Statistics and Its Interface Volume 8 (2015) 331–345

Distance-weighted Support Vector Machine

Xingye Qiao[*,†] and Lingsong Zhang[‡]

A novel linear classification method that possesses the merits of both the Support Vector Machine (SVM) and the Distance-weighted Discrimination (DWD) is proposed in this article. The proposed Distance-weighted Support Vector Machine method can be viewed as a hybrid of SVM and DWD that finds the classification direction by minimizing mainly the DWD loss, and determines the intercept term in the SVM manner. We show that our method inheres the merit of DWD, and hence, overcomes the data-piling and overfitting issue of SVM. On the other hand, the new method is not subject to the imbalanced data issue which was a main advantage of SVM over DWD. It uses an unusual loss which combines the Hinge loss (of SVM) and the DWD loss through a trick of axillary hyperplane. Several theoretical properties, including Fisher consistency and asymptotic normality of the DWSVM solution are developed. We use some simulated examples to show that the new method can compete DWD and SVM on both classification performance and interpretability. A real data application further establishes the usefulness of our approach.

AMS 2000 subject classifications: Primary 62H30.
Keywords and phrases: Discriminant analysis, Fisher consistency, Imbalanced data, High-dimensional, low-sample size data, Support Vector Machine.

1. INTRODUCTION

Classification is a very important research topic in statistical machine learning, and has many useful applications in various scientific and social research areas. In this article, we focus on the binary linear classification problem, in which a classification rule is to be found that maps a point in $\mathcal{X}$ to a class label chosen from $\mathcal{Y}$, $\phi: \mathcal{X} \mapsto \mathcal{Y}$ where $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \{+1, -1\}$. We focus on linear classification methods instead of nonlinear ones because they are easy to interpret due to simple formulations. In particular, each linear classification rule is associated with a linear discriminant function $f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{\omega} + \beta$, where the coefficient direction vector $\boldsymbol{\omega} \in \mathbb{R}^d$ has unit L_2 norm, and $\beta \in \mathbb{R}$ is the intercept term. The classification rule is then $\phi(\boldsymbol{x}) = \text{sign}(f(\boldsymbol{x}))$, that is, the sample space $\mathbb{R}^d$ is divided into halves by the separating hyperplane defined by $\{\boldsymbol{x} : f(\boldsymbol{x}) \equiv \boldsymbol{x}^T\boldsymbol{\omega} + \beta = 0\}$. The coefficient direction vector $\boldsymbol{\omega}$ determines the orientation of the hyperplane (as a matter of fact, it is the normal vector of this hyperplane), and the intercept term β determines its location.

There is a large body of literature on linear classification. See Duda, Hart and Stork (2001) and Hastie, Tibshirani and Friedman (2009) for comprehensive introductions. Among many linear classification methods, the Support Vector Machine (SVM; Cortes and Vapnik, 1995; Vapnik, 1998; Cristianini and Shawe-Taylor, 2000) and the Distance-weighted Discrimination (DWD; Marron, Todd and Ahn, 2007; Qiao *et al.*, 2010) are two state-of-the-art instances and have received a lot of attention. A brief review of these two methods will be given in Section 2.

In the high-dimensional, low-sample size (HDLSS) data setting, a so-called "data-piling" phenomenon has been observed for SVM (Marron, Todd and Ahn, 2007) and some other classifiers (for example, Ahn and Marron, 2010). Data-piling is referred to the phenomenon that after projected to the direction vector $\boldsymbol{\omega}$ given by a linear classifier, a large portion of the data vectors pile upon each other and concentrate on two points. Data-piling reflects severe overfitting in the HDLSS data setting and is an indicator that the direction is driven by artifacts in the data, and hence the direction as well as the classification performance can be stochastically volatile. Moreover, it turns out that the directions from these linear classification methods are much deviated from the Bayes rule direction (when the Bayes rule exists and is linear). To this end, DWD was proposed largely to overcome the data-piling issue in the HDLSS setting and has been quite successful on that.

While DWD overcomes the data-piling and mitigates the overfitting effect, it is sensitive to the imbalanced sample sizes between the two classes (Qiao *et al.*, 2010). In particular, when the sample size of one class is much greater than the other one, the classification boundary would be pushed towards the minority class and consequently, all future data vectors will be classified into the majority class.

Qiao and Zhang (2013) have thoroughly studied the high-dimensional overfitting issue of SVM and the imbalanced data issue of DWD. Moreover, they proposed a new family of classifiers called FLAME which both SVM and DWD belong to. To illustrate the main points of the data-piling

[*] Corresponding author.

[†] Supported by Binghamton University Harpur College Dean's New Faculty Start-up Funds, Simons Foundation Collaboration Grants for Mathematicians #246649 and Statistical and Applied Mathematical Sciences Institute (SAMSI).

[‡] Supported by Statistical and Applied Mathematical Sciences Institute (SAMSI).

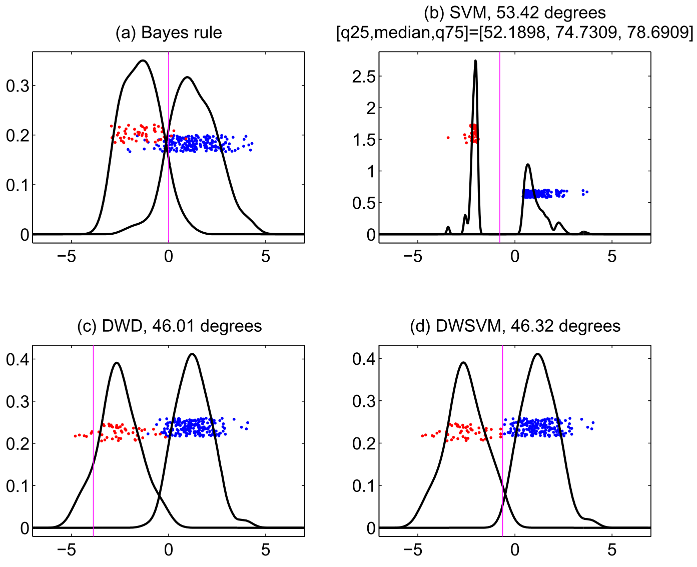

Figure 1. Plots of projections to: (a) the true mean difference (Bayes rule) direction, (b) the SVM direction, (c) the DWD direction and (d) the proposed DWSVM direction. The angles (in degree) between the last three directions and the first direction are shown in the titles. Projections of the separating hyperplanes of different methods are depicted by the magenta vertical lines. Panel (a) shows the Bayes direction and the separating hyperplane to be compared with. SVM in Panel (b) demonstrates a very good separation between the two classes, but a severe data-piling phenomenon also appears. The projected data vectors are nowhere near Gaussian, which suggests that the direction is too much deviated from the Bayes direction (Panel (a)). For SVM, the 25% and 75% quantiles and the median of the angles for 15 different tuning parameter values are reported in the subtitle of Panel (b) as well. Panel (c) shows that DWD has no data-piling issue, and the projection plot preserves the Gaussian pattern. However, the separating hyperplane is pushed towards the red class because of its relatively small sample size. Our proposed DWSVM approach (Panel (d)) combines the merits of SVM and DWD. It preserves a good direction by showing the Gaussian pattern in the projections while finds a good intercept term which is not subject to imbalanced sample sizes. Note that the SVM classifier in Panel (b) is tuned based on the misclassification rate for a large test set, while the tuning parameters for DWD and DWSVM are fixed.

and imbalanced issues, we show projection plots of a toy example to four different discriminant direction vectors in Figure 1. In this example, the data vectors from the two classes are generated from multivariate normal distributions $N_d(\pm\mu\mathbf{1}_d, \mathbf{I}_d)$, where the dimension $d = 300$, $\mu = 1.35/\sqrt{d} = 0.07794229$, $\mathbf{1}_d$ is a d-dimensional vector of all 1's and $\mathbf{I}_d$ is the $d \times d$ identity matrix. The Bayes rule in this example has direction $\boldsymbol{\omega}_B = \mathbf{1}_d/\sqrt{d}$ and the Bayes intercept $\beta_B = 0$. Here the sample size of the positive class (with $Y = +1$) is $n_+ = 200$ and the negative class sample size is $n_- = 50$.

Panel (a) in Figure 1 shows the true mean difference direction (which in fact is the Bayes direction) and the projections of the data vectors therein. They serve as the benchmark to be compared with. Panel (b) is for the SVM direction (whose corresponding tuning parameter has been selected based on misclassification errors for a large test set) and it demonstrates a very dramatic separation between the two classes. This could be an alarming bell for overfitting. Indeed, severe data-piling is visible. The projected data vectors are nowhere near Gaussian, which suggests that the direction is too much deviated from the true direction in Panel (a). This deviation is also measured by the angle between the SVM direction and the Bayes direction (53.42 degrees, shown in the subtitle). We also report the 25% and 75% quantiles and the median of the angles between the Bayes direction and the SVM directions for 15 different tuning parameter values in the subtitle of Panel (b). Panel (c) shows that DWD has no data-piling issue, and the projection plot preserves the Gaussian pattern, which means that there is some potential to interpret the data using the DWD direction. However, because the blue class (positive class with $Y = +1$) has four times the sample size as the red class, the separating hyperplane is therefore pushed towards the red class. Expectedly, its classification performance is not good.

In this article, we propose a new method which integrates the merits of SVM and DWD, and thus can address the data-piling issue and the imbalanced data issue at the same time. Our proposed method is named Distance-weighted Support Vector Machine (DWSVM) to salute the above two classical methods. As shown in Panel (d) of Figure 1, DWSVM preserves a good direction by showing the Gaussian pattern in the projections while finds a good intercept term which is not subject to the imbalanced sample sizes. In addition, we prove in theory that the DWSVM is Fisher consistent and asymptotically normal, and that its intercept term is not sensitive to imbalanced sample size as DWD is.

The rest of the article is organized as follows. Section 2 gives a brief introduction to the SVM and the DWD methods. Our DWSVM method is proposed in Section 3. Simulated examples and a real application are studied in Sections 4 and 5. Several theoretical results are given in Section 6. Some concluding remarks are made in Section 7. Technical proofs and details of computational algorithms are included in the appendix.

2. CLASSICAL METHODS

In this section, we give a brief introduction to SVM and DWD, their formulations and the discussion on the roles of different terms.

