JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2025, VOL. 120, NO. 550, 1164–1177: Theory and Methods
https://doi.org/10.1080/01621459.2024.2402567

Taylor & Francis
Taylor & Francis Group

# Neyman-Pearson Multi-Class Classification via Cost-Sensitive Learning

Ye Tian[a] and Yang Feng[b]

[a]Department of Statistics, Columbia University, New York, NY; [b]Department of Biostatistics, School of Global Public Health, New York University, New York, NY

**ABSTRACT**
Most existing classification methods aim to minimize the overall misclassification error rate. However, in applications such as loan default prediction, different types of errors can have varying consequences. To address this asymmetry issue, two popular paradigms have been developed: the Neyman-Pearson (NP) paradigm and the cost-sensitive (CS) paradigm. Previous studies on the NP paradigm have primarily focused on the binary case, while the multi-class NP problem poses a greater challenge due to its unknown feasibility. In this work, we tackle the multi-class NP problem by establishing a connection with the CS problem via strong duality and propose two algorithms. We extend the concept of NP oracle inequalities, crucial in binary classifications, to NP oracle properties in the multi-class context. Our algorithms satisfy these NP oracle properties under certain conditions. Furthermore, we develop practical algorithms to assess the feasibility and strong duality in multi-class NP problems, which can offer practitioners the landscape of a multi-class NP problem with various target error levels. Simulations and real data studies validate the effectiveness of our algorithms. To our knowledge, this is the first study to address the multi-class NP problem with theoretical guarantees. The proposed algorithms have been implemented in the R package `npcs`, which is available on CRAN. Supplementary materials for this article are available online, including a standardized description of the materials available for reproducing the work.

**ARTICLE HISTORY**
Received October 2023
Accepted September 2024

**KEYWORDS**
Cost-sensitive learning; Confusion matrix; Duality; Feasibility; Multi-class classification; Neyman-Pearson paradigm

## 1. Introduction

### 1.1. Asymmetric Classification Errors and An Example in Loan Default Prediction

Classification is one of the central tasks in machine learning, in which we train a classifier on training data to accurately predict the labels of unseen test data based on predictors. In practice, we rarely achieve a perfect classifier that can correctly classify all the unknown data. There are different types of errors that a classifier can make. In binary classification with classes 1 and 2, denote the predictor vector $X \in \mathcal{X} \subseteq \mathbb{R}^p$ and the label $Y \in \{1,2\}$. For any classifier $\phi : \mathcal{X} \to \{1,2\}$, we usually define Type-I error $R_1 = \mathbb{P}_{X|Y=1}(\phi(X) \neq 1)$ and type-II error $R_2 = \mathbb{P}_{X|Y=2}(\phi(X) \neq 2)$, where $\mathbb{P}_{X|Y=k}$ represents the probability measure induced by the conditional distribution of $X$ given $Y = k$, $k = 1$ or 2. The overall misclassification error can be viewed as a weighted sum of Type-I and Type-II errors.

In many classification approaches, classifiers are designed to minimize the overall misclassification error. However, in many scenarios, different types of errors can have varying degrees of consequences, rendering the overall misclassification error minimization inappropriate. One such example is loan default prediction, where a default borrower is denoted as class 1 and a borrower who pays the full amount on time as class 2. In this context, making a Type-I error, that is, misclassifying a default borrower as a non-default borrower and lending money to them, is typically more serious than making a Type-II error, that is, misclassifying a non-default borrower person as a default one and refusing to lend money to them. In such cases, the criterion of overall misclassification error minimization may need to be revised. Consequently, researchers developed two paradigms—the Neyman-Pearson paradigm and the cost-sensitive learning paradigm—to address this error asymmetry. In the following two sections, we introduce these paradigms separately.

### 1.2. Neyman-Pearson Paradigm

The Neyman-Pearson (NP) paradigm changes the classical classification framework by prioritizing different types of errors differently. In binary classification, the NP paradigm seeks the classifier $\phi$ that solves the following optimization problem

$$\begin{aligned} \min_{\phi} \quad & \mathbb{P}_{X|Y=2}(\phi(X) \neq 2) \\ \text{s.t.} \quad & \mathbb{P}_{X|Y=1}(\phi(X) \neq 1) \leq \alpha_1, \end{aligned} \tag{1}$$

with a given target error level $\alpha_1 \in [0,1]$.

There have been many studies on the binary NP paradigm, and researchers have developed many useful tools to solve problem (1). Cannon et al. (2002) initiated the theoretical analysis of NP classification. Scott and Nowak (2005) proved theoretical properties of the empirical error minimization (ERM) approach, including the so-called NP oracle inequalities. Scott (2007) com-

**CONTACT** Yang Feng yang.feng@nyu.edu Department of Biostatistics, School of Global Public Health, New York University, 708 Broadway, New York, NY 10003.
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JASA.

bined two types of errors to measure the performance under the NP paradigm. Rigollet and Tong (2011) transformed the original problem into a convex problem through some convex surrogates. They solved the new problem and proved that the optimal classifier could successfully control the Type-I error with high probability. Tong (2013) tackled this problem by combining the Neyman-Pearson lemma with the kernel density estimation and developed the so-called plug-in method, which enjoys the NP oracle inequalities. Zhao et al. (2016) extended the NP framework into the high-dimensional case via naïve Bayes classifier, where the number of predictors can grow with the sample size. More recently, Tong, Feng, and Li (2018) proposed an umbrella NP algorithm that can adapt to any scoring-type classifier, including linear discriminant analysis (LDA), support vector machines (SVM), and random forests. Using order statistics and some thresholding strategy, the umbrella algorithm can provide high probability control for all classifiers under some sample size requirements. Tong et al. (2020) further studied both parametric and nonparametric ways to adjust the classification threshold for an LDA classifier, which were proved to solve (1) with NP oracle inequalities. More recently, Wang et al. (2021) introduced an LDA-based NP classifier that does not depend on sample splitting. Scott (2019) proposed a generalized Neyman-Pearson criterion and argued that a broader class of transfer learning problems could be solved under this criterion. Li, Tong, and Li (2020) first connected binary NP problems with CS problems and proposed a way to construct a CS classifier with Type-I error control. Xia et al. (2021) applied the NP umbrella method proposed by Tong, Feng, and Li (2018) into a social media text classification problem. Li, Chen, and Tong (2021) proposed a model-free feature ranking method based on the NP framework. The works we list may be incomplete. We refer interested readers to the survey paper by Tong, Feng, and Zhao (2016) and another recent paper discussing the relationship between hypothesis testing and NP binary classification by Li and Tong (2020).

However, all the works mentioned above primarily focus on the binary NP paradigm. In many real-world scenarios, for example, the loan default prediction problem, there may be more than two possible outcomes, such as default, fully paid, and late payment but not default. Controlling errors under certain target levels in the multi-class scenario is a less explored yet more practically relevant problem. In this article, we consider such a *multi-class* classification problem and propose algorithms to solve it under the NP paradigm. Suppose there are $K$ classes ($K \geq 2$), and we denote them as classes 1 to $K$. The training sample $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ are i.i.d. copies of $(X, Y) \subseteq \mathcal{X} \otimes \{1, \ldots, K\}$, where $\mathcal{X} \subseteq \mathbb{R}^p$. Denote $\pi_k^* = \mathbb{P}(Y = k)$ and we assume $\pi_k^* \in (0, 1)$ for all $k$'s. Also denote $\boldsymbol{\pi}^* = (\pi_1^*, \ldots, \pi_K^*)^T$. To formulate a multi-class NP problem, we need to extend the two types of errors in binary classification to the multi-class case. We now introduce two possible formulations.

- Mossman (1999) and Dreiseitl, Ohno-Machado, and Binder (2000) extended binary receiver operating characteristic (ROC) to multi-class ROC by considering $\mathbb{P}_{X|Y=k}(\phi(X) \neq k)$ as the $k$th error rate of classifier $\phi$ for any $k \in \{1, \ldots, K\}$. Then the NP problem can be constructed to minimize a weighted sum of $\{\mathbb{P}_{X|Y=k}(\phi(X) \neq k)\}_{k=1}^K$ while controlling $\mathbb{P}_{X|Y=k}(\phi(X) \neq k)$ for $k \in \mathcal{A} \subseteq \{1, \ldots, K\}$.
- Another way is to consider the confusion matrix $\Gamma = [\Gamma_{rk}]_{K \times K}$, where $\Gamma_{rk} = \mathbb{P}_{X|Y=k}(\phi(X) = r)$ for $r \neq k$ (Edwards, Metz, and Kupinski 2004). Then we can formulate the NP problem as minimizing a weighted sum of $\{\mathbb{P}_{X|Y=k}(\phi(X) = r)\}_{r,k=1}^K$ while controlling $\Gamma_{rk}$ for $(r, k) \in \mathcal{A} \subseteq [K] \otimes [K]$.

To begin, we focus on the first formulation, which aims to minimize a weighted sum of $\{\mathbb{P}_{X|Y=k}(\phi(X) \neq k)\}_{k=1}^K$ and controls $\mathbb{P}_{X|Y=k}(\phi(X) \neq k)$ for $k \in \mathcal{A}$, where $\mathcal{A} \subseteq \{1, \ldots, K\}$. The more general confusion matrix control problem is more complicated and will be discussed in Section S.2 of the supplementary materials due to space constraints. We formally present the Neyman-Pearson *multi-class* classification (NPMC) problem as

$$\begin{aligned} \min_{\phi} \quad & J(\phi) = \sum_{k=1}^K w_k \mathbb{P}_{X|Y=k}(\phi(X) \neq k) \\ \text{s.t.} \quad & \mathbb{P}_{X|Y=k}(\phi(X) \neq k) \leq \alpha_k, \quad k \in \mathcal{A}, \end{aligned} \tag{2}$$

where $\phi : \mathcal{X} \to \{1, \ldots, K\}$ is a (measurable) classifier, $\alpha_k \in [0, 1]$, $w_k \geq 0$, $\sum_{k=1}^K w_k = 1$ [1], and $\mathcal{A} \subseteq \{1, \ldots, K\}$. The linear combination format of the objective function $J(\phi)$ is chosen for ease of interpretation. Here, $w_k$ represents the "cost" of misclassifying an observation from class $k$. If we set $w_k = \pi_k^*$ for all $k$, then $J(\phi)$ equals the overall misclassification error rate $\mathbb{P}(\phi(X) \neq Y)$. Furthermore, our analysis and proposed algorithms can be extended to the case of $J(\phi) = \max_{k \notin \mathcal{A}} \mathbb{P}_{X|Y=k}(\phi(X) \neq k)$, which represents the worst performance among classes not in $\mathcal{A}$[2], and the details can be found in Section S.3 in the supplements.

The formulation of (2) is closely connected to the distributional hypothesis testing problem with a composite null hypothesis consisting of finite arguments. For example, suppose that we have collected data $X_n = (x_1, \ldots, x_n)^T \sim$ some distribution $\mathbb{P}$ and we would like to test $H_0 : \mathbb{P} \in \{\mathbb{P}^{(k)}\}_{k=1}^K$ v.s. $H_1 : \mathbb{P} = \mathbb{P}^{(K+1)}$. The goal is to find the optimal deterministic testing function $\varphi : X_n \mapsto \{0, 1\}$ that maximizes the statistical power $\mathbb{P}^{(K+1)}(\varphi(X_n) = 1)$ and controls the Type-I error rate under level $\alpha$, that is, $\max_{k=1:K} \mathbb{P}^{(k)}(\varphi(X_n) = 1) \leq \alpha$. These two problems are interconnected, and both necessitate control over multiple errors. However, there are some intrinsic differences between these two problems. First, in the hypothesis testing problem, $\mathbb{P}^{(k)}$ is *known*, whereas in the NP problem (2), the distribution of $X$ given $Y = k$ is *unknown*. Second, multiple $\mathbb{P}^{(k)}$'s belong to the same null hypothesis $H_0$, inherently constituting a *binary* problem. Consequently, the hypothesis testing problem is always feasible. However, in the NP problem (2), $K$ classes are distinct and are associated with potentially different target control levels $\alpha_k$'s, rendering it a *multi-class* problem where feasibility is not guaranteed (as elaborated later). More comparisons between the hypothesis testing and NP problems

[1] This is without loss of generality as we can always normalize the weights $\{w_k\}_{k=1}^K$ by $\sum_{k=1}^K w_k$.

