Journal of the American Statistical Association

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

Set-Valued Support Vector Machine with Bounded Error Rates

Wenbo Wang & Xingye Qiao

To cite this article: Wenbo Wang & Xingye Qiao (2022): Set-Valued Support Vector Machine with Bounded Error Rates, Journal of the American Statistical Association, DOI: 10.1080/01621459.2022.2089573

To link to this article: https://doi.org/10.1080/01621459.2022.2089573

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2022, VOL. 00, NO. 0, 1–13: Theory and Methods
https://doi.org/10.1080/01621459.2022.2089573

Taylor & Francis
Taylor & Francis Group

Set-Valued Support Vector Machine with Bounded Error Rates

Wenbo Wang and Xingye Qiao

Department of Mathematics and Statistics at Binghamton University, State University of New York, Binghamton, New York, NY

ABSTRACT

This article concerns cautious classification models that are allowed to predict a set of class labels or reject to make a prediction when the uncertainty in the prediction is high. This set-valued classification approach is equivalent to the task of acceptance region learning, which aims to identify subsets of the input space, each of which guarantees to cover observations in a class with at least a predetermined probability. We propose to directly learn the acceptance regions through risk minimization, by making use of a truncated hinge loss and a constrained optimization framework. Collectively our theoretical analyses show that these acceptance regions, with high probability, satisfy simultaneously two properties: (a) they guarantee to cover each class with a noncoverage rate bounded from above; (b) they give the least ambiguous predictions among all the acceptance regions satisfying (a). An efficient algorithm is developed and numerical studies are conducted using both simulated and real data. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received February 2021
Accepted June 2022

KEYWORDS
Acceptance region learning;
Cautious classification;
Set-valued classification;
Statistical learning theory;
Support vector machine

1. Introduction

The advancement of statistics and machine learning is reshaping many fields. Increasingly, many critical decisions are made based on advanced statistical and machine learning methods, especially classification methods. It, therefore, has been important to make reliable classification and avoid making a misclassification when it is known that the chance of misclassification is high. Standard classification methods often cannot meet this demand. This is partially because a standard classifier has the goal of minimizing the overall misclassification rate and it assigns a single class label to each observation regardless of the perceived high uncertainty for some observations. However, in practice, it is often the case that an accurate single-valued prediction is difficult or impossible to obtain for some observations due to high uncertainty and lack of information. Moreover, in many applications, the consequence of misclassification for even one instance is too severe to bear for those who are affected. Examples of this kind include using classification methods to guide parole decisions, to evaluate school teachers (O'Neil 2016) and to diagnose cancers. In these high-stake domains, it is safer and more appropriate for the classifier to return a set of most plausible outcomes (e.g., class labels) for each observation and leave the final decision to a human expert or a secondary model to validate. It is desirable that this prediction set contains the true class label with high probability. Moreover, one can expect that the classifier should not make predictions at all for observations that it is highly unsure about.

In this article, we propose a set-valued multicategory classification method based on the support vector machine approach. The size of the prediction set is adaptive to the confidence that the classifier has on each observation. When it has high confidence on an observation, a single class label may be given as the prediction; otherwise, multiple class labels will be reported. Rejections may be viewed as the extreme case that all the possible class labels are predicted for an observation.[1] The standard classification method may be viewed as a special case in which the prediction set only contains one label. Therefore, standard classification should ideally only be used when there is high confidence for all the observations; unfortunately, it is rarely the case in practice.

The main difference between the standard and the set-valued classification is that the latter can no longer be framed as an (unconstrained) minimization problem of the overall misclassification rate. Set-valued classification is best understood using the following tradeoff: the larger the prediction set is, the more likely that it contains the true class label, and yet the less information such a prediction has. One way to precisely formulate this tradeoff is the acceptance region learning framework. Let the training data consist of independent and identically distributed pairs of data points $(\boldsymbol{X}_i, Y_i)$, $i = 1, 2, \ldots, n$, from an unknown distribution P, with $\boldsymbol{X}_i \in \mathcal{X} \subset \mathbb{R}^p$, and $Y_i \in \mathcal{Y} = \{1, \ldots, k\}$. The goal of acceptance region learning is to identify acceptance regions $C_j \subset \mathcal{X}$, $j = 1, \ldots, k$, one for each class, which satisfy some nice coverage properties (see details in Section 2.) Collectively these acceptance regions are equivalent to a set-valued classifier $\phi : \mathcal{X} \to 2^{\mathcal{Y}}$, defined as $\phi(\boldsymbol{x}) = \{j : \boldsymbol{x} \in C_j\}$, namely, observation $\boldsymbol{x}$ is predicted to be from a set of class labels consisting of all classes with acceptance regions that contain $\boldsymbol{x}$. Reversely, given a set-valued classifier ϕ, the equivalent acceptance regions are $C_j = \{\boldsymbol{x} : j \in \phi(\boldsymbol{x})\}$, $j = 1, \ldots, k$.

CONTACT Xingye Qiao qiao@math.binghamton.edu Department of Mathematics and Statistics at Binghamton University, State University of New York, Binghamton, New York, NY 13902.

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

[1] Abstentions refer to the situation in which the test observation is unlike any of the classes seen before, and hence is slightly different from rejections. Abstentions are beyond the scope of this article.

Lei (2014) and Sadinle, Lei, and Wasserman (2017) defined acceptance regions using two competing quantities, *confidence* and *efficiency*. The notion of *confidence* is defined as the probability that set C_j ($j = 1, \ldots, k$) covers a random observation from class j. The notion of *efficiency* is inversely related to *ambiguity*, defined as the expected number of acceptance regions C_j's that contain a random observation (equivalently, the expected size of prediction set for a random observation.) As the confidence of C_j's increases, the efficiency decreases (i.e., the ambiguity increases). The Bayes-optimal acceptance regions minimize the ambiguity with the noncoverage rate for each C_j constrained. It was shown (Sadinle, Lei, and Wasserman 2017) that the Bayes-optimal acceptance regions (or their equivalent set-valued classifier), is obtained through the conditional class probability $\eta_j(\boldsymbol{x}) \triangleq \mathsf{P}(Y = j \mid \boldsymbol{X} = \boldsymbol{x})$. Sadinle, Lei, and Wasserman (2017) proposed to use the plug-in method to estimate this set-valued classifier, that is, to first estimate $\eta_j(\boldsymbol{x})$ using a consistent estimator, then plug the estimated $\widehat{\eta}_j(\boldsymbol{x})$ into the Bayes-optimal rule. The empirical performance of the resulting set-valued classifier highly depends on the estimation accuracy of $\eta(\boldsymbol{x})$. However, as pointed out by many authors (Wang, Shen, and Liu 2007; Fürnkranz and Hüllermeier 2010; Wu, Zhang, and Liu 2010), probability estimation can be more difficult than the prediction of the class label, especially for high-dimensional data. While the requirement on estimation accuracy is somewhat relaxed in the classification context, how accurate the probability estimation needs to be is still an open question.

In this article, we propose to estimate acceptance regions and the equivalent set-valued classifiers by minimizing some empirical risk based on the support vector machine (SVM; Scholkopf and Smola 2001), bypassing the step of estimating $\eta_j(\boldsymbol{x})$. It takes advantage of the great prediction power of the SVM in both the linear and nonlinear cases. We show in theory the Fisher consistency, that is, the population minimizer of the proposed optimization is equivalent to the Bayes-optimal classifier. Moreover, in the finite-sample case, we show that the resulting classifier can control the noncoverage rates while minimizing the ambiguity.

A related problem is the Neyman-Pearson (NP) classification problem (Cannon et al. 2002; Rigollet and Tong 2011). Given a null hypothesis class, NP classification aims to identify an acceptance region for the null class which minimizes the probability that an observation from an alternative class falls into it (the Type II error) while controlling the chance that an observation from the null class is not covered by the region, and hence, is misclassified to the alternative (i.e., the Type I error). See Tong, Feng, and Zhao (2016) for a survey. The problem studied here can be regarded as solving k NP classification problems jointly.

The problem of identifying acceptance regions and its connection with the NP classification have attracted increasing attention from the statistics and machine learning communities. Dümbgen, Igl, and Munk (2008) framed it as a general p-value for classification problem. Lei (2014) proposed a framework for the binary case; Sadinle, Lei, and Wasserman (2017) extended it to the multicategory classification. Denis and Hebiri (2015) and Denis and Hebiri (2017) studied a dual problem, in which they minimized the overall noncoverage rates while controlling the ambiguity. Recently Hechtlinger, Póczos, and Wasserman (2018) and Guan and Tibshirani (2019) generalized this problem to conduct outlier detection (that is, the abstention problem). In this article, we do not consider the abstention/outlier detection problem; in other words, we assume that there is no unseen class in the training data that might appear later in the test data.

Popular ways to achieve set-valued classification include classification with reject options and conformal learning. Unlike the constrained minimization framework considered in this article, the classification with rejection methods often try to balance the ambiguity and confidence using a weighted sum of the costs of misclassification and rejection, given a predetermined weight (in the binary case, an ambiguous prediction is the same as a rejection, while confidence is related to classification accuracy). The binary version of this problem has been extensively studied (Herbei and Wegkamp 2006; Bartlett and Wegkamp 2008; Yuan and Wegkamp 2010); Zhang, Wang, and Qiao (2017) has studied the multicategory case. The conformal learning inference aims to find a set-valued prediction for each new observation to guarantee the probability that the prediction set contains its true class label (Shafer and Vovk 2008; Lei, Robins, and Wasserman 2013; Vovk et al. 2017; Lei et al. 2018). Both the approaches taken by Lei (2014) and Sadinle, Lei, and Wasserman (2017) may be viewed as special cases of conformal learning.

The rest of this article is organized as follows. Section 2 gives an overview of the underlying problems. Our main algorithm is introduced in Section 3, followed by a study of the theoretical properties in Section 4. Section 5 offers some numerical experiments. Concluding remarks are given in Section 6. Proofs are in the supplementary materials.

2. Background

Sadinle, Lei, and Wasserman (2017) extended the binary acceptance region learning problem (Lei 2014) to the multicategory case. Under this framework, one tries to balance the *efficiency* and *confidence* of acceptance regions. The *efficiency* can be measured by the *ambiguity*, defined as the expected cardinality of the set-valued prediction, $\mathsf{E}(|\phi(\boldsymbol{X})|)$, where $|\cdot|$ is the cardinality of a set. Note that this is the same as $\mathsf{E}(\sum_{j=1}^{k} \mathbb{1}[\boldsymbol{X} \in C_j])$, the expected number of acceptance regions that cover a random observation. The *confidence* refers to the requirement that each acceptance region C_j must cover at least $(1-\alpha_j)100\%$ of the population in class j, $\mathsf{P}_j(C_j) \geq 1-\alpha_j$, where $\mathsf{P}_j(\cdot) \triangleq \mathsf{P}(\cdot|Y=j)$ is the probability measure conditional on $Y = j$. Note that this constraint may be written as the class-specific classification accuracy guarantee for class j: $\mathsf{P}_j(Y \in \phi(\boldsymbol{X})) \geq 1-\alpha_j$. In summary, we minimize the ambiguity when maintaining the confidence by controlling the noncoverage rates (or the class-specific error rates),

$$\min_{\phi \in \boldsymbol{\Phi}} \quad \mathsf{E}(|\phi(\boldsymbol{X})|), \quad \text{subject to } 1 - \mathsf{P}_j(C_j) \leq \alpha_j,\ j \in \{1, \ldots, k\}. \tag{1}$$

Here α_j's are predetermined. For example, if one wants the set-valued classifier to correctly classify at least 95% of the population from class j, then she can set $\alpha_j = 0.05$.

