Published in final edited form as:

J Multivar Anal. 2018 July ; 166: 282–299. doi:10.1016/j.jmva.2018.03.004.

Adaptively weighted large-margin angle-based classifiers

Sheng Fu^{a,b}, Sanguo Zhang^{a,b}, and Yufeng Liu^{c,*}

^aSchool of Mathematical Science, University of the Chinese Academy of Sciences, Beijing 100049, China

^bKey Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, China

^cDepartment of Statistics and Operations Research, Department of Genetics, Department of Biostatistics, Carolina Center for Genome Sciences, Lineberger Comprehensive Cancer Center, University of North Carolina at Chapel Hill, NC 27599, USA

Abstract

Large-margin classifiers are powerful techniques for classification problems. Although binary large-margin classifiers are heavily studied, multiclass problems are more complicated and challenging. A common approach is to construct k different decision functions for a k -class problem with a sum-to-zero constraint. However, such a constraint can be inefficient. Moreover, many large-margin classifiers can be sensitive to outliers in the training sample. In this article, we use the angle-based classification framework to avoid the explicit sum-to-zero constraint, and we propose two adaptively weighted large-margin classification techniques. Our new methods are Fisher consistent and more robust against outliers under suitable conditions. Numerical experiments further indicate that our methods give competitive and stable performance when compared with existing approaches.

Keywords

Fisher Consistency; Multiclass Classification; Robustness; SVM; Weighted Learning

1. Introduction

Classification is an archetypal example of supervised learning problem with many diverse applications, e.g., to handwritten digit recognition, disease diagnosis, and fraud detection. Many classification methods are available in the literature, including logistic regression, Fisher linear discriminant analysis (LDA), support vector machine (SVM), tree-based methods, and many more; see [2, 11] for comprehensive reviews.

*Correspondence to: Yufeng Liu, Department of Statistics and Operations Research, CB3260, University of North Carolina, Chapel Hill, NC 27599, USA.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Large-margin classifiers have attracted a lot of attention in recent years. The SVM is an appealing and powerful method which enjoys great success in various situations [3, 5, 6, 8, 22]. It was originally designed for two-class problems. The basic idea is to search for a single classification function that maximally separates the two classes. One appealing property of the SVM is that the resulting classification function only depends on a subset of the training data. The kernel trick implicitly maps the training data from the original input space into a more flexible, possibly high-dimensional, feature space, which enables the SVM to handle high-dimensional data and nonlinear classification problems. These three characteristics — maximum margin principle, support vector property, and kernel trick — enable the SVM to be successful and explain its popularity.

Since it is frequent to have more than two classes in practice, extensions dealing with multicategory classification problems are necessary and challenging. Two general approaches are used to handle k -class classification problems:

- (i) Reduce the multicategory problem through a sequence of binary classifiers, e.g., using a one-against-one [22] or a one-against-rest [9] strategy. Both of them have obvious drawbacks. The one-against-one approach evaluates all possible pairwise classifiers and then combines $k(k-1)/2$ individual binary classifiers; these classifiers may suffer from the tie-in-vote problem. The one-against-rest approach trains k classifiers to separate each class from the combined remaining classes, and often leads to imbalanced training data [13, 14].
- (ii) Consider all classes simultaneously [4, 22, 23]. This type of method uses a k -tuple function vector $\mathbf{f}$ which maps the covariates to $\mathbb{R}^k$. The prediction rule assigns the class label to the category that corresponds to the maximum element within the function vector. To reduce the parameter space and ensure the uniqueness of optimal solution, a sum-to-zero constraint is often employed on the decision vector $\mathbf{f}$. Many existing Multicategory SVM (MSVM) methods fall within this framework; see, e.g., [7, 13, 17, 21].

Despite their great success, the SVM methods still suffer from some drawbacks in certain situations. In particular, the SVM uses the unbounded hinge loss and consequently the resulting classifiers may be affected by points far away from their own classes, namely “outliers” in the training data. To overcome these problems, Shen et al. [20] and Liu et al. [15] proposed ψ -learning for binary and multicategory problems, respectively. Wu and Liu [24, 25] applied the truncated hinge loss to obtain robust multicategory SVMs. However, the ψ -learning and robust truncated MSVMs both follow the non-convex optimization framework. Wu and Liu [26] proposed a weighted large-margin classification technique to gain robustness, which can be more computationally efficient than methods in [15, 20, 24, 25].

For binary classification, the two classes can be separated by a single decision function. Analogously, it suffices to use $k-1$ decision functions for k -class problems. Recently, Zhang and Liu [28] and Zhang et al. [29] proposed a new angle-based large-margin classification framework. They used a regular k -vertex simplex structure centered at the origin in $\mathbb{R}^{k-1}$. Each vertex stands for one class and is a $(k-1)$ -dimensional vector. The

classification function vector $\mathbf{f}$ maps the covariates of a given instance to $\mathbb{R}^{k-1}$. The prediction rule assigns a new instance to the category whose corresponding vertex vector has the smallest angle with respect to the mapped classification function vector $\mathbf{f}$.

Although multicategory ψ -learning [15] and robust MSVMs [24, 25] are less sensitive to outliers, there are still drawbacks. Because the ψ loss and the truncated hinge loss are non-convex, the corresponding optimization problems are more complex than the convex quadratic programming or linear programming for typical MSVMs. Furthermore, these methods need to estimate k functions with the sum-to-zero constraint.

In this paper, we first provide two types of general loss functions under the angle-based framework and study their properties. To ensure robustness while reducing computational complexity, we apply weighted learning techniques to two multicategory large-margin angle-based classification formulations without the sum-to-zero constraint. Our methods have several attractive properties. First, the angle-based classification strategy avoids the sum-to-zero constraint, which makes the formulation more compact and efficient. Second, the adaptively weighted methods we propose are robust and computationally effective, given that the objective function is convex. Third, the new formulations are Fisher consistent under suitable conditions. Finally, our numerical experiments demonstrate that the proposed methods offer competitive classification prediction accuracy compared to various simultaneous multicategory classifiers.

The rest of the paper is organized as follows. In Section 2, we briefly review several existing MSVM methods, and then introduce the angle-based classification framework with both convex and non-convex loss functions. Some consistency results for general loss functions are provided as well. In Section 3, we propose convex weighted learning techniques to achieve robustness and discuss their connections with non-convex robust methods. The results of our numerical studies showing the effectiveness of the new methods are reported in Section 4. Some discussion is given in Section 5. All proofs and some details are included in the Appendix.

2. Multicategory large-margin classification

Let $P(\mathbf{X}, Y)$ be some unknown probability distribution of $(\mathbf{X}, Y)$, where $\mathbf{X}$ is a vector of predictors, and Y is a label. Consider a k -category classification problem with $k \geq 2$. Suppose the training set is $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ from $P(\mathbf{X}, Y)$, where $\mathbf{x}_i$ is a p -dimensional covariate, and $y_i \in \{1, \dots, k\}$ represents the corresponding label. Our target is to build a classification model based on the training data, which can be used to predict the class label of the new observed data. In this section, we first review the simultaneous large-margin classifiers in Section 2.1, and then introduce the angle-based classification framework with general loss functions in Section 2.2. Robust non-convex loss functions are provided in Section 2.3. Some statistical properties of the angle-based methods are given in Section 2.4.

2.1. Multicategory classification with k functions

Define a decision function $\mathbf{f} = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))^\top \in \mathbb{R}^k$ in the vector form, where each component corresponds to one class. The prediction rule is $\hat{y} = \arg\max_{j \in \{1, \dots, k\}} f_j(\mathbf{x})$ for

any new data point $\mathbf{x}$. Then, the corresponding optimization of large-margin classifiers can typically be written as

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \ell\{f(\mathbf{x}_i), y_i\} + \lambda J(f), \quad \text{subject to } \sum_{j=1}^k f_j(\mathbf{x}) = 0 \text{ for all } \mathbf{x},$$

where $\mathcal{F}$ is the functional space for f , the first part of the objective function is the empirical loss term with a general surrogate loss $\ell\{f(\mathbf{x}), y\}$, J denotes the penalty term to prevent overfitting, and λ is a nonnegative tuning parameter that balances the loss and penalty terms. Since $(\mathbf{x}, y)$ is misclassified when $\hat{y} = \operatorname{argmax}_{j \in \{1, \dots, k\}} f_j(\mathbf{x}) \neq y$, a good loss function should make f_y the maximum among $f_1, \dots, f_k$. In the literature, the following loss functions are commonly used versions to extend the hinge loss of the binary SVM to the MSVMs:

- a) MSVM1 (Naive hinge loss) $\ell\{f(\mathbf{x}), y\} = [1 - f_y(\mathbf{x})]_+$, where $[u]_+ = \max(0, u)$.
- b) MSVM2 $\ell\{f(\mathbf{x}), y\} = \sum_j [1 - \{f_j(\mathbf{x}) - f_y(\mathbf{x})\}]_+$; see [23].
- c) MSVM3 $\ell\{f(\mathbf{x}), y\} = [1 - \min_j \{f_j(\mathbf{x}) - f_y(\mathbf{x})\}]_+$; see [7].
- d) MSVM4 $\ell\{f(\mathbf{x}), y\} = \sum_j [1 + f_j(\mathbf{x})]_+$; see [13].
- e) MSVM5 $\ell\{f(\mathbf{x}), y\} = \gamma[(k-1) - f_y(\mathbf{x})]_+ + (1 - \gamma) \sum_j [1 + f_j(\mathbf{x})]_+$, $\gamma \in [0, 1]$; [17].

The sum-to-zero constraint is imposed for MSVM1, 4 and 5, and is automatically satisfied for the solutions of MSVM2 and MSVM3. Interestingly, although the binary SVM is known to be Fisher consistent, not all MSVMs are. In particular, MSVMs 1–3 are not always consistent. In contrast, MSVM5 is always Fisher consistent when $\gamma \in [0, 0.5]$, which also covers MSVM4 as a special case with $\gamma = 0$.

2.2. Multicategory angle-based classification with $k - 1$ functions

The sum-to-zero constraint used to reduce the degrees of freedom of the k -component function f to $k - 1$ can be inefficient. To address this problem, one can build a classifier with $k - 1$ functions directly. Recently, Zhang and Liu [28] proposed a multicategory angle-based classification framework. Consider a k -vertex regular simplex in $\mathbb{R}^{k-1}$, whose center is at the origin and with equal pairwise distances among the vertices. We use the k vertices to represent the k classes. Let $\mathbf{W}$ be the collection $\{\mathbf{W}_1, \dots, \mathbf{W}_k\}$ of vertex vectors with elements

$$\mathbf{W}_j = \begin{cases} \frac{1}{\sqrt{k-1}} \mathbf{1} & \text{if } j = 1, \\ -\frac{1 + \sqrt{k}}{(k-1)^{3/2}} \mathbf{1} + \sqrt{\frac{k}{k-1}} \mathbf{e}_{j-1} & \text{if } j \in \{2, \dots, k\}, \end{cases}$$

where $\mathbf{1} \in \mathbb{R}^{k-1}$ is a vector of 1s, and $\mathbf{e}_j \in \mathbb{R}^{k-1}$ is a vector of 0s except the j th element is 1. Without loss of generality, assume that class j is assigned to $\mathbf{W}_j$. The angle-based classifiers map $\mathbf{x}$ into $\mathbb{R}^{k-1}$ using a $(k-1)$ -component function $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_{k-1}(\mathbf{x}))^\top$. Then the vector $\mathbf{f}(\mathbf{x})$ can generate k angles with respect to $\mathbf{W}_j$, namely $\{\angle(\mathbf{f}(\mathbf{x}), \mathbf{W}_1), \dots, \angle(\mathbf{f}(\mathbf{x}), \mathbf{W}_k)\}$.

The prediction for a new $\mathbf{x}$ is the class whose corresponding angle is the smallest. One can verify that the least-angle rule is equivalent to the largest inner product rule for k inner products $\{\langle \mathbf{f}(\mathbf{x}), \mathbf{W}_1 \rangle, \dots, \langle \mathbf{f}(\mathbf{x}), \mathbf{W}_k \rangle\}$, i.e.,

$$\hat{y} = \operatorname{argmin}_{j \in \{1, \dots, k\}} \angle(\mathbf{f}(\mathbf{x}), \mathbf{W}_j) = \operatorname{argmax}_{j \in \{1, \dots, k\}} \langle \mathbf{f}(\mathbf{x}), \mathbf{W}_j \rangle.$$

Note that $\langle \mathbf{f}(\mathbf{x}), \mathbf{W}_1 \rangle + \dots + \langle \mathbf{f}(\mathbf{x}), \mathbf{W}_k \rangle = 0$ for all $\mathbf{x}$, and hence the angle-based methods are free of the sum-to-zero constraint. The goal is then to maximize the angle-based margin $\langle \mathbf{f}(\mathbf{x}), \mathbf{W}_y \rangle$ for any $(\mathbf{x}, y)$. Given a binary large-margin loss function $\ell(u)$, the angle-based classifier can be expressed as

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \ell(\langle \mathbf{f}(\mathbf{x}_i), \mathbf{W}_{y_i} \rangle) + \lambda J(f). \quad (1)$$

Because it has only $k-1$ components and hence fewer parameters, it can be optimized more efficiently than other traditional methods. Furthermore, the angle-based classification in (1) is Fisher consistent if the derivative ℓ' exists and $\ell'(u) < 0$ for all u . When the naive hinge loss $\ell(u) = [1-u]_+$ is applied, one also gets a direct multicategory extension of the binary SVM; however, such a naive SVM extension is not always Fisher consistent [28].

