Asymmetric error control under imperfect supervision: a label-noise-adjusted Neyman-Pearson umbrella algorithm

Shunan Yao[1], Bradley Rava[2], Xin Tong[2,3], and Gareth James[2,3]

[1]Department of Mathematics, Dana and David Dornsife College of Letters, Arts and Sciences, University of Southern California.

[2]Department of Data Sciences and Operations, Marshall School of Business, University of Southern California.

[3]To whom correspondence should be addressed. xint@marshall.usc.edu, gareth.james@marshall.usc.edu

October 10, 2021

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.

KEY WORDS: label noise, classification, 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, 1999*b*). Specifically, in the medical field, a 2011 survey of more than 6,000 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 et al., 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 et al. (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 et al. (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 malig-

nant 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.

In this paper, 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 (Tong et al., 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

[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.

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 paper 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 Appendix.

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 $X^0 \triangleq X \mid (Y = 0)$ and $X^1 \triangleq X \mid (Y = 1)$. Similarly, denote $\tilde{X}^0 \triangleq X \mid (\tilde{Y} = 0)$ and $\tilde{X}^1 \triangleq X \mid (\tilde{Y} = 1)$. 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.

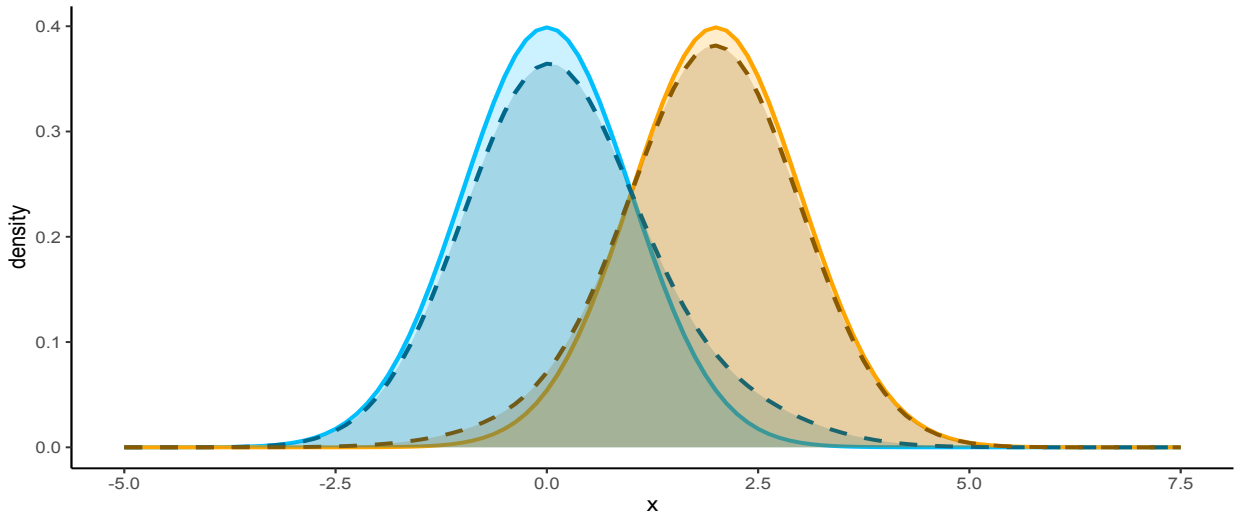

Figure 1: Density plots in Example 1. True (lighter and solid) and corrupted (darker and dashed).

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

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

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* $m_0^{\#} \geq m_0$ *and* $m_1^{\#} \leq m_1$.

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 $m_0^{\#} \geq m_0$ and $m_1^{\#} \leq m_1$ mean that we know some *lower bounds* of the corruption levels.

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 papers, 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

[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.

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 Appendix, 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 et al., 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. i.e., 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 $\phi : \mathcal{X} \to \{0, 1\}$ 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 paper, *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 towards 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 et al., 2018) has attracted 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 1-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, i.e., $R_1(\hat{\phi}_\alpha) - R_1(\phi^*_\alpha) \to 0$. 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 $\mathcal{S}^0 = \{X^0_j\}_{j=1}^{M_0}$ and $\mathcal{S}^1 = \{X^1_j\}_{j=1}^{M_1}$, 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: $\mathcal{S}^0_{\text{b}}$ and $\mathcal{S}^0_{\text{t}}$. The first part, $\mathcal{S}^0_{\text{b}}$, together with $\mathcal{S}^1$, is used to train the *base* algorithm, while the second part $\mathcal{S}^0_{\text{t}}$ 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 $\mathcal{S}^0_{\text{b}} \cup \mathcal{S}^1$, apply $\hat{T}(\cdot)$ on $\mathcal{S}^0_{\text{t}}$ ($|\mathcal{S}^0_{\text{t}}| = n$) 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 $\hat{\phi}_{k_*}(\cdot) = \mathbb{1}\{\hat{T}(\cdot) > t_{(k_*)}\}$, where

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

in which δ is a user-specified tolerance probability of the type I error exceeding α. The key to this approach is that Tong et al. (2018) established, for all $\hat{\phi}_k(\cdot) = \mathbb{1}\{\hat{T}(\cdot) > t_{(k)}\}$ where $k \in \{1, \ldots, n\}$,

[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 non-parametric 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.

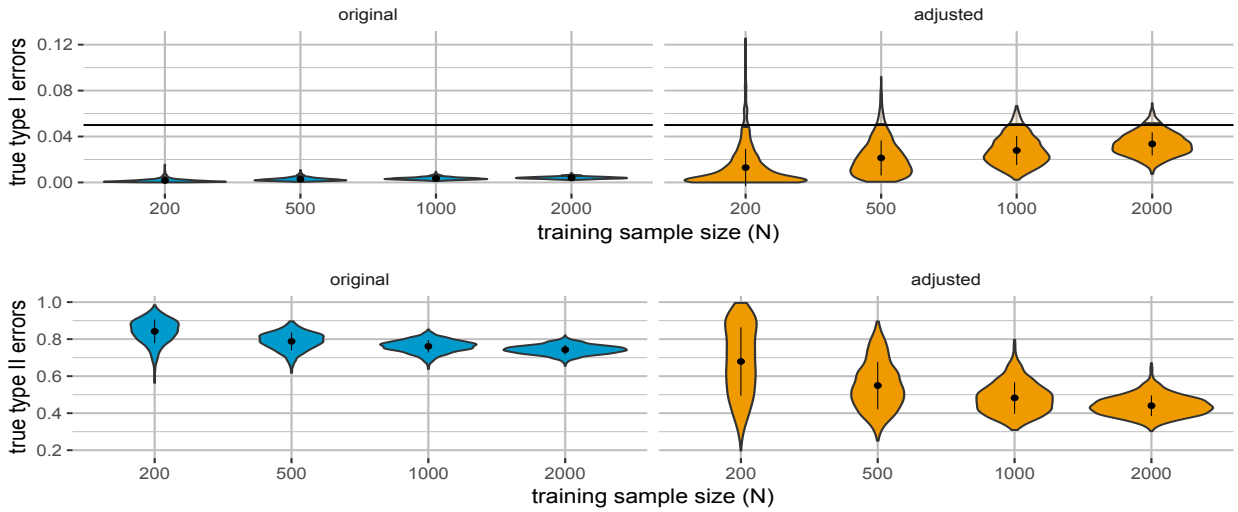

Figure 2: The original NP umbrella algorithm vs. a label-noise-adjusted version for Example 2. The plots in the left panel (blue) are the true type I and II errors for the original NP umbrella algorithm. The plots in the right panel (orange) are the true type I and II errors for the label-noise-adjusted 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.

it holds $\mathbb{P}(R_0(\hat{\phi}_k) > \alpha) \leq \sum_{j=k}^{n} \binom{n}{j}(1-\alpha)^j \alpha^{(n-j)}$, 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 $\mathbb{P}(R_0(\hat{\phi}_{k_*}) > \alpha) \leq \delta$, 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* $N \in$

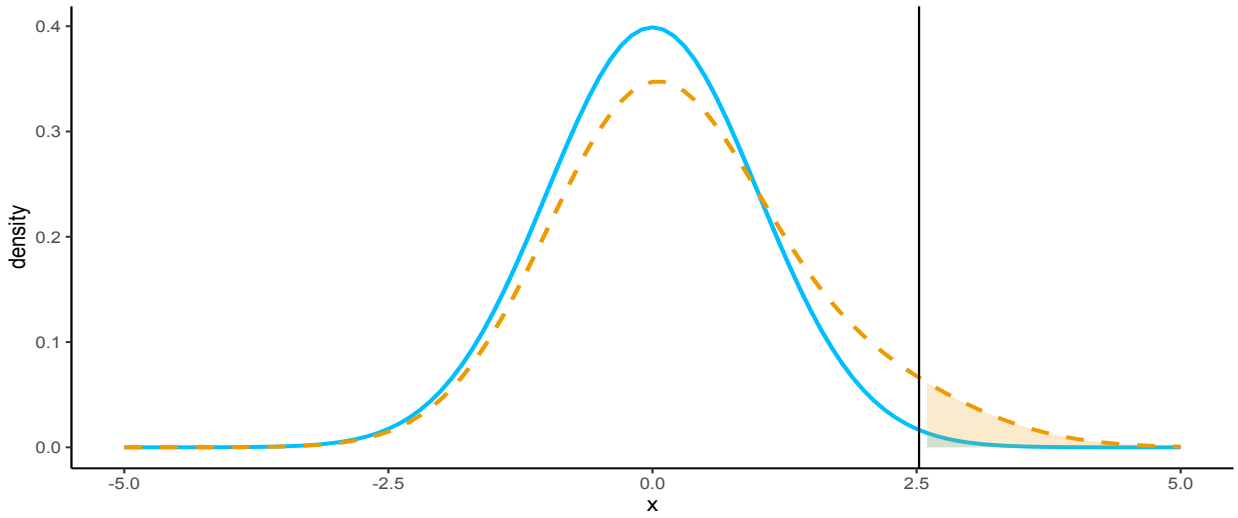

Figure 3: The blue solid curve is the density of true class 0 (i.e., $\mathcal{N}(0,1)$) and the orange dashed curve is the density of corrupted class 0 (i.e., a mixture of $\mathcal{N}(0,1)$ and $\mathcal{N}(2,1)$ with $m_0 = 0.85$). The black vertical line marks the threshold of the classifier $\mathbb{1}\{X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

$\{200, 500, 1000, 2000\}$ *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 $\mathbb{1}\{X > 2.52\}$ 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

[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.

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 non-random 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 $\tilde{\mathcal{S}}_{\mathrm{b}}^0$ and $\tilde{\mathcal{S}}_{\mathrm{t}}^0$ are used to train the *base* algorithm and determine the *threshold* candidates, respectively. The third part $\tilde{\mathcal{S}}_{\mathrm{e}}^0$ is used to *estimate* a correction term to account for the label noise. Similarly, we randomly divide $\tilde{\mathcal{S}}^1$ into *two* disjoint non-empty subsets: $\tilde{\mathcal{S}}_{\mathrm{b}}^1$ and $\tilde{\mathcal{S}}_{\mathrm{e}}^1$.

Let $\hat{T}(\cdot)$ be a scoring function trained on $\tilde{\mathcal{S}}_{\mathrm{b}} = \tilde{\mathcal{S}}_{\mathrm{b}}^0 \cup \tilde{\mathcal{S}}_{\mathrm{b}}^1$. We apply $\hat{T}(\cdot)$ to elements in $\tilde{\mathcal{S}}_{\mathrm{t}}^0$ and sort them in an increasing order: $\{t_{(1)}, \ldots, t_{(n)}\}$, where $n = |\tilde{\mathcal{S}}_{\mathrm{t}}^0|$[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

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

$\hat{D}^+(\cdot) = \hat{D}(\cdot) \vee 0 := \max(\hat{D}(\cdot), 0)$ 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 $\tilde{\mathcal{S}}_{\mathrm{e}}^0$ and $\tilde{\mathcal{S}}_{\mathrm{e}}^1$, 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 (Algorithm 2 in

[7] In Appendix 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 Appendix.

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
α: type I error upper bound, $0 < \alpha < 1$
δ: 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(δ, k, n) ; // compute $\alpha_{k,\delta}$ through binary search
  $\hat{\tilde{F}}^{\hat{T}}_0(t_{(k)}) \leftarrow \left|\mathcal{T}^0_{\mathrm{e}}\right|^{-1} \cdot \sum_{t \in \mathcal{T}^0_{\mathrm{e}}} \mathbb{1}\{t \le t_{(k)}\}$
  $\hat{\tilde{F}}^{\hat{T}}_1(t_{(k)}) \leftarrow \left|\mathcal{T}^1_{\mathrm{e}}\right|^{-1} \cdot \sum_{t \in \mathcal{T}^1_{\mathrm{e}}} \mathbb{1}\{t \le t_{(k)}\}$; // 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
$\hat{\phi}_{k^*}(\cdot) \leftarrow \mathbb{1}\{\hat{T}(\cdot) > t_{(k^*)}\}$; // construct an NP classifier
Output: $\hat{\phi}_{k^*}(\cdot)$

Appendix 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 Appendix 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 $\hat{\phi}_{k^*}(\cdot)$.

