Journal of the American Statistical Association

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uasa20

Asymmetric Error Control Under Imperfect Supervision: A Label-Noise-Adjusted Neyman–Pearson Umbrella Algorithm

Shunan Yao, Bradley Rava, Xin Tong & Gareth James

To cite this article: Shunan Yao, Bradley Rava, Xin Tong & Gareth James (2022): Asymmetric Error Control Under Imperfect Supervision: A Label-Noise-Adjusted Neyman–Pearson Umbrella Algorithm, Journal of the American Statistical Association, DOI: 10.1080/01621459.2021.2016423

To link to this article: https://doi.org/10.1080/01621459.2021.2016423

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2022, VOL. 00, NO. 0, 1–13: Theory and Methods
https://doi.org/10.1080/01621459.2021.2016423

Taylor & Francis
Taylor & Francis Group

Asymmetric Error Control Under Imperfect Supervision: A Label-Noise-Adjusted Neyman–Pearson Umbrella Algorithm

Shunan Yao[a], Bradley Rava[b], Xin Tong[b], and Gareth James[b]

[a]Department of Mathematics, Dana and David Dornsife College of Letters, Arts and Sciences, University of Southern California, Los Angeles, CA; [b]Department of Data Sciences and Operations, Marshall School of Business, University of Southern California, Los Angeles, CA

ABSTRACT

Label noise in data has long been an important problem in supervised learning applications as it affects the effectiveness of many widely used classification methods. Recently, important real-world applications, such as medical diagnosis and cybersecurity, have generated renewed interest in the Neyman–Pearson (NP) classification paradigm, which constrains the more severe type of error (e.g., the Type I error) under a preferred level while minimizing the other (e.g., the Type II error). However, there has been little research on the NP paradigm under label noise. It is somewhat surprising that even when common NP classifiers ignore the label noise in the training stage, they are still able to control the Type I error with high probability. However, the price they pay is excessive conservativeness of the Type I error and hence a significant drop in power (i.e., 1− Type II error). Assuming that domain experts provide lower bounds on the corruption severity, we propose the first theory-backed algorithm that adapts most state-of-the-art classification methods to the training label noise under the NP paradigm. The resulting classifiers not only control the Type I error with high probability under the desired level but also improve power.

ARTICLE HISTORY
Received January 2021
Accepted November 2021

KEYWORDS
Classification; Label noise; Neyman–Pearson (NP) paradigm; Type I error; Umbrella algorithm

1. Introduction

Most classification methods assume a perfectly labeled training dataset. Yet, it is estimated that in real-world databases around five percent of labels are incorrect (Orr 1998; Redman 1998). Labeling errors might come from insufficient guidance to human coders, poor data quality, or human mistakes in decisions, among others (Brazdil and Konolige 1990; Hickey 1996; Brodley and Friedl 1999b). Specifically, in the medical field, a 2011 survey of more than 6000 physicians found that half said they encountered diagnostic errors at least once a month (MacDonald 2011). The existence of labeling errors in training data is often referred to as *label noise*, *imperfect labels*, or *imperfect supervision*. It belongs to a more general *data corruption* problem, which refers to "anything which obscures the relationship between description and class" (Hickey 1996).

The study of label noise in supervised learning has been a vibrant field in academia. *On the empirical front*, researchers have found that some statistical learning methods such as quadratic discriminant analysis (Lachenbruch 1979) and k-NN (Okamoto and Yugami 1997), can be greatly affected by label noise and have accuracy seriously reduced, while other methods, such as linear discriminant analysis (Lachenbruch 1966), are more label noise tolerant. Moreover, one can modify AdaBoost (Cao, Kwong, and Wang 2012), perceptron algorithm (Khardon and Wachman 2007), and neural networks (Sukhbaatar and Fergus 2014), so that they are more tolerant to label noise. Data cleansing techniques were also developed, such as in Guyon et al. (1996) and Brodley and Friedl (1999*a*). *On the theoretical front*, Natarajan et al. (2013) provided a guarantee for risk minimization in the setting of convex surrogates. Manwani and Sastry (2013) proved label noise tolerance of risk minimization for certain types of loss functions, and Ghosh, Manwani, and Sastry (2015) extended the result by considering more loss types. Liu and Tao (2016) proposed learning methods with importance-reweighting which can minimize the risk. Blanchard et al. (2016) studied intensely the *class-conditional corruption model*, a model that many works on label noise are based on. In particular, theoretical results about parameter estimation and consistency of classifiers under this model were presented in their work. Most recently, Cannings, Fan, and Samworth (2020) derived innovative theory of excess risk for general classifiers.

In many classification settings, one type of error may have far worse consequences than the other. For example, a biomedical diagnosis/prognosis that misidentifies a benign tumor as malignant will cause distress and potentially unnecessary medical procedures, but the alternative, where a malignant tumor is classified as benign, will have far worse outcomes. Other related predictive applications include cybersecurity and finance. Despite great advances in the label-noise classification literature, to our knowledge, no classifier has been constructed to deal with this asymmetry in error importance under label noise so as to control the level of the more severe error type.

CONTACT Xin Tong xint@marshall.usc.edu; Gareth James gareth@usc.edu Department of Data Sciences and Operations, Marshall School of Business, University of Southern California, Los Angeles, CA.

Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JASA.

In this article, we concentrate on the classification setting involving both mislabeled outcomes and error importance asymmetry. The Neyman–Pearson (NP) paradigm (Cannon et al. 2002; Scott and Nowak 2005), which controls the false-negative rate (FNR, a.k.a., Type I error[1]) under some desired level while minimizing the false-positive rate (FPR, a.k.a., Type II error), provides a natural approach to this problem. However, to the best of our knowledge, there has been no work that studies how label noise issues affect the control of the more severe FNR. We show that if one trains a standard NP classifier on corrupted labels (e.g., the NP umbrella algorithm in Tong, Feng, and Li 2018), then the actual achieved FNR is far below the control target, resulting in a very high, and undesirable, FPR.

This problem motivates us to devise a new label-noise-adjusted umbrella algorithm that corrects for the labeling errors to produce a lower FPR while still controlling the FNR. The construction of such an algorithm is challenging because we must identify the optimal correction level without any training data from the uncorrupted distribution. To address this challenge, we employ a common class-conditional noise model and derive the population-level difference between the Type I errors of the true and corrupted labels. Based on this difference, we propose a sample-based correction term that, even without observing any uncorrupted labels, can correctly adjust the NP umbrella algorithm to significantly reduce the FPR while still controlling the FNR.

Our approach has several advantages. First, it is the first theory-backed methodology in the label noise setting to control population-level Type I error (i.e., FNR) regarding the true labels. Concretely, we can show analytically that the new algorithm produces classifiers that have a high probability of controlling the FNR below the desired threshold with a FPR lower than that provided by the original NP umbrella algorithm. Second, when there are no labeling errors, our new algorithm reduces to the original NP algorithm. Finally, we demonstrate on both simulated and real-world data, that under the NP paradigm the new algorithm dominates the original unadjusted one and competes favorably against existing methods which handle label noise in classification.

The rest of the article is organized as follows. In Section 2, we introduce some notation and a corruption model to study the label noise. In Section 3, we demonstrate the ineffectiveness of the original NP umbrella algorithm under label noise and propose a new label-noise-adjusted version. The validity and the high-probability Type I error control property of the new algorithm are established in Section 4. Simulation and real data analysis are conducted in Section 5, followed by a Discussion section. All proofs, additional numerical results, and technical results are relegated to the supplementary materials.

2. Notation and Corruption Model

Let $(X, Y, \tilde{Y})$ be a random triplet, where $X \in \mathcal{X} \subset \mathbb{R}^d$ represents features, $Y \in \{0,1\}$ encodes the true class labels and $\tilde{Y} \in \{0,1\}$ the corrupted ones. Note that in our setting, we cannot observe Y; the observations come from $(X, \tilde{Y})$. Denote $X^0 \triangleq X|(Y=0)$ and $X^1 \triangleq X|(Y=1)$. Similarly, denote $\tilde{X}^0 \triangleq X|(\tilde{Y}=0)$ and $\tilde{X}^1 \triangleq X|(\tilde{Y}=1)$. Denote by $\mathbb{P}$ and $\mathbb{E}$ generic probability measure and expectation whose meanings depend on the context. For any Borel set $A \subset \mathcal{X}$, we denote

$$P_0(A) = \mathbb{P}(X \in A | Y=0)\,,\ P_1(A) = \mathbb{P}(X \in A | Y=1)\,,$$
$$\tilde{P}_0(A) = \mathbb{P}(X \in A | \tilde{Y}=0)\,,\ \tilde{P}_1(A) = \mathbb{P}(X \in A | \tilde{Y}=1)\,.$$

Then, we denote by $F_0, F_1, \tilde{F}_0$ and $\tilde{F}_1$ their respective distribution functions and by $f_0, f_1, \tilde{f}_0$ and $\tilde{f}_1$ the density functions, assuming they exist. Moreover, for a measurable function $T : \mathcal{X} \to \mathbb{R}$, we denote, for any $z \in \mathbb{R}$,

$$F_0^T(z) = P_0(T(X) \le z)\,,\ F_1^T(z) = P_1(T(X) \le z)\,,$$
$$\tilde{F}_0^T(z) = \tilde{P}_0(T(X) \le z)\,,\ \tilde{F}_1^T(z) = \tilde{P}_1(T(X) \le z)\,.$$

Since the effect of, and adjustment to, the label noise depend on the type and severity of corruption, we need to specify a corruption model to work with. Our choice for this work is the *class-conditional noise (contamination) model*, which is specified in the next assumption.

Assumption 1. There exist constants $m_0, m_1 \in [0,1]$ such that for any Borel set $A \subset \mathcal{X}$,

$$\tilde{P}_0(A) = m_0 P_0(A) + (1-m_0) P_1(A) \quad \text{and}$$
$$\tilde{P}_1(A) = m_1 P_0(A) + (1-m_1) P_1(A)\,. \tag{1}$$

Furthermore, assume $m_0 > m_1$ but both quantities can be unknown. Moreover, let $m_0^\#, m_1^\# \in [0,1]$ be known constants such that $m_0^\# \ge m_0$ and $m_1^\# \le m_1$.

Example 1 (An example of Assumption 1). Let $X^0 \sim \mathcal{N}(\mu_0, \sigma^2)$ and $X^1 \sim \mathcal{N}(\mu_1, \sigma^2)$, where $\mu_0, \mu_1 \in \mathbb{R}$ and $\sigma > 0$. Then $\tilde{F}_0(z) = m_0 \Phi(\frac{z-\mu_0}{\sigma}) + (1-m_0)\Phi(\frac{z-\mu_1}{\sigma})$ and $\tilde{F}_1(z) = m_1 \Phi(\frac{z-\mu_0}{\sigma}) + (1-m_1)\Phi(\frac{z-\mu_1}{\sigma})$, where $\Phi(\cdot)$ is the distribution function of $\mathcal{N}(0,1)$. With the choice of $\mu_0 = 0, \mu_1 = 1, \sigma = 1$, $m_0 = 0.9$, and $m_1 = 0.05$, the density functions $f_0, \tilde{f}_0, f_1$ and $\tilde{f}_1$ are plotted in Figure 1.

