Another Look at DWD: Thrifty Algorithm and Bayes Risk Consistency in RKHS

Boxiang Wang* and Hui Zou †

August 21, 2015

Abstract

Distance weighted discrimination (DWD) is a margin-based classifier with an interesting geometric motivation. DWD was originally proposed as a superior alternative to the support vector machine (SVM), however DWD is yet to be popular compared with the SVM. The main reasons are twofold. First, the state-of-the-art algorithm for solving DWD is based on the second-order-cone programming (SOCP), while the SVM is a quadratic programming problem which is much more efficient to solve. Second, the current statistical theory of DWD mainly focuses on the linear DWD for the high-dimension-low-sample-size setting and data-piling, while the learning theory for the SVM mainly focuses on the Bayes risk consistency of the kernel SVM. In fact, the Bayes risk consistency of DWD is presented as an open problem in the original DWD paper. In this work, we advance the current understanding of DWD from both computational and theoretical perspectives. We propose a novel efficient algorithm for solving DWD, and our algorithm can be several hundred times faster than the existing state-of-the-art algorithm based on the SOCP. In addition, our algorithm can handle the generalized DWD, while the SOCP algorithm only works well for a special DWD but not the generalized DWD. Furthermore, we consider a natural kernel DWD in a reproducing kernel Hilbert space and then establish the Bayes risk consistency of the kernel DWD. We compare DWD and the SVM on several benchmark data sets and show that the two have comparable classification accuracy, but DWD equipped with our new algorithm can be much faster to compute than the SVM.

*School of Statistics, University of Minnesota.

†Corresponding author, zouxx019@umn.edu. School of Statistics, University of Minnesota.

Key words: Bayes risk consistency, Classification, DWD, Kernel methods, MM principle, SOCP.

1 Introduction

Binary classification problems appear from diverse practical applications, such as, financial fraud detection, spam email classification, medical diagnosis with genomics data, drug response modeling, among many others. In these classification problems, the goal is to predict class labels based on a given set of variables. Suppose that we observe a training data set consisting of n pairs, where $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, $\boldsymbol{x}_i \in \mathbb{R}^p$, and $y_i \in \{-1, 1\}$. A classifier fits a discriminant function f and constructs a classification rule to classify data point $\boldsymbol{x_i}$ to either class 1 or class -1 according to the sign of $f(\boldsymbol{x}_i)$. The decision boundary is given by $\{\boldsymbol{x} : f(\boldsymbol{x}) = 0\}$. Two canonical classifiers are linear discriminant analysis and logistic regression. Modern classification algorithms can produce flexible non-linear decision boundaries with high accuracy. The two most popular approaches are ensemble learning and support vector machines/kernel machines. Ensemble learning such as boosting (Freund and Schapire, 1997) and random forest (Breiman, 2001) combine many weak learners like decision trees into a powerful one. The support vector machine (SVM) (Vapnik, 1995, 1998) fits an optimal separating hyperplane in the extended kernel feature space which is non-linear in the original covariate spaces. In a recent extensive numerical study by Fernández-Delgado et al. (2014), the kernel SVM is shown to be one of the best among 179 commonly used classifiers.

Motivated by "data-piling" in the high-dimension-low-sample-size problems, Marron et al. (2007) invented a new classification algorithm named distance weighted discrimination (DWD) that retains the elegant geometric interpretation of the SVM and delivers competitive performance. Since then much work has been devoted to the development of DWD. The readers are referred to Marron (2015) for an up-to-date list of work on DWD. On the other hand, we notice that DWD has not attained the popularity it deserves. We can think of two reasons for that. First, the current state-of-the-art algorithm for DWD is based on second-order-cone programming (SOCP) proposed in Marron et al. (2007). SOCP was an essential part of the DWD development. As acknowledged in Marron et al. (2007), SOCP was then much less well-known than quadratic programming, even in optimization. Furthermore, SOCP is generally more computationally demanding than quadratic programming. There are two existing implementations of the SOCP algorithm: Marron (2013) in Matlab and