Under certain continuity conditions and the assumption that $\mathsf{P}(Y = j) > 0$ for all j's, Sadinle, Lei, and Wasserman (2017) gave

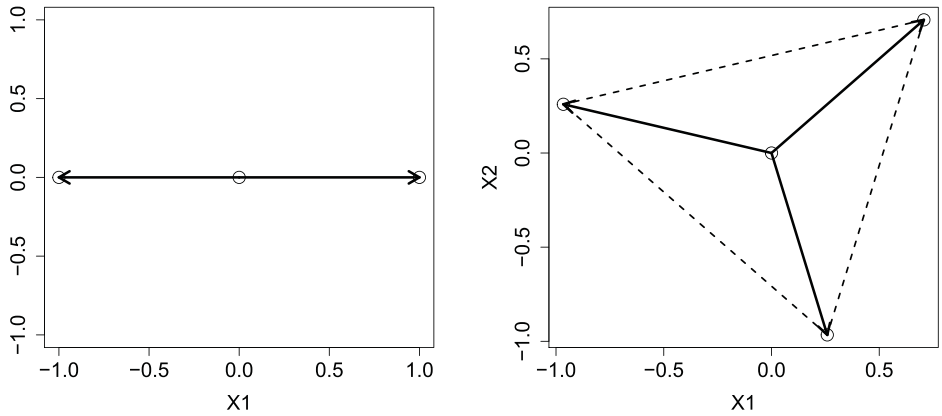

Figure 1. Configurations for $\boldsymbol{w}_j$'s. Left $k = 2$; right $k = 3$.

the following Bayes acceptance regions as solutions to problem (1).

Definition 1 (*Bayes acceptance regions*). Given α_j's, a solution to problem (1) is

$$C_j = \{\boldsymbol{x} : \eta_j(\boldsymbol{x}) \triangleq \mathsf{P}(Y = j \mid \boldsymbol{X} = \boldsymbol{x}) \geq t_j\},$$

where t_j is chosen to have $\mathsf{P}_j(C_j) = 1 - \alpha_j$.

Intuitively, each Bayes acceptance region contains all the observations for which the corresponding conditional class probability is large enough. In practice, Sadinle, Lei, and Wasserman (2017) suggested to employ the plug-in principle: first obtain $\widehat{\eta}_j$, the estimation of η_j, by methods such as the penalized logistic regression or k-nearest neighbors; then estimate t_j by the $\lfloor n_j\alpha_j \rfloor$th smallest value of $\{\widehat{\eta}(\boldsymbol{x}_{j,1}), \ldots, \widehat{\eta}(\boldsymbol{x}_{j,n_j})\}$, where $\boldsymbol{x}_{j,1}, \ldots, \boldsymbol{x}_{j,n_j}$ are training data from class j. As a result, the estimated acceptance regions take the form of $\widehat{C}_j = \{\boldsymbol{x} : \widehat{\eta}_j(\boldsymbol{x}) \geq \widehat{t}_j\}$.

3. Set-valued Multicategory Support Vector Machine

A fundamental challenge of the plug-in method is that in many contemporary data analyses, it is very difficult to estimate η_j at the first place. In this work, we propose to solve (1) directly via a risk minimization procedure avoiding estimating η_j. We introduce a general formulation in Section 3.1, and then focus on specifics in Sections 3.2 and 3.3.

3.1. Formulation

For a k-class problem, our set-valued classifier will be characterized by a vector-valued discriminant function $\boldsymbol{f} : \mathcal{X} \mapsto \mathbb{R}^{k-1}$ and a threshold $\varepsilon \in \mathbb{R}$. To obtain $\boldsymbol{f}$, we adopt the angle-based classification method (Zhang and Liu 2013), which has been shown to be very effective and computational efficient for large-scale multicategory classification in the high-dimensional space. We first define k unit vectors, $\boldsymbol{w}_j \in \mathbb{R}^{k-1}$, $j = 1, \ldots, k$, which form a regular simplex and sum to 0. Each vector represents a class and they are equiangular from one another. One possible configuration of $\boldsymbol{w}_j$'s is,

$$\boldsymbol{w}_j = \begin{cases} (k-1)^{-1/2}\mathbb{1} & j = 1, \\ -(1 + k^{1/2})/\{(k-1)^{3/2}\} & \\ \quad \mathbb{1} + \{k/(k-1)\}^{1/2}\boldsymbol{e}_{j-1} & 2 \leq j \leq k \end{cases}$$

where $\mathbb{1} \in \mathbb{R}^{k-1}$ is the vector of all ones and $\boldsymbol{e}_j \in \mathbb{R}^{k-1}$ is the vector of all zeros, with the jth element being 1. Figure 1 gives an illustration of this configuration for $k = 2$ and $k = 3$.

The angle margin, defined as $\langle \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{w}_j \rangle$, measures the proximity from $\boldsymbol{f}(\boldsymbol{x})$ to vector $\boldsymbol{w}_j$. A large angle margin indicates a small angle between vectors $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{w}_j$, and hence a close proximity between the observation $\boldsymbol{x}$ and class j. We will conduct the optimization with respect to $\boldsymbol{f}$ so that $\langle \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{w}_j \rangle$ is large for $j = y$ and small for $j \neq y$. Motivated by this intuition, we define the acceptance regions and the set-valued classifier to be,

$$\widehat{C}_j = \{\boldsymbol{x} : \langle \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{w}_j \rangle \geq -\varepsilon\}, \ \phi(\boldsymbol{x}) = \{j : \langle \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{w}_j \rangle \geq -\varepsilon\}. \quad (2)$$

Intuitively, the acceptance region for class j consists of all those observations whose $\boldsymbol{f}(\boldsymbol{x})$ are close enough to $\boldsymbol{w}_j$. Re-expressing (1) in terms of $\boldsymbol{f}$ and ε, we have

$$\begin{aligned} \min_{\varepsilon \geq 0, \boldsymbol{f} \in \mathcal{F}} \quad & R(\boldsymbol{f}, \varepsilon) \triangleq \mathsf{E}\left(\sum_{j=1}^{k} \mathbb{1}\left\{ \langle \boldsymbol{f}(\boldsymbol{X}), \boldsymbol{w}_j \rangle \geq -\varepsilon \right\} \right), \quad (3) \\ \text{subject to} \quad & \mathsf{P}_j(\langle \boldsymbol{f}(\boldsymbol{X}), \boldsymbol{w}_j \rangle < -\varepsilon) \leq \alpha_j, \quad j = 1, \ldots, k. \end{aligned}$$

When the constraints attain equality at the minimizer, one can show that the minimizer coincides with the solution to the following modified optimization,

$$\begin{aligned} \min_{\varepsilon \geq 0, \boldsymbol{f} \in \mathcal{F}} \quad & \bar{R}(\boldsymbol{f}, \varepsilon) \triangleq \mathsf{E}\left(\sum_{j \neq Y} \mathbb{1}\left\{ \langle \boldsymbol{f}(\boldsymbol{X}), \boldsymbol{w}_j \rangle \geq -\varepsilon \right\} \right), \quad (4) \\ \text{subject to} \quad & \mathsf{P}_j(\langle \boldsymbol{f}(\boldsymbol{X}), \boldsymbol{w}_j \rangle < -\varepsilon) \leq \alpha_j, \quad j = 1, \ldots, k. \end{aligned}$$

Since $\boldsymbol{w}_j$'s sum to 0, we have that $\sum_{j=1}^{k} \langle \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{w}_j \rangle = 0$ for any $\boldsymbol{x}$. In this case, requiring $\varepsilon \geq 0$ implies that $\langle \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{w}_j \rangle \geq -\varepsilon$ for at least one j, and hence $|\phi(\boldsymbol{x})| \geq 1$. There has been previous work (Hechtlinger, Póczos, and Wasserman 2018; Guan and Tibshirani 2019) in which $\phi(\boldsymbol{x}) = \emptyset$ may occur for some $\boldsymbol{x}$, implying that the class label for $\boldsymbol{x}$ has never been seen before.

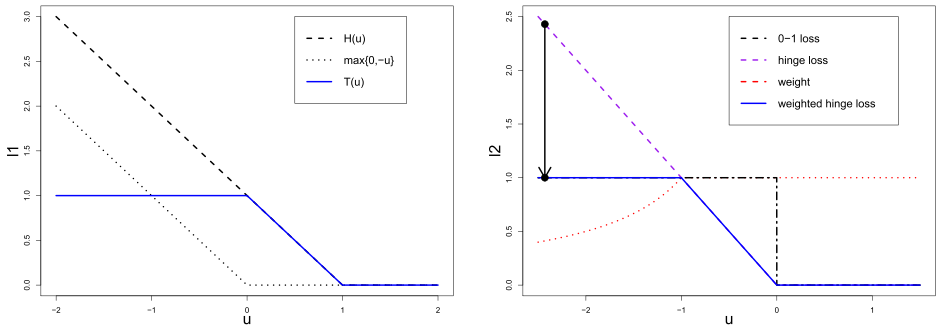

Figure 2. The left panel illustrates the truncated hinge loss. The right penal illustrates the proposed weight function and the resulting weighted hinge loss.

We do not consider this setting in this article: specifically, we assume that $\{1, 2, \ldots, k\}$ are the only possible classes in the test data.

In practice, the indicator functions in both the objective and in the constraints of (4) may cause difficulties for numerical optimizations (Hoffgen, Simon, and Vanhorn 1995). A common practice is to replace the indicator function in the objective by a convex surrogate loss. Moreover, a stream of work on NP classification (Rigollet and Tong 2011) also suggests to use a surrogate loss to bound the noncoverage rates such as the one in the constraints of (4). In general, it can be any decreasing surrogate loss used in the literature. Let ℓ_1 and ℓ_2 be the surrogate losses to be deployed in the objective and in the constraints, respectively. Our proposed set-valued classifier can be obtained by the following optimization,

$$\min_{\varepsilon\geq 0, \boldsymbol{f}\in\mathcal{F}} \quad R_{\ell_1}(\boldsymbol{f},\varepsilon) \triangleq \mathsf{E}\left(\sum_{j\neq Y}\ell_1\Big(-(\langle \boldsymbol{f}(\boldsymbol{X}),\boldsymbol{w}_j\rangle+\varepsilon)\Big)\right), \tag{5}$$

$$\text{subject to} \quad \mathsf{E}\Big[\ell_2(\langle \boldsymbol{f}(\boldsymbol{X}),\boldsymbol{w}_j\rangle+\varepsilon)\mid Y=j\Big]\leq\alpha_j, \quad j=1,\ldots,k.$$

Conceptually, the value $(\langle \boldsymbol{f}(\boldsymbol{x}),\boldsymbol{w}_j\rangle+\varepsilon)$ in the argument of ℓ_1 in the objective measures the closeness between observation $\boldsymbol{X}$ and the jth acceptance region (the larger the closer). Minimization of the objective leads to small values of $(\langle \boldsymbol{f}(\boldsymbol{x}),\boldsymbol{w}_j\rangle+\varepsilon)$ for $j\neq y$ and a large value for $j=y$. When ℓ_1 is the hinge loss, the new objective function resembles the loss function in multicategory SVM (Lee, Lin, and Wahba 2004), except for the important absence of the sum-to-zero constraint from our work, thanks to the use of the angle-based framework. In practice, given training data $\{(\boldsymbol{x}_i, y_i), i=1,\ldots,n\}$, one solves the empirical version of (5),

$$\min_{\varepsilon\geq 0,\boldsymbol{f}} \quad \frac{1}{n}\sum_{i=1}^{n}\sum_{j\neq y_i}\ell_1\Big(-(\langle \boldsymbol{f}(\boldsymbol{x}_i),\boldsymbol{w}_j\rangle+\varepsilon)\Big) \tag{6}$$

$$\text{subject to} \quad \frac{1}{n_j}\sum_{y_i=j}\ell_2(\langle \boldsymbol{f}(\boldsymbol{x}_i),\boldsymbol{w}_j\rangle+\varepsilon)\leq\alpha_j, \quad j=1,\ldots,k, \quad J(\boldsymbol{f})\leq s,$$

where n_j is the subsample size for class j and $J(\boldsymbol{f}) \leq s$ is a regulatory constraint added to make the solutions identifiable.

