

Supplementary Materials for **Neyman-Pearson classification algorithms and NP receiver operating characteristics**

Xin Tong, Yang Feng, Jingyi Jessica Li

Published 2 February 2018, *Sci. Adv.* **4**, eaao1659 (2018)

DOI: 10.1126/sciadv.aao1659

The PDF file includes:

- 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).

Other Supplementary Material for this manuscript includes the following:
(available at advances.sciencemag.org/cgi/content/full/4/2/eaao1659/DC1)

- R codes and data sets

Supplementary Materials

Proof of Proposition 1

Proof: In our context, let $T_{(k)}$ be the k -th ordered classification score of a left-out class 0 sample (i.e., class 0 sample not used to train a base algorithm). Suppose $T_{(k)}$ is the chosen threshold on classification scores. Then the classifier is $\hat{\Phi}_k = (T > T_{(k)})$, where T is the classification score of an independent observation from class 0, and the population type I error of $\hat{\Phi}_k$ given $T_{(k)}$ is

$$\mathbb{P}[T > T_{(k)} | T_{(k)}] = 1 - F(T_{(k)}),$$

where F is the cumulative distribution function of T_i 's.

Hence, the violation rate (the probability that the population type I error of $\hat{\Phi}_k$ exceeds α) is

$$\begin{aligned} \mathbb{P}[1 - F(T_{(k)}) > \alpha] &= \mathbb{P}[F(T_{(k)}) < 1 - \alpha] = \mathbb{P}[T_{(k)} < F^{-1}(1 - \alpha)] \\ &= \mathbb{P}[T_{(1)} < F^{-1}(1 - \alpha), \dots, T_{(k)} < F^{-1}(1 - \alpha)] \\ &= \mathbb{P}[\text{at least } k \text{ of the } T_i\text{'s are less than } F^{-1}(1 - \alpha)] \\ &= \sum_{j=k}^n \mathbb{P}[\text{exactly } j \text{ of the } T_i\text{'s are less than } F^{-1}(1 - \alpha)] \\ &= \sum_{j=k}^n \binom{n}{j} \mathbb{P}[T_i < F^{-1}(1 - \alpha)]^j (1 - \mathbb{P}[T_i < F^{-1}(1 - \alpha)])^{n-j} \\ &\leq \sum_{j=k}^n \binom{n}{j} (1 - \alpha)^j \alpha^{n-j}, \quad (\text{S1}) \end{aligned}$$

where the last inequality holds because $\mathbb{P}[T_i < F^{-1}(1 - \alpha)] \leq 1 - \alpha$, and it becomes an equality when F is continuous.

Remark: The proof does not have any assumptions on F . Regardless of the continuity of F , we define its inverse as $F^{-1}(\cdot) = \inf\{x: F(x) \leq \cdot\}$, which has the property: $x \leq F(y)$ if and only if $F^{-1}(x) \leq y$, for any $x \in [0,1]$ and y in the domain of F .

Conditional type II error bounds in NP-ROC Bands

Given training data $\mathcal{S} = \mathcal{S}^0 \cup \mathcal{S}^1$, where \mathcal{S}^0 and \mathcal{S}^1 are class 0 and class 1 samples, respectively, we randomly split \mathcal{S}^0 into \mathcal{S}_1^0 and \mathcal{S}_2^0 and split \mathcal{S}^1 into \mathcal{S}_1^1 and \mathcal{S}_2^1 . For simplicity, we let $|\mathcal{S}_1^0| = |\mathcal{S}_2^0| = n$ and $|\mathcal{S}_1^1| = |\mathcal{S}_2^1| = m$, and express the two left-out samples as $\mathcal{S}_2^0 = \{x_1^0, \dots, x_n^0\}$ and $\mathcal{S}_2^1 = \{X_1^1, \dots, X_m^1\}$. In the following discussion, we treat \mathcal{S}_1^0 , \mathcal{S}_2^0 and \mathcal{S}_1^1 as fixed (by conditioning on them) and only consider the m data points in \mathcal{S}_2^1 as random variables.