Huang et al. (2012) in R. With these two implementations, we find that DWD is usually more time-consuming than the SVM. Therefore, SOCP contributes to both the success and unpopularity of DWD. Second, the kernel extension of DWD and the corresponding kernel learning theory are under-developed compared to the kernel SVM. Although Marron et al. (2007) proposed a version of non-linear DWD by mimicking the kernel trick used for deriving the kernel SVM, theoretical justification of such a kernel DWD is still absent. On the contrary, the kernel SVM as well as the kernel logistic regression (Wahba et al., 1994; Zhu and Hasite, 2005) have mature theoretical understandings built upon the theory of reproducing kernel Hilbert space (RKHS) (Wahba, 1999; Hastie et al., 2009). Most learning theories of DWD succeed to Hall et al. (2005)'s geometric view of HDLSS data and assume that $p \to \infty$ and n is fixed, as opposed to the learning theory for the SVM where $n \to \infty$ and p is fixed. We are not against the fixed n and $p \to \infty$ theory but it would be desirable to develop the canonical learning theory for the kernel DWD when p is fixed and $n \to \infty$. In fact, how to establish the Bayes risk consistency of the DWD and kernel DWD was proposed as an open research problem in the original DWD paper (Marron et al., 2007). Nearly a decade later, the problem still remains open.

In this paper, we aim to resolve the aforementioned issues. We show that the kernel DWD in a RKHS has the Bayes risk consistency property if a universal kernel is used. This result should convince those who are less familiar with DWD to treat the kernel DWD as a serious competitor to the kernel SVM. To popularize the DWD, it is also important to allow practitioners to easily try DWD collectively with the SVM in real applications. To this end, we develop a novel fast algorithm to solve the linear and kernel DWD by using the majorization-minimization (MM) principle. Compared with the SOCP algorithm, our new algorithm has multiple advantages. First, our algorithm is much faster than the SOCP algorithm. In some examples, our algorithm can be several hundred times faster. Second, DWD equipped with our algorithm can be faster than the SVM. Third, our algorithm is easier to understand than the SOCP algorithm, especially for those who are not familiar with semi-definite and second-order-cone programming. This could help demystify the DWD and hence may increase its popularity.

To give a quick demonstration, we use a simulation example to compare the kernel DWD and the kernel SVM. We drew 10 centers $\{\boldsymbol{\mu}_{k+}\}$ from $N((1,0)^T, \boldsymbol{I})$. For each data point in the positive class, we randomly picked up a center $\boldsymbol{\mu}_{k+}$ and then generated the point from $N(\boldsymbol{\mu}_{k+}, \boldsymbol{I}/5)$. The negative class was assembled in the same way except that 10 centers $\boldsymbol{\mu}_{k-}$