2.1 Classification and loss functions

In classification, one is given a training data set, $\mathcal{D} \equiv \{(\boldsymbol{x}_i, y_i) \in \mathcal{X} \otimes \mathcal{Y}, i = 1, \ldots, n\}$ and the goal is to find a rule, $\phi(\boldsymbol{x}) \equiv \operatorname{sign}(f(\boldsymbol{x}))$, depending on $\mathcal{D}$, so that the classification error $\mathbb{E}(\phi(\boldsymbol{X}) \neq Y)$ is minimized. A natural estimate of the classification error is $\frac{1}{n}\sum_{i=1}^n \mathbb{1}_{[\operatorname{sign}(f(\boldsymbol{x}_i)) \neq y_i]} = \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{[y_i f(\boldsymbol{x}_i)<0]}$. However, even in the simple case of linear classification where $f(\boldsymbol{x})$ is assumed to have the form $f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{\omega} + \beta$, searching for $(\boldsymbol{\omega}, \beta)$ to minimize $\sum_{i=1}^n \mathbb{1}_{[y_i f(\boldsymbol{x}_i)<0]}$ is intractable due to the discontinuity and nonconvexity of the objective function. In statistical learning, a common practice to avoid these issues is to use a convex surrogate function to approximate/upper-bound the 0-1 loss function $\mathbb{1}_{[yf(\boldsymbol{x})<0]}$. For any discriminant function $f(\boldsymbol{x})$, let us define $u \equiv yf(\boldsymbol{x})$ the functional margin which can be viewed as the signed distance (up to a constant) from data point $\boldsymbol{x}$ to the separating hyperplane $\{\boldsymbol{x} : f(\boldsymbol{x}) = 0\}$. A convex surrogate $\psi(u) : \mathbb{R} \mapsto \mathbb{R}^+$ can be used in the place of $\mathbb{1}_{[u<0]}$. For example, a classification rule can be obtained by minimizing over $(\boldsymbol{\omega}, \beta)$

$$\sum_{i=1}^n \psi\left(y_i(\boldsymbol{x}_i^T\boldsymbol{\omega} + \beta)\right) + \frac{\lambda}{2}\|\boldsymbol{\omega}\|^2.$$

Here, the first term in the objective function bounds the empirical classification error and the $\|\boldsymbol{\omega}\|^2$ term in the second term measures the complexity of the model. The choice of the tuning parameter λ balances the two main concerns. Equivalently, this optimization problem can be cast to $\min_{\boldsymbol{\omega},\beta}\sum_{i=1}^n \psi(y_i(\boldsymbol{x}_i^T\boldsymbol{\omega} + \beta))$, s.t. $\|\boldsymbol{\omega}\|^2 \leq C$ due to standard optimization theory. Many classification methods fall into this category, such as Support Vector Machine, AdaBoost (Freund and Schapire, 1997), and logistic regression (Friedman, Hastie and Tibshirani, 2000). See Bartlett, Jordan and McAuliffe (2006) and the references therein for more sophisticated discussions on convex loss functions and their implications for risk bounds.

2.2 Support Vector Machine (SVM)

By choosing the Hinge loss function $(1-u)_+$ as the convex surrogate, where $(a)_+ \equiv \max(a, 0)$ is the positive part of a, the SVM method is defined to maximize the smallest distances of all observations to the separating hyperplane. Mathematically, for some positive λ, the optimization problem of SVM can be written as $\min_{\tilde{\boldsymbol{\omega}},\tilde{\beta}} \sum_{i=1}^n (1 - y_i(\boldsymbol{x}_i^T\tilde{\boldsymbol{\omega}} + \tilde{\beta}))_+ + \frac{\lambda}{2}\|\tilde{\boldsymbol{\omega}}\|^2$. Here, in addition to measuring the model complexity, $\|\tilde{\boldsymbol{\omega}}\|^2$ also defines a notion of gap between the two classes for SVM. In particular, $2/\|\tilde{\boldsymbol{\omega}}\|$ is the distance between the classes (up to a constant). Hence, to minimize $\|\tilde{\boldsymbol{\omega}}\|^2$ is the same as to maximize the gap between classes. The notion of gap will play a central role in the derivation of methods in this article.

The formulation above can be equivalently written as $\min_{\tilde{\boldsymbol{\omega}},\tilde{\beta}} \sum_{i=1}^n (1 - y_i(\boldsymbol{x}_i^T\tilde{\boldsymbol{\omega}} + \tilde{\beta}))_+$, s.t. $\|\tilde{\boldsymbol{\omega}}\|^2 \leq C$. Here the coefficient vector $\tilde{\boldsymbol{\omega}}$ does not necessarily have unit norm. We let $\boldsymbol{\omega} = \tilde{\boldsymbol{\omega}}/\sqrt{C}$ and $\beta = \tilde{\beta}/\sqrt{C}$. Then the SVM solution is given by $\operatorname{argmin}_{\boldsymbol{\omega},\beta} \sum_{i=1}^n (\sqrt{C} - Cy_i(\boldsymbol{x}_i^T\boldsymbol{\omega} + \beta))_+$, s.t. $\|\boldsymbol{\omega}\|^2 \leq 1$. In this formulation, a modified Hinge loss function,

$$(1) \qquad H_C(u) = \begin{cases} \sqrt{C} - Cu & \text{if } u \leq \frac{1}{\sqrt{C}}, \\ 0 & \text{otherwise,} \end{cases}$$

is used, such that SVM can be viewed as to minimize $\sum_{i=1}^n H_C(u_i)$, subject to $\|\boldsymbol{\omega}\|^2 \leq 1$, where the functional margin u_i for the ith data is $u_i = y_i(\boldsymbol{x}_i^T\boldsymbol{\omega} + \beta)$. In order to align this formulation with that of DWD, we introduce a slack variable ξ_i and rewrite SVM as,

$$(2) \qquad \operatorname*{argmin}_{\boldsymbol{\omega},\beta,\xi_i} \sum_{i=1}^n \xi_i\ ,$$

$$(3) \qquad \text{s.t. } Cy_i(\boldsymbol{x}_i^T\boldsymbol{\omega} + \beta) + \xi_i \geq \sqrt{C},\ \xi_i \geq 0,$$

$$(4) \qquad \|\boldsymbol{\omega}\|^2 \leq 1.$$

2.3 Distance-weighted Discrimination (DWD)

DWD method was proposed by Marron, Todd and Ahn (2007) to improve the performance of SVM in the HDLSS setting. It also maximizes a notion of gap between classes: the harmonic mean of the distances of all data vectors to the separating hyperplane. Let $r_i = y_i(\boldsymbol{x}_i^T\boldsymbol{\omega} + \beta) + \eta_i$ be the (adjusted) distance of the ith data vector to the separating hyperplane. Mathematically, the solution of DWD is

$$(5) \qquad \operatorname*{argmin}_{\boldsymbol{\omega},\beta,\eta_i} \sum_{i=1}^n \left(\frac{1}{r_i} + C\eta_i\right)\ ,$$

$$(6) \qquad \text{s.t.} \quad r_i = y_i(\boldsymbol{x}_i^T\boldsymbol{\omega} + \beta) + \eta_i,\ r_i \geq 0 \text{ and } \eta_i \geq 0,$$

$$(7) \qquad \|\boldsymbol{\omega}\|^2 \leq 1.$$

When $\eta_i = 0$ and $y_i(\boldsymbol{x}_i^T\boldsymbol{\omega} + \beta) > 0$, $r_i = y_i(\boldsymbol{x}_i^T\boldsymbol{\omega} + \beta)$ is the positive distance from each data vector to the separating hyperplane, due to (6). Thus $\sum_{i=1}^{n} 1/r_i$ defines a different notion of gap between classes from that by SVM (which was $2/\|\boldsymbol{\omega}\|$.)

If a positive distance $y_i(\boldsymbol{x}_i^T\boldsymbol{\omega}+\beta)$ is not achievable for a data vector, then a positive slack variable η_i is added to make r_i positive. Note that the value of correction η_i corresponds to the amount of misclassification for the ith vector, and hence in order to minimize the misclassification, we must control $\sum_{i=1}^{n}\eta_i$ in the objective function.

We will use this formulation and combine it with that of the SVM method in (2)–(4). Here, in order to understand the underlying DWD loss function for later use, we modify (5)–(7) as follows. For each i, the term in the objective function $(\frac{1}{r_i}+C\eta_i)$ can be minimized over η_i. Some algebraic manipulations reveal that the optimization problem (about $\boldsymbol{\omega}$ and β) becomes

$$\operatorname*{argmin}_{\boldsymbol{\omega},\beta} \quad \sum_{i=1}^{n} V_C\left(y_i(\boldsymbol{x}_i^T\boldsymbol{\omega}+\beta)\right) \ , \tag{8}$$

$$\text{s.t.} \quad \|\boldsymbol{\omega}\|^2 \le 1, \tag{9}$$

where the DWD loss function is defined as

$$V_C(u) = \begin{cases} 2\sqrt{C}-Cu & \text{if } u \le \frac{1}{\sqrt{C}}, \\ 1/u & \text{otherwise.} \end{cases} \tag{10}$$

One key observation is to be made here. There are two main tasks in a binary linear classification method:

1) a *notion of gap* which is to be maximized so as to make the two classes more separated; and
2) a *measure of misclassification* which is to be minimized.

Recall that in the SVM formulation, the notion of gap is $2/\|\boldsymbol{\omega}\|$, and the misclassification is measured by the Hinge loss function. SVM jointly minimizes the sum of these two components to search for a solution. In contrast, the DWD loss function in (10) (derived from the objective function (5)) has two functionalities: the first term $\sum_i r_i^{-1}$ in (5), the sum of inverse distance, is a notion of gap, and the second term $\sum_{i=1}^{n}\eta_i$ in (5) measures misclassification. Unlike its counterpart in SVM, the constraint $\|\boldsymbol{\omega}\|^2\le 1$ in DWD (7) merely serves as a regulator but it does not maximize the gap. This appears to be a reason that DWD fails to provide a sensible intercept term for classification cutoff point: it cannot accomplish both tasks at the same time!

The main motivation of our DWSVM approach is to extract the role of misclassification controller from the DWD loss, and assign this role to a SVM component. As will be shown in the next section, we carefully design our formulation to allow a DWD component to define a notion of gap between the two classes, which helps to find a good direction vector. Meanwhile, we let an SVM component control the misclassification, which helps to search for a better intercept term.

3. DISTANCE-WEIGHTED SUPPORT VECTOR MACHINE

In Section 3.1, we first introduce a method which can be intuitively viewed as the prototype of the hybridization between SVM and DWD. Our proposed main method will be discussed in Section 3.2. Some explanations to our method are given in Section 3.3.

3.1 Simple prototype: Naive DWSVM

Before we introduce the DWSVM method, we discuss an intuitive hybridization between SVM and DWD, which is called the naive DWSWD method (nDWSVM). Based on the previous discussion and other results in the literature, a linear classifier with a direction given by DWD and an intercept term found by SVM is desirable. However, naively matching a DWD direction and an SVM intercept together would be problematic because the intercept would lose its context without the corresponding discriminant direction. Instead, we could train a DWD classifier on the data set, discard the DWD intercept, keep the DWD direction, and project all the data vectors to the 1-dimensional DWD direction to obtain a set of 1-dimensional data points. Lastly, find an intercept (a cutoff) by applying SVM to this 1-dimensional data set. Following this paradigm, we can get a DWD direction, which is thought to be better than an SVM direction in overcoming overfitting, and then given this DWD direction, search for an intercept in an SVM manner so as to mitigate the imbalanced data issue. We name this two-step procedure as nDWSVM. The two-step nDWSVM method is a simple prototype of DWSVM, where the DWD component and the SVM component are trained at the same time.

3.2 DWSVM

In this subsection, we formally define the Distance-weighted Support Vector Machine (DWSVM). DWSVM simultaneously minimizes both the SVM loss function and the DWD loss function, to identify a common discriminant direction. The less-imbalance-sensitive SVM-driven intercept term will be used to identify the location of the optimal separating hyperplane. Mathematically, the optimization problem can be written as follows: Let $C_{dwd}>0$, $C_{svm}>0$ and $\alpha\in[0,1)$. The DWSVM solution is given by

$$\operatorname*{argmin}_{\boxed{\boldsymbol{\omega},\beta}\,\beta_0,\xi_i,\eta_i} \sum_{i=1}^{n}\left\{\alpha\left(\frac{1}{r_i}+C_{dwd}\cdot\eta_i\right)+(1-\alpha)\xi_i\right\}, \tag{11}$$

$$\text{s.t.} \quad r_i = y_i(\boldsymbol{x}_i^T\boldsymbol{\omega}+\beta_0)+\eta_i,\ \ r_i\ge 0 \text{ and } \eta_i\ge 0, \tag{12}$$

$$C_{svm}y_i(\boldsymbol{x}_i^T\boldsymbol{\omega}+\beta)+\xi_i \ge \sqrt{C_{svm}},\ \ \xi_i\ge 0, \tag{13}$$

$$\|\boldsymbol{\omega}\|^2 \le 1. \tag{14}$$

Importantly, in the end, we let $f(\boldsymbol{x}) \equiv \boldsymbol{x}^T\boldsymbol{\omega} + \beta$ and use $\text{sign}(f(\boldsymbol{x})) = \text{sign}(\boldsymbol{x}^T\boldsymbol{\omega} + \beta)$ as the classification rule instead of $\text{sign}(\boldsymbol{x}^T\boldsymbol{\omega} + \beta_0)$. Thus $\boldsymbol{\omega}$ and β are the only two variables that really participate in classifying future data vectors, while β_0 is not involved. However, it does not mean that β_0 is of no significance. We will elaborate this point later.