We train a base classification algorithm (e.g., SVM) on $\mathcal{S}_1^0 \cup \mathcal{S}_1^1$ and denote the resulting classification scoring function as f . Because f is a function of $\mathcal{S}_1^0 \cup \mathcal{S}_1^1$, in our discussion here, f is a fixed function that maps \mathcal{X} to \mathbb{R} .

After applying f to the left-out samples \mathcal{S}_2^0 and \mathcal{S}_2^1 , we denote the resulting classification scores as $t_i^0 = f(x_i^0)$, $i = 1, \dots, n$, and $T_j^1 = f(X_j^1)$, $j = 1, \dots, m$, respectively.

Suppose that we decide to use the k -th ordered left-out class 0 score, $t_{(k)}^0$, as the score threshold. We then construct an NP classifier $\hat{\Phi}_k$ (based on one ($M = 1$) random split) as

$$\hat{\Phi}_k(X) = I(f(X) > t_{(k)}^0).$$

We then find the corresponding rank of $t_{(k)}^0$ among the left-out class 1 scores T_1^1, \dots, T_m^1 .

There are three scenarios.

Scenario 1

If $T_{(1)}^1 \leq t_{(k)}^0 \leq T_{(m)}^1$, we define the lower bound rank r_L and the upper bound rank r_U as

$$r_L = \max\{r \in \{1, \dots, m\}: T_{(r)}^1 \leq t_{(k)}^0\}, \quad (\text{S2})$$

$$r_U = \min\{r \in \{1, \dots, m\}: T_{(r)}^1 \geq t_{(k)}^0\}, \quad (\text{S3})$$

and denote their corresponding classifiers as

$$\tilde{\Phi}_{r_L}(X) = I(f(X) > T_{(r_L)}^1), \quad (\text{S4})$$

$$\tilde{\Phi}_{r_U}(X) = I(f(X) > T_{(r_U)}^1). \quad (\text{S5})$$

We define the conditional (here the conditioning is on training data \mathcal{S}_1^0 , \mathcal{S}_2^0 and \mathcal{S}_1^1) type II errors of $\tilde{\Phi}_{r_L}$ and $\tilde{\Phi}_{r_U}$ as

$$R_1^c(\tilde{\Phi}_{r_L}) := \mathbb{P}[f(X^1) \leq T_{(r_L)}^1 | T_{(r_L)}^1] = F_1(T_{(r_L)}^1), \quad (\text{S6})$$

$$R_1^c(\tilde{\Phi}_{r_U}) := \mathbb{P}[f(X^1) \leq T_{(r_U)}^1 | T_{(r_U)}^1] = F_1(T_{(r_U)}^1), \quad (\text{S7})$$

where X^1 represents a new class 1 observation, and F_1 is the cumulative distribution function of classification scores of class 1 observations.

Because $T_{(r_L)}^1 \leq t_{(k)}^0 \leq T_{(r_U)}^1$, we have

$$R_1^c(\tilde{\Phi}_{r_L}) \leq R_1^c(\hat{\Phi}_k) \leq R_1^c(\tilde{\Phi}_{r_U}), \quad (\text{S8})$$

where $R_1^c(\cdot)$ stands for the conditional type II error, and $R_1^c(\hat{\Phi}_k) = \mathbb{P}[f(X^1) \leq t_{(k)}^0] = F_1(t_{(k)}^0)$. In (S8), the left inequality becomes tight when $T_{(r_L)}^1 = t_{(k)}^0$ and the right inequality is tight when $T_{(r_U)}^1 = t_{(k)}^0$.

Given a pre-specified tolerance level δ , denote by $\beta_L(\hat{\Phi}_k)$ and $\beta_U(\hat{\Phi}_k)$ the $(1 - \delta)$ high probability lower and upper bounds of $R_1^c(\hat{\Phi}_k)$. Specifically, $\beta_L(\hat{\Phi}_k)$ and $\beta_U(\hat{\Phi}_k)$ are defined as

$$\beta_L(\hat{\Phi}_k) := \sup \left\{ \beta \in [0,1]: \sum_{j=r_L}^m \binom{m}{j} \beta^j (1 - \beta)^{m-j} \leq \delta \right\}, \quad (\text{S9})$$

$$\beta_U(\hat{\Phi}_k) := \inf \left\{ \beta \in [0,1]: \sum_{j=r_U}^m \binom{m}{j} \beta^j (1 - \beta)^{m-j} \geq 1 - \delta \right\}. \quad (\text{S10})$$

The reason that (S9) and (S10) give valid $(1 - \delta)$ high probability lower and upper bounds is as follows.