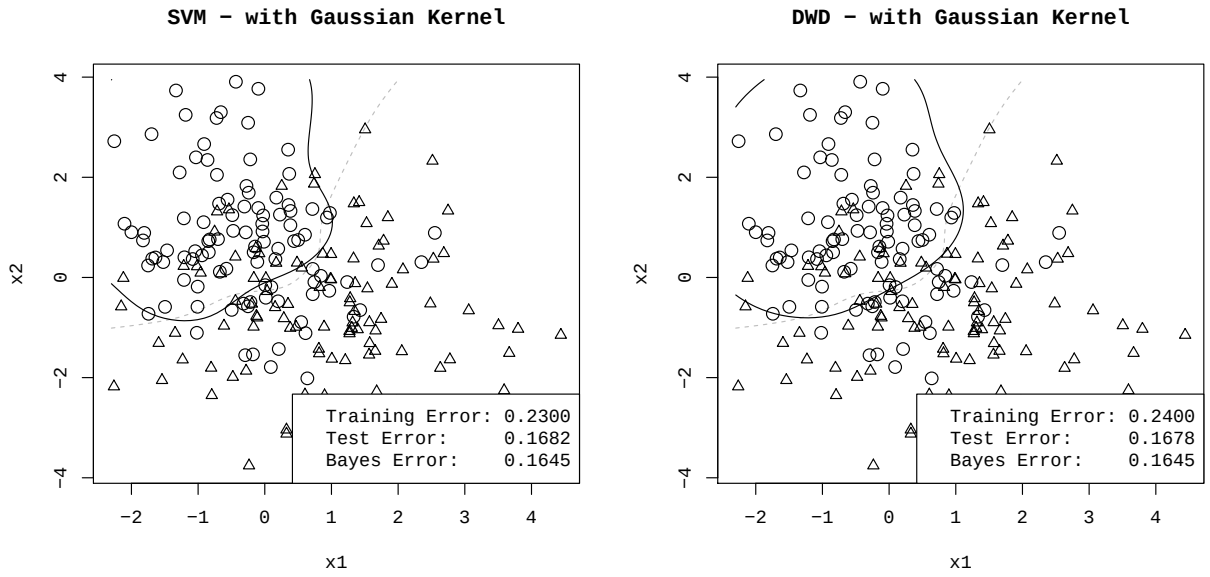

Figure 1. *Nonlinear SVM and DWD with Gaussian kernel. The broken curves are the Bayes decision boundary. The R package* **kerndwd** *used 2.396 second to solve the kernel DWD, and* **kernlab** *took 7.244 second to solve the kernel SVM. The timings include tuning parameters and they are averaged over 100 runs.*

were drawn from $N((0,1)^T, \boldsymbol{I})$. For this model the Bayes rule is nonlinear [1]. Figure 1 displays the training data from the simulation model where 100 observations are from the positive class (plotted as triangles) and another 100 observations are from the negative class (plotted as circles). We fitted the SVM and DWD using Gaussian kernels. We have implemented our new algorithm for DWD in a publicly available R package `kerndwd`. We computed the kernel SVM by using the R package `kernlab` (Karatzoglou et al., 2004). We recorded their training errors and test errors. From Figure 1, we observe that like the kernel SVM, the kernel DWD has a test error close to the Bayes error, which is consistent with the Bayes risk consistency property of the kernel DWD established in section 4.2. Notably, the kernel DWD is about three times as fast as the kernel SVM in this example.

The rest of the paper is organized as follows. To be self-contained, we first review the SVM and DWD in section 2. We then derive the novel algorithm for DWD in section 3. We introduce the kernel DWD in a reproducing kernel Hilbert space and establish the learning theory of kernel DWD in section 4. Real data examples are given in section 5 to compare DWD and the SVM. Technical proofs are provided in the appendix.

[1] The Bayes decision boundary is a curve: $\left\{\boldsymbol{z} : \sum_k \exp\left(-5||\boldsymbol{z}-\boldsymbol{\mu}_{k+}||^2/2\right) = \sum_k \exp\left(-5||\boldsymbol{z}-\boldsymbol{\mu}_{k-}||^2/2\right)\right\}$.

2 Review of SVMs and DWD

2.1 SVM

The introduction of the SVM usually begins with its geometric interpretation as a maximum margin classifier (Vapnik, 1995). Consider a case when two classes are separable by a hyperplane $\{\boldsymbol{x} : f(x) = \omega_0 + \boldsymbol{x}^T\boldsymbol{\omega} = 0\}$ such that $y_i(\omega_0 + \boldsymbol{x}_i^T\boldsymbol{\omega})$ are all non-negative. Without loss of generality, we assume that $\boldsymbol{\omega}$ is a unit vector, i.e., $\boldsymbol{\omega}^T\boldsymbol{\omega} = 1$, and we observe that each $d_i \equiv y_i(\omega_0 + \boldsymbol{x}_i^T\boldsymbol{\omega})$ is equivalent to the Euclidean distance between the data point $\boldsymbol{x}_i$ and the hyperplane. The reason is that $d_i = (\boldsymbol{x}_i - \boldsymbol{x}_0)^T\boldsymbol{\omega}$ and $\omega_0 + \boldsymbol{x}_0^T\boldsymbol{\omega} = 0$, where $\boldsymbol{x}_0$ is any data point on the hyperplane and $\boldsymbol{\omega}$ is the unit normal vector. The SVM classifier is defined as the optimal separating hyperplane that maximizes the smallest distance of each data point to the separating hyperplane. Mathematically, the SVM can be written as the following optimization problem (for the separable data case):

