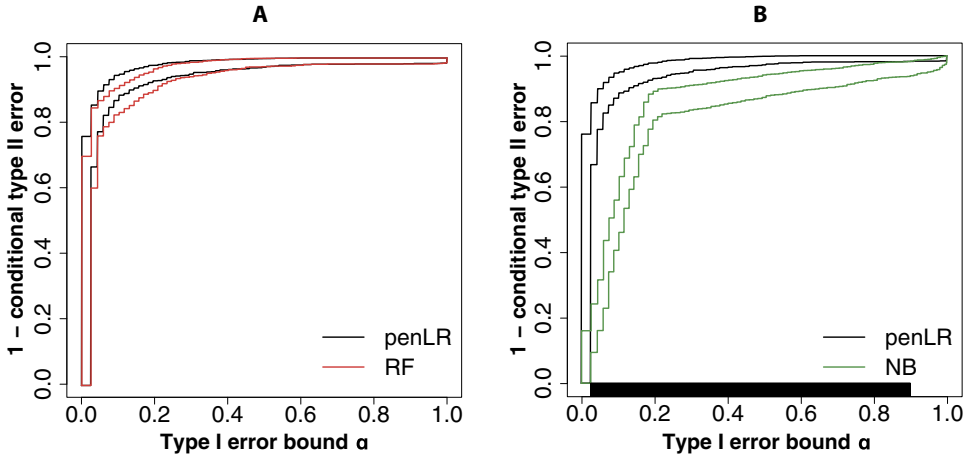

Fig. 6. NP-ROC bands of three classification methods in real data application 2. (**A**) penLR versus RFs. No method dominates the other for any **α** values. (**B**) penLR versus NB. The black bar at the bottom indicates the **α** values where penLR is better than NB with high probability.

samples, if predicted similar, will be used for further biological interpretation (*40*). Other potential applications include biomedical applications such as disease diagnosis (*18*, *19*), engineering applications such as network security control (*20*, *21*), financial applications such as loan screening (*30*) and financial data forecasting (*41*–*44*), and social applications such as prediction of regional and international conflicts (*23*–*25*).

Besides the advantage of enabling user-specific control on the prioritized type of classification error, another advantage of the NP classification paradigm is that its resulting classifier will not be dominated by the majority class in imbalanced classification scenarios, whereas classifiers constructed by the classical paradigm, which seeks to minimize the classification risk, might predict all observations to be the majority class in an extremely imbalanced case. Moreover, in applications with imbalanced classes, down-sampling techniques or oversampling methods, such as Random OverSampling Examples (ROSE) (*45*) and Synthetic Minority Oversampling Technique (SMOTE) (*46*), that can potentially improve the training of classification algorithms can be easily incorporated into the NP umbrella algorithm (Step 14, the training step in Algorithm 1) to potentially reduce the type II error given α.

We note that $n_{\min} = \log \delta / \log(1 - \alpha)$ is the minimum left-out class 0 sample size to ensure that the algorithm outputs a classifier with the type I error below α with probability at least $(1 - \delta)$. Both the type I error upper bound α and the tolerance level δ are subject to user's preferences. In practice, if a user has a small class 0 sample size but would like to achieve a type I error control with high probability using our umbrella algorithm, then he or she must either increase α or δ such that the left-out class 0 sample size is above the corresponding $n_{\min}$. For commonly used α and δ values, $n_{\min}$ is satisfied in most applications. For example, $n_{\min} = 59$ when $\alpha = 0.05$ and $\delta = 0.05$, $n_{\min} = 45$ when $\alpha = 0.1$ and $\delta = 0.05$, and $n_{\min} = 29$ when $\alpha = 0.05$ and $\delta = 0.1$. Having large class 1 and class 0 samples to train the scoring function will reduce the type II error, but it will not affect the type I error control.

One limitation of the current umbrella algorithm is the requirement of sample homogeneity, that is, the data points in each class are independently and identically distributed. For future studies, we will generalize the umbrella algorithm for dependent data and investigate the optimality of the sample splitting ratio used in the umbrella algorithm under specific model settings.