- A constant β serves as a valid $(1 - \delta)$ high probability lower bound of $R_1^c(\hat{\Phi}_k)$ if

$$\mathbb{P}[R_1^c(\hat{\Phi}_k) \geq \beta] \geq 1 - \delta. \quad (\text{S11})$$

Since by (S8)

$$\mathbb{P}[R_1^c(\hat{\Phi}_k) \geq \beta] \geq \mathbb{P}[R_1^c(\tilde{\Phi}_{r_L}) \geq \beta].$$

in order to have (S11) hold it is sufficient to have

$$\mathbb{P}[R_1^c(\tilde{\Phi}_{r_L}) \geq \beta] \geq 1 - \delta.$$

By (S6)

$$\begin{aligned} \mathbb{P}[R_1^c(\tilde{\Phi}_{r_L}) \geq \beta] &= \mathbb{P}[F_1(T_{(r_L)}^1) \geq \beta] = 1 - \mathbb{P}[F_1(T_{(r_L)}^1) < \beta] \\ &\geq 1 - \sum_{j=r_L}^m \binom{m}{j} \beta^j (1 - \beta)^{m-j}, \end{aligned}$$

where the inequality in the second line follows from (S1) in the proof of Proposition 1. Hence, it suffices to have

$$\sum_{j=r_L}^m \binom{m}{j} \beta^j (1 - \beta)^{m-j} \leq \delta$$

to make (S11) hold. Among all the β values that satisfy (S11), we would choose the supremum as the $(1 - \delta)$ high probability lower bound of $R_1^c(\hat{\Phi}_k)$, leading to (S9).

- A constant β serves as a valid $(1 - \delta)$ high probability upper bound of $R_1^c(\hat{\Phi}_k)$ if

$$\mathbb{P}[R_1^c(\hat{\Phi}_k) \leq \beta] \geq 1 - \delta. \quad (\text{S12})$$

Since by (S8)

$$\mathbb{P}[R_1^c(\hat{\Phi}_k) \leq \beta] \geq \mathbb{P}[R_1^c(\tilde{\Phi}_{r_U}) \leq \beta],$$

in order to have (S12) hold it is sufficient to have

$$\mathbb{P}[R_1^c(\tilde{\Phi}_{r_U}) \leq \beta] \geq 1 - \delta.$$

By (S7),

$$\begin{aligned}\mathbb{P}[R_1^c(\tilde{\Phi}_{r_U}) \leq \beta] &= \mathbb{P}[F_1(T_{(r_U)}^1) \leq \beta] \\ &= \mathbb{P}[F_1(T_{(r_U)}^1) < \beta] = \sum_{j=r_U}^m \binom{m}{j} \beta^j (1-\beta)^{m-j}\end{aligned}$$

where the two equalities in the second line follow from (S1) in the proof of Proposition 1, under the assumption that F_1 is continuous, which holds for most classification algorithms. Hence, it suffices to have

$$\sum_{j=r_U}^m \binom{m}{j} \beta^j (1-\beta)^{m-j} \geq 1 - \delta$$

to make (S12) hold. Among all the β values that satisfy (S12), we would choose the infimum as the $(1 - \delta)$ high probability upper bound of $R_1^c(\hat{\Phi}_k)$, leading to (S10).

Scenario 2

If $t_{(k)}^0 > T_{(m)}^1$, we define the lower bound rank r_L the same as in (S2) and the $(1 - \delta)$ high probability lower bound $\beta_L(\hat{\Phi}_k)$ the same as in (S9). We set the $(1 - \delta)$ high probability upper bound $\beta_U(\hat{\Phi}_k) = 1$.