$$\begin{aligned} \max_{\omega_0,\boldsymbol{\omega}} \quad & \min d_i, \\ \text{subject to} \quad & d_i = y_i(\omega_0 + \boldsymbol{x}_i^T\boldsymbol{\omega}) \geq 0, \ \forall i, \ \text{and } \boldsymbol{\omega}^T\boldsymbol{\omega} = 1. \end{aligned} \tag{2.1}$$

The smallest distance $\min d_i$ is called the *margin*, and the SVM is thereby regarded as a *large-margin classifier*. The data points closest to the hyperplane, i.e., $d_i = \min d_i$, are dubbed the *support vectors*.

In general, the two classes are not separable, and thus $y_i(\omega_0 + \boldsymbol{x}_i^T\boldsymbol{\omega})$ cannot be non-negative for all $i = 1, \ldots, n$. To handle this issue, non-negative slack variables η_i, $1 \leq i \leq n$, are introduced to ensure all $y_i(\omega_0 + \boldsymbol{x}_i^T\boldsymbol{\omega}) + \eta_i$ to be non-negative. With these slack variables, the optimization problem (2.1) is generalized as follows,

$$\begin{aligned} \max_{\omega_0,\boldsymbol{\omega}} \quad & \min d_i, \\ \text{subject to} \quad & d_i = y_i(\omega_0 + \boldsymbol{x}_i^T\boldsymbol{\omega}) + \eta_i \geq 0, \ \forall i, \\ & \eta_i \geq 0, \ \forall i, \ \sum_{i=1}^{n} \eta_i < \text{constant}, \ \text{and } \boldsymbol{\omega}^T\boldsymbol{\omega} = 1. \end{aligned} \tag{2.2}$$

To compute SVMs, the optimization problem (2.2) is usually rephrased as an equivalent

quadratic programming (QP) problem,

$$\begin{aligned} \min_{\beta_0,\boldsymbol{\beta}} \quad & \left[\frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + c\sum_{i=1}^{n}\xi_i\right], \\ \text{subject to} \quad & y_i(\beta_0 + \boldsymbol{x}_i^T\boldsymbol{\beta}) + \xi_i \geq 1,\ \xi_i \geq 0,\ \forall i, \end{aligned} \tag{2.3}$$

and it can be solved by maximizing its Lagrange dual function,

$$\begin{aligned} \max_{\mu_i} \quad & \left[\sum_{i=1}^{n}\mu_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{i'=1}^{n}\mu_i\mu_{i'}y_iy_{i'}\langle\boldsymbol{x}_i,\boldsymbol{x}_{i'}\rangle\right], \\ \text{subject to} \quad & \mu_i \geq 0 \text{ and } \sum_{i=1}^{n}\mu_i y_i = 0. \end{aligned} \tag{2.4}$$

By solving (2.4), one can show that the solution of (2.3) has the form

$$\hat{\boldsymbol{\beta}} = \sum_{i=1}^{n}\hat{\mu}_i y_i \boldsymbol{x}_i, \text{ and thus } \hat{f}(\boldsymbol{x}) = \hat{\beta}_0 + \sum_{i=1}^{n}\hat{\mu}_i y_i\langle\boldsymbol{x},\boldsymbol{x}_i\rangle, \tag{2.5}$$

$\hat{\mu}_i$ being zero only when $\boldsymbol{x}_i$ lies on the support vectors.