Note that Equation (1) specifies perhaps the simplest model for label noise in supervised learning. Here, m_0 and m_1 represent the severity of corruption levels. Concretely, m_0 can be interpreted as the proportion of true 0 observations among corrupted 0 observations, and m_1 the proportion of true 0 observations among corrupted 1 observations. The assumption $m_0 > m_1$ means that corrupted class 0 resembles true class 0 more than corrupted class 1 does, and that corrupted class 1 resembles true class 1 more than corrupted class 0 does. However, this assumption does not mean that corrupted class 0 resembles true class 0 more than it resembles true class 1 (i.e., $m_0 > 1/2$) or that corrupted class 1 resembles true class 1 more than it resembles true class 0 (i.e., $m_1 < 1/2$). Note that by the way our model is written, $m_0 = 1$ and $m_1 = 0$ correspond to the no label noise situation; as such, the roles of m_0 and m_1 are not symmetric. Hence, the assumptions $m_0^\# \ge m_0$ and $m_1^\# \le m_1$ mean that we know some *lower bounds* of the corruption levels.

[1] Note that Type I error in our work is defined to be the conditional probability of misclassifying a 0 instance as class 1. Moreover, we code the more severe class as class 0. In the disease diagnosis example, the disease class would be class 0.

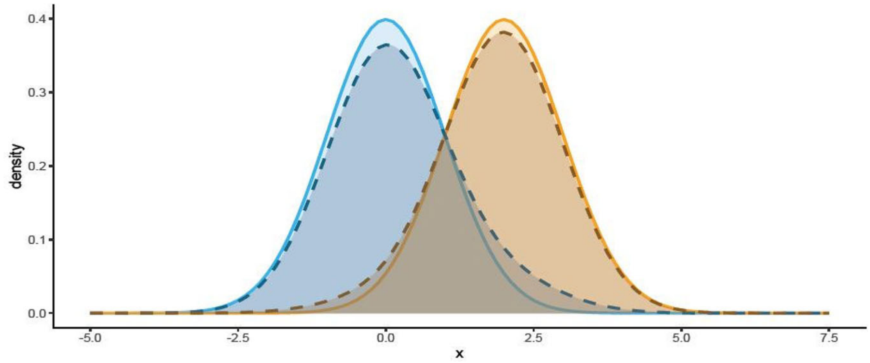

Figure 1. Density plots in Example 1. True (lighter and solid) and corrupted (darker and dashed).

The class-conditional label noise model has been widely adopted in the literature (Natarajan et al. 2013; Liu and Tao 2016; Blanchard et al. 2016). We note here that the assumption $m_0 > m_1$ aligns with the *total noise assumption* $\pi_0 + \pi_1 < 1$ in Blanchard et al. (2016) as π_0 and π_1 in their work correspond to $1 - m_0$ and m_1 in Assumption 1, respectively. In Natarajan et al. (2013) and Liu and Tao (2016), the label noise was modeled through the label flipping probabilities: $\mu_i = \mathbb{P}(\tilde{Y} = 1 - i | Y = i)$, $i = 0, 1$. This alternative formulation is related to our formulation via Bayes' rule. An in-depth study of the class-conditional label noise model, including mutual irreducibility and identifiability, was presented in Blanchard et al. (2016). Moreover, Blanchard et al. (2016) developed a noisy label trained classifier based on weighted cost-sensitive surrogate loss and established its consistency. Similarly, Natarajan et al. (2013) provided two methods to train classifiers, both relying on classification-calibrated surrogate loss; bounds for respective excess risks of these two methods were also given. Moreover, Liu and Tao (2016) proposed an importance reweighting method and extended the result in Natarajan et al. (2013) to all surrogate losses. Other than Blanchard et al. (2016), which briefly discussed the NP paradigm at the population level, in all aforementioned articles, though loss functions vary, the goal of classification is to minimize the overall risk. Our work focuses on the NP paradigm. Moreover, we focus on high probability control on the Type I error based on finite samples, in contrast to asymptotic results in the literature.

In this work, we take the perspective that the domain experts can provide under-estimates of corruption levels. In the literature, there are existing methods to estimate these levels. For example, Liu and Tao (2016) and Blanchard et al. (2016) developed methods to estimate π_i's and μ_i's, and showed consistency of their estimators. In numerical studies, we apply the method in Liu and Tao (2016) to estimate m_0 and m_1[2]. Numerical evidence shows that using these estimators in our proposed algorithm fails to establish a high probability control of the true Type I error. In fact, even using consistent and unbiased estimators of m_0 and m_1 as inputs of our proposed algorithm would not be able to control the true Type I error with high probability. One such case is demonstrated in Simulation 8 of the supplementary materials, where estimators for m_0 and m_1 are normally distributed and centered at the true values. To have high probability control on the true Type I error, we do need the "under-estimates" of corruption levels as in Assumption 1.

3. Methodology

In this section, we first formally introduce the Neyman–Pearson (NP) classification paradigm and review the NP umbrella algorithm (Tong, Feng, and Li 2018) for the uncorrupted label scenario (Section 3.1). Then we provide an example demonstrating that in the presence of label noise, naively implementing the NP umbrella algorithm leads to excessively conservative Type I error. That is, Type I error much smaller than the control target α. We analyze and capitalize on this phenomenon, and present new noise-adjusted versions of the NP umbrella algorithm, Algorithm 1 for known corruption levels (Section 3.2) and Algorithm $1^\#$ for unknown corruption levels (Section 3.3). Algorithm 1 can be considered as a special case of Algorithm $1^\#$: $m_0^\# = m_0$ and $m_1^\# = m_1$.

A few additional notations are introduced to facilitate our discussion. A classifier $\phi : \mathcal{X} \rightarrow \{0, 1\}$ maps from the feature space to the label space. The (population-level) Type I and II errors of $\phi(\cdot)$ regarding the *true* labels (a.k.a., true Type I and II errors) are respectively $R_0(\phi) = P_0(\phi(X) \neq Y)$ and $R_1(\phi) = P_1(\phi(X) \neq Y)$. The (population-level) Type I and II errors of $\phi(\cdot)$ regarding the *corrupted* labels (a.k.a., corrupted Type I and II errors) are, respectively, $\tilde{R}_0(\phi) = \tilde{P}_0(\phi(X) \neq \tilde{Y})$ and $\tilde{R}_1(\phi) = \tilde{P}_1(\phi(X) \neq \tilde{Y})$. In verbal discussion in this article, *Type I error* without any suffix refers to Type I error regarding the true labels.

3.1. The NP Umbrella Algorithm Without Label Noise

The NP paradigm (Cannon et al. 2002; Scott and Nowak 2005) aims to mimic the NP oracle

$$\phi_\alpha^* \in \underset{\phi:\, R_0(\phi) \leq \alpha}{\arg\min}\ R_1(\phi)\,,$$

where $\alpha \in (0, 1)$ is a user-specified level that reflects the priority toward the Type I error. In practice, with or without label noise, based on training data of finite sample size, it is usually impossible to ensure $R_0(\cdot) \leq \alpha$ almost surely. Instead, we aim to control the Type I error with high probability. Recently, the NP umbrella algorithm (Tong, Feng, and Li 2018) has attracted

[2] Note that though their method targets at μ_i's, estimates of m_i's in equation (1) can be constructed from those of μ_i's by the Bayes' theorem.

significant attention.[3] This algorithm works in conjunction with any score based classification method (e.g., logistic regression, support vector machines, or random forest) to compress a d-dimensional feature measurement to a one-dimensional score, and then threshold the score to classify. Specifically, *given a (score based) classification method*, the NP umbrella algorithm uses a model-free order statistics approach to decide the threshold, attaining a high probability control on Type I error with minimum Type II error *for that method*. Moreover, when coupling with a classification method that matches the underlying data distribution, the NP umbrella algorithm also achieves a diminishing excess Type II error, that is, $R_1(\hat{\phi}_\alpha) - R_1(\phi^*_\alpha) \to 0$. For example, Tong et al. (2020) showed that under a linear discriminant analysis (LDA) model, an LDA classifier with the score threshold determined by the NP umbrella algorithm satisfies both the control on Type I error and a diminishing excess Type II error.[4] Next we will review the implementation of the NP umbrella algorithm.

Let $\mathcal{S}^0 = \{X_j^0\}_{j=1}^{M_0}$ and $\mathcal{S}^1 = \{X_j^1\}_{j=1}^{M_1}$, respectively be the *uncorrupted* observations in classes 0 and 1, where M_0 and M_1 are the number of observations from each class.[5] Then, given a classification method (i.e., base algorithm, e.g., logistic regression), the NP umbrella algorithm is implemented by randomly splitting the class 0 data $\mathcal{S}^0$ into two parts: $\mathcal{S}^0_{\mathrm{b}}$ and $\mathcal{S}^0_{\mathrm{t}}$. The first part, $\mathcal{S}^0_{\mathrm{b}}$, together with $\mathcal{S}^1$, is used to train the *base* algorithm, while the second part $\mathcal{S}^0_{\mathrm{t}}$ determines the *threshold* candidates. Specifically, we train a base algorithm with scoring function $\hat{T}(\cdot)$ (e.g., the sigmoid function in logistic regression) using $\mathcal{S}^0_{\mathrm{b}} \cup \mathcal{S}^1$, apply $\hat{T}(\cdot)$ on $\mathcal{S}^0_{\mathrm{t}}$ ($|\mathcal{S}^0_{\mathrm{t}}| = n$) to get threshold candidates $\{t_1, \ldots, t_n\}$, and sort them in an increasing order $\{t_{(1)}, \ldots, t_{(n)}\}$. Then the NP umbrella algorithm proposes classifier $\hat{\phi}_{k_*}(\cdot) = \mathbb{I}\{\hat{T}(\cdot) > t_{(k_*)}\}$, where

$$k_* = \min\left\{k \in \{1, \ldots, n\} : \sum_{j=k}^{n} \binom{n}{j} (1-\alpha)^j \alpha^{(n-j)} \le \delta \right\}, \tag{2}$$

in which δ is a user-specified tolerance probability of the Type I error exceeding α. The key to this approach is that Tong, Feng, and Li (2018) established, for all $\hat{\phi}_k(\cdot) = \mathbb{I}\{\hat{T}(\cdot) > t_{(k)}\}$ where $k \in \{1, \ldots, n\}$, it holds $\mathbb{P}(R_0(\hat{\phi}_k) > \alpha) \le \sum_{j=k}^{n} \binom{n}{j} (1-\alpha)^j \alpha^{(n-j)}$, where $\mathbb{P}$ corresponds to random draws of $\mathcal{S}^0$ and $\mathcal{S}^1$, as well as potential randomness in the classification method (e.g., random forest), and the inequality becomes an equality when $\hat{T}$ is continuous almost surely. In view of this inequality and the definition for k_*, we have $\mathbb{P}(R_0(\hat{\phi}_{k_*}) > \alpha) \le \delta$, and $\hat{\phi}_{k_*}$ achieves the smallest type II error among the $\hat{\phi}_k$'s that respect the $(1-\delta)$ probability control of the Type I error. We call this algorithm the *original* NP umbrella algorithm to contrast with the newly developed versions.

3.2. Algorithm 1: Label-Noise-Adjusted NP Umbrella Algorithm With Known Corruption Levels

Returning to our errors in labels problem leads one to ask what would happen if we were to directly apply the original NP umbrella algorithm to the label noise setting? The results are mixed. While this algorithm successfully controls Type I error, it tends to be massively conservative, leading to very low Type I errors, but high Type II errors. The next example illustrates this phenomenon.