Comparing (11)–(14) with (2)–(4) and (5)–(7), we can see that the first term in (11) and the constraint (12) are similar to (5) and (6), while the second term in (11) and the constraint (13) are similar to (2) and (3). Thus we may write the DWSVM formulation (11)–(14) as

$$\underset{\boldsymbol{\omega},\beta,\beta_0}{\text{argmin}} \sum_{i=1}^{n} \left\{ \alpha V_{C_{dwd}}\left(y_i(\boldsymbol{x}_i^T\boldsymbol{\omega} + \beta_0)\right) + (1-\alpha) H_{C_{svm}}\left(y_i(\boldsymbol{x}_i^T\boldsymbol{\omega} + \beta)\right) \right\}, \tag{15}$$

$$\text{s.t.} \quad \|\boldsymbol{\omega}\|^2 \leq 1. \tag{16}$$

One might think that our DWSWM is just an optimization problem with the objective function equaling to a weighted average of the DWD loss and the SVM loss. However, it is more sophisticated than that. In the next subsection, we give some explanations to different components and parameters in DWSVM to help understand the new method.

3.3 Understanding DWSVM

Below we review some important aspects of the one-step DWSVM classifier.

Two hyperplanes

First of all, the most significant difference of DWSVM from previous methods is that there are two intercept terms β_0 and β and only one direction vector $\boldsymbol{\omega}$ in the DWSVM method, that is, there are two hyperplanes that are parallel to each other, $\{\boldsymbol{x} : \boldsymbol{x}^T\boldsymbol{\omega} + \beta = 0\}$ and $\{\boldsymbol{x} : \boldsymbol{x}^T\boldsymbol{\omega} + \beta_0 = 0\}$. For convenience, we call them the main hyperplane and the axillary hyperplane, respectively, and their corresponding discriminant functions $f \equiv \boldsymbol{x}^T\boldsymbol{\omega}+\beta$ and $f_0 \equiv \boldsymbol{x}^T\boldsymbol{\omega} + \beta_0$. See Figure 2 for an illustration using a two-dimensional toy example. In the plot, the magenta solid line is the main hyperplane and the magenta dashed line is the axillary hyperplane.

The axillary hyperplane

Note that r_i's are the adjusted distances of data vectors to the axillary hyperplane $\{\boldsymbol{x} : f_0(\boldsymbol{x}) = 0\}$, shown as dot-dashed line segments in Figure 2. Similar to its role in DWD, $\sum_{i=1}^n(1/r_i)$ controls the gap between the two classes. In particular, the smaller $\sum_{i=1}^n(1/r_i)$ is, the more separated the two classes are.

In words, the purpose of the axillary hyperplane is not for classifying data vectors, but to make it possible to define a number of distances (from data vectors to itself) so that we can minimize the sum of the inverse distances, which leads to a notion of gap. In contrast, in the ordinary DWD, this axillary hyperplane has to coincide with the hyperplane that is actually used for classification.

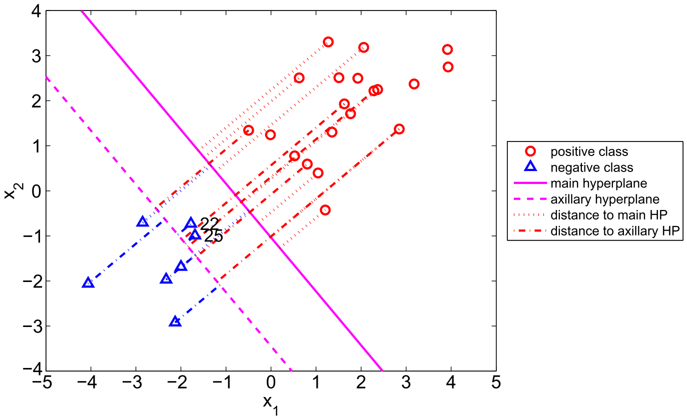

Figure 2. The main separating hyperplane (magenta solid line) and the axillary hyperplane (magenta dashed line) for DWSVM applied to a two-dimensional toy example. The distance from each data vector to the main hyperplane is depicted as a dotted line segment while the distance to the axillary hyperplane is depicted as a dotted-dashed line segment. Although the data vectors #22 and #25 are on the wrong side of the axillary hyperplane, they are not treated as misclassified by this method as they are both on the correct side of the main hyperplane. A positive η_i is added to each negative functional margin $y_i(\boldsymbol{x}_i^T\boldsymbol{\omega} + \beta_0)$, $i = 22, 25$, to make the sum positive.

Slack variables η_i v.s. ξ_i

There are two slack variables in this optimization problem: η_i is with respect to the axillary hyperplane, and ξ_i is with respect to the main hyperplane. Since the axillary hyperplane does not actually classify any future data vector, the slack variable η_i does not measure misclassification.

On the other hand, it is the variable ξ_i that measures the misclassification of the ith data vector. In particular, ξ_i serves as a proxy of the modified Hinge loss $(\sqrt{C_{svm}} - C_{svm}u_i)_+$.

Necessity of the slack variable η_i

In DWSVM, the slack variable η_i is used to adjust the sign of $y_i f_0(\boldsymbol{x}_i)$. When $y_i f_0(\boldsymbol{x}_i) < 0$, the (signed) distance from the data vector to the axillary hyperplane is negative. In this case, a positive η_i is added to $y_i f_0(\boldsymbol{x}_i)$ to make their sum r_i positive. For example, in Figure 2, the functional margins $y_i f_0(\boldsymbol{x}_i)$ for data vectors #22 and #25 are negative. The DWSVM optimization adds some positive η_i's to make the sum $r_i = y_i f_0(\boldsymbol{x}_i) + \eta_i$ positive. It is the sum of $1/r_i$ that we minimize, instead of the sum of $1/(y_i f_0(\boldsymbol{x}_i))$.

Recall that in the ordinary DWD, we control the sum of the slack variables, since they measures misclassification.

Now that in DWSVM, η_i does not measure misclassification (see the discussion in the previous part above), does it mean that the slack variable η_i is no longer needed? The answer is no. If not for the η_i, one can always make β_0 to be infinity, that is, the axillary hyperplane is infinitely far from the data so that all the distances $y_i f_0(\boldsymbol{x}_i)$'s are infinity (whether positive or negative), and hence $1/(y_i f_0(\boldsymbol{x}_i)) = 0$. This is certainly not a desired situation because it would make the direction vector trivial (because the minimal of the objective function would always be 0 regardless of the choice of the direction). For these reasons, the addition of η_i and the inclusion of $\sum_{i=1}^{n} \eta_i$ in the objective function are necessary to make the optimization problem meaningful.

Summary

In summary, the hyperplane defined by $\boldsymbol{\omega}$ and β_0 is an axillary hyperplane which is useful for finding the *best* direction, and the one defined by $\boldsymbol{\omega}$ and β is the main hyperplane that is useful for minimizing the Hinge loss for classification. By the trick of allowing two intercept terms, we gain some flexibility and manage to get two hyperplanes to do their own jobs.

Empirically, nDWSVM can be used to mimic the idea of DWSVM. As a matter of fact, nDWSVM is very easy to implement, so long as the user has accessible implementations for both SVM and DWD (both are now available in R and MATLAB). The differences between DWSVM and nDWSVM are that in the two-step prototype nDWSVM, the direction is determined *only* by the DWD algorithm, and the intercept is found by SVM based on the projections given by the DWD direction. However, in DWSVM, the optimization is done all at once in DWSVM and both the SVM and DWD components jointly optimize the direction.

Between DWD and DWSVM, the latter inherits the direction of the former, and adopts a more effective intercept term from its SVM component. Compared with SVM, the DWSVM method has a direction that is much improved due to the DWD component.

4. SIMULATIONS

In this section, we first compare the classification and the interpretability performance between the DWSVM approaches and the original SVM and DWD. The classification performance is measured by the misclassification rate for a large test data set with 4,000 observations. The interpretability is a concept that is more or less vague. We partially measure it by the angle between the discriminant direction vector for the classifier under investigation and for the Bayes classifier. We believe the closer to the Bayes rule direction, the better the interpretability of the linear classifier is.

4.1 Performance comparison

We consider two different simulation settings. In each setting, samples from the two classes are generated from multivariate normal distributions $N_d(\pm\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

1. **Example 1**: Constant mean difference, identity covariance matrix example. $\boldsymbol{\mu} \equiv c\mathbf{1}_d$, and $\boldsymbol{\Sigma} \equiv \mathbf{I}_d$, where $c > 0$ is a scaling factor which makes $2c\|\mathbf{1}_d\|_2 = 2.7$. This corresponds to the Mahalanobis distance between the two classes and represents a reasonable difficulty of classification using the Bayes rule.
2. **Example 2**: Decreasing mean difference, block-diagonal interchangeable covariance matrix example. Here we let $\boldsymbol{\mu} \equiv c\boldsymbol{v}_d$, where $\boldsymbol{v}_d = (\sqrt{50}, \sqrt{49}, \ldots, \sqrt{1}, 0, 0, \ldots, 0)^T \in \mathbb{R}^d$, and $\boldsymbol{\Sigma} \equiv$ Block-Diag$\{\Sigma, \Sigma, \ldots, \Sigma\}$, where each Σ is an 50×50 interchangeable sub-covariance matrix whose diagonal entries are all 1 and off-diagonal entries are 0.8. The scaling factor c is chosen to make the Mahalanobis distance $\{(2c\boldsymbol{v}_d)\boldsymbol{\Sigma}^{-1}(2c\boldsymbol{v}_d)^T\}^{1/2} = 2.7$.

In both simulation settings, we let the positive class sample size be 200 and the negative class sample size be 50. We vary the dimensions d among $100, 200, 300, 500$ and 1,000, thus the last three cases correspond to the HDLSS data settings.

4.1.1 Example 1

In the top-left panel of Figure 3, we report the misclassification error of DWSVM, nDWSVM, DWD and SVM applied to a test data set with 2,000 data points in each class which are generated according to the Constant mean difference, identity covariance matrix example. We conduct the simulation for 100 times and report the averages of the measurements. The standard error of the mean measurement is shown as error bars. Our DWSVM approach gives the best classification results in most cases. The two-step alternative nDWSVM has very similar performance for dimensions 100, 200 and 500, but its performance is downgraded for higher dimensions. For all dimensions, unsurprisingly, the original DWD has misclassification rate close to almost 50%, which is largely due to its intercept term which is subject to the imbalanced data.