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 $\hat{D}_{\#}(t_{(k)}) = \frac{1-m_0^{\#}}{m_0^{\#}-m_1^{\#}} \left(\hat{\tilde{F}}_0^{\hat{T}}(t_{(k)}) - \hat{\tilde{F}}_1^{\hat{T}}(t_{(k)}) \right)$ and $\hat{D}_{\#}^{+}(t_{(k)}) = \hat{D}_{\#}(t_{(k)}) \vee 0$. Then, Algorithm $1^{\#}$ delivers the NP classifier $\hat{\phi}_{k_{\#}^*}(\cdot) = \mathbb{1}\{\hat{T}(\cdot) > t_{(k_{\#}^*)}\}$, where $k_{\#}^* = \min\{k \in \{1, \ldots, n\} : \alpha_{k,\delta} - \hat{D}_{\#}^{+}(t_{(k)}) \leq \alpha\}$. 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., $m_0^{\#} \geq m_0$ and $m_1^{\#} \leq m_1$ 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 $\tilde{\mathcal{S}}_{\mathrm{b}}$. Applying $\hat{T}(\cdot)$ to every element in $\tilde{\mathcal{S}}_{\mathrm{t}}^0$, we get a set of scores. Order these scores and denote them by $\{t_{(1)}, t_{(2)}, \ldots, t_{(n)}\}$, in which $t_{(1)} \leq t_{(2)} \leq \ldots \leq t_{(n)}$. Then, for any $\alpha \in (0, 1)$ and*

$k \in \{1, 2, \ldots, n\}$, the classifier $\hat{\phi}_k(\cdot) = \mathbb{1}\{\hat{T}(\cdot) > t_{(k)}\}$ satisfies

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

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 et al. (2018). For $\alpha, \delta \in (0, 1)$, recall that the original NP umbrella algorithm selects $k_* = \min\{k \in \{1, \ldots, n\} : \sum_{j=k}^{n} \binom{n}{j}(1-\alpha)^j \alpha^{(n-j)} \leq \delta\}$. The smallest k among all that satisfy $\sum_{j=k}^{n} \binom{n}{j}(1-\alpha)^j \alpha^{(n-j)} \leq \delta$ 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 $(1-\alpha)^n \leq \delta$. This translates to Assumption 2 on n, the sample size of $\tilde{\mathcal{S}}_{\mathrm{t}}^0$. This is a mild requirement. For instance, when $\alpha = \delta = 0.05$, n should be at least 59.

Assumption 2. *$n \geq \lceil \log \delta / \log(1-\alpha) \rceil$, 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 $1 - \delta_1(n_{\mathrm{b}})$, where $n_{\mathrm{b}} = |\tilde{\mathcal{S}}_{\mathrm{b}}|$ and $\delta_1(n_{\mathrm{b}})$ 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 Appendix C (Example 3). 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 $\tilde{\mathcal{S}}_{\mathrm{b}}$ 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, 2 and 3, for given α and δ, it holds that*

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

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

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

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, 2 and 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 $\hat{\phi}_{\tilde{k}^*}(\cdot) = \mathbb{I}\{\hat{T}(\cdot) > t_{(\tilde{k}^*)}\}$. 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 Appendix). 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 $\tilde{\mathcal{S}}_{\mathrm{e}}^1$, i.i.d. 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 $c \in \mathbb{R}$,

$$\hat{D}(c) = \frac{1-m_0}{m_0-m_1}\left(\hat{\tilde{F}}_0^{\hat{T}}(c) - \hat{\tilde{F}}_1^{\hat{T}}(c)\right) \text{ and } k^{**} = \min\{k \in \{1,\ldots,n\} : \alpha_{k,\delta} - \hat{D}(t_{(k)}) \leq \alpha - \varepsilon\},$$

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* $\hat{\phi}_{k^*}(\cdot) = \mathbb{I}\{\hat{T}(\cdot) > t_{(k^*)}\}$, 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.

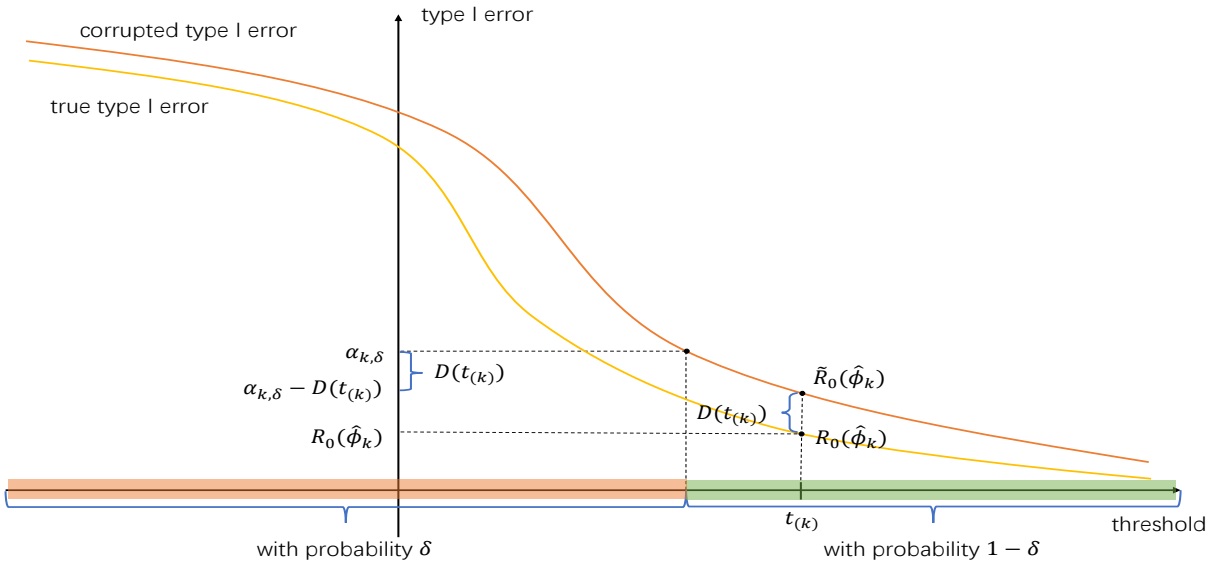

Figure 4: A cartoon illustration of $1-\delta$ probability upper bound of type I error.

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, 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 $\min\{k \in \{1, \ldots, n\} : \alpha_{k,\delta} - \hat{D}(t_{(k)}) \leq \alpha\} \geq \min\{k \in \{1, \ldots, n\} : \alpha_{k,\delta} \leq \alpha\}$. 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 $k^* \geq k_*$ will reverse this effort and worsen the conservativeness. To solve this issue, we force the empirical version of D to be non-negative 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,* $k^* = \min\{k \in \{1, \ldots, n\} : \alpha_{k,\delta} - \hat{D}^+(t_{(k)}) \leq \alpha\}$ *in Algorithm 1*

exists. Moreover, this label-noise-adjusted order is no larger than that chosen by the original NP umbrella algorithm; that is $k^* \leq k_*$.

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, i.e., when $m_0 = 1$ and $m_1 = 0$, we have $k^* = \min\{k \in \{1, \ldots, n\} : \alpha_{k,\delta} \leq \alpha\} = k_*$. 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,* $\sup_{\hat{T}\in\mathcal{T}} \|\tilde{f}_0^{\hat{T}} \vee \tilde{f}_1^{\hat{T}}\|_\infty \leq C$ *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* $1 - \delta_2(n_{\mathrm{b}})$*, where* $\mathcal{D}_{\hat{T}}$ *is the support of* $\tilde{f}_0^{\hat{T}}$ *and is a closed interval, and* $\delta_2(n_{\mathrm{b}})$ *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 $k^* = \min\{k \in \{1, \ldots, n\} : \alpha_{k,\delta} - \hat{D}^+(t_{(k^*)}) \leq \alpha\}$ without an ε term. Setting $\varepsilon = 0$ intuitively seems reasonable since, when the sample size is small, the sets $\{k \in \{1, \ldots, n\} : \alpha_{k,\delta} - \hat{D}^+(t_{(k^*)}) \leq \alpha - \varepsilon\}$ and $\{k \in \{1, \ldots, n\} : \alpha_{k,\delta} - \hat{D}^+(t_{(k^*)}) \leq \alpha\}$ 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, 2, 3 and 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*

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

in which $n_{\mathrm{b}} = |\tilde{\mathcal{S}}_{\mathrm{b}}|$, $n = |\tilde{\mathcal{S}}_{\mathrm{t}}^0|$, $n_{\mathrm{e}}^0 = |\tilde{\mathcal{S}}_{\mathrm{e}}^0|$, *and* $n_{\mathrm{e}}^1 = |\tilde{\mathcal{S}}_{\mathrm{e}}^1|$.

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 Appendix 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", i.e., $m_0^{\#} = 1$ 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_**, i.e.,* $k^* \leq k_{\#}^* \leq k_*$.

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*

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

in which $n_{\mathrm{b}} = |\tilde{\mathcal{S}}_{\mathrm{b}}|$, $n = |\tilde{\mathcal{S}}_{\mathrm{t}}^0|$, $n_{\mathrm{e}}^0 = |\tilde{\mathcal{S}}_{\mathrm{e}}^0|$, *and* $n_{\mathrm{e}}^1 = |\tilde{\mathcal{S}}_{\mathrm{e}}^1|$.

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

[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.

II errors. Besides the simulations in this section, we have additional simulations in Appendix D.1. Furthermore, the violin plots associated with selected simulation are presented in Appendix D.3.

As a justification of the minor discrepancy between our theory and implementation, readers can find in Appendix 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 Appendix 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 2,000. 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 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 linear discriminant analysis (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.

Table 1: (Approximate) type I error violation rates over 1,000 repetitions for Simulation 1. Standard errors ($\times 10^{-3}$) in parentheses.

N	$m_0 = .95, m_1 = .05$ $\alpha = .05, \delta = .05$		$m_0 = .9, m_1 = .1$ $\alpha = .05, \delta = .05$		$m_0 = .95, m_1 = .05$ $\alpha = .1, \delta = .1$		$m_0 = .9, m_1 = .1$ $\alpha = .1, \delta = .1$	
	adjusted	original	adjusted	original	adjusted	original	adjusted	original
200	.026 (5.03)	.001 (1.00)	.033 (5.65)	0 (0)	.078 (8.84)	.003 (1.73)	.073 (8.23)	0 (0)
500	.031 (5.40)	0 (0)	.046 (6.63)	0 (0)	.090 (9.05)	.001 (1.00)	.085 (8.82)	0 (0)
1,000	.038 (5.97)	0 (0)	.049 (6.83)	0 (0)	.105 (9.70)	0 (0)	.081 (8.63)	0 (0)
2,000	.053 (6.96)	0 (0)	.046 (6.63)	0 (0)	.087 (8.92)	0 (0)	.099 (9.45)	0 (0)

Table 2: Averages of (approximate) true type II errors over 1,000 repetitions for Simulation 1. Standard errors ($\times 10^{-3}$) in parentheses.

N	$m_0 = .95, m_1 = .05$ $\alpha = .05, \delta = .05$		$m_0 = .9, m_1 = .1$ $\alpha = .05, \delta = .05$		$m_0 = .95, m_1 = .05$ $\alpha = .1, \delta = .1$		$m_0 = .9, m_1 = .1$ $\alpha = .1, \delta = .1$	
	adjusted	original	adjusted	original	adjusted	original	adjusted	original
200	.685 (7.16)	.706 (4.65)	.697 (7.06)	.826 (3.54)	.333 (3.93)	.403 (3.56)	.369 (4.93)	.537 (4.03)
500	.481 (4.08)	.590 (2.99)	.512 (4.92)	.743 (2.79)	.249 (1.94)	.307 (1.83)	.257 (2.21)	.436 (2.48)
1,000	.396 (2.53)	.534 (2.19)	.387 (2.37)	.663 (1.68)	.218 (1.18)	.287 (1.22)	.213 (1.01)	.381 (1.28)
2,000	.350 (1.51)	.491 (1.45)	.371 (1.99)	.651 (1.45)	.201 (.76)	.268 (.77)	.205 (.87)	.375 (1.01)

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 ($m_0 = .95$, $m_1 = .05$, $\alpha = .1$ and $\delta = .1$) in Table 3.*

Table 3: (Approximate) type I error violation rates, and averages of (approximate) true type II errors over 1,000 repetitions for Simulation 2 ($m_0 = .95$, $m_1 = .05$, $\alpha = .1$ and $\delta = .1$). Standard errors ($\times 10^{-3}$) in parentheses.

N	(approximate) violation rate		averages of (approximate) true type II errors	
	adjusted	original	adjusted	original
200	.079 (8.53)	.006 (2.44)	.164 (2.77)	.226 (3.35)
500	.086 (8.87)	.001 (1.00)	.123 (.92)	.161 (.80)
1,000	.085 (8.82)	0 (0)	.109 (.61)	.151 (.58)
2,000	.085 (8.82)	0 (0)	.101 (.44)	.142 (.39)

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 ($m_0 = .95$, $m_1 = .05$, $\alpha = .1$ and $\delta = .1$) in Table 4.

Table 4: (Approximate) type I error violation rates, and averages of (approximate) true type II errors over 1,000 repetitions for Simulation 3 ($m_0 = .95$, $m_1 = .05$, $\alpha = .1$ and $\delta = .1$). Standard errors ($\times 10^{-3}$) in parentheses.

N	(approximate) violation rate		average of (approximate) true type II errors	
	adjusted	original	adjusted	original
200	.068 (7.96)	.008 (2.82)	.526 (5.67)	.575 (4.32)
500	.085 (8.82)	.002 (1.41)	.398 (3.32)	.472 (2.59)
1,000	.090 (9.05)	0 (0)	.345 (2.07)	.432 (1.78)
2,000	.093 (9.19)	0 (0)	.314 (1.24)	.401 (1.18)

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 one combination ($m_0 = .95$, $m_1 = .05$, $\alpha = .1$ and $\delta = .1$) 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

Table 5: (Approximate) type I error violation rates over 1,000 repetitions for Simulation 4. Standard errors ($\times 10^{-3}$) in parentheses.