Scenario 3

If $t_{(k)}^0 < T_{(1)}^1$, we define the upper bound rank r_U the same as in (S3) and the $(1 - \delta)$ high probability upper bound $\beta_U(\hat{\Phi}_k)$ the same as in (S10). We set the $(1 - \delta)$ high probability lower bound $\beta_L(\hat{\Phi}_k) = 0$.

In all the above three scenarios, we have $\mathbb{P}[R_1^c(\hat{\Phi}_k) < \beta_L(\hat{\Phi}_k)] \leq \delta$ and $\mathbb{P}[R_1^c(\hat{\Phi}_k) > \beta_U(\hat{\Phi}_k)] \leq \delta$, leading to

$$\mathbb{P}[\beta_L(\hat{\Phi}_k) \leq R_1^c(\hat{\Phi}_k) \leq \beta_U(\hat{\Phi}_k)] \geq 1 - 2\delta.$$

Empirical ROC curves versus NP-ROC bands in guiding users to choose classifiers to satisfy type I error control

In practice, the popular ROC curves cannot provide proper guidance for comparing two classifiers whose type I errors are bounded from above by some α , because ROC curves are constructed based on empirical type I and type II errors, which are calculated from test data or cross-validation on training data and do not display population type I error information. We refer to such ROC curves as empirical ROC curves in the main text to differentiate them from the oracle ROC curves. Concretely, given an empirical ROC curve of a classification method, it is unclear how users should decide which point on the curve corresponds to a classifier satisfying the type I error bound α . Through Simulation 1 and Fig. 2, we showed that users cannot simply pick the point that has empirical type I error (i.e., horizontal axis of the ROC curve) no greater than and closest to α , a seemingly intuitive but actually improper practice. We further illustrate this point in Simulation S1 and Fig. 3. Due to the lack of direct information on population type I errors, existing methods for constructing ROC confidence bands cannot serve the purpose either. **Simulation S1.** From the same setup as in Simulation 1:

$$(X|Y = 0) \sim N(0,1) \text{ and } (X|Y = 1) \sim N(2,1), \text{ with } \mathbb{P}(Y = 0) = \mathbb{P}(Y = 1) = 0.5$$

we simulate $2D = 2000$ data sets $\{(x_i^{(m)}, y_i^{(m)})\}_{i=1}^N$, where $m = 1, \dots, 2D$ and $N = 1000$. The first D data sets are used as the training data, and the other D data sets are the test data. On the m -th training data set, we construct N classifiers, $I(X > x_i^{(m)})$, $i = 1, \dots, N$, and evaluate their empirical type I and II errors on the m -th test data set (i.e., the $(D + m)$ -th simulated data set), resulting in one ROC curve. We also use the m -th training data set to calculate one NP-ROC lower curve, i.e., the lower curve of an NP-ROC band. Fig. 3 illustrates the $D = 1000$ ROC and NP-ROC lower curves. Suppose that users would like to find a classifier respecting a type I error bound $\alpha = 0.05$ with tolerance level $\delta = 0.05$ from an ROC curve. An intuitive choice is to pick the classifier at the intersection of the ROC curve and the vertical line at α . If there is no classifier right at the intersection, a reasonable idea is to pick the first classifier to the left of the intersection. For the D classifiers chosen in this way, we summarize their empirical type I errors (on the test data) and their population type I errors as histograms (Fig. 3 left panel). The results suggest that although the classifiers have no *empirical* type I errors greater than α , approximately 30% of the classifiers have *population* type I errors greater than α , violating users' desire for controlling type I error under α with at least 0.95 probability. On the other hand, the NP-ROC lower curves provide a natural way for users to choose classifiers given α , as the horizontal coordinates of the NP-ROC curves are type I error upper bounds. Users can simply pick the classifier with horizontal coordinate α . For the D chosen NP classifiers, we summarize their empirical type I errors on the test data and their population type I errors as histograms (Fig. 3 right panel). It is clear that the violation rate of population type I error is under $\delta = 0.05$.