3.2. Choices of Surrogate Loss in Objective and Constraints

The choice of the surrogate losses is an important issue. Ideally, the surrogate loss ℓ_1 should enjoy the Fisher consistency property; on the other hand, an appropriately chosen ℓ_2 should guarantee that each acceptance region cover each class with at least the promised rate.

We propose to use a truncated hinge loss for ℓ_1 to achieve the Fisher consistency. Define the hinge loss as $H(u) = (1-u)_+$ and the truncated hinge loss as $T(u) = (1-u)_+ - (-u)_+$, where $(a)_+ = \max\{a, 0\}$. The latter loss truncates the conventional hinge loss to have a height not exceeding 1. The blue solid line in the left panel of Figure 2 gives an illustration of this truncated loss, which can be regarded as the difference of two hinge-type loss functions (the dashed and dotted lines). Theorem 1 shows that with the truncated hinge loss, our proposed method is Fisher consistent.

With a truncated loss, the resulting optimization is not convex due to the non-convexity of T. However, one can use the difference of convex function (DC) algorithm (Le Thi Hoai and Tao 1997; Wu and Liu 2007). A brief description of this algorithm is shown below.

Algorithm 1 (DC algorithm). To minimize $Q(\Theta) = Q_{\text{vex}}(\Theta) + Q_{\text{cav}}(\Theta)$, do the following:
1. Initialize Θ with Θ_0.
2. Repeat $\Theta_{t+1} = \text{argmin}_\Theta\,(Q_{\text{vex}}(\Theta) + \langle Q'_{\text{cav}}(\Theta), \Theta - \Theta_t\rangle)$ until convergence of Θ_t.

In (6), $Q_{\text{vex}} = \sum_{j\neq Y} H(-(\langle \boldsymbol{f}(\boldsymbol{X}),\boldsymbol{w}_j\rangle+\varepsilon))$, $Q_{\text{cav}} = -\sum_{j\neq Y} H(1-(\langle \boldsymbol{f}(\boldsymbol{X}),\boldsymbol{w}_j\rangle+\varepsilon))$. Θ stands for the parameters in $\boldsymbol{f}$ and ε. This algorithm is an example of the Majorize-Minimization (MM) algorithm as we replace Q_{cav} by its affine approximation in each iteration (Hunter and Lange 2004). The DC algorithm was used by Wu and Liu (2007) to build a Fisher consistent robust multicategory SVM.

Next we discuss the loss function in the constraints. We aim to bound the empirical noncoverage rate $(1/n_j)\sum_{j=y_i}\mathbb{1}\big[\langle \boldsymbol{f}(\boldsymbol{x}_i),\boldsymbol{w}_j\rangle < -\varepsilon\big]$ through bounding the empirical risk under the surrogate loss, $(1/n_j)\sum_{j=y_i}\ell_2(\langle \boldsymbol{f}(\boldsymbol{x}_i),\boldsymbol{w}_j\rangle+\varepsilon)$. The hinge loss may not be ideal for this purpose because it may have a much greater value than the indicator $\mathbb{1}[u<0]$, deteriorating the performance. For example, an observation with a very small functional margin $\langle \boldsymbol{f}(\boldsymbol{x}_i),\boldsymbol{w}_j\rangle \ll 0$ will give a large hinge loss and make the left-hand-side of the inequality $(1/n_j)\sum_{j=y_i}\ell_2(\langle \boldsymbol{f}(\boldsymbol{x}_i),\boldsymbol{w}_j\rangle+\varepsilon) \leq \alpha_j$ to be very close to or even exceed the right-hand-side, even though it is associated with only one instance of noncoverage. In general, using the hinge loss to bound the noncoverage in the constraints will lead to *overly conservative* solution (set-valued classifiers

being too ambiguous). A potentially useful alternative is the truncated hinge loss, $\min\{1, H(\langle \boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{w}_{y_i}\rangle + \varepsilon)\}$. However, the use of the (nonconvex) truncated hinge loss will add to another layer of computational challenge. To mimic the truncated hinge loss, we propose to combine the hinge loss with an adaptive weight in an iterative algorithm to alleviate this issue. Observations are assigned with weights, chosen to be $w_i = \max\{1, H(\langle \boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{w}_{y_i}\rangle + \varepsilon)\}^{-1}$ based on the solution to $(\boldsymbol{f}, \varepsilon)$ from the previous iteration, which, when multiplied by the hinge loss, resembles the truncated hinge loss. See the right panel of Figure 2 for an illustration: the blue bold line stands for the weighted hinge loss, which is the result of multiplying the weight (red dotted) by the hinge loss (purple dashed); the weighted hinge loss is close to the indicator function (black two-dashed).

Our proposed set-valued SVM (SSVM) is $\phi(\boldsymbol{x}) \triangleq \phi_{(\boldsymbol{f},\varepsilon)}(\boldsymbol{x}) = \{j : \langle \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{w}_j\rangle \geq -\varepsilon\}$, where $(\boldsymbol{f}, \varepsilon)$ is the final solution in an iterative algorithm. In each iteration, we solve

$$\begin{aligned}(\boldsymbol{f}, \varepsilon) \in \underset{\boldsymbol{f}, \varepsilon \geq 0}{\operatorname{argmin}} \quad & \frac{1}{n}\sum_{i=1}^{n}\sum_{j\neq y_i} T(-(\langle \boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{w}_j\rangle + \varepsilon)) \qquad (7)\\ \text{subject to} \quad & \frac{1}{n_j}\sum_{y_i=j} w_i H(\langle \boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{w}_j\rangle + \varepsilon) \leq \alpha_j,\\ & j = 1, \ldots, k, \quad J(\boldsymbol{f}) \leq s,\end{aligned}$$

given the weights. In the initial step, $w_i \equiv 1$ for all i. In the subsequent steps, we define

$$w_i = \max\{1, H(\langle \boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{w}_j y_i\rangle + \varepsilon)\}^{-1}$$

given $(\boldsymbol{f}, \varepsilon)$ from the previous step. The algorithm stops when the solution converges or the number of iterations has reached a preset maximum. Though there is no theoretical guarantee on the convergence of this iterative algorithm, in our numerical studies, it often converges after two or three iterations. Wu and Liu (2013) used a similar iterative idea in their adaptive weighted large margin classifiers for the purpose of robust classification.

3.3. Implementation Algorithms

In this section we discuss the implementation algorithm, to be used in the numerical experiments in Section 5. For computational convenience, we move the constraint $J(\boldsymbol{f}) \leq s$ to the objective as an additional regularization term, and obtain

$$\begin{aligned}(\boldsymbol{f}, \varepsilon) \in \underset{\boldsymbol{f}, \varepsilon \geq 0}{\operatorname{argmin}} \quad & \frac{1}{n}\sum_{i=1}^{n}\sum_{j\neq y_i} T(-(\langle \boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{w}_j\rangle) + \varepsilon) + \lambda J(\boldsymbol{f}). \qquad (8)\\ \text{subject to} \quad & \frac{1}{n_j}\sum_{y_i=j} w_i H(\langle \boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{w}_j\rangle + \varepsilon) \leq \alpha_j,\\ & j = 1, \ldots, k.\end{aligned}$$

We will consider both linear and kernel learning. In linear learning, let $\boldsymbol{f}(\boldsymbol{x}) = \mathbf{B}^T\boldsymbol{x} + \boldsymbol{v}$ where $\mathbf{B} = [\beta_{r,q}]$ is a $p \times (k-1)$ matrix of coefficients and $\boldsymbol{v}$ is a $(k-1) \times 1$ vector of intercepts. The regularization $J(\boldsymbol{f})$ is defined as the squared Frobenius norm of $\mathbf{B}$, $J(\boldsymbol{f}) = \sum_{r=1}^{p}\sum_{q=1}^{k-1}\beta_{r,q}^2$. The parameter $\Theta = (vec(\mathbf{B}), \boldsymbol{v}^T, \varepsilon)$, with the vectorized $\mathbf{B}$, $vec(\mathbf{B}) = [\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T, \ldots, \boldsymbol{\beta}_{k-1}^T]$, where $\boldsymbol{\beta}_q$ is the qth column of $\mathbf{B}$. Following the standard routine in the SVM literature, we introduce slack variables $\xi_{i,j}$ for the hinge-like functions in the objective function, and slack variables $\eta_{i,j}$ for the hinge-like function in the constraints. The entire algorithm entails two loops. In the outer loop, we update the weight for the constraints; in the inner loop, we use the DC algorithm given a fixed value of the weight. At each iteration of the DC algorithm, we aim to solve

$$\begin{aligned}\min_{\varepsilon, \boldsymbol{v}, \mathbf{B}, \{\xi_i\}_{i,j}, \{\eta_i\}_{i,j}} \quad & \frac{1}{2}\sum_{q=1}^{k-1}\boldsymbol{\beta}_q^T\boldsymbol{\beta}_q + C\sum_{i=1}^{n}\sum_{j\neq y_i}\xi_{i,j} \qquad (9)\\ & - \sum_{q=1}^{k-1}\sum_{r=1}^{p} c_{r,q}\beta_{r,q} - \sum_{q=1}^{k-1} c_q b_q - c\varepsilon,\\ \text{subject to} \quad & \xi_{i,j} \geq 1 + \varepsilon + \langle \mathbf{B}^T\boldsymbol{x}_i + \boldsymbol{v}, \boldsymbol{w}_j\rangle, \text{ for all } j \neq y_i,\\ & \eta_{i,j} \geq 1 - \varepsilon - \langle \mathbf{B}^T\boldsymbol{x}_i + \boldsymbol{v}, \boldsymbol{w}_j\rangle, \text{ for all } j = y_i,\\ & \xi_{i,j} \geq 0, \quad \eta_{i,j} \geq 0, \quad \varepsilon \geq 0\\ & \sum_{y_i=j} w_i\eta_{i,j} \leq n_j\alpha_j, \text{ for all } j,\end{aligned}$$

where $C = (2n\lambda)^{-1}$, and $c_{r,q}$, c_q and c are sub-gradients of $\sum_{i=1}^{n}\sum_{j\neq y_i} H(-\langle \boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{w}_j\rangle - (\varepsilon - 1))$ with respect to $\beta_{r,q}$, b_q and ε evaluated at the parameter set value from the previous iteration. Problem (9) is equivalent to the following dual problem,

$$\begin{aligned}\min_{\mathbf{Z}, \{\delta_j\}_{j=1}^{k}} \quad & \frac{1}{2}\sum_{i=1}^{n}\sum_{i'=1}^{n}((\mathbf{Z}\circ\mathbf{Y})\mathbf{W}^T\mathbf{W}(\mathbf{Z}\circ\mathbf{Y})^T)_{i,i'}\langle \boldsymbol{x}_i, \boldsymbol{x}_{i'}\rangle\\ & - \sum_{i=1}^{n}\sum_{j=1}^{k}(1 - \boldsymbol{x}_i^T\mathbf{C}\boldsymbol{w}_j Y_{i,j})\zeta_{i,j} + \sum_{j=1}^{k}\delta_j\theta_j\\ \text{subject to} \quad & 0 \leq \zeta_{i,j} \leq C, \text{ for all } y_i \neq j, \qquad (10)\\ & 0 \leq \zeta_{i,j} \leq \delta_j w_i, \text{ for all } y_i = j,\\ & \sum_{i=1}^{n}\sum_{j=1}^{k}\zeta_{i,j}Y_{i,j}w_{j,q} + c_q = 0, \ q = 1, \ldots, k-1,\\ & \sum_{i=1}^{n}\sum_{j=1}^{k}\zeta_{i,j}Y_{i,j} + c \leq 0,\end{aligned}$$

where $\circ$ stands for the Hadamard product, $\mathbf{Z} = [\zeta_{i,j}]$ is a $n \times k$ matrix, $\mathbf{W} = [w_{i,j}]$ is a $(k-1)\times k$ matrix with jth column $\boldsymbol{w}_j$, $\mathbf{Y} = [Y_{i,j}]$ is a $n \times k$ matrix and $Y_{i,j} = \mathbb{1}\{j = y_i\} - \mathbb{1}\{j \neq y_i\}$, $\theta_j = n_j\alpha_j$ and $\mathbf{C} = [c_{i,j}]$ is a $p \times (k-1)$ matrix. This dual problem can be solved by many standard off-the-shelf quadratic programming routines. After obtaining the solution to $\mathbf{Z}$, denoted as $\widehat{\mathbf{Z}}$, we have $\widehat{\mathbf{B}} = \boldsymbol{X}^T(\widehat{\mathbf{Z}}\circ\mathbf{Y})\mathbf{W}^T + \mathbf{C}$. Then we can plug $\widehat{\mathbf{B}}$ back into (9), which becomes a linear programming problem for $\boldsymbol{v}$ and ε, solvable by standard routines.