N	$m_0^\# = .93$, $m_1^\# = .07$	$m_0^\# = .95$, $m_1^\# = .05$	$m_0^\# = .97$, $m_1^\# = .03$	original
200	.136(10.85)	.078(8.48)	.055(7.21)	.003(1.73)
500	.218(13.06)	.090(9.05)	.038(6.05)	.001(1.00)
1,000	.324(14.81)	.105(9.70)	.012(3.44)	0(0)
2,000	.462(15.77)	.087(8.92)	.005(2.23)	0(0)

Table 6: (Approximate) type II error violation rates over 1,000 repetitions for Simulation 4. Standard errors ($\times 10^{-3}$) in parentheses.

N	$m_0^\# = .93$, $m_1^\# = .07$	$m_0^\# = .95$, $m_1^\# = .05$	$m_0^\# = .97$, $m_1^\# = .03$	original
200	.287(3.43)	.333(3.92)	.373(4.62)	.403(3.56)
500	.215(1.61)	.249(1.94)	.285(2.22)	.307(1.83)
1,000	.189(1.02)	.218(1.18)	.250(1.37)	.287(1.22)
2,000	.174(.65)	.201(.76)	.230(.86)	.268(.77)

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 Appendix 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 Appendix 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 1 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 ($m_0 = .95$, $m_1 = .05$, $\alpha = .1$ and $\delta = .1$) are reported in Table 7 in the main and Table 16 in Appendix D.4, respectively.*

Table 7: (Approximate) type I error violation rates over 1,000 repetitions for Simulation 5 ($m_0 = .95$, $m_1 = .05$, $\alpha = .1$ and $\delta = .1$). Standard errors ($\times 10^{-3}$) in parentheses.

algorithms	N			
	200	500	1,000	2,000
T-revision	.713(14.31)	.675(14.82)	.651(15.08)	.621(15.35)
backward loss correction (known corruption levels)	.994(2.44)	.977(4.74)	.770(13.31)	.127(10.53)
backward loss correction (unknown corruption levels)	.984(3.97)	.793(5.20)	.320(6.89)	.131(3.60)

In Simulation 5, the benchmark algorithms fail to control the true type I error with the pre-specified 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 4,601 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 1 observations with probability r_1. Note that m_0 and m_1 are $\mathbb{P}(Y = 0 \mid \tilde{Y} = 0)$ and $\mathbb{P}(Y = 0 \mid \tilde{Y} = 1)$, respectively, while $r_0 = \mathbb{P}(\tilde{Y} = 1 \mid Y = 0)$ and $r_1 = \mathbb{P}(\tilde{Y} = 0 \mid Y = 1)$. 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 3,680 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 1,000 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.

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 1,000 repetitions for the email spam data. Standard errors ($\times 10^{-3}$) in parentheses.

	(approximate) violation rate		average of (approximate) true type II errors	
	adjusted	original	adjusted	original
penalized logistic regression	.082(8.68)	0(0)	.205(2.65)	.272(2.71)
linear discriminant analysis	.096(9.32)	0(0)	.226(3.05)	.314(2.77)
support vector machine	.093(9.19)	.004(2.00)	.183(3.15)	.218(1.93)
random forests	.080(8.58)	0(0)	.120(1.13)	.152(1.54)

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

[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.

Table 9: (Approximate) type I error violation rates by Algorithm $1^{\#}$ over 1,000 repetitions for the email spam data. Standard errors ($\times 10^{-3}$) in parentheses.

	$m_0^{\#} = 0.93$, $m_1^{\#} = 0.07$	$m_0^{\#} = 0.95$, $m_1^{\#} = 0.05$	$m_0^{\#} = 0.97$, $m_1^{\#} = 0.03$	original
penalized logistic regression	.231(13.33)	.082(8.68)	.028(5.22)	0(0)
linear discriminant analysis	.223(13.17)	.096(9.32)	.023(4.74)	0(0)
support vector machine	.220(13.11)	.093(9.19)	.026(5.03)	.004(2.00)
random forest	.238(13.47)	.080(8.58)	.019(4.32)	0(0)

Table 10: Averages of (approximate) true type II errors by Algorithm $1^{\#}$ over 1,000 repetitions for the email spam data. Standard errors ($\times 10^{-3}$) in parentheses.

	$m_0^{\#} = 0.93$, $m_1^{\#} = 0.07$	$m_0^{\#} = 0.95$, $m_1^{\#} = 0.05$	$m_0^{\#} = 0.97$, $m_1^{\#} = 0.03$	original
penalized logistic regression	.165(2.04)	.205(2.65)	.254(3.10)	.272(2.71)
linear discriminant analysis	.213(2.54)	.226(3.05)	.314(3.37)	.314(2.77)
support vector machine	.138(1.20)	.183(3.15)	.199(2.11)	.218(1.93)
random forest	.102(.78)	.120(1.13)	.143(1.41)	.152(1.54)

17 in Appendix 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 (Appendix 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 linear discriminant analysis model, i.e., $X^0 \sim \mathcal{N}(\mu_0, \Sigma)$ and $X^1 \sim \mathcal{N}(\mu_1, \Sigma)$.

Appendices

A. SUMMARY OF SAMPLING SCHEME

This section summarizes our sampling scheme and related notations for the readers' convenience. First, to review the NP paradigm and to make a contrast with the corrupted setting, we introduced the notation for uncorrupted samples: 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. To construct the original NP umbrella algorithm (for uncorrupted data), $\mathcal{S}^0$ is randomly split into $\mathcal{S}^0 = \mathcal{S}_{\mathrm{b}}^0 \cup \mathcal{S}_{\mathrm{t}}^0$, where the subscript b reinforces that this part is to train a *base algorithm* (e.g., logistic regression, random forest), and the subscript t reinforces that this part of the data is to find the *threshold*. For the uncorrupted scenario, we do not split $\mathcal{S}^1$. All $\mathcal{S}^1$ are used together with $\mathcal{S}_{\mathrm{b}}^0$ to train a base algorithm.

For the corrupted scenario, which is the focus of our paper, we assume the following sampling scheme for methodology and theory development. Let $\tilde{\mathcal{S}}^0 = \{\tilde{X}_j^0\}_{j=1}^{N_0}$ be *corrupted* class 0 observations and $\tilde{\mathcal{S}}^1 = \{\tilde{X}_j^1\}_{j=1}^{N_1}$ be *corrupted* class 1 observations. The sample sizes N_0 and N_1 are considered to be non-random numbers. The split for the corrupted scenario is more complicated than the uncorrupted counterpart. Concretely, we split $\tilde{\mathcal{S}}^0$ into three parts: $\tilde{\mathcal{S}}^0 = \tilde{\mathcal{S}}_{\mathrm{b}}^0 \cup \tilde{\mathcal{S}}_{\mathrm{t}}^0 \cup \tilde{\mathcal{S}}_{\mathrm{e}}^0$, and split $\tilde{\mathcal{S}}^1$ into two parts $\tilde{\mathcal{S}}^1 = \tilde{\mathcal{S}}_{\mathrm{b}}^1 \cup \tilde{\mathcal{S}}_{\mathrm{e}}^1$. The subscripts b and t have the same meaning as the uncorrupted case while the subscript e stands for *estimation*, and $\tilde{\mathcal{S}}_{\mathrm{e}}^0$ and $\tilde{\mathcal{S}}_{\mathrm{e}}^1$ are used to estimate a correction term to account for the label noise.

Given the above decomposition of $\tilde{\mathcal{S}}^0$ and $\tilde{\mathcal{S}}^1$, we also used $\tilde{S}_{\mathrm{b}} = \tilde{\mathcal{S}}_{\mathrm{b}}^0 \cup \tilde{\mathcal{S}}_{\mathrm{b}}^1$ to denote all corrupted class 0 and class 1 observations that are used to train the base algorithm in the label-noise-adjusted NP umbrella algorithm. The sample size n is reserved for $|\mathcal{S}_{\mathrm{t}}^0|$ in the uncorrupted scenario, or for $|\tilde{\mathcal{S}}_{\mathrm{t}}^0|$ in the corrupted scenario. The other sub-sample size notations are all for the corrupted scenario. In particular, $n_{\mathrm{b}} = |\tilde{S}_{\mathrm{b}}| = |\tilde{\mathcal{S}}_{\mathrm{b}}^0 \cup \tilde{\mathcal{S}}_{\mathrm{b}}^1|$, $n_{\mathrm{e}}^0 = |\tilde{S}_{\mathrm{e}}^0|$, and $n_{\mathrm{e}}^1 = |\tilde{S}_{\mathrm{e}}^1|$.

B. BINARY SEARCH ALGORITHM

Here r is an error for stopping criterion of this binary search. That is, the algorithm stops when $\left|\sum_{j=k}^{n} \binom{n}{j} (1-\alpha_{\mathrm{middle}})^j \alpha_{\mathrm{middle}}^{n-j} - \delta\right| \leq r$.

Algorithm 2: Binary Search For $\alpha_{k,\delta}$

Input : δ: a small tolerance level, $0 < \delta < 1$
k, n: two integers such that $k \leq n$
r: a small number for error (we implement $r = 10^{-5}$ in our numerical analysis)

$\alpha_{\min} \leftarrow 0$
$\alpha_{\max} \leftarrow 1$
$\delta_{\max} \leftarrow \sum_{j=k}^{n} \binom{n}{j}(1-\alpha_{\min})^j \alpha_{\min}^{n-j}$
$\delta_{\min} \leftarrow \sum_{j=k}^{n} \binom{n}{j}(1-\alpha_{\max})^j \alpha_{\max}^{n-j}$
$E \leftarrow 2$
while $E > r$ **do**
  $\alpha_{\text{middle}} \leftarrow (\alpha_{\min} + \alpha_{\max})/2$
  $\delta_{\text{middle}} \leftarrow \sum_{j=k}^{n} \binom{n}{j}(1-\alpha_{\text{middle}})^j \alpha_{\text{middle}}^{n-j}$
  if $\delta_{\text{middle}} = \delta$ **then**
    Output: α_{middle}
  else if $\delta_{\text{middle}} > \delta$ **then**
    $\alpha_{\text{middle}} \leftarrow \alpha_{\min}$
  else
    $\alpha_{\text{middle}} \leftarrow \alpha_{\max}$
  end
  $E \leftarrow |\delta_{\text{middle}} - \delta|$
end
Output: α_{middle}

C. AN EXAMPLE FOR ASSUMPTION 3

Example 3. *Under the same distributional setting as in Example 1, let $\hat{T}$ be trained by linear discriminant analysis (LDA) on $\tilde{\mathcal{S}}_{\text{b}}$; that is $\hat{T}(X) = \hat{\tilde{\sigma}}^{-2}(\hat{\tilde{\mu}}_1 - \hat{\tilde{\mu}}_0)X$, in which $\hat{\tilde{\mu}}_0$ and $\hat{\tilde{\mu}}_1$ are the sample means of corrupted class 0 and 1 observations, respectively, and $\hat{\tilde{\sigma}}^2$ is the pooled sample variance. For any $z \in \mathbb{R}$, by Lemma 4 in the Appendix, we have*

$$\tilde{F}_0^{\hat{T}}(z) - \tilde{F}_1^{\hat{T}}(z) = (m_0 - m_1)\left(F_0^{\hat{T}}(z) - F_1^{\hat{T}}(z)\right).$$

Therefore, when $m_0 > m_1$ (as assumed in Assumption 1), $\tilde{F}_0^{\hat{T}}(z) > \tilde{F}_1^{\hat{T}}(z)$ is equivalent to $F_0^{\hat{T}}(z) > F_1^{\hat{T}}(z)$. We first fix $\tilde{\mathcal{S}}_{\text{b}}$, then $\hat{T}(X^0) \sim \mathcal{N}(\hat{\tilde{\sigma}}^{-2}(\hat{\tilde{\mu}}_1 - \hat{\tilde{\mu}}_0)\mu_0, \hat{\tilde{\sigma}}^{-4}(\hat{\tilde{\mu}}_1 - \hat{\tilde{\mu}}_0)^2\sigma^2)$ and $\hat{T}(X^1) \sim \mathcal{N}(\hat{\tilde{\sigma}}^{-2}(\hat{\tilde{\mu}}_1 - \hat{\tilde{\mu}}_1)\mu_0, \hat{\tilde{\sigma}}^{-4}(\hat{\tilde{\mu}}_1 - \hat{\tilde{\mu}}_0)^2\sigma^2)$. Since these two distributions are two normal with the same variance and different means, $F_0^{\hat{T}}(z) > F_1^{\hat{T}}(z)$ as long as $\hat{\tilde{\sigma}}^{-2}(\hat{\tilde{\mu}}_1 - \hat{\tilde{\mu}}_0)\mu_0 < \hat{\tilde{\sigma}}^{-2}(\hat{\tilde{\mu}}_1 - \hat{\tilde{\mu}}_0)\mu_1$, or equivalently, $(\hat{\tilde{\mu}}_1 - \hat{\tilde{\mu}}_0)(\mu_1 - \mu_0) > 0$. By Lemma 4 in the Appendix, this condition can be written as $(\hat{\tilde{\mu}}_1 - \hat{\tilde{\mu}}_0)(\tilde{\mu}_1 - \tilde{\mu}_0)/(m_0 - m_1) > 0$, where $\tilde{\mu}_0$ and $\tilde{\mu}_1$ are the means of $\tilde{X}^0$ and $\tilde{X}^1$ respectively.