[2] We thank one of the reviewers for pointing this out.

can be found in Li and Tong (2020). Additional discussions will be provided in Section S.1.4 of supplementary materials.

Previously, there have been few works on solving the NPMC problem. Landgrebe and Duin (2005) proposed a general empirical method to solve the NPMC problem, which relies on the multi-class ROC estimation. Our work tackles the NPMC problem by linking it with the cost-sensitive learning problem (to be introduced), which is partly motivated by their approach. However, there are notable differences between our work and theirs. First, their algorithm requires a grid search to determine the appropriate cost parameters. When dealing with a large number of classes $K$ and demanding high accuracy, the computation cost will be too high to be affordable. Despite the efficient multi-class ROC approximation via decomposition and sensitivity analysis proposed in a later work (Landgrebe and Duin 2008), it remains somewhat restrictive without a formal connection to a cost-sensitive learning problem. Our algorithms connect the NPMC problem to cost-sensitive learning by duality and search the optimal costs in cost-sensitive learning by a direct optimization procedure, which is much more straightforward than their method. Second, their approach lacks theoretical guarantees, whereas we prove the multi-class NP oracle properties for our methods under certain conditions. More recently, Ma, Lin, and Yang (2020) developed a regularized sub-gradient method on non-convex optimization problems, which can be applied to solve the NPMC problem with specific linear classifiers with non-convex losses. Their method is only suitable for linear classifiers with certain loss functions, while our methods are adaptable to any classification method. To our knowledge, our work is the first to solve the NPMC problem via cost-sensitive learning techniques with theoretical guarantees.

Compared to the binary NP problem (1), the multi-class version (2) is significantly more challenging to solve. One of the major challenges lies in the fact that the binary NP problem (1) is always feasible (in the most extreme case, all observations can be classified to the class whose error rate is to be controlled) while the problem (2) can be infeasible. To provide readers with insight into how feasibility interacts with target error levels and the conditional distribution of $X$ given $Y$, let's consider a simple example: a 3-class NPMC problem with $X|Y=k \sim N(\boldsymbol{\mu}_k, \boldsymbol{I}_p)$ for $k=1,2,3$, $\mathcal{A}=\{1,2\}$, and the target levels $\alpha_1, \alpha_2$. Even in this basic setup, characterizing the feasibility condition remains challenging because problem (2) encompasses all deterministic classifiers. However, thanks to our Theorem 1 (to be introduced in Section 3.1), we can derive the following lemma, which explicitly provides the feasibility condition.

*Lemma 1.* The 3-class NPMC problem (2) with $X|Y=k \sim N(\boldsymbol{\mu}_k, \boldsymbol{I}_p)$ for $k=1,2,3$, $\mathcal{A}=\{1,2\}$, and the target levels $\alpha_1$, $\alpha_2 \in [0,1]$, is feasible if and only if

$$\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \geq \Phi^{-1}(1-\alpha_1) + \Phi^{-1}(1-\alpha_2),$$

where $\Phi^{-1}$ is the inverse CDF function of $N(0,1)$.

We observe a tradeoff between $\alpha_1$ and $\alpha_2$ given $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, indicating that we cannot make both arbitrarily small. In general, it is difficult to characterize the feasibility condition on the joint distribution of $(X, Y)$ for the NPMC problem (2).

### 1.3. Cost-Sensitive Learning

As discussed in Section 1.1, cost-sensitive learning (CS) provides another approach to addressing asymmetric errors in classification. There are two types of cost-sensitive learning problems where the cost is associated with features or classes, respectively (Fernández et al. 2018). Here, we focus on the second type, where the cost is associated with different classes. Ling and Sheng (2008) further divided methods dealing with this type of CS problem into two categories: direct and meta-learning methods. Direct methods design the algorithm structure for specific classifiers, for example, support vector machines (Katsumata and Takeda 2015), $k$-nearest neighbors (Qin et al. 2013), and neural networks (Zhou and Liu 2005). Meta-learning methods create a wrapper that converts an existing classifier into a cost-sensitive one. Instances of this type of approach include rescaling (Domingos 1999; Zhou and Liu 2010), thresholding (Elkan 2001; Sheng and Ling 2006; Tian and Zhang 2019), and weighted-likelihood methods (Dmochowski, Sajda, and Parra 2010), among others.

Similar to the multi-class NP problem, there are also two ways to formulate the multi-class CS problem. One is to consider per-class error rates $\mathbb{P}_{X|Y=k}(\phi(X) \neq k|Y=k)$ for $k=1,\ldots,K$, and the other one is to consider the confusion matrix. In this article, we would like to connect (2) to the following cost-sensitive (CS) multi-class classification problem

$$\min_{\phi} \quad \mathrm{Cost}(\phi) = \sum_{k=1}^{K} \pi_k^* c_k \mathbb{P}_{X|Y=k}(\phi(X) \neq k), \tag{3}$$

where $\phi : \mathcal{X} \to \{1,\ldots,K\}$, $\pi_k^* = \mathbb{P}(Y=k)$, and $\{c_k\}_{k=1}^K$ are the costs associated with each class. The relationship between the NPMC problem with the confusion matrix control and the CS problem will be discussed in Section S.2 of supplementary materials.

The following lemma shows that CS problem (3) has an explicit solution.

*Lemma 2.* Define classifier $\bar{\phi}^* : \boldsymbol{x} \mapsto \arg\max_k\{c_k \mathbb{P}_{Y|X=\boldsymbol{x}}(Y=k)\}$ [3]. Then $\bar{\phi}^*$ is an optimal classifier of (3) in the following sense: For any classifier $\phi$, $\mathrm{Cost}(\bar{\phi}^*) \leq \mathrm{Cost}(\phi)$.

### 1.4. Multi-Class NP Oracle Properties

In this section, we extend the NP oracle inequalities proposed in Scott and Nowak (2005) to the multi-class case for problem (2). We call them the multi-class NP oracle properties. Algorithms satisfying these two properties satisfied are desirable. For any classifier $\phi$, we denote $R_k(\phi) = \mathbb{P}_{X|Y=k}(\phi(X) \neq k)$.

**Multi-class NP oracle properties for the NPMC problem**:

(i) If the NPMC problem is feasible and has an optimal solution $\phi^*$, then the algorithm outputs a solution $\hat{\phi}$ that satisfies

(a) $R_k(\hat{\phi}) \leq \alpha_k + \mathcal{O}_{\mathbb{P}}(\epsilon(n)), \forall k \in \mathcal{A}$;
(b) $J(\hat{\phi}) \leq J(\phi^*) + \mathcal{O}_{\mathbb{P}}(\epsilon_J(n))$,

where $\epsilon(n)$ and $\epsilon_J(n) \to 0$ as $n \to \infty$.

[3] If there is a tie, let $\bar{\phi}^*(\boldsymbol{x})$ be the smallest index within the tie.

(ii) Denote the event that the algorithm indicates infeasibility of NPMC problem given $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ as $\mathcal{G}_n$. If the NPMC problem is infeasible, then $\mathbb{P}(\mathcal{G}_n) \to 1$, as $n \to \infty$.

It is important to remark that multi-class NP oracle properties can only guarantee an "approximate" control for problem (2), in the sense that the actual error rate could fluctuate around the target level, and the scale of fluctuation vanishes with high probability as the sample size $n \to \infty$. This form is motivated from the NP oracle inequalities in the binary case used in literature (e.g., Cannon et al. 2002; Scott and Nowak 2005; Scott 2019; Kalan and Kpotufe 2023, 2024). Therefore, our goal is to obtain a classifier $\phi$ which can control $\mathbb{P}_{X|Y=k}(\phi(X) \neq k)$ around $\alpha_k$ with high probability for all $k \in \mathcal{A}$.

### 1.5. Organization

We organize the rest of this article as follows. In Section 2, we develop two algorithms to solve the NPMC problem (2), denoted as NPMC-CX (ConveX) and NPMC-ER (Empirical Risk), respectively. In Section 3, we show that NPMC-CX enjoys multi-class NP oracle properties under Rademacher classes, and NPMC-ER satisfies multi-class NP oracle properties under a broader class of models, as long as the model can fit the data well enough. We validate the effectiveness of our approaches via simulations and real data experiments in Section 4. Section 5 summarizes our contributions and points out a few potential future research directions. Due to the page limit, some additional discussions, extra numerical results, and all the proofs are provided in the supplementary materials.

### 1.6. Notations

Before closing the introduction, we summarize the notations used throughout this article. For any set $D$, $|D|$ represents its cardinality. For any real number $a$, $\lfloor a \rfloor$ denotes the maximum integer no larger than $a$. Define the nonnegative half space in $\mathbb{R}^p$ as $\mathbb{R}^p_+ = \{\boldsymbol{x} = (x_1, \dots, x_p)^T \in (\mathbb{R} \cup \{+\infty\})^p : \min_j x_j \geq 0\}$. For a $p$-dimensional vector $\boldsymbol{x} = (x_1, \dots, x_p)^T$, its $\ell_2$-norm is defined as $\|\boldsymbol{x}\|_2 = \sqrt{\sum_{j=1}^p x_j^2}$. For a $p \times p$ matrix $A$, $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ represent its maximum and minimum eigenvalues, respectively. We mean $A$ is positive-definite or negative-definite by writing $A \succ 0$ or $A \prec 0$, respectively. For a function $f : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X}$ is some metric space, we define its sup-norm as $\|f\|_\infty = \sup_{\boldsymbol{x} \in \mathcal{X}} |f(\boldsymbol{x})|$. For the empty set $\emptyset$, we define $\min_{\boldsymbol{x} \in \emptyset} f(\boldsymbol{x}) = +\infty$. For two nonzero real sequences $\{a_n\}_{n=1}^\infty$ and $\{b_n\}_{n=1}^\infty$, we denote $\sup_n |a_n/b_n| < \infty$ by $a_n \lesssim b_n$. For two random sequences $\{a_n\}_{n=1}^\infty$ and $\{b_n\}_{n=1}^\infty$, $a_n = \mathcal{O}_{\mathbb{P}}(b_n)$ indicates that for any $\epsilon > 0$, there exists a positive constant $M$ such that $\sup_n \mathbb{P}(|a_n/b_n| > M) \leq \epsilon$. We use $\mathbb{P}$ and $\mathbb{E}$ to represent probabilities and expectations. Sometimes we add subscripts to emphasize the source of randomness. For example, $\mathbb{P}_{Y|X=\boldsymbol{x}}(Y = k)$ means the probability of $Y = k$ given $X = \boldsymbol{x}$. $\mathbb{E}_X$ means the expectation is taken w.r.t. the distribution of $X$. If there is no subscript, we mean the probability and expectation are calculated w.r.t. all randomness.

## 2. Methodology

### 2.1. The First Algorithm: NPMC-CX

Before formally introducing our first algorithm, we would like to derive it through heuristic calculations. For problem (2), consider its Lagrangian function as

$$
\begin{aligned}
L(\boldsymbol{\lambda}, \phi) &= \sum_{k \notin \mathcal{A}} w_k \mathbb{P}_{X|Y=k}(\phi(X) \neq k) \\
&\quad + \sum_{k \in \mathcal{A}} (w_k + \lambda_k) \mathbb{P}_{X|Y=k}(\phi(X) \neq k) - \sum_{k \in \mathcal{A}} \lambda_k \alpha_k \\
&= -\sum_{k \notin \mathcal{A}} w_k \mathbb{P}_{X|Y=k}(\phi(X) = k) \\
&\quad - \sum_{k \in \mathcal{A}} (w_k + \lambda_k) \mathbb{P}_{X|Y=k}(\phi(X) = k) + \sum_{k=1}^K w_k \\
&\quad + \sum_{k \in \mathcal{A}} \lambda_k (1 - \alpha_k),
\end{aligned}
\tag{4}
$$

where $\boldsymbol{\lambda} = \{\lambda_k\}_{k \in \mathcal{A}}$. Then, the dual problem of (2) can be written as

$$
\max_{\boldsymbol{\lambda} \in \mathbb{R}_+^{|\mathcal{A}|}} \min_{\phi} L(\boldsymbol{\lambda}, \phi). \tag{5}
$$

We can see that (5) looks for a lower bound of the objective function in (2), that is, $\sup_{\boldsymbol{\lambda} \in \mathbb{R}_+^{|\mathcal{A}|}} \min_\phi L(\boldsymbol{\lambda}, \phi) \leq \inf_{\phi \in \mathfrak{C}} \sum_{k=1}^K w_k \mathbb{P}_{X|Y=k}(\phi(X) \neq k)$, where $\mathfrak{C}$ includes all feasible classifiers for problem (2). We often call this fact as *weak duality*. In many cases, the exact equality holds, which is called *strong duality*. Under strong duality, (2) and (5) can be seen as two different approaches to address the same problem. If one has an optimal solution, the other one will have an optimal solution as well. If the original NPMC problem (2) is infeasible, then (5) must be unbounded from above, and vice versa. Another key observation is that, for a given $\boldsymbol{\lambda} \in \mathbb{R}_+^{|\mathcal{A}|}$, looking for $\phi$ that minimizes $L(\boldsymbol{\lambda}, \phi)$ in (4) effectively translates into a CS problem (3) with costs

$$
c_k(\boldsymbol{\lambda}, \boldsymbol{\pi}^*) = \begin{cases} w_k/\pi_k^*, & k \notin \mathcal{A}; \\ (w_k + \lambda_k)/\pi_k^*, & k \in \mathcal{A}. \end{cases}
$$

This observation motivates our first algorithm, where we endeavor to solve the more tractable CS problem (5) to address the more challenging original problem (2).