Another use of the NP-ROC lower curve is that it provides a conservative point-wise estimate of the oracle ROC curve. For an NP classifier $\hat{\phi}$, the corresponding point on an NP-ROC lower curve is, with high probability, below and to the right of the point on the oracle ROC curve. To explain this phenomenon, suppose that the classifier corresponds to the point $(\alpha(\hat{\phi}), 1 - \beta_U(\hat{\phi}))$ on an NP-ROC lower curve and the point $(R_0(\hat{\phi}), 1 - R_1(\hat{\phi}))$ on an oracle ROC curve. Then, by the definition of $\alpha(\hat{\phi})$ (Equation (3)) and $\beta_U(\hat{\phi})$ (Equation (S10)) as the high probability upper bounds on $R_0(\hat{\phi})$ and $R_1^c(\hat{\phi})$ (the conditional type II error, conditioning on the training data), respectively, we know that $\alpha(\hat{\phi}) \geq R_0(\hat{\phi})$ and $1 - \beta_U(\hat{\phi}) \leq 1 - R_1^c(\hat{\phi})$ hold with high

probability. As $R_1(\hat{\Phi}) = \mathbb{E}[R_1^c(\hat{\Phi})]$, where the expectation is with respect to the joint distribution of the training data, we have $1 - \beta_U(\hat{\Phi}) \leq 1 - R_1(\hat{\Phi})$ with high probability. Hence, the point on the NP-ROC lower curve is below and to the right of the oracle ROC curve with high probability. In other words, for an NP classifier, coordinates on the NP-ROC lower curve provide a conservative estimate of the corresponding coordinates on the oracle ROC curve. This phenomenon is visualized in the top right panel of Fig. 3.

Effects of majority voting on the type I and II errors of the ensemble classifier

In the following simulation, we demonstrate that the ensemble classifiers from multiple random splits maintain the type I error violation rates under δ and achieve reduced average type II errors with smaller standard errors over multiple simulations, as compared with the NP classifiers from just one random split.

Simulation S2. Our first generative model is a logistic regression (LR) model with $d = 3$ independent features and $n = 1000$ observations. For each feature, 1000 values are independently drawn from a standard Normal distribution. The three features have non-zero coefficients as 3, 2.4 and 1.8, respectively. Denoting the feature matrix by $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$ and the coefficient vector by $\beta \in \mathbb{R}^d$, we simulate the response vector as

$$\mathbf{Y} = (Y_1, \dots, Y_n)^T, \text{ with } Y_i \stackrel{\text{indep}}{\sim} \text{Bernoulli}\left(\frac{1}{1 + \exp(-\mathbf{x}_i^T \beta)}\right).$$

Our second generative model is a linear discriminant analysis (LDA) model with $d = 3$ independent features and $n = 1000$ observations. We first simulate the response vector as

$$\mathbf{Y} = (Y_1, \dots, Y_n)^T, \text{ with } Y_i \stackrel{\text{indep}}{\sim} \text{Bernoulli}(0.5).$$

Then we simulate the feature matrix $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$ as

$$(\mathbf{x}_i | Y_i = 0) \sim N((0, 0, 0)^T, \mathbf{I}_3); (\mathbf{x}_i | Y_i = 1) \sim N((2, 1.6, 1.2)^T, \mathbf{I}_3).$$

We simulate 1000 datasets from each of the above two models, obtaining 2000 datasets. With $\alpha = 0.05$, $\delta = 0.05$, and varying number of splits $M \in \{1, 5, 9, 11, 15\}$, we construct five NP classifiers for each of six methods (logistic regression, support vector machines, random forests, naïve Bayes, linear discriminant analysis, and AdaBoost) with different numbers of splits based on each data set. We separately simulate two large test data sets with 10^6 observations from the two models, so that we can use the test data set to calculate the test (empirical) type I and type II errors to approximate the population type I and type II errors.

For each combination of M value, classification method and generative model, we use its corresponding 1000 NP classifiers to calculate the average of the approximate type I errors and its standard error, the percentage of classifiers with approximate type I errors greater than α (type I error violation rate), and the average of the approximate type II errors and its standard error. The results summarized in tables S1-S6 show that the type I error violation rates of the ensemble classifiers stay below $\delta = 5\%$, the average type I and type II errors, which approximate the average of the population type I and type II errors of the ensemble classifiers, and their standard deviations decrease from $M = 1$ to $M > 1$.