One widely used method to extend the linear SVM to non-linear classifiers is the kernel method (Aizerman et al., 1964), which replaces the dot product $\langle\boldsymbol{x}_i,\boldsymbol{x}_{i'}\rangle$ in the Lagrange dual problem (2.4) with a kernel function $K(\boldsymbol{x}_i,\boldsymbol{x}_i')$, and hence the solution has the form

$$\hat{f}(\boldsymbol{x}) = \hat{\beta}_0 + \boldsymbol{x}^T\hat{\boldsymbol{\beta}} = \hat{\beta}_0 + \sum_{i=1}^{n}\hat{\mu}_i y_i K(\boldsymbol{x},\boldsymbol{x}_i).$$

Some popular examples of the kernel function K include: $K(\boldsymbol{x},\boldsymbol{x}') = \langle\boldsymbol{x},\boldsymbol{x}'\rangle$ (linear kernel), $K(\boldsymbol{x},\boldsymbol{x}') = (a + \langle\boldsymbol{x},\boldsymbol{x}'\rangle)^d$ (polynomial kernel), and $K(\boldsymbol{x},\boldsymbol{x}') = \exp(-\sigma||\boldsymbol{x}-\boldsymbol{x}'||_2^2)$ (Gaussian kernel), among others.

2.2 DWD

2.2.1 Motivation

Distance weighted discrimination was originally proposed by Marron et al. (2007) to resolve the *data-piling* issue. Marron et al. (2007) observed that many data points become support vectors when the SVM is applied on the so-called high-dimension-low-sample-size (HDLSS)

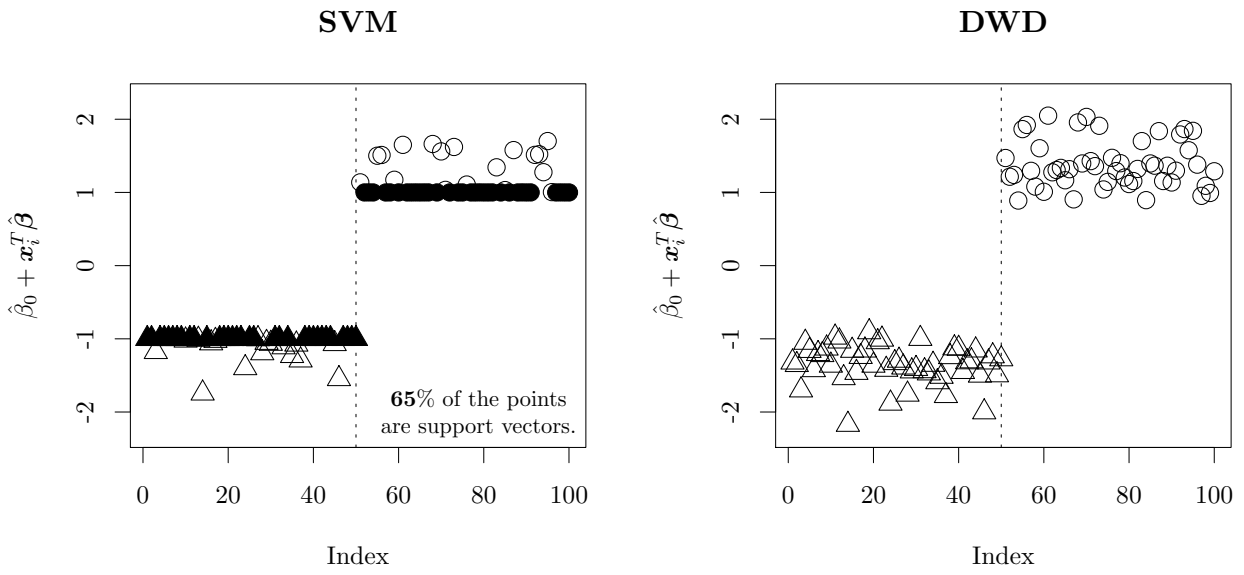

Figure 2. *A toy example illustrating the data-piling. Values $\hat{\beta}_0 + \boldsymbol{x}_i^T\hat{\boldsymbol{\beta}}$ are plotted for SVM and DWD. Indices 1 to 50 represent negative class (triangles) and indices 51 to 100 are for positive class (circles). In the left panel, data points belonging to the support vectors are depicted as solid circles and triangles.*