In the bottom-left panel of Figure 3, we calculate the angles between the directions from different classifiers and the Bayes direction (for both simulation settings in this article, the Bayes classifiers are linear and the Bayes directions are well defined.) It shows that all the DWD related classifiers give very similar angles. As a matter of fact, the angles from DWSVM, DWSVM and DWD almost overlap with each other in this plot, except for low dimensional case where the DWSVM angle is a bit larger than the other two. On the other hand, the SVM directions are significantly more different from the Bayes direction than the DWD family directions are.

The observations so far verify the conjecture that DWD is worse at misclassification rate and SVM is worse at giving interpretable classification direction. DWSVM and nDWSVM appear to be able to address both issues simultaneously.

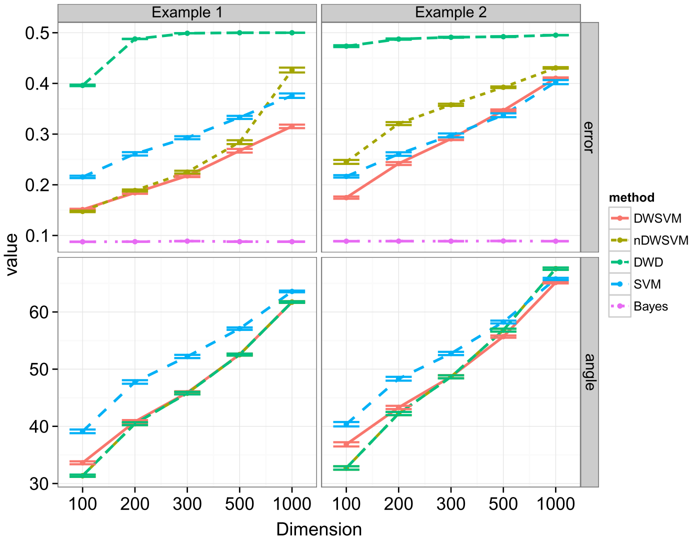

Figure 3. Comparison between four methods for Example 1 (the left panel) and Example 2 (the right panel). The misclassification error rates are shown on the top row and the angles between the classification directions and the Bayes direction are shown on the bottom. For Example 1 (left), for smaller dimensions, the two DWSVM approaches are better than SVM and DWD in terms of classification. For a large dimension, SVM outperforms the nDWSVM approach. The one-step DWSVM approach dominates all the other approaches in terms of classification performance. In terms of the interpretability (bottom) in Example 1, the two DWSVM approaches and the DWD approach all give similar and better results than the SVM approach. For Example 2 (right), DWSVM has similarly good classification performance to SVM and similarly good interpretability performance to DWD and nDWSVM. For small and moderate dimensions, the classification performance of DWSVM is significantly better than SVM.

In the simulations, we tune the parameter C_{svm} for SVM from a grid of possible values $2^{-5}, 2^{-4} \ldots, 2^{11}, 2^{12}$ and choose the one which gives rise to the smallest misclassification rate for a tuning data set that is identical to the training data set in terms of sample size and underlying distributions. For the DWD family of classifiers (DWSVM, nDWSVM and DWD), we let C_{dwd} be 100 divided by a scaling factor that counts for the scale of the data, which was recommended by Marron, Todd and Ahn (2007). We fix $C_{svm} = 100$ for DWSVM and nDWSVM. Lastly, we let $\alpha = 0.5$ for DWSVM in our simulation study. Thus, the tuning parameter for SVM has been optimized while tuning parameters for our DWSVM methods are not tuned. Yet, our DWSVM method can achieve the performance as good as, sometimes even much better than, the other methods, for multiple criteria (classification and interpretability). This suggests a great potential of the DWSVM method.

4.1.2 Example 2

We have conducted the same comparison for the Decreasing mean difference, block-diagonal interchangeable covariance matrix example (Example 2) and the results are shown in the right panel of Figure 3. This time, the classification performance of DWSVM and SVM are closely competing with each other. For dimensions $d = 100, 200, 300$, the DWSVM misclassification rates are smaller than SVM. But for dimensions $d = 500$ and 1,000, its classification error rates are slightly greater than SVM (not visually significant). In terms of the angles between the classification direction vectors and the Bayes direction, the DWSVM direction are similar to those from nDWSVM and DWD; all three are better than SVM for dimensions 100, 200, 300, 500 and DWSVM is better than SVM in all cases (with an insignificant margin when $d = 1000$). For the highest dimension case, all four directions are much different from the Bayes direction. However, the DWSVM direction may be the best in this situation (with an insignificant margin).

4.2 Sensitivity to parameter values

In this subsection, we investigate the impacts of different parameter values on DWSVM. Note that there are three tuning parameters in DWSVM, namely, α, C_{dwd} and C_{svm}. We have the following strategy for each parameter.

a) For C_{dwd}: Marron, Todd and Ahn (2007) suggested that the tuning parameter in DWD (the counterpart of C_{dwd} in DWSVM) should be 100 divided by a typical distance measure to count for the scale of the data. Qiao *et al.* (2010) verified this for the weighted DWD classifier. In particular, they suggested 100 divided by the median of the pairwise distances among data vectors. Here we consider 100 divided by the 25% quantile, the median and the 75% quantile of the pairwise distances in the training data set for C_{dwd} (shown in the left, middle and right columns in Figures 4 and 5.)

b) For C_{svm}: We use a fine grid of 35 values $C_{svm} = 2^{-5}, 2^{-4.5}, 2^{-4}, 2^{-3.5} \ldots, 2^{11}, 2^{11.5}, 2^{12}$.

c) For α: We use a grid of 19 values, namely, $\alpha = 0.05, 0, 1, \ldots, 0.95$.

Overall, we have tried $3 \times 19 \times 35$ parameter triplets in each simulation replication. We apply DWSVM to 100 replications from the simulated examples defined in the last subsection (Example 1 and Example 2) respectively. The performance of the resulting DWSVM is measured by (1) the misclassification rate for a large test data with 4,000 observations (2,000 in each class), and (2) the angle (in degree) between the DWSVM direction and the Bayes rule direction.

Figures 4 and 5 report the average error rates and average angles over 100 replications for the two examples. The left, middle and right columns in each figure are for the three different C_{dwd} values. The grayscale images in the top panels show the misclassification rates. The darker the pixel,

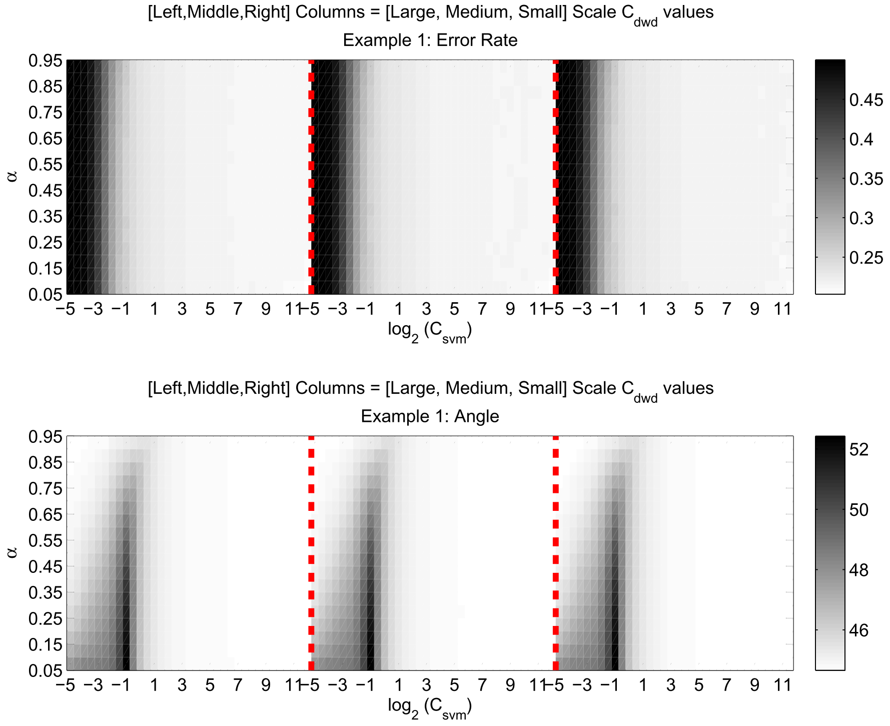

Figure 4. Sensitivity analysis for Example 1. The left, middle, right columns are for different C_{dwd} values. The grayscale images in the first row show misclassification rates and those in the second row indicate the angles between the DWSVM directions and the Bayes direction. Darker pixels correspond to high values (color bars for the images are shown aside). The analysis suggests that the α and C_{dwd} parameters have much smaller impacts on the performance of DWSVM than the C_{svm} parameter does.

the higher the error rate. Similarly, the grayscale images in the bottom panels show the angles. The darker the pixel is, the more deviated the DWSVM direction is from the Bayes direction.

Firstly, one can easily have the impression that the images in the left, middle and right columns are almost identical; hence the C_{dwd} value seems to have very little influence. Given any C_{dwd} value, between the parameter α and the parameter C_{svm}, one can see that change of the overall pattern of the image depends more on C_{svm} than on α. In particular, in Example 1 (Figure 4), the grayscale image for the error rate almost does not change as α changes; in Example 2 (Figure 5), it does not change much except when α is very close to 1, at which point, the error rate corresponding to $\log_2(C_{svm}) = 1$ seems to increase. As far as the angle is concerned, in both examples, it does appear that for $\log_2(C_{svm})$ between -5 and 1, the patterns change as α changes; however, such changes are clearly not as drastic as those caused by the change of C_{svm}.

One may wonder about the ideal choice of parameter values in each case. For Example 1 (Figure 4), a C_{svm} value greater than 2^3 (with almost any α value) is easily the winner, since it leads to a small error and a small angle. For Example 2 (Figure 5), it seems that a "small C_{svm}-large α" combination can lead to a better direction, but a worse classification error. What complicates even more is that the best C_{svm} values for the error, namely, $\log_2(C_{svm})$ between 1 and 3, may also correspond to the worst angles. Nonetheless, one can still choose a C_{svm} value greater than 2^3 (with almost any α value) to achieve a balance between the classification performance and the interpretability (with slight compromises on both ends). Note that in reality, the angle images cannot be obtained and the error images are based on a much smaller tuning set or by a cross-validated error.

With this sensitivity analysis in mind, we have recommended that the C_{dwd} be fixed at the value suggested by Marron, Todd and Ahn (2007). Moreover, although the α does change the angle (hence the direction) a little bit in some situations, the difference it makes is quite small so that the additional effort to tune it may not be needed. In the article, we have used a noninformative choice of $\alpha = 1/2$, indicating that we have no *a priori* preference between the two components. More discussion on the parameter α will be given in Section 7.