Example 2. Let $X^0 \sim \mathcal{N}(0, 1)$ and $X^1 \sim \mathcal{N}(2, 1)$, $m_0 = 0.85$, $m_1 = 0.15$, $\alpha = 0.05$ and $\delta = 0.05$. For simplicity, we use the identity scoring function: $\hat{T}(X) = X$. We generate $N \in \{200, 500, 1000, 2000\}$ corrupted class 0 observations and train a classifier $\hat{\phi}_{k_*}(\cdot)$ based on them. Due to normality, we can analytically calculate the Type I and II errors regarding the true labels. The above steps are repeated 1,000 times for every value of N to graph the violin plots of both errors as shown in the left panel of Figure 2. Clearly, all the achieved true Type I errors are much lower than the control target α and true Type II errors are very high[6].

The phenomenon illustrated in the left panel of Figure 2 is not a contrived one. Indeed, under the class-conditional noise model (i.e., Assumption 1), at the same threshold level, the tail probability of corrupted class 0 is greater than that of true class 0 since the corrupted 0 distribution is a mixture of true 0 and 1 distributions. Figure 3 provides further illustration. In this figure, the black vertical line ($x = 2.52$) marks the threshold of the classifier $\mathbb{I}\{X > 2.52\}$ whose corrupted Type I error (i.e., the right tail probability under the orange dashed curve) is 0.05. In contrast, its true Type I error (i.e., the right-tail probability under the blue solid curve) is much smaller.

The above observation motivates us to create new label-noise-adjusted NP umbrella algorithms by carefully studying the discrepancy between true and corrupted Type I errors, whose population-level relation is channeled by the class-conditional noise model and can be estimated based on data with corrupted labels alone. We will first develop a version for known corruption levels (i.e., Algorithm 1) and then a variant for unknown corruption levels (i.e., Algorithm $1^{\#}$). Although the latter variant is suitable for most applications, we believe that presenting first the known corruption level version streamlines the reasoning and presentation.

For methodology and theory development, we assume the following sampling scheme. Let $\tilde{\mathcal{S}}^0 = \{\tilde{X}_j^0\}_{j=1}^{N_0}$ be *corrupted* class 0 observations and $\tilde{\mathcal{S}}^1 = \{\tilde{X}_j^1\}_{j=1}^{N_1}$ *corrupted* class 1 ones. The sample sizes N_0 and N_1 are considered to be nonrandom numbers, and we assume that all observations in $\tilde{\mathcal{S}}^0$ and $\tilde{\mathcal{S}}^1$ are independent. Then, we divide $\tilde{\mathcal{S}}^0$ into *three* random disjoint non-empty subsets. The first two parts $\tilde{\mathcal{S}}^0_{\mathrm{b}}$ and $\tilde{\mathcal{S}}^0_{\mathrm{t}}$ are used to train the *base* algorithm and determine the *threshold*

[3] At the time of writing, the NP umbrella package has been downloaded over 35,000 times.

[4] These two properties together were coined as the *NP oracle inequalities* by Rigollet and Tong (2011). Classifiers with these properties were constructed with nonparametric assumptions in Tong (2013) and Zhao et al. (2016).

[5] Note that the uncorrupted data $\mathcal{S}^0$ and $\mathcal{S}^1$ are not available in our present label noise setting and we only use them here for review purposes.

[6] To make a contrast, we also plot in the right panel of Figure 2, the true Type I and II errors of $\hat{\phi}_{k^*}(\cdot)$, the classifier constructed by the label-noise-adjusted NP umbrella algorithm with known corruption levels to be introduced in the next section. The details to generate $\hat{\phi}_{k^*}(\cdot)$'s are skipped here, except we reveal that corrupted class 1 observations, in addition to the corrupted class 0 observations, are also needed to construct the thresholds.

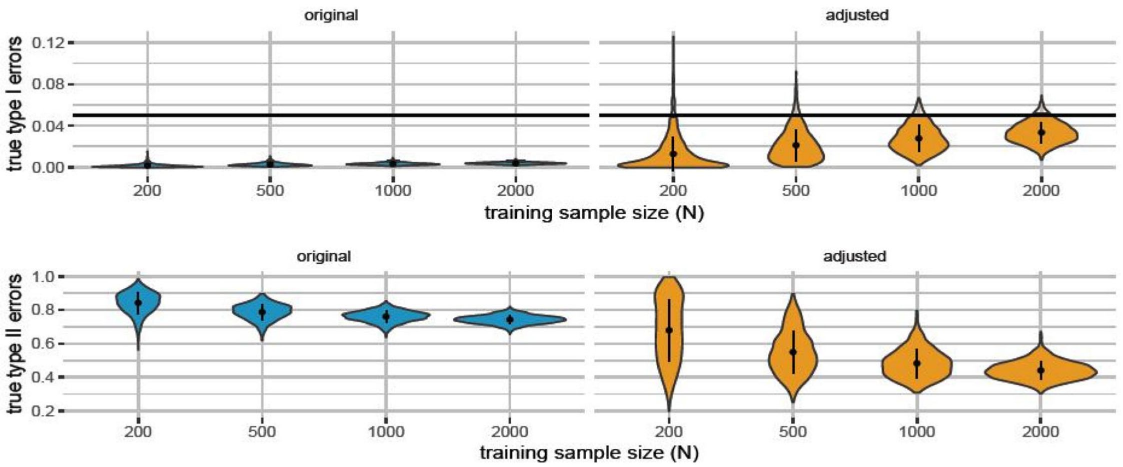

Figure 2. The original NP umbrella algorithm vs. a label-noise-adjusted version for Example 2. The plots in the left panel (blue) are the true Type I and II errors for the original NP umbrella algorithm. The plots in the right panel (orange) are the true Type I and II errors for the label-noise-adjusted NP umbrella algorithm with known corruption levels. The black dot and vertical bar in every violin represent mean and standard deviation, respectively. In the top row, the horizontal black line is $\alpha = 0.05$ and the boundaries between lighter and darker color in each violin plot mark the $1 - \delta = 95\%$ quantiles.

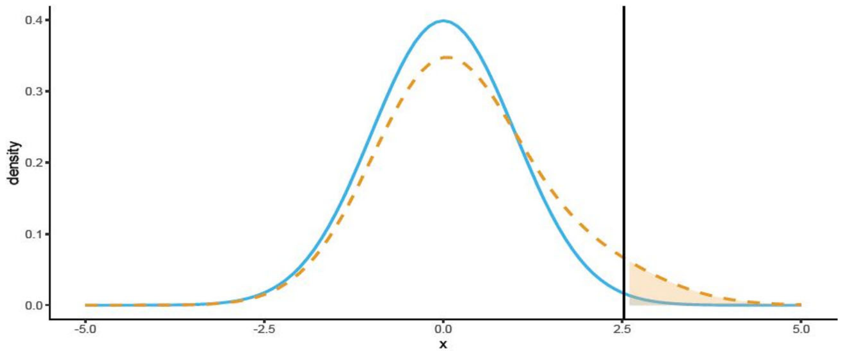

Figure 3. The blue solid curve is the density of true class 0 (i.e., $\mathcal{N}(0,1)$) and the orange dashed curve is the density of corrupted class 0 (i.e., a mixture of $\mathcal{N}(0,1)$ and $\mathcal{N}(2,1)$ with $m_0 = 0.85$). The black vertical line marks the threshold of the classifier $\mathbb{1}\{X > 2.52\}$ whose corrupted Type I error is 0.05.

candidates, respectively. The third part $\tilde{\mathcal{S}}^0_{\rm e}$ is used to *estimate* a correction term to account for the label noise. Similarly, we randomly divide $\tilde{\mathcal{S}}^1$ into *two* disjoint nonempty subsets: $\tilde{\mathcal{S}}^1_{\rm b}$ and $\tilde{\mathcal{S}}^1_{\rm e}$.

Let $\hat{T}(\cdot)$ be a scoring function trained on $\tilde{\mathcal{S}}_{\rm b} = \tilde{\mathcal{S}}^0_{\rm b} \cup \tilde{\mathcal{S}}^1_{\rm b}$. We apply $\hat{T}(\cdot)$ to elements in $\tilde{\mathcal{S}}^0_{\rm t}$ and sort them in an increasing order: $\{t_{(1)}, \ldots, t_{(n)}\}$, where $n = |\tilde{\mathcal{S}}^0_{\rm t}|$.[7] These will serve as the threshold candidates, just as in the original NP umbrella algorithm. However, instead of k_*, the label-noise-adjusted NP umbrella algorithm with known corruption levels will take the order k^* defined by

$$k^* = \min\{k \in \{1, \ldots, n\} : \alpha_{k,\delta} - \hat{D}^+(t_{(k)}) \leq \alpha\},$$

where $\alpha_{k,\delta}$[8] satisfies

$$\sum_{j=k}^{n} \binom{n}{j} \alpha_{k,\delta}^{n-j} (1 - \alpha_{k,\delta})^j = \delta\,, \qquad (3)$$

$\hat{D}^+(\cdot) = \hat{D}(\cdot) \vee 0 := \max(\hat{D}(\cdot), 0)$ and $\hat{D}(\cdot) = \frac{1-m_0}{m_0-m_1}\left(\hat{\tilde{F}}_0^{\hat{T}}(\cdot) - \hat{\tilde{F}}_1^{\hat{T}}(\cdot)\right)$, in which $\hat{\tilde{F}}_0^{\hat{T}}(\cdot)$ and $\hat{\tilde{F}}_1^{\hat{T}}(\cdot)$ are empirical estimates of $\tilde{F}_0^{\hat{T}}(\cdot)$ and $\tilde{F}_1^{\hat{T}}(\cdot)$ based on $\tilde{\mathcal{S}}^0_{\rm e}$ and $\tilde{\mathcal{S}}^1_{\rm e}$, respectively.

The entire construction process of $\hat{\phi}_{k^*}(\cdot) = \mathbb{1}\{\hat{T}(\cdot) > t_{(k^*)}\}$ is summarized and detailed in Algorithm 1. In this algorithm, to solve $\alpha_{k,\delta}$, we use a binary search subroutine (in the supplementary materials B) on the function $x \mapsto \sum_{j=k}^{n} \binom{n}{k} x^{n-j}(1-x)^j$, leveraging its strict monotone decreasing property in x. Interested readers are referred to the proof of Lemma 5 in the supplementary materials for further reasoning. Currently, we randomly split $\tilde{\mathcal{S}}^0$ and $\tilde{\mathcal{S}}^1$ respectively into three and two equal sized subgroups. An optimal splitting strategy could be a subject for future research.

The key to the new algorithm is $\hat{D}^+(\cdot)$, which adjusts for the label corruption. Indeed, the original NP umbrella algorithm can be seen as a special case of our approach where $\hat{D}^+(\cdot) = 0$. The numerical advantage of the new algorithm is demonstrated in the right panel of Figure 2 and in Section 5. We will prove in the next section that the *label-noise-adjusted NP classifier* $\hat{\phi}_{k^*}(\cdot) = \mathbb{1}\{\hat{T}(\cdot) > t_{(k^*)}\}$ controls true Type I error with high probability while avoiding the excessive conservativeness of the original NP umbrella algorithm. Note that in contrast to the deterministic order k_* in the original NP umbrella algorithm, the new order k^* is random, calling for much more involved technicalities to establish the theoretical properties of $\hat{\phi}_{k^*}(\cdot)$.

[7] In supplementary materials A, we summarize the notations related to the sampling scheme for the readers' convenience.

[8] The existence and uniqueness of $\alpha_{k,\delta}$ are ensured by Lemma 5 in the supplementary materials.