To derive our first algorithm, let's rewrite (4) as

$$
\begin{aligned}
L(\boldsymbol{\lambda}, \phi) = &-\mathbb{E}_X \left[ c_{\phi(X)}(\boldsymbol{\lambda}, \boldsymbol{\pi}^*) \cdot \mathbb{P}_{Y|X}(Y = \phi(X)) \right] \\
&+ \sum_{k=1}^K w_k + \sum_{k \in \mathcal{A}} \lambda_k (1 - \alpha_k).
\end{aligned}
$$

Then by Lemma 2, we can define

$$
\begin{aligned}
\phi_{\boldsymbol{\lambda}}^* : \boldsymbol{x} \mapsto &\operatorname*{arg\,max}_k \{c_k(\boldsymbol{\lambda}, \boldsymbol{\pi}^*) \mathbb{P}_{Y|X=\boldsymbol{x}}(Y = k)\} \\
&\in \operatorname*{arg\,min}_\phi L(\boldsymbol{\lambda}, \phi),
\end{aligned}
\tag{7}
$$

$$
G(\boldsymbol{\lambda}) = \min_\phi L(\boldsymbol{\lambda}, \phi) = L(\boldsymbol{\lambda}, \phi_{\boldsymbol{\lambda}}^*). \tag{8}
$$

Therefore, on the population level, we can find $\boldsymbol{\lambda}$ which maximizes $G(\boldsymbol{\lambda})$, then plug $\boldsymbol{\lambda}$ into (7) to obtain the final classifier. On the other hand, due to weak duality, since the objective function in (2) is no larger than 1 when it's feasible, we must have $\sup_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} G(\boldsymbol{\lambda}) \leq 1$. Thus, if $\sup_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} G(\boldsymbol{\lambda}) > 1$, the original NP problem (2) must be infeasible.

In practice, there is no access to $L(\boldsymbol{\lambda},\phi)$ and $G(\boldsymbol{\lambda})$ since we do not know the true model. We estimate $L(\boldsymbol{\lambda},\phi)$ by training data as

$$\widehat{L}^{\text{CX}}(\boldsymbol{\lambda},\phi) = -\frac{1}{n}\sum_{i=1}^{n} c_{\phi(\boldsymbol{x}_i)}(\boldsymbol{\lambda},\hat{\boldsymbol{\pi}})\widehat{\mathbb{P}}_{Y|X=\boldsymbol{x}_i}(Y=\phi(\boldsymbol{x}_i)) + \sum_{k=1}^{K} w_k + \sum_{k\in\mathcal{A}}\lambda_k(1-\alpha_k),$$

where

$$c_k(\boldsymbol{\lambda},\hat{\boldsymbol{\pi}}) = \begin{cases} w_k/\hat{\pi}_k, & k\notin\mathcal{A}; \\ (w_k+\lambda_k)/\hat{\pi}_k, & k\in\mathcal{A}, \end{cases}$$

$\hat{\pi}_k = n_k/n$ with $n_k = \#\{i: y_i = k\}$, $\hat{\boldsymbol{\pi}} = \{\hat{\pi}_k\}_{k=1}^K$, and $\widehat{\mathbb{P}}_{Y|X}$ is the estimated conditional probability. $\widehat{\mathbb{P}}_{Y|X}$ can be obtained from any function class by fitting the data, and we do not impose any conditions on it here. Here are two examples.

- For a parametric example, we may use the data to fit a multinomial logistic regression model and obtain the estimates of $(K-1)$ contrast coefficients $\{\hat{\boldsymbol{\beta}}^{(k)}\}_{k=1}^{K-1}$ with $\hat{\boldsymbol{\beta}}^{(k)} \in \mathbb{R}^p$. Then $\widehat{\mathbb{P}}_{Y|X=\boldsymbol{x}}(Y=k) = \frac{\exp\{\boldsymbol{x}^T\hat{\boldsymbol{\beta}}^{(k)}\}}{\sum_{k=1}^K \exp\{\boldsymbol{x}^T\hat{\boldsymbol{\beta}}^{(k)}\}}$ where $\hat{\boldsymbol{\beta}}^{(K)} = \boldsymbol{0}_p$.
- For a nonparametric example, we may use the $k$-nearest neighbors ($k$NN) to obtain the estimate $\widehat{\mathbb{P}}_{Y|X=\boldsymbol{x}}$. Given such an $\boldsymbol{x}$ and the number of the nearest neighbors $k_0$, we can use the proportion of training observations of class $k$ among $k_0$ nearest neighbors to $\boldsymbol{x}$ as an estimate $\widehat{\mathbb{P}}_{Y|X=\boldsymbol{x}}(Y=k)$.

Similar to Lemma 2, it is easy to show that one of the optimal classifiers that minimize $\widehat{L}^{\text{CX}}(\boldsymbol{\lambda},\phi)$ for a given $\boldsymbol{\lambda}$ is

$$\hat{\phi}_{\boldsymbol{\lambda}}: \boldsymbol{x} \mapsto \arg\max_k\{c_k(\boldsymbol{\lambda},\hat{\boldsymbol{\pi}})\widehat{\mathbb{P}}_{Y|X=\boldsymbol{x}}(Y=k)\} \in \arg\min_\phi \widehat{L}^{\text{CX}}(\boldsymbol{\lambda},\phi). \tag{10}$$

Denote

$$\widehat{G}^{\text{CX}}(\boldsymbol{\lambda}) := \widehat{G}^{\text{CX}}(\boldsymbol{\lambda};\widehat{\mathbb{P}}_{Y|X},\hat{\boldsymbol{\pi}}) = \min_\phi \widehat{L}^{\text{CX}}(\boldsymbol{\lambda},\phi) = \widehat{L}^{\text{CX}}(\boldsymbol{\lambda},\hat{\phi}_{\boldsymbol{\lambda}}),$$

which is a well-defined function of $\boldsymbol{\lambda}$ given $\widehat{\mathbb{P}}_{Y|X}$ and $\hat{\boldsymbol{\pi}}$. Similar to (5), we solve

$$\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}}\min_\phi \widehat{L}^{\text{CX}}(\boldsymbol{\lambda},\phi) = \max_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} \widehat{L}^{\text{CX}}(\boldsymbol{\lambda},\hat{\phi}_{\boldsymbol{\lambda}}) = \max_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} \widehat{G}^{\text{CX}}(\boldsymbol{\lambda})$$

to find solution $\hat{\boldsymbol{\lambda}}$, then plug it in (10) to obtain the final solution $\hat{\phi}_{\hat{\boldsymbol{\lambda}}}$ to the original NPMC problem (2). On the other hand, considering the estimation error, if $\sup_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} \widehat{G}^{\text{CX}}(\boldsymbol{\lambda}) > 1+\delta$ with a small positive constant $\delta$, we declare that the NPMC problem (2) is infeasible.

```
Algorithm 1: NPMC-CX
Input: training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, target upper bounds of
       errors $\boldsymbol{\alpha}$, the weighting vector of objective
       function $\boldsymbol{w}$, a function class $\mathcal{M}$ to estimate $\mathbb{P}_{Y|X}$,
       a small constant $\delta > 0$
Output: the fitted classifier $\hat{\phi}$ or report the NP problem
        as infeasible
$\widehat{\mathbb{P}}_{Y|X}, \hat{\boldsymbol{\pi}} \leftarrow$ the estimates of $\mathbb{P}_{Y|X}$ (chosen from $\mathcal{M}$) and
    $\boldsymbol{\pi}^*$ on training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$
if $\sup_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} \widehat{G}^{\text{CX}}(\boldsymbol{\lambda};\widehat{\mathbb{P}}_{Y|X},\hat{\boldsymbol{\pi}}) \leq 1+\delta$ then
    $\hat{\boldsymbol{\lambda}} \in \arg\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} \widehat{G}^{\text{CX}}(\boldsymbol{\lambda};\widehat{\mathbb{P}}_{Y|X},\hat{\boldsymbol{\pi}})$        (13)
    Report the NP problem as feasible and output the
     solution $\hat{\phi}(\boldsymbol{x}) = \arg\max_k\{c_k(\hat{\boldsymbol{\lambda}},\hat{\boldsymbol{\pi}})\widehat{\mathbb{P}}_{Y|X=\boldsymbol{x}}(Y=k)\}$
else
    Report the NP problem as infeasible
end
```

Note that $\widehat{G}^{\text{CX}}(\boldsymbol{\lambda})$ is a concave function (as we will show in Proposition 1), which implies that the optimization problem (13) is convex. Therefore, we refer to the algorithm above as NPMC-CX, summarized in Algorithm 1. It is worth noting that $\widehat{G}^{\text{CX}}(\boldsymbol{\lambda})$ is also a piecewise linear function on $\mathbb{R}_+^{|\mathcal{A}|}$. In practice, despite the concavity of $\widehat{G}^{\text{CX}}(\boldsymbol{\lambda})$, the common convex optimization methods are difficult to apply due to the difficulty in calculating the gradient of $\widehat{G}^{\text{CX}}(\boldsymbol{\lambda})$ w.r.t. $\boldsymbol{\lambda}$. Instead, we implement the optimization step via direct search methods like the Hooke-Jeeves method (Hooke and Jeeves 1961) and Nelder-Mead method (Nelder and Mead 1965). More implementation details will be described in Section 4 and Section S.4 of the supplementary materials.

### 2.2. The Second Algorithm: NPMC-ER

In Section 2.1, we introduced an estimator (9) for the Lagrangian function (4). In the literature on NP classification, a more popular estimator is constructed using empirical error rates on a separate dataset (Landgrebe and Duin 2005; Tong 2013). In this section, we develop a new algorithm, NPMC-ER, based on a different estimator for (4) using empirical error rates. We will compare NPMC-CX and NPMC-ER both theoretically (Section 3) and empirically (Section 4). Some take-away messages will be summarized in Section 5.