The kernel trick is often used in SVM like (7) to allow nonlinear classifiers. In the proposed method, $\boldsymbol{f}$ is a vector of nonlinear functions $(f_q)_{q=1}^{k-1}$ where f_q's belong to the same reproducing kernel Hilbert space (RKHS) with respect to a positive definite kernel function $K(\cdot,\cdot)$. By the representer theorem (Kimeldorf and Wahba 1970), we can focus on functions with form $f_q(\boldsymbol{x}) =$

$\sum_{r=1}^{n} \beta_{r,q} K(\boldsymbol{x}_r, \boldsymbol{x}) + b_q$ and the coefficients matrix B now become $n \times (k-1)$. Then the dual problem at each iteration of the DC algorithm becomes,

$$\begin{aligned} \min_{\mathbf{Z},\{\delta_j\}_{j=1}^k} \; & \frac{1}{2} \sum_{i=1}^{n} \sum_{r=1}^{n} ((\mathbf{Z} \circ \mathbf{Y}) \mathbf{W}^T \mathbf{W} (\mathbf{Z} \circ \mathbf{Y})^T)_{r,i} K(\boldsymbol{x}_r, \boldsymbol{x}_i) \\ & - \sum_{i=1}^{n} \sum_{j=1}^{k} (1 - (\mathbf{K}\widetilde{\mathbf{C}})_{i,\cdot} \boldsymbol{w}_j Y_{i,j}) \zeta_{i,j} + \sum_{j=1}^{k} \delta_j \theta_j, \\ \text{subject to} \quad & 0 \le \zeta_{i,j} \le C, \text{ for all } y_i \ne j, \\ & 0 \le \zeta_{i,j} \le \delta_j w_i, \text{ for all } y_i = j, \\ & \sum_{i=1}^{n} \sum_{j=1}^{k} \zeta_{i,j} Y_{i,j} w_{j,q} + c_q = 0, \; q = 1, \ldots, k-1, \\ & \sum_{i=1}^{n} \sum_{j=1}^{k} \zeta_{i,j} Y_{i,j} + c \le 0. \end{aligned} \tag{11}$$

After the solution to (11) is found, we have $\widehat{\mathbf{B}} = (\widehat{\mathbf{Z}} \circ \mathbf{Y}) \mathbf{W}^T + \widetilde{\mathbf{C}}$. Here $\widetilde{\mathbf{C}} = [\widetilde{c}_{i,q}]$ is a $n \times (k-1)$ matrix whose (i,q)th term $\widetilde{c}_{i,q}$ is the sub-gradient of $\sum_{i=1}^{n} \sum_{j \ne y_i} H(-\langle \boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{w}_j \rangle - (\varepsilon - 1))$ with respect to $f_q(\boldsymbol{x}_i)$ at the last iteration Θ_{t-1}. $\mathbf{K} = [K(\boldsymbol{x}_i, \boldsymbol{x}_j)]$ is a $n \times n$ matrix whose entries are the kernel function K evaluated on pairs of observations from the data set.

3.4. Relation with Classification using Reject and Refine Options

Classification with reject and refine options (CRR) (Bartlett and Wegkamp 2008; Manwani et al. 2015; Zhang, Wang, and Qiao 2017) also allows set-valued classification. Bartlett and Wegkamp (2008) proposed a Fisher consistent surrogate loss in the binary case. Zhang, Wang, and Qiao (2017) extended CRR to the multicategory case and introduced the refined option, which allows a set-valued prediction whose size is between 1 and k.

Typically CRR classifiers aim to balance the cost of misclassification and the cost of rejection. Some CRR work uses a weighted combination of both costs in the objective function; others consider minimizing the misclassification rate, subject to a budget of rejections. There is an underlying connection between the level of rejection allowed in CRR and the confidence achieved in a set-valued classifier. Though CRR may lead to set-valued predictions, the notion of confidence is not explicitly accounted for in the algorithm. The main motivation of the current work is precise quantifications of the confidence (or class-specific accuracy) of the set-valued classifier. To this end, one may view CRR and the set-valued classification as dual problems to each other.

4. Theoretical Studies

In this section, we first study the Fisher consistency in the set-valued classification setting. Then we bound the excess ambiguity by the excess surrogate ambiguity, in parallel to the excess risk bound seen in Bartlett, Jordan, and McAuliffe (2006). Lastly, we study finite sample bounds for the noncoverage rate and the excess ambiguity.

4.1. Fisher Consistency and Excess Risk Bound

We follow the same assumptions in Sadinle, Lei, and Wasserman (2017). Assume the underlying distribution $P(\boldsymbol{X}, Y)$ is absolute continuous with respect to $\nu_X \times \nu_Y$, where ν_X is the Lebesgue measure in $\mathbb{R}^p$ and ν_Y is the counting measure on $\{1, \ldots, k\}$. Moreover, assume p_j, the density function of the distribution of $\boldsymbol{X}$ conditional on $Y = j$, is positive on $\mathcal{X}$. Let $\pi_j = \mathsf{P}(Y = j)$ be the prior probability of class j and assume $\pi_j > 0$. In addition, we assume $\eta_j(\boldsymbol{X})$ is a continuous random variable with $\mathsf{P}(\eta_i(\boldsymbol{X}) = \eta_j(\boldsymbol{X})) = 0$ for all pairs $i \ne j$.

Our first main result is the Fisher consistency of the surrogate function, which suggests that the population minimizer of the surrogate loss function coincides with the Bayes solution given in Sadinle, Lei, and Wasserman (2017), with constraints that $\mathsf{P}_j(\langle \boldsymbol{f}(\boldsymbol{X}), \boldsymbol{w}_j \rangle < -\varepsilon) \le \alpha_j$, $j = 1, \ldots, k$, which correspond to the optimization,

$$\begin{aligned} \min_{\boldsymbol{f}} \quad & R_T(\boldsymbol{f}, \varepsilon) \triangleq \mathsf{E}\Big(\sum_{j \ne Y} T(-(\langle \boldsymbol{f}(\boldsymbol{X}), \boldsymbol{w}_j \rangle + \varepsilon))\Big) \\ \text{subject to} \quad & \mathsf{P}_j(\langle \boldsymbol{f}(\boldsymbol{X}), \boldsymbol{w}_j \rangle < -\varepsilon) \le \alpha_j, \quad j = 1, \ldots, k. \end{aligned} \tag{12}$$

One subtlety here is that the true Bayes solution may involve null set, that is, the union of all acceptance regions in the Bayes solution $\cup_j C_j^*$ may not cover the whole feature space $\mathcal{X}$, or equivalently, $\phi^*(\boldsymbol{x})$ may be empty for some observation $\boldsymbol{x}$. This may happen for relatively easy classification tasks in which data points from different classes are far away from each; this may also happen when the noncoverage rates α_j's are chosen to be large so that the acceptance regions are relatively small. Note that in these cases, set-valued classification methods becomes less relevant since a traditional classification method can meet the needs and perform just as well. Hence, to show Fisher consistency of the proposed method in settings relevant to set-valued classification, we consider the following assumption.

Assumption 1. Given α_j, $j = 1, \ldots, k$, assume that $\cup_j \{\boldsymbol{x} : \eta_j(\boldsymbol{x}) \ge t_j\} = \mathcal{X}$, where t_j satisfies $\mathsf{P}_j(\eta_j(\boldsymbol{X}) \ge t_j) = 1 - \alpha_j$.

Under this assumption, the Bayes acceptance regions in Sadinle, Lei, and Wasserman (2017), as given in Definition 1, satisfy $\cup_{j=1}^k C_j = \mathcal{X}$.

Theorem 1. Under Assumption 1, for a fixed $\varepsilon \ge 0$, let $\mathcal{F}^*$ be the class of functions that solve (12). Then any $\boldsymbol{f}^* \in \mathcal{F}^*$ satisfies that

$$\begin{cases} \langle \boldsymbol{f}^*(\boldsymbol{x}), \boldsymbol{w}_j \rangle \ge -\varepsilon, & \text{if } \eta(\boldsymbol{x}) \ge t_j \\ \langle \boldsymbol{f}^*(\boldsymbol{x}), \boldsymbol{w}_j \rangle \le -(1+\varepsilon), & \text{if } \eta(\boldsymbol{x}) < t_j \end{cases} \tag{13}$$

almost surely, where t_j satisfies $\mathsf{P}_j(\eta_j(\boldsymbol{X}) \ge t_j) = 1 - \alpha_j$. Hence, $\phi_{(\boldsymbol{f}^*, \varepsilon)}$ is equivalent to the Bayes acceptance regions in Definition 1, that is, the truncated hinge loss is Fisher consistent.

The next theorem provides a bound quantification of the excess risk (defined as the classification ambiguity) using the excess surrogate risk as assessed using the truncated hinge loss function. The same bound quantification framework was proposed by Bartlett, Jordan, and McAuliffe (2006) and used by Wang and Qiao (2018).

Theorem 2. Suppose there exists $c \in (0,1)$ such that for any $j \in \{1,\ldots,k\}$, $\{\boldsymbol{x} : \eta_j(\boldsymbol{x}) > 1-c\} \subset \{\boldsymbol{x} : \eta_j(\boldsymbol{x}) \geq t_j\}$, where t_j satisfies $\mathsf{P}_j(\eta_j(\boldsymbol{X}) \geq t_j) = 1-\alpha_j$. For a fixed threshold ε, let $\widehat{\boldsymbol{f}}$ be another function that satisfies the coverage rate constraints in (12). Under Assumption 1, we have

$$\frac{1}{c}(R_T(\widehat{\boldsymbol{f}},\varepsilon) - R_T(\boldsymbol{f}^*,\varepsilon)) \geq R(\widehat{\boldsymbol{f}},\varepsilon) - R(\boldsymbol{f}^*,\varepsilon).$$

Although Theorem 2 is proved given a fixed ε, the bound does not depend on ε.

4.2. Finite-Sample Properties for Error Rates and the Ambiguity

In this section, we discuss two properties of the proposed set-valued classifier (6) based on a finite sample. Our discussion focuses on kernel learning given a set of nonstochastic weights in the constraint. To simplify the theoretical analysis, we consider the case of equal weights, and assume that the sample size for each class is non-stochastic (i.e., they are fixed). In particular, instead of sampling n points directly from $P(\boldsymbol{X}, Y)$, we choose the sample size for each class and then sample from each subpopulation. The theorems in this section can be extended to unequal weights or stochastic weights with the assumption that $\pi_j > 0$.