When $m_0 > m_1$, this is further equivalent to $(\hat{\tilde{\mu}}_1 - \hat{\tilde{\mu}}_0)(\tilde{\mu}_1 - \tilde{\mu}_0) > 0$. Then Assumption 3 follows from the law of large numbers.

D. ADDITIONAL NUMERICAL RESULTS

D.1 Additional Simulations

We apply Algorithm $1^{\#}$ in Simulation 6. For $m_0^{\#}$ and $m_1^{\#}$ needed in Algorithm $1^{\#}$, we use the estimators proposed by Liu and Tao (2016). Technically, Liu and Tao (2016) estimates the "flip rates" $\mathbb{P}\left(\tilde{Y} = 1 \mid Y = 0\right)$ and $\mathbb{P}\left(\tilde{Y} = 0 \mid Y = 1\right)$. Our corruption levels can be derived from flip rates by the Bayes theorem.

Simulation 6. *The distributional setting is the same as in Simulation 2. For different $(m_0, m_1, \alpha, \delta)$ combinations, the (approximate) type I error violation rates and averages of (approximate) true type II errors generated by Algorithm $1^{\#}$ are reported in Tables 11 and 12, respectively.*

Table 11: (Approximate) type I error violation rates over 1,000 repetitions for Simulation 6. Standard errors ($\times 10^{-3}$) in parentheses.

N	$m_0 = .95, m_1 = .05$ $\alpha = .05, \delta = .05$	$m_0 = .9, m_1 = .1$ $\alpha = .05, \delta = .05$	$m_0 = .95, m_1 = .05$ $\alpha = .1, \delta = .1$	$m_0 = .9, m_1 = .1$ $\alpha = .1, \delta = .1$
200	.067(7.91)	.068(7.96)	.131(10.67)	.101(9.53)
500	.084(8.78)	.083(8.73)	.134(10.78)	.115(10.09)
1,000	.463(15.78)	.182(12.21)	.497(15.82)	.197(12.58)
2,000	.665(14.93)	.190(12.41)	.695(14.57)	.209(12.86)

Table 12: Averages of (approximate) true type II errors over 1,000 repetitions for Simulation 6. Standard errors ($\times 10^{-3}$) in parentheses.

N	$m_0 = .95, m_1 = .05$ $\alpha = .05, \delta = .05$	$m_0 = .9, m_1 = .1$ $\alpha = .05, \delta = .05$	$m_0 = .95, m_1 = .05$ $\alpha = .1, \delta = .1$	$m_0 = .9, m_1 = .1$ $\alpha = .1, \delta = .1$
200	.431(9.36)	.589(9.79)	.150(2.44)	.221(4.86)
500	.219(3.60)	.391(7.25)	.115(.95)	.145(1.49)
1,000	.140(.99)	.190(2.53)	.082(.72)	.107(1.01)
2,000	.128(.82)	.175(1.75)	.073(.65)	.101(.88)

In this simulation, Algorithm $1^{\#}$ fails to control the type I error with pre-specified high probability. Similar results on additional distributional settings can be found in Simulation 7 of Appendix D.1. One might wonder: if we were to use other estimators of m_0 and m_1, will the result be different? The answer is that the usually "good" estimators do not serve for the purpose of high

probability control on type I error. For example, Simulation 8 in Appendix D.1 uses consistent and unbiased estimators of m_0 and m_1, but Algorithm $1^{\#}$ still fails to control the type I error.

Simulation 7. *The distributional setting is the same as in Simulation 1. For different* $(m_0, m_1, \alpha, \delta)$ *combinations, the (approximate) true type I errors generated by Algorithm* $1^{\#}$ *are reported in Table 13.*

Table 13: (Approximate) type I error violation rates over 1,000 repetitions for Simulation 6. Standard errors ($\times 10^{-3}$) in parentheses.

N	$m_0 = .95, m_1 = .05$ $\alpha = .05, \delta = .05$	$m_0 = .9, m_1 = .1$ $\alpha = .05, \delta = .05$	$m_0 = .95, m_1 = .05$ $\alpha = .1, \delta = .1$	$m_0 = .9, m_1 = .1$ $\alpha = .1, \delta = .1$
200	.430(15.66)	.512(15.81)	.530(15.79)	.504(15.82)
500	.694(14.58)	.488(15.81)	.758(13.55)	.570(15.66)
1,000	.940(7.51)	.788(13.47)	.953(6.70)	.805(12.54)
2,000	.950(6.90)	.792(12.80)	.957(6.42)	.818(12.21)

Simulation 8. *The distributional setting is the same as in Simulation 1. The* $m_0^{\#}$ *and* $m_1^{\#}$ *are generated from* $\mathcal{N}(m_0, 1/N)$ *and* $\mathcal{N}(m_1, 1/N)$*, respectively. The (approximate) type I error violation rates generated by Algorithm* $1^{\#}$ *for one combination (*$m_0 = .95$*,* $m_1 = .05$*,* $\alpha = .1$ *and* $\delta = .1$*) are reported in Table 14.*

D.2 CIFAR10 data analysis

In this section we apply Algorithm 1 to the CIFAR10 dataset (Krizhevsky et al., 2009). As we focus on binary classification problems, we merge the ten categories of the CIFAR10 dataset into two: "vehicles" and "non-vehicles." The class "vehicles," encoded as 0, contains the original "automobile" and "truck" classes, and the class "non-vehicles," encoded as 1, contains the other eight original classes. Then type I/II errors are defined accordingly. We employ the NP paradigm to this modified dataset to prioritize control over the chance of failing to detect vehicles.

The original CIFAR10 dataset has pre-specified training and test sets, but the number of class 0 observations in the test set is too small (2,000 in total) to produce a reliable approximation to population-level type I error. Furthermore, given that the train-test procedure has to be repeated multiple times to approximate the type I error violation rate, a fixed train-test split throughout all repetitions does not serve our purpose. As such, we perform stratified splits to the whole modified CIFAR10 dataset (with the newly assigned labels). In particular, 20% true class 0 observations and

Table 14: (Approximate) type I error violation rates over 1,000 repetitions for Simulation 8. Standard errors ($\times 10^{-2}$) in parentheses.

N	(approximate) violation rate
200	.193(1.25)
500	.208(1.28)
1,000	.186(1.23)
2,000	.203(1.27)

20% true class 1 observations are randomly selected to form the new training set and the remaining observations form the evaluation set. The training and evaluation sets contain 12,000 and 48,000 observations, respectively. Moreover, the labels of all training observations are artificially corrupted by the same method as in Section 5.2 with $m_0 = 0.95$ and $m_1 = 0.05$. By Bayes theorem, the flip rates $r_0 = \mathbb{P}\left(\tilde{Y} = 1 \mid Y = 0\right)$ and $r_1 = \mathbb{P}\left(\tilde{Y} = 0 \mid Y = 1\right)$ are 0.2083 and 0.0104, respectively. We apply Algorithm 1 (with the parameter choice $\alpha = \delta = 0.1$ and CNN as the base algorithm) to the training set with corrupted labels and obtain a classifier. Then, the classifier is applied to the untouched evaluation set to calculate the (approximate) true type I and II errors. This procedure is repeated 1,000 times.

In the main text, we have shown that Algorithm $1^{\#}$ with under-estimates of corruption levels fulfills the goal of high-probability control over the type I error, while other benchmark algorithms do not. To avoid delivering redundant messages, we only apply Algorithm 1 to the modified CIFAR10 dataset since our primary interest is the type I error control. The (approximate) type I error violation rate and average of (approximate) true type II errors are presented in Table 15. Clearly, Algorithm 1 is able to achieve high probability control of the true type I error under the specified level.

Table 15: (Approximate) type I error violation rate, and average of (approximate) true type II errors by Algorithm 1 over 1,000 repetitions for the modified CIFAR10 dataset. Standard errors ($\times 10^{-3}$) in parentheses.

	(approximate) violation rate	average of (approximate) true type II errors
Algorithm 1 with CNN as base algorithm	.099(9.45)	.150(.87)

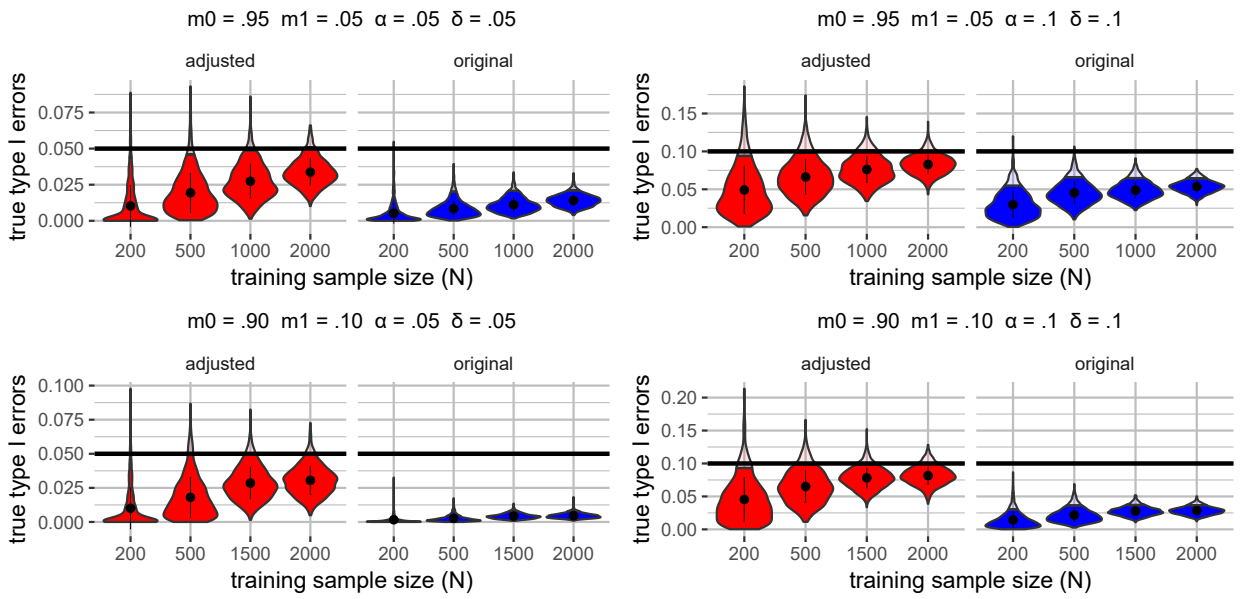

Figure 5: Violin plots for (approximate) true type I errors of Simulation 1.

D.3 Violin plots for Section 5

In this section, we present the violin plots (Figures 5 - 10) for Simulations 1-3 in Section 5. The violin plots for the (approximate) true type I and type II errors over these 1,000 repetitions are plotted for each $(m_0, m_1, \alpha, \delta)$ combination. Take Figures 5 and 6 as an example, the two rows in each figure respectively correspond to the $m_1 = 0.95, m_1 = 0.05$ and $m_0 = 0.85, m_1 = 0.15$ settings, while the two columns respectively correspond to $\alpha = 0.05, \delta = 0.05$ and $\alpha = 0.10, \delta = 0.10$. The area of every plot with lighter color represents true type I errors above the $1 - \delta$ quantile while the area with darker color represents true type I errors below the $1 - \delta$ quantile. The black dots represent the average of true type I/II errors and the bars above and below the dots represent standard deviations.

D.4 Tables for Section 5

In this section, we present Table 16 for in Simulation 5 in Section 5.1 and Table 17 for the email spam data analysis in Section 5.2.

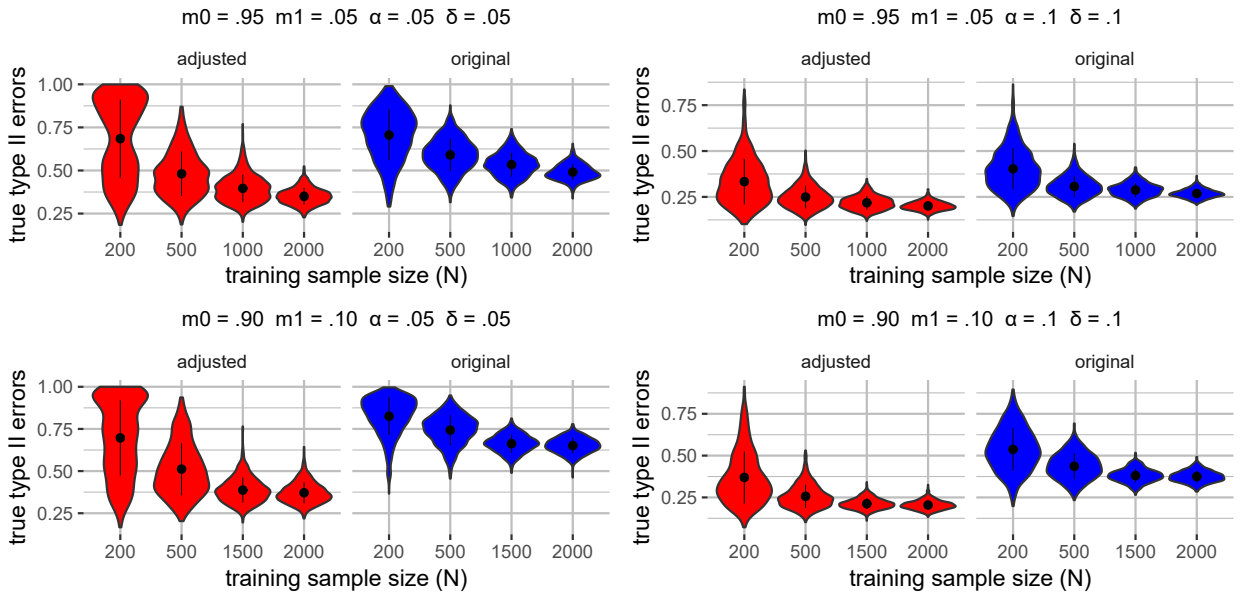

Figure 6: Violin plots for (approximate) true type II errors of Simulation 1.