For convenience, throughout this section, we assume the training sample size to be $2n$. Consider the following procedure. First, we divide the training data randomly into two parts of size $n$: $\mathcal{D}_1 = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ and $\mathcal{D}_2 = \{(\boldsymbol{x}_i, y_i)\}_{i=n+1}^{2n}$. $\mathcal{D}_1$ will be used to calculate the value of $\widehat{L}^{\text{ER}}(\boldsymbol{\lambda},\phi)$ (to be defined), and $\mathcal{D}_2$ will be used to estimate $\widehat{\mathbb{P}}_{Y|X}$ and $\hat{\boldsymbol{\pi}}$. We estimate (4) on $\mathcal{D}_1 = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n = \{\{(\boldsymbol{x}_i^{(k)}, y_i^{(k)})\}_{i=1}^{n_k}\}_{k=1}^K$ by

$$\begin{aligned}\widehat{L}^{\mathrm{ER}}(\boldsymbol{\lambda},\phi) = &-\sum_{k\notin\mathcal{A}} w_k \cdot \frac{1}{n_k}\sum_{i=1}^{n_k}\mathbb{1}(\phi(\boldsymbol{x}_i^{(k)})=k) \\ &-\sum_{k\in\mathcal{A}}(w_k+\lambda_k)\cdot\frac{1}{n_k}\sum_{i=1}^{n_k}\mathbb{1}(\phi(\boldsymbol{x}_i^{(k)})=k) \\ &+\sum_{k=1}^{K} w_k + \sum_{k\in\mathcal{A}}\lambda_k(1-\alpha_k),\end{aligned}$$

where $\{(\boldsymbol{x}_i^{(k)}, y_i^{(k)})\}_{i=1}^{n_k}$ are the observations from class $k$ in $\mathcal{D}_1$. Then, similar to (12), we solve

$$\hat{\boldsymbol{\lambda}} \in \arg\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} \widehat{L}^{\mathrm{ER}}(\boldsymbol{\lambda},\hat{\phi}_{\boldsymbol{\lambda}}),[4]$$

where $\hat{\phi}_{\boldsymbol{\lambda}}$ is defined as in (10) while $\widehat{\mathbb{P}}_{Y|X}$ and $\hat{\boldsymbol{\pi}}$ are calculated by data in $\mathcal{D}_2$. Define

$$\widehat{G}^{\mathrm{ER}}(\boldsymbol{\lambda}) := \widehat{G}^{\mathrm{ER}}(\boldsymbol{\lambda};\widehat{\mathbb{P}}_{Y|X},\hat{\boldsymbol{\pi}}) = \widehat{L}^{\mathrm{ER}}(\boldsymbol{\lambda},\hat{\phi}_{\boldsymbol{\lambda}}). \tag{16}$$

Note that in NPMC-CX, given any $\boldsymbol{\lambda}$, $\hat{\phi}_{\boldsymbol{\lambda}}$ is a minimizer of $\widehat{L}^{\mathrm{CX}}(\boldsymbol{\lambda},\phi)$ w.r.t. any classifier $\phi$. In contrast, for NPMC-ER, given $\boldsymbol{\lambda}$, we still define $\hat{\phi}_{\boldsymbol{\lambda}}$ as in (10), which is not necessarily a minimizer of $\widehat{L}^{\mathrm{ER}}(\boldsymbol{\lambda},\phi)$, and $\widehat{G}^{\mathrm{ER}}(\boldsymbol{\lambda})$ is not necessarily equal to $\sup_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}}\min_\phi \widehat{L}^{\mathrm{ER}}(\boldsymbol{\lambda},\phi)$. The remaining steps are the same as NPMC-CX.

The reason we do not define $\hat{\phi}_{\boldsymbol{\lambda}}$ as $\arg\min_\phi \widehat{L}^{\mathrm{ER}}(\boldsymbol{\lambda},\phi)$ is that there might be many (even infinitely many) minimizers, leading to instability in the estimated model. This issue often arises when fitting models via minimizing the training error. For instance, rescaling all coefficient components in logistic regression does not change the classification results and error rates.

**Algorithm 2:** NPMC-ER

```
Input: training data ${(\boldsymbol{x}_i, y_i)}_{i=1}^{2n}$, target upper bound of
       errors $\boldsymbol{\alpha}$, the weighting vector of objective
       function $\boldsymbol{w}$, a search range $R > 0$, a function class
       $\mathcal{M}$ to estimate $\mathbb{P}_{Y|X}$, a small constant $\delta > 0$
Output: the fitted classifier $\hat{\phi}$ or report the NP problem
        as infeasible
Randomly divide the whole training data (and reindex
  them) into $\mathcal{D}_1 \bigcup \mathcal{D}_2 = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n} \bigcup \{(\boldsymbol{x}_i, y_i)\}_{i=n+1}^{2n}$
$\widehat{\mathbb{P}}_{Y|X}, \hat{\boldsymbol{\pi}}$ ← the estimates of $\mathbb{P}_{Y|X}$ (chosen from $\mathcal{M}$) and
  $\boldsymbol{\pi}^*$ on $\mathcal{D}_2 = (\boldsymbol{x}_i, y_i)_{i=n+1}^{2n}$
$\hat{\boldsymbol{\lambda}}$ ← $\arg\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}, \|\boldsymbol{\lambda}\|_\infty \le R} \widehat{G}^{\mathrm{ER}}(\boldsymbol{\lambda}; \widehat{\mathbb{P}}_{Y|X}, \hat{\boldsymbol{\pi}})$, where $\widehat{G}^{\mathrm{ER}}$
  is estimated on $\mathcal{D}_1 = (\boldsymbol{x}_i, y_i)_{i=1}^{n}$          (17)
if $\widehat{G}^{\mathrm{ER}}(\hat{\boldsymbol{\lambda}}) \le 1 + \delta$ then
    Report the NP problem as feasible and output the
      solution $\hat{\phi}(\boldsymbol{x}) = \arg\max_k \{c_k(\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\pi}}) \widehat{\mathbb{P}}_{Y|X=\boldsymbol{x}}(Y = k)\}$
else
    Report the NP problem as infeasible
end
```

We name the second algorithm NPMC-ER because it uses the empirical error to estimate the true error rate, and we summarize it as Algorithm 2. Similar to $\widehat{G}^{\mathrm{CX}}(\boldsymbol{\lambda})$ defined in (11), $\widehat{G}^{\mathrm{ER}}(\boldsymbol{\lambda})$ in (16) is also a piecewise linear function of $\boldsymbol{\lambda}$. However, it is not necessarily concave. Similar to NPMC-CX, we use the direct search method to conduct the optimization step (17) in practice. Note that since $\widehat{G}^{\mathrm{ER}}(\boldsymbol{\lambda})$ is not necessarily concave, for technical reasons, we need to restrict the search range of the best $\boldsymbol{\lambda}$ to a bounded region. Hence, compared to NPMC-CX (Algorithm 1), there is an additional argument representing the search range $R$ in NPMC-ER (Algorithm 2). The condition on $R$ in the theoretical analysis will be described in the next section. The empirical results are not very sensitive to the choice of $R$, and we pick $R = 1000$ in all numerical studies.

## 3. Theory

In this section, we delve into the theoretical properties of the two algorithms introduced in Section 2. We begin with Section 3.1, where we establish sufficient and necessary conditions for strong duality, shedding light on the circumstances under which it holds. Sections 3.2 and 3.3 are dedicated to presenting the theoretical foundations of NPMC-CX and NPMC-ER, respectively. In Section 3.4, we undertake a theoretical comparison of the two algorithms, unearthing additional insights that encompass discussions on the assumptions and other essential properties of NP algorithms. The additional details omitted in this section can be found in Section S.1 of the supplementary materials.

### *3.1. Checking Strong Duality and Feasibility*

As described in the heuristic arguments in Section 2.1, strong duality between the original NPMC problem (2) and the dual problem (5) is vital for our algorithms to work well. Therefore, we formalize the requirement of strong duality through the following assumption.

*Assumption 1 (Strong duality for the NPMC problem).* It holds that

$$\inf_{\phi\in\mathfrak{C}} J(\phi) = \sup_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} G(\boldsymbol{\lambda}),$$

where $\mathfrak{C}$ includes all feasible classifiers for the NPMC problem (2). If $\mathfrak{C} \neq \emptyset$, the infimum over $\phi \in \mathfrak{C}$ is achievable, and the supremum over $\boldsymbol{\lambda} \in \mathbb{R}_+^{|\mathcal{A}|}$ can be attained at a finite $\boldsymbol{\lambda}$.

There are various sufficient conditions for strong duality in literature, for example, Slater's condition (Luenberger 1997; Boyd and Vandenberghe 2004). However, most of them are applicable only to convex problems, while the original NPMC problem (2) is not necessarily convex. The following theorem elucidates a tight relationship between the feasibility of the induced classifier from the dual CS problem (5) and the strong duality in the NPMC problem (2).

*Theorem 1 (Sufficient and necessary conditions for NPMC strong duality).* Suppose $\{X|Y = k\}_{k=1}^K$ are continuous random variables (i.e. have Lebesgue densities).

(i) When the NPMC problem (2) is feasible, the strong duality holds if and only if there exists $\boldsymbol{\lambda}^{(0)} = \{\lambda_k^{(0)}\}_{k\in\mathcal{A}}$ such

[4] This $\hat{\lambda}$ is different from the $\hat{\lambda}$ estimated in NPMC-CX. We ignore the superscript for simplicity.

that $\phi^*_{\boldsymbol{\lambda}^{(0)}}$ is feasible for the NPMC problem (2), that is, $\mathbb{P}_{X|Y=k}(\phi^*_{\boldsymbol{\lambda}^{(0)}}(X) \neq k) \leq \alpha_k$ for all $k \in \mathcal{A}$.

(ii) Suppose $\mathbb{P}_{Y|X=\boldsymbol{x}}(Y = k) \geq a > 0$ for a.s. $\boldsymbol{x}$ (w.r.t. the distribution of $X$) and all $k \in \mathcal{A}$. When the NPMC problem (2) is infeasible, the strong duality holds (i.e., $\sup_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} G(\boldsymbol{\lambda})$ is unbounded from above) if and only if for an arbitrary $\boldsymbol{\lambda} \in \mathbb{R}_+^{|\mathcal{A}|}$, $\phi^*_{\boldsymbol{\lambda}}$ is infeasible for NPMC problem (2), that is, $\exists$ at least one $k \in \mathcal{A}$ such that $\mathbb{P}_{X|Y=k}(\phi^*_{\boldsymbol{\lambda}}(X) \neq k) > \alpha_k$.

Building upon Theorem 1, we derive the following corollary, which proves to be very useful in practical assessments of feasibility and strong duality.

*Corollary 1.* Suppose $\{X|Y = k\}_{k=1}^K$ are continuous random variables (i.e. have Lebesgue densities). The following equivalences hold:

(i) The NPMC problem is feasible, and strong duality holds $\Leftrightarrow$ $\exists$ a finite $\boldsymbol{\lambda}^* \in \arg\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} G(\boldsymbol{\lambda})$ and $\phi^*_{\boldsymbol{\lambda}^*}$ is feasible;

(ii) The NPMC problem is infeasible, and strong duality holds $\Leftrightarrow$ $\exists$ an infinite $\boldsymbol{\lambda}^*$ and $G(\boldsymbol{\lambda}^*) = +\infty$;

(iii)
- The NPMC problem is feasible, and strong duality fails $\Rightarrow$ For any $\boldsymbol{\lambda}^* \in \arg\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} G(\boldsymbol{\lambda})$, $\boldsymbol{\lambda}^*$ is infinite [5] or $\boldsymbol{\lambda}^*$ is finite but $\phi^*_{\boldsymbol{\lambda}^*}$ is infeasible, and $G(\boldsymbol{\lambda}^*) \leq 1$;
- For any $\boldsymbol{\lambda}^* \in \arg\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} G(\boldsymbol{\lambda})$, $\boldsymbol{\lambda}^*$ is infinite or $\boldsymbol{\lambda}^*$ is finite but $\phi^*_{\boldsymbol{\lambda}^*}$ is infeasible, and $G(\boldsymbol{\lambda}^*) \leq 1 \Rightarrow$ strong duality fails, and the NPMC problem can be either feasible or infeasible;

(iv)
- The NPMC problem is infeasible, and strong duality fails $\Rightarrow$ For any $\boldsymbol{\lambda}^* \in \arg\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} G(\boldsymbol{\lambda})$, $\boldsymbol{\lambda}^*$ is infinite or $\boldsymbol{\lambda}^*$ is finite but $\phi^*_{\boldsymbol{\lambda}^*}$ is infeasible, and $G(\boldsymbol{\lambda}^*) < +\infty$.
- For any $\boldsymbol{\lambda}^* \in \arg\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} G(\boldsymbol{\lambda})$, $\boldsymbol{\lambda}^*$ is infinite or $\boldsymbol{\lambda}^*$ is finite but $\phi^*_{\boldsymbol{\lambda}^*}$ is infeasible, and $1 < G(\boldsymbol{\lambda}^*) < +\infty \Rightarrow$ strong duality fails, the NPMC problem is infeasible.