Let $\mathcal{H}_K(s) = \{\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^{k-1} \mid \sum_{q=1}^{k-1} \|f_q\|^2_{\mathcal{H}_K} \leq s\}$, the reproducing kernel Hilbert space (RKHS) for $(k-1)$ dimensional vector-valued functions with norm bounded by s. Here K is a positive definite kernel function which induces $\mathcal{H}_K$ and we assume that $\sup_{\boldsymbol{x}} K(\boldsymbol{x},\boldsymbol{x}) \leq r$.

Theorem 3. Given the training data $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{n_j} | Y = j\}$ from the jth class, and a fixed $\varepsilon \geq 0$, any function $\boldsymbol{f} \in \mathcal{H}_K(s)$ uniformly satisfies that,

$$\mathsf{E}[\ell(\langle \boldsymbol{f}(\boldsymbol{X}), \boldsymbol{w}_j\rangle + \varepsilon) \mid Y = j] \leq \frac{1}{n_j}\sum_{y_i=j} \ell(\langle \boldsymbol{f}(\boldsymbol{X}_i), \boldsymbol{w}_j\rangle + \varepsilon) + 3T_{n_j}(\zeta) + Z(n_j)$$

for any j, with probability at least $1 - k\zeta$ (the probability is with respect to the distribution of the training data.) Here $Z(n) = \sqrt{sr(k-1)}/\sqrt{n}$, $T_n(\zeta) = \{2sr(k-1)\log(1/\zeta)n^{-1}\}^{1/2}$, the expectation on the left-hand side is with respect to a test observation $(\boldsymbol{X}, Y)$, and $\ell(\cdot)$ can be either the hinge loss $H(\cdot)$ or the truncated hinge loss $T(\cdot)$.

Note that the left-hand side of the inequality in Theorem 3 satisfies

$$\mathsf{P}_j(\langle \boldsymbol{f}(\boldsymbol{X}), \boldsymbol{w}_j\rangle < -\varepsilon) \leq \mathsf{E}[\ell(\langle \boldsymbol{f}(\boldsymbol{X}), \boldsymbol{w}_j\rangle + \varepsilon) \mid Y = j]$$

due to the definition of $\ell(\cdot)$. Together with this observation, Theorem 3 suggests a way to control the noncoverage rate for each class at a desirable level, say, α_j. To this end, one should identify a data-dependent function $\widehat{\boldsymbol{f}}$, by solving (7) and searching for tuning parameter properly, so that

$$\frac{1}{n_j}\sum_{y_i=j} \ell(\langle \widehat{\boldsymbol{f}}(\boldsymbol{x}_i), \boldsymbol{w}_j\rangle + \varepsilon) \leq \alpha_j - 3T_{n_j}(\zeta) - Z(n_j).$$

This amounts to setting the right-hand side of the constraint in (7) to be slightly smaller than the desired level α_j; after replacing $\boldsymbol{f}$ in the inequality in Theorem 3 by $\widehat{\boldsymbol{f}}$, we can see that the left-hand side of the inequality, and hence $\mathsf{P}_j(\langle \widehat{\boldsymbol{f}}(\boldsymbol{X}), \boldsymbol{w}_j\rangle < -\varepsilon \mid \mathcal{D})$, is bounded by α_j. Note that the remainder terms $3T_{n_j}(\zeta) + Z(n_j)$ converges to 0 at the rate of $n_j^{-1/2}$.

By setting the arbitrary $\boldsymbol{f}$ to be the data-dependent $\widehat{\boldsymbol{f}}$, Theorem 3 implies the multiple-use validity in the sense of Dümbgen, Igl, and Munk (2008). Specifically, let $\widehat{C}_j$ be the acceptance region for class j induced by $\widehat{\boldsymbol{f}}$; we have that, with probability at least $1 - k\zeta$, $\mathsf{P}(\boldsymbol{X} \in \widehat{C}_j | Y = j, \mathcal{D}) \equiv \mathsf{P}_j(\langle \widehat{\boldsymbol{f}}(\boldsymbol{X}), \boldsymbol{w}_j\rangle \geq -\varepsilon \mid \mathcal{D}) \geq 1 - \alpha_j$, for each j. Hence, by making $\zeta \to 0$ as $n_j \to \infty$, we can obtain the multiple-use validity in Dümbgen, Igl, and Munk (2008), in principle. Note that in practice one may implement a different calibration method to select α_j; in our numerical studies, we use split-conformal calibrations for all methods to achieve fair comparisons.

We note that this convergence rate does not depend on the dimension of the data, although it does depend on the number of classes. In contrast, the estimation error of probability estimation could quickly diverge as the dimension increases, which may undermine the performance of plug-in based methods in high-dimensional settings.

The next theorem quantifies the excess T-ambiguity based on a finite sample. We define the function space with noncoverage rates bounded by α_j less a small term $\frac{\kappa}{\sqrt{n_j}}$ by

$$\mathcal{F}_\varepsilon(\boldsymbol{\alpha}, \kappa, s) = \{\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^{k-1} \mid \mathsf{E}(H(\langle \boldsymbol{f}(\boldsymbol{X}), \boldsymbol{w}_j\rangle + \varepsilon) \mid Y = j) \leq \alpha_j - \frac{\kappa}{\sqrt{n_j}}, \forall j\} \bigcap \mathcal{H}_K(s),$$

and its empirical version as

$$\widehat{\mathcal{F}}_\varepsilon(\boldsymbol{\alpha}, \kappa, s) = \{\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^{k-1} \mid \frac{1}{n_j}\sum_{i:y_i=j} H(\langle \boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{w}_j\rangle + \varepsilon) \leq \alpha_j - \frac{\kappa}{\sqrt{n_j}}, \forall j\} \bigcap \mathcal{H}_K(s).$$

Theorem 4. For a fixed ε, let $\widehat{\boldsymbol{f}}$ be a solution of the optimization problem

$$\min_{\boldsymbol{f} \in \widehat{\mathcal{F}}_\varepsilon(\boldsymbol{\alpha}, \kappa, s)} \sum_{i=1}^{n}\sum_{j \neq y_i} T(-(\langle \boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{w}_j\rangle + \varepsilon)) \tag{14}$$

with $\kappa = (2\sqrt{\log(\frac{1}{\zeta})} + 1)\sqrt{sr(k-1)}$. With probability $1 - 2k^2\zeta$, we have

(i) $\widehat{\boldsymbol{f}} \in \mathcal{F}_\varepsilon(\boldsymbol{\alpha}, 0, s)$.

(ii) $R_T(\widehat{\boldsymbol{f}}, \varepsilon) - \min_{\boldsymbol{f} \in \mathcal{F}_\varepsilon(\boldsymbol{\alpha}, 2\kappa, s)} R_T(\boldsymbol{f}, \varepsilon) \leq 2k\kappa(n^{-1/2})$.

Problem (14) is almost equivalent to (7), except that ε is fixed and the nominal noncoverage is set to be α_j less a small quantity $\frac{\kappa}{\sqrt{n_j}}$. Part (i) of Theorem 4 has a similar implication to Theorem 3: if one imposes a more stringent constraint (that is, $\boldsymbol{f} \in \widehat{\mathcal{F}}_\varepsilon(\boldsymbol{\alpha}, \kappa, s)$, or precisely speaking, $\frac{1}{n_j}\sum_{i:y_i=j} H(\langle \boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{w}_j\rangle + \varepsilon) \leq \alpha_j - \frac{\kappa}{\sqrt{n_j}}$, with the gap term $\frac{\kappa}{\sqrt{n_j}}$ vanishes as the sample size increases), then it is possible to make $\mathsf{E}(H(\langle \boldsymbol{f}(\boldsymbol{X}), \boldsymbol{w}_j\rangle + \varepsilon) \mid Y = j)$ bounded by the desired rate α_j. Part (ii) further

shows that the T-ambiguity of our proposed method based on a finite sample converges to the T-ambiguity of the theoretically optimal classifier that minimizes the T-ambiguity subject to the true noncoverage rate being bounded. The difference between the two (that is, the excess T-ambiguity) is at most $2k\kappa(n^{-1/2})$, which vanishes as the sample size grows, and does not depend on the dimension. Though both Theorems 3 and 4 are under a fixed ε which is usually unknown in real applications, the convergence rate does not depend on the value of ε. Hence, given a dataset, the proposed method can achieve at least the convergence rate shown in the theorems.

Remark 1. The convergence rate of the excess T-ambiguity for our proposed method is $O(n^{-1/2})$, whereas the plug-in method has a convergence rate of $O(\epsilon_n^{\gamma} + \log(n)n^{-1/2})$ for the excess ambiguity (Sadinle, Lei, and Wasserman 2017), where ϵ_n is related to the estimation error of the η_j functions and γ is the margin exponent in the low-noise margin condition of the underlying distribution for $\eta(\boldsymbol{X})$. While this is not an apples-to-apples comparison, our proposed method has a faster convergence rate, and does not require the estimation of η_j. From the methodological perspective our method does not require data splitting for the purpose of calibration as is required by the plug-in method. In practice, we recommend to use the proposed method when the dimension is high and sample size is limited; this is the scenario in which the plug-in method may have difficulty estimating η_j accurately.

5. Numerical Studies

In this section, we compare our confidence-based set-valued multicategory support vector machine (SSVM) method and various methods using the plug-in principle (Sadinle, Lei, and Wasserman 2017) on both simulated and real data. The baseline models include L_2 penalized logistic regression (Le Cessie and Van Houwelingen 1992; Zhang and Liu 2013), kernel logistic regression (Zhu and Hastie 2005), kNN (Altman 1992), random forest (Liaw and Wiener 2002) and MSVM (multicategory SVM) (Cortes and Vapnik 1995; Platt 1999; Lee, Lin, and Wahba 2004). MSVM does not directly provide an estimate of the probability, but provides a list of scores that preserve the order among the estimated probabilities. For the proposed SSVM model, we use the implementation that solves the optimization problem (8).

In the study, we use solver `Cplex` and `lpsolve` to solve the quadratic and linear programming problem arising in SSVM. For other methods, we use existing R packages `glmnet`, `gelnet`, `class`, `randomForest`, `e1071` and the solver provided in Lee, Lin, and Wahba (2004).

5.1. Simulations

We study the empirical performance of the proposed method over a variety of simulated data with different sample sizes. In each case, an independent tuning set with the same sample size as the training set is generated for parameter tuning. The test set has 10,000 observations for each class. We run the simulation 100 times and report the mean and standard error. Nominal noncoverage rates are set to be 0.05.

We select the tuning parameter $C = (2n\lambda)^{-1}$ and the hyperparameters in kernel learning for the proposed SSVM method as follows. We search for the optimal ρ in the Gaussian kernel $\exp(-\|x-y\|^2/\rho^2)$ from the grid $10^{\{-0.5,0.25,0,0.25,0.5\}}$ and the optimal degree for polynomial kernel from $\{2,3,4\}$. For each fixed candidate hyperparameter, we choose C from a grid of candidate values ranging from 10^{-4} to 10^2 by the following two-step searching scheme. We first do a rough search with a larger stride $\{10^{-4}, 10^{-3.5}, \ldots, 10^2\}$ and get the best parameter C_1. In the next step, we do a fine search from $C_1 \times \{10^{-0.5}, 10^{-0.4}, \ldots, 10^{0.5}\}$. After that, we choose the optimal pair which gives the smallest tuning ambiguity among those which have the tuning set noncoverage rates being smaller than or equal to the nominal rate α_j.

For the plug-in methods, we employ both one-versus-rest classification and multicategory classification to estimate the posterior probability η_j as done in Sadinle, Lei, and Wasserman (2017). In one-versus-rest classification, we train k separate classifiers to classify between class j and all the other classes. All k classifiers share the same tuning parameter. All the plug-in methods are tuned in the same way as SSVM, that is, choosing the tuning parameter(s) that minimizes the ambiguity among those which satisfy the nominal noncoverage rates. For logistic regression and SVM, we use the same grid as SSVM when grid-searching their tuning parameters. For random forest, we choose the best number of trees from {50, 100, 150,…, 300} and subsampling rate for the number of variables from {0.05, 0.1, 0.2,…, 0.8}. For kNN, we choose the best k from {6, 8,…, 40}.