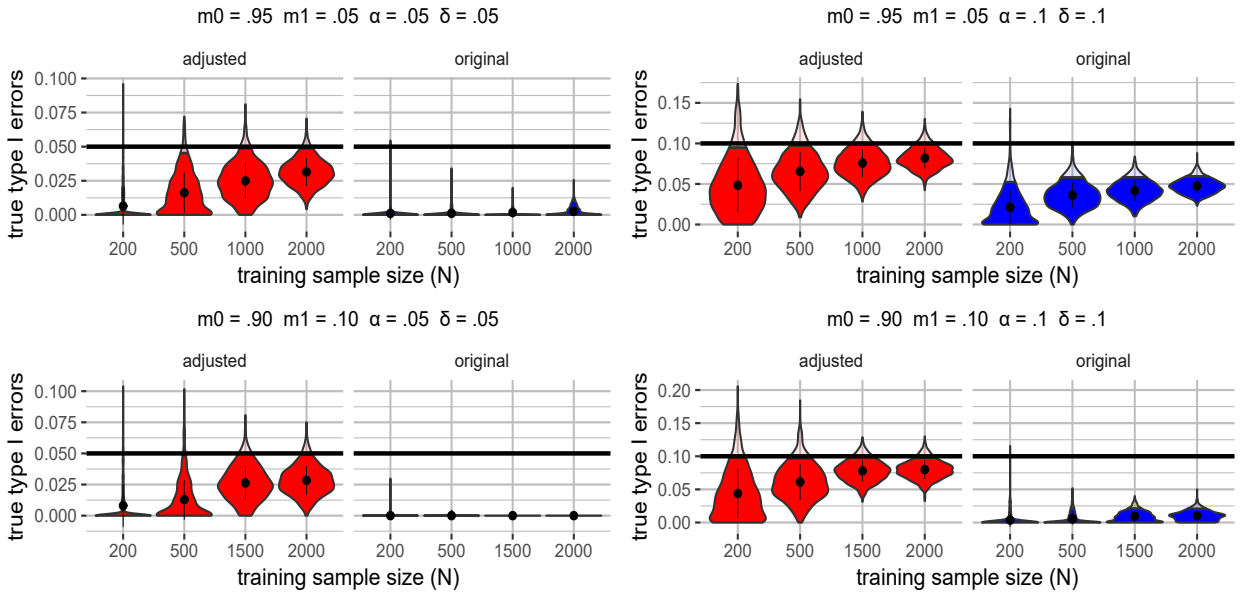

Figure 7: Violin plots for (approximate) true type I errors of Simulation 2.

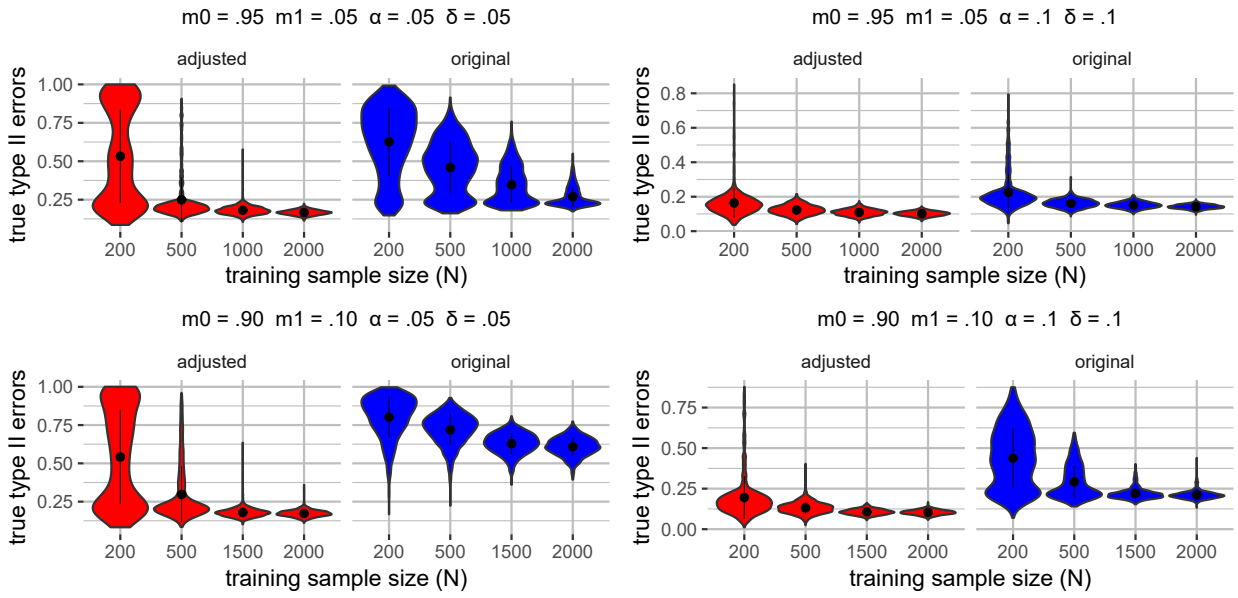

Figure 8: Violin plots for (approximate) true type II errors of Simulation 2.

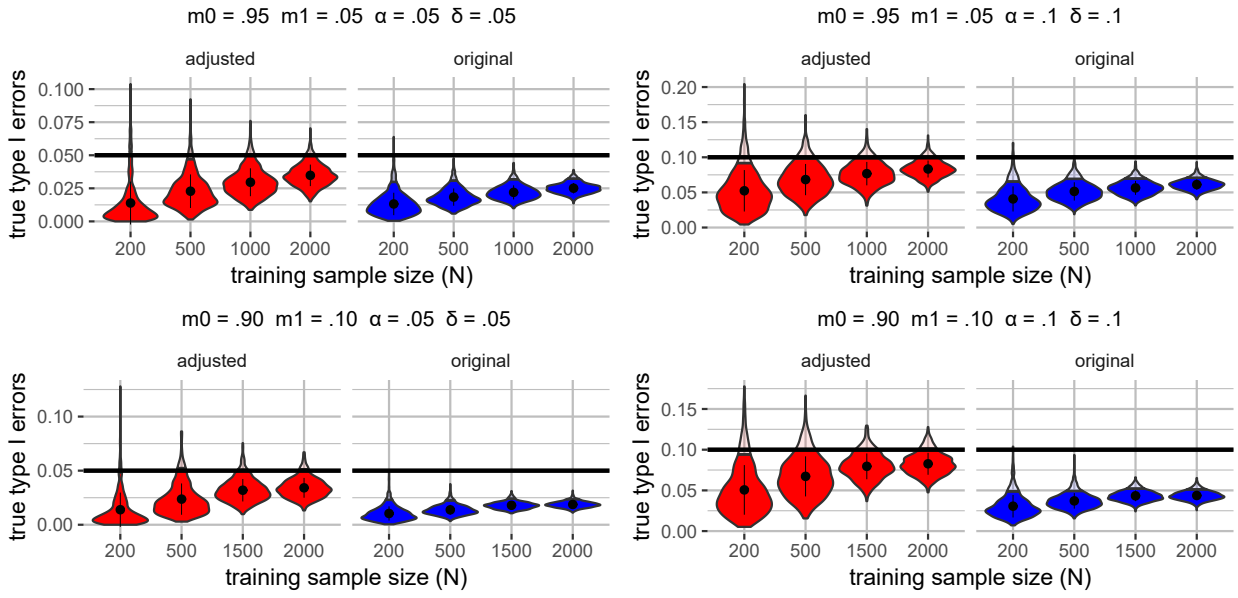

Figure 9: Violin plots for (approximate) true type I errors of Simulation 3.

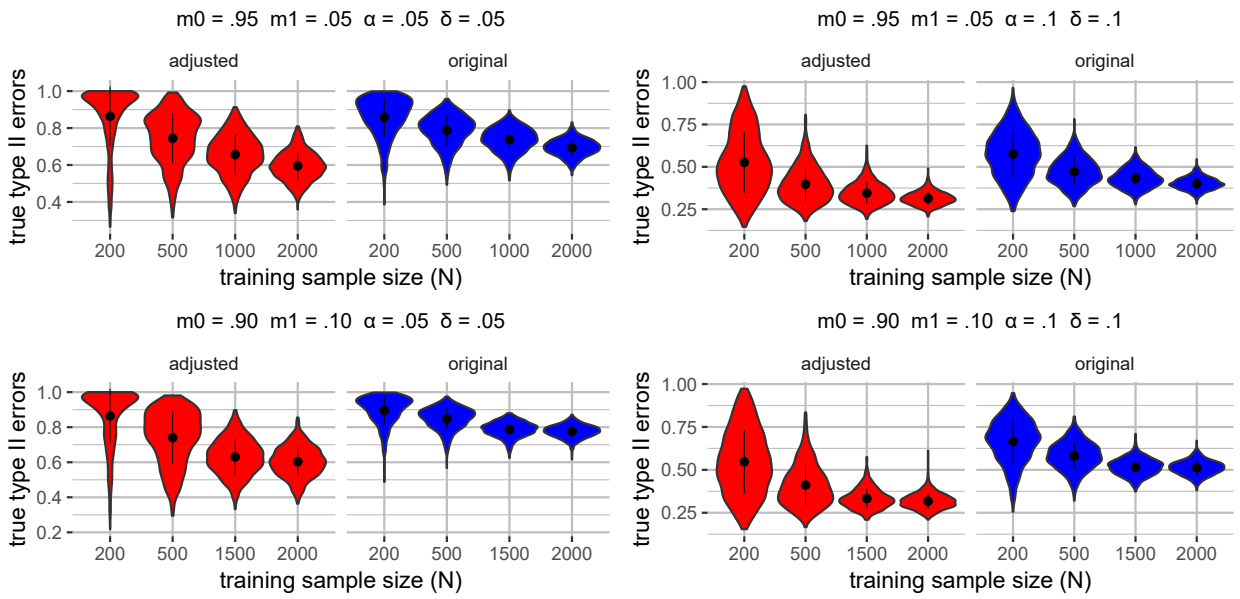

Figure 10: Violin plots for (approximate) true type II errors of Simulation 3.

Table 16: Averages of (approximate) true type II errors over 1,000 repetitions for Simulation 5 ($m_0 = .95$, $m_1 = .05$, $\alpha = .1$ and $\delta = .1$). Standard errors ($\times 10^{-3}$) in parentheses.

algorithms	N			
	200	500	1,000	2,000
T-revision	.165(4.32)	.153(4.08)	.146(3.52)	.147(4.27)
backward loss correction (known corruption level)	.151(.77)	.139(.70)	.161(.71)	.199(.69)
backward loss correction (unknown corruption level)	.158(.02)	.163(.02)	.186(.01)	.192(.01)

D.5 Alternative implementation with a positive ε

In this section, we repeat the numerical studies for Simulations 1-3 in Section 5 but replace k^* in Algorithm 1 by $\min\{k \in \{1, \ldots, n\} : \alpha_{k,\delta} - \hat{D}^+(t_{(k)}) \leq \alpha - \varepsilon\}$ where $\varepsilon = 0.0001$. The results are presented in Figures 11 - 16. Numerical evidence shows that whether to have a small positive ε in selection of k^* does not actually affect much the performance of label-noise-adjusted umbrella algorithm. Thus, as a simpler algorithm is always preferred, we recommend taking $\varepsilon = 0$.

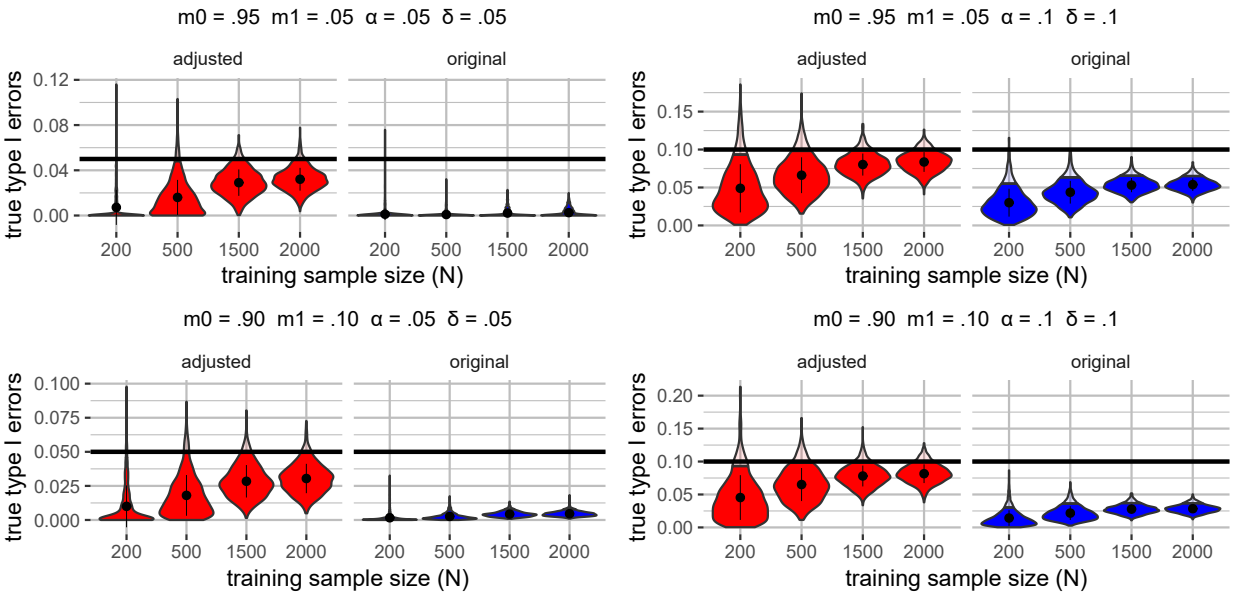

Figure 11: Violin plots for (approximate) true type I errors of Simulation 1.