Define a general hinge loss $H_s(u) = [s-u]_+$ with location parameter s . Note that H_s includes the original hinge loss and its horizontal shifts. Recently, Zhang et al. [29] proposed a new MSVM family to enrich the angle-based methods, namely reinforced angle-based multicategory SVM (RAMSVM). The RAMSVM solves the following problem

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \{\gamma H_{k-1}(\langle \mathbf{f}(\mathbf{x}_i), \mathbf{W}_{y_i} \rangle) + (1-\gamma) \sum_{j \neq y_i} H_1(-\langle \mathbf{f}(\mathbf{x}_i), \mathbf{W}_j \rangle)\} + \lambda J(f), \quad (2)$$

where $\gamma \in [0, 1]$ is the convex combination parameter. Zhang et al. [29] showed that RAMSVM is Fisher consistent with $\gamma \in [0, 0.5]$. Problem (1) with the hinge loss $\ell(u) = H_{k-1}(u)$ is equivalent to problem (2) with $\gamma = 1$. The value $\gamma = 0.5$ was suggested by Zhang et al. [29] based on numerical and real data experiments.

In this paper, we propose more general large-margin unified angle-based loss families. Consider two convex large-margin loss functions ℓ_1 and ℓ_2 , and the convex combination parameter $\gamma \in [0, 1]$. Then a multicategory angle-based loss family covering (1) and (2) can be written as

$$\phi_1\{\mathbf{f}(\mathbf{x}), y\} = \gamma \ell_1(\langle \mathbf{f}(\mathbf{x}), \mathbf{W}_y \rangle) + (1-\gamma) \sum_{j \neq y} \ell_2(-\langle \mathbf{f}(\mathbf{x}), \mathbf{W}_j \rangle). \quad (3)$$

A key motivation for using the loss function in (3) is the argmax rule for multicategory classification. In order to obtain a correct classification for $(\mathbf{x}, y)$, we need $\langle \mathbf{f}(\mathbf{x}), \mathbf{W}_y \rangle$ to be

the maximum among $\langle f(\mathbf{x}), \mathbf{W}_1 \rangle, \dots, \langle f(\mathbf{x}), \mathbf{W}_k \rangle$. On one hand, ℓ_1 in (3) encourages $\langle f(\mathbf{x}), \mathbf{W}_y \rangle$ to be large. Since $\sum_{j=1}^k \langle f(\mathbf{x}), \mathbf{W}_j \rangle = 0$ for all $\mathbf{x}$, then $\langle f(\mathbf{x}), \mathbf{W}_y \rangle = -\sum_{j \neq y} \langle f(\mathbf{x}), \mathbf{W}_j \rangle$. The second loss ℓ_2 encourages $\langle f(\mathbf{x}), \mathbf{W}_j \rangle$ to be small for all $j \neq y$. Consequently, both terms in (3) try to make $\langle f(\mathbf{x}), \mathbf{W}_y \rangle$ large for correct classification in different directions.

Motivated by the generalized functional margin in [7, 15, 25, 26], we develop its angle-based version $g\{f(\mathbf{x}), y\} = \min_{j \neq y} \langle f(\mathbf{x}), \mathbf{W}_y - \mathbf{W}_j \rangle$. A given point $(\mathbf{x}, y)$ is misclassified if $g\{f(\mathbf{x}), y\} \leq 0$. Using the newly defined angle-based functional margin for any binary convex large-margin loss function ℓ , we can get another multicategory angle-based loss family as follows

$$\phi_2\{f(\mathbf{x}), y\} = \ell_1 \left\{ \min_{j \neq y} \langle f(\mathbf{x}), \mathbf{W}_y - \mathbf{W}_j \rangle \right\}, \quad (4)$$

which encourages the new angle-based margin $g\{f(\mathbf{x}), y\}$ to be large.

The main difference between loss functions (3) and (4) lies in their types of functional margins. Loss (3) focuses on the k inner products independently, and loss (4) considers pairwise comparisons of all inner products. Thus the former margin is “absolute” while the latter is “relative”. Both of these two margins help to ensure correct classification. Our theoretical and numerical studies suggest that (3) and (4) yield comparable classification performance.

The two multicategory large-margin loss families $\phi_1\{f(\mathbf{x}), y\}$ and $\phi_2\{f(\mathbf{x}), y\}$ are well designed for many applications. However, due to the use of unbounded convex loss functions, the resulting classifiers can be sensitive to outliers. Truncation of loss functions can help to obtain more robust classifiers as we discuss next.

2.3. Large-margin angle-based classification with truncated loss

In practice, one may encounter classification problems with very noisy data, and robust classifiers can be necessary in such situations. Wu and Liu [24, 25] showed that using the bounded truncated loss can lead to robust classifiers. Motivated by this, we consider two families of truncated loss functions for large-margin angle-based classification.

For the first type of classifiers corresponding to $\phi_1\{f(\mathbf{x}), y\}$, let $s, v \geq 0$ be the truncation locations for $\ell_1(u)$ and $\ell_2(u)$, and write $T_s(u) = \min\{\ell_1(s), \ell_1(u)\}$ and $R_v(u) = \min\{\ell_2(v), \ell_2(u)\}$. Then a new family of truncated loss functions for (3) can be written as

$$\phi_1^t\{f(\mathbf{x}), y\} = \gamma T_s(\langle f(\mathbf{x}), \mathbf{W}_y \rangle) + (1 - \gamma) \sum_{j \neq y} R_v(-\langle f(\mathbf{x}), \mathbf{W}_j \rangle). \quad (5)$$

For the second type of classifiers corresponding to $\phi_2\{f(\mathbf{x}), y\}$, using the truncated loss function $T_s(u)$, the proposed truncated loss function for (4) becomes

$$\phi_2^t\{f(\mathbf{x}), y\} = T_s \left\{ \min_{j \neq y} \langle f(\mathbf{x}), \mathbf{W}_j - \mathbf{W}_y \rangle \right\}. \quad (6)$$

Since the objective functions with truncated loss functions (5) and (6) are non-convex, the corresponding optimization becomes more involved than the original convex problems.

Wu and Liu [24, 25] employed the truncated hinge loss for robust multicategory SVMs. Furthermore, Zhang et al. [30] employed two truncated hinge loss functions for the angle-based classifiers, which can be viewed as special cases of the loss functions (5) and (6). In particular, Zhang et al. [30] designed a special truncated loss with $T_{(k-1),s}(u) = H_{k-1}(u) - H_{(k-1),s}(u)$ and $R_s(u) = H_1(u) - H_s(u)$ in (5), and $T_s(u) = H_1(u) - H_s(u)$ in (6). They showed that the corresponding classifiers can enjoy sound properties, and alleviate the impact of outliers as well.

Note that the truncated loss function can be decomposed as the difference of two convex functions. For loss (5), we have $T_s(u) = \ell_1(u) - [\ell_1(u) - \ell_1(s)]_+$ and $R_s(u) = [\ell_2(u) - \ell_2(v)]_+$, where the two parts of such decompositions are convex. To solve the problems with truncated loss (5) and (6), one can apply the DCA [1, 16]. This algorithm solves the non-convex minimization problem by minimizing a sequence of convex subproblems.

So far, we have introduced two multicategory angle-based loss families including both convex and non-convex loss functions. In Section 2.4, we provide some statistical properties of these loss functions. Then we introduce our robust weighted convex learning techniques, and their connections with our proposed truncated loss functions in Section 3.

2.4. Statistical properties

Fisher consistency is a fundamental property of a classification loss function. A Fisher consistent classification method can achieve the best classification performance asymptotically, when the underlying functional space is rich enough. For studies on Fisher consistency of multicategory classifiers, see [13, 14, 17, 25, 31, 32].

For a given $\mathbf{x}$, define the j th class conditional probability $P_j(\mathbf{x}) = \Pr(Y = j | \mathbf{x} = \mathbf{x})$ for each $j \in \{1, \dots, k\}$. Fisher consistency requires that the theoretical predicted label $\hat{y}$ matches the Bayes rule $\arg\max_{j \in \{1, \dots, k\}} P_j(\mathbf{x})$. Assume $\phi\{f(\mathbf{x}), y\}$ is an arbitrary loss function or its truncation for angle-based classifiers, and denote the minimizer of $E[\phi\{f(\mathbf{x}), y\} | \mathbf{x} = \mathbf{x}]$ by $f^*(\mathbf{x})$. Under the angle-based classification framework, Fisher consistency requires that

$$\arg\max_{j \in \{1, \dots, k\}} \langle f^*(\mathbf{x}), \mathbf{W}_j \rangle = \arg\max_{j \in \{1, \dots, k\}} P_j(\mathbf{x}).$$

If the right-hand side is unique, so is the left-hand side. Thus, Fisher consistency ensures that the optimal classifier $f^*(\mathbf{x})$ assigns $\mathbf{x}$ to the class with the largest class conditional probability.

Next we establish Fisher consistency for the proposed two robust large-margin angle-based classifiers with truncated loss functions (5) and (6) in the following theorems.

Theorem 1. *Let ℓ_1, ℓ_2 be differential loss functions. If $\ell_1'(u) < 0$ and $\ell_2'(u) < 0$ for all u , then our first truncated large-margin angle-based classifier with truncated loss (5) is Fisher consistent with $\gamma \in [0, 1]$, $s = 0$ and $v = 0$. $\square$*

Note that ℓ_1 and ℓ_2 are not required to be the same; they can be different types of loss functions. If $s \rightarrow -\infty$, and $v \rightarrow -\infty$, the truncation effect vanishes. Then the large-margin angle-based classifier with untruncated loss (3) is also Fisher consistent. Theorem 1 extends the family of existing multicategory classifiers, such as [17, 27–29]. For example, the Large-margin Unified Machine (LUM) loss function [18] meets the condition in Theorem 1 and hence the angle-based classifier with LUM loss or its truncation is Fisher consistent.

Theorem 2. *Let ℓ_1 be a nonincreasing loss function with $\ell_1'(0) < 0$. A necessary and sufficient condition for our second truncated large-margin angle-based classifier with the truncated loss (6) to be Fisher consistent is that the truncation location s satisfies $\sup_{u>0} \{\ell_1(0) - \ell_1(u)\} / \{T_s(-u) - \ell_1(0)\} \geq k - 1$. $\square$*

When $s \rightarrow -\infty$, no truncation is applied for $\ell_1(u)$. The sufficient and necessary condition for Fisher consistency of loss (4) then becomes

$$\sup_{u>0} \{\ell_1(0) - \ell_1(-u)\} / \{\ell_1(-u) - \ell_1(0)\} \geq k - 1.$$

The condition for Fisher consistency given in Theorem 2 is more general than in [25, 30]. In particular, considering the hinge loss, when $s \in [-1/(k-1), 0]$, the corresponding classifier using the truncated hinge loss is Fisher consistent. Without truncation, the hinge loss does not Fisher consistent unless $k = 2$. A similar conclusion was reached in [25].

3. Robust convex weighted learning

Wu and Liu [24,25] showed that truncation of unbounded and convex loss functions can help to obtain robustness. Since large-margin classifiers with truncated loss solve non-convex optimization problems, the corresponding computations are more difficult. Wu and Liu [26] suggested that data adaptively weighted learning can convert non-convex problems to convex optimization and achieve robustness. They further showed that the iterative weighted MSVM can be equivalent to the truncated hinge loss SVM in terms of fixed points. For the large-margin angle-based classifiers with truncated loss functions (5) and (6), the optimization problems involve non-convex objective functions, and can be hard to optimize. To address this problem, we apply weighted learning for large-margin angle-based classifiers.

For illustration, we focus on the SVM and its extensions under the angle-based classification framework. In Section 3.1, we propose two corresponding weighted angle-based classifiers. The choices of weights are discussed in Section 3.2, which includes the one-step and iteratively weighting strategies. We summarize and compare these two weighting strategies

in Section 3.3, where connections between robust learning and weighted learning are highlighted.