Corollary 1 establishes a connection between NPMC strong duality and feasibility with the optimal $\boldsymbol{\lambda}^*$ and the value of $G(\boldsymbol{\lambda}^*)$. In practice, $\boldsymbol{\lambda}^*$ and $G(\boldsymbol{\lambda}^*)$ can be estimated by $\hat{\boldsymbol{\lambda}}$ and $\widehat{G}^{\mathrm{CX}}(\hat{\boldsymbol{\lambda}})$ from NPMC-CX or $\hat{\boldsymbol{\lambda}}$ and $\widehat{G}^{\mathrm{ER}}(\hat{\boldsymbol{\lambda}})$ from NPMC-ER. The equivalences in Corollary 1 can then be used to assess whether feasibility and strong duality hold. Due to space constraints, further details are provided in Section S.1.1 in the supplements, while related empirical results will be discussed in Section 4.

### 3.2. Analysis on NPMC-CX

It is well-known that regardless of the primal problem, the Lagrangian dual function is always concave (Luenberger 1997; Boyd and Vandenberghe 2004), implying that $G(\boldsymbol{\lambda})$ in (8) is concave w.r.t. $\boldsymbol{\lambda}$. For NPMC-CX, the empirical version $\widehat{G}(\boldsymbol{\lambda})$ in (11) is a concave function as well, making (13) a convex optimization problem.

*Proposition 1.* $G(\boldsymbol{\lambda})$ and $\widehat{G}^{\mathrm{CX}}(\boldsymbol{\lambda})$ are concave and continuous on $\mathbb{R}_+^{|\mathcal{A}|}$.

Suppose we estimate $\mathbb{P}_{Y|X=\boldsymbol{x}}(Y = k)$ with a function class $\mathcal{M}$ that can be indexed by an index $\boldsymbol{\beta} \in \mathcal{B}$, where $\mathcal{B}$ is some metric space. Suppose the data dimension $p$ is fixed.

To prove the NP oracle properties of NPMC-CX, we impose the following assumptions.

*Assumption 2 (Model consistency).* $\max_k \mathbb{E}|\widehat{\mathbb{P}}_{Y|X}(Y = k) - \mathbb{P}_{Y|X}(Y = k)| \to 0$ as $n \to \infty$.

*Assumption 3 (Strict concavity).* $G(\boldsymbol{\lambda})$ is continuously twice-differentiable at $\boldsymbol{\lambda}^*$ and $\nabla^2 G(\boldsymbol{\lambda}^*) \prec 0$, where $\boldsymbol{\lambda}^* = \arg\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} G(\boldsymbol{\lambda})$ is unique.

*Assumption 4 (Rademacher classes).* The function class for estimating conditional probability $\mathcal{M} = \{\{\widehat{\mathbb{P}}_{Y|X=\boldsymbol{x}}(Y = k; \boldsymbol{\beta})\}_{k=1}^K : \boldsymbol{\beta} \in \mathcal{B}\}$ has a vanishing Rademacher complexity

$$C_{\mathrm{Rad}}(n) := \max_{k=1:K} \mathbb{E} \sup_{\boldsymbol{\beta}\in\mathcal{B}} \left| \frac{1}{n} \sum_{i=1}^n \epsilon_i \widehat{\mathbb{P}}_{Y|X=\boldsymbol{x}_i}(Y = k; \boldsymbol{\beta}) \right| \to 0,$$

as $n \to \infty$, where $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)^T$ is a vector of independent Rademacher variables.

*Assumption 5 (Margin condition).* Denote the function characterizing the decision boundary of class $k$ as $\varphi_k(\boldsymbol{x}) = c_k(\boldsymbol{\lambda}^*, \boldsymbol{\pi}^*)\mathbb{P}_{Y|X=\boldsymbol{x}}(Y = k) - \max_{j\neq k}\{c_j(\boldsymbol{\lambda}^*, \boldsymbol{\pi}^*)\mathbb{P}_{Y|X=\boldsymbol{x}}(Y = j)\}$, where $\boldsymbol{\lambda}^* = \arg\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} G(\boldsymbol{\lambda})$ is unique. It holds that

$$\max_{k=1:K} \mathbb{P}_{X|Y=k}(|\varphi_k(X)| \leq \tau) \lesssim \tau^{\bar{\gamma}},$$

with some $\bar{\gamma} > 0$ and any nonnegative $\tau$ smaller than some constant $C \in (0, 1)$.

*Remark 1.* Assumption 2 ensures that the conditional probability can be accurately estimated. Assumption 3 is motivated by the second-order information condition used in proving MLE consistency (Wald 1949; Van der Vaart 2000).

Algorithm 4 restricts the model complexity [6]. Many parametric model classes fulfill this condition, such as the multinomial logistic model with bounded coefficients when $\mathbb{P}_X$ has second-order moments. Additionally, certain nonparametric classes also satisfy this requirement, such as Lipschitz function classes with $\widehat{\mathbb{P}}_{Y|X=\boldsymbol{x}}(Y = k; \boldsymbol{\beta})$ Lipschitz in $\boldsymbol{x}$ when $\mathbb{P}_X$ is supported on a bounded set of $\mathbb{R}^p$. Note that the function class $\mathcal{M}$ does not necessarily correspond to the underlying true model, and we do not require the true model to belong to a Rademacher class.

Assumption 5 is commonly referred to as "margin condition" in literature (Audibert and Tsybakov 2007; Tong 2013; Zhao et al. 2016), and it requires most data points to be away from the optimal decision boundary. In many cases, this assumption can lead to convergence rates faster than $\mathcal{O}_{\mathbb{P}}(n^{-1/2})$. Previous binary NP classification papers such as Tong (2013), Zhao et al. (2016) and Tong et al. (2020) do not require it when arbitrary convergence rates are acceptable. Besides, it is often employed

[5] When we say infinite $\boldsymbol{\lambda}^*$, we refer to a sequence $\{(\boldsymbol{\lambda}^*)^{(m)}\}_{m=1}^\infty$ s.t. $\|(\boldsymbol{\lambda}^*)^{(m)}\|_\infty \to +\infty$, $\lim_{m\to\infty} G((\boldsymbol{\lambda}^*)^{(m)}) = \sup_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} G(\boldsymbol{\lambda})$ exists and is denoted as $G(\boldsymbol{\lambda}^*)$.

[6] More precisely, such a restriction also depends on $\mathbb{P}_X$ because $\mathbb{E}$ is w.r.t. all the random ness.

with an opposite condition called "detection condition" (Tong 2013; Zhao et al. 2016; Tong et al. 2020) to aid in accurately estimating the optimal classification threshold. Here, we do not need such a detection condition, but Assumption 5 is crucial and required to hold.

More discussions can be found in Section S.1.3 of the supplementary materials.

Next, we establish that NPMC-CX satisfies the multi-class NP oracle properties under the conditions above.

*Theorem 2 (Multi-class NP oracle properties of NPMC-CX).* NPMC-CX satisfies multi-class NP oracle properties in the following senses.

(i) When the NPMC problem (2) is feasible, if Assumptions 1-5 hold, and $\delta \gtrsim \left[R_{\mathrm{Rad}}(n)+\max_k \mathbb{E}|\widehat{\mathbb{P}}_{Y|X}(Y=k)-\mathbb{P}_{Y|X}(Y=k)|\right]^{\bar{\gamma}/2}$, then there exist a solution $\phi^*$ and a constant $C>0$ such that

$$\max_k \mathbb{P}(|R_k(\hat{\phi})-R_k(\phi^*)|>\tau) \lesssim \exp\{-Cn\tau^{4/\bar{\gamma}}\}$$
$$+\tau^{-\frac{2\vee(1+\bar{\gamma})}{\bar{\gamma}}} \max_k \mathbb{E}\left|\widehat{\mathbb{P}}_{Y|X}(Y=k)-\mathbb{P}_{Y|X}(Y=k)\right|,$$

when $1 \geq \tau \gtrsim \left[C_{\mathrm{Rad}}(n)+\max_k \mathbb{E}|\widehat{\mathbb{P}}_{Y|X}(Y=k)-\mathbb{P}_{Y|X}(Y=k)|\right]^{\bar{\gamma}/2}$.

(ii) When the NPMC problem (2) is infeasible, if Assumptions 1, 2, and 4 hold, and $\delta \gtrsim \left[R_{\mathrm{Rad}}(n)+\max_k \mathbb{E}|\widehat{\mathbb{P}}_{Y|X}(Y=k)-\mathbb{P}_{Y|X}(Y=k)|\right]^{1/2}$, then there exists a constant $C>0$ such that

$$\mathbb{P}\left(\widehat{G}^{\mathrm{CX}}(\hat{\boldsymbol{\lambda}}) \leq 1+\delta\right) \lesssim \exp\{-Cn\}$$
$$+\max_k \mathbb{E}\left|\widehat{\mathbb{P}}_{Y|X}(Y=k)-\mathbb{P}_{Y|X}(Y=k)\right|,$$

where $\delta$ is an input parameter in Algorithm 1.

*Remark 2.* Observe that $J(\hat{\phi})-J(\phi^*)$ is a linear combination of $\{R_k(\hat{\phi})-R_k(\phi^*)\}_{k=1}^K$. Hence, when the NPMC problem (2) is feasible,

$$R_k(\hat{\phi})-\alpha_k \leq R_k(\hat{\phi})-R_k(\phi^*) \leq \mathcal{O}_{\mathbb{P}}(\epsilon(n)), \quad \forall k \in \mathcal{A},$$
$$J(\hat{\phi})-J(\phi^*) \leq \mathcal{O}_{\mathbb{P}}(\epsilon(n)),$$

where $\epsilon(n) = n^{-\bar{\gamma}/4} + \left(\max_k \mathbb{E}|\widehat{\mathbb{P}}_{Y|X}(Y=k)-\mathbb{P}_{Y|X}(Y=k)|\right)^{\bar{\gamma}/(2\vee(1+\bar{\gamma}))} + (C_{\mathrm{Rad}}(n))^{\bar{\gamma}/2} \to 0$. Theorem 2 verifies multi-class NP oracle properties as defined in Section 1.4.

### 3.3. Analysis on NPMC-ER

One advantage of NPMC-ER over NPMC-CX is that it does not require $\widehat{\mathbb{P}}_{Y|X=\boldsymbol{x}}(Y=k)$ to belong to a Rademacher class. We will explain the intuition in the next section.

Unlike NPMC-CX, for NPMC-ER, the empirical dual function $\widehat{G}(\boldsymbol{\lambda})$ in (16) is not necessarily concave. This discrepancy arises from the "mismatch" of $\widehat{L}^{\mathrm{ER}}(\boldsymbol{\lambda},\phi)$ and $\hat{\phi}_{\boldsymbol{\lambda}}$. As discussed in Section 2.2, given $\boldsymbol{\lambda}$, $\hat{\phi}_{\boldsymbol{\lambda}}$ is not necessarily a minimizer of $\widehat{L}^{\mathrm{ER}}(\boldsymbol{\lambda},\phi)$, leading to a dual function not of the "max-min" type and hence not necessarily concave. Nonetheless, the multi-class NP oracle properties still hold under similar conditions.

*Theorem 3 (Multi-class NP oracle properties of NPMC-ER).* NPMC-ER satisfies multi-class NP oracle properties in the following senses.

(i) When the NPMC problem (2) is feasible, if Assumptions 1, 2, 3, and 5 hold, $\delta \gtrsim n^{-\bar{\gamma}/4}$, and $R \geq \|\boldsymbol{\lambda}^*\|_\infty$ with $\boldsymbol{\lambda}^* = \arg\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^{|\mathcal{A}|}} G(\boldsymbol{\lambda})$, then there exist a solution $\phi^*$ and a constant $C>0$ such that

$$\max_k \mathbb{P}(|R_k(\hat{\phi})-R_k(\phi^*)|>\tau) \lesssim \exp\{-Cn\tau^{4/\bar{\gamma}}\}$$
$$+\tau^{-\frac{2\vee(1+\bar{\gamma})}{\bar{\gamma}}} \max_k \mathbb{E}\left|\widehat{\mathbb{P}}_{Y|X}(Y=k)-\mathbb{P}_{Y|X}(Y=k)\right|,$$

when $1 \geq \tau \gtrsim n^{-\bar{\gamma}/4}$.