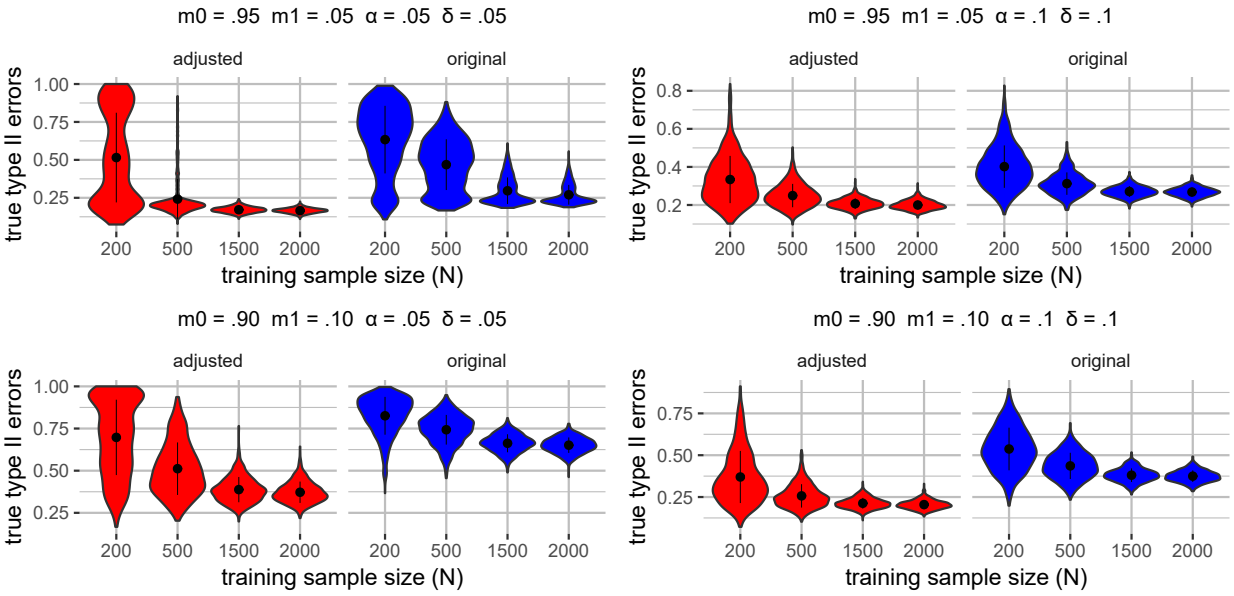

Figure 12: Violin plots for (approximate) true type II errors of Simulation 1.

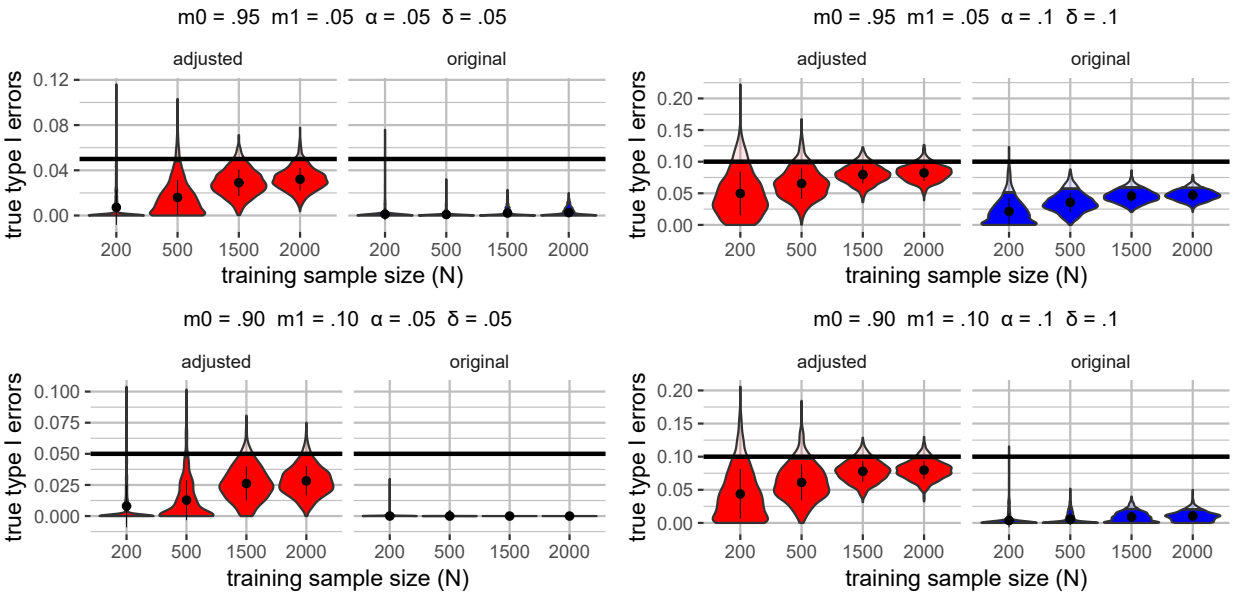

Figure 13: Violin plots for (approximate) true type I errors of Simulation 2.

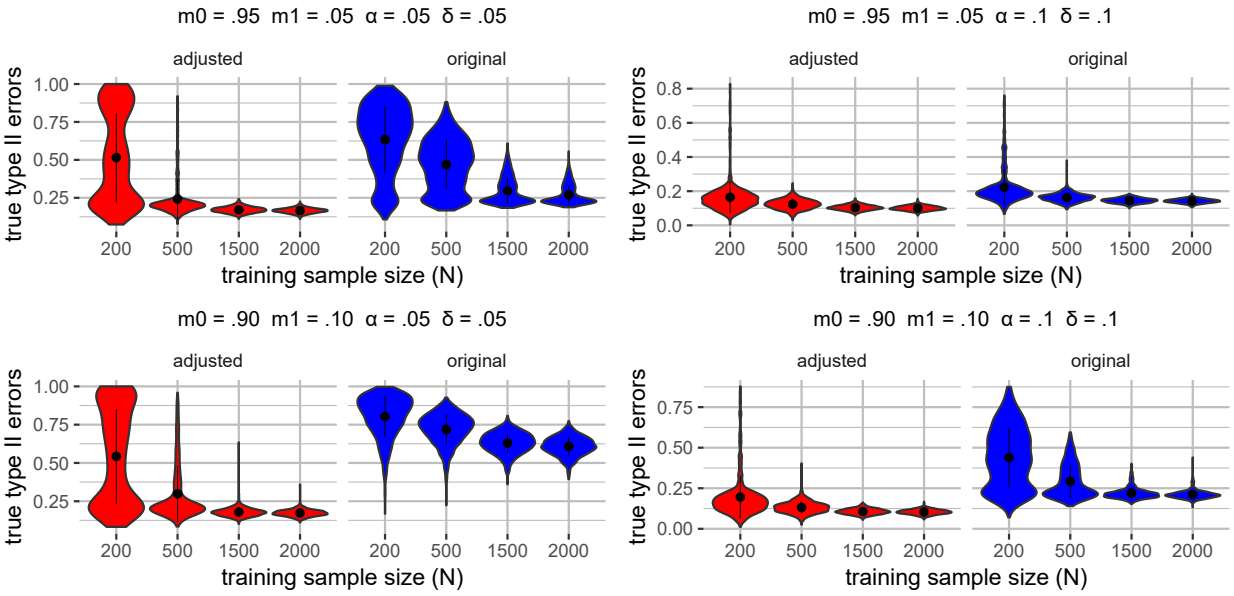

Figure 14: Violin plots for (approximate) true type II errors of Simulation 2.

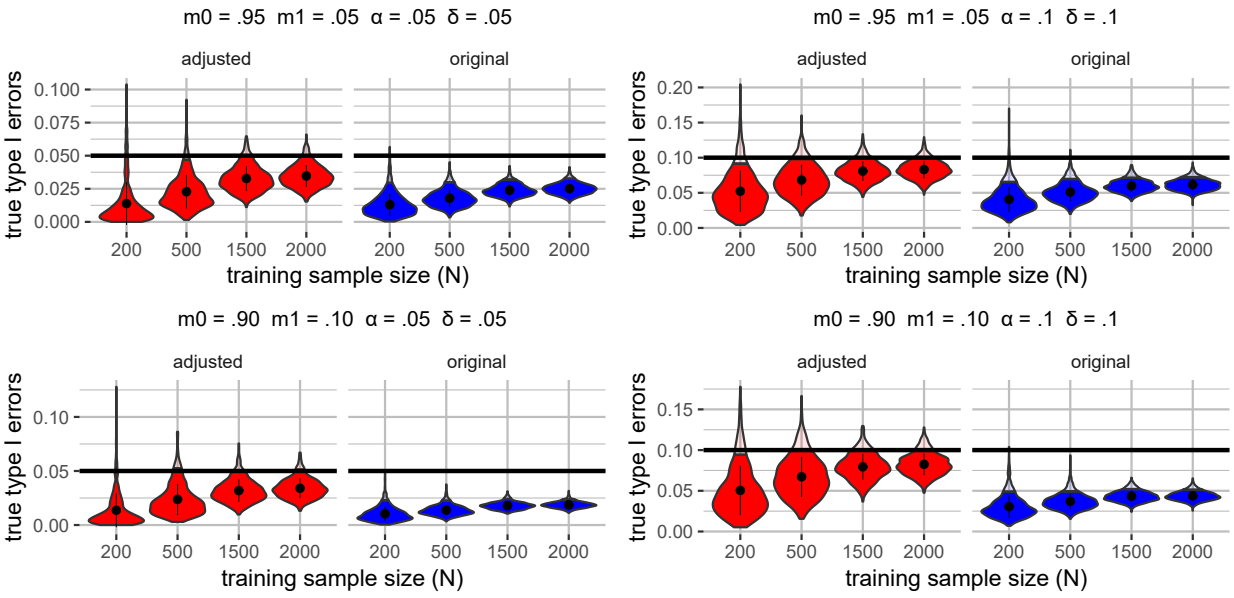

Figure 15: Violin plots for (approximate) true type I errors of Simulation 3.

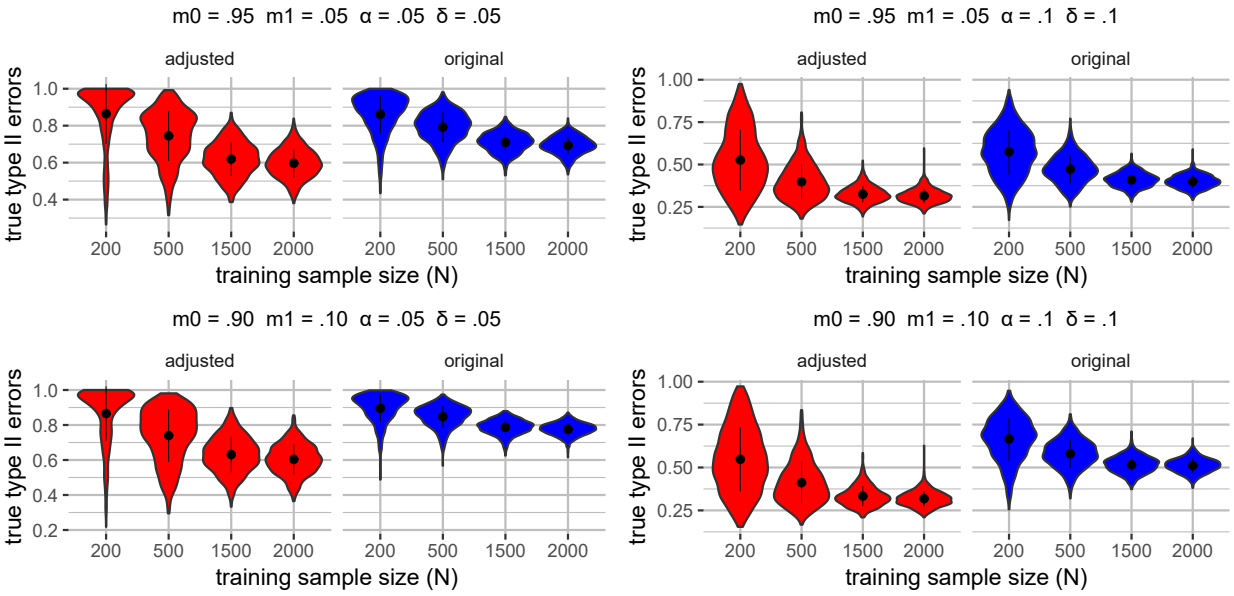

Figure 16: Violin plots for (approximate) true type II errors of Simulation 3.