3.1. Weighted angle-based classifiers

We first design weighted learning for loss (3). Suppose that for each $i \in \{1, \dots, n\}$, we have k given nonnegative weights $w_{i,1}, \dots, w_{i,k}$ for $(\mathbf{x}_i, y_i)$. The weighted learning for classifier with loss (3) solves the problem

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \left[\gamma w_{i,y_i} \ell_1 \{ \langle \mathbf{f}(\mathbf{x}_i), \mathbf{W}_{y_i} \rangle \} + (1-\gamma) \sum_{j \neq y_i} w_{i,j} \ell_2 \{ -\langle \mathbf{f}(\mathbf{x}_i), \mathbf{W}_j \rangle \} \right] + \lambda J(f). \quad (7)$$

We call this type of classifier the weighted angle-based classifier I (WAC1). Problem WAC1 can be regarded as the weighted learning of loss (3). The standard problem (3) corresponds to the case where $w_{i,j} = 1$ for all $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, k\}$.

Set $\ell_1(u) = H_{k-1}(u)$ and $\ell_2(u) = H_1(u)$. Such two loss functions are used in [17, 29]. We call this classifier weighted angle-based SVM I (WASVM1), and its optimization problem can be rewritten as

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \left[\gamma w_{i,y_i} H_{k-1} \{ \langle \mathbf{f}(\mathbf{x}_i), \mathbf{W}_{y_i} \rangle \} + (1-\gamma) \sum_{j \neq y_i} w_{i,j} H_1 \{ -\langle \mathbf{f}(\mathbf{x}_i), \mathbf{W}_j \rangle \} \right] + \lambda J(f). \quad (8)$$

Next, we design weighted learning for the loss (4). In contrast to the setting discussed above, we only define, for each $i \in \{1, \dots, n\}$, one nonnegative weight w_i associated with $(\mathbf{x}_i, y_i)$. Then the proposed weighted learning for the large-margin angle-based classifier with loss (4) solves the optimization problem

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n w_i \ell_1 \left\{ \min_{j \neq y_i} \langle \mathbf{f}(\mathbf{x}_i), \mathbf{W}_{y_i} - \mathbf{W}_j \rangle \right\} + \lambda J(f). \quad (9)$$

We name such a classifier the weighted angle-based classifier II (WAC2). If $w_1 = \dots = w_n = 1$, it reduces to the standard large-margin angle-based classification framework with loss (4).

When the commonly used hinge loss $H_1(u)$ is applied for $\ell_1(u)$, we call such classifiers the weighted angle-based SVM II (WASVM2). The corresponding problem can be formulated as

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n w_i H_1 \left\{ \min_{j \neq y_i} \langle \mathbf{f}(\mathbf{x}_i), \mathbf{W}_{y_i} - \mathbf{W}_j \rangle \right\} + \lambda J(f). \quad (10)$$

All technical details of WASVM1 (8) and WASVM2 (10) for linear and nonlinear learning are provided in the Appendix.

3.2. Choices of Weights

To achieve robustness, we discuss two different strategies for data adaptive weighting, the one-step weighted learning and the iteratively weighted learning, which are both convex optimization problems and can be solved efficiently. Through proper choices of weights, our weighted learning can achieve robustness as classifiers with truncated loss functions. We study the corresponding weighted learning for the angle-based SVM, and name them the one-step weighted angle-based SVM and iteratively weighted angle-based SVM as OWASVM and IWASVM, respectively.

3.2.1. One-step weighted angle-based classifiers—The standard large-margin angle-based classifiers use the same weights for all training data. However, when there are outliers far from their own classes, it will be ideal to give smaller weights to such outliers. One direct weighting strategy is to make use of the solution from standard angle-based learning, which can help to learn the relative locations of the training data points to the classification boundary.

The main problem of nonrobust classifiers is the unboundedness of the employed loss functions. To this end, we propose the weighting function which can help to obtain bounded target loss functions. For a general positive loss function ℓ , let the weighting function be η_ℓ and the target loss be $\ell^*(\cdot) = \eta_\ell(\cdot) \times \ell(\cdot)$. The proposed weighting function and target loss can be respectively expressed as

$$\eta_\ell(u) = \begin{cases} \ell(0) / \ell(u) & \text{if } u \leq 0, \\ \ell(0) / \ell(-u) & \text{if } u > 0, \end{cases} \quad \text{and} \quad \ell^*(u) = \begin{cases} \ell(0) & \text{if } u \leq 0, \\ \ell(u)\ell(0) / \ell(-u) & \text{if } u > 0. \end{cases}$$

One can check that the target loss is bounded above by $\ell(0)$.

For WAC1 (7), we first train the standard classifiers with all weights equal using a properly tuned regularization parameter λ . Denote the optimal solution as $\hat{f}_0(x)$. According to the weighting functions η_{ℓ_1} and η_{ℓ_2} , we can assign one-step weights for all training data. For all $i \in \{1, \dots, n\}$, set $w_{i,y_i} = \eta_{\ell_1}(\langle \hat{f}_0(x_i), \mathbf{W}_{y_i} \rangle)$ and $w_{i,j} = \eta_{\ell_2}(\langle \hat{f}_0(x_i), \mathbf{W}_j \rangle)$ for all $j \neq y_i$. In particular, for OWASVM1 (8), employing the weighting function for the hinge loss, we obtain the two types of weighting functions $w_{i,y_i} = (k-1) / \{k-1 + |\langle \hat{f}_0(x_i), \mathbf{W}_{y_i} \rangle|\}$ and $w_{i,j} = 1 / \{1 + |\langle \hat{f}_0(x_i), \mathbf{W}_j \rangle|\}$ for all $j \neq y_i$. We can express the weighting functions as $w^1(u) = (k-1)/(k-1 + |u|)$ and $w^2(u) = 1/(1 + |u|)$. To further illustrate the weighting functions, we display them for the hinge loss in Figure 1. One can see that the two weighted hinge losses are bounded above by $k-1$ and 1, respectively.

For MAC2 (9), by a similar procedure, we can design the corresponding weighting function for the loss function ℓ which takes the general functional margin into consideration. For the

hinge loss H_1 , the weighting function becomes $1/(1 + |u|)$. The figures of such a weighting function and the target loss function are very similar as before, and they are bounded above by 1. For OWASVM2 (10), the one-step weight for observation $(\mathbf{x}_i, y_i)$ is

$$w_i = 1 / \left\{ 1 + \min_{j \neq y_i} \left| \hat{f}_0(\mathbf{x}_i, \mathbf{W}_{y_i} - \mathbf{W}_j) \right| \right\} \text{ for all } i \in \{1, \dots, n\}.$$

The two OWASVM1 and OWASVM2 procedures involve two steps: the first step is to apply the standard learning to calculate all weights for the training instances, and the second step is to apply weighted learning for all observations. As we show in Section 4, our proposed choices of weights work very well in numerical examples.

3.2.2. Iteratively weighted angle-based classifiers—We now discuss the IWASVM procedures. For truncated loss (5), note that $T_s(u)$ is flat over $(-\infty, s)$ and $R_s(-u)$ is flat over $(-v, \infty)$. Wu and Liu [25] observed that the optimizer of (5) does not depend on any observation falling inside either $(-\infty, s)$ or $(-v, \infty)$. The corresponding conclusion for loss (6) is similar. We can develop the iteratively weighted learning for two types of truncated loss functions.

For the truncated loss function (5), consider the aforementioned truncated hinge loss $T_{(k-1)s}$ and R_s in Section 2, which was discussed and used by Zhang et al. [30]. These authors proposed to use DCA, denoted as DCA1, to solve the corresponding non-convex problem. One can find the optimizer only depends on such observations falling in $\{i: (k-1)s \leq \langle \mathbf{f}(\mathbf{x}_i), \mathbf{W}_{y_i} \rangle \leq k-1 \text{ or } -1 \leq \langle \mathbf{f}(\mathbf{x}_i), \mathbf{W}_j \rangle \leq -s, j \neq y_i\}$.

Motivated from the connection between robust truncated SVM and weighted SVM [26], we propose an alternative algorithm, namely IWASVM1, by iteratively updating the weights and solving the corresponding weighted convex SVM subproblems. The IWASVM1 algorithm can be summarized as follows:

Step 1: (Initialization): Solve the WASVM1 problem (8) with weights $w_{i,j} = 1$ for all $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, k\}$, and denote the optimal solution $\hat{\mathbf{f}}^{(0)}$.

Step 2: (Iteration): At iteration $t = 1$, for all $i \in \{1, \dots, n\}$, define weights as $w_{i,y_i}^{(t)} = 1$ if $(k-1)s \leq \langle \hat{\mathbf{f}}^{(t-1)}(\mathbf{x}_i), \mathbf{W}_{y_i} \rangle \leq k-1$ and 0 otherwise, and as $w_{i,j}^{(t)} = 1$ if $-1 \leq \langle \hat{\mathbf{f}}^{(t-1)}(\mathbf{x}_i), \mathbf{W}_j \rangle \leq -s$ and 0 otherwise for $j \neq y_i$. Solve the WASVM1 (8) with the updated weights $w_{i,j}^{(t)}$ and denote the corresponding optimizer $\hat{\mathbf{f}}^{(t)}$.

Step 3: (Convergence): Repeat Step 2 until convergence. We have convergence when $w_{i,j}^{(t)} = w_{i,j}^{(t-1)}$ for all $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, k\}$.

Note that in Step 2, the special weighting technique only selects some observations to be involved in the optimization problem. Thus the actual size of the variables is much smaller than nk and consequently it is much more efficient to solve the corresponding optimization problem.

As a remark, we note that both the DCA1 and the proposed IWASVM1 algorithm can solve the robust truncated hinge loss classifiers iteratively. From optimization theory, the algorithms of non-convex problems may only guarantee local optimizers. There may exist multiple local minima, and thus we cannot ensure that these two algorithms always have the same local optimizer. However, we can show their equivalence in terms of *fixed points* as stated in the following theorem. For an iterative algorithm such as the DCA1 and IWASVM1, a solution is called a *fixed point* if it satisfies the convergence criterion and no further iteration is needed.

Theorem 3. *The DCA1 and IWASVM1 algorithms are equivalent in terms of fixed points. Namely, the local solution of the DCA1 is a fixed point of the proposed IWASVM1 algorithm, and vice versa.*

For the truncated loss function (6), we focus on the truncated hinge loss function, where $T_s(u) = \min\{H_1(u), H(s)\}$. Note that T_s is flat over $(-\infty, s)$ and $(1, \infty)$ and is also non-convex. To solve the corresponding non-convex optimization problem, Zhang et al. [30] developed another iteratively difference of convex functions algorithm, namely DCA2. Note that only the observations within $[s, 1]$ are involved in the optimizer, and we can design the iteratively weighted learning for the problem solving in a similar manner. The algorithm, called IWASVM2, can be summarized as follows:

Step 1: (Initialization): Solve the WASVM2 problem (10) with weights $w_1 = \dots = w_n = 1$ and denote the optimal solution $\hat{f}^{(0)}$.

Step 2: (Iteration): At iteration $t \geq 1$ define, for each $i \in \{1, \dots, n\}$, weight $w_i^{(t)} = 1$ if $s \leq \min_{j \neq y_i} \left\langle \hat{f}^{(t-1)}(x_i), \mathbf{W}_{y_i} - \mathbf{W}_j \right\rangle \leq 1$ and 0 otherwise. Solve the WASVM2 (10) with the updated weights $w_i^{(t)}$ and denote the corresponding optimizer $\hat{f}^{(t)}$.

Step 3: (Convergence): Repeat Step 2 until convergence. We have convergence when $w_i^{(t)} = w_i^{(t-1)}$ for all $i \in \{1, \dots, n\}$.

Based on the weighting strategy in Step 2, the number of involved observations for optimization of (10) is less than n , and its computation can be more efficient. Analogously, we also establish the close connection between the DCA2 and the proposed IWASVM2 in the following theorem.

Theorem 4. *The DCA2 and IWASVM2 algorithms are equivalent in terms of fixed points. Namely, the local solution of the DCA2 is a fixed point of the proposed IWASVM2 algorithm, and vice versa.*

3.3. Truncated loss and weighted learning

In Section 2.3, we introduced two angle-based classifiers using robust truncated loss functions, which can be solved by the DCA1 and DCA2. For further illustrations, we also proposed two weighted angle-based SVMs in Section 3.1. Two different weighted strategies for each WASVM were discussed in Section 3.2, one is the one-step weighting algorithm

(OWASVM), and the other is the iterative weighting algorithm (IWASVM). As we have shown before, the IWASVM can be viewed as another algorithm for classifiers with non-convex truncated hinge loss functions. The DCA and the proposed IWASVM algorithms are equivalent to each other in terms of fixed points. They both involve several iterations, and thus take more computational time than the OWASVM.

We take an example of WASVM1 (8) for discussion, and the corresponding conclusions for WASVM2 (10) are similar. To further illustrate these two different weighted learning algorithms, we plot their weighting functions in Figure 2 for comparisons. Note that the weighting functions for the IWASVM1 are asymmetric step functions with value 1 or 0, and they jump in different intervals $[(k-1)s, k-1]$ and $[-1, -s]$, respectively. In contrast, the weighting functions for the OWASVM1 are smooth and convex on both positive and negative parts of the horizontal axis, and are symmetric about the vertical axis. Another important difference is that the OWASVM1 algorithm is a one-step procedure, and does not iterate. Both of these two types of weighting functions can guarantee the boundedness of target loss functions to achieve robust classifiers.