Note that optimality result cannot be established for the type II error without assumptions on the data distributions. In contrast to our previous work (*9*, *16*), the current work does not aim to construct an optimal classifier by estimating the two-class density ratio, which has the optimality guaranteed by the Neyman-Pearson Lemma for hypothesis testing. The reason is that the plug-in density ratio approach has difficulty with density estimation in high dimensions without restrictive assumptions, and in practice, multiple machine learning methods have been widely applied to binary classification problems with high-dimensional features. Therefore, in our current work, we developed the umbrella algorithm to integrate popular binary classification methods into the NP paradigm. Given each method, its trained scoring function, and a left-out class 0 sample, our umbrella algorithm outputs a classifier that satisfies the type I error control and has the minimum type II error. In practice, users can use the NP umbrella algorithm with different classification methods and choose the method that dominates other methods at a specific α value or in a range of α values based on NP-ROC bands. Otherwise, if NP-ROC bands do not suggest clear dominance of a classification method at the users' choice of α, then users can choose the method that gives the lowest type II error by CV.

It is possible to extend the NP umbrella algorithm to the multiclass classification problem. Here, we describe a simple but common scenario where users have three classes in a decreasing order of priority (for example, class 0, cancer of a more dangerous subtype; class 1, cancer of a less dangerous subtype; class 2, benign tumor) and would like to first control the error of misclassifying class 0 and then the error of misclassifying class 1. In this scenario, we can adopt our umbrella algorithm to address the needs. A recent work provided that breaking a multiclass classification problem into multiple binary classification problems may lead to poor performance (*47*). One advantage of our umbrella algorithm is that it can be used with any good multiclass classification algorithms, and it does not need to decompose the classification problem into multiple binary classification problems if the algorithm does not do so. The basic idea of using the umbrella algorithm in the multiclass case is as follows. Given a trained multiclass classification algorithm, we first apply it to the left-out class 0 data to obtain the probabilities of assigning them to class 0. With a prespecified $(1 - \delta)$ probability upper bound α_0 on the error $\mathbb{P}(\phi(X) \neq Y \,|\, Y = 0)$, where $\phi(X)$ denotes the predicted class label, and Y denotes the actual class label, we can use Proposition 1 to find a threshold c_0 to assign a new data point to class 0. Similarly, we can apply the trained algorithm to the left-out class 1 data to obtain the probabilities of assigning

them to class 1. With a prespecified $(1-\delta)$ probability upper bound α_1 on the error $\mathbb{P}(\phi(X) \neq Y | Y = 1)$, we can use Proposition 1 to find another threshold c_1 to assign a new data point to class 1. With these two thresholds, we can construct the following classifier: Given a new data point, we use the trained algorithm to estimate its probabilities of being assigned to classes 0 and 1, denoted as $\hat{p}_0$ and $\hat{p}_1$, respectively. If $\hat{p}_0 \geq c_0$, then we will assign this new data point to class 0. If $\hat{p}_0 < c_0$ and $\hat{p}_1 \geq c_1$, then we will assign the new data point to class 1. Otherwise, we will assign it to class 2.

MATERIALS AND METHODS

Data

We used two real public data sets to demonstrate the use of our proposed NP umbrella algorithm and NP-ROC bands. The first data set was from the Early Warning Project (www.earlywarningproject.com/) (*25*), which collected historical data over 50 years around the world and aimed to produce risk assessments of the potential for mass atrocities. The data set, which we provided in the Supplementary Materials containing our R codes, contained 9496 observations and 243 variables. We considered the binary variable mkl.start.1 as the response, which denoted the onset of state-led mass killing episode in the next year, and we used 32 variables as predictors, which are summarized in table S7. We formulated the prediction of (1 − mkl.start.1), for which a value 0 indicates mass killing and a value 1 indicates otherwise, as a binary classification problem. After we removed the observations with NA (not available) values in the response or any of the 32 variables, there remained 6364 observations, among which only 60 had responses as 0's. The second data set contained 43,827 gene expression measurements from Illumina RNA sequencing of 498 neuroblastoma samples (Gene Expression Omnibus accession number GSE62564, with file name GSE62564_SEQC_NB_RNA-Seq_log2RPM.txt.gz) generated by the Sequencing Quality Control (SEQC) consortium (*48*–*51*). We also provided this data set in the Supplementary Materials containing our R codes. Each neuroblastoma sample was labeled as HR or non-HR, indicating whether the sample belonged to a HR patient based on clinical evidence. There were 176 HR samples and 322 non-HR samples.