(ii) When the NPMC problem (2) is infeasible, if Assumptions 1 and 2 hold, $\delta \gtrsim n^{-1/4}$, and $R$ satisfies $\sup_{\|\boldsymbol{\lambda}\|_\infty \leq R} G(\boldsymbol{\lambda}) > 1+\delta$, then there exists a constant $C>0$ such that

$$\mathbb{P}\left(\widehat{G}^{\mathrm{ER}}(\hat{\boldsymbol{\lambda}}) \leq 1+\delta\right) \lesssim \exp\{-Cn\}$$
$$+\max_k \mathbb{E}\left|\widehat{\mathbb{P}}_{Y|X}(Y=k)-\mathbb{P}_{Y|X}(Y=k)\right|,$$

where $\delta$ is an input parameter in Algorithm 1.

Analyzing similarly in Remark 2, we conclude that Theorem 3 confirms the multi-class NP oracle properties of NPMC-ER. As discussed in Section 2.2, because $\widehat{G}^{\mathrm{ER}}(\boldsymbol{\lambda})$ is not necessarily concave, for technical reasons, we can only search for optimal $\boldsymbol{\lambda}$ within a bounded region $\|\boldsymbol{\lambda}\|_\infty \leq R$ where $R>0$ is a constant. On the other hand, to ensure that this search region covers the true optimal $\boldsymbol{\lambda}^*$ (when the NPMC problem is feasible) or is large enough to find a large $G(\boldsymbol{\lambda})$ value (when the NPMC problem is infeasible), we need to ensure that $R$ is not very small, leading to the conditions $R \geq \|\boldsymbol{\lambda}^*\|_\infty$ and $\sup_{\|\boldsymbol{\lambda}\|_\infty \leq R} G(\boldsymbol{\lambda}) > 1+\delta$ in (i) and (ii), respectively. The empirical results are not very sensitive to the choice of $R$, and we set $R = 1000$ in all numerical studies.

### 3.4. Comparison of NPMC-CX and NPMC-ER from Theoretical Perspective

We now summarize the difference between the two algorithms from theoretical perspectives.

- Both NPMC-CX and NPMC-ER exhibit NP oracle properties under certain conditions.
- NPMC-CX assumes the function class used to estimate the posterior $\mathbb{P}_{Y|X=\boldsymbol{x}}(Y=k)$ has a vanishing Rademacher complexity, while NPMC-ER does not impose such a restriction. This distinction arises because NPMC-CX uses all training data simultaneously, necessitating control over model complexity for certain uniform convergence results. In contrast, NPMC-ER leverages sample splitting, creating independence that only requires pointwise convergence instead of uniform convergence, regardless of the model class considered. Further details are available in the corresponding proofs provided in supplementary materials.

## 4. Numerical Experiments

We demonstrate the effectiveness of NPMC-CX and NPMC-ER through a simulation example and a real data study on loan

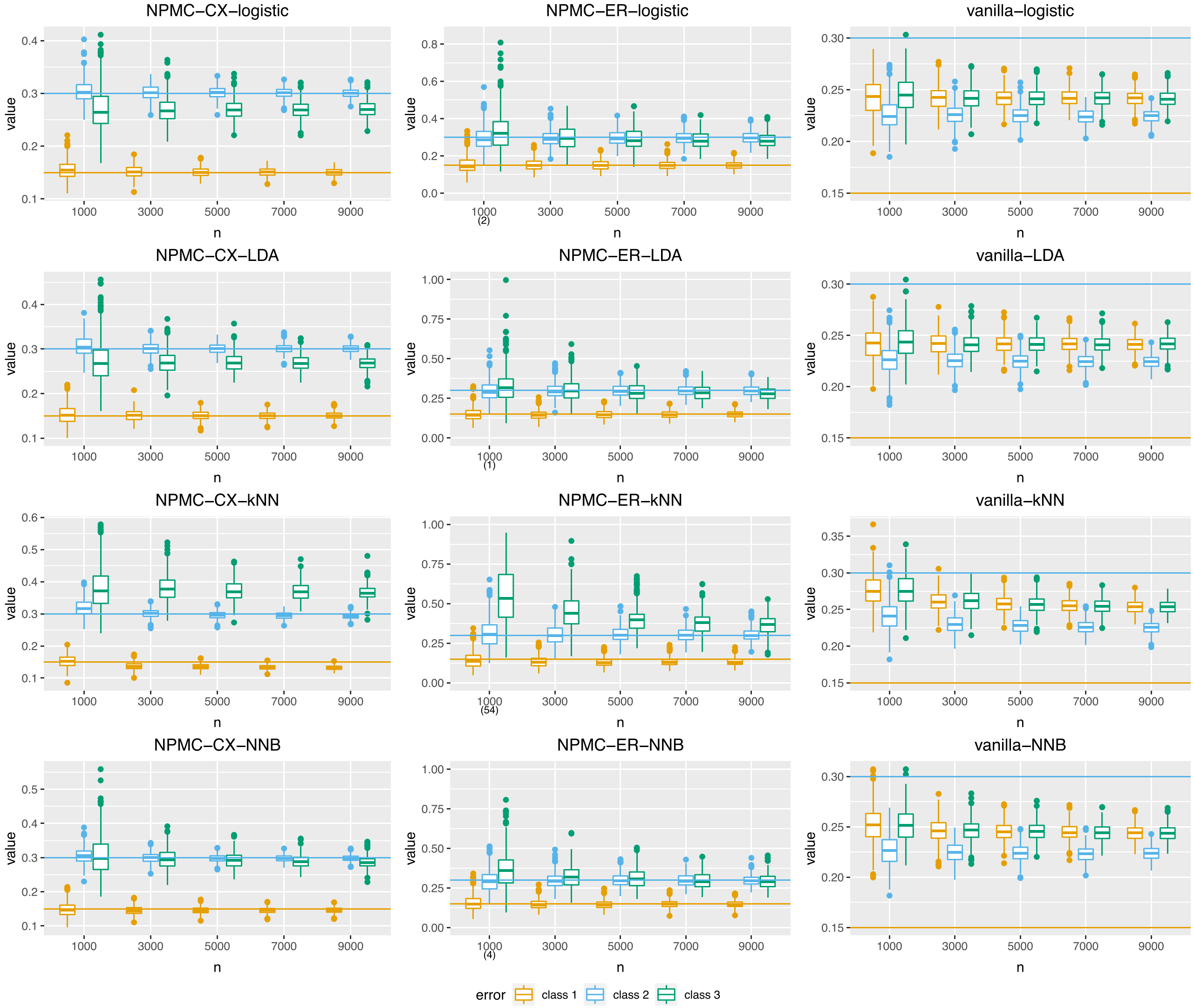

**Figure 1.** Per-class error rates under each classifier and training sample size setting in simulation. Horizontal lines in corresponding colors mark the target control levels. In some graphs, additional values are displayed in brackets beneath the training sample size, $n$. These values represent the number of instances where the algorithms reported infeasibility during evaluation.

default prediction. All numerical experiments were conducted using R. Our proposed algorithms, NPMC-CX and NPMC-ER, have been implemented in the package npcs (*https://CRAN.R-project.org/package=npcs*). In the simulations, we vary the training sample size $n$ from 1000 to 9000 with an increment of 2000, while keeping the test sample size fixed at 20,000. Without specific notice, each setting in both simulations and real data studies is repeated 500 times. Due to space constraints, we provide additional numerical results and more implementation details, including the choice of tuning parameters in Section S.4 of the supplementary materials. All the code is available at *https://github.com/ytstat/NPMC*.

### 4.1. Simulation

Consider a three-class independent Gaussian conditional distributions $X|Y = k \sim N(\boldsymbol{\mu}_k, \boldsymbol{I}_p)$, where $p = 5$, $\boldsymbol{\mu}_1 = (-1, 2, 1, 1, 1)^T$, $\boldsymbol{\mu}_2 = (0, 1, 0, 1, 0)^T$, $\boldsymbol{\mu}_3 = (1, 1, -1, 0, 1)^T$ and $\boldsymbol{I}_p$ is the $p$-dimensional identity matrix. The marginal distribution of $Y$ is $\mathbb{P}(Y = 1) = \mathbb{P}(Y = 3) = 0.3$ and $\mathbb{P}(Y = 2) = 0.4$.

We aim to solve the following NPMC problem:

$$\begin{aligned} \min_{\phi} \quad & \mathbb{P}_{X|Y=3}(\phi(X) \neq 3) \\ \text{s.t.} \quad & \mathbb{P}_{X|Y=1}(\phi(X) \neq 1) \leq 0.15, \quad \mathbb{P}_{X|Y=2}(\phi(X) \neq 2) \leq 0.3. \end{aligned}$$

We run the proposed algorithms NPMC-CX and NPMC-ER with four function classes to estimate $\mathbb{P}_{Y|X}$, including logistic regression, LDA, $k$NN, and nonparametric naïve Bayes model with Gaussian kernel. For comparison, we also fit four corresponding vanilla classifiers trained without error controls as benchmarks. Boxplots show the per-class error rates under each method and training sample size setting in Figure 1.

One can see that vanilla classifiers fail to control the error of class 1 and "over-control" the error of class 2. In contrast, NPMC-CX and NPMC-ER work very well by controlling the error rates around the target control level, which matches our theoretical results in Section 3. By comparing the error rates of class 2 between NPMC methods and vanilla classifiers, we observe that to achieve a successful control over $\mathbb{P}_{X|Y=1}(\phi(X) \neq$

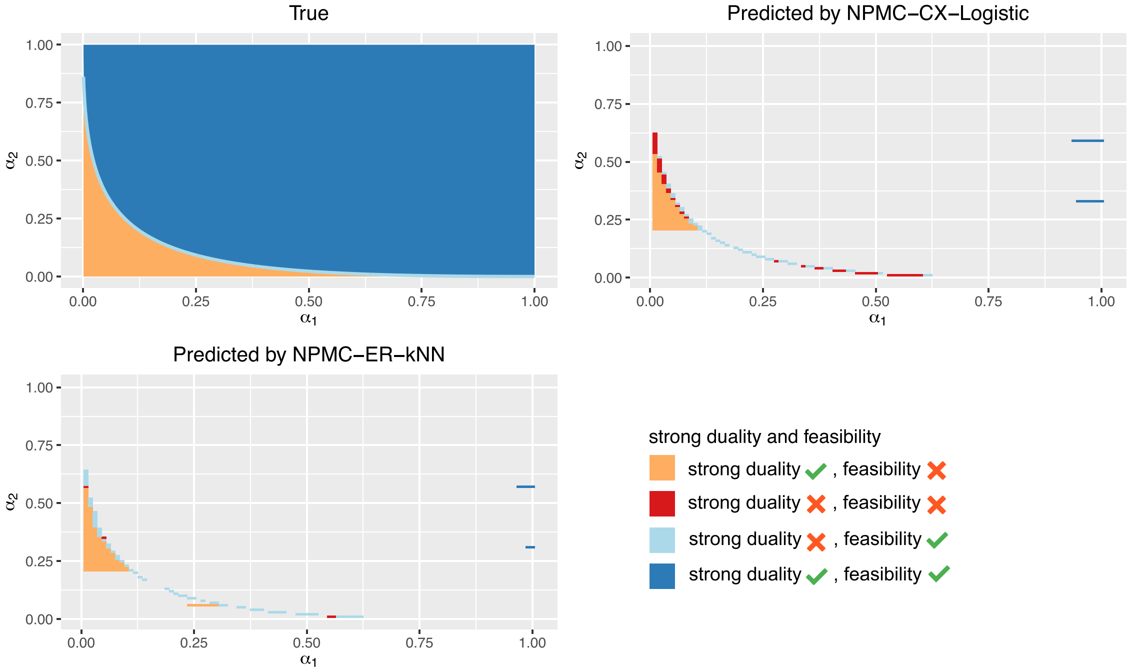

**Figure 2.** Strong duality and feasibility of simulation: ground truth and predicted results.

1) [7] and $\mathbb{P}_{X|Y=2}(\phi(X) \neq 2)$ around the corresponding levels, there is a cost in terms of the performance on class 3. When the training sample size $n$ increases, the variance of error rates for each method tends to decrease. For NPMC-CX-LDA and NPMC-CX-NNB, when $n$ is small, sometimes the algorithm outputs the infeasibility warning. For NPMC-CX-LDA, this behavior might be due to LDA's higher sample size requirements (because of the need to estimate the covariance matrix) compared to other methods like logistic regression. For NPMC-CX-NNB, this phenomenon could be caused by the improper choice of bandwidth.