Algorithm 1: Label-noise-adjusted NP Umbrella Algorithm with known corruption levels

```
Input : $\tilde{\mathcal{S}}^0$: sample of corrupted 0 observations
        $\tilde{\mathcal{S}}^1$: sample of corrupted 1 observations
        $\alpha$: Type I error upper bound, $0 < \alpha < 1$
        $\delta$: Type I error violation rate target, $0 < \delta < 1$
        $m_0$: probability of a corrupted class 0 sample being of true class 0
        $m_1$: probability of a corrupted class 1 sample being of true class 0
$\tilde{\mathcal{S}}^0_{\mathrm{b}}, \tilde{\mathcal{S}}^0_{\mathrm{t}}, \tilde{\mathcal{S}}^0_{\mathrm{e}} \leftarrow$ random split on $\tilde{\mathcal{S}}^0$
$\tilde{\mathcal{S}}^1_{\mathrm{b}}, \tilde{\mathcal{S}}^1_{\mathrm{e}} \leftarrow$ random split on $\tilde{\mathcal{S}}^1$
$\tilde{\mathcal{S}}_{\mathrm{b}} \leftarrow \tilde{\mathcal{S}}^1_{\mathrm{b}} \cup \tilde{\mathcal{S}}^0_{\mathrm{b}}$ ;  // combine $\tilde{\mathcal{S}}^0_{\mathrm{b}}$ and $\tilde{\mathcal{S}}^1_{\mathrm{b}}$ as $\tilde{\mathcal{S}}_{\mathrm{b}}$
$\hat{T}(\cdot) \leftarrow$ base classification algorithm$(\tilde{\mathcal{S}}_{\mathrm{b}})$ ;  // train a scoring function on $\tilde{\mathcal{S}}_{\mathrm{b}}$
$\mathcal{T}_{\mathrm{t}} = \{t_1, t_2, \ldots, t_n\} \leftarrow \hat{T}(\tilde{\mathcal{S}}^0_{\mathrm{t}})$ ;  // apply $\hat{T}$ to every entry in $\tilde{\mathcal{S}}_{\mathrm{t}}$
$\{t_{(1)}, t_{(2)}, \ldots, t_{(n)}\} \leftarrow$ sort$(\mathcal{T}_{\mathrm{t}})$
$\mathcal{T}^0_{\mathrm{e}} \leftarrow \hat{T}(\tilde{\mathcal{S}}^0_{\mathrm{e}})$
$\mathcal{T}^1_{\mathrm{e}} \leftarrow \hat{T}(\tilde{\mathcal{S}}^1_{\mathrm{e}})$ ;  // apply $\hat{T}$ to all elements in $\tilde{\mathcal{S}}^0_{\mathrm{e}}$ and $\tilde{\mathcal{S}}^1_{\mathrm{e}}$
for $k$ in $\{1, \ldots, n\}$ do
    $\alpha_{k,\delta} \leftarrow$ BinarySearch$(\delta, k, n)$ ;  // compute $\alpha_{k,\delta}$ through binary search
    $\hat{\tilde{F}}^{\hat{T}}_0(t_{(k)}) \leftarrow \left|\mathcal{T}^0_{\mathrm{e}}\right|^{-1} \cdot \sum_{t \in \mathcal{T}^0_{\mathrm{e}}} \mathbb{I}\{t \le t_{(k)}\}$
    $\hat{\tilde{F}}^{\hat{T}}_1(t_{(k)}) \leftarrow \left|\mathcal{T}^1_{\mathrm{e}}\right|^{-1} \cdot \sum_{t \in \mathcal{T}^1_{\mathrm{e}}} \mathbb{I}\{t \le t_{(k)}\}$ ;
    // compute the empirical distributions
    $\hat{D}(t_{(k)}) \leftarrow \frac{1-m_0}{m_0-m_1}\left(\hat{\tilde{F}}^{\hat{T}}_0(t_{(k)}) - \hat{\tilde{F}}^{\hat{T}}_1(t_{(k)})\right)$ ;
    // compute an estimate of $\tilde{R}_0 - R_0$
    $\hat{D}^+(t_{(k)}) \leftarrow \hat{D}(t_{(k)}) \vee 0$ ;  // if $\hat{D}(t_{(k)})$ is negative, then set it to 0
end
$k^* \leftarrow \min\{k \in \{1, \ldots, n\} : \alpha_{k,\delta} - \hat{D}^+(t_{(k)}) \le \alpha\}$ ;
// select the order
$\hat{\phi}_{k^*}(\cdot) \leftarrow \mathbb{I}\{\hat{T}(\cdot) > t_{(k^*)}\}$ ;  // construct an NP classifier
Output: $\hat{\phi}_{k^*}(\cdot)$
```

3.3. Algorithm 1$^\#$: Label-Noise-Adjusted NP Umbrella Algorithm With Unknown Corruption Levels

For most applications in practice, accurate corruption levels m_0 and m_1 are inaccessible. To address this, we propose Algorithm 1$^\#$, a simple variant of Algorithm 1 that replaces m_0 and m_1 with estimates $m_0^\#$ and $m_1^\#$. In all other respects, the two algorithms are identical. Specifically, when estimating $\tilde{R}_0 - R_0$, Algorithm 1$^\#$ uses $\hat{D}_\#(t_{(k)}) = \frac{1-m_0^\#}{m_0^\# - m_1^\#}\left(\hat{\tilde{F}}^{\hat{T}}_0(t_{(k)}) - \hat{\tilde{F}}^{\hat{T}}_1(t_{(k)})\right)$ and $\hat{D}^+_\#(t_{(k)}) = \hat{D}_\#(t_{(k)}) \vee 0$. Then, Algorithm 1$^\#$ delivers the NP classifier $\hat{\phi}_{k^*_\#}(\cdot) = \mathbb{I}\{\hat{T}(\cdot) > t_{(k^*_\#)}\}$, where $k^*_\# = \min\{k \in \{1, \ldots, n\} : \alpha_{k,\delta} - \hat{D}^+_\#(t_{(k)}) \le \alpha\}$. Due to the similarity with Algorithm 1, we do not re-produce the other steps of Algorithm 1$^\#$ to write it out in a full algorithm format.

Rather than supplying unbiased estimates for m_0 and m_1, we will demonstrate that it is important that $m_0^\#$ and $m_1^\#$ are under-estimates of the corruption levels (i.e., $m_0^\# \ge m_0$ and $m_1^\# \le m_1$ as in Assumption 1). In this work, we assume that domain experts supply these under-estimates. While it would be unrealistic to assume that these experts know m_0 and m_1 exactly, in many scenarios one can provide accurate bounds on these quantities. It would be interesting to investigate data-driven estimators that have such a property for future work.

4. Theory

In this section, we first elaborate the rationale behind Algorithm 1 (Section 4.1), and then show that under a few technical conditions, this new algorithm induces well-defined classifiers whose Type I errors are bounded from above by the desired level with high probability (Section 4.2). Then we establish a similar result for its unknown-corruption-level variant, Algorithm 1$^\#$ (Section 4.3).

4.1. Rationale Behind Algorithm 1

Proposition 1. Let $\hat{T}(\cdot)$ be a scoring function (e.g., sigmoid function in logistic regression) trained on $\tilde{\mathcal{S}}_b$. Applying $\hat{T}(\cdot)$ to every element in $\tilde{\mathcal{S}}^0_t$, we get a set of scores. Order these scores and denote them by $\{t_{(1)}, t_{(2)}, \ldots, t_{(n)}\}$, in which $t_{(1)} \le t_{(2)} \le \ldots \le t_{(n)}$. Then, for any $\alpha \in (0,1)$ and $k \in \{1, 2, \ldots, n\}$, the classifier $\hat{\phi}_k(\cdot) = \mathbb{I}\{\hat{T}(\cdot) > t_{(k)}\}$ satisfies

$$\mathbb{P}\left(\tilde{R}_0(\hat{\phi}_k) > \alpha\right) \le \sum_{j=k}^{n} \binom{n}{j} (1-\alpha)^j \alpha^{(n-j)},$$

in which $\mathbb{P}$ is regarding the randomness in all training observations, as well as additional randomness if we adopt certain random classification methods (e.g., random forest). Moreover, when $\hat{T}(\cdot)$ is continuous almost surely, the above inequality obtains the equal sign.

Recall that $\tilde{R}_0(\cdot)$ denotes Type I error regarding the *corrupted* labels. We omit a proof for Proposition 1 as it follows the same proof as its counterpart in Tong, Feng, and Li (2018). For $\alpha, \delta \in (0,1)$, recall that the original NP umbrella algorithm selects $k_* = \min\{k \in \{1, \ldots, n\} : \sum_{j=k}^{n} \binom{n}{j}(1-\alpha)^j \alpha^{(n-j)} \le \delta\}$. The smallest k among all that satisfy $\sum_{j=k}^{n} \binom{n}{j}(1-\alpha)^j \alpha^{(n-j)} \le \delta$ is desirable because we also wish to minimize the Type II error. There is a sample size requirement for this order statistics approach to work because a finite order k_* should exist. Precisely, an order statistics approach works if the last order does; that is $(1-\alpha)^n \le \delta$. This translates to Assumption 2 on n, the sample size of $\tilde{\mathcal{S}}^0_t$. This is a mild requirement. For instance, when $\alpha = \delta = 0.05$, n should be at least 59.

Assumption 2. $n \ge \lceil \log \delta / \log(1-\alpha) \rceil$, in which $\lceil \cdot \rceil$ denotes the ceiling function.

In view of Proposition 1, the choice of k_* guarantees $\mathbb{P}\left(\tilde{R}_0(\hat{\phi}_{k_*}) \leq \alpha\right) \geq 1-\delta$. In other words, if we were to ignore the label noise presence and apply the original NP umbrella algorithm, the Type I error regarding the *corrupted* labels, $\tilde{R}_0$, is controlled under level α with probability at least $1-\delta$. Moreover, the achieved $\tilde{R}_0$ is usually not far from α when the sample size n is much larger than the lower bound requirement. However, this is not our main target; what we really want is to control R_0. Example 2 in Section 3.1 convincingly demonstrates that in the presence of label noise, the achieved R_0 after naive implementation of the original NP umbrella algorithm can be much lower than the control target α. This is no exception. To aid in analyzing the gap between R_0 and $\tilde{R}_0$, we make the following assumption.

Assumption 3. The scoring function $\hat{T}$ is trained such that $\tilde{F}_0^{\hat{T}}(z) > \tilde{F}_1^{\hat{T}}(z)$ for all $z \in \mathbb{R}$ with probability at least $1-\delta_1(n_b)$, where $n_b = |\tilde{\mathcal{S}}_b|$ and $\delta_1(n_b)$ converges to 0 as n_b goes to infinity.

Loosely, Assumption 3 means that the scoring function trained on corrupted data still has the "correct direction." For any classifier of the form $\hat{\phi}_c(\cdot) = \mathbb{1}\{\hat{T}(\cdot) > c\}$, Assumption 3 implies that with probability at least $1-\delta_1(n_\mathrm{b})$, $\tilde{P}_0(\hat{\phi}_c(X) = 0) > \tilde{P}_1(\hat{\phi}_c(X) = 0)$, which means that a corrupted class 0 observation is more likely to be classified as 0 than a corrupted class 1 observation is. Interested readers can find a concrete example that illustrates this mild assumption in the supplementary material C. Now we are ready to describe the discrepancy between R_0 and $\tilde{R}_0$.