data, and Marron et al. (2007) coined the term data-piling to describe this phenomenon. We delineate it in Figure 2 through a simulation example. Let $\boldsymbol{\mu} = (3, 0, \ldots, 0)$ be a 200-dimension vector. We generated 50 points (indexed from 1 to 50 and represented as triangles) from $N(-\boldsymbol{\mu}, \boldsymbol{I}_p)$ as the negative class and another 50 points (indexed from 51 to 100 and represented as circles) from $N(\boldsymbol{\mu}, \boldsymbol{I}_p)$ as the positive class. We computed $\hat{\beta}_0$ and $\hat{\boldsymbol{\beta}}$ for SVM (2.3). In the left panel of Figure 2, we plotted $\hat{\beta}_0 + \boldsymbol{x}_i^T\hat{\boldsymbol{\beta}}$ for each data point, and we portrayed the support vectors by solid triangles and circles. We observe that 65 out of 100 data points become support vectors. The right panel of Figure 2 corresponds to DWD (will be defined shortly), where data-piling is attenuated. A real example revealing the data-piling can be seen in Figure 1 of Ahn and Marron (2010).

Marron et al. (2007) viewed "data-piling" as a drawback of the SVM, because the SVM classifier (2.5) is a function of only support vectors. Another popular classifier logistic regression does classification by using all the data points. However, the classical logistic regression classifier is derived by following the maximum likelihood principle, not based on a nice margin-maximization motivation[2]. Marron et al. (2007) wanted to have a new method

[2]Zhu and Hasite (2005) later showed that the limiting ℓ_2 penalized logistic regression approaches the margin-maximizing hyperplane for the separable data case. DWD was first proposed in 2002.

that is directly formulated by a SVM-like margin-maximization picture and also uses all data points for classification. To this end, Marron et al. (2007) proposed DWD which finds a separating hyperplane minimizing the total inverse margins of all the data points:

$$\begin{aligned} \min_{\omega_0,\boldsymbol{\omega}} \quad & \left[\sum_{i=1}^{n}\frac{1}{d_i}+c\sum_{i=1}^{n}\eta_i\right], \\ \text{subject to} \quad & d_i = y_i(\omega_0+\boldsymbol{x}_i^T\boldsymbol{\omega})+\eta_i \geq 0,\ \eta_i \geq 0,\ \forall i,\ \text{and } \boldsymbol{\omega}^T\boldsymbol{\omega}=1. \end{aligned} \tag{2.6}$$

There has been much work on variants of the standard DWD. We can only give an incomplete list here. Qiao et al. (2010) introduced the weighted DWD to tackle unequal cost or sample sizes by imposing different weights on two classes. Huang et al. (2013) extended the binary DWD to the multiclass case. Wang and Zou (2015) proposed the sparse DWD for high-dimensional classification. In addition, the work connecting DWD with other classifiers, e.g., SVM, includes but not limited to LUM (Liu et al., 2011), DWSVM (Qiao and Zhang , 2015a), and FLAME (Qiao and Zhang , 2015b). Marron (2015) provided a more comprehensive review of the current DWD literature.

2.2.2 Computation

Marron et al. (2007) solved the standard DWD by reformulating (2.6) as a second-order cone programming (SOCP) program (Alizadeh and Goldfarb, 2004; Boyd and Vandenberghe, 2004), which has a linear objective, linear constraints, and second-order-cone constraints. Specifically, for each i, let $\rho_i = (1/d_i + d_i)/2$, $\sigma_i = (1/d_i - d_i)/2$, and then $\rho_i + \sigma_i = 1/d_i$, $\rho_i - \sigma_i = d_i$, and $\rho_i^2 - \sigma_i^2 = 1$. Hence the original optimization problem (2.6) becomes