4. Numerical results

We use both simulated examples in Sections 4.1–4.2 and a real data example in Section 4.3 to investigate the behaviors of the proposed two weighted angle-based SVMs. We compare the performances of several methods ranging from the traditional SVMs to angle-based SVMs, and also include the non-weighted leaning and weighted learning.

For simplicity, we create some abbreviations for these methods. The methods considered include RMSVM in [17] and MSVM in [7], and they are both non-weighted and non-angle-based methods. For the proposed two types of weighted angle-based SVM (WASVM1, WASVM2), we consider the non-weighted angle-based SVM with equal weights, one-step weighting, iteratively weighting with the truncation location parameter $s = 0$ or $s = -1/(k-1)$, and denote them NWASVM1, OWASVM1, IWASVM1₀ and IWASVM1_k for WASVM1, respectively. The abbreviations NWASVM2, OWASVM2, IWASVM2₀ and IWASVM2_k for WASVM2 are defined in a similar manner. The two different truncation locations ($s \in \{0, -1/(k-1)\}$) can guarantee Fisher consistency for the corresponding WASVMs.

We proceed as in [17, 29, 30] and use $\gamma = 0.5$ to obtain best performance for the first type of SVMs. This can help to avoid extensive tuning of the convex combination parameter γ for efficient computation.

4.1. Linear example

Consider linear learning with a three-class response ($k = 3$), and conduct a series of example settings to demonstrate the numerical performances of our new adaptively weighted angle-based classifiers. We generate simulated datasets in the following manner. First, generate $(X_1, X_2)^T$ from the distribution $\Pr(Y = j) = 1/3$, $\Pr(X | Y = j) \sim \mathcal{N}(\mu_j, 0.16 \mathbf{I}_2)$ with $j \in \{1, 2, 3\}$, where $\mathbf{I}_2$ is the 2×2 identity matrix, and the μ_j s are set to be $(1, 0)^T$, $(-1/2, \sqrt{3}/2)^T$, $(-1/2, -\sqrt{3}/2)^T$.

$-\sqrt{3}/2)^T$ for the corresponding three classes. Second, denote the contamination percentage of the data be $prec \in \{5\%, 10\%\}$, and select the instances of $prec$ randomly and change their predictors and labels as outliers. For certain contaminated observations $(\mathbf{x}, y)$, the marginal distribution of $(X_1, X_2)^T$ is $\mathcal{N}((3, 0)^T, 0.16 \mathbf{I}_2)$, and we switch its label to one of the remaining classes $\{1, 2, 3\} \setminus y$ with equal probabilities. For the case $p > 2$, the remaining input $X_3, \dots, X_p$ are generated independently from $\mathcal{U}[-1, 1]$ as noise variables. We study the cases $p \in \{2, 20, 200\}$.

We simulate 100 observations for training, 1000 observations for tuning, and 10,000 observations for testing. We use the training set to build corresponding classifiers, then use the separate tuning set to choose the best λ via a grid search among the 30 different parameters. Once the tuning parameter is selected, we use the test set to evaluate the corresponding test error rates of the tuned classifiers. For each simulation setting, we run the experiment 100 times. In particular, we investigate the performance of iteratively weighted methods with truncations at $s = 0$ and $s = -1/(k - 1)$ to achieve Fisher consistency. Note that only the first two predictors contain useful classification signals, and the remaining are noise variables. We report the average misclassification error rates for each scenario in Table 1, and show the box plots of the classification results in Figure 3.

From Table 1 and Figure 3, one can see that the proposed WASVMs 1,2 perform better than the original SVMs. In addition, the iteratively WASVMs perform better than one-step WASVMs, which are better than the non-weighted SVMs and the original SVMs. When the dimension of predictors becomes higher, the misclassification errors get larger as well. When the percentage of contamination grows larger, the performance of all classifiers becomes worse. The IWASVMs with $s = 0$ and $s = -1/(k - 1)$ have similar classification accuracy. These two different iteratively weighted angle-based classifiers have competitive performance. Furthermore, to avoid extensive tuning of s and to obtain more stable results, we recommend to use $s = -1/(k - 1)$ in practice.

To compare the computational time used for different methods, we report average running time for one replication in Table 2. Since the IWASVMs involve iterations till convergence, while the OWASVMs only use one-step weighting, the IWASVMs take longer to compute than the OWSVMs. However, they are still relatively efficient to implement as shown in Table 2.

4.2. Nonlinear example

In this section, we demonstrate the performances of the proposed methods for nonlinear classification. We generate three classes with $\Pr(Y = j) = 1/3$ for all $j \in \{1, 2, 3\}$ on $\mathbb{R}^2$. Specifically,

$$\Pr(X|Y = 1) \sim \mathcal{N}[(2, 0)^T, 1.5^2 \mathbf{I}_2], \Pr(X|Y = 2) \sim \frac{1}{2} \mathcal{N}[(0, 2)^T, 1.5^2 \mathbf{I}_2] + \frac{1}{2} \mathcal{N}[(0, -2)^T, 1.5^2 \mathbf{I}_2]$$

and $\Pr(X | Y = 3) \sim \mathcal{N}[(- 2, 0)^T, 1.5^2 I_2]$. Here, we employ the Gaussian kernel $K(\mathbf{u}, \mathbf{v}) = \exp\{ - \|\mathbf{u} - \mathbf{v}\|_2^2 / (2\sigma^2) \}$. As to the second parameter σ of the Gaussian kernel, we use the median of the between-class pairwise Euclidean distances of training inputs to avoid extensive grid search.

The average misclassification error rates and running time over 100 repetitions are reported in Tables 3 and 4, respectively, and the box plots of classification results are shown in Figure 4. We can see that the angle-based methods perform better than non-angle-based ones, and the weighted methods perform better than unweighted classifiers as well. As the contamination percentage increases, all methods worsen to some degree, and we can also find the weighted angle-based methods have smaller misclassification error rates. The improvements of the OWASVM1,2 and IWASVM1,2 over the SVMs get smaller in terms of misclassification errors compared with the linear examples. One potential reason is that the nonlinear kernel learning enriches the flexibility of the classification functions. Thus, the effect of weighted learning on kernel learning can be smaller than that on linear learning. The implementation of iteratively weighted methods takes longer to compute than one-step weighted methods as shown in Table 4.

4.3. Real data

In this section, we apply the proposed methods to the Vehicle dataset. More information about the data can be found at the UCI Machine Learning Repository (<https://archive.ics.uci.edu/ml/datasets.html>). This dataset has a total of 846 observations with four classes and 18 integer-valued input variables. The numbers of four classes are 218, 212, 217 and 199, respectively. Before applying the methods, we standardize each input variable with mean 0 and standard deviation 1 and randomly split the dataset into training, tuning, and testing sets, with sizes 141, 282, and 423, correspondingly. We investigate WASVMs1, 2 under all equal unit weights, one-step weighting and iteratively weighing with truncation location $s = 0$ and $s = -1/3$. To further study robustness of the truncated hinge loss for both the training and tuning sets, we contaminate the class output by randomly choosing a certain percentage (*perc*) of their observations and changing the class label to one of the other class labels. We consider three different levels of contamination percentage, namely 0%, 5%, 10%. This allows us to show the potential robustness against outliers for all classifiers. For the analysis, linear learning is applied on this dataset, and the results of average misclassification error rates and computational time over 50 replications are summarized in Table 5.

From Table 5, we can see that when there is no data contamination, the non-weighted angle-based SVMs perform better than other classifiers. As the percentage of data contamination increases, the proposed IWASVM1,2 with $s = -1/3$ perform better than other methods. However, the IWASVM1,2 take more computational time when there are outliers. These results indicate that our weighted angle-based methods are more robust against outliers. This is consistent with the simulation results. In addition, the IWASVMs take longer time to compute than the OWASVMs.

5. Conclusion

In this article, we propose two unified robust truncated loss function classifiers under the angle-based framework to alleviate the impact of outliers. We give conditions for truncation of a general class of loss functions to achieve Fisher consistency. The corresponding non-convex optimization problems can be solved by DCA. As an alternative to non-convex optimization, we apply two different weighted extensions of the framework, the one-step weighted and iteratively weighted learning, which involve efficient convex optimization and can also gain robustness. We establish the equivalence between the weighted angle-based SVMs and the truncated hinge loss angle-based SVMs as well. Numerical results with simulated and real data demonstrate that the proposed weighted classifiers are very competitive, especially at the presence of outliers. In general, when the computational time is not a concern, we recommend to use the IWASVM1 since it appears to perform slightly better than IWASVM2 on the numerical examples.

Acknowledgments

The authors would like to thank the Editor-in-Chief, Christian Genest, the Associate Editor, and reviewers for their valuable comments and suggestions which led to a much improved presentation. Sheng Fu's research was supported in part by Chinese Scholarship Council. Yufeng Liu's research was supported in part by National Science Foundation Grant IIS1632951 and National Institute of Health Grant R01GM126550.

Appendix

Appendix A. Technical Details on Weighted Angle-based SVMs

For the WASVM1 (8), consider linear learning. We assume $f_q(x) = \beta_q^T x$ for each $q \in \{1, \dots, k-1\}$ with the intercepts included, where $\beta_q \in \mathbb{R}^{p+1}$ are the vectors of parameters of interest. For brevity, we denote $f(x)$ by Bx , where $B = (\beta_1, \dots, \beta_{k-1})^T$ is a $(k-1) \times (p+1)$ parameter matrix. We choose the ℓ_2 penalty term with the intercepts penalized [10], i.e., $J(f) = \sum_{q=1}^{k-1} \beta_q^T \beta_q / 2$. Solving problem (8) in its dual form, the corresponding dual problem becomes

$$\begin{aligned} \min_{\substack{\alpha_{i,j} \\ i \in \{1, \dots, n\} \\ j \in \{1, \dots, k\}}} \quad & \frac{1}{2n\lambda} \sum_{q=1}^{k-1} \left\| \sum_{i=1}^n \sum_{j=1}^k (-1)^{\mathbf{1}(y_i \neq j)} \alpha_{i,j} W_{j,q} x_i \right\|_2^2 - (k-1) \sum_{i=1}^n \alpha_{i,y_i} - \sum_{i=1}^n \sum_{j \neq y_i} \alpha_{i,j} \\ \text{subject to } & \forall_{i \in \{1, \dots, n\}} \forall_{j \in \{1, \dots, k\}} \quad 0 \leq \alpha_{i,j} \leq A_{i,j}, \end{aligned} \quad (\text{A.1})$$

where $W_{j,q}$ is the q th element of W_j , $A_{i,j} = w_{i,j} \{ \gamma \mathbf{1}(y_i = j) + (1 - \gamma) \mathbf{1}(y_i \neq j) \}$, and $\mathbf{1}$ is the indicator function.

Next, we briefly discuss nonlinear learning with kernel $K(\cdot, \cdot)$. To begin with, denote by K the corresponding kernel function, and by $\mathbf{K} = (K(x_i, x_{i'}))$ for all $i, i' \in \{1, \dots, n\}$ the Gram

matrix. Since $f_q \in (\{1\} + H_K)$ for all $q \in \{1, \dots, k-1\}$, we can write $f_q(x) = \theta_{q,0} + \sum_{i=1}^n \theta_{q,i} K(x_i, x)$ using the representer theorem [12]. To simplify calculations, we use the penalty term $J(f) = \sum_{q=1}^{k-1} \theta_q^T K \theta_q + \sum_{q=1}^{k-1} \theta_{q,0}^2$. Here $\theta_q = (\theta_{q,1}, \dots, \theta_{q,n})^T$ is the kernel product coefficient vector for all $q \in \{1, \dots, k-1\}$.

Without loss of generality, assume that the Gram matrix K is invertible. Then the optimization problem for kernel learning can be expressed as follows:

$$\begin{aligned} \min_{\substack{\alpha_{i,j} \\ i \in \{1, \dots, n\} \\ j \in \{1, \dots, k\}}} \quad & \frac{1}{2n\lambda} \sum_{q=1}^{k-1} \left\{ \sum_{i=1}^n \sum_{j=1}^k (-1)^{\mathbf{1}(y_i \neq j)} \alpha_{i,j} W_{j,q} \begin{pmatrix} 1 \\ \mathbf{K}_i \end{pmatrix} \right\}^T \begin{pmatrix} 1 & 0 \\ 0 & \mathbf{K}^{-1} \end{pmatrix} \times \left\{ \sum_{i=1}^n \sum_{j=1}^k (-1)^{\mathbf{1}(y_i \neq j)} \alpha_{i,j} W_{j,q} \begin{pmatrix} 1 \\ \mathbf{K}_i \end{pmatrix} \right\} - (k-1) \sum_{i=1}^n \alpha_{i,y_i} - \sum_{i=1}^n \sum_{j \neq y_i} \alpha_{i,j} \\ \text{subject to } & \forall_{i \in \{1, \dots, n\}} \forall_{j \in \{1, \dots, k\}} \quad 0 \leq \alpha_{i,j} \leq A_{i,j}, \end{aligned} \quad (\text{A.2})$$

where $\mathbf{K}_{i1}$ is the i_1 th column vector of K , and A_{ij} has the same definition as the linear case. Note that $\mathbf{K}^{-1} \mathbf{K}_{i2}$ is the i_2 th column of the identity matrix, and $\mathbf{K}_{i1} \mathbf{K}^{-1} \mathbf{K}_{i2} = K(x_{i1}, x_{i2})$. Therefore, we do not need to compute the inverse matrix $\mathbf{K}^{-1}$ in the optimization.