To robustly control the error, we make use of the split-conformal inference approach (the so-called *robust implementation*) suggested in Lei (2014) for all the methods. We split the data into training and tuning sets. Using the training data, we first obtain an estimate of η_j (by methods such as logistic regression, kNN and random forest) or an monotone proxy of it so that the order is preserved (such as the scores in MSVM, and $\langle \boldsymbol{f}, \boldsymbol{w}_j \rangle$, the jth angle margin in SSVM). For each class j, we choose thresholds $\widehat{t}_j$ to be the $(\alpha_j \times 100)$th sample percentile of $\widehat{\eta}_j(\boldsymbol{x})$ among the tuning data in class j so that the noncoverage rates for the tuning set match the nominal rates. The estimated acceptance regions are defined as $\widehat{C}_j = \{\boldsymbol{x} : \widehat{\eta}_j(\boldsymbol{x}) \geq \widehat{t}_j\}$ and equivalently, the set-valued predictions $\phi(\boldsymbol{x}) = \{j : \widehat{\eta}_j(\boldsymbol{x}) \geq \widehat{t}_j\}$. Ideally, the plug-in procedure requires two extra datasets other than the training data: One is used to select thresholds $\widehat{t}_j$'s and the other is for hyperparameter tuning. However, to achieve fair comparison with the proposed method, we use the tuning set for both purposes. This method was introduced in Lei (2014) which works well in practice.

We include MSVM approaches whose discriminant functions are obtained either in the traditional one-versus-rest way or in the all-at-once multicategory (Lee, Lin, and Wahba 2004) way. We induce acceptance regions from MSVM by thresholding in the same way described above. It is well-known that SVM does not provide an accurate estimation of the posterior probabilities (Platt 1999). The comparison between these MSVM methods, not originally designed for set-valued classification, and our proposed method, highlights that even using robust

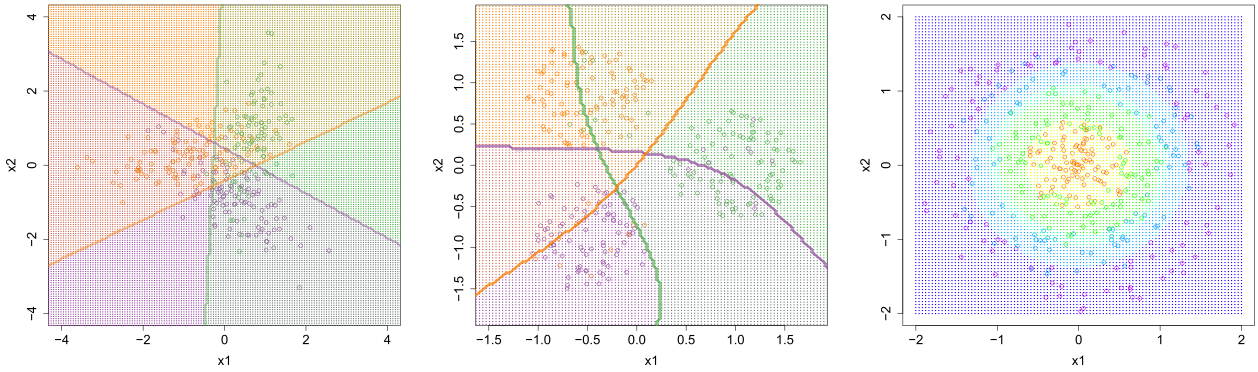

Figure 3. Scatterplots of the first two dimensions for the simulated data with different colors showing the overlapping acceptance regions suggested by the SSVM method.

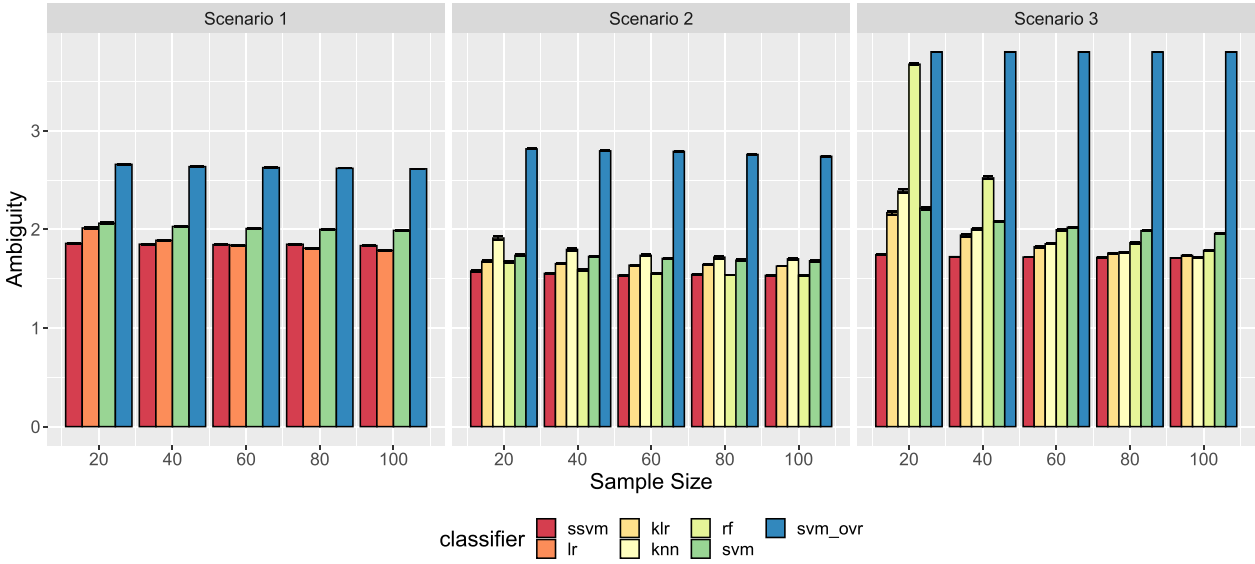

Figure 4. Empirical ambiguities in three settings. Empirical noncoverage rates are aligned among different methods and are not shown. SSVM has the smallest ambiguity.

implementation directly on either kind of MSVM methods will not provide a successful set-valued classifier; that is to say that the better performance of our method is attributed to factors beyond the use of the robust implementation scheme.

Because there are k noncoverage rates and one ambiguity, how to make fair comparisons between methods becomes a tricky problem since one method can have small test data ambiguity but higher test data noncoverage rates. It is unfair to claim that this method is better simply because it has a smaller test data ambiguity. To resolve this conflict, we further adjust the thresholds in each method after the initial training stage, so that the test data empirical noncoverage rates of all the methods are aligned with the nominal noncoverage rate. As a result, the noncoverage rates for almost all methods are the same so that we only need to compare them based on their test data ambiguity (kNN and random forest have slightly smaller noncoverage rates because there are many ties exactly at the threshold). Given the same noncoverage rate, a smaller ambiguity means the classifier performs better.

We consider three different simulation scenarios. In the first scenario we compare the linear approaches (SSVM, naive SVM and penalized logistic regression), while in the next two scenarios we consider nonlinear methods. In all cases, we add additional noisy dimensions to the data to test the robustness of all the methods. These noisy covariates are normally distributed with mean $\mathbf{0}$ and $\mathbf{\Sigma} = \text{diag}(1/p)$, where p is the total dimension of the data.

Example 1 (Linear model with nonlinear Bayes rule). In this scenario, we generate three normally distributed classes with different covariance matrices as shown in the left panel of Figure 3. In particular, we have $\boldsymbol{X} \mid Y = j \sim \mathcal{N}(\boldsymbol{\mu}_j, \mathbf{\Sigma}_j)$. Given $\boldsymbol{w}_4 := \boldsymbol{w}_1$, for $j = 1, 2, 3$, we have $\boldsymbol{\mu}_j = \|\boldsymbol{w}_j - \boldsymbol{w}_{j+1}\|_2^{-1}(\boldsymbol{w}_j - \boldsymbol{w}_{j+1})$, and $\mathbf{\Sigma}_j = \mathbf{S}_j \text{diag}(1, 0.2)\mathbf{S}_j^T$, with $\mathbf{S}_j = [\boldsymbol{\mu}_j, \widetilde{\boldsymbol{\mu}}_j]$ and $\widetilde{\boldsymbol{\mu}}_j = [-\mu_{j,2}, \mu_{j,1}]^T$. Here $\boldsymbol{w}_j$'s are those class representative vectors in angle-based learning. The prior probabilities of all classes are the same. Lastly, we add eight dimensions of noisy covariates to the data. We compare linear SSVM, and the plug-in methods based on L_2 penalized logistic regression and naive linear SVM.

Example 2 (Moderate dimensional uniform balls). We first generate a two-dimensional data uniformly distributed in three disks with radius 2/3 as shown in the middle panel of Figure 3. The centers of three disks are $c_1 = [1, 0]^T$, $c_2 = [\cos(\frac{2\pi}{3}), \sin(\frac{2\pi}{3})]^T$ and $c_3 = [\cos(\frac{4\pi}{3}), \sin(\frac{4\pi}{3})]^T$. Then we contaminate each disk by relabeling 10% of the observations within each class to a different class, so that the Bayes acceptance region should include the own disk for each class and one of the

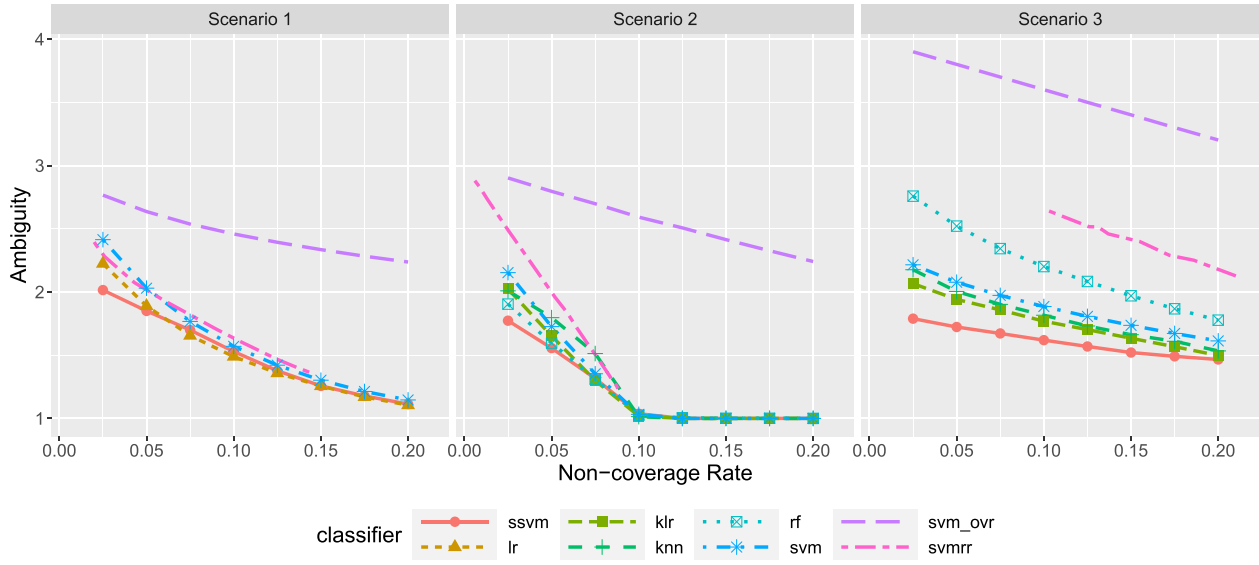

Figure 5. Ambiguity under different noncoverage rates. The advantage of the proposed method is more obvious when the noncoverage rates are small.

rest two disks. We then add 98 noisy covariates on top of the two-dimensional signal. We use the Gaussian kernel for all the kernel-based methods.