Table 17: (Approximate) type I error violation rates, and averages of (approximate) true type II error by benchmark algorithms over 1,000 repetitions for the email spam data. Standard errors ($\times 10^{-3}$) in parentheses.

	(approximate) violation rate	average of (approximate) true type II errors
T-revision	.829(11.91)	.414(7.01)
backward loss correction (known corruption level)	.831(11.86)	.573(5.49)
backward loss correction (unknown corruption level)	.750(13.70)	.631(5.51)

E. EXTRA LEMMAS

Lemma 4. *Under Assumption 1, for any measurable function $T : \mathbb{R}^d \to \mathbb{R}$ and arbitrary number $z \in \mathbb{R}$, we have*

$$\tilde{F}_0^T(z) = m_0 F_0^T(z) + (1 - m_0) F_1^T(z) \quad \textit{and} \quad \tilde{F}_1^T(z) = m_1 F_0^T(z) + (1 - m_1) F_1^T(z)\,.$$

Furthermore,

$$\mathbb{E}\tilde{X}^0 = m_0 \mathbb{E}X^0 + (1 - m_0)\mathbb{E}X^1 \quad \textit{and} \quad \mathbb{E}\tilde{X}^1 = m_1 \mathbb{E}X^0 + (1 - m_1)\mathbb{E}X^1\,.$$

Proof. The first two equations can be proved in the similar way. So we will only show the first equation. By Assumption 1, for any Borel set A,

$$\tilde{P}_0(T^{-1}(A)) = m_0 P_0(T^{-1}(A)) + (1 - m_0) P_1(T^{-1}(A))\,.$$

Then, select $A = (-\infty, z]$ and the result follows.

Similarly, the proof of last two equations are similar in nature. So we are going to show

$\mathbb{E}\tilde{X}^0 = m_0\mathbb{E}X^0 + (1-m_0)\mathbb{E}X^1$. Note that by Assumption 1,

$$\begin{aligned}
\mathbb{E}\tilde{X}^0 &= \int_0^\infty (1-\tilde{P}_0(X \le x))dx - \int_{-\infty}^0 \tilde{P}_0(X \le x)dx \\
&= m_0\left(\int_0^\infty (1-P_0(X \le x))dx - \int_{-\infty}^0 P_0(X \le x)dx\right) \\
&+ (1-m_0)\left(\int_0^\infty (1-P_1(X \le x))dx - \int_{-\infty}^0 P_1(X \le x)dx\right) \\
&= m_0\mathbb{E}X^0 + (1-m_0)\mathbb{E}X^1\,.
\end{aligned}$$

□

Lemma 5. *For any $k \in \{1,\dots,n\}$ and $\delta \in (0,1)$, a unique $\alpha_{k,\delta}$ exists. Moreover, under Assumption 2, $k_* = \min\{k \in \{1,\dots,n\} : \alpha_{k,\delta} \le \alpha\}$.*

Proof. Let $h_k(x) = \sum_{j=k}^n \binom{n}{k} x^{n-j}(1-x)^j$ for any $k \in \{1,\dots,n\}$. Then, one can show, for $k \le n-1$ and $x \in (0,1)$,

$$\begin{aligned}
h_k'(x) &= \sum_{j=k}^{n-1}(n-j)\binom{n}{j}x^{n-j-1}(1-x)^j - \sum_{j=k}^{n} j\binom{n}{j}x^{n-j}(1-x)^{j-1} \\
&= n\sum_{i=k+1}^{n}\binom{n}{i-1}x^{n-i}(1-x)^{i-1} - n\sum_{j=k}^{n}\binom{n}{j-1}x^{n-j}(1-x)^{j-1} \\
&= -n\binom{n}{k-1}x^{n-k}(1-x)^{k-1}\,,
\end{aligned}$$

which is negative. Thus, $h_k(x)$ is strictly decreasing on $(0,1)$ for $k \le n-1$. Furthermore, $h_n(x) = (1-x)^n$ which is also strictly decreasing on $(0,1)$. Since for any k, $h_k(0) = 1$ and $h_k(1) = 0$, there exists a unique $\alpha_{k,\delta}$ such that $h_k(\alpha_{k,\delta}) = \delta$.

Recall that k_* is defined as the smallest k such that $h_k(\alpha) \le \delta$. Meanwhile, by monotonicity, for any k, the inequality $h_k(\alpha) \le \delta$ is equivalent to $\alpha_{k,\delta} \le \alpha$. Assumption 2 guarantees the existence of k such that $h_k(\alpha)$. Therefore it also guarantees the existence of k such that $\alpha_{k,\delta} \le \alpha$. Then, for any δ, $\{k \in \{1,\dots,n\} : h_k(\alpha) \le \delta\} = \{k \in \{1,\dots,n\} : \alpha_{k,\delta} \le \alpha\}$. Then, $k_* = \min\{k \in \{1,\dots,n\} : \alpha_{k,\delta} \le \alpha\}$. □

Lemma 6. *Given a random variable $X \in \mathbb{R}^d$ with probability measure P and a deterministic*

measurable function $T : \mathbb{R}^d \to \mathbb{R}$*, define probability measure* $P^T(B) = P(T(X) \in B)$ *for any Borel set* B*. Furthermore, denote the distribution functions of* P *and* P^T *as* F *and* F^T*, respectively. Let* $X_1, X_2, \ldots, X_n \sim X$ *be i.i.d. random variables. Moreover, let* $\hat{F}^T(z) = \frac{1}{n}\sum_{j=1}^n \mathbb{1}\{T(X_j) \le z\}$ *for any* $z \in \mathbb{R}$*. Then, for any* $\xi > 0$

$$P\left(\sup_{z\in\mathbb{R}}\left|\hat{F}^T(z) - F^T(z)\right| > \xi\right) \le 2e^{-2n\xi^2}.$$

Proof. Note that $X_1, X_2, \ldots, X_n$ are i.i.d. random variables, then so are $T(X_1), T(X_2), \ldots, T(X_n)$. Denote $T_j = T(X_j)$, then T_j has the probability measure P^T. Note that the Dvoretsky-Kiefer-Wolfowitz inequality says,

$$P^T\left(\sup_{z\in\mathbb{R}}\left|\frac{1}{n}\sum_{j=1}^n \mathbb{1}\{T_j \le z\} - F^T(z)\right| > \xi\right) \le 2e^{-2n\xi^2}.$$

Then, it suffices to show the left hand side of above inequality equals $P\left(\sup_{z\in\mathbb{R}}\left|\hat{F}^T(z) - F^T(z)\right| > \xi\right)$. Towards that, denote

$$f_n(x_1, x_2, \ldots, x_n) = \mathbb{1}\left\{\sup_{z\in\mathbb{R}}\left|\frac{1}{n}\sum_{j=1}^n \mathbb{1}\{T(x_j) \le z\} - F^T(z)\right| > \xi\right\},$$

and

$$f_0(t_1, t_2, \ldots, t_n) = \mathbb{1}\left\{\sup_{z\in\mathbb{R}}\left|\frac{1}{n}\sum_{j=1}^n \mathbb{1}\{t_j \le z\} - F^T(z)\right| > \xi\right\}.$$

By Fubini's theorem, it holds that

$$P\left(\sup_{z\in\mathbb{R}}\left|\hat{F}^T(z) - F^T(z)\right| > \xi\right) = \mathbb{E}_1\mathbb{E}_2\ldots\mathbb{E}_n f_n(X_1, X_2, \ldots, X_n),$$

and

$$P^T\left(\sup_{z\in\mathbb{R}}\left|\frac{1}{n}\sum_{j=1}^n \mathbb{1}\{T_j \le z\} - F^T(z)\right| > \xi\right) = \mathbb{E}_1^T\mathbb{E}_2^T\ldots\mathbb{E}_n^T f_0(T_1, T_2, \ldots, T_n),$$

where $\mathbb{E}_j$ and $\mathbb{E}_j^T$ are the expectations taken with respect to X_j and T_j under the probability measures P and P^T, respectively. Thus, it suffices to show

$$\mathbb{E}_1\mathbb{E}_2\ldots\mathbb{E}_n f_n(X_1,X_2,\ldots,X_n)=\mathbb{E}_1^T\mathbb{E}_2^T\ldots\mathbb{E}_n^T f_0(T_1,T_2,\ldots,T_n)\,,$$

and we will show this by induction. Denote

$$f_l(x_1,x_2,\ldots,x_l,t_{l+1},t_{l+2},\ldots,t_n)=\mathbb{1}\left\{\sup_{z\in\mathbb{R}}\left|\frac{1}{n}\left(\sum_{j=1}^{l}\mathbb{1}\{T(x_j)\le z\}+\sum_{j=l+1}^{n}\mathbb{1}\{t_j\le z\}\right)-F^T(z)\right|>\xi\right\}\,,$$

for any $l\in\{1,2,\ldots,n-1\}$ and $A_{n-1}(x_1,x_2,\ldots,x_{n-1})=\{t_n: f_{n-1}(x_1,x_2,\ldots,x_{n-1},t_n)=1\}$. Then, for any fixed values of $x_1,x_2,\ldots,x_{n-1}$,

$$\begin{aligned}\mathbb{E}_n f_n(x_1,x_2,\ldots,x_{n-1},X_n)&=P(T(X_n)\in A_{n-1}(x_1,x_2,\ldots,x_{n-1}))\\&=P^T(A_{n-1}(x_1,x_2,\ldots,x_{n-1}))\\&=\mathbb{E}_n^T f_{n-1}(x_1,x_2,\ldots,x_{n-1},T_n)\,,\end{aligned}$$

and thus,

$$\mathbb{E}_1\mathbb{E}_2\ldots\mathbb{E}_n f_n(X_1,X_2,\ldots,X_n)=\mathbb{E}_1^T\mathbb{E}_2^T\ldots\mathbb{E}_{n-1}\mathbb{E}_n^T f_{n-1}(X_1,X_2,\ldots,X_{n-1},T_n)\,.$$

Now, assume that for some $l\in\{2,3,\ldots,n\}$,

$$\begin{aligned}&\mathbb{E}_1\mathbb{E}_2\ldots\mathbb{E}_n f_n(X_1,X_2,\ldots,X_n)\\&\qquad=\mathbb{E}_1\mathbb{E}_2\ldots\mathbb{E}_{l-1}\mathbb{E}_l^T\mathbb{E}_{l+1}^T\ldots\mathbb{E}_n^T f_{l-1}(X_1,X_2,\ldots,X_{l-1},T_l,T_{l+1},\ldots,T_n)\,.\end{aligned}$$

Therefore, for any fixed values of $x_1,x_2,\ldots,x_{l-2}$, denote

$$A_{l-2}(x_1,x_2,\ldots,x_{l-2})=\{t_{l-1}:\mathbb{E}_l^T\mathbb{E}_{l+1}^T\ldots\mathbb{E}_n^T f_{l-2}(x_1,x_2,\ldots,x_{t-2},t_{l-1},T_l,\ldots,T_n)=1\}\,,$$

we can have

$$\mathbb{E}_{l-1}\mathbb{E}_l^T\mathbb{E}_{l+1}^T\dots\mathbb{E}_n^T f_{l-1}(x_1,x_2,\dots,x_{l-2},X_{l-1},T_l,T_{l+1},\dots,T_n)$$
$$= P(T(X_{l-1})\in A_{l-2}(x_1,x_2,\dots,x_{l-2})) = P^T(A_{l-2}(x_1,x_2,\dots,x_{l-2}))\,,$$

and thus,

$$\mathbb{E}_{l-1}\mathbb{E}_l^T\mathbb{E}_{l+1}^T\dots\mathbb{E}_n^T f_{l-1}(x_1,x_2,\dots,x_{l-2},X_{l-1},T_l,T_{l+1},\dots,T_n)$$
$$= \mathbb{E}_{l-1}^T\mathbb{E}_l^T\dots\mathbb{E}_n^T f_{l-2}(x_1,x_2,\dots,x_{t-2},T_{l-1},\dots,T_n)\,.$$

Therefore, by the assumption, we have

$$\mathbb{E}_1\mathbb{E}_2\dots\mathbb{E}_n f_n(X_1,X_2,\dots,X_n)$$
$$= \mathbb{E}_1\mathbb{E}_2\dots\mathbb{E}_{l-2}\mathbb{E}_{l-1}^T\mathbb{E}_l^T\dots\mathbb{E}_n^T f_{l-2}(X_1,X_2,\dots,X_{l-2},T_{l-1},T_l,\dots,T_n)\,.$$

We conclude the proof by induction. □

F. PROOFS

Lemma 1. Let's focus on the event of the statement of Assumption 3, whose complement holds with probability at most $\delta_1(n_{\mathrm{b}})$. Meanwhile, by Lemma 4, for any $z\in\mathbb{R}$,

$$\begin{aligned}\tilde{F}_0^{\hat{T}}(z)-\tilde{F}_1^{\hat{T}}(z) &= \left[m_0F_0^{\hat{T}}(z)+(1-m_0)F_1^{\hat{T}}(z)\right]-\left[m_1F_0^{\hat{T}}(z)+(1-m_1)F_1^{\hat{T}}(z)\right]\\ &= (m_0-m_1)\left(F_0^{\hat{T}}(z)-F_1^{\hat{T}}(z)\right)\,.\end{aligned}$$