Problems (A.1) and (A.2) are standard quadratic programming with box constraints. They can be solved by the efficient coordinate descent algorithm [29].

For linear learning of WASVM2 (10), the symbols and notations here are the same as before. We solve the optimization problem (10) by its dual form, and the dual problem is

$$\begin{aligned} \min_{\substack{\alpha_{i,j} \\ i \in \{1, \dots, n\} \\ j \in \{1, \dots, k\}}} \quad & \frac{1}{2n\lambda} \sum_{q=1}^{k-1} \left\| \sum_{i=1}^n \sum_{j \neq y_i} \alpha_{i,j} (W_{y_i,q} - W_{j,q}) x_i \right\|_2^2 - \sum_{i=1}^n \sum_{j \neq y_i} \alpha_{i,j} \\ \text{subject to } & \forall_{i \in \{1, \dots, n\}} \sum_{j \neq y_i} \alpha_{i,j} \leq w_i, \quad \forall_{i \in \{1, \dots, n\}} \forall_{j \neq y_i} \alpha_{i,j} \geq 0. \end{aligned} \quad (\text{A.3})$$

For nonlinear learning of WASVM2 (10), we have the same symbols and notations as aforementioned. The corresponding dual problem is

$$\begin{aligned} \min_{\substack{\alpha_{i,j} \\ i \in \{1, \dots, n\} \\ j \in \{1, \dots, k\}}} \quad & \frac{1}{2n\lambda} \sum_{q=1}^{k-1} \left\{ \sum_{i=1}^n \sum_{j \neq y_i} \alpha_{i,j} (W_{y_i,q} - W_{j,q}) \begin{pmatrix} 1 \\ \mathbf{K}_i \end{pmatrix} \right\}^T \begin{pmatrix} 1 & 0 \\ 0 & \mathbf{K}^{-1} \end{pmatrix} \times \left\{ \sum_{i=1}^n \sum_{j \neq y_i} \alpha_{i,j} (W_{y_i,q} - W_{j,q}) \begin{pmatrix} 1 \\ \mathbf{K}_i \end{pmatrix} \right\} - \sum_{i=1}^n \sum_{j \neq y_i} \alpha_{i,j} \\ \text{subject to } & \forall_{i \in \{1, \dots, n\}} \forall_{j \neq y_i} \alpha_{i,j} \geq 0. \end{aligned} \quad (\text{A.4})$$

The optimization problems (A.3) and (A.4) both involve quadratic programming with box constraints and linear inequalities. Due to the block separability structure of the linear inequality constraints, we can solve them by the block coordinate descent method [19].

Appendix B. Proof of Theorem 1

The proof of Theorem 1 can be decomposed into several steps as follows.

Lemma 1. Suppose we have an arbitrary $\mathbf{f} \in \mathbb{R}^{k-1}$. For any $u, v \in \{1, \dots, k\}$ such that $u \neq v$, define $\mathbf{T}_{u,v} = \mathbf{W}_u - \mathbf{W}_v$. For any scalar $z \in \mathbb{R}$, $\langle \mathbf{f} + z\mathbf{T}_{u,v}, \mathbf{W}_w \rangle = \langle \mathbf{f}, \mathbf{W}_w \rangle$, where $w \in \{1, \dots, k\}$ and $w \neq u, v$. Furthermore, we have that $\langle \mathbf{f} + z\mathbf{T}_{u,v}, \mathbf{W}_u \rangle - \langle \mathbf{f}, \mathbf{W}_u \rangle = -\langle \mathbf{f} + z\mathbf{T}_{v,u}, \mathbf{W}_v \rangle + \langle \mathbf{f}, \mathbf{W}_v \rangle$.

The proof of Lemma 1 can be found in Zhang and Liu [28]. Note that

$$\begin{aligned} L(\mathbf{f}, \mathbf{x}) &= \mathbb{E}\{\phi_1^t(\mathbf{f}(\mathbf{X}), Y | \mathbf{X} = \mathbf{x})\} = \sum_{j=1}^k P_j \left\{ \gamma T_s(\langle \mathbf{f}, \mathbf{W}_j \rangle) + (1-\gamma) \sum_{i \neq j} R_v(-\langle \mathbf{f}, \mathbf{W}_i \rangle) \right\} \\ &= \sum_{j=1}^k P_j \left\{ \gamma T_s(\langle \mathbf{f}, \mathbf{W}_j \rangle) - (1-\gamma) R_v(-\langle \mathbf{f}, \mathbf{W}_j \rangle) \right\} + (1-\gamma) \sum_{j=1}^k R_v(-\langle \mathbf{f}, \mathbf{W}_j \rangle) \\ &= \sum_{j=1}^k P_j h_\gamma(\langle \mathbf{f}, \mathbf{W}_j \rangle) + (1-\gamma) \sum_{j=1}^k R_v(-\langle \mathbf{f}, \mathbf{W}_j \rangle), \end{aligned}$$

where $h_\gamma(u) \triangleq \gamma T_s(u) - (1-\gamma) R_v(-u)$. By definition, we know that $h_\gamma(u)$ is nonincreasing. Denote the optimal minimizer for $\mathcal{L}(\mathbf{f}, \mathbf{x})$ as $\mathbf{f}^*$.

Lemma 2. Suppose $P_1 > P_j$ for any $j \neq 1$, we have $h_\gamma(\langle \mathbf{f}^*, \mathbf{W}_1 \rangle) = h_\gamma(\langle \mathbf{f}^*, \mathbf{W}_j \rangle)$.

Proof. If the conclusion is not true, there must exist $h_\gamma(\langle \mathbf{f}^*, \mathbf{W}_1 \rangle) > h_\gamma(\langle \mathbf{f}^*, \mathbf{W}_j \rangle)$ for some $j \neq 1$. Then we can define $\mathbf{f}'(\mathbf{x}) \in \mathbb{R}^{k-1}$ such that $\langle \mathbf{f}', \mathbf{W}_j \rangle = \langle \mathbf{f}^*, \mathbf{W}_1 \rangle$ and $\langle \mathbf{f}', \mathbf{W}_1 \rangle = \langle \mathbf{f}^*, \mathbf{W}_j \rangle$. The existence of such $\mathbf{f}'$ is guaranteed by Lemma 1. One can verify that $\mathbf{f}'$ is the minimizer of $\mathcal{L}(\mathbf{f}, \mathbf{x})$, not $\mathbf{f}^*$. This is a contradiction to the definition of $\mathbf{f}^*$. $\square$

Lemma 3. Suppose $P_1 > P_j$ for any $j \neq 1$, we have $\mathbf{f}^* = \mathbf{0}$.

Proof. We can easily check that $P_1 > 1/k$. It is sufficient to show that $\mathcal{L}(\mathbf{0}, \mathbf{x})$ is not a minimizer of L . Let $a > 0$ be small enough. From Lemma 1, we can set a $\mathbf{f}'$ such that $\langle \mathbf{f}', \mathbf{W}_j \rangle = -a$ for any $j \neq 1$ and $\langle \mathbf{f}', \mathbf{W}_1 \rangle = (k-1)a$. When $a \downarrow 0$, one can see that $\mathbf{f}' \rightarrow \mathbf{0}$. We consider the following four cases to complete the proof.

1. If $s = v = 0$, we have

$$\ell(\mathbf{f}', \mathbf{x}) = \gamma P_1 \ell_1\{(k-1)a\} + \gamma(1-P_1) \ell_1(0) + (1-\gamma)(1-P_1) \ell_2(0) + (1-\gamma)(k-2+P_1) \ell_2(a).$$

$$\text{Then } dL(\mathbf{f}', \mathbf{x}) \Big|_{a=0_+} da = \gamma P_1(k-1) \ell'_1(0) + (1-\gamma)(k-2+P_1) \ell'_2(0) < 0.$$

2. If $s = 0, v < 0$, we have

$$\ell(f', x) = \gamma P_1 \ell_1\{(k-1)a\} + \gamma(1-P_1)\ell_1(0) + (1-\gamma)(1-P_1)\ell_2\{-(k-1)a\} \\ + (1-\gamma)(k-2+P_1)\ell_2(a).$$

$$\text{Then } dL(f', x)/da \Big|_{a=0_+} = \gamma P_1(k-1)\ell'_1(0) + (1-\gamma)(kP_1-1)\ell'_2(0) < 0.$$

3. If $s < 0$, $v = 0$, we have

$$\ell(f', x) = \gamma P_1 \ell_1\{(k-1)a\} + \gamma(1-P_1)\ell_1(-a) + (1-\gamma)(1-P_1)\ell_2(0) + (1-\gamma)(k-2+P_1)\ell_2(a).$$

$$\text{Then } dL(f', x)/da \Big|_{a=0_+} = \gamma(kP_1-1)\ell'_1(0) + (1-\gamma)(k-2+P_1)\ell'_2(0) < 0.$$

4. If $s < 0$, $v < 0$, we have

$$\ell(f', x) = \gamma P_1 \ell_1\{(k-1)a\} + \gamma(1-P_1)\ell_1(-a) + (1-\gamma)(1-P_1)\ell_2\{-(k-1)a\} + (1-\gamma)(k-2 \\ + P_1)\ell_2(a).$$

$$\text{Then } dL(f', x) / da \Big|_{a=0_+} = (kP_1-1)\{\gamma\ell'_1(0) + (1-\gamma)\ell'_2(0)\} < 0.$$

The above four cases indicate that $\ell(f', x)$ has a direction of decrease at $f = \mathbf{0}$. $\square$

From Lemma 3, we obtain that the k inner products $\langle f^*, W_1 \rangle, \dots, \langle f^*, W_k \rangle$ may have different signs.

Lemma 4. Suppose $P_1 > P_j$ for any $j = 1, \dots, k$, we have $\langle f^*, W_1 \rangle > \langle f^*, W_j \rangle$.

Proof. We argue by contradiction. Suppose there exists $i = 1, \dots, k$ such that $\langle f^*, W_1 \rangle < \langle f^*, W_i \rangle$. According to the monotone property of $h_\gamma(u)$, we have $h_\gamma(\langle f^*, W_1 \rangle) < h_\gamma(\langle f^*, W_i \rangle)$. Based on the conclusion in Lemma 2, then $h_\gamma(\langle f^*, W_1 \rangle) = h_\gamma(\langle f^*, W_i \rangle)$. We consider the following cases.

1. Case $\gamma = 1$, $h_\gamma(u) = T_s(u) = \min\{\ell_1(s), \ell_1(u)\}$. Then $h_\gamma(u)$ is flat in $(-\infty, s]$, and its maximum is $\ell_1(s)$. Thus $\langle f^*, W_1 \rangle < \langle f^*, W_i \rangle \leq s$, and $h_\gamma(\langle f^*, W_j \rangle) = \ell_1(s)$ for all $j \in \{1, \dots, k\}$. We have all $\langle f^*, W_j \rangle \leq s$. From Lemma 3, we know $f^* = \mathbf{0}$, then all $\langle f^*, W_j \rangle < 0$. This contradicts the sum-to-zero property.
2. Case $\gamma = 0$, $h_\gamma(u) = -R_v(-u) = \max\{-\ell_2(-v), -\ell_2(-u)\}$. Then $h_\gamma(u)$ is flat in $[-v, \infty)$ and its minimum is $-\ell_2(-v)$. Thus $0 \leq -v < \langle f^*, W_1 \rangle < \langle f^*, W_i \rangle$. There exists some $q = 1, \dots, k$ with $\langle f^*, W_q \rangle < 0$, because of the sum-to-zero constraint. Now one can increase $\langle f^*, W_1 \rangle$ by a small amount, and decrease $\langle f^*, W_q \rangle$ by the same amount (Lemma 1), and this would decrease the conditional loss $\ell(f, x)$ at f^* .
3. Case $\gamma \in (0, 1)$. Then $h_\gamma(u)$ is a strictly monotone decreasing function. According to the assumption $h_\gamma(\langle f^*, W_1 \rangle) = h_\gamma(\langle f^*, W_i \rangle)$, one has $\langle f^*, W_1 \rangle = \langle f^*, W_i \rangle = f_0$. We can define $f^{**} \in \mathbb{R}^{k-1}$ with

$$\langle f^{**}, \mathbf{w}_1 \rangle = f_0 + \epsilon, \langle f^{**}, \mathbf{w}_i \rangle = f_0 - \epsilon; \langle f^{**}, \mathbf{w}_j \rangle = \langle f^*, \mathbf{w}_j \rangle,$$

for $j = 1, i$, where $\epsilon > 0$ is sufficiently small. Set $L(f^{**}, \mathbf{x}) - L(f^*, \mathbf{x}) = A$. Then we have

- a) $f_0 = s, A = \{\gamma P_1 \ell'_1(s) + (1 - \gamma)(P_1 - P_i) \ell'_2(-s)\} \in +o(\epsilon) < 0;$
- b) $f_0 = -v, A = \{\gamma(P_1 - P_i) \ell'_1(-v) + (1 - \gamma)(1 - P_i) \ell'_2(v)\} \in +o(\epsilon) < 0;$
- c) $f_0 \neq s, f_0 \neq -v, A = (P_1 - P_i)\{\gamma \ell'_1(f_0) + (1 - \gamma) \ell'_2(-f_0)\} \in +o(\epsilon) < 0$

Thus, we obtain that $\langle f^*, \mathbf{w}_1 \rangle > \langle f^*, \mathbf{w}_j \rangle$ for any $j = 1$ with $\gamma \in [0, 1]$, $s = 0$ and $v = 0$. $\square$

Lemmas 1–4 together imply Fisher consistency for the first truncated large-margin angel-based classifiers.