R package implementation

In our nproc R package, we implemented the NP umbrella algorithm in the npc function, which output a classifier given the input training data, user-specified classification method, type I error upper bound α, and tolerance level δ. We implemented the NP-ROC bands in the nproc function, which output an NP-ROC band given the input training data, user-specified classification method, and tolerance level δ. The detailed input information is summarized in table S9.

Software

The R package nproc is available at https://cran.r-project.org/web/packages/nproc/index.html. The Shiny version with graphical user interface is available online at https://yangfeng.shinyapps.io/nproc/ and also available for use on local computers after users install the R package. The R codes for generating figures and tables are available in a supporting .zip file within the Supplementary Materials.

SUPPLEMENTARY MATERIALS

Supplementary material for this article is available at http://advances.sciencemag.org/cgi/content/full/4/2/eaao1659/DC1

Proof of Proposition 1
Conditional type II error bounds in NP-ROC bands
Empirical ROC curves versus NP-ROC bands in guiding users to choose classifiers to satisfy type I error control
Effects of majority voting on the type I and II errors of the ensemble classifier
table S1. Results of LR in Simulation S2.
table S2. Results of SVMs in Simulation S2.
table S3. Results of RFs in Simulation S2.
table S4. Results of NB in Simulation S2.
table S5. Results of LDA in Simulation S2.
table S6. Results of AdaBoost in Simulation S2.
table S7. Description of variables used in real data application 1.
table S8. The performance of the NP umbrella algorithm in real data application 2.
table S9. Input information of the nproc package (version 2.0.9).
R codes and data sets

REFERENCES AND NOTES

1. S. B. Kotsiantis, Supervised machine learning: A review of classification techniques. *Informatica* **31**, 249–268 (2007).
2. K. B. Meyer, S. G. Pauker, Screening for HIV: Can we afford the false positive rate? *N. Engl. J. Med.* **317**, 238–241 (1987).
3. S. Liu, C. F. Babbs, E. J. Delp, Multiresolution detection of spiculated lesions in digital mammograms. *IEEE Trans. Image Process.* **10**, 874–884 (2001).
4. M. Dettling, P. Bühlmann, Boosting for tumor classification with gene expression data. *Bioinformatics* **19**, 1061–1069 (2003).
5. E. A. Freeman, G. G. Moisen, A comparison of the performance of threshold criteria for binary classification in terms of predicted prevalence and kappa. *Ecol. Model.* **217**, 48–58 (2008).
6. A. Cannon, J. Howse, D. Hush, C. Scovel, Learning with the Neyman-Pearson and min-max criteria, Los Alamos National Laboratory, Tech. Rep. LA-UR-02-2951 (2002).
7. C. Scott, R. Nowak, A Neyman-Pearson approach to statistical learning. *IEEE Trans. Inf. Theory* **51**, 3806–3819 (2005).
8. P. Rigollet, X. Tong, Neyman-Pearson classification, convexity and stochastic constraints. *J. Mach. Learn. Res.* **12**, 2831–2855 (2011).
9. X. Tong, A plug-in approach to Neyman-Pearson classification. *J. Mach. Learn. Res.* **14**, 3011–3040 (2013).
10. C. Elkan, The foundations of cost-sensitive learning. *Proc. 17th Int. Jt. Conf. Artif. Intell.* **2**, 973–978 (2001).
11. B. Zadrozny, J. Langford, N. Abe, Cost-sensitive learning by cost-proportionate example weighting, in *Third IEEE International Conference on Data Mining (ICDM'03)*, Melbourne, FL, 22 November 2003.
12. S.-H. Wu, K.-P. Lin, H.-H. Chien, C.-M. Chen, M.-S. Chen, On generalizable low false-positive learning using asymmetric support vector machines. *IEEE Trans. Knowl. Data Eng.* **25**, 1083–1096 (2013).
13. D. Casasent, X.-w. Chen, Radial basis function neural networks for nonlinear Fisher discrimination and Neyman–Pearson classification. *Neural Netw.* **16**, 529–535 (2003).
14. C. Scott, Comparison and design of Neyman-Pearson classifiers (2005); http://www.stat.rice.edu/~cscott/pubs/npdesign.pdf.
15. M. Han, D. Chen, Z. Sun, Analysis to Neyman-Pearson classification with convex loss function. *Anal. Theory Appl.* **24**, 18–28 (2008).
16. A. Zhao, Y. Feng, L. Wang, X. Tong, Neyman-Pearson classification under high-dimensional settings. *J. Mach. Learn. Res.* **17**, 1–39 (2016).
17. X. Tong, Y. Feng, A. Zhao, A survey on Neyman-Pearson classification and suggestions for future research. *WIREs Comput. Stat.* **8**, 64–81 (2016).
18. Y. Zheng, M. Loziczonek, B. Georgescu, S. K. Zhou, F. Vega-Higuera, D. Comaniciu, Machine learning based vesselness measurement for coronary artery segmentation in cardiac CT volumes. *Proc. SPIE* **7962**, 79621K (2011).
19. K. Bourzac, Diagnosis: Early warning system. *Nature* **513**, S4–S6 (2014).
20. C. Goues, W. Weimer, Specification mining with few false positives, in *Proceedings of the 15th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'09)*, York, UK, 22 to 29 March 2009.
21. D. Kong, D. Tian, Q. Pan, P. Liu, D. Wu, Semantic aware attribution analysis of remote exploits. *Secur. Commun. Netw.* **6**, 818–832 (2013).
22. National Research Council, *A Review of the EPA Water Security Research and Technical Support Action Plan: Parts I and II* (National Academies Press, 2004).
23. P. T. Brandt, J. R. Freeman, P. A. Schrodt, Real time, time series forecasting of inter- and intra-state political conflict. *Confl. Manag. Peace Sci.* **28**, 41–64 (2011).
24. R. Kennedy, Making useful conflict predictions: Methods for addressing skewed classes and implementing cost-sensitive learning in the study of state failure. *J. Peace Res.* **52**, 649–664 (2015).
25. Early Warning Project (2017); www.earlywarningproject.com/.