5. REAL APPLICATION

In this section, we compare DWSVM with the competing classifiers by applying them to the Golub data set (Golub *et al.*, 1999). This gene expression data has 3,051 genes

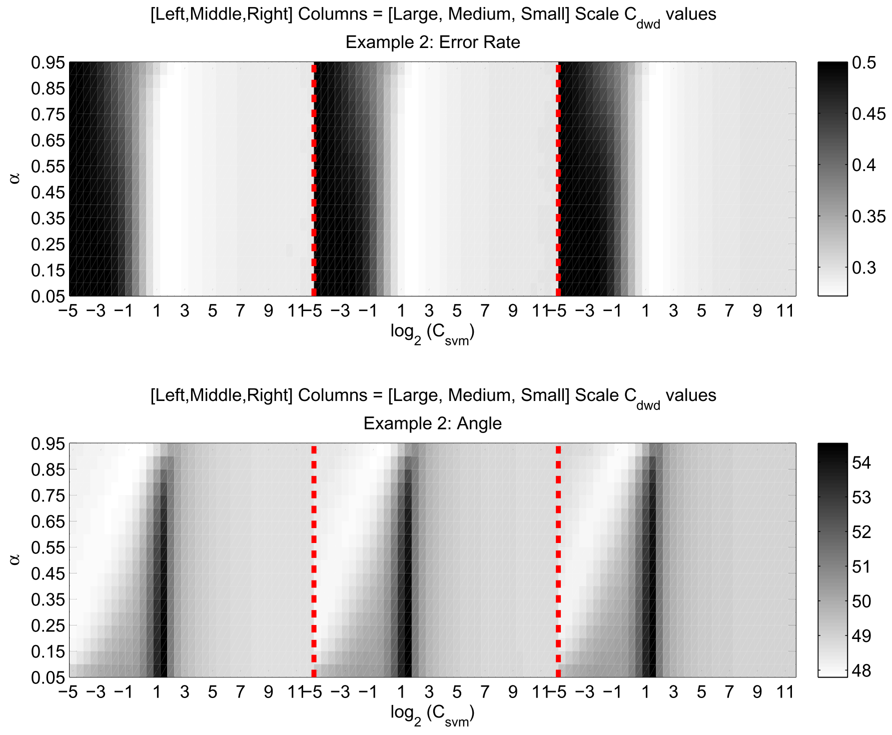

Figure 5. Sensitivity analysis for Example 2. The left, middle, right columns are for different C_{dwd} values. The grayscale images in the first row show misclassification rates and those in the second row indicate the angles between the DWSVM directions and the Bayes direction. Darker pixels correspond to high values (color bars for the images are shown aside). A similar conclusion to that in Figure 4 may be drawn, although a subtle change of pattern in the angle due to change of α seems to be visible.

and 38 tumor mRNA samples from the leukemia microarray study of Golub *et al.* (1999). Pre-processing was done as described in Dudoit, Fridlyand and Speed (2002).

As there are 11 and 27 observations from both classes, we expect the SVM and the DWSVM classifiers will give better result than DWD because the latter is subject to the imbalanced sample size. Moreover, because the dimension is much higher than the sample size, we expect severe overfitting in this data. We apply SVM, DWD, DWSVM and nDWSVM to the data set and use 3-fold cross validation to find the best C_{svm} tuning parameter value. The C_{dwd} and α values are fixed. In the left panel of Figure 6, we report the average cross-validated (CV) number of misclassified observations and the standard error over 100 random foldings. Both SVM and DWSVM give very good results (CV error almost zero), although the DWSVM method is a little better. The nDWSVM error is almost twice that of the SVM and the DWD error is almost four times.

In order to see the extent to which our DWSVM avoids overfitting, we perturb the original data set as follows. We randomly switch the class labels of k pairs of observations (k observations from each class) ($k = 1, 2$). Then we conduct parameter tuning (via cross-validation) and training based on the perturbed data. Then, we calculate the cross-validated error for the resulting classifier: we use two folds (2/3) of the perturbed data to training a classifier, and evaluate the number of misclassified observation for the remaining fold using the true class labels (the label before perturbation). Because we randomly add in noise into such settings, the CV errors increases. However, a classifier which is subject to overfitting would have a greater CV error in this setting. In the middle and the right panels of Figure 6, we report the CV error for the perturbed data where one pair and two pairs of data vectors are mislabeled respectively. As we can see, although all classifiers perform worse here than for the original data, the DWSVM classifier gives the lowest CV errors for the perturbed data. Even the performance of the two-step nDWSVM is on the par with SVM. The performance of DWD is always the worse in all three setting because of the imbalanced data issue.

6. THEORETICAL PROPERTIES

We will show some theoretical properties of DWSVM in three different favors. First, we derive the Fisher consistency of the DWSVM loss function. Note that the loss function of DWSVM is not a typical large-margin loss function. Second, we derive the asymptotic normality of the DWSVM coefficient vector. Third, we show that the intercept of DWSVM does not diverge, even in an extremely imbalanced setting.

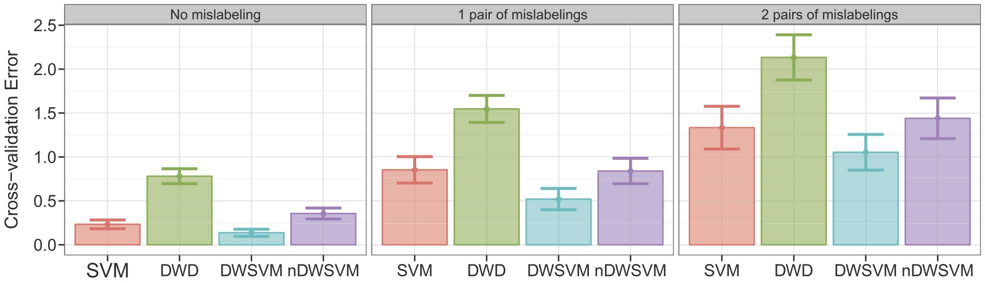

Figure 6. Cross-validated number of misclassified observations for SVM, DWD, DWSVM and nDWSVM for the original Golub data set, the Golub data with a pair of mislabeled observations, and the data with 2 pairs of mislabeled observations. For the original data, both the SVM and the DWSVM methods have CV error almost 0, with DWSVM being a little better. When there are mislabeled observations, the advantage of DWSVM becomes more obvious: it can be seen that DWSVM has the smallest CV errors while nDWSVM is on a par with SVM. The DWD classifier is always worse than the others in terms of the classification performance.

6.1 Fisher consistency

The DWSVM method can be estimated from equations (15)–(16). Thus the underlying loss function can be written as $L(yf(\boldsymbol{x}), yf_0(\boldsymbol{x})) = \alpha V_{C_{dwd}}(yf_0(\boldsymbol{x})) + (1-\alpha)H_{C_{svm}}(yf(\boldsymbol{x}))$. Because there are two functions involved, the underlying loss function is not a traditional margin-based loss function which involves only one function, such as that considered in Lin (2004). Moreover, the two hyperplanes implied by f and f_0 in our methods are parallel to each other. In general cases (beyond linear functions), this can be interpreted as the difference of these two functions is a constant, that is, $f(\boldsymbol{x}) - f_0(\boldsymbol{x})$ is independent of $\boldsymbol{x}$. Theorem 1 below shows the Fisher consistency of the DWSVM loss function.

Theorem 1. *For any given $C_{svm}, C_{dwd} > 0$ and $\alpha \in [0,1)$, if $\mathbb{E}[L\{Yf(\boldsymbol{X}), Yf_0(\boldsymbol{X})\}]$ has a global minimizer $(f^*(\boldsymbol{x}), f_0^*(\boldsymbol{x}))$ subject to $f(\boldsymbol{x}) - f_0(\boldsymbol{x})$ is a constant, then $\text{sign}[f^*(\boldsymbol{x})] = \text{sign}[q(\boldsymbol{x}) - 1/2]$, where $q(\boldsymbol{x}) \equiv \mathbb{P}(Y = +1 \mid \boldsymbol{X} = \boldsymbol{x})$.*

Fisher consistency of the DWSVM loss function ensures that the sign of the minimizer of the expected loss function (subject to the parallel condition) coincides with the Bayes rule.

6.2 Asymptotic normality

Koo *et al.* (2008) has studied the asymptotic normality of the coefficient vector for the SVM classifier. We follow the same direction and prove the corresponding results for the DWSVM classifier.

For ease of presentation of the theorem, we let $\boldsymbol{\omega}_+$ denote the augmented parameter vector $(\beta_0, \beta, \boldsymbol{\omega}^T)^T \in \mathbb{R}^{d+2}$, $\boldsymbol{x}_+$, $\boldsymbol{x}_\dagger$ and $\boldsymbol{x}_\ddagger$ the augmented data vectors $(0, 1, \boldsymbol{x}^T)^T \in \mathbb{R}^{d+2}$, $(1, 0, \boldsymbol{x}^T)^T \in \mathbb{R}^{d+2}$ and $(1, 1, \boldsymbol{x}^T)^T \in \mathbb{R}^{d+2}$. Consequently, the main discriminant function $f(\boldsymbol{x}; \boldsymbol{\omega}_+) \equiv {\boldsymbol{x}_+}^T \boldsymbol{\omega}_+ = \boldsymbol{x}^T\boldsymbol{\omega} + \beta$, and the axillary discriminant function $f_0(\boldsymbol{x}; \boldsymbol{\omega}_+) \equiv {\boldsymbol{x}_\dagger}^T \boldsymbol{\omega}^+ = \boldsymbol{x}^T\boldsymbol{\omega} + \beta_0$.

We cast DWSVM to an optimization problem with an unconstrained objective function.

$$q_{\lambda,n}(\boldsymbol{\omega}_+) \equiv \frac{1}{n}\sum_{i=1}^{n} L(\boldsymbol{x}_i, y_i, \boldsymbol{\omega}_+) + \frac{\lambda}{2}\|\boldsymbol{\omega}\|^2 \tag{17}$$

$$= \frac{1}{n}\sum_{i=1}^{n} \{\alpha V_{C_d}(y_i f_0(\boldsymbol{x}_i; \boldsymbol{\omega}_+)) + (1-\alpha) H_{C_s}(y_i f(\boldsymbol{x}_i; \boldsymbol{\omega}_+))\} + \frac{\lambda}{2}\|\boldsymbol{\omega}\|^2 \tag{18}$$

The solution to the optimization problem can be scaled by the norm of $\boldsymbol{\omega}$ so as to make it have unit norm.

The population version of (18) without the penalty term is defined as

$$Q(\boldsymbol{\omega}_+) \equiv \mathbb{E}\{\alpha V_{C_d}(Yf_0(\boldsymbol{X}; \boldsymbol{\omega}_+)) + (1-\alpha) H_{C_s}(Yf(\boldsymbol{X}; \boldsymbol{\omega}_+))\},$$

whose minimizer is defined as $\boldsymbol{\omega}_+^* \equiv \text{argmin}_{\boldsymbol{\omega}_+} Q(\boldsymbol{\omega}_+)$.

For easy presentation, let

$$\begin{aligned}
g(\boldsymbol{x}, y, \boldsymbol{\omega}_+) &\equiv \alpha\left(-\mathbb{1}_{\{yf_0(\boldsymbol{x};\boldsymbol{\omega}_+)\le 1/\sqrt{C_d}\}} C_d \right.\\
&\qquad \left. - \mathbb{1}_{\{yf_0(\boldsymbol{x};\boldsymbol{\omega}_+)>1/\sqrt{C_d}\}} 1/[yf_0(\boldsymbol{x};\boldsymbol{\omega}_+)]^2\right),\\
h(\boldsymbol{x}, y, \boldsymbol{\omega}_+) &\equiv (1-\alpha)\left(-\mathbb{1}_{\{yf(\boldsymbol{x};\boldsymbol{\omega}_+)\le 1/\sqrt{C_s}\}} C_s\right),\\
v(\boldsymbol{x}, y, \boldsymbol{\omega}_+) &\equiv \alpha\left(\mathbb{1}_{\{yf_0(\boldsymbol{x};\boldsymbol{\omega}_+)>1/\sqrt{C_d}\}} 1/[yf_0(\boldsymbol{x};\boldsymbol{\omega}_+)]^3\right),\\
w(\boldsymbol{x}, y, \boldsymbol{\omega}_+) &\equiv (1-\alpha)\delta\left(1/\sqrt{C_s} - yf(\boldsymbol{x};\boldsymbol{\omega}_+)\right) C_s,
\end{aligned}$$

where $\delta(\cdot)$ denotes the Dirac delta function. Furthermore, let

$$\begin{aligned}
S(\boldsymbol{\omega}_+) &\equiv \mathbb{E}\{g(\boldsymbol{X}, Y, \boldsymbol{\omega}_+) Y \boldsymbol{X}_\dagger + h(\boldsymbol{X}, Y, \boldsymbol{\omega}_+) Y \boldsymbol{X}_+\} \text{ and}\\
U(\boldsymbol{\omega}_+) &\equiv \mathbb{E}\left\{v(\boldsymbol{X}, Y, \boldsymbol{\omega}_+)\boldsymbol{X}_\dagger \boldsymbol{X}_\dagger^T + w(\boldsymbol{x}, y, \boldsymbol{\omega}_+)\boldsymbol{X}_+\boldsymbol{X}_+^T\right\}.
\end{aligned}$$

Let $\Omega(\boldsymbol{X}_i, Y_i, \boldsymbol{\omega}_+^*) = \mathrm{diag}\{g(\boldsymbol{X}_i, Y_i, \boldsymbol{\omega}_+^*), h(\boldsymbol{X}_i, Y_i, \boldsymbol{\omega}_+^*), [g(\boldsymbol{X}_i, Y_i, \boldsymbol{\omega}_+^*) + h(\boldsymbol{X}_i, Y_i, \boldsymbol{\omega}_+^*)]\mathbf{I}_d\}$, where $\mathbf{I}_d$ is $d \times d$ identity matrix.