Appendix C. Proof of Theorem 2

Note that

$$L(f, \mathbf{x}) = \mathbb{E}\{\phi_2^t(f(X), Y) | X = \mathbf{x}\} = \sum_{j=1}^k P_j T_s(g_j),$$

where $g_j = \min_q \langle f(\mathbf{x}), \mathbf{w}_j - \mathbf{w}_q \rangle$. By definition and the fact that $\sum_{j=1}^k \langle f, \mathbf{w}_j \rangle = 0$, we can conclude that $\max(g_1, \dots, g_k) = 0$ and at most one of the g_j s is positive. Assume that $j_p = \arg\max_{j \in \{1, \dots, k\}} P_j$ is unique. Then using the nonincreasing property of T_s and $\ell'_1(0) < 0$, the minimizer f^* satisfies that $g_{j_p}^* \geq 0$.

We are now left to show $g_{j_p}^* \neq 0$ or, equivalently, that $\mathbf{0}$ cannot be a minimizer. Without loss of generality, assume that $P_{j_p} > 1/k$. Then it is sufficient to show that there exists a solution with $g_{j_p}^* > 0$. By assumption, there exists $u > 0$ such that $\{\ell'_1(0) - \ell'_1(u)\} / \{T_s(-u) - \ell'_1(0)\} < k - 1$. Consider a solution f^0 with $\langle f^0(\mathbf{x}), \mathbf{w}_{j_p} \rangle = (k - 1)u/k$ and $\langle f^0(\mathbf{x}), \mathbf{w}_j \rangle = -u/k$ for $j \neq j_p$. We want to show that f^0 yields a smaller expected loss than $\mathbf{0}$, i.e., $P_{j_p} \ell'_1(u) + (1 - P_{j_p}) T_s(-u) < \ell'_1(0)$. Equivalently, $\{\ell'_1(0) - \ell'_1(u)\} / \{T_s(-u) - \ell'_1(0)\} > (1 - P_{j_p}) / P_{j_p}$, which holds due to the fact that $(1 - P_{j_p}) / P_{j_p} < k - 1$. This implies sufficiency of the condition.

To prove necessity of the condition, it is sufficient to show that if $\{\ell'_1(0) - \ell'_1(u)\} / \{T_s(-u) - \ell'_1(0)\} < k - 1$ for all $u > 0$, then $\mathbf{0}$ is a minimizer of $\sum_{j=1}^k P_j T_s(g_j)$. Equivalently, we need to show that there exists $(P_1, \dots, P_k)$ such that $\sum_{j=1}^k P_j T_s(g_j) \geq \ell'_1(0)$ for all f . Without loss of generality, assume that $\langle f, \mathbf{w}_1 \rangle = \dots = \langle f, \mathbf{w}_k \rangle$. Then

$$\sum_{j=1}^k P_j T_s(g_j) = \sum_{j=1}^{k-1} P_j T_s(\langle f, \mathbf{w}_j - \mathbf{w}_k \rangle) + P_k T_s(\langle f, \mathbf{w}_k - \mathbf{w}_{k-1} \rangle) \geq (1 - P_k) T_s(\langle f, \mathbf{w}_{k-1} - \mathbf{w}_k \rangle) + P_k T_s(\langle f, \mathbf{w}_k - \mathbf{w}_{k-1} \rangle),$$

because $T_s(u) = \min\{\ell(u), \ell(s)\}$ is nonincreasing. Thus it is sufficient to show that $P_k \ell(u) + (1 - P_k) T_s(-u) > \ell(0)$ for all $u > 0$, i.e., $(1 - P_k)\{T_s(-u) - \ell(0)\} > P_k\{\ell(u) - \ell(0)\}$. By assumption, we can set $T_s(-u) - \ell(0) = \{\ell(0) - \ell(u)\}/(k-1) + a$ for some $a > 0$. Denote $\ell(0) - \ell(u) = A$. Then we need to have $(1 - P_k)\{A/(k-1) + a\} > P_k A$. Let $P_k = 1/k + \epsilon$. Then $\{(k-1)/k - \epsilon\}\{A/(k-1) + a\} > (1/k + \epsilon)A$, or equivalently,

$$a \frac{k-1}{k\epsilon} > \frac{k}{k-1}A + a. \quad (\text{C.1})$$

For any given $a > 0$ and $A > 0$, we can always find a small $\epsilon > 0$ to satisfy (C.1). The desired results then follow.

Appendix D. Proof of Theorem 3

Since linear learning can be achieved by using the linear kernel, it is sufficient to prove Theorem 3 under a general kernel setting. For the kernel $K(\cdot, \cdot)$, let $\mathbf{K} = (K(\mathbf{x}_i, \mathbf{x}_j))$ be an $n \times n$ matrix, and $\tilde{\mathbf{K}}_i = (1, K(\mathbf{x}_1, \mathbf{x}_i), \dots, K(\mathbf{x}_n, \mathbf{x}_i))^T \in \mathbb{R}^{n+1}$ for all $i \in \{1, \dots, n\}$. Denote $\boldsymbol{\theta}_q = (\theta_{q,0}, \theta_{q,1}, \dots, \theta_{q,n})^T$ be the coefficients of interest for all $q \in \{1, \dots, k-1\}$, and $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_{k-1})^T \in \mathbb{R}^{(k-1) \times (n+1)}$. The kernel learning for the truncated hinge loss angle-based SVM (A. 2) solves

$$\min_{\boldsymbol{\Theta}} \frac{1}{n} \sum_{i=1}^n \left\{ \gamma T_{(k-1)s}(\langle \boldsymbol{\Theta} \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle) + (1 - \gamma) \sum_{j \neq y_i} R_s(-\langle \boldsymbol{\Theta} \tilde{\mathbf{K}}_i, \mathbf{w}_j \rangle) \right\} + \frac{\lambda}{2} \sum_{q=1}^{k-1} \boldsymbol{\theta}_q^T \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{K} \end{pmatrix} \boldsymbol{\theta}_q. \quad (\text{D.1})$$

1)

The DCA1 for (D.1) includes three steps, as follows.

Step 1: Initialize by setting the initial solution $\hat{\boldsymbol{\Theta}}^{(0)}$ to be the solution of

$$\min_{\boldsymbol{\Theta}} \frac{1}{n} \sum_{i=1}^n \left\{ \gamma H_{k-1}(\langle \boldsymbol{\Theta} \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle) + (1 - \gamma) \sum_{j \neq y_i} H_1(-\langle \boldsymbol{\Theta} \tilde{\mathbf{K}}_i, \mathbf{w}_j \rangle) \right\} + \frac{\lambda}{2} \sum_{q=1}^{k-1} \boldsymbol{\theta}_q^T \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{K} \end{pmatrix} \boldsymbol{\theta}_q.$$

Step 2: At the i th iteration, solve

$$\begin{aligned}
& \min_{\Theta} \frac{1}{n} \sum_{i=1}^n \quad (D.2) \\
& \left[\gamma \left[H_{k-1}(\langle \Theta \tilde{K}_i, \mathbf{w}_{y_i} \rangle) + \mathbf{1}\{\langle \widehat{\Theta}^{(t-1)} \tilde{K}_i, \mathbf{w}_{y_i} \rangle < (k-1)s\} \langle \Theta - \widehat{\Theta}^{(t-1)} \tilde{K}_i, \mathbf{w}_{y_i} \rangle \right] \right. \\
& \left. + (1-\gamma) \sum_{j \neq y_i} \left[H_1(-\langle \Theta \tilde{K}_i, \mathbf{w}_j \rangle) - \mathbf{1}\{\langle \widehat{\Theta}^{(t-1)} \tilde{K}_i, \mathbf{w}_j \rangle > -s\} \langle \Theta - \widehat{\Theta}^{(t-1)} \tilde{K}_i, \mathbf{w}_j \rangle \right] \right] \\
& + \frac{\lambda}{2} \sum_{q=1}^{k-1} \theta_q^T \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{K} \end{pmatrix} \theta_q,
\end{aligned}$$

and denote its solution by $\widehat{\Theta}^{(t)}$.

Step 3: Repeat Step 2 until convergence. We have convergence when

$$\mathbf{1}\{\langle \widehat{\Theta}^{(t)} \tilde{K}_i, \mathbf{w}_{y_i} \rangle < (k-1)s\} \text{ and } \mathbf{1}\{\langle \widehat{\Theta}^{(t)} \tilde{K}_i, \mathbf{w}_j \rangle > -s\}$$

with $j \neq y_i$ for all $i \in \{1, \dots, n\}$, do not change.

The corresponding three steps of the IWASVM1 for (D.1) are as follows.

Step 1: Same as Step 1 of the above DCA1.

Step 2: At the t th iteration, solve

$$\begin{aligned}
& \min_{\Theta} \frac{1}{n} \sum_{i=1}^n \left\{ \gamma w_{i,y_i}^{(t)} H_{k-1}(\langle \Theta \tilde{K}_i, \mathbf{w}_{y_i} \rangle) + (1-\gamma) \sum_{j \neq y_i} w_{i,j}^{(t)} H_1(-\langle \Theta \tilde{K}_i, \mathbf{w}_j \rangle) \right\} \quad (D.3) \\
& + \frac{\lambda}{2} \sum_{q=1}^{k-1} \theta_q^T \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{K} \end{pmatrix} \theta_q,
\end{aligned}$$

with weight $w_{i,y_i}^{(t)} = 1$ if $(k-1)s \leq \langle \widehat{\Theta}^{(t-1)} \tilde{K}_i, \mathbf{w}_{y_i} \rangle \leq k-1$ and 0 otherwise, and

$w_{i,j}^{(t)} = 1 (j \neq y_i)$ if $-1 \leq \langle \widehat{\Theta}^{(t-1)} \tilde{K}_i, \mathbf{w}_j \rangle \leq -s$ and 0 otherwise. Denote the solution of (D.3) by $\widehat{\Theta}^{(t)}$.

Step 3: Repeat Step 2 until convergence. Here the procedure stops when $w_{i,j}^{(t)} = w_{i,j}^{(t-1)}$ for all $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, k\}$.

We first prove that the local solution of the DCA1 is a fixed point for the IWASVM1. To be more precise, by a fixed point, we mean that the local solution of the DCA1 satisfies the

convergence criterion of the IWASVM1. Denote the DCA1 solution at convergence by $\widehat{\Theta}^{DCA1}$. Define

$$\mathcal{L}_1 = \{i: \langle \widehat{\Theta}^{DCA1} \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle < (k-1)s\}, \quad \mathcal{R}_1 = \{i: \langle \widehat{\Theta}^{DCA1} \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle > k-1\},$$

and

$$\mathcal{L}_2 = \{(i, j): \langle \widehat{\Theta}^{DCA1} \tilde{\mathbf{K}}_i, \mathbf{w}_j \rangle < -1, j \neq y_i\}, \quad \mathcal{R}_2 = \{(i, j): \langle \widehat{\Theta}^{DCA1} \tilde{\mathbf{K}}_i, \mathbf{w}_j \rangle > -s, j \neq y_i\}.$$

At the convergence step, for any $i \in \mathcal{L}_1$, $H_{k-1}(\langle \Theta \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle) = k-1 - \langle \Theta \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle$ in a small neighborhood of the optimal solution $\widehat{\Theta}^{DCA1}$ of (D.2). This implies that

$$H_{k-1}(\langle \Theta \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle) + \mathbf{1}(\langle \widehat{\Theta}^{(t-1)} \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle < (k-1)s) \langle \Theta - \widehat{\Theta}^{(t-1)} \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle$$

reduces to a constant in a small neighborhood of the optimal solution $\widehat{\Theta}^{DCA1}$ and does not contribute to the objective function for $i \in \mathcal{L}_1$ while solving optimization problem (D.2).