This simulation experiment and its results demonstrate the validity and effectiveness of the ensemble approach by majority voting.

table S1. Results of LR in Simulation S2. The first column indicates the data generative model (LR: logistic regression; LDA: linear discriminant analysis). The second column indicates the number of random splits on the class 0 training data to construct each ensemble classifier. The third column indicates the (approximate) averages (avg) and standard deviations (sd) of the population type I errors of the ensemble classifiers. The fourth column indicates the proportions of the (approximate) population type I errors exceeding α , i.e., the violation rates. The fifth column indicates the (approximate) averages (avg) and standard deviations (sd) of the population type II errors of the ensemble classifiers.

Model	M	Type I error avg (sd)	Type I error violation rate	Type II error avg (sd)
LR	1	2.78% (1.01%)	2.30%	33.42% (5.19%)
	5	2.72% (0.80%)	1.00%	33.38% (4.12%)
	9	2.71% (0.75%)	0.50%	33.35% (3.89%)
	11	2.71% (0.74%)	0.50%	33.37% (3.86%)
	15	2.70% (0.73%)	0.50%	33.35% (3.80%)
LDA	1	2.82% (1.06%)	3.20%	19.22% (4.62%)
	5	2.74% (0.82%)	0.70%	19.14% (3.63%)
	9	2.72% (0.78%)	0.40%	19.13% (3.39%)
	11	2.72% (0.77%)	0.40%	19.09% (3.33%)
	15	2.72% (0.75%)	0.40%	19.10% (3.27%)

table S2. Results of SVMs in Simulation S2. The column meanings are the same as those of table S1.

Model	M	Type I error avg (sd)	Type I error violation rate	Type II error avg (sd)
LR	1	2.81% (1.04%)	3.00%	43.22% (9.40%)
	5	2.70% (0.77%)	0.90%	42.85% (7.53%)
	9	2.69% (0.73%)	0.50%	42.76% (7.25%)
	11	2.68% (0.72%)	0.70%	42.81% (7.20%)
	15	2.68% (0.71%)	0.50%	42.74% (7.15%)
LDA	1	2.81% (1.08%)	3.40%	34.11% (15.42%)
	5	2.72% (0.78%)	0.90%	33.25% (12.95%)
	9	2.72% (0.76%)	0.60%	33.05% (12.75%)
	11	2.72% (0.75%)	0.60%	33.10% (12.73%)
	15	2.71% (0.74%)	0.50%	33.15% (12.58%)

table S3. Results of RFs in Simulation S2. The column meanings are the same as those of table S1.

Model	<i>M</i>	Type I error avg (sd)	Type I error violation rate	Type II error avg (sd)
LR	1	2.79% (1.03%)	2.70%	45.08% (7.52%)
	5	2.39% (0.72%)	0.20%	45.05% (5.52%)
	9	2.32% (0.66%)	0.00%	44.97% (5.11%)
	11	2.31% (0.65%)	0.10%	44.96% (5.04%)
	15	2.29% (0.64%)	0.00%	44.93% (5.00%)
LDA	1	2.80% (1.08%)	3.30%	26.00% (7.16%)
	5	2.60% (0.75%)	0.20%	25.08% (4.92%)
	9	2.58% (0.73%)	0.40%	24.84% (4.60%)
	11	2.57% (0.72%)	0.40%	24.81% (4.52%)
	15	2.56% (0.70%)	0.30%	24.77% (4.42%)

table S4. Results of NB in Simulation S2. The column meanings are the same as those of table S1.