26. D. R. Cox, The regression analysis of binary sequences. *J. R. Stat. Soc. Ser. B* **20**, 215–242 (1958).
27. C. Cortes, V. Vapnik, Support-vector networks. *Mach. Learn.* **20**, 273–297 (1995).
28. Y. Freund, R. E. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting, in *Computational Learning Theory*, P. Vitanyi, Ed. (Springer-Verlag, 1995), pp. 23–37.
29. L. Breiman, Random forests. *Mach. Learn.* **45**, 5–32 (2001).
30. D. J. Hand, W. E. Henley, Statistical classification methods in consumer credit scoring: A review. *J. R. Stat. Soc. Ser. A* **160**, 523–541 (1997).
31. T. Fawcett, An introduction to ROC analysis. *Pattern Recogn. Lett.* **27**, 861–874 (2006).
32. J. A. Hanley, B. J. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* **143**, 29–36 (1982).
33. A. P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recogn.* **30**, 1145–1159 (1997).
34. M. J. Pencina, R. B. D'Agostino Sr., R. B. D'Agostino Jr., R. S. Vasan, Evaluating the added predictive ability of a new marker: From area under the ROC curve to reclassification and beyond. *Stat. Med.* **27**, 157–172 (2008).
35. R. A. Hilgers, Distribution-free confidence bounds for ROC curves. *Methods Inf. Med.* **30**, 96–101 (1991).
36. G. Ma, W. J. Hall, Confidence bands for receiver operating characteristic curves. *Med. Decis. Making* **13**, 191–197 (1993).
37. G. Campbell, Advances in statistical methodology for the evaluation of diagnostic and laboratory tests. *Stat. Med.* **13**, 499–508 (1994).
38. S. A. Macskassy, F. Provost, S. Rosset, ROC confidence bands: An empirical evaluation, in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 07 to 11 August 2005, pp. 537–544.
39. W. J. Youden, Index for rating diagnostic tests. *Cancer* **3**, 32–35 (1950).
40. W. V. Li, Y. Chen, J. J. Li, TROM: A testing-based method for finding transcriptomic similarity of biological samples. *Stat. Biosci.* **9**, 105–136 (2017).
41. K.-j. Kim, Financial time series forecasting using support vector machines. *Neurocomputing* **55**, 307–319 (2003).
42. R. Choudhry, K. Garg, A hybrid machine learning system for stock market forecasting. *World Acad. Sci. Eng. Technol.* **39**, 315–318 (2008).
43. L. Yu, H. Chen, S. Wang, K. K. Lai, Evolving least squares support vector machines for stock market trend mining. *IEEE Trans. Evol. Comput.* **13**, 87–102 (2009).
44. Y. Kara, M. Acar Boyacioglu, Ö. K. Baykan, Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *Expert Syst. Appl.* **38**, 5311–5319 (2011).
45. N. Lunardon, G. Menardi, N. Torelli, ROSE: A package for binary imbalanced learning. *R J.* **6**, 79–89 (2014).
46. N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002).
47. Q. Mai, Y. Yang, H. Zou, Multiclass sparse discriminant analysis. https://arxiv.org/abs/1504.05845 (2015).