Similarly, at convergence, for any $i \in \mathcal{R}_1$, $H_{k-1}(\langle \Theta \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle) = 0$, in a small neighborhood of the optimal solution. Thus

$$H_{k-1}(\langle \Theta \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle) + \mathbf{1}(\langle \widehat{\Theta}^{(t-1)} \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle < (k-1)s) \langle \Theta - \widehat{\Theta}^{(t-1)} \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle = 0.$$

It does not contribute the optimization (D.2) for $i \in \mathcal{R}_1$ at the convergence step since

$$\mathbf{1}(\langle \widehat{\Theta}^{(t-1)} \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle < (k-1)s) = 0 \text{ for } i \in \mathcal{R}_1 \text{ due to the definition of } \mathcal{R}_1.$$

Analogously, based on the property of the truncated hinge loss, we can also claim that for all $(i, j) \in \mathcal{L}_2 \cup \mathcal{R}_2$,

and they do not contribute the optimization (D.2). To summarize, at the convergence step of the DCA1, (D.2) reduces to

$$\min_{\Theta} \frac{1}{n} \left[\gamma \sum_{i \notin \mathcal{L}_1 \cup \mathcal{R}_1} \left[H_{k-1}(\langle \Theta \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle) + \mathbf{1}(\langle \widehat{\Theta}^{(t-1)} \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle < (k-1)s) \langle \Theta - \widehat{\Theta}^{(t-1)} \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle \right] \right. \\ \left. + (1-\gamma) \sum_{(i,j) \notin \mathcal{L}_2 \cup \mathcal{R}_2} \left[H_1(-\langle \Theta \tilde{\mathbf{K}}_i, \mathbf{w}_j \rangle) - \mathbf{1}(\langle \widehat{\Theta}^{(t-1)} \tilde{\mathbf{K}}_i, \mathbf{w}_j \rangle > -s) \langle \Theta - \widehat{\Theta}^{(t-1)} \tilde{\mathbf{K}}_i, \mathbf{w}_j \rangle \right] \right] \\ + \frac{\lambda}{2} \sum_{q=1}^{k-1} \theta_q^T \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{K} \end{pmatrix} \theta_q,$$

which is equivalent to

$$\min_{\Theta} \frac{1}{n} \left\{ \gamma \sum_{i \in \mathcal{L}_1 \cup \mathcal{R}_1} H_{k-1}(\langle \Theta \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle) + (1-\gamma) \sum_{(i,j) \in \mathcal{L}_2 \cup \mathcal{R}_2} H_1(-\langle \Theta \tilde{\mathbf{K}}_i, \mathbf{w}_j \rangle) \right\} \quad (\text{D.4})$$

$$+ \frac{\lambda}{2} \sum_{q=1}^{k-1} \theta_q^T \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{K} \end{pmatrix} \theta_q,$$

as $\mathbf{1}\{\langle \widehat{\Theta}^{(t-1)} \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle < (k-1)s\} = 0$, for any $i \notin \mathcal{L}_1$, and $\mathbf{1}\{\langle \widehat{\Theta}^{(t-1)} \tilde{\mathbf{K}}_i, \mathbf{w}_j \rangle > -s\} = 0$ for any $(i, j) \notin \mathcal{R}_2$ at the convergence. Note that (D.4) is equivalent to assigning w_{i,y_i} weight 1 for observation $i \notin \mathcal{L}_1 \cup \mathcal{R}_1$ and 0 otherwise, and $w_{i,j}$ weight 1 for observation $(i, j) \notin \mathcal{L}_2 \cup \mathcal{R}_2$ and 0 otherwise, which is exactly the same as our IWASVM1. Moreover, these weights do not change at the convergence of DCA1 and thus satisfy the convergence criterion of the IWASVM1 algorithm as well. This establishes the first part of the theorem.

Next, we try to prove the second part of the theorem, i.e., the local solution of the DCA1 is a fixed point for the IWASVM1. Denote the IWASVM1 solution at convergence by $\widehat{\Theta}^{\text{IWASVM1}}$. Using a similar argument above, we can show that, in a small neighborhood of the optimal solution $\widehat{\Theta}^{\text{IWASVM1}}$ of (D.3), $w_{i,y_i}^{(t)} H_{k-1}(\langle \Theta \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle)$ is same as

$$H_{k-1}(\langle \Theta \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle) + \mathbf{1}\{\langle \widehat{\Theta}^{(t-1)} \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle < (k-1)s\} \langle \Theta - \widehat{\Theta}^{(t-1)} \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle$$

up to a constant, $w_{i,j}^{(t)} H_1(-\langle \Theta \tilde{\mathbf{K}}_i, \mathbf{w}_j \rangle)$ is same as

$$H_1(-\langle \Theta \tilde{\mathbf{K}}_i, \mathbf{w}_j \rangle) - \mathbf{1}\{\langle \widehat{\Theta}^{(t-1)} \tilde{\mathbf{K}}_i, \mathbf{w}_j \rangle < (k-1)s\} \langle \Theta - \widehat{\Theta}^{(t-1)} \tilde{\mathbf{K}}_i, \mathbf{w}_j \rangle$$

up to a constant, and thus we have the equivalency between (D.3) and (D.2). Furthermore, at the convergence of the IWASVM1, the sets $\{i : (k-1)s \leq \langle \widehat{\Theta}^{(t)} \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} \rangle \leq k-1\}$ and

$\{(i, j) : -1 \leq \langle \widehat{\Theta}^{(t)} \tilde{\mathbf{K}}_i, \mathbf{w}_j \rangle \leq -s, j \neq y_i\}$ do not change, consequently $\widehat{\Theta}^{\text{IWASVM1}}$ satisfies the convergence criterion of the DCA1. This completes the proof.

Appendix E. Proof of Theorem 4

We use the same symbols and notations as the proof of Theorem 3. Considering the kernel learning for truncated hinge loss angle-based SVM (A.4), it solves

$$\min_{\Theta} \frac{\lambda}{2} \sum_{q=1}^{k-1} \theta_q^T \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{K} \end{pmatrix} \theta_q + \frac{1}{n} \sum_{i=1}^n T_s \left(\min_{j \neq y_i} \left\langle \Theta \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} - \mathbf{w}_j \right\rangle \right). \quad (\text{E.1})$$

The DCA2 for (E.1) includes three steps.

Step 1: Initialize by setting the initial solution $\widehat{\Theta}^{(0)}$ to be the solution of

$$\min_{\Theta} \frac{\lambda}{2} \sum_{q=1}^{k-1} \theta_q^T \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{K} \end{pmatrix} \theta_q + \frac{1}{n} \sum_{i=1}^n H_1 \left(\min_{j \neq y_i} \left\langle \Theta \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} - \mathbf{w}_j \right\rangle \right).$$

Step 2: At the t th iteration, solve

$$\begin{aligned} \min_{\Theta} \quad & \frac{1}{n} \sum_{i=1}^n \left\{ H_1 \left(\min_{j \neq y_i} \left\langle \Theta \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} - \mathbf{w}_j \right\rangle \right) \right. \\ & \left. - \mathbf{1} \left(\min_{j \neq y_i} \left\langle \widehat{\Theta}^{(t-1)} \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} - \mathbf{w}_j \right\rangle < s \right) \left\langle G_i(\widehat{\Theta}^{(t-1)}), \Theta - \widehat{\Theta}^{(t-1)} \right\rangle_F \right\} \\ & + \frac{\lambda}{2} \sum_{q=1}^{k-1} \theta_q^T \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{K} \end{pmatrix} \theta_q, \end{aligned} \quad (\text{E.2})$$

where $G_i(\Theta) = \partial H_s(\min_{j \neq y_i} \left\langle \Theta \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} - \mathbf{w}_j \right\rangle) / \partial \Theta$ is the subgradient matrix, and $\langle \cdot, \cdot \rangle_F$ is the Frobenius product of two matrices. Denote its solution by $\widehat{\Theta}^{(t)}$.

Step 3: Repeat Step 2 until convergence. Convergence is reached when

$$\mathbf{1} \left\{ \min_{j \neq y_i} \left\langle \widehat{\Theta}^{(t-1)} \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} - \mathbf{w}_j \right\rangle < s \right\}$$

does not change for all $i \in \{1, \dots, n\}$.

The corresponding three steps of the IWASVM2 for (E.1) are:

Step 1: Same as Step 1 of the above DCA2.

Step 2: At the t th iteration, solve

$$\min_{\Theta} \quad \frac{1}{n} \sum_{i=1}^n w_i^{(t)} H_1 \left(\min_{j \neq y_i} \left\langle \Theta \tilde{\mathbf{K}}_i, \mathbf{w}_{y_i} - \mathbf{w}_j \right\rangle \right) + \frac{\lambda}{2} \sum_{q=1}^{k-1} \theta_q^T \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{K} \end{pmatrix} \theta_q, \quad (\text{E.3})$$

with weight $w_i^{(t)} = 1$ if $s \leq \min_{j \neq y_i} \left| \Theta^{(t-1)} \tilde{\mathbf{K}}_i \mathbf{w}_{y_i} - \mathbf{w}_j \right| \leq 1$ and 0 otherwise. Denote the solution of (E.3) by $\hat{\Theta}^{(t)}$.

Step 3: Repeat Step 2 until convergence. Here we have the convergence when $w_i^{(t)} = w_i^{(t-1)}$ for all $i \in \{1, \dots, n\}$.

We first prove that the local solution of the DCA2 is a fixed point for the IWASVM2.

Denote the DCA2 solution at convergence by $\hat{\Theta}^{DCA2}$. Define

$$\mathcal{L} = \left\{ i : \min_{j \neq y_i} \left| \hat{\Theta}^{DCA2} \tilde{\mathbf{K}}_i \mathbf{w}_{y_i} - \mathbf{w}_j \right| < s \right\}, \quad \mathcal{R} = \left\{ i : \min_{j \neq y_i} \left| \hat{\Theta}^{DCA2} \tilde{\mathbf{K}}_i \mathbf{w}_{y_i} - \mathbf{w}_j \right| > 1 \right\}.$$

Similarly, according to similar discussions in the proof of Theorem 3, for any $i \in \mathcal{L} \cup \mathcal{R}$, we can also claim that they do not contribute to the optimization (E.2). To summarize, at the convergence step of the DCA2, (E.2) reduces to

$$\min_{\Theta} \frac{1}{n} \sum_{i \notin \mathcal{L} \cup \mathcal{R}} \left\{ H_1(\min_{j \neq y_i} \left| \Theta \tilde{\mathbf{K}}_i \mathbf{w}_{y_i} - \mathbf{w}_j \right|) - \mathbf{1}(\min_{j \neq y_i} \langle \Theta^{(t-1)} \tilde{\mathbf{K}}_i \mathbf{w}_{y_i} - \mathbf{w}_j \rangle < s) \langle G(\hat{\Theta}^{(t-1)}), \Theta - \hat{\Theta}^{(t-1)} \rangle_F \right\} \\ + \frac{\lambda}{2} \sum_{q=1}^{k-1} \theta_q^T \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{K} \end{pmatrix} \theta_q,$$

which is equivalent to

$$\min_{\Theta} \frac{1}{n} \sum_{i \notin \mathcal{L} \cup \mathcal{R}} H_1(\min_{j \neq y_i} \left| \Theta \tilde{\mathbf{K}}_i \mathbf{w}_{y_i} - \mathbf{w}_j \right|) + \frac{\lambda}{2} \sum_{q=1}^{k-1} \theta_q^T \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{K} \end{pmatrix} \theta_q, \quad (\text{E.4})$$

at convergence. Note that (E.4) is equivalent to assigning w_i weight 1 for observation $i \notin \mathcal{L} \cup \mathcal{R}$ and 0 otherwise, which is exactly the same as our IWASVM2. Furthermore, these weights do not change at the convergence of DCA2 and thus satisfy the convergence criterion of the IWASVM2 algorithm as well. This establishes the first part of the theorem.

Next, we prove the second part of the theorem, i.e., the local solution of the DCA2 is a fixed point for the IWASVM2. Denote the IWASVM2 solution at convergence by $\hat{\Theta}^{IWASVM2}$.