Model	<i>M</i>	Type I error avg (sd)	Type I error violation rate	Type II error avg (sd)
LR	1	2.79% (1.01%)	2.00%	35.31% (5.32%)
	5	2.69% (0.76%)	0.50%	35.18% (4.11%)
	9	2.68% (0.72%)	0.40%	35.13% (3.87%)
	11	2.67% (0.71%)	0.30%	35.15% (3.84%)
	15	2.67% (0.70%)	0.20%	35.11% (3.75%)
LDA	1	2.81% (1.05%)	3.50%	19.20% (4.58%)
	5	2.73% (0.82%)	0.80%	19.11% (3.62%)
	9	2.72% (0.77%)	0.40%	19.06% (3.35%)
	11	2.72% (0.75%)	0.40%	19.03% (3.28%)
	15	2.71% (0.75%)	0.40%	19.05% (3.24%)

table S5. Results of LDA in Simulation S2. The column meanings are the same as those of table S1.

Model	M	Type I error avg (sd)	Type I error violation rate	Type II error avg (sd)
LR	1	2.78% (1.02%)	3.00%	33.48% (5.19%)
	5	2.72% (0.80%)	1.40%	33.43% (4.14%)
	9	2.71% (0.75%)	0.80%	33.42% (3.90%)
	11	2.70% (0.74%)	0.50%	33.44% (3.85%)
	15	2.70% (0.73%)	0.30%	33.43% (3.82%)
LDA	1	2.81% (1.06%)	3.60%	19.15% (4.59%)
	5	2.73% (0.82%)	0.70%	19.11% (3.64%)
	9	2.72% (0.77%)	0.40%	19.05% (3.36%)
	11	2.72% (0.76%)	0.30%	19.04% (3.29%)
	15	2.71% (0.75%)	0.40%	19.05% (3.25%)

table S6. Results of AdaBoost in Simulation S2. The column meanings are the same as those of table S1.

Model	M	Type I error avg (sd)	Type I error violation rate	Type II error avg (sd)
LR	1	2.81% (1.02%)	2.90%	41.05% (6.20%)
	5	2.39% (0.70%)	0.00%	40.87% (4.51%)
	9	2.33% (0.65%)	0.00%	40.74% (4.19%)
	11	2.31% (0.64%)	0.00%	40.77% (4.14%)
	15	2.29% (0.62%)	0.00%	40.72% (4.06%)
LDA	1	2.80% (1.09%)	3.60%	24.39% (6.21%)
	5	2.55% (0.74%)	0.80%	23.39% (4.20%)
	9	2.51% (0.71%)	0.50%	23.21% (3.93%)
	11	2.50% (0.70%)	0.30%	23.17% (3.86%)
	15	2.49% (0.68%)	0.20%	23.11% (3.78%)

table S7. Description of variables used in real data application 1. The data and description are from the Early Warning Project (<http://www.earlywarningproject.com/>).

	Variable	Description	Values
Response	mkl.start.1	Onset of state-led mass killing episode in next year ($t + 1$)	{0, 1}
	reg.afr	US Dept State region: Sub-Saharan Africa	{0, 1}
	reg.eap	US Dept State region: East Asia and Pacific	{0, 1}
	reg.eur	US Dept State region: Europe and Eurasia	{0, 1}
	reg.mna	US Dept State region: Middle East and North Africa	{0, 1}
	reg.sca	US Dept State region: South and Central Asia	{0, 1}
	reg.amr	US Dept State region: Americas	{0, 1}
	mkl.ongoing	Any ongoing episodes of state-led mass killing	{0, 1}
	mkl.ever	Any state-led mass killing since WWII (cumulative)	{0, 1}
	countryage.ln	Country age, logged	[0, 7.712891]
	wdi.popsizeln	Population size, logged	[4.781189, 14.130934]
	imr.normed.ln	Infant mortality rate relative to annual global median, logged	[-2.721325, 1.798977]
	gdppcgrow.sr	Annual % change in GDP per capita, meld of IMF and WDI, square root	[-8.002944, 13.777878]
	wdi.trade.ln	Trade openness, logged	[-3.863269, 6.276150]
	ios.iccpr1	ICCPR 1st Optional Protocol signatory	{0, 1}
	postcw	Post-Cold War period (year ≥ 1991)	{0, 1}
Predictors	pol.cat.fl1	Autocracy (Fearon and Laitin)	{0, 1}
	pol.cat.fl2	Anocracy (Fearon and Laitin)	{0, 1}
	pol.cat.fl3	Democracy (Fearon and Laitin)	{0, 1}
	pol.cat.fl7	Other (Fearon and Laitin)	{0, 1}
	pol.durable.ln	Regime duration, logged (Polity)	[0, 5.332719]
	dis.l4pop.ln	Percent of population subjected to state-led discrimination, logged	[0, 4.49981]
	elf.ethnicc1	Ethnic fractionalization: low	{0, 1}
	elf.ethnicc2	Ethnic fractionalization: medium	{0, 1}
	elf.ethnicc3	Ethnic fractionalization: high	{0, 1}
	elf.ethnicc9	Ethnic fractionalization: missing	{0, 1}
	elc.eleth1	Salient elite ethnicity: majority rule	{0, 1}
	elc.eleth2	Salient elite ethnicity: minority rule	{0, 1}
	elc.eliti	Ruling elites espouse an exclusionary ideology	{0, 1}
	cou.tries5d	Any coup attempts in past 5 years ($(t - 4)$ to (t))	{0, 1}
	pit.sftpuhv12.10.ln	Sum of max annual magnitudes of PITF instability other than genocide from past 10 yrs ($(t - 9)$ to (t)), logged	[0, 4.51086]
	mev.regac.ln	Scalar measure of armed conflict in geographic region, logged	[0, 4.174387]
	mev.civtot.ln	Scale of violent civil conflict, logged	[0, 2.397895]