Another noteworthy observation is the higher variances of error rates on class 3 compared to the other two classes, particularly evident when $n$ is small. This phenomenon arises because the decision boundary of NP classifiers traverses the densely populated area for class 3 but not for classes 1 and 2 when stringent error controls are imposed on the latter. Consequently, even a small change in the decision boundary can lead to a relatively bigger change in the error of class 3 compared to classes 1 and 2.

To validate the feasibility and strong duality checking algorithms induced by Corollary 1 (see the algorithms in Section S.1.1 of the supplements and note that the feasibility prediction is the same as in NPMC-CX and NPMC-ER), we conducted experiments for them with NPMC-CX-logistic and NPMC-ER-$k$NN by fixing the random training data of size $n = 10^5$ and considering all choices of $(\alpha_1, \alpha_2)$ within range $[0.01, 1]^2$ with a grid size 0.01. Note that the feasibility and strong duality can be theoretically verified for any specific $(\alpha_1, \alpha_2)$ in this example. The following lemma, in conjunction with Lemma 1, establishes the ground truth regarding strong duality and feasibility.

*Lemma 3.* The strong duality in Assumption 1 holds for 3-class NPMC problem (2) with $X|Y = k \sim N(\boldsymbol{\mu}_k, \boldsymbol{I}_p)$ for $k = 1, 2, 3$, $\mathcal{A} = \{1, 2\}$, and the target levels $\alpha_1, \alpha_2 \in [0, 1]$, if and only if

$$\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2 \neq \Phi^{-1}(1 - \alpha_1) + \Phi^{-1}(1 - \alpha_2),$$

where $\Phi^{-1}$ is the inverse CDF function of $N(0, 1)$.

We then compared the true feasibility and strong duality with the predictions generated by our feasibility and strong duality checking algorithms in Figure 2. It shows that our algorithms can accurately predict the feasibility and strong duality with sufficient data. Hence, practitioners can first use algorithms in Section S.1.1 to assess the feasibility and strong duality for various target error levels, thereby gaining insights into the problem difficulty, especially when they are unsure about the appropriate target levels for error controls. In other words, our feasibility and strong duality checking algorithms offer a prediction of the *landscape* of an NPMC problem with various target levels.

### 4.2. A Real Data Study: Loan Default Prediction

Identifying high-risk customers prone to late payments or default is paramount for banks and lending institutions in managing risk. Providing loans to high-risk customers often results in greater losses than denying loans to low-risk customers, underscoring the importance of effective risk assessment strategies. The Neyman-Pearson classification framework is particularly valuable in this context for its ability to address asymmetric errors.

LendingClub, a peer-to-peer lending company, caters to borrowers seeking personal loans ranging from $1000 to $40,000. The LendingClub dataset (*https://www.kaggle.com/code/emmaruyiyang/lending-club-loan-default-prediction-eda/input*) encompasses loan data spanning from 2007 to 2015. It includes details such as loan amount, term length, current status,

[7] To be more precise, the graphs only show the empirical error rates on the test data.

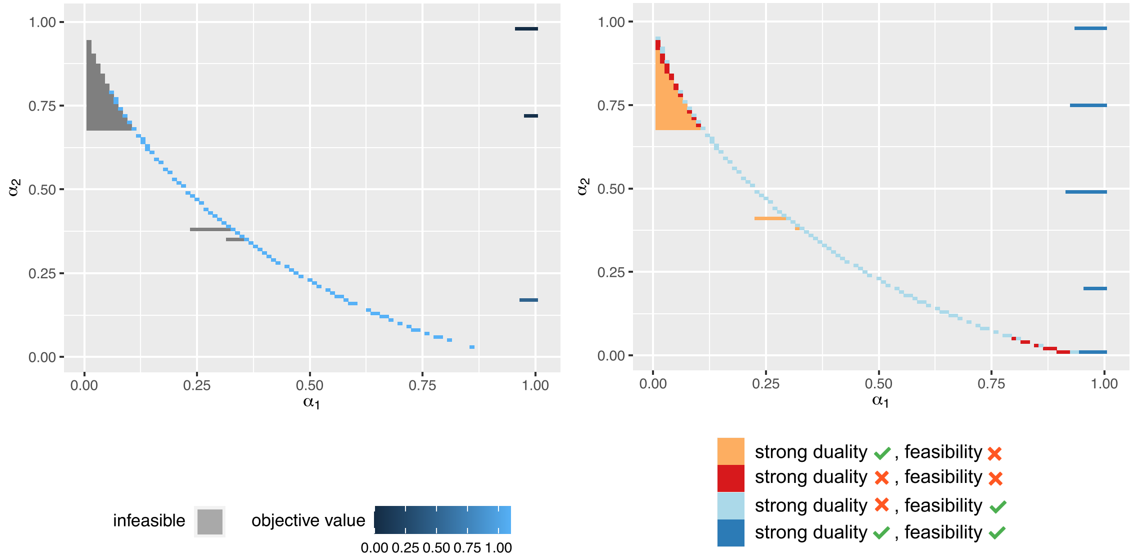

**Figure 3.** Strong duality and feasibility of NPMC problem for the LendingClub dataset with different target error levels: predicted by Algorithm 3 with NPMC-CX-logistic.

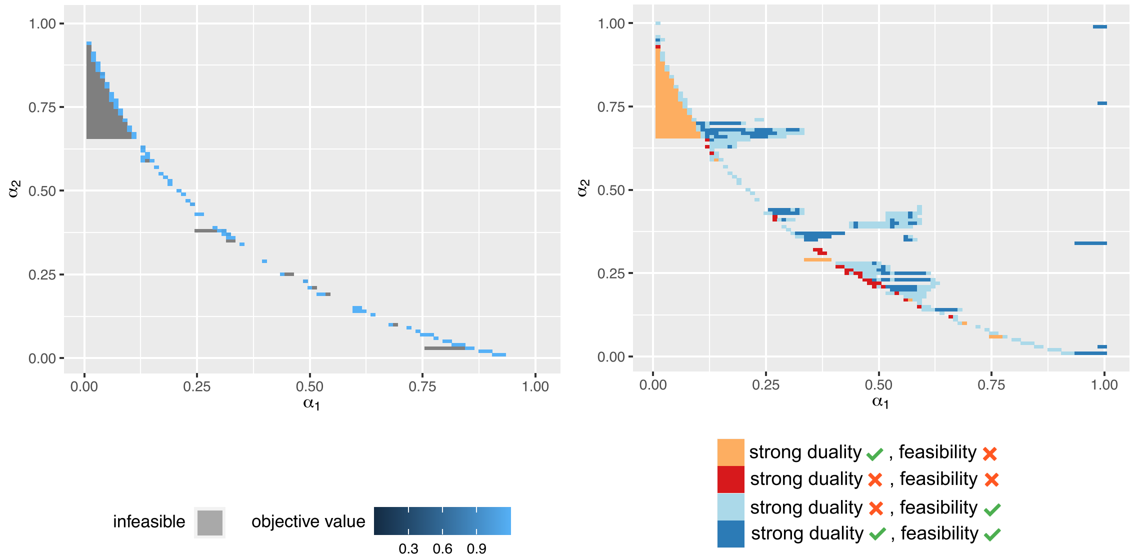

**Figure 4.** Strong duality and feasibility of NPMC problem for the LendingClub dataset with different target error levels: predicted by Algorithm 4 with NPMC-ER-RF.

and borrower information like annual income and number of bankcard accounts. The objective is to predict the loan status based on these variables. The original dataset contains various labels for loan status, including “fully paid”, “late payment” with varying durations, “in grace period”,“default”, and “charge off”. For simplicity, we categorize them into three groups: class 1 (bad status: default or charge off), class 2 (fair status: late payment but not default), and class 3 (excellent status: fully paid). Following some preprocessing steps (refer to Section S.4.3.1 for details), the dataset comprises 264,274 observations with 25 features and 1 response variable. The sample sizes for the three classes are 45,072 (17.1%), 19,265 (7.3%), and 199,937 (75.6%), respectively. The significant class imbalance poses an additional challenge in addressing this problem.

We would like to solve the following NPMC problem

$$\min_{\phi} \quad \mathbb{P}_{X|Y=3}(\phi(X) \neq 3)$$

$$\text{s.t.} \quad \mathbb{P}_{X|Y=1}(\phi(X) \neq 1) \leq \alpha_1, \quad \mathbb{P}_{X|Y=2}(\phi(X) \neq 2) \leq \alpha_2,$$

where $\alpha_1$ is typically chosen to be smaller than $\alpha_2$ because misclassifying observations of class 1 is more detrimental than misclassifying those of class 2.

As described in Section 4.1, practitioners can experiment with various target levels $(\alpha_1, \alpha_2)$ using our feasibility and strong duality checking algorithms (Algorithms 3 and 4 in Section S.1.1) to assess the problem’s complexity and select the target level based on feasibility and practical considerations. We present the predicted strong duality, feasibility, and objective values using Algorithm 3 with NPMC-CX-logistic (NPMC-CX with $\mathcal{M}$ as logistic regression) and Algorithm 4 with NPMC-ER-RF (NPMC-ER with $\mathcal{M}$ as random forests) on the entire dataset for different $(\alpha_1, \alpha_2) \in [0, 1]^2$, in Figures 3 and 4, respectively. These figures illustrate the tradeoff between error rates for the three classes. To ensure the feasibility of the NP problem, the error thresholds $(\alpha_1, \alpha_2)$ must not be set too low. This requirement largely stems from the intrinsic complexity of the task, especially the challenge of distinguishing between

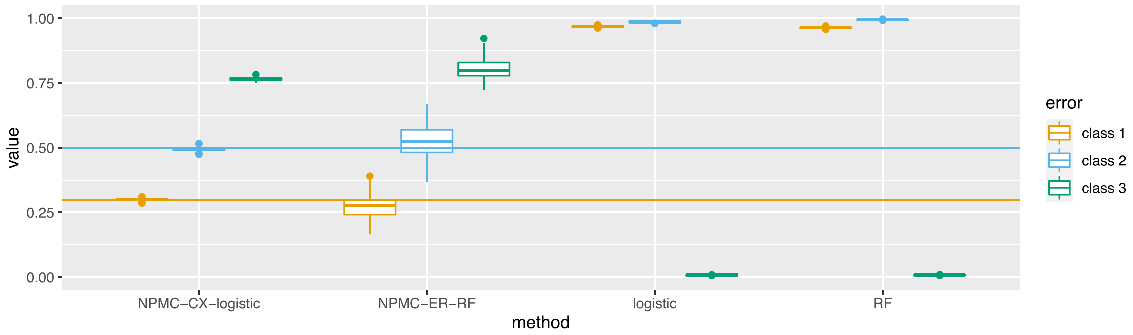

**Figure 5.** Per-class error rates and objective function values under each classifier for the NPMC problem on the LendingClub dataset. Horizontal lines in corresponding colors mark the target control levels.

classes 1 and 2. When logistic regression and random forests are trained solely on data from these classes, both methods display a binary misclassification error rate approaching 30%, underscoring the inherent difficulty.

Next, we fix $\alpha_1 = 0.3$ and $\alpha_2 = 0.5$, and conduct experiments with NPMC-CX-logistic and NPMC-ER-RF, alongside vanilla logistic regression and random forests as benchmarks. We randomly split the entire data into 50% training and 50% testing data over 500 replications. Boxplots in Figure 5 display the per-class error rates under each classifier and across various training sample size settings. Notably, vanilla logistic regression and random forests tend to assign all observations to class 3 due to the significant imbalance in sample sizes. In contrast, NPMC-CX-logistic and NPMC-ER-RF effectively control the error rates of classes 1 and 2 around the specified target levels.

We also run similar experiments on the confusion matrix control problem outlined in Section 1.2 and the detailed results are presented in Section S.4.3.1.

### 4.3. Comparison of NPMC-CX and NPMC-ER from Experimental Perspective

From the previous numerical results, we can observe that:

- NPMC-CX works better under parametric models (e.g., logistic and LDA) by controlling the error rates well and achieving a lower objective function value compared to NPMC-ER, but can sometimes fail to control error rates under target levels for nonparametric models (e.g., *k*NN, RF, and SVM with RBF kernel).
- Compared to NPMC-CX, NPMC-ER requires a larger sample size to perform well due to sample splitting in Algorithm 2, but it is more robust to different model types.