Then, define

$$T_n \equiv \sum_{i=1}^{n} \left\{ g(\boldsymbol{X}_i, Y_i, \boldsymbol{\omega}_+^*) Y_i (\boldsymbol{X}_i)_\dagger + h(\boldsymbol{X}_i, Y_i, \boldsymbol{\omega}_+^*) Y_i (\boldsymbol{X}_i)_+ \right\},$$
$$= \sum_{i=1}^{n} Y_i \left\{ \Omega(\boldsymbol{X}_i, Y_i, \boldsymbol{\omega}_+^*)(\boldsymbol{X}_i)_\ddagger \right\}.$$

Lastly, define $G(\boldsymbol{\omega}_+^*) \equiv \mathbb{E}[(\boldsymbol{X}_i)_\ddagger \Omega^2(\boldsymbol{X}_i, Y_i, \boldsymbol{\omega}_+^*)(\boldsymbol{X}_i)_\ddagger{}^T]$.

Some regularity conditions are needed. We state the conditions in the appendix. Note that conditions (A1), (A2) and (A4) are the same as in Koo *et al.* (2008). Our new (A3) is tailored for DWSVM and incorporates the DWD component. In particular, (A1) ensures that $U(\boldsymbol{\omega}^+)$ is well-defined and is continuous in $\boldsymbol{\omega}_+$ while (A1) and (A2) ensure that the minimizer $\boldsymbol{\omega}_+^*$ exists. (A3) is a sufficient condition to that $\boldsymbol{\omega}_+^*$ is not zero. (A4) guarantees the positive-definiteness of $U(\boldsymbol{\omega}_+)$ around $\boldsymbol{\omega}_+^*$.

Under these regularity conditions, we obtain a Bahadur representation of $\widehat{\boldsymbol{\omega}_{\lambda,n}}_+$ in Theorem 2, the asymptotic normality in Theorem 3, and consequently, the asymptotic normality of the discriminant function $f(\boldsymbol{x}; \widehat{\boldsymbol{\omega}_{\lambda,n}}_+)$ at $\boldsymbol{x}$ in Corollary 4.

Theorem 2. *Suppose that (A1)–(A4) are met. For* $\lambda = o(n^{-1/2})$*, we have*

$$\sqrt{n}(\widehat{\boldsymbol{\omega}_{\lambda,n}}_+ - \boldsymbol{\omega}_+^*) = -\frac{1}{\sqrt{n}} U(\boldsymbol{\omega}_+^*)^{-1} T_n + o_P(1).$$

Theorem 3. *Suppose that (A1)–(A4) are met. For* $\lambda = o(n^{-1/2})$*, we have*

$$\sqrt{n}(\widehat{\boldsymbol{\omega}_{\lambda,n}}_+ - \boldsymbol{\omega}_+^*) = N\left(\mathbf{0}, U(\boldsymbol{\omega}_+^*)^{-1} G(\boldsymbol{\omega}_+^*) U(\boldsymbol{\omega}_+^*)^{-1}\right)$$

This will lead to the following corollary.

Corollary 4. *Under the same conditions as in Theorem 3, for* $\lambda = o(n^{-1/2})$ *and any* $\boldsymbol{x} \in \mathbb{R}^d$*,*

$$\sqrt{n}\left(f(\boldsymbol{x}, \widehat{\boldsymbol{\omega}_{\lambda,n}}_+) - f(\boldsymbol{x}, \boldsymbol{\omega}_+^*)\right)$$
$$\xrightarrow{d} N\left(\mathbf{0}, \boldsymbol{x}_+^T U(\boldsymbol{\omega}_+^*)^{-1} G(\boldsymbol{\omega}_+^*) U(\boldsymbol{\omega}_+^*)^{-1} \boldsymbol{x}_+\right)$$

6.3 Extremely imbalanced data

Owen (2007) discussed the behavior of the intercept term in the logistic regression when the sample size of one class is extremely large while that of the other class is fixed. Moreover, Qiao and Zhang (2013) also showed that the intercept term of DWD diverges. In this subsection, we prove that the intercept term for the DWSVM classifier does not diverge. Without loss of generality, we assume that $n_- \gg n_+$, that is, the negative class is the majority class.

Lemma 5. *Suppose that the negative majority class is sampled from a distribution with compact support* $\mathcal{S}$*. Then the intercept term* β *in SVM does not diverge to negative infinity when* $n_- \to \infty$*.*

Corollary 6. *Suppose that the negative majority class is sampled from a distribution with compact support* $\mathcal{S}$*. Then the intercept term* β *in DWSVM does not diverge to negative infinity when* $n_- \to \infty$*.*

The assumption of compact support $\mathcal{S}$ is essential here, but it is fairly weak and is true in many real applications. Note that this result does not ensure that the sensitivity issue is completely overcome by SVM or DWSVM. Instead, it suggests that in the $n_- \to \infty$ asymptotics, the impact of the imbalanced sample size is limited to some extent.

7. CONCLUSION

Both SVM and DWD are subject to certain disadvantages and enjoy certain advantages. The DWSVM combines the merits of both methods by creatively deploying an axillary intercept term. We have shown standard asymptotic results for the DWSVM classifier. The simulations and real data application establish the superiority of the DWSVM method over SVM and DWD in some situations. In particular, the DWSVM method can lead to a discriminant direction vector that, like the DWD direction, preserve important features of the data set. More importantly, the DWSVM also performs very well in terms of classification. As a bottom line, its performance is just as good as the SVM. In special settings such as the perturbed data, we have demonstrated that DWSVM can overcome overfitting and is more robust against perturbation/mislabeling of the data.

We have shown some asymptotic properties of DWSVM in this article. More work can be done to investigate its statistical properties, for example, in the line of Blanchard, Bousquet and Massart (2008).

In DWSVM, the direction is jointly optimized by both the DWD component and the SVM component. The relative contributions of both are controlled by the tuning parameter α. When $\alpha = 0$, DWSVM is exactly identical to SVM. On the other hand, one would not want to have $\alpha = 1$, since that would make the Hinge loss equal to zero and the choice of the main intercept term β meaningless (since it will disappear from the objective function). A sensitivity analysis in Section 4.2 suggests that, within a large range between 0 and 1, the impact of the value of α on the classification performance and interpretability of the resulting DWSVM classifier is very small. One may notice from Figures 4 and 5 that, ideally, an α value quite close to 1 (but not too close so as to cause the problem as above) can potentially produce improved interpretability (smaller angle) over a range of C_{svm} values, while maintaining classification performance. However, as the potential improvement seems to be limited, and since a wise tuning of C_{svm} can potentially find a DWSVM classifier almost as good, the effort of pursuing the direction of tuning α may not be justified. This is the reason we have used $\alpha = 1/2$ in the paper, which turns out to be quite useful.

An instant extension of the DWSVM classifier is multiclass classification. For example, for a multiclass classification problem with K classes, the following optimization problem accomplishes such an extension.

$$\underset{\boxed{\boldsymbol{\omega}_j,\beta_j}\,\beta_{j0},\boldsymbol{\xi},\boldsymbol{\eta}}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{y_i=j,\ k\neq j} \left\{\alpha\left(\frac{1}{r^i_{jk}} + C_{dwd}\cdot\eta^i_{jk}\right) + (1-\alpha)\xi^i_{jk}\right\},$$

$$\begin{aligned}
\text{s.t.}\quad & r^i_{jk} = y_i\{\boldsymbol{x}_i^T(\boldsymbol{\omega}_j - \boldsymbol{\omega}_k) + (\beta_{j0} - \beta_{k0})\} + \eta^i_{jk},\\
& r^i_{jk} \ge 0 \text{ and } \eta^i_{jk} \ge 0,\\
& C_{svm} y_i\{\boldsymbol{x}_i^T(\boldsymbol{\omega}_j - \boldsymbol{\omega}_k) + (\beta_j - \beta_k)\} + \xi^i_{jk} \ge \sqrt{C_{svm}},\\
& \xi^i_{jk} \ge 0,\\
& \sum_{j=1}^{K} \|\boldsymbol{\omega}_j\|^2 \le 1,\\
& \sum_{j=1}^{K} \boldsymbol{\omega}_j = \boldsymbol{0},\ \sum_{j=1}^{K} \beta_j = 0,\ \sum_{j=1}^{K} \beta_{j0} = 0.
\end{aligned}$$

Other extensions such as kernel DWSVM or sparse DWSVM are also readily in order.

In summary, DWSVM integrates the merits of classical classification methods. Its numerical performance is very good and it is theoretically justified. These show evidence that it is a very promising linear learner which has great potential in many applications.

The DWSVM classifier has been implemented in MATLAB. We have used the second order cone programming to implement it, whose computational cost is comparable to that of DWD. See the appendix for details on the implementation of this algorithm. Future work will also concentrate on developing more efficient implementations of DWSVM.

APPENDICES

Proof of Theorem 1

For any $\boldsymbol{x}$, denote $q(\boldsymbol{x}) = \mathbb{P}(Y = +1|\boldsymbol{X} = \boldsymbol{x})$. The conditional risk is

$$\begin{aligned}
R(f, f_0) &\equiv E[L\{Yf(\boldsymbol{X}), Yf_0(\boldsymbol{X})\} \mid \boldsymbol{X} = \boldsymbol{x}]\\
&= \{\alpha V_{C_{dwd}}(f_0) + (1-\alpha)H_{C_{svm}}(f)\}\, q(\boldsymbol{x})\\
&\quad + \{\alpha V_{C_{dwd}}(-f_0) + (1-\alpha)H_{C_{svm}}(-f)\}\,\{1 - q(\boldsymbol{x})\},
\end{aligned}$$

where for simplicity we write $f(\boldsymbol{x})$ and $f_0(\boldsymbol{x})$ as f and f_0.