Lemma 1. Let $\hat{T}$ be a scoring function trained on $\tilde{\mathcal{S}}_b$ and $\hat{\phi}_c(\cdot) = \mathbb{1}\{\hat{T}(\cdot) > c\}$ be a classifier that thresholds the scoring function at $c \in \mathbb{R}$. Denote $D(c) = \tilde{R}_0(\hat{\phi}_c) - R_0(\hat{\phi}_c)$. Then, under Assumptions 1–3, for given α and δ, it holds that

$$\mathbb{P}\left(\inf_{c\in\mathbb{R}} D(c) \geq 0\right) \geq 1-\delta_1(n_b) \quad \text{and}$$
$$\mathbb{P}\left(R_0(\hat{\phi}_{k_*}) > \alpha - D(t_{(k_*)})\right) \leq \delta + \delta_1(n_b)\,,$$

where k_* and δ are related via Equation (2). Moreover, we have

$$D(c) = M\left(\tilde{F}_0^{\hat{T}}(c) - \tilde{F}_1^{\hat{T}}(c)\right), \qquad (4)$$

where $M = (1-m_0)(m_0-m_1)^{-1}$.

Note that $D(c)$ measures the discrepancy between the *corrupted* Type I error and the *true* Type I error of the classifier $\hat{\phi}_c(\cdot)$. Lemma 1 implies that with high probability, $\hat{\phi}_{k_*}(\cdot)$ has R_0, the Type I error regarding *true* labels, under a level that is smaller than the target value α, and that the gap is measured by $D(t_{(k_*)})$. It is important to note that $D(c)$ is solely a function of the distributions of the corrupted data, and does not require any knowledge of the uncorrupted scores, so we are able to estimate this quantity from our observed data.

As argued previously, excessive conservativeness in Type I error is not desirable because it is usually associated with a high Type II error. Therefore, a new NP umbrella algorithm should adjust to the label noise, so that the resulting classifier respects the true Type I error control target, but is not excessively conservative. Motivated by Lemma 1, our central plan is to choose some less conservative (i.e., smaller) order than that in the original NP umbrella algorithm, in view of the difference between R_0 and $\tilde{R}_0$. Recall that $\delta \in (0,1)$ is the target Type I error violation rate. In the presence of label noise, we do not expect to control at this precise violation rate, but just some number around it.

For any $\hat{\phi}_k(\cdot)$, under Assumptions 1–3, Lemma 1 implies $\tilde{R}_0(\hat{\phi}_k) \geq R_0(\hat{\phi}_k)$ with probability at least $1-\delta_1(n_\mathrm{b})$. Note that the $\delta_1(n_\mathrm{b})$ term is small and asymptotically 0; we will ignore it in this section when motivating our new strategy. With this simplification, $\tilde{R}_0(\hat{\phi}_k)$ is always greater than $R_0(\hat{\phi}_k)$, as illustrated in Figure 4. The definition of $\alpha_{k,\delta}$ in equation (3) and Proposition 1 imply with probability at least $1-\delta$, $\alpha_{k,\delta} \geq \tilde{R}_0(\hat{\phi}_k)$, which corresponds to the green region (the region on the right) in Figure 4. Since we only need $1-\delta$ probability control on R_0, it suffices to control R_0 corresponding to this region. Combining the results $\alpha_{k,\delta} \geq \tilde{R}_0(\hat{\phi}_k)$ and $\tilde{R}_0(\hat{\phi}_k) \geq R_0(\hat{\phi}_k)$, we have the inequalities $\alpha_{k,\delta} \geq \alpha_{k,\delta} - D(t_{(k)}) \geq R_0(\hat{\phi}_k)$ on our interested region (Recall $D(t_{(k)}) = \tilde{R}_0(\hat{\phi}_k) - R_0(\hat{\phi}_k)$). By the previous argument, $\alpha_{k,\delta}$ can be used as an upper bound for R_0, but to have a good Type II error, a better choice is clearly the smaller $\alpha_{k,\delta} - D(t_{(k)})$. So if $D(t_{(k)})$ were a known quantity, we can set the order to be $\tilde{k}^* = \min\{k \in \{1\ldots,n\} : \alpha_{k,\delta} - D(t_{(k)}) \leq \alpha\}$ and propose a classifier $\hat{\phi}_{\tilde{k}^*}(\cdot) = \mathbb{1}\{\hat{T}(\cdot) > t_{(\tilde{k}^*)}\}$. This is to be compared with the order k_* chosen by the original NP umbrella algorithm, which can be equivalently expressed as $k_* = \min\{k \in \{1\ldots,n\} : \alpha_{k,\delta} \leq \alpha\}$ (Lemma 5 in the supplementary materials). Then we have $\tilde{k}^* \leq k_*$, and so $\hat{\phi}_{\tilde{k}^*}(\cdot)$ is less conservative than $\hat{\phi}_{k_*}(\cdot)$ in terms of Type I error.

However, $\hat{\phi}_{\tilde{k}^*}(\cdot)$ is not accessible because D is unknown. Instead, we estimate D by replacing $\tilde{F}_0^{\hat{T}}$ and $\tilde{F}_1^{\hat{T}}$ in (4) with their empirical distributions $\hat{\tilde{F}}_0^{\hat{T}}$ and $\hat{\tilde{F}}_1^{\hat{T}}$, which are calculated using $\tilde{\mathcal{S}}_\mathrm{e}^0$ and $\tilde{\mathcal{S}}_\mathrm{e}^1$, iid samples from the corrupted 0 and 1 observations. Note that these estimates are independent of $\tilde{\mathcal{S}}_\mathrm{b}$ and $\tilde{\mathcal{S}}_\mathrm{t}^0$. For a given $\hat{T}$, we define for every $c \in \mathbb{R}$,

$$\hat{D}(c) = \frac{1-m_0}{m_0-m_1}\left(\hat{\tilde{F}}_0^{\hat{T}}(c) - \hat{\tilde{F}}_1^{\hat{T}}(c)\right) \quad \text{and}$$

$$k^{**} = \min\{k \in \{1,\ldots,n\} : \alpha_{k,\delta} - \hat{D}(t_{(k)}) \leq \alpha - \varepsilon\}\,,$$

in which a small $\varepsilon > 0$ is introduced to compensate for the randomness of $\hat{D}$ in the theory proofs. For simulation and real data, we actually just use $\varepsilon = 0$. Finally, *the proposed new label-noise-adjusted NP classifier with known corruption levels is* $\hat{\phi}_{k^*}(\cdot) = \mathbb{1}\{\hat{T}(\cdot) > t_{(k^*)}\}$, in which k^* is a small twist from k^{**} by replacing $\hat{D}$ with its positive part. The construction of $\hat{\phi}_{k^*}(\cdot)$ was detailed in Algorithm 1.

We have two comments on the implementation of Algorithm 1. First, though the ε compensation for the randomness is necessary for the theory proof, our empirical results suggest almost identical performance between $\varepsilon = 0$ relative to any small ε, so we recommend setting ε to 0 for simplicity, and we do not use the ε compensation in Algorithm 1. Second,

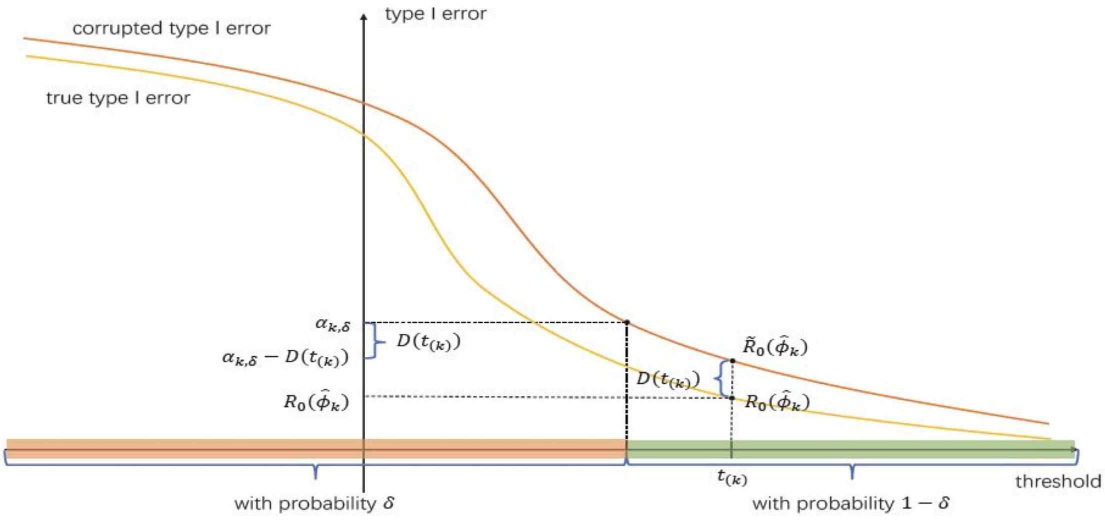

Figure 4. A cartoon illustration of $1-\delta$ probability upper bound of Type I error.

in the order selection criterion of k^* in Algorithm 1, we use $\hat{D}^+ = \hat{D} \vee 0 := \max(\hat{D}, 0)$ instead of $\hat{D}$, because empirically, although highly unlikely, $\hat{D}$ can be negative, which results in $\min\{k \in \{1,\dots,n\} : \alpha_{k,\delta} - \hat{D}(t_{(k)}) \le \alpha\} \ge \min\{k \in \{1,\dots,n\} : \alpha_{k,\delta} \le \alpha\}$. In this case, the new order could be greater than k_*. Since we aim to reduce the conservativeness of the original NP umbrella algorithm, the possibility of $k^* \ge k_*$ will reverse this effort and worsen the conservativeness. To solve this issue, we force the empirical version of D to be nonnegative by replacing $\hat{D}$ with $\hat{D}^+$ in Algorithm 1.

4.2. *Theoretical Properties of Algorithm 1*

In this subsection, we first formally establish that Algorithm 1 gives rise to valid classifiers (Lemma 2) and then show that these classifiers have the true Type I errors controlled under α with high probability (Theorem 1).

Lemma 2. Under Assumption 2, $k^* = \min\{k \in \{1,\dots,n\} : \alpha_{k,\delta} - \hat{D}^+(t_{(k)}) \le \alpha\}$ in Algorithm 1 exists. Moreover, this label-noise-adjusted order is no larger than that chosen by the original NP umbrella algorithm; that is $k^* \le k_*$.

Lemma 2 implies that Algorithm 1 reduces the excessive conservativeness of the original NP umbrella algorithm on the Type I error by choosing a smaller order statistic as the threshold. Moreover, if there is no label noise, that is, when $m_0 = 1$ and $m_1 = 0$, we have $k^* = \min\{k \in \{1,\dots,n\} : \alpha_{k,\delta} \le \alpha\} = k_*$. That is, Algorithm 1 reduces to the original NP umbrella algorithm.

Another important question is whether Algorithm 1 can control the true Type I error with high probability. The following condition is assumed for the rest of this section.

Assumption 4. The scoring function $\hat{T}$ is trained from a class of functions $\mathcal{T}$ such that the density functions for both $\hat{T}(\tilde{X}^0)$ and $\hat{T}(\tilde{X}^1)$ exist for every $\hat{T} \in \mathcal{T}$. Then, we denote these two densities by $\tilde{f}_0^{\hat{T}}$ and $\tilde{f}_1^{\hat{T}}$, respectively. Furthermore, $\sup_{\hat{T}\in\mathcal{T}} \|\tilde{f}_0^{\hat{T}} \vee \tilde{f}_1^{\hat{T}}\|_\infty \le C$ and $\inf_{\hat{T}\in\mathcal{T}} \inf_{z\in\mathcal{D}_{\hat{T}}} \tilde{f}_0^{\hat{T}}(z) > c$ for some positive c and C with probability $1-\delta_2(n_b)$, where $\mathcal{D}_{\hat{T}}$ is the support of $\tilde{f}_0^{\hat{T}}$ and is a closed interval, and $\delta_2(n_b)$ converges to 0 as n_b goes to infinity.