Using a similar argument above, we can show that, in a small neighborhood of the optimal solution $\hat{\Theta}^{IWASVM2}$ of (E.3), $w_i^{(t)} H_1(\min_{j \neq y_i} \left| \Theta \tilde{\mathbf{K}}_i \mathbf{w}_{y_i} - \mathbf{w}_j \right|)$ is the same as

$$H_1(\min_{j \neq y_i} \left\| \tilde{\mathbf{\Theta}}_t \tilde{\mathbf{K}}_t \mathbf{w}_{y_i} - \mathbf{w}_j \right\|) + \mathbf{1}(\min_{j \neq y_i} \left\| \hat{\mathbf{\Theta}}^{(t-1)} \tilde{\mathbf{K}}_t \mathbf{w}_{y_i} - \mathbf{w}_j \right\| < s) \left\langle G(\hat{\mathbf{\Theta}}^{(t-1)}), \mathbf{\Theta} - \hat{\mathbf{\Theta}}^{(t-1)} \right\rangle_F$$

up to a constant, and thus we have the equivalency between (E.3) and (E.2). Furthermore, at the convergence of the IWASVM2, the set

$$\{i: s \leq \min_{j \neq y_i} \left\| \min_{j \neq y_i} \left\| \mathbf{\Theta}^{DCA2} \tilde{\mathbf{K}}_t \mathbf{w}_{y_i} - \mathbf{w}_j \right\| \right\} \leq 1\}$$

does not change, consequently $\hat{\mathbf{\Theta}}^{IWASVM2}$ satisfies the convergence criterion of the DCA2. This completes the proof.

References

- [1]. An LTH, Tao PD, Solving a class of linearly constrained indefinite quadratic problems by DC algorithms, J. Global Optim 11 (1997) 253–285.
- [2]. Bishop CM, Pattern Recognition and Machine Learning, Springer, New York, 2006.
- [3]. Boser BE, Guyon IM, Vapnik VN, A training algorithm for optimal margin classifiers, In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, ACM, 1992.
- [4]. Bredensteiner EJ, Bennett KP, Multicategory classification by support vector machines, In: Computational Optimization, Springer, New York, 1999, pp. 53–79.
- [5]. Burges CJ, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery 2 (1998) 121–167.
- [6]. Cortes C, Vapnik V, Support-vector networks, Machine Learning 20 (1995) 273–297.
- [7]. Crammer K, Singer Y, On the algorithmic implementation of multiclass kernel-based vector machines, J. Mach. Learn. Res 2 (2001) 265–292.
- [8]. Cristianini N, Shawe-Taylor J, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, 2000.
- [9]. Dietterich TG, Bakiri G, Solving multiclass learning problems via error-correcting output codes, J. Artif. Intell. Res (1995) 263–286.
- [10]. Fan R-E, Chang K-W, Hsieh C-J, Wang X-R, Lin C-J, Liblinear: A library for large linear classification, J. Mach. Learn. Res 9 (2008) 1871–1874.
- [11]. Hastie TJ, Tibshirani RJ, Friedman JH, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, New York, 2009.
- [12]. Kimeldorf GS, Wahba G, Some results on Tchebycheffian spline functions, J. Math. Anal. Appl 33 (1971) 82–95.
- [13]. Lee Y, Lin Y, Wahba G, Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data, J. Amer. Statist. Assoc 99 (2004) 67–81.
- [14]. Liu Y, Fisher consistency of multicategory support vector machines, In: International Conference on Machine Learning, 2007.
- [15]. Liu Y, Shen X, Multicategory ψ -learning, J. Amer. Statist. Assoc 101 (2006) 500–509.
- [16]. Liu Y, Shen X, Doss H, Multicategory ψ -learning and support vector machine: computational tools, J. Comput. Graph. Stat 14 (2005) 219–236.
- [17]. Liu Y, Yuan M, Reinforced multicategory support vector machines, J. Comput. Graphical. Statist 20 (2011) 901–919.
- [18]. Liu Y, Zhang HH, Wu Y, Hard or soft classification? Large-margin unified machines, J. Amer. Statist. Assoc 106 (2011) 166–177.

- [19]. Nesterov Y, Efficiency of coordinate descent methods on huge-scale optimization problems, *SIAM J. Optim* 22 (2012) 341–362.
- [20]. Shen X, Tseng GC, Zhang X, Wong WH, On ψ -learning, *J. Amer. Statist. Assoc* 98 (2003) 724–734.
- [21]. Tang Y, Zhang HH, Multiclass proximal support vector machines, *J. Comput. Graph. Stat* 15 (2006) 339–355.
- [22]. Vapnik VN, *Statistical Learning Theory*, vol. 1, Wiley, New York, 1998.
- [23]. Weston J, Watkins C, Support vector machines for multi-class pattern recognition, In: *ESANN'1999 Proceedings — European Symposium on Artificial Neural Networks Bruges (Belgium)*, 21–23 April 1999, pp. 219–224, D-Facto public., ISBN 2–600049–9–X.
- [24]. Wu Y, Liu Y, On multicategory truncated hinge loss support vector machines, In: *Verducci JS, Shen X, Lafferty J (Eds.), Prediction and Discovery: AMS-IMS-SIAM Joint Summer Research Conference, Machine and Statistical Learning*, vol. 443, American Mathematical Society, Snowbird, Utah, 2006, pp. 49–58.
- [25]. Wu Y, Liu Y, Robust truncated hinge loss support vector machines, *J. Amer. Statist. Assoc* 102 (2007) 974–983.
- [26]. Wu Y, Liu Y, Adaptively weighted large margin classifiers, *J. Comput. Graph. Stat* 22 (2013) 416–432.
- [27]. Zhang C, Liu Y, Multicategory large-margin unified machines, *J. Mach. Learn. Res* 14 (2013) 1349–1386. [PubMed: 24415909]
- [28]. Zhang C, Liu Y, Multicategory angle-based large-margin classification, *Biometrika* 101 (2014) 625–640. [PubMed: 26538663]
- [29]. Zhang C, Liu Y, Wang J, Zhu H, Reinforced angle-based multicategory support vector machines, *J. Comput. Graph. Stat* 25 (2016) 806–825. [PubMed: 27891045]
- [30]. Zhang C, Pham M, Fu S, Liu Y, Robust multicategory support vector machines using difference convex algorithm, *Mathematical Programming*, in press.
- [31]. Zhang T, Statistical analysis of some multi-category large margin classification methods, *J. Mach. Learn. Res* 5 (2004) 1225–1251.
- [32]. Zou H, Zhu J, Hastie T, New multicategory boosting algorithms based on multicategory Fisher-consistent losses, *Ann. Appl. Statist* 2 (2008) 1290–1306.

Figure 1:
Plots of the hinge loss $H_{k-1}(u)$ (left) and $H_1(-u)$ (right), the weight functions for the OWASVM1, and the target functions $w^1(u)H_{k-1}(u)$ and $w^2(u)H_1(-u)$ for $k = 3$.

Figure 2:
Plots of the weight functions w^1 (left) and w^2 (right) for the OWASVM1 and IWASVM1
with $k = 3$ and $s = -1/(k - 1)$.

Figure 3:
Box plots of misclassification rates for different methods (denoted M1, ..., M10) on the simulated Example 1.

Figure 4:
Box plots of misclassification rates for different methods (denoted M1, ..., M10) on the simulated Example 2.

Table 1:

Misclassification error rates for RMSVM, MSVM and proposed WASVM1,2 on the simulated data of Example 1 with $k = 3$, $perc \in \{5\%, 10\%\}$ and $p \in \{2, 20, 200\}$.

Contaminated Percentage		5%			10%		
Estimated Bayes Error		0.0601			0.0911		
Dimension (p)		2	20	200	2	20	200
Methods Linear	RMSVM	0.0655	0.0906	0.2691	0.1063	0.1299	0.3031
	NWASVM1	0.0641	0.0846	0.2619	0.1038	0.1281	0.2996
	OWASVM1	0.0633	0.0838	0.2580	0.0953	0.1195	0.2940
	IWASVM1 ₀	0.0628	0.0829	0.2543	0.0948	0.1161	0.2913
	IWASVM1 _k	0.0627	0.0829	0.2536	0.0944	0.1159	0.2908
	MSVM	0.0646	0.0915	0.2686	0.1010	0.1341	0.3013
	NWASVM2	0.0645	0.0908	0.2641	0.0990	0.1322	0.2975
	OWASVM2	0.0626	0.0849	0.2593	0.0950	0.1244	0.2964
	IWASVM2 ₀	0.0623	0.0834	0.2567	0.0946	0.1192	0.2933
	IWASVM2 _k	0.0623	0.0833	0.2561	0.0945	0.1173	0.2923

Table 2:

Running time (sec) for various classifiers on the simulated data of Example 1.

Contaminated Percentage		5%			10%		
Dimension (p)		2	20	200	2	20	200
Methods Linear	RMSVM	0.8347	1.0258	1.2460	0.7597	1.0258	1.2597
	NWASVM1	0.6814	0.8289	1.0299	0.6950	0.8241	1.0576
	OWASVM1	1.3920	1.7113	2.1099	1.4204	1.8528	2.1391
	IWASVM1 ₀	3.9923	5.3037	7.1675	4.0922	5.5514	6.9542
	IWASVM1 _k	4.0820	5.5169	7.1950	4.1891	5.6093	7.1557
	MSVM	0.7572	0.8837	0.9827	0.8038	0.9192	1.1188
	NWASVM2	1.0450	1.0987	1.1923	1.2867	1.3179	1.4205
	OWASVM2	2.5530	2.6243	2.8878	2.5814	2.6356	2.8683
	IWASVM2 ₀	4.2928	5.3973	5.2176	4.8806	6.3315	7.5830
	IWASVM2 _k	4.3788	5.8550	6.0306	5.2700	6.5058	8.0396

Table 3:

Misclassification error rates for RMSVM, MSVM and proposed WASVM1,2 on the nonlinear simulated data in Section 4.2 with the fixed dimension $p = 2$ and $perc \in \{5\%, 10\%, 15\%, 20\%\}$.

Contaminated Percentage		5%	10%	15%	20%
Estimated Bayes Error		0.3163	0.3447	0.3731	0.4015
Methods Kernel	RMSVM	0.3452	0.3747	0.4061	0.4413
	NWASVM1	0.3416	0.3716	0.4033	0.4374
	OWASVM1	0.3402	0.3684	0.4007	0.4331
	IWASVM1 ₀	0.3385	0.3670	0.3967	0.4310
	IWASVM1 _k	0.3378	0.3662	0.3963	0.4303
	MSVM	0.3433	0.3753	0.4072	0.4421
	NWASVM2	0.3424	0.3704	0.4035	0.4384
	OWASVM2	0.3401	0.3691	0.4000	0.4364
	IWASVM2 ₀	0.3381	0.3666	0.3975	0.4309
	IWASVM2 _k	0.3376	0.3662	0.3968	0.4306

Table 4:

Running time (sec) of one replication for various classifiers on the simulated data of Example 2.

Contaminated Percentage		5%	10%	15%	20%
Methods Kernel	RMSVM	1.7350	1.7683	1.7752	1.8163
	NWASVM1	0.8058	0.7699	0.7897	0.7855
	OWASVM1	1.5872	1.5716	1.5397	1.5690
	IWASVM1 ₀	3.8802	4.1833	4.5198	5.3417
	IWASVM1 _k	4.7210	4.5306	4.5776	5.1359
	MSVM	1.2100	1.2047	1.2120	1.2069
	NWASVM2	1.5502	1.5471	1.5538	1.5621
	OWASVM2	3.0931	3.1115	3.1155	3.0972
	IWASVM2 ₀	7.7522	7.1315	7.5132	7.3654
	IWASVM2 _k	7.4143	7.0985	6.9237	7.1127

Table 5:

Misclassification error rates and running time in seconds for each method on the Vehicle dataset in Section 4.3. The corresponding standard errors for misclassification error rates are in parentheses.

		Misclassification error rate			Running time		
Contaminated Percentage		0%	5%	10%	0%	5%	10%
Methods Linear	RMSVM	0.2663 (0.0017)	0.3079 (0.0014)	0.3517 (0.0020)	1.8154	1.8518	1.8863
	NWASVM1	0.2629 (0.0012)	0.3073 (0.0013)	0.3478 (0.0019)	1.4133	1.4731	1.4559
	OWASVM1	0.2690 (0.0014)	0.3049 (0.0011)	0.3457 (0.0018)	2.8699	2.9124	2.8881
	IWASVM1 ₀	0.2798 (0.0017)	0.3031 (0.0013)	0.3432 (0.0016)	18.2805	18.3228	18.8719
	IWASVM1 _k	0.2805 (0.0014)	0.3021 (0.0015)	0.3419 (0.0019)	17.0342	16.2044	20.1709
	MSVM	0.2692 (0.0012)	0.3103 (0.0016)	0.3515 (0.0018)	1.8060	1.8011	1.8300
	NWASVM2	0.2576 (0.0015)	0.3073 (0.0017)	0.3496 (0.0017)	2.3441	2.4883	2.5840
	OWASVM2	0.2669 (0.0015)	0.3040 (0.0016)	0.3482 (0.0018)	4.6910	4.9787	4.9422
	IWASVM2 ₀	0.3150 (0.0014)	0.3026 (0.0015)	0.3446 (0.0018)	18.0893	19.4397	20.1780
	IWASVM2 _k	0.3003 (0.0018)	0.3018 (0.0015)	0.3430 (0.0019)	19.5693	20.6168	20.4720