For the global minimizer (f^*, f_0^*), since $f^* - f_0^* = \Delta^*$ is independent of $\boldsymbol{x}$, we can consider another feasible (but not optimal) solution $(-f^*, -f^* - \Delta^*)$. Due to the optimality of $(f^*, f_0^*) = (f^*, f^* - \Delta^*)$, we can show that

$$\begin{aligned}
0 &\ge R(f^*, f^* - \Delta^*) - R(-f^*, -f^* - \Delta^*)\\
&= \{2q(\boldsymbol{x}) - 1\}\,[\{\alpha V_{C_{dwd}}(f^* - \Delta^*) + (1-\alpha)H_{C_{svm}}(f^*)\}\\
&\qquad - \{\alpha V_{C_{dwd}}(-f^* - \Delta^*) + (1-\alpha)H_{C_{svm}}(-f^*)\}]\\
&= \{2q(\boldsymbol{x}) - 1\}\,[\alpha\{V_{C_{dwd}}(f^* - \Delta^*) - V_{C_{dwd}}(-f^* - \Delta^*)\}\\
&\qquad + (1-\alpha)\{H_{C_{svm}}(f^*) - H_{C_{svm}}(-f^*)\}]
\end{aligned}$$

Thus if $q(\boldsymbol{x}) > 1/2$, then

$$\begin{aligned}
&\alpha\{V_{C_{dwd}}(f^* - \Delta^*) - V_{C_{dwd}}(-f^* - \Delta^*)\}\\
&\quad + (1-\alpha)\{H_{C_{svm}}(f^*) - H_{C_{svm}}(-f^*)\} \le 0.
\end{aligned}$$

Because $V_{C_{dwd}}(\cdot)$ is strictly decreasing everywhere, and $H_{C_{svm}}(\cdot)$ is strictly decreasing around 0, we have that $V_{C_{dwd}}(f^* - \Delta^*) - V_{C_{dwd}}(-f^* - \Delta^*)$ and $H_{C_{svm}}(f^*) - H_{C_{svm}}(-f^*)$ have the same sign, and hence $f^* \ge 0$. By a similar argument, if $q(\boldsymbol{x}) < 1/2$, then $f^* \le 0$. Lastly, it is easy to show that $f^* \neq 0$. Hence we have $\operatorname{sign}(f^*) = \operatorname{sign}(q(\boldsymbol{x}) - 1/2)$.

Regularity conditions

We state the regularity conditions for the asymptotics below. We use C_1, C_2, ... to denote positive constants independent of n.

A1 The densities p_+ and p_- are continuous and have finite second moments.

A2 There exists $B(\boldsymbol{x}_0, \delta_0)$, a ball centered at $\boldsymbol{x}_0$ with radius δ_0 such that $p_1(\boldsymbol{x}) > C_1$ and $p_2(\boldsymbol{x}) > C_1$ for every $\boldsymbol{x} \in B(\boldsymbol{x}_0, \delta_0)$.

A3 For some $1 \le l \le d$,

$$\mathbb{E}\left(\mathbb{1}_{\{X_l \ge F_-^L\}} X \mid Y = -1\right) < \mathbb{E}\left(\mathbb{1}_{\{X_l \le F_+^U\}} X \mid Y = +1\right)$$

or

$$\mathbb{E}\left(\mathbb{1}_{\{X_l \le F_-^U\}} X \mid Y = -1\right) > \mathbb{E}\left(\mathbb{1}_{\{X_l \ge F_+^L\}} X \mid Y = +1\right),$$

where F_+^L andF_-^L (F_+^U and F_-^U, respectively) are the lower bounds (upper bounds, respectively) for the positive and negative classes. They are defined as

$$\begin{aligned}
&\mathbb{P}\left(X_l \ge F_+^L \mid Y = +1\right)\\
&\quad = \min\left(1, \frac{\pi_+\{\alpha C_d + (1-\alpha)C_s\}}{\pi_-(1-\alpha)C_s}\right),\\
&\mathbb{P}\left(X_l \ge F_-^L \mid Y = +1\right)\\
&\quad = \min\left(1, \frac{\pi_-\{\alpha C_d + (1-\alpha)C_s\}}{\pi_+(1-\alpha)C_s}\right),\\
&\mathbb{P}\left(X_l \le F_+^U \mid Y = +1\right)\\
&\quad = \min\left(1, \frac{\pi_+(1-\alpha)C_s}{\pi_-\{\alpha C_d + (1-\alpha)C_s\}}\right),\\
&\mathbb{P}\left(X_l \le F_-^U \mid Y = +1\right)\\
&\quad = \min\left(1, \frac{\pi_-(1-\alpha)C_s}{\pi_+\{\alpha C_d + (1-\alpha)C_s\}}\right).
\end{aligned}$$

A4 For an orthogonal transformation A_l that maps $\boldsymbol{\omega}^*/\|\boldsymbol{\omega}^*\|$ to the lth unit basis vector e_l for some $1 \leq l \leq d$, there exist rectangles

$$\begin{aligned}\mathcal{D}^+ = \{&\boldsymbol{x} \in M^+ : l_s \\ &\leq (A_l\boldsymbol{x})_s \leq v_s \text{ with } l_s < v_s \text{ for } s \neq l\}\end{aligned}$$

and

$$\begin{aligned}\mathcal{D}^- = \{&\boldsymbol{x} \in M^- : l_s \\ &\leq (A_l\boldsymbol{x})_s \leq v_s \text{ with } l_s < v_s \text{ for } s \neq l\}\end{aligned}$$

such that $p_+(\boldsymbol{x}) \geq C_2 > 0$ on $\mathcal{D}^+$ and $p_-(\boldsymbol{x}) \geq C_3 > 0$ on $\mathcal{D}^-$, where $M^+ \equiv \{\boldsymbol{x} : \boldsymbol{x}^T\boldsymbol{\omega}^* + \beta = 1/\sqrt{C_s}\}$ and $M^- \equiv \{\boldsymbol{x} : \boldsymbol{x}^T\boldsymbol{\omega}^* + \beta = -1/\sqrt{C_s}\}$.

Proof of Theorems 2 and 3 and Corollary 4

For fixed $\boldsymbol{\theta} \in \mathbb{R}^{d+2}$, define

$$\begin{aligned}&\Lambda_n(\boldsymbol{\theta}) \equiv n\left\{q_{\lambda,n}(\boldsymbol{\omega}_+^* + \boldsymbol{\theta}/\sqrt{n}) - q_{\lambda,n}(\boldsymbol{\omega}_+^*)\right\}, \text{ and} \\ &\Gamma_n(\boldsymbol{\theta}) \equiv \mathbb{E}\Lambda_n(\boldsymbol{\theta}).\end{aligned}$$

Observe that

$$\begin{aligned}\Gamma_n(\boldsymbol{\theta}) =& n\left\{Q(\boldsymbol{\omega}_+^* + \boldsymbol{\theta}/\sqrt{n}) - Q(\boldsymbol{\omega}_+^*)\right\} \\ &+ \frac{\lambda}{2}\left(\|\boldsymbol{\theta}_{3:(d+2)}\|^2 + 2\sqrt{n}\boldsymbol{\omega}^{*T}\boldsymbol{\theta}_{3:(d+2)}\right)\end{aligned}$$

By Taylor series expansion of Q around $\boldsymbol{\omega}_+^*$, we obtain, for some $0 < t < 1$,

$$\begin{aligned}\Gamma_n(\boldsymbol{\theta}) =& \frac{1}{2}\boldsymbol{\theta}^T U\left(\boldsymbol{\omega}_+^* + (t/\sqrt{n})\boldsymbol{\theta}\right)\boldsymbol{\theta} \\ &+ \frac{\lambda}{2}\left(\|\boldsymbol{\theta}_{3:(d+2)}\|^2 + 2\sqrt{n}\boldsymbol{\omega}^{*T}\boldsymbol{\theta}_{3:(d+2)}\right).\end{aligned}$$

Because $U(\boldsymbol{\omega}_+)$ is continuous in $\boldsymbol{\omega}_+$, due to condition (A1), we have

$$\frac{1}{2}\boldsymbol{\theta}^T U\left(\boldsymbol{\omega}_+^* + (t/\sqrt{n})\boldsymbol{\theta}\right)\boldsymbol{\theta} = \frac{1}{2}\boldsymbol{\theta}^T U\left(\boldsymbol{\omega}_+^*\right)\boldsymbol{\theta} + o(1).$$

This, combined with $\lambda = o(n^{-1/2})$, results in

$$\Gamma_n(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^T U\left(\boldsymbol{\omega}_+^*\right)\boldsymbol{\theta} + o(1).$$

Now, observe that $\mathbb{E}T_n = nS(\boldsymbol{\omega}_+^*) = \mathbf{0}$ and $\mathbb{E}(T_nT_n^T) = \sum_{i=1}^n \mathbb{E}[(\boldsymbol{X}_i)_\ddagger\Omega^2(\boldsymbol{X}_i, Y_i, \boldsymbol{\omega}_+^*)(\boldsymbol{X}_i)_\ddagger{}^T] = nG(\boldsymbol{\omega}_+^*)$. Hence, $\frac{1}{\sqrt{n}}T_n$ follows $N(0, G(\boldsymbol{\omega}_+^*))$ asymptotically by central limit theorem.

Next, we define

$$\begin{aligned}R_{i,n}(\boldsymbol{\theta}) \equiv& L_{i,n}(\boldsymbol{\omega}_+^* + \boldsymbol{\theta}/\sqrt{n}) - L_{i,n}(\boldsymbol{\omega}_+^*) \\ &- \left(\frac{\partial L_{i,n}}{\partial\boldsymbol{\omega}_+}(\boldsymbol{\omega}_+)\bigg|_{\boldsymbol{\omega}_+=\boldsymbol{\omega}_+^*}\right)^T\boldsymbol{\theta}/\sqrt{n},\end{aligned}$$

where $L_{i,n}(\boldsymbol{\omega}_+) \equiv \alpha V_{C_d}(Y_i(\boldsymbol{X}_i)_\dagger^T\boldsymbol{\omega}_+) + (1 - \alpha)H_{C_s}(Y_i(\boldsymbol{X}_i)_+^T\boldsymbol{\omega}_+)$.

We continue by splitting $R_{i,n}$ to two parts $R_{i,n} = \alpha R_{i,n}^d + (1-\alpha)R_{i,n}^s$, where the first term concerns the DWD component and the second term concerns the SVM component.

For the DWD component,

$$\begin{aligned}R_{i,n}^d(\boldsymbol{\theta}) \equiv& V(Y_i(\boldsymbol{X}_i)_\dagger^T(\boldsymbol{\omega}_+^* + \boldsymbol{\theta}/\sqrt{n})) - V(Y_i(\boldsymbol{X}_i)_\dagger^T\boldsymbol{\omega}_+^*) \\ &- \left(\frac{\partial V}{\partial\boldsymbol{\omega}_+}(\boldsymbol{\omega}_+)\bigg|_{\boldsymbol{\omega}_+=\boldsymbol{\omega}_+^*}\right)^T\boldsymbol{\theta}/\sqrt{n}\end{aligned}$$

Because the DWD loss V has first order continuous derivative, $R_{i,n}^d(\boldsymbol{\theta}) = O(n^{-1})$.