Note that Assumption 4 summarizes assumptions that we make for technical convenience in establishing the next theorem. In particular, we assume the existence of densities $\tilde{f}_0^{\hat{T}}$ and $\tilde{f}_1^{\hat{T}}$, which holds if $\tilde{X}^0$ and $\tilde{X}^1$ have densities and $\hat{T}(\cdot)$ is smooth. Moreover, we assume that with high probability, both the densities are uniformly bounded from above and $\tilde{f}_0^{\hat{T}}(\cdot)$ is bounded uniformly from below.

Recall that in Algorithm 1, we set $k^* = \min\{k \in \{1,\dots,n\} : \alpha_{k,\delta} - \hat{D}^+(t_{(k^*)}) \le \alpha\}$ without an ε term. Setting $\varepsilon = 0$ intuitively seems reasonable since, when the sample size is small, the sets $\{k \in \{1,\dots,n\} : \alpha_{k,\delta} - \hat{D}^+(t_{(k^*)}) \le \alpha - \varepsilon\}$ and $\{k \in \{1,\dots,n\} : \alpha_{k,\delta} - \hat{D}^+(t_{(k^*)}) \le \alpha\}$ agree with high probability, and, when the sample size is large, concentration of random variables takes effect so there is little need for compensation for randomness. Our simulation results further reinforce this intuition. However, we include an ε term in the next theorem as this is required in our proof for the theory to hold.

Theorem 1. Under Assumptions 1–4, the classifier $\hat{\phi}_{k^*}(\cdot)$, given by Algorithm 1 with $k^* = \min\{k \in \{1,\dots,n\} : \alpha_{k,\delta} - \hat{D}^+(t_{(k)}) \le \alpha - \varepsilon\}$, satisfies

$$\mathbb{P}\left(R_0(\hat{\phi}_{k^*}) > \alpha\right) \le \delta + \delta_1(n_b) + \delta_2(n_b) + 2e^{-8^{-1}nM^{-2}C^{-2}c^2\varepsilon^2} + 2e^{-8^{-1}n_e^0M^{-2}\varepsilon^2} + 2e^{-8^{-1}n_e^1M^{-2}\varepsilon^2},$$

in which $n_b = |\tilde{\mathcal{S}}_b|$, $n = |\tilde{\mathcal{S}}_t^0|$, $n_e^0 = |\tilde{\mathcal{S}}_e^0|$, and $n_e^1 = |\tilde{\mathcal{S}}_e^1|$.

Note that the upper bound of $\mathbb{P}\left(R_0(\hat{\phi}_{k^*}) > \alpha\right)$ is δ, our violation rate control target, plus a few terms which converge to zero as the sample sizes increase. To establish this inequality, we first exclude the complement of the events described in Assumption 3 and 4. Then, we further restrict ourselves on the event constructed by a Glivenko–Cantelli-type inequality where $\hat{D}$ and D only differ by $2^{-1}\varepsilon$. There, the order selection criterion can be written as $k^* = \min\{k \in \{1,\ldots,n\} : \alpha_{k,\delta} - D(t_{(k)}) \leq \alpha - 2^{-1}\varepsilon\}$. The main difficulty of the proof is to handle the randomness of the threshold $t_{(k^*)}$. Unlike the deterministic order k_* in the original NP umbrella algorithm, the new order k^* is stochastic. As such, even when conditioning on $\hat{T}$, $t_{(k^*)}$ is sill random and cannot be handled as a normal order statistic. Our solution is to find a high probability deterministic lower bound for $t_{(k^*)}$. To do this, we introduce c_k, the k/n quantile of $\tilde{F}_0^{\hat{T}}$, which is a deterministic value if we consider $\hat{T}$ to be fixed. Then, we show that $D(t_{(k)})$ only differs from $D(c_k)$ by $4^{-1}\varepsilon$ for all k and that $\alpha_{k^*,\delta} - D(c_{k^*}) \leq \alpha - 4^{-1}\varepsilon$. Then, we define $k_0 = \min\{k \in \{1,\ldots,n\} : \alpha_{k,\delta} - D(c_k) \leq \alpha - 4^{-1}\varepsilon\}$, which is another deterministic value, given that $\hat{T}$ is considered to be fixed. Then, we find that $k_0 \leq k^*$ and $\alpha_{k_0,\delta} - D(t_{(k_0)}) \leq \alpha$ with high probability. Therefore, $t_{(k_0)}$ is a high probability lower bound for $t_{(k^*)}$. Moreover, $t_{(k_0)}$ is an order statistic with deterministic order (for fixed $\hat{T}$) and thus its distribution can be written as a binomial probability. The fact $\alpha_{k_0,\delta} - D(t_{(k_0)}) \leq \alpha$ combined with Proposition 1 yields that the violation rate of $\hat{\phi}_{k_0}(\cdot)$ is smaller than δ. The readers are referred to supplementary materials F for a complete proof.

4.3. Theoretical Properties of Algorithm 1#

In this subsection, we discuss the properties of Algorithm $1^\#$. Recall that $m_0^\# \geq m_0$ and $m_1^\# \leq m_1$ in Assumption 1 mean that the corruption levels are "underestimated." As such, Algorithm $1^\#$ produces a more conservative result than Algorithm 1. To see this, note that the only difference between two algorithms is that $(1 - m_0)(m_0 - m_1)^{-1}$ in Algorithm 1 is replaced with $(1 - m_0^\#)(m_0^\# - m_1^\#)^{-1}$ in Algorithm $1^\#$. The latter is no larger than the former, so we have a threshold in Algorithm $1^\#$ larger than or equal to that in Algorithm 1.

On the other hand, under Assumption 1, Algorithm $1^\#$ is still less conservative than the original NP umbrella algorithm. To digest this, we first consider the case where the label noise is totally "ignored," that is, $m_0^\# = 1$ and $m_1^\# = 0$. In this case, Algorithm $1^\#$ is equivalent to the original NP umbrella algorithm. Then, since usually $m_0^\# < 1$ and $m_1^\# > 0$, Algorithm $1^\#$ produces a smaller threshold than the NP original umbrella algorithm. Therefore, Algorithm $1^\#$ overcomes, at least partially, the conservativeness issue of the original NP umbrella algorithm.

These insights are formalized in the following lemma.

Lemma 3. Under Assumptions 1 - 2, $k_\#^* = \min\{k \in \{1,\ldots,n\} : \alpha_{k,\delta} - \hat{D}_\#^+(t_{(k)}) \leq \alpha\}$ in Algorithm $1^\#$ exists. Moreover, the order $k_\#^*$ is between k^* and k_*, that is, $k^* \leq k_\#^* \leq k_*$.

Next we establish a high probability control on Type I error for Algorithm $1^\#$. Recall that a high probability control on Type I error for Algorithm 1 was established in Theorem 1. In view of Lemma 3, $\hat{\phi}_{k_\#^*}(\cdot)$ produced in Algorithm $1^\#$ has a larger threshold, and thus smaller true Type I error, than that of $\hat{\phi}_{k^*}(\cdot)$ produced by Algorithm 1. Then, a high probability control on true Type I error of $\hat{\phi}_{k_\#^*}(\cdot)$ naturally follows. This result is summarized in the following corollary.

Corollary 1. Under Assumptions 1–4, the classifier $\hat{\phi}_{k_\#^*}(\cdot)$ given by Algorithm $1^\#$ with $k_\#^* = \min\{k \in \{1,\ldots,n\} : \alpha_{k,\delta} - \hat{D}_\#^+(t_{(k)}) \leq \alpha - \varepsilon\}$, satisfies

$$\mathbb{P}\left(R_0(\hat{\phi}_{k_\#^*}) > \alpha\right) \leq \delta + \delta_1(n_b) + \delta_2(n_b) + 2e^{-8^{-1}nM^{-2}C^{-2}c^2\varepsilon^2} + 2e^{-8^{-1}n_e^0M^{-2}\varepsilon^2} + 2e^{-8^{-1}n_e^1M^{-2}\varepsilon^2}.$$

in which $n_b = |\tilde{\mathcal{S}}_b|$, $n = |\tilde{\mathcal{S}}_t^0|$, $n_e^0 = |\tilde{\mathcal{S}}_e^0|$, and $n_e^1 = |\tilde{\mathcal{S}}_e^1|$.

5. Numerical Analysis

In this section, we apply Algorithms 1 (known corruption levels) and $1^\#$ (unknown corruption levels) on simulated and real datasets, and compare with other methods in the literature. We present the (approximate) Type I error violation rates[9] and the averages of (approximate) true type II errors. Besides the simulations in this section, we have additional simulations in Supplementary materials D.1. Furthermore, the violin plots associated with selected simulation are presented in Supplementary materials D.3.

As a justification of the minor discrepancy between our theory and implementation, readers can find in Supplementary materials D.5 the results for a slightly different implementation of Algorithm 1, in which $k^* = \min\{k \in \{1,\ldots,n\} : \alpha_{k,\delta} - \hat{D}^+(t_{(k)}) \leq \alpha - \varepsilon\}$ and $\varepsilon = 0.0001$. In principle, it is possible that setting $\varepsilon > 0$ will make k^* larger than when $\varepsilon = 0$ as $\{k \in \{1,2,\ldots,n\}, \alpha_{k,\delta} - \hat{D}^+(t_{(k)}) \leq \alpha_{k,\delta} - \varepsilon\}$ is a subset of $\{k \in \{1,2,\ldots,n\}, \alpha_{k,\delta} - \hat{D}^+(t_{(k)}) \leq \alpha_{k,\delta}\}$. This will make the threshold larger and the Type I error and the violation rate smaller. However, since $\varepsilon = 0.0001$ is a very small value, its effect on k^* is very minor. In numerical studies, two implementations ($\varepsilon = 0.0001$ in the supplementary materials vs. $\varepsilon = 0$ in this section) give nearly identical results for all examples. Both implementations generate the same Type I errors and Type II errors for most (at least 95%) cases. Moreover, the difference in violation rates of the two implementations is no larger than a very small number 0.1δ.

5.1. Simulation

5.1.1. Algorithm 1

We present three distributional settings for Algorithm 1 (known m_0 and m_1). In each setting, $2N$ observations are generated as a training sample, of which half are from the *corrupted* class 0 and half from the *corrupted* class 1. The number N varies from 200 to 2000. To approximate the true Type I and II errors, we generate 20,000 *true* class 0 observations and 20,000 *true* class 1 observations as the evaluation set. For each distribution and sample size combination, we repeat the procedure 1,000 times. Algorithm 1 ("adjusted") and the original NP umbrella

[9] Strictly speaking, the observed Type I error violation rate is only an approximation to the real violation rate. The approximation is two-fold: (i). in each repetition of an experiment, the population Type I error is approximated by empirical Type I error on a large test set; (ii). the violation rate should be calculated based on infinite repetitions of the experiment, but we only calculate it based on a finite number of repetitions. However, such approximation is unavoidable in numerical studies.

Table 1. (Approximate) Type I error violation rates over 1000 repetitions for Simulation 1.