Example 3 (*High-dimensional donut*). We first generate data using radius-angle pairs (r_i, θ_i) where $\theta_i \sim \text{Unif}[0, 2\pi]$, and $r_i|Y = 1 \sim \text{Unif}[0, 0.65]$, $r_i|Y = 2 \sim \text{Unif}[0.45, 1.1]$, $r_i|Y = 3 \sim \text{Unif}[0.9, 1.55]$, $r_i|Y = 4 \sim \text{Unif}[1.35, 2]$. We define the two-dimensional $\boldsymbol{X}_i = (r_i \cos(\theta_i), r_i \sin(\theta_i))$ as shown in right panel of Figure 3. We then add 398 covariates on top of the two-dimensional signal. We use the polynomial kernel for all the kernel-based methods.

Simulation results are reported in Figure 4. In all three settings, the proposed method (denoted as "ssvm") outperforms all the plug-in methods when the number of observations are small and are comparable to the best plug-in method (logistic regression in Example 1, random forest in Example 2, and kNN in Example 3) when the sample size becomes large. The naive SVM method is significantly worse than the proposed methods in all scenarios. The noncoverage rates (not shown here) of SSVM, random forest, kernel logistic regression and naive SVM methods are close to 0.05 while kNN have a smaller noncoverage rates (due to such technicalities as too many ties of $\widehat{\eta}_j(\boldsymbol{x})$ exactly at the threshold.)

5.2. Accuracy and Ambiguity Tradeoff

Although 0.05 is a popular noncoverage rate in practice, it is of interest to study the trade-off between ambiguity and noncoverage rates as the noncoverage rate varies. In this section, we compare the proposed method with other plug-in methods as well as the CRR method (Zhang, Wang, and Qiao 2017) with various noncoverage rates. In particular, we studied which method has the smallest ambiguity under different noncoverage rates.

We fix the training sample size at 40 for each class and vary the noncoverage rates from 0.025 to 0.2 for SSVM and plug-in methods. We align the empirical noncoverage rates for SSVM and plug-in methods and compare their ambiguity as in the previous section. For the CRR classifier, we vary the reject price d and report the ambiguity and the average noncoverage rates over all the classes. The results are shown in Figure 5.

In Figure 5 (where "svmrr" stands for the SVM with reject and refine), we can see that the SSVM gives a much smaller ambiguity than the plug-in methods when the noncoverage rates are small. However, with the noncoverage rates grow, the gap between the proposed method and the plug-in methods become smaller. In the first example, SSVM is even outperformed by a certain plug-in method. This may not be surprising. One major advantage of the proposed method is to incorporate the noncoverage rate consideration into the risk minimization. In contrast, the discriminant functions of plug-in methods, such as the logistic regression, is not affected by the choice of the noncoverage rate. As a result, when the noncoverage rates are small, our proposed method will optimize its discriminant function to accommodate the noncoverage rates; when the noncoverage rates gradually grow larger, the effect of the noncoverage rate level becomes weaker and the gaps between the proposed method and the plug-in methods vanish. When the noncoverage rate is set to be a very large value, the coverage constraints are not active (that is, they no longer matter because they can be achieved by most classifiers easily) and therefore most of these methods perform similarly (as they do in the standard classification setting.)

5.3. Real Data Analysis

We study the performance of the proposed method on a few benchmark datasets. We compare the proposed method SSVM with L_2 penalized logistic regression, kernel logistic regression, kNN, random forest and MSVM. For the sake of brevity, we do not consider methods based on the One-versus-One or One-versus-Rest paradigm.

CNAE-9 Data: The CNAE-9 data (Ciarelli and Oliveira 2009) contains 1080 documents of free text business descriptions

Table 1. Rows annotated with $\widehat{\alpha}_j$ are empirical noncoverage rates given class.

CNAE data—$\alpha_j = 0.05$						
Classifier		SSVM	LR	*k*NN	RF	SVM-Naive
$\widehat{\alpha}_j$	$Y = 1$	2.900(.323)	2.050(.250)	4.200(.442)	3.950(.386)	4.325(.438)
	$Y = 2$	4.425(.448)	3.000(.354)	5.500(.517)	4.875(.434)	4.450(.420)
	$Y = 3$	4.925(.409)	4.425(.436)	4.275(.465)	4.725(.445)	4.300(.4258)
	$Y = 4$	6.525(.507)	4.975(.475)	5.450(.493)	5.675(.517)	4.975(.429)
	$Y = 5$	2.250(.237)	3.550(.324)	4.900(.448)	3.175(.329)	4.000(.348)
	$Y = 6$	5.075(.453)	4.425(.477)	5.600(.569)	6.400(.523)	4.075(.401)
	$Y = 7$	5.275(.456)	4.150(.444)	5.275(.543)	5.575(.474)	5.150(.518)
	$Y = 8$	3.325(.343)	3.675(.353)	3.950(.413)	3.950(.356)	3.800(.361)
	$Y = 9$	5.475(.474)	5.025(.507)	5.175(.583)	5.075(.499)	5.150(.471)
Ambiguity		1.280(.011)	1.414(.018)	2.373(.032)	1.456(.013)	1.813(.043)
Amb. Aligned		**1.426**(.020)	1.428(.020)	2.684(.050)	1.596(.020)	1.878(.042)
Zip code data—$\alpha_j = 0.01$						
Classifier		SSVM	LR	*k*NN	RF	SVM-Naive
$\widehat{\alpha}_j$	$Y = 0$	0.976(.088)	1.037(.088)	0.832(.081)	1.134(.099)	0.959(.082)
	$Y = 6$	1.233(.111)	1.303(.120)	0.926(.091)	1.263(.105)	1.229(.121)
	$Y = 8$	1.346(.108)	1.362(.120)	0.978(.089)	1.524(.112)	1.305(.102)
	$Y = 9$	0.890(.083)	1.060(.095)	0.791(.092)	0.953(.085)	0.903(.086)
Ambiguity		1.120(.006)	1.178(.008)	1.333(.023)	1.107(.005)	1.125(.006)
Amb. Aligned		**1.108**(.004)	1.175(.005)	1.442(.032)	1.109(.003)	1.112(.005)
Vehicle data—$\alpha_j = 0.04$						
Classifier		SSVM	LR	*k*NN	RF	SVM-Naive
$\widehat{\alpha}_j$	$Y = 1$	4.280(.308)	4.211(.429)	2.771(.280)	3.975(.325)	4.297(.348)
	$Y = 2$	5.089(.369)	3.982(.332)	3.500(.283)	6.250(.388)	3.938(.344)
	$Y = 3$	5.103(.367)	3.872(.346)	2.761(.267)	5.009(.336)	3.590(.331)
	$Y = 4$	4.000(.270)	3.727(.344)	3.434(.312)	3.949(.311)	4.232(.333)
Ambiguity		1.798(.013)	2.127(.023)	2.482(.016)	1.777(.010)	3.253(.013)
Amb. Aligned		1.924(.015)	2.150(.018)	2.713(.019)	**1.891**(.010)	3.290(.011)

NOTE: Rows annotated with "Ambiguity" give the ambiguity for each classifier, and "Amb. Aligned" give the ambiguity with the empirical noncoverage rates aligned with the nominal rates by adjusting the threshold. Numbers in the parenthesis are standard errors across 100 runs. SSVM has a comparable performance to the best plug-in method (logistic regression in CNAE dataset and Random Forest in zip-code and Vehicle dataset). The boldface indicates the smallest aligned ambiguity.

of Brazilian companies from nine categories. Each document was represented as a vector, where the weight of each word is its frequency in the document. This dataset is highly sparse (99.22% of the matrix is filled with zeros) with 856 predictors. The dimension of this data is much more than the number of observations. There are 120 observations for each class in the original dataset. We evenly split the observations to training, tuning and test set, which makes 360 documents for each set. The noncoverage rate is set to be 0.05 for all the classes. We apply linear SSVM, and compare with linear logistic regression, random forest, *k*NN and naive linear SVM on this dataset.

Zipcode Data: We conduct the comparison on the well-known hand-written zip code data (LeCun et al. 1989) widely used in the classification literature. The original dataset consists of 9298 16 × 16 (hence 256 predictos) pixel images of handwritten digits. There are both training and test sets defined in it. Lei (2014), Sadinle, Lei, and Wasserman (2017), Wang and Qiao (2018) used the same dataset for illustrating the set-valued classification. Following Lei (2014) and Wang and Qiao (2018), we only use a subset of the data containing digits $\{0, 6, 8, 9\}$. Previous studies (Shafer and Vovk 2008) pointed out that there were discrepancies between the training and test sets. In this study we first mixed the training and test data and then randomly split into new training, tuning and test data. The training and tuning data both have sample size 400, with 100 from each class.

Although Lei (2014) and Sadinle, Lei, and Wasserman (2017) sets nominal noncoverage rates to be 0.05 in their study, many nonlinear classifiers, such as SVM with Gaussian kernel, can achieve this noncoverage rate without introducing any ambiguity. Therefore, we reduce the noncoverage rate to 0.01 for both classes to make the task more challenging.

We apply Gaussian kernel for SSVM, and compare with kernel logistic regression with Gaussian kernel, random forest, *k*NN and naive SVM with Gaussian kernel on this dataset.

Vehicle Data: The Vehicle dataset (Siebert 1987) can be found in the UCI Machine Learning Repository. It is a four-class multicategory classification task with 946 observations and 18 predictors in total. We discriminate between silhouettes of model cars, vans and buses. We randomly split the data into training, tuning and test sets. The training and tuning sets are both of size 200 (50 for each class), and the rest is used as test set. We learn set-valued classifiers using both the proposed method and the plug-in methods and evaluate the performance with the test set. The noncoverage rate is set to be 0.04 for all the classes. We apply linear SSVM, and compare with linear logistic regression, random forest, *k*NN and linear naive SVM with on this dataset.

For each example, we repeat splitting 100 times and report the mean and standard error. We shows the results in Tables 1 and 2. In Table 1, we report the class-specific noncoverage rate. Ideally, they should be less than or equal to the nominal rates. The rows "Amb. Aligned" and "Ambiguity" show the ambiguity of the set-valued classifiers with and without aligning the non-

Table 2. The column $|\phi(\boldsymbol{X})|$ stands for different cardinalities of set-valued predictions.