table S8. The performance of the NP umbrella algorithm in real data application 2. Given $\alpha = 0.1$ and $\delta = 0.1$, after randomly splitting the data into training data with a size 374 (3/4 of the observations) and test data with a size 124 (1/4 of the observations) for 1000 times, we calculate the empirical type I and type II errors on the test data. The violation rates of empirical type I errors (the percentage of empirical type I errors greater than α) of different approaches are summarized in the 2nd column. The averages and standard errors of empirical type II errors in percentages are summarized in the 3rd column. Three classification methods: penalized logistic regression (penLR), random forests (RF), and naïve Bayes (NB) are considered here. For each method, we considered three ways to choose the cutoff: *default*, the classical approach to minimize the overall classification error; *naïve*, the naïve approach to choose the threshold whose empirical type I error on the training data is no greater than and closest to α ; *NP*, the NP umbrella approach.

	Type I error avg (se)	Type I error violation rate	Type II error avg (se)
penLR (<i>default</i>)	9.35% (0.14%)	42%	3.75% (0.07%)
penLR (<i>naïve</i>)	20.53% (0.22%)	94%	1.25% (0.04%)
penLR (NP)	4.48% (0.11%)	7%	6.47% (0.09%)
RF (<i>default</i>)	12.45% (0.16%)	66%	5.72% (0.08%)
RF (<i>naïve</i>)	53.15% (0.24%)	100%	0.34% (0.02%)
RF (NP)	5.09% (0.12%)	11%	12.88% (0.12%)
NB (<i>default</i>)	10.61% (0.14%)	54%	14.43% (0.11%)
NB (<i>naïve</i>)	10.90% (0.17%)	57%	18.32% (0.62%)
NB (NP)	1.87% (0.11%)	4%	76.27% (1.16%)

table S9. Input information of the nproc package (version 2.0.9).

Argument	Description
x	an $n \times d$ design matrix with n observations and d covariates
y	an n dimensional vector with binary responses
method	<p>The base classification method. The following options are provided in nproc package version 2.0.9:</p> <ul style="list-style-type: none"> 'logistic': logistic regression 'penlog': penalized logistic regression with LASSO penalty, depending on the glmnet package version 2.0.5 'svm': support vector machines, depending on the e1071 package version 1.6.7. Arguments of the svm function can be passed from the npc and nproc functions 'randomforest': random forests, depending on the e1071 package version 1.6.7. Arguments of the randomforest function can be passed from the npc and nproc functions 'lda': linear discriminant analysis 'nb': naïve Bayes, depending on the e1071 package version 1.6.7 'ada': AdaBoost, depending on the ada package version 2.0-5
alpha	type I error upper bound, with default value 0.05
delta	the violation rate of type I error, with default value 0.05
split	the number of splits M , with default value 1