N	$m_0 = 0.95, m_1 = 0.05$ $\alpha = 0.05, \delta = 0.05$		$m_0 = 0.9, m_1 = 0.1$ $\alpha = 0.05, \delta = 0.05$		$m_0 = 0.95, m_1 = 0.05$ $\alpha = 0.1, \delta = 0.1$		$m_0 = 0.9, m_1 = 0.1$ $\alpha = 0.1, \delta = 0.1$	
	Adjusted	Original	Adjusted	Original	Adjusted	Original	Adjusted	Original
200	0.026 (5.03)	0.001 (1.00)	0.033 (5.65)	0 (0)	0.078 (8.84)	0.003 (1.73)	0.073 (8.23)	0 (0)
500	0.031 (5.40)	0 (0)	0.046 (6.63)	0 (0)	0.090 (9.05)	0.001 (1.00)	0.085 (8.82)	0 (0)
1000	0.038 (5.97)	0 (0)	0.049 (6.83)	0 (0)	0.105 (9.70)	0 (0)	0.081 (8.63)	0 (0)
2000	0.053 (6.96)	0 (0)	0.046 (6.63)	0 (0)	0.087 (8.92)	0 (0)	0.099 (9.45)	0 (0)

NOTE: Standard errors ($\times 10^{-3}$) in parentheses.

Table 2. Averages of (approximate) true Type II errors over 1000 repetitions for Simulation 1.

N	$m_0 = 0.95, m_1 = 0.05$ $\alpha = 0.05, \delta = 0.05$		$m_0 = 0.9, m_1 = 0.1$ $\alpha = 0.05, \delta = 0.05$		$m_0 = 0.95, m_1 = 0.05$ $\alpha = 0.1, \delta = 0.1$		$m_0 = 0.9, m_1 = 0.1$ $\alpha = 0.1, \delta = 0.1$	
	Adjusted	Original	Adjusted	Original	Adjusted	Original	Adjusted	Original
200	0.685 (7.16)	0.706 (4.65)	0.697 (7.06)	0.826 (3.54)	0.333 (3.93)	0.403 (3.56)	0.369 (4.93)	0.537 (4.03)
500	0.481 (4.08)	0.590 (2.99)	0.512 (4.92)	0.743 (2.79)	0.249 (1.94)	0.307 (1.83)	0.257 (2.21)	0.436 (2.48)
1000	0.396 (2.53)	0.534 (2.19)	0.387 (2.37)	0.663 (1.68)	0.218 (1.18)	0.287 (1.22)	0.213 (1.01)	0.381 (1.28)
2000	0.350 (1.51)	0.491 (1.45)	0.371 (1.99)	0.651 (1.45)	0.201 (0.76)	0.268 (0.77)	0.205 (0.87)	0.375 (1.01)

NOTE: Standard errors ($\times 10^{-3}$) in parentheses.

Table 3. (Approximate) Type I error violation rates, and averages of (approximate) true Type II errors over 1000 repetitions for Simulation 2 ($m_0 = 0.95, m_1 = 0.05$, $\alpha = 0.1$ and $\delta = 0.1$).

N	(Approximate) violation rate		Averages of (approximate) true Type II errors	
	Adjusted	Original	Adjusted	Original
200	0.079 (8.53)	0.006 (2.44)	0.164 (2.77)	0.226 (3.35)
500	0.086 (8.87)	0.001 (1.00)	0.123 (0.92)	0.161 (0.80)
1000	0.085 (8.82)	0 (0)	0.109 (0.61)	0.151 (0.58)
2000	0.085 (8.82)	0 (0)	0.101 (0.44)	0.142 (0.39)

NOTE: Standard errors ($\times 10^{-3}$) in parentheses.

Table 4. (Approximate) Type I error violation rates, and averages of (approximate) true Type II errors over 1000 repetitions for Simulation 3 ($m_0 = 0.95, m_1 = 0.05$, $\alpha = 0.1$ and $\delta = 0.1$).

N	(Approximate) violation rate		Averages of (approximate) true Type II errors	
	Adjusted	Original	Adjusted	Original
200	0.068 (7.96)	0.008 (2.82)	0.526 (5.67)	0.575 (4.32)
500	0.085 (8.82)	0.002 (1.41)	0.398 (3.32)	0.472 (2.59)
1000	0.090 (9.05)	0 (0)	0.345 (2.07)	0.432 (1.78)
2000	0.093 (9.19)	0 (0)	0.314 (1.24)	0.401 (1.18)

NOTE: Standard errors ($\times 10^{-3}$) in parentheses.

algorithm (“original”) are both applied, paired with different base algorithms.

Simulation 1 (Gaussian Distribution). Let $X^0 \sim \mathcal{N}(\mu_0, \Sigma)$ and $X^1 \sim \mathcal{N}(\mu_1, \Sigma)$, where $\mu_0 = (0,0,0)^\top, \mu_1 = (1,1,1)^\top$ and

$$\Sigma = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix},$$

and the base algorithm is LDA. For different $(m_0, m_1, \alpha, \delta)$ combinations, the (approximate) Type I error violation rates and the averages of (approximate) true Type II errors generated by Algorithm 1 are reported in Tables 1 and 2, respectively.

Simulation 2 (Uniform Distribution within Circles). Let X^0 and X^1 be uniformly distributed within unit circles respectively centered at $(0,0)^\top$ and $(1,1)^\top$. The base algorithm is logistic regression. We only report (approximate) Type I error violation rates and the averages of (approximate) true Type II errors generated by Algorithm 1 for one combination ($m_0 = 0.95$, $m_1 = 0.05, \alpha = 0.1$ and $\delta = 0.1$) in Table 3.

Simulation 3 (T Distribution). Let X^0 and X^1 be t-distributed with shape matrix Σ, which was specified in Simulation 1, 4 degrees of freedom, and centered at $(0,0,0)^\top$ and $(1,1,1)^\top$ respectively. The base algorithm is LDA. Similar to the previous simulation, we only report (approximate) Type I error violation rates and the averages of (approximate) true Type II errors generated by Algorithm 1 for one combination ($m_0 = 0.95$, $m_1 = 0.05, \alpha = 0.1$, and $\delta = 0.1$) in Table 4.

The results from Simulations 1–3 confirm that the original NP umbrella algorithm is overly conservative on Type I error when there is label noise in the training data, resulting in Type I error violation rates (close to) 0 in all settings. In contrast, the label-noise-adjusted Algorithm 1 has Type I errors controlled at the specified level with high probability and achieves much better Type II errors.

5.1.2. *Algorithm 1[#].*

In this section, we show numerically that under the NP paradigm, the “under-estimates” of corruption levels serve Algorithm 1# well, while “over-estimates” do not.

Simulation 4. The distributional setting is the same as in Simulation 1. Different combinations of $m_0^\#$ and $m_1^\#$ are used. the (approximate) Type I error violation rates and the averages of (approximate) true Type II errors generated by Algorithm 1# for

Table 5. (Approximate) Type I error violation rates over 1000 repetitions for Simulation 4.

N	$m_0^\# = 0.93$, $m_1^\# = 0.07$	$m_0^\# = 0.95$, $m_1^\# = 0.05$	$m_0^\# = 0.97$, $m_1^\# = 0.03$	original
200	0.136(10.85)	0.078(8.48)	0.055(7.21)	0.003(1.73)
500	0.218(13.06)	0.090(9.05)	0.038(6.05)	0.001(1.00)
1000	0.324(14.81)	0.105(9.70)	0.012(3.44)	0(0)
2000	0.462(15.77)	0.087(8.92)	0.005(2.23)	0(0)

NOTE: Standard errors ($\times 10^{-3}$) in parentheses.

Table 6. (Approximate) Type II error violation rates over 1000 repetitions for Simulation 4.

N	$m_0^\# = 0.93$, $m_1^\# = 0.07$	$m_0^\# = 0.95$, $m_1^\# = 0.05$	$m_0^\# = 0.97$, $m_1^\# = 0.03$	original
200	0.287(3.43)	0.333(3.92)	0.373(4.62)	0.403(3.56)
500	0.215(1.61)	0.249(1.94)	0.285(2.22)	0.307(1.83)
1000	0.189(1.02)	0.218(1.18)	0.250(1.37)	0.287(1.22)
2000	0.174(0.65)	0.201(0.76)	0.230(0.86)	0.268(0.77)

NOTE: Standard errors ($\times 10^{-3}$) in parentheses.

one combination ($m_0 = 0.95, m_1 = 0.05, \alpha = 0.1$ and $\delta = 0.1$) are reported in Tables 5 and 6.

The second to the last column in Table 5 confirms that, using strict under-estimates of corruption levels (i.e., $m_0^\# > m_0$ and $m_1^\# < m_1$), the Type I error control objective is satisfied. Note that we also include the strict over-estimate scenarios in the second column (i.e., $m_0^\# < m_0$ and $m_1^\# > m_1$), where we see that the Type I violation rates exceed the target δ. Hence, the under-estimate requirement in the theory part is not merely for technical convenience. Table 6 confirms that the using strict under-estimates would lead to higher Type II errors than using the true corruption levels. This is a necessary price to pay for not knowing the exact levels, but still it is better than totally ignoring the label corruption and applying the original NP umbrella algorithm.

We state again that in this work, we rely on domain experts to supply under-estimates of corruption levels. In the literature, there are existing estimators. For example, we implement estimators proposed by Liu and Tao (2016) in Simulations 6 and 7 in Supplementary material D.1. There, we would see that those estimators do not help Algorithm $1^\#$ achieve the Type I error control objective. But this is not a problem with these estimators themselves. Even "oracle" consistent and unbiased estimators that center at m_0 and m_1 do not serve the purpose either, as revealed in Simulation 8 in supplementary material D.1. As expected, given our discussion about the need for under-estimates of the corruption levels (i.e., $m_0^\# \geq m_0$ and $m_1^\# \leq m_1$), Algorithm $1^\#$ performs poorly using these unbiased estimates. It could be an interesting topic for future research to identify an efficient method for producing biased estimates which will satisfy (with high probability) the bounds necessary to ensure correct Type I error control.

5.1.3. Benchmark Algorithms

In the next simulation, we apply existing state-of-the-art algorithms that perform classification on data with label noise. In particular, we apply the backward loss correction algorithm in Patrini et al. (2017) and the T-revision method in Xia et al. (2019). Since we focus on the NP paradigm, we will report the same (approximate) Type I error violation rates and averages of (approximate) true Type II errors as for our own methods.

Simulation 5. The distributional setting is the same as in Simulation 1. The (approximate) Type I error violation rates and averages of (approximate) true Type II errors generated by benchmark algorithms for one combination ($m_0 = 0.95, m_1 = 0.05, \alpha = 0.1$ and $\delta = 0.1$) are reported in Table 7 in the main and Table 16 in Supplementary materials D.4, respectively.

In Simulation 5, the benchmark algorithms fail to control the true Type I error with the prespecified high probability. This is understandable, as none of the benchmark algorithms have α or δ as inputs. As such, these algorithms, unlike Algorithms 1 or $1^\#$, are not designed for the NP paradigm.

5.2. Real Data Analysis

We analyze a canonical email spam dataset (Hopkins et al. 1999), which consists of 4601 observations including 57 attributes describing characteristics of emails and a $0-1$ class label. Here, 1 represents *spam* email while 0 represents *non-spam*, and the Type I/II error is defined accordingly. The labels in the dataset are all assumed to be correct.

We create corrupted labels according to the class-conditional noise model. Concretely, we flip the labels of true class 0 observations with probability r_0 and flip the labels of true class

Table 8. (Approximate) Type I error violation rates, and averages of (approximate) true Type II errors by Algorithm 1 and original NP umbrella algorithm over 1000 repetitions for the email spam data.

	(approximate) violation rate		average of (approximate) true type II errors	
	adjusted	original	adjusted	original
Penalized logistic regression	0.082(8.68)	0(0)	0.205(2.65)	0.272(2.71)
LDA	0.096(9.32)	0(0)	0.226(3.05)	0.314(2.77)
Support vector machine	0.093(9.19)	0.004(2.00)	0.183(3.15)	0.218(1.93)
Random forests	0.080(8.58)	0(0)	0.120(1.13)	0.152(1.54)

NOTE: Standard errors ($\times 10^{-3}$) in parentheses.