Furthermore, for any classifier $\phi(X)=\mathbb{1}\{\hat{T}(X)>z\}$

$$\begin{aligned}\tilde{R}_0(\phi)-R_0(\phi) &= \left(1-\tilde{F}_0^{\hat{T}}(z)\right)-\left(1-F_0^{\hat{T}}(z)\right)\\ &= F_0^{\hat{T}}(z)-m_0F_0^{\hat{T}}(z)-(1-m_0)F_1^{\hat{T}}(z)\\ &= (1-m_0)\left(F_0^{\hat{T}}(z)-F_1^{\hat{T}}(z)\right)\,,\end{aligned}$$

which is positive by Assumption 3. Now, let $D(z) = \tilde{R}_0(\phi) - R_0(\phi) > 0$ and therefore $R_0(\hat{\phi}_{k_*}) > \alpha - D(t_{(k_*)})$ is equivalent to $\tilde{R}_0(\hat{\phi}_{k_*}) > \alpha$, whose probability is δ by Proposition 1. To this end, we have shown

$$\mathbb{P}\left(R_0(\hat{\phi}_{k_*}) > \alpha - D(t_{(k_*)})\right) \leq \delta + \delta_1(n_{\mathrm{b}})\,.$$

□

Proof of Lemma 2. By Lemma 5, the set $\{k \in \{1,\dots,n\} : \alpha_{k,\delta} \leq \alpha\}$ is non-empty. Then, the set $\{k \in \{1,\dots,n\} : \alpha_{k,\delta} - \hat{D}^+(t_{(k)}) \leq \alpha\}$ is non-empty since $\hat{D}^+(t_{(k)})$ is non-negative. Then $k^* = \min\{k \in \{1,\dots,n\} : \alpha_{k,\delta} - \hat{D}^+(t_{(k)}) \leq \alpha\}$ exists. Note that $k_* = \min\{k \in \{1,\dots,n\} : \alpha_{k,\delta} \leq \alpha\}$ by Lemma 5. Since $\{k \in \{1,\dots,n\} : \alpha_{k,\delta} \leq \alpha\}$ is a subset of $\{k \in \{1,\dots,n\} : \alpha_{k,\delta} - \hat{D}^+(t_{(k)}) \leq \alpha\}$ by the non-negativeness of $\hat{D}^+$, it can be concluded that $k^* \leq k_*$. □

Proof of Lemma 3. Assumption 1 implies $0 \leq M_{\#} := \frac{1-m_0^{\#}}{m_0^{\#}-m_1^{\#}} \leq M = \frac{1-m_0}{m_0-m_1}$, and thus, $0 \leq \hat{D}^+_{\#}(c) \leq \hat{D}^+(c)$. Then, $\{k \in \{1,\dots,n\} : \alpha_{k,\delta} \leq \alpha\}$, which is non-empty by Assumption 2, is a subset of $\{k \in \{1,\dots,n\} : \alpha_{k,\delta} - \hat{D}^+_{\#}(t_{(k)}) \leq \alpha\}$. This implies $k^*_{\#}$ exists and is smaller than or equal to k_*. Furthermore, $\{k \in \{1,\dots,n\} : \alpha_{k,\delta} - \hat{D}^+_{\#}(t_{(k)}) \leq \alpha\}$ is also a subset of $\{k \in \{1,\dots,n\} : \alpha_{k,\delta} - \hat{D}^+(t_{(k)}) \leq \alpha\}$ and thus, $k^*_{\#} = \min\{k \in \{1,\dots,n\} : \alpha_{k,\delta} - \hat{D}^+_{\#}(t_{(k)}) \leq \alpha\}$ is larger than or equal to k^*. □

Proof of Theorem 1. Let's focus on the event where statement of both Assumption 3 and 4 hold, whose complement has probability less that $\delta_1(n_{\mathrm{b}}) + \delta_2(n_{\mathrm{b}})$. Then, let

$$\mathcal{B}_e = \left\{\sup_{z\in\mathbb{R}} \left|\hat{D}(z) - D(z)\right| \leq 2^{-1}\varepsilon\right\}.$$

It follows from Lemma 6 that

$$\begin{aligned}\mathbb{P}(\mathcal{B}_e^c) &\leq \mathbb{P}\left(\sup_{z\in\mathbb{R}} \left|\hat{\tilde{F}}_0^{\hat{T}}(z) - \tilde{F}_0^{\hat{T}}(z)\right| > \frac{\varepsilon}{4}\right) + \mathbb{P}\left(\sup_{z\in\mathbb{R}} \left|\hat{\tilde{F}}_1^{\hat{T}}(z) - \tilde{F}_1^{\hat{T}}(z)\right| > \frac{\varepsilon}{4}\right) \\ &\leq 2e^{-8^{-1}n_{\mathrm{e}}^0 M^{-2}\varepsilon^2} + 2e^{-8^{-1}n_{\mathrm{e}}^1 M^{-2}\varepsilon^2}\,.\end{aligned}$$

Note that since $D(z)$ is non-negative by Lemma 1, $\left|\hat{D}^+(z) - D(z)\right| \leq \left|\hat{D}(z) - D(z)\right| \leq 2^{-1}\varepsilon$ on $\mathcal{B}_e$.

So, one can conclude that on the event $\mathcal{B}_e$, k^* is chosen from all k such that $\alpha_{k,\delta} - D(t_{(k)}) \leq \alpha - 2^{-1}\varepsilon$. Furthermore, denote $c_k = \inf\{y : \tilde{F}_0^{\hat{T}}(y) \geq kn^{-1}\}$ and $k_0 = \min\{k \in \{1, \ldots, n\} : \alpha_{k,\delta} - D(c_k) \leq \alpha - 4^{-1}\varepsilon\}$. Note that since $\mathcal{D}_T$ is a closed interval, thus c_k is well-defined. Let $\tilde{F}_n^{\hat{T}}$ be the empirical distribution induced by $\mathcal{T}_{\mathrm{t}}$, i.e., for any $z \in \mathbb{R}$,

$$\tilde{F}_n^{\hat{T}}(z) = \frac{1}{n} \sum_{t \in \mathcal{T}_t} \mathbb{1}\{t \leq z\}\,.$$

Denote $\mathcal{B}_t = \{\sup_{z \in \mathbb{R}} |\tilde{F}_n^{\hat{T}}(z) - \tilde{F}_0^{\hat{T}}(z)| \leq 4^{-1}M^{-1}C^{-1}c\varepsilon\}$. Then, by Lemma 6, $\mathbb{P}(\mathcal{B}_t^c) \leq 2e^{-8^{-1}nM^{-2}C^{-2}c^2\varepsilon^2}$. So, it remains to show the probability of true type I error exceeding α is bounded by δ on the set $\mathcal{B}_t \cap \mathcal{B}_e$. Thus, till the end of the proof, we will focus on the intersection of both sets. Note that we have $\tilde{F}_n^{\hat{T}}(t_{(k)}) = kn^{-1}$. Then, Taylor expansion implies

$$\tilde{F}_n^{\hat{T}}(t_{(k)}) - \tilde{F}_0^{\hat{T}}(t_{(k)}) = \tilde{F}_n^{\hat{T}}(t_{(k)}) - \tilde{F}_0^{\hat{T}}(c_k) - \tilde{f}_0^{\hat{T}}(c_k^*)(t_{(k)} - c_k) = -\tilde{f}_0^{\hat{T}}(c_k^*)(t_{(k)} - c_k)\,,$$

where c_k^* is bounded by c_k and $t_{(k)}$. Then the above equation implies $\left|t_{(k)} - c_k\right| \leq 4^{-1}M^{-1}C^{-1}\varepsilon$ for any k according to the lower bound provided by Assumption 2. Furthermore, $D(t_{(k)}) - D(c_k) = M(\tilde{f}_0^{\hat{T}}(c_k^{**}) - \tilde{f}_1^{\hat{T}}(c_k^{**}))(t_{(k)} - c_k)$ for some c_k^{**} bounded by c_k and $t_{(k)}$. Therefore, Assumption 2 implies $|D(t_{(k)}) - D(c_k)| \leq 4^{-1}\varepsilon$. Suppose $k^* = k'$, then,

$$\alpha_{k',\delta} - D(c_{k'}) \leq \alpha_{k',\delta} - D(t_{(k')}) + 4^{-1}\varepsilon \leq \alpha - 4^{-1}\varepsilon\,,$$

and thus $k^* \geq k_0$. Furthermore, we also have

$$\alpha_{k_0,\delta} - D(t_{(k_0)}) \leq \alpha_{k_0,\delta} - D(c_{k_0}) + 4^{-1}\varepsilon \leq \alpha\,.$$

Recall that $D(t_{(k_0)}) = \tilde{R}_0(\hat{\phi}_{k_0}) - R_0(\hat{\phi}_{k_0})$. Therefore, $R_0(\hat{\phi}_{k_0}) > \alpha$ implies $\tilde{R}_0(\hat{\phi}_{k_0}) > \alpha_{k_0,\delta}$ whose probability is bounded by δ by Proposition 1. □

Proof of Corollary 1. By Lemma 3, $k_{\#}^* \geq k^*$ and thus, $t_{(k^*)} \leq t_{(k_{\#}^*)}$. Therefore, $R_0(\hat{\phi}_{(k^*)}) \geq$

$R_0(\hat{\phi}_{(k^*_{\#})})$. Combined with Theorem 1, the previous result yields

$$\begin{aligned}\mathbb{P}\left(R_0(\hat{\phi}_{(k^*_{\#})}) > \alpha\right) &\leq \mathbb{P}\left(R_0(\hat{\phi}_{(k^*)}) > \alpha\right) \\ &\leq \delta + \delta_1(n_{\mathrm{b}}) + \delta_2(n_{\mathrm{b}}) + 2e^{-8^{-1}nM^{-2}C^{-2}c^2\varepsilon^2} + 2e^{-8^{-1}n_{\mathrm{e}}^0 M^{-2}\varepsilon^2} + 2e^{-8^{-1}n_{\mathrm{e}}^1 M^{-2}\varepsilon^2}\,.\end{aligned}$$

□

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(2), 2780–2824.

Brazdil, P., and Konolige, K. (1990), "Machine Learning, Meta-Reasoning and Logics," *Springer*, .

Brodley, C. E., and Friedl, M. A. (1999*a*), "Identifying mislabeled training data," *Journal of Artificial Intelligence Research*, 11, 131–167.

Brodley, C., and Friedl, M. (1999*b*), "Identifying mislabeled training data," *Journal of Artificial Intelligence Research*, 11, 131–167.

Cannings, T. I., Fan, Y., and Samworth, R. J. (2020), "Classification with imperfect training labels," *Biometrika*, 107(2), 311–330.

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.

Cao, J., Kwong, S., and Wang, R. (2012), "A noise-detection based AdaBoost algorithm for mislabeled data," *Pattern Recognition*, 45(12), 4451–4465.

Ghosh, A., Manwani, N., and Sastry, P. (2015), "Making risk minimization tolerant to label noise," *Neurocomputing*, 160, 93–107.

Guyon, I., Matic, N., Vapnik, V. et al. (1996), "Discovering Informative Patterns and Data Cleaning.,".

Hickey, R. J. (1996), "Noise modelling and evaluating learning from examples," *Artificial Intelligence*, 82(1-2), 157–179.

Hopkins, M., Reeber, E., Forman, G., and Suermondt, J. (1999), "Spambase data set," *Hewlett-Packard Labs*, 1(7).

Khardon, R., and Wachman, G. (2007), "Noise tolerant variants of the perceptron algorithm," *Journal of Machine Learning Research*, 8(Feb), 227–248.

Krizhevsky, A., Hinton, G. et al. (2009), "Learning multiple layers of features from tiny images," , .

Lachenbruch, P. A. (1966), "Discriminant analysis when the initial samples are misclassified," *Technometrics*, 8(4), 657–662.

Lachenbruch, P. A. (1979), "Note on initial misclassification effects on the quadratic discriminant function," *Technometrics*, 21(1), 129–132.

Liu, T., and Tao, D. (2016), "Classification with noisy labels by importance reweighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3), 447–461.

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

Manwani, N., and Sastry, P. (2013), "Noise tolerance under risk minimization," *IEEE Transactions on Cybernetics*, 43(3), 1146–1151.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. (2013), Learning with noisy labels,, in *Advances in Neural Information Processing Systems*, pp. 1196–1204.

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.

Orr, K. (1998), "Data quality and systems theory," *Communications of the ACM*, 41(2), 66–71.

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.

Redman, T. (1998), "The impact of poor data quality on the typical enterprise," *Communications of the ACM*, 2(2), 79–82.

Rigollet, P., and Tong, X. (2011), "Neyman-pearson classification, convexity and stochastic constraints," *Journal of Machine Learning Research*, 12(Oct), 2831–2855.

Scott, C., and Nowak, R. (2005), "A Neyman-Pearson approach to statistical learning," *IEEE Transactions on Information Theory*, 51(11), 3806–3819.

Sukhbaatar, S., and Fergus, R. (2014), "Learning from noisy labels with deep neural networks," *arXiv preprint arXiv:1406.2080*, 2(3), 4.

Tong, X. (2013), "A plug-in approach to Neyman-Pearson classification," *Journal of Machine Learning Research*, 14(1), 3011–3040.

Tong, X., Feng, Y., and Li, J. J. (2018), "Neyman-Pearson classification algorithms and NP receiver operating characteristics," *Science Advances*, 4(2), eaao1659.

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.

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.

Zhao, A., Feng, Y., Wang, L., and Tong, X. (2016), "Neyman-Pearson classification under high-dimensional settings," *Journal of Machine Learning Research*, 17(213), 1–39.