These observations align well with our intuition from theoretical analysis (Section 3.4). Therefore, for practitioners, if a Rademacher class (usually parametric) is believed to be suitable for the problem at hand, we suggest using NPMC-CX. If the nonparametric model is believed to work better and sample size is not very small, we suggest using NPMC-ER.

## 5. Discussions

### 5.1. Summary

In this article, we connect Neyman-Pearson multi-class classification (NPMC) problems with cost-sensitive learning (CS) problems, and propose two algorithms, NPMC-CX and NPMC-ER, to solve the NPMC problem (2) via CS techniques. To our knowledge, this is the first work solving NPMC problems with theoretical guarantees. We have presented some theoretical results, including conditions for strong duality and multi-class NP oracle properties for the two algorithms. Furthermore, we propose practical algorithms to verify the NPMC feasibility and strong duality, which can offer practitioners a landscape of the NPMC problem with various target error levels. Our algorithms are shown to be effective through extensive simulations and real data studies.

Comparing NPMC-CX and NPMC-ER, we find:

- Both algorithms are shown to satisfy multi-class NP properties. However, NPMC-CX necessitates a function class with a vanishing Rademacher complexity for estimating $\mathbb{P}_{Y|X=\boldsymbol{x}}(Y = k)$, while NPMC-ER has no such constraints.
- In practice, NPMC-CX works well for parametric models but may struggle with some nonparametric models. Due to data splitting, NPMC-ER requires a larger sample size but is more robust to diverse model types.
- Therefore, we suggest the practitioners go with NPMC-CX when a parametric model is favored. When the nonparametric model is believed to work better, and there is enough training data, we suggest using NPMC-ER.

Furthermore, the general confusion matrix control problem outlined in Section 1.2 is discussed in detail in Section S.2 of supplementary materials, and we extended our two NPMC algorithms to solve that problem. The theoretical results are also provided.

### 5.2. Future Research Directions

There are many interesting future avenues to explore. Here, we list three of them.

(i) There are many approaches to fitting a CS classifier. We use (10) to fit the CS classifier in our NPMC algorithms, which sometimes is called the thresholding strategy in binary CS problems (Dmochowski, Sajda, and Parra 2010). It might be interesting to explore other approaches and replace (10) accordingly.

(ii) Li, Tong, and Li (2020) studied the methodological relationship between the binary NP paradigm and CS paradigm, and constructed a CS classifier with Type-I error controls. In this article, we focus on the multi-class NP paradigm and build a multi-class NP classifier via CS learning, which can be viewed as the inverse to Li, Tong, and Li (2020). Exploring the other direction in the multi-class cases would be interesting: developing multi-class CS classifiers with specific error controls.

(iii) As one reviewer pointed out, the current multi-class NP oracle properties might not be strong enough in some degenerated cases where the NPMC problem can vary with $n$ and $J(\phi^*) = \mathcal{O}(1)$ or $\alpha_k = \mathcal{O}(1)$ for some $k \in \mathcal{A}$. It would be intriguing to generalize the existing multi-class NP oracle properties from $R_k(\hat{\phi}) \leq \alpha_k + \mathcal{O}_{\mathbb{P}}(1), \forall k \in \mathcal{A}$ and $J(\hat{\phi}) \leq J(\phi^*) + \mathcal{O}_{\mathbb{P}}(1)$ to $R_k(\hat{\phi}) \leq \alpha_k + \mathcal{O}_{\mathbb{P}}(\alpha_k), \forall k \in \mathcal{A}$ and $J(\hat{\phi}) \leq J(\phi^*) + \mathcal{O}_{\mathbb{P}}(J(\phi^*))$.

## Supplementary Materials

The supplementary materials contain additional technical details of the NPMC problem, extension to the confusion matrix control problem, extension to worst-case object functions, additional numerical results, technical lemmas and propositions, and all proofs.

## Acknowledgments

We thank the Coeditor, the AE, and two anonymous reviewers for their insightful comments, which greatly improved a prior version of the article.

## Funding

This research was partially supported by NIH Grant 1R21AG074205-01, NSF Grant DMS-2324489, a grant from the New York University School of Global Public Health, NYU University Research Challenge Fund. All experiments were conducted on Ginsburg HPC Cluster of Columbia University.

## Disclosure Statement

The authors report there are no competing interests to declare.

## ORCID

Ye Tian http://orcid.org/0000-0002-4772-1951
Yang Feng http://orcid.org/0000-0001-7746-7598

## References

Audibert, J.-Y., and Tsybakov, A. B. (2007), "Fast Learning Rates for Plug-in Classifiers," *The Annals of Statistics*, 35, 608–633. [1170]

Boyd, S., and Vandenberghe, L. (2004), *Convex Optimization*, Cambridge: Cambridge University Press. [1169,1170]

Cannon, A., Howse, J., Hush, D., and Scovel, C. (2002), "Learning with the Neyman-Pearson and Min-Max Criteria," *Los Alamos National Laboratory, Technical Report LA-UR*, pp. 02–2951. [1164,1167]

Dmochowski, J. P., Sajda, P., and Parra, L. C. (2010), "Maximum Likelihood in Cost-Sensitive Learning: Model Specification, Approximations, and Upper Bounds," *Journal of Machine Learning Research*, 11, 3313–3332. [1166,1175]

Domingos, P. (1999), "Metacost: A General Method for Making Classifiers Cost-Sensitive," in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 155–164. [1166]

Dreiseitl, S., Ohno-Machado, L., and Binder, M. (2000), "Comparing Three-Class Diagnostic Tests by Three-Way ROC Analysis," *Medical Decision Making*, 20, 323–331. [1165]

Edwards, D. C., Metz, C. E., and Kupinski, M. A. (2004), "Ideal Observers and Optimal ROC Hypersurfaces in n-Class Classification," *IEEE Transactions on Medical Imaging*, 23, 891–895. [1165]

Elkan, C. (2001), "The Foundations of Cost-Sensitive Learning," in *International Joint Conference on Artificial Intelligence* (Vol. 17), pp. 973–978, Lawrence Erlbaum Associates Ltd. [1166]

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., and Herrera, F. (2018), "Cost-Sensitive Learning," in *Learning from Imbalanced Data Sets*, pp. 63–78, Springer. [1166]

Hooke, R., and Jeeves, T. A. (1961), "Direct Search" Solution of Numerical and Statistical Problems," *Journal of the ACM (JACM)*, 8, 212–229. [1168]

Kalan, M. M., and Kpotufe, S. (2023), "Tight Rates in Supervised Outlier Transfer Learning," in *The Twelfth International Conference on Learning Representations*. [1167]

Kalan, M. M., and Kpotufe, S. (2024), "Distribution-Free Rates in Neyman-Pearson Classification," arXiv preprint arXiv:2402.09560. [1167]

Katsumata, S., and Takeda, A. (2015), "Robust Cost Sensitive Support Vector Machine," in *Artificial Intelligence and Statistics*, pp. 434–443, PMLR. [1166]

Landgrebe, T., and Duin, R. (2005), "On Neyman-Pearson Otimisation for Multiclass Classifiers," in *Proceedings 16th Annual Symposium of the Pattern Recognition Association of South Africa. PRASA*, pp. 165–170. [1166,1168]

Landgrebe, T. C., and Duin, R. P. (2008), "Efficient Multiclass ROC Approximation by Decomposition via Confusion Matrix Perturbation Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30, 810–822. [1166]

Li, J. J., Chen, Y. E., and Tong, X. (2021), "A Flexible Model-Free Prediction-based Framework for Feature Ranking," *Journal of Machine Learning Research*, 22, 1–54. [1165]

Li, J. J., and Tong, X. (2020), "Statistical Hypothesis Testing versus Machine Learning Binary Classification: Distinctions and Guidelines," *Patterns*, 1, 100115. [1165,1166]

Li, W. V., Tong, X., and Li, J. J. (2020), "Bridging Cost-Sensitive and Neyman-Pearson Paradigms for Asymmetric Binary Classification," arXiv preprint arXiv:2012.14951. [1165,1175,1176]

Ling, C. X., and Sheng, V. S. (2008), "Cost-Sensitive Learning and the Class Imbalance Problem," *Encyclopedia of Machine Learning*, 2011, 231–235. [1166]

Luenberger, D. G. (1997), *Optimization by Vector Space Methods*, Wiley. [1169,1170]

Ma, R., Lin, Q., and Yang, T. (2020), "Quadratically Regularized Subgradient Methods for Weakly Convex Optimization with Weakly Convex Constraints," in *International Conference on Machine Learning*, pp. 6554–6564, PMLR. [1166]

Mossman, D. (1999), "Three-Way ROCs," *Medical Decision Making*, 19, 78–89. [1165]

Nelder, J. A., and Mead, R. (1965), "A Simplex Method for Function Minimization," *The Computer Journal*, 7, 308–313. [1168]

Qin, Z., Wang, A. T., Zhang, C., and Zhang, S. (2013), "Cost-Sensitive Classification with k-nearest Neighbors," in *International Conference on Knowledge Science, Engineering and Management*, pp. 112–131, Springer. [1166]

Rigollet, P., and Tong, X. (2011), "Neyman-Pearson Classification, Convexity and Stochastic Constraints," *Journal of Machine Learning Research*, 12, 2831–2855. [1165]

Scott, C. (2007), "Performance Measures for Neyman–Pearson Classification," *IEEE Transactions on Information Theory*, 53, 2852–2863. [1164]

——— (2019), "A Generalized Neyman-Pearson Criterion for Optimal Domain Adaptation," in *Algorithmic Learning Theory*, pp. 738–761, PMLR. [1165,1167]

Scott, C., and Nowak, R. (2005), "A Neyman-Pearson Approach to Statistical Learning," *IEEE Transactions on Information Theory*, 51, 3806–3819. [1164,1166,1167]

Sheng, V. S., and Ling, C. X. (2006), "Thresholding for Making Classifiers Cost-Sensitive," in *AAAI* (Vol. 6), pp. 476–481. [1166]

Tian, Y., and Zhang, W. (2019), "THORS: An Efficient Approach for Making Classifiers Cost-Sensitive," *IEEE Access*, 7, 97704–97718. [1166]

Tong, X. (2013), "A Plug-in Approach to Neyman-Pearson Classification," *Journal of Machine Learning Research*, 14, 3011–3040. [1165,1168,1170,1171]

Tong, X., Feng, Y., and Li, J. J. (2018), "Neyman-Pearson Classification Algorithms and NP Receiver Operating Characteristics," *Science Advances*, 4, eaao1659. [1165]

Tong, X., Feng, Y., and Zhao, A. (2016), "A Survey on Neyman-Pearson Classification and Suggestions for Future Research," *Wiley Interdisciplinary Reviews: Computational Statistics*, 8, 64–81. [1165]

Tong, X., Xia, L., Wang, J., and Feng, Y. (2020), "Neyman-Pearson Classification: Parametrics and Sample Size Requirement," *Journal of Machine Learning Research*, 21, 1–48. [1165,1170,1171]

Van der Vaart, A. W. (2000), *Asymptotic Statistics* (Vol. 3), Cambridge: Cambridge University Press. [1170]

Wald, A. (1949), "Note on the Consistency of the Maximum Likelihood Estimate," *The Annals of Mathematical Statistics*, 20, 595–601. [1170]

Wang, J., Xia, L., Bao, Z., and Tong, X. (2021), "Non-Splitting Neyman-Pearson Classifiers," arXiv preprint arXiv:2112.00329. [1165]

Xia, L., Zhao, R., Wu, Y., and Tong, X. (2021), "Intentional Control of Type I Error Over Unconscious Data Distortion: A Neyman–Pearson Approach to Text Classification," *Journal of the American Statistical Association*, 116, 68–81. [1165]

Zhao, A., Feng, Y., Wang, L., and Tong, X. (2016), "Neyman-Pearson Classification Under High-Dimensional Settings," *Journal of Machine Learning Research*, 17, 7469–7507. [1165,1170,1171]

Zhou, Z.-H., and Liu, X.-Y. (2005), "Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem," *IEEE Transactions on Knowledge and Data Engineering*, 18, 63–77. [1166]

Zhou, Z.-H., and Liu, X.-Y. (2010), "On Multi-Class Cost-Sensitive Learning," *Computational Intelligence*, 26, 232–257. [1166]