Table 7. (Approximate) Type I error violation rates over 1000 repetitions for Simulation 5 ($m_0 = 0.95, m_1 = 0.05, \alpha = 0.1$ and $\delta = 0.1$).

algorithms	N			
	200	500	1000	2000
T-revision	0.713(14.31)	0.675(14.82)	0.651(15.08)	0.621(15.35)
Backward loss correction (known corruption levels)	0.994(2.44)	0.977(4.74)	0.770(13.31)	0.127(10.53)
Backward loss correction (unknown corruption levels)	0.984(3.97)	0.793(5.20)	0.320(6.89)	0.131(3.60)

NOTE: Standard errors ($\times 10^{-3}$) in parentheses

Table 9. (Approximate) Type I error violation rates by Algorithm 1$^\#$ over 1000 repetitions for the email spam data.

	$m_0^\# = 0.93$,	$m_0^\# = 0.95$,	$m_0^\# = 0.97$,	original
	$m_1^\# = 0.07$	$m_1^\# = 0.05$	$m_1^\# = 0.03$	
Penalized logistic regression	0.231(13.33)	0.082(8.68)	0.028(5.22)	0(0)
LDA	0.223(13.17)	0.096(9.32)	0.023(4.74)	0(0)
Support vector machine	0.220(13.11)	0.093(9.19)	0.026(5.03)	0.004(2.00)
Random forest	0.238(13.47)	0.080(8.58)	0.019(4.32)	0(0)

NOTE: Standard errors ($\times 10^{-3}$) in parentheses.

Table 10. Averages of (approximate) true Type II errors by Algorithm 1$^\#$ over 1000 repetitions for the email spam data.

	$m_0^\# = 0.93$,	$m_0^\# = 0.95$,	$m_0^\# = 0.97$,	original
	$m_1^\# = 0.07$	$m_1^\# = 0.05$	$m_1^\# = 0.03$	
Penalized logistic regression	0.165(2.04)	0.205(2.65)	0.254(3.10)	0.272(2.71)
LDA	0.213(2.54)	0.226(3.05)	0.314(3.37)	0.314(2.77)
Support vector machine	0.138(1.20)	0.183(3.15)	0.199(2.11)	0.218(1.93)
Random forest	0.102(0.78)	0.120(1.13)	0.143(1.41)	0.152(1.54)

NOTE: Standard errors ($\times 10^{-3}$) in parentheses.

1 observations with probability r_1. Note that m_0 and m_1 are $\mathbb{P}(Y = 0|\tilde{Y} = 0)$ and $\mathbb{P}(Y = 0|\tilde{Y} = 1)$, respectively, while $r_0 = \mathbb{P}(\tilde{Y} = 1|Y = 0)$ and $r_1 = \mathbb{P}(\tilde{Y} = 0|Y = 1)$.

In our analysis, we choose $m_0 = 0.95$ and $m_1 = 0.05$, which implies setting $r_0 = 0.032$ and $r_1 = 0.078$[10]. For each training and evaluation procedure, we split the data by stratified sampling into training and evaluation sets. Specifically, 20% of the true class 0 observations and 20% of the true class 1 observations are randomly selected to form the training dataset, and the rest of the observations form the evaluation dataset. In total, the training set contains 921 observations and the evaluation set contains 3680 observations. The larger evaluation set is reserved to better approximate (population-level) true Type I/II error. We leave the evaluation data untouched, but randomly flip the training data label according to the calculated r_0 and r_1. Four base algorithms are coupled with the original and new NP umbrella algorithms, with $\alpha = \delta = 0.1$. We repeat the procedure 1000 times.

The (approximate) Type I error violation rates and averages of (approximate) true Type II errors generated by Algorithm 1 and the original NP umbrella algorithm are summarized in Table 8. Similar to the simulation studies, we observe that Algorithm 1 correctly controls Type I error at the right level, while the original NP umbrella algorithm is significantly overly conservative on Type I error, and consequently has much higher Type II error. We also summarize the results generated by Algorithm 1$^\#$ in Tables 9 and 10. Clearly, while strict under-estimates lead to higher Type II errors than using exact corruption levels, the Type I error control objective is achieved, and the Type II error is better than just ignoring label corruption and applying the original NP umbrella algorithm.

To make a comparison, we also apply the loss correction algorithm in Patrini et al. (2017) and the T-revision method in Xia et al. (2019) to the email spam data, with results summarized in Table 17 in supplementary material D.4. Since these benchmark algorithms are not designed for the NP paradigm, as discussed in Section 5.1, none of the (approximate) true Type I error violation rates are controlled as we desire. In addition to the email spam data, we also apply Algorithm 1 to the CIFAR10 dataset (Krizhevsky et al. 2009) and successfully have the Type I error controlled (Supplementary material D.2).

6. Discussion

Under the NP paradigm, we developed the first label-noise-adjusted umbrella algorithms. There are several interesting directions for future research. First, we can consider a more complex noise model in which the corruption levels depend on both the class and features. Another direction is to consider data-driven "under-estimates" of the corruption levels in the class-conditional noise model and develop (distributional) model-specific adjustment algorithms. For instance, we can adopt the LDA model, that is, $X^0 \sim \mathcal{N}(\mu_0, \Sigma)$ and $X^1 \sim \mathcal{N}(\mu_1, \Sigma)$.

Supplementary Material

The supplementary material contains technical and additional numerical results.

Acknowledgments

We thank to Dr. Xiao Han from the University of Science and Technology of China and Dr. Min Zhou from United International College for inspirational discussions.

Funding

This work was partially supported by U.S. NSF grant DMS 2113500.

References

Blanchard, G., Flaska, M., Handy, G., Pozzi, S., and Scott, C. (2016), "Classification With Asymmetric Label Noise: Consistency and Maximal Denoising," *Electronic Journal of Statistics*, 10, 2780–2824. [1,3]

Brazdil, P., and Konolige, K. (1990), *Machine Learning, Meta-Reasoning and Logics*, Boston, MA: Springer. [1]

[10] This is an application of the Bayes theorem with $\mathbb{P}(Y = 0)$ estimated to be 0.610, which is the proportion of class 0 observations in the whole dataset.

Brodley, C. E., and Friedl, M. A. (1999a), "Identifying Mislabeled Training Data," *Journal of Artificial Intelligence Research*, 11, 131–167. [1]

Brodley, C., and Friedl, M. (1999b), "Identifying Mislabeled Training Data," *Journal of Artificial Intelligence Research*, 11, 131–167. [1]

Cannings, T. I., Fan, Y., and Samworth, R. J. (2020), "Classification With Imperfect Training Labels," *Biometrika*, 107, 311–330. [1]

Cannon, A., Howse, J., Hush, D., and Scovel, C. (2002), "Learning With the Neyman–Pearson and Min–Max Criteria," Los Alamos National Laboratory, Tech. Rep. LA-UR, pp. 02-2951. [2,3]

Cao, J., Kwong, S., and Wang, R. (2012), "A Noise-Detection Based AdaBoost Algorithm for Mislabeled Data," *Pattern Recognition*, 45, 4451–4465. [1]

Ghosh, A., Manwani, N., and Sastry, P. (2015), "Making Risk Minimization Tolerant to Label Noise," *Neurocomputing*, 160, 93–107. [1]

Guyon, I., Matic, N., and Vapnik, V. (1996), "Discovering Informative Patterns and Data Cleaning," in *Advances in Knowledge Discovery and Data Mining*, edited by U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Menlo Park, CA: American Association for Artificial Intelligence, pp. 181–203. [1]

Hickey, R. J. (1996), "Noise Modelling and Evaluating Learning From Examples," *Artificial Intelligence*, 82, 157–179. [1]

Hopkins, M., Reeber, E., Forman, G., and Suermondt, J. (1999), "Spambase Data Set," *Hewlett-Packard Labs*, 1. Available at: *https://archive.ics.uci.edu/ml/datasets/spambase* [11]

Khardon, R., and Wachman, G. (2007), "Noise Tolerant Variants of the Perceptron Algorithm," *Journal of Machine Learning Research*, 8, 227–248. [1]

Krizhevsky, A. (2009), "Learning Multiple Layers of Features From Tiny Images," available at http://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf. [12]

Lachenbruch, P. A. (1966), "Discriminant Analysis When the Initial Samples Are Misclassified," *Technometrics*, 8, 657–662. [1]

——— (1979), "Note on Initial Misclassification Effects on the Quadratic Discriminant Function," *Technometrics*, 21, 129–132. [1]

Liu, T., and Tao, D. (2016), "Classification With Noisy Labels by Importance Reweighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38, 447–461. [1,3,11]

MacDonald, O. (2011), "Physician Perspectives on Preventing Diagnostic Errors." Available at: *https://www.kff.org/wp-content/uploads/sites/2/2013/05/quantiamd_preventingdiagnosticerrors_whitepaper_1.pdf*. [1]

Manwani, N., and Sastry, P. (2013), "Noise Tolerance Under Risk Minimization," *IEEE Transactions on Cybernetics*, 43, 1146–1151. [1]

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. (2013), "Learning With Noisy Labels," in *Advances in Neural Information Processing Systems*, *NIPS'13: Proceedings of the 26th International Conference on Neural Information Processing Systems* (Vol. 1), pp. 1196–1204. [1,3]

Okamoto, S., and Yugami, N. (1997), "An Average-Case Analysis of the k-Nearest Neighbor Classifier for Noisy Domains," in *IJCAI (1)*, pp. 238–245. [1]

Orr, K. (1998), "Data Quality and Systems Theory," *Communications of the ACM*, 41, 66–71. [1]

Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., and Qu, L. (2017), "Making Deep Neural Networks Robust to Label Noise: A Loss Correction Approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1944–1952. [11,12]

Redman, T. (1998), "The Impact of Poor Data Quality on the Typical Enterprise," *Communications of the ACM*, 2, 79–82. [1]

Rigollet, P., and Tong, X. (2011), "Neyman–Pearson Classification, Convexity and Stochastic Constraints," *Journal of Machine Learning Research*, 12, 2831–2855. [4]

Scott, C., and Nowak, R. (2005), "A Neyman–Pearson Approach to Statistical Learning," *IEEE Transactions on Information Theory*, 51, 3806–3819. [2,3]

Sukhbaatar, S., and Fergus, R. (2014), "Learning From Noisy Labels With Deep Neural Networks," arXiv:1406.2080, 2, 4. [1]

Tong, X. (2013), "A Plug-in Approach to Neyman-Pearson Classification," *Journal of Machine Learning Research*, 14, 3011–3040. [4]

Tong, X., Feng, Y., and Li, J. J. (2018), "Neyman–Pearson Classification Algorithms and NP Receiver Operating Characteristics," *Science Advances*, 4, eaao1659. [2,3,4,6]

Tong, X., Xia, L., Wang, J., and Feng, Y. (2020), "Neyman–Pearson Classification: Parametrics and Sample Size Requirement," *Journal of Machine Learning Research*, 21, 1–18. [4]

Xia, X., Liu, T., Wang, N., Han, B., Gong, C., Niu, G., and Sugiyama, M. (2019), "Are Anchor Points Really Indispensable in Label-Noise Learning?" *Advances in Neural Information Processing Systems*, 32, 6838–6849. [11,12]

Zhao, A., Feng, Y., Wang, L., and Tong, X. (2016), "Neyman–Pearson Classification Under High-Dimensional Settings," *Journal of Machine Learning Research*, 17, 1–39. [4]