48. C. Wang, B. Gong, P. R. Bushel, J. Thierry-Mieg, D. Thierry-Mieg, J. Xu, H. Fang, H. Hong, J. Shen, Z. Su, J. Meehan, X. Li, L. Yang, H. Li, P. P. Łabaj, D. P. Kreil, D. Megherbi, S. Gaj, F. Caiment, J. van Delft, J. Kleinjans, A. Scherer, V. Devanarayan, J. Wang, Y. Yang, H.-R. Qian, L. J. Lancashire, M. Bessarabova, Y. Nikolsky, C. Furlanello, M. Chierici, D. Albanese, G. Jurman, S. Riccadonna, M. Filosi, R. Visintainer, K. K. Zhang, J. Li, J.-H. Hsieh, D. L. Svoboda, J. C. Fuscoe, Y. Deng, L. Shi, R. S. Paules, S. S. Auerbach, W. Tong, The concordance between RNA-seq and microarray data depends on chemical treatment and transcript abundance. *Nat. Biotechnol.* **32**, 926–932 (2014).
49. SEQC/MAQC-III Consortium, A comprehensive assessment of RNA-seq accuracy, reproducibility and information content by the Sequencing Quality Control Consortium. *Nat. Biotechnol.* **32**, 903–914 (2014).
50. S. A. Munro, S. P. Lund, P. S. Pine, H. Binder, D.-A. Clevert, A. Conesa, J. Dopazo, M. Fasold, S. Hochreiter, H. Hong, N. Jafari, D. P. Kreil, P. P. Łabaj, S. Li, Y. Liao, S. M. Lin, J. Meehan, C. E. Mason, J. Santoyo-Lopez, R. A. Setterquist, L. Shi, W. Shi, G. K. Smyth, N. Stralis-Pavese, Z. Su, W. Tong, C. Wang, J. Wang, J. Xu, Z. Ye, Y. Yang, Y. Yu, M. Salit, Assessing technical performance in differential gene expression experiments with external spike-in RNA control ratio mixtures. *Nat. Commun.* **5**, 5125 (2014).
51. Z. Su, H. Fang, H. Hong, L. Shi, W. Zhang, W. Zhang, Y. Zhang, Z. Dong, L. J. Lancashire, M. Bessarabova, X. Yang, B. Ning, B. Gong, J. Meehan, J. Xu, W. Ge, R. Perkins, M. Fischer, W. Tong, An investigation of biomarkers derived from legacy microarray data for their utility in the RNA-seq era. *Genome Biol.* **15**, 523 (2014).
52. J. J. Li, X. Tong, Genomic applications of the Neyman–Pearson classification paradigm, in *Big Data Analytics in Genomics*, K.-C. Wong, Ed. (Springer, 2016), pp. 145–167.

Acknowledgments

Funding: This work was supported by Zumberge Individual Research Award (to X.T.); NSF grants DMS-1613338 (to X.T. and J.J.L.), DMS-1308566 (to Y.F.), and DMS-1554804 (to Y.F.); Hellman Fellowship (to J.J.L.); NIH grant R01GM120507 (to J.J.L. and X.T.); and the PhRMA Foundation Research Starter Grant in Informatics (to J.J.L.). **Author contributions:** X.T., Y.F., and J.J.L. designed the study, conducted the simulation and real data analysis, and wrote the paper. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper and/or the Supplementary Materials. Additional data related to this paper may be requested from the authors.

Submitted 21 June 2017
Accepted 3 January 2018
Published 2 February 2018
10.1126/sciadv.aao1659

Citation: X. Tong, Y. Feng, J. J. Li, Neyman-Pearson classification algorithms and NP receiver operating characteristics. *Sci. Adv.* **4**, eaao1659 (2018).

Neyman-Pearson classification algorithms and NP receiver operating characteristics

Xin Tong, Yang Feng and Jingyi Jessica Li

Sci Adv **4** (2), eaao1659.
DOI: 10.1126/sciadv.aao1659