CNAE data

$\lvert\phi(\boldsymbol{X})\rvert$	SSVM		LR		kNN		RF		SVM-Naive	
	Prop.	Acc.	Prop.	Acc.	Prop.	Acc.	Prop.	Acc.	Prop.	Acc.
1	66.0	98.4	65.2	97.9	19.5	98.7	60.6	98.7	37.9	97.8
2	26.6	96.7	27.3	97.7	26.7	98.2	23.7	95.9	41.4	97.3
3	6.3	95.6	6.7	97.8	26.6	96.8	11.2	95.7	16.4	95.5
4	1.1	98.3	0.7	99.5	21.3	96.8	4.2	97.8	3.8	98.0
≥ 5	0.04	100	0.02	100	5.9	98.0	0.03	98.8	0.6	94.3
$E(\lvert\phi(\boldsymbol{X})\rvert)$	1.43		1.43		2.68		1.60		1.88	
$P(Y \in \phi(\boldsymbol{X}))$	97.8		97.9		97.6		97.7		97.2	

Zip code data

$\lvert\phi(\boldsymbol{X})\rvert$	SSVM		LR		kNN		RF		SVM-Naive	
	Prop.	Acc.	Prop.	Acc.	Prop.	Acc.	Prop.	Acc.	Prop.	Acc.
1	88.8	99.2	86.0	99.3	71.1	99.5	89.9	99.1	88.5	99.2
2	10.4	96.9	10.8	96.3	24.7	97.9	9.4	96.3	10.6	96.8
3	0.7	98.5	2.6	95.0	4.0	98.0	0.6	98.9	0.9	97.8
4	0.04	100	0.6	100	0.2	100	0.01	100	0.07	100
$E(\lvert\phi(\boldsymbol{X})\rvert)$	1.12		1.18		1.33		1.11		1.13	
$P(Y \in \phi(\boldsymbol{X}))$	98.9		98.9		99.0		98.8		99.0	

Vehicle data

$\lvert\phi(\boldsymbol{X})\rvert$	SSVM		LR		kNN		RF		SVM-Naive	
	Prop.	Acc.	Prop.	Acc.	Prop.	Acc.	Prop.	Acc.	Prop.	Acc.
1	23.0	95.9	17.2	92.1	2.7	97.2	27.5	97.2	0.01	33.3
2	61.6	97.7	52.4	97.9	38.4	98.6	56.1	97.5	18.5	90.7
3	15.2	99.2	28.5	99.2	43.7	98.7	16.1	98.8	34.0	96.6
4	0.1	100	1.9	100	15.1	100	0.2	100	47.5	100
$E(\lvert\phi(\boldsymbol{X})\rvert)$	1.92		2.15		2.71		1.89		3.29	
$P(Y \in \phi(\boldsymbol{X}))$	97.3		98.8		97.7		97.1		97.1	

NOTE: For each classifier, we report the proportion of predictions with different cardinalities and the accuracy for each cardinality. Row $E(|\phi(\boldsymbol{X})|)$ is the average cardinality of set-valued predictions, which is the same as "Amb. Aligned." in Table 1. Row $P(Y \in \phi(\boldsymbol{X}))$ gives the overall accuracy for each classifier. They are very similar after the alignment.

coverage rates to be the nominal rates using the test data. If all the empirical noncoverage rates match the nominal rates, then one could simply compare the ambiguities. Unfortunately it is rarely the case. Instead, it is fair to compare the ambiguities after aligning the test data noncoverage rates: the smaller ambiguity, the better.

The effectiveness of the proposed method (SSVM) can be seen from the aligned ambiguity in Table 1. Overall, no single method is the best over all cases, but the proposed SSVM is either the best or comparable to the best plug-in methods. In low-dimensional datasets (Vehicle), SSVM outperforms the logistic regression, *k*NN and naive MSVM methods and comes close to random forest. In relatively high-dimensional settings, SSVM's performance improves. In particular, it is slightly better comparing to the best plug-in methods, random forest and naive MSVM in the zip code data. In the high-dimensional CNAE-9 data, it is slightly better than logistic regression and significantly better than all the plug-in methods.

Table 2 provides a different perspective to this study. It shows the proportions and accuracy of the set-valued predictions conditional on the size of the prediction set. It also shows the expected size of the prediction set and the overall accuracy for each method. Here we define accuracy as the probability that the true label is contained in the prediction set.

The naive SVM method does not give successful acceptance regions on most of the datasets. Although the proposed method also uses the hinge loss as the surrogate, it performs much better. This illustrates the potential power of the proposed risk minimization framework that explicitly incorporates the noncoverage consideration.

6. Conclusion

In this work, we propose to learn multicategory acceptance regions to achieve set-valued classification using empirical minimization. We make use of a general large-margin framework for the learning task. It is important to choose appropriate surrogate losses for the proposed problem. In particular, we use truncated hinge loss in the objective with proven Fisher consistency and use the weighted hinge loss to obtain a close approximation to the noncoverage rates. The angle-based learning approach is used to effectively learn the classifier in the high-dimensional setting. Theoretical and numerical studies have shown the effectiveness of our approach in controlling the noncoverage rate and minimizing the ambiguity. Other surrogate losses can be considered in this framework as future work.

In our proposed framework of set-valued classification, we optimize the ambiguity while imposing a constraint on the noncoverage rate (equivalently, the class-specific accuracy). A separate stream of research in the machine learning community (Denis and Hebiri 2015, 2017; Shekhar, Ghavamzadeh, and Javidi 2019) consider the paradigm in which one optimizes the accuracy with an constraint on how many ambiguous predictions (prediction sets with size greater than 1) can be made. It

will be interesting to investigate the statistical properties of set-valued classifiers in this alternative setting.

Supplementary Materials

1. A .PDF file that contains proofs of Theorems in the main paper and additional numerical studies.
2. Computer program codes needed to reproduce the numerical studies.

Acknowledgments

The authors thank the editors, the associate editor, and anonymous reviewers for their helpful comments and suggestions which led to a much improved presentation. This research work was conducted when Wenbo Wang was a Ph.D. candidate at the Department of Mathematical Sciences at Binghamton University, State University of New York, Binghamton, New York, 13902. The authors report there are no competing interests to declare.

References

Altman, N. S. (1992), "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression," *The American Statistician*, 46, 175–185. [8]

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. (2006), "Convexity, Classification, and Risk Bounds," *Journal of the American Statistical Association*, 101, 138–156. [6]

Bartlett, P. L., and Wegkamp, M. H. (2008), "Classification with a Reject Option Using a Hinge Loss," *Journal of Machine Learning Research*, 9, 1823–1840. [2,6]

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*, 02–2951. [2]

Ciarelli, P. M., and Oliveira, E. (2009), "Agglomeration and Elimination of Terms for Dimensionality Reduction," in *2009 Ninth International Conference on Intelligent Systems Design and Applications*, IEEE, pp. 547–552. [10]

Cortes, C., and Vapnik, V. (1995), "Support-Vector Networks," *Machine Learning*, 20, 273–297. [8]

Denis, C., and Hebiri, M. (2015), "Consistency of Plug-in Confidence Sets for Classification in Semi-supervised Learning," arXiv preprint arXiv:1507.07235. [2,12]

——— (2017), "Confidence Sets with Expected Sizes for Multiclass Classification," *The Journal of Machine Learning Research*, 18, 3571–3598. [2,12]

Dümbgen, L., Igl, B.-W., and Munk, A. (2008), "*P*-values for Classification," *Electronic Journal of Statistics*, 2, 468–493. [2,7]

Fürnkranz, J., and Hüllermeier, E. (2010), "Preference Learning: An Introduction," in *Preference Learning*, eds. J. Fürnkranz and E. Hällermeier, pp. 1–17, Berlin: Springer. [2]

Guan, L., and Tibshirani, R. (2019), "Prediction and Outlier Detection: A Distribution-Free Prediction Set with a Balanced Objective," arXiv preprint arXiv:1905.04396. [2,3]

Hechtlinger, Y., Póczos, B., and Wasserman, L. (2018), "Cautious Deep Learning," arXiv preprint arXiv:1805.09460. [2,3]

Herbei, R., and Wegkamp, M. H. (2006), "Classification with Reject Option," *Canadian Journal of Statistics*, 34, 709–721. [2]

Hoffgen, K.-U., Simon, H.-U., and Vanhorn, K. S. (1995), "Robust Trainability of Single Neurons," *Journal of Computer and System Sciences*, 50, 114–125. [4]

Hunter, D. R., and Lange, K. (2004), "A Tutorial on MM Algorithms," *The American Statistician*, 58, 30–37. [4]

Kimeldorf, G. S., and Wahba, G. (1970), "A Correspondence between Bayesian Estimation on Stochastic Processes and Smoothing by Splines," *The Annals of Mathematical Statistics*, 41, 495–502. [5]

Le Cessie, S., and Van Houwelingen, J. C. (1992), "Ridge Estimators in Logistic Regression," *Applied statistics*, 41, 191–201. [8]

Le Thi Hoai, A., and Tao, P. D. (1997), "Solving a Class of Linearly Constrained Indefinite Quadratic Problems by DC Algorithms," *Journal of Global Optimization*, 11, 253–285. [4]

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989), "Backpropagation Applied to Handwritten zip Code Recognition," *Neural computation*, 1, 541–551. [11]

Lee, Y., Lin, Y., and Wahba, G. (2004), "Multicategory Support Vector Machines: Theory and Application to the Classification of Microarray Data and Satellite Radiance Data," *Journal of the American Statistical Association*, 99, 67–81. [4,8]

Lei, J. (2014), "Classification with Confidence," *Biometrika*, 101, 755–769. [2,8,11]

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. (2018), "Distribution-Free Predictive Inference for Regression," *Journal of the American Statistical Association*, 113, 1094–1111. [2]

Lei, J., Robins, J., and Wasserman, L. (2013), "Distribution-Free Prediction Sets," *Journal of the American Statistical Association*, 108, 278–287. [2]

Liaw, A., and Wiener, M. (2002), "Classification and Regression by RandomForest," *R news*, 2, 18–22. [8]

Manwani, N., Desai, K., Sasidharan, S., and Sundararajan, R. (2015), "Double Ramp Loss based Reject Option Classifier," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, pp. 151–163. [6]

O'Neil, C. (2016), *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*, New York: Broadway Books. [1]

Platt, J. (1999), "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," *Advances in large margin classifiers*, 10, 61–74. [8]

Rigollet, P., and Tong, X. (2011), "Neyman-Pearson Classification, Convexity and Stochastic Constraints," *Journal of Machine Learning Research*, 12, 2831–2855. [2,4]

Sadinle, M., Lei, J., and Wasserman, L. (2017), "Least Ambiguous Set-Valued Classifiers with Bounded Error Levels," *Journal of the American Statistical Association*, 114, 223–224. [2,3,6,8,11]

Scholkopf, B., and Smola, A. J. (2001), *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, Cambridge, MA: MIT Press. [2]

Shafer, G., and Vovk, V. (2008), "A Rutorial on Conformal Prediction," *Journal of Machine Learning Research*, 9, 371–421. [2,11]

Shekhar, S., Ghavamzadeh, M., and Javidi, T. (2019), "Binary Classification with Bounded Abstention Rate," arXiv preprint arXiv:1905.09561. [12]

Siebert, J. P. (1987), "Vehicle Recognition Using Rule Based Methods," Technical report. [11]

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. [2]

Vovk, V., Nouretdinov, I., Fedorova, V., Petej, I., and Gammerman, A. (2017), "Criteria of Efficiency for Set-Valued Classification," *Annals of Mathematics and Artificial Intelligence*, 81, 21–46. [2]

Wang, J., Shen, X., and Liu, Y. (2007), "Probability Estimation for Large-Margin Classifiers," *Biometrika*, 95, 149–167. [2]

Wang, W., and Qiao, X. (2018), "Learning Confidence Sets using Support Vector Machines," in *Advances in Neural Information Processing Systems*, pp. 4929–4938. [6,11]

Wu, Y., and Liu, Y. (2007), "Robust Truncated Hinge Loss Support Vector Machines," *Journal of the American Statistical Association*, 102, 974–983. [4]

——— (2013), "Adaptively Weighted Large Margin Classifiers," *Journal of Computational and Graphical Statistics*, 22, 416–432. [5]

Wu, Y., Zhang, H. H., and Liu, Y. (2010), "Robust Model-Free Multiclass Probability Estimation," *Journal of the American Statistical Association*, 105, 424–436. [2]

Yuan, M., and Wegkamp, M. (2010), "Classification Methods with Reject Option based on Convex Risk Minimization," *Journal of Machine Learning Research*, 11, 111–130. [2]

Zhang, C., and Liu, Y. (2013), "Multicategory Large-Margin Unified Machines," *The Journal of Machine Learning Research*, 14, 1349–1386. [3,8]

Zhang, C., Wang, W., and Qiao, X. (2017), "On Reject and Refine Options in Multicategory Classification," *Journal of the American Statistical Association* (accepted). [2,6,10]

Zhu, J., and Hastie, T. (2005), "Kernel Logistic Regression and the Import Vector Machine," *Journal of Computational and Graphical Statistics*, 14, 185–205. [8]