For the SVM component,

$$\begin{aligned}R_{i,n}^s(\boldsymbol{\theta}) \equiv& H[Y_i(\boldsymbol{X}_i)_+^T(\boldsymbol{\omega}_+^* + \boldsymbol{\theta}/\sqrt{n})] - H[Y_i(\boldsymbol{X}_i)_+^T\boldsymbol{\omega}_+^*] \\ &+ \sqrt{C_s}\mathbb{1}_{\{Y_i(\boldsymbol{X}_i)_+^T\boldsymbol{\omega}_+^* < 1/\sqrt{C_s}\}}Y_i(\boldsymbol{X}_i)_+^T\boldsymbol{\theta}/\sqrt{n}.\end{aligned}$$

Following the argument by Koo *et al.* (2008) and combining the fact that $R_{i,n}^d(\boldsymbol{\theta}) = O(n^{-1})$, we can show that $\sum_{i=1}^n \mathbb{E}(|R_{i,n}(\boldsymbol{\theta}) - \mathbb{E}R_{i,n}(\boldsymbol{\theta})|^2) \to 0$, as $n \to 0$

We note that $\Lambda_n(\boldsymbol{\theta}) = \Gamma_n(\boldsymbol{\theta}) + T_n^T\boldsymbol{\theta}/\sqrt{n} + \sum_{i=1}^n(R_{i,n}(\boldsymbol{\theta}) - \mathbb{E}R_{i,n}(\boldsymbol{\theta}))$. Thus

$$\Lambda_n(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^T U\left(\boldsymbol{\omega}_+^*\right)\boldsymbol{\theta} + T_n^T\boldsymbol{\theta}/\sqrt{n} + o_P(1).$$

By the Convexity Lemma in Pollard (1991), we have for any fixed θ,

$$\begin{aligned}\Lambda_n(\boldsymbol{\theta}) =& \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\zeta}_n)^T U\left(\boldsymbol{\omega}_+^*\right)(\boldsymbol{\theta} - \boldsymbol{\zeta}_n) \\ &+ \frac{1}{2}\boldsymbol{\zeta}_n^T U\left(\boldsymbol{\omega}_+^*\right)\boldsymbol{\zeta}_n + r_n(\theta),\end{aligned}$$

where $\boldsymbol{\zeta}_n \equiv -U(\boldsymbol{\omega}_+^*)^{-1}T_n/\sqrt{n}$, and for each compact set $K \in \mathbb{R}^d$,

$$\sup_{\boldsymbol{\theta}\in K}|r_n(\boldsymbol{\theta})| \xrightarrow{p} 0.$$

We then follow the argument in Koo *et al.* (2008) and have for each $\varepsilon > 0$ and $\widehat{\boldsymbol{\theta}}_{\lambda,n} = \sqrt{n}(\widehat{\boldsymbol{\omega}_{\lambda,n}}_+ - \boldsymbol{\omega}_+^*)$,

$$\mathbb{P}\left(\|\widehat{\boldsymbol{\theta}}_{\lambda,n} - \boldsymbol{\zeta}_n\| > \varepsilon\right) \xrightarrow{p} 0,$$

which completes the proof. □

Proof of Lemma 5

We prove the result for the simpler and more intuitive case of $d = 1$. In this case $\boldsymbol{\omega} \in \mathbb{R}$ does not need to be optimized. We can simply assume that $\boldsymbol{\omega} = 1$. Moreover, we can consider the worst case scenario where $n_+ = 1$. This is the worse case because this represents the most imbalanced sample sizes. We let x_0 denote the sole data vector in the positive minority class.

Since the negative class is extremely large compared to the positive, we can assume that the functional margin with respect to the main hyperplane $u \equiv y_0(x_0+\beta) = x_0+\beta$ for the data vectors from the positive minority class are always less than $1/\sqrt{C_s}$, that is $\beta \leq 1/\sqrt{C_s} - x_0$.

Write the objective function of SVM as

$$\begin{aligned}\mathcal{L}^s(\beta) \equiv & (\sqrt{C_s} - C_s x_0 - C_s\beta) \\ & + \sum_{i=1}^{n} \left\{ \mathbb{1}_{\{y_i=-1\}} (\sqrt{C_s} + C_s x_i + C_s\beta)_+ \right\} \\ \approx & (\sqrt{C_s} - C_s x_0 - C_s\beta) \\ & + n_- \mathbb{E} \left\{ (\sqrt{C_s} + C_s X + C_s\beta)_+ \mid Y = -1 \right\}\end{aligned}$$

Note that

$$\begin{aligned}\frac{\partial \mathcal{L}^s}{\partial \beta}(\beta) \equiv & -C_s + n_- C_s \mathbb{E}\left[\mathbb{1}_{\{\sqrt{C_s}+C_s X + C_s\beta > 0\}} \mid Y = -1 \right] \\ = & -C_s + n_- C_s \mathbb{P}\left[\sqrt{C_s} + C_s X + C_s \beta > 0 \mid Y = -1 \right].\end{aligned}$$

This leads to that $\lim_{\beta\to-\infty} \frac{\partial \mathcal{L}^s}{\partial \beta}(\beta) = -C_s < 0$, and

$$\begin{aligned}&\frac{\partial \mathcal{L}^s}{\partial \beta}\left(-M - 1/\sqrt{C_s}\right) \\ = & -C_s + n_- C_s \mathbb{P}\Big[\sqrt{C_s} + C_s X \\ & \quad + C_s(-M - 1/\sqrt{C_s}) > 0 \mid Y = -1 \Big] \\ = & -C_s + n_- C_s \mathbb{P}\left[X > M \mid Y = -1 \right] = -C_s < 0\end{aligned}$$

Thus if $1/\sqrt{C_s} - x_0 \leq -M - 1/\sqrt{C_s}$, then

$$\begin{aligned}&\frac{\partial \mathcal{L}^s}{\partial \beta}\left(1/\sqrt{C_s} - x_0\right) \\ = & -C_s + n_- C_s \mathbb{P}\Big[\sqrt{C_s} + C_s X \\ & \quad + C_s(1/\sqrt{C_s} - x_0) > 0 \mid Y = -1 \Big] \\ = & -C_s + n_- C_s \mathbb{P}\left[X > x_0 - 2/\sqrt{C_s} \mid Y = -1 \right] \\ = & -C_s < 0,\end{aligned}$$

and $\beta = 1/\sqrt{C_s} - x_0$ is the minimizer of $\mathcal{L}^s$. On the other hand, if $1/\sqrt{C_s} - x_0 > -M - 1/\sqrt{C_s}$, then the minimizer β^* will be greater than $-M - 1/\sqrt{C_s}$ but less than or equal to $1/\sqrt{C_s} - x_0$. This means that the intercept term β in SVM does not diverge to $-\infty$. □

Implementation of the DWSVM method

In this subsection, we describe how to implement the DWSVM by using the second-order cone programming. Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^T$ be a d-dimensional vector. We call that $\boldsymbol{x}$ lies in a d-dimensional second-order cone $\mathcal{S}_d$, if $x_1 \geq \sqrt{x_2^2 + \cdots + x_d^2}$. Similar to the DWD implementation, we will introduce triplets $(\rho_i, \sigma_i, \tau_i)$ so that

$$\rho_i + \sigma_i = \frac{1}{r_i}, \quad \rho_i - \sigma_i = r_i.$$

Thus, $\rho_i^2 = \sigma_i^2 + 1$. Then if we let $\tau_i = 1$, we have $(\rho_i, \sigma_i, \tau_i) \in \mathcal{S}_3$. Thus, Equations (11–14) will be

$$\begin{aligned}&\underset{\boxed{\boldsymbol{\omega}, \beta}\, \beta_0, \xi_i, \eta_i}{\text{argmin}} \sum_{i=1}^{n} \{\alpha(\rho_i + \sigma_i + C_{dwd}\eta_i) + (1-\alpha)\xi_i\}, \\ \text{s.t.} \quad & y_i(\boldsymbol{x}_i^T \boldsymbol{\omega} + \beta_0) + \eta_i - \rho_i + \sigma_i = 0 \\ & C_{svm} y_i(\boldsymbol{x}_i^T w + \beta) + \xi_i - \sqrt{C_{svm}} \geq 0 \\ & \tau = 1 \text{ and } \tau_i = 1 \\ & (\tau, \boldsymbol{\omega}) \in \mathcal{S}_{d+1}, \ (\rho_i, \sigma_i, \tau_i) \in \mathcal{S}_3, \ \eta_i \geq 0, \ \xi_i \geq 0.\end{aligned}$$

Standard second order cone programming packages can be used to implement this optimization problem. In our implementation, we used CVX, a package for specifying and solving convex programs (Grant and Boyd, 2008, 2013). For high-dimensional, low-sample size data, the computing cost of the optimization above is in the same order as that of DWD. In particular, the number of variable is $d + 2 + 2n$ compared to $d + 1 + n$. When $d \gg n$, the increment is relatively small. The number of constraints doubles compared to DWD.

ACKNOWLEDGEMENTS

The first author's work was partially supported by Binghamton University Harpur College Dean's New Faculty Start-up Funds and a collaboration grant from the Simons Foundation (#246649 to Xingye Qiao). Both authors thank the Statistical and Applied Mathematical Sciences Institute (the 2012–13 Program on Statistical and Computational Methodology for Massive Datasets and the 2013–14 Program on Low-dimensional Structure in High-dimensional Systems) for their generous support where both authors have spent considerable amount of time when writing this article.

This material is based upon work partially supported by the National Science Foundation under Grant DMS-1127914 to the Statistical and Applied Mathematical Sciences Institute. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Received 23 February 2014

REFERENCES

Ahn, J. and Marron, J. (2010). The maximal data piling direction for discrimination. *Biometrika* **97** 254–259. MR2594434

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. MR2268032

Blanchard, G., Bousquet, O. and Massart, P. (2008). Statistical performance of support vector machines. *The Annals of Statistics* 489–531. MR2396805

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning* **20** 273–297.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods*. Cambridge University Press.

Duda, R. O., Hart, P. E. and Stork, D. G. (2001). *Pattern Classification*. Wiley. MR1802993

Dudoit, S., Fridlyand, J. and Speed, T. P. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association* **97** 77–87. MR1963389

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* **55** 119–139. MR1473055

Friedman, J., Hastie, T. and Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *Annals of Statistics* 337–374. MR1790002

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A. et al. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* **286** 531.

Grant, M. C. and Boyd, S. P. (2008). Graph implementations for nonsmooth convex programs. In *Recent advances in learning and control* 95–110. Springer. MR2409077

Grant, M. and Boyd, S. (2013). CVX: Matlab software for disciplined convex programming, version 2.0 beta. http://cvxr.com/cvx.

Hastie, T., Tibshirani, R. and Friedman, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (second edition)*. Springer. MR2722294

Koo, J. Y., Lee, Y., Kim, Y. and Park, C. (2008). A Bahadur Representation of the Linear Support Vector Machine. *Journal of Machine Learning Research* **9** 1343–1368. MR2426045

Lin, Y. (2004). A note on margin-based loss functions in classification. *Statistics & Probability Letters* **68** 73–82. MR2064687

Marron, J. S., Todd, M. J. and Ahn, J. (2007). Distance-weighted discrimination. *Journal of the American Statistical Association* **102** 1267–1271. MR2412548

Owen, A. B. (2007). Infinitely imbalanced logistic regression. *The Journal of Machine Learning Research* **8** 761–773. MR2320678

Pollard, D. (1991). Asymptotics for least absolute deviation regression estimators. *Econometric Theory* **7** 186–199. MR1128411

Qiao, X. and Zhang, L. (2013). Flexible high-dimensional classification machines and their asymptotic properties.

Qiao, X., Zhang, H. H., Liu, Y., Todd, M. J. and Marron, J. S. (2010). Weighted distance weighted discrimination and its asymptotic properties. *Journal of the American Statistical Association* **105** 401–414. MR2656058

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley. MR1641250

Xingye Qiao
Department of Mathematical Sciences
Binghamton University, State University of New York
Binghamton, NY 13902-6000
USA
E-mail address: qiao@math.binghamton.edu

Lingsong Zhang
Department of Statistics
Purdue University
West Lafayette, IN 47907
USA
E-mail address: lingsong@purdue.edu