$$\begin{aligned} \min_{\omega_0,\boldsymbol{\omega}} \quad & \left[\mathbf{1}^T\boldsymbol{\rho}+\mathbf{1}^T\boldsymbol{\sigma}+c\mathbf{1}^T\boldsymbol{\eta}\right], \\ \text{subject to} \quad & \boldsymbol{\rho}-\boldsymbol{\sigma} = \tilde{\boldsymbol{Y}}\boldsymbol{X}\boldsymbol{\omega}+\omega_0\cdot\boldsymbol{y}+\boldsymbol{\eta}, \\ & \eta_i \geq 0,\ (\rho_i;\sigma_i,1)\in S_3,\ \forall i,\ (1;\boldsymbol{\omega})\in S_{p+1}, \end{aligned} \tag{2.7}$$

where $\tilde{\boldsymbol{Y}}$ is an $n\times n$ diagonal matrix with the ith diagonal element y_i, $\boldsymbol{X}$ is an $n\times p$ data matrix with the ith row $\boldsymbol{x}_i^T$, and $S_{m+1} = \{(\psi,\boldsymbol{\phi})\in\mathbb{R}^{m+1} : \psi^2 \geq \boldsymbol{\phi}^T\boldsymbol{\phi}\}$ is the form of the second-order cones. After solving $\hat{\omega}_0$ and $\hat{\boldsymbol{\omega}}$ from (2.7), a new observation $\boldsymbol{x}_{\text{new}}$ is classified by $\text{sign}(\hat{\omega}_0+\boldsymbol{x}_{\text{new}}^T\hat{\boldsymbol{\omega}})$.

2.2.3 Non-linear extension

Note that the kernel SVM was derived from applying the kernel trick to the dual formulation (2.5). Marron et al. (2007) followed the same approach to consider a version of kernel DWD for achieving non-linear classification. The dual function of the problem (2.7) is (Marron et al., 2007)

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \left[-\sqrt{\boldsymbol{\alpha}^T \tilde{\boldsymbol{Y}} \boldsymbol{X} \boldsymbol{X}^T \tilde{\boldsymbol{Y}} \boldsymbol{\alpha}} + 2 \cdot \mathbf{1}^T \sqrt{\boldsymbol{\alpha}} \right], \\ \text{subject to} \quad & \boldsymbol{y}^T \boldsymbol{\alpha} = 0, \ \mathbf{0} \leq \boldsymbol{\alpha} \leq c \cdot \mathbf{1}, \end{aligned} \tag{2.8}$$

where $(\sqrt{\boldsymbol{\alpha}})_i = \sqrt{\alpha_i}$, $i = 1, 2, \ldots, n$. Note that (2.8) only uses $\boldsymbol{X}\boldsymbol{X}^T$, which makes it easy to employ the kernel trick to get a nonlinear extension of the linear DWD. For a given kernel function K, define the kernel matrix as $(\boldsymbol{K})_{ij} = K(X_i, X_j)$, $1 \leq i, \ j \leq n$. Then a kernel DWD can be defined as (Marron et al., 2007)

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \left[-\sqrt{\boldsymbol{\alpha}^T \tilde{\boldsymbol{Y}} \boldsymbol{K} \tilde{\boldsymbol{Y}} \boldsymbol{\alpha}} + 2 \cdot \mathbf{1}^T \sqrt{\boldsymbol{\alpha}} \right], \\ \text{subject to} \quad & \boldsymbol{y}^T \boldsymbol{\alpha} = 0, \ \mathbf{0} \leq \boldsymbol{\alpha} \leq c \cdot \mathbf{1}. \end{aligned} \tag{2.9}$$

To solve (2.9), Marron et al. (2007) used the Cholesky decomposition of the kernel matrix, i.e., $\boldsymbol{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T$ and then replaced the predictors $\boldsymbol{X}$ in (2.7) with $\boldsymbol{\Phi}$. Marron et al. (2007) also carefully discussed several algorithmic issues that ensure the equivalent optimality in (2.7) and (2.8).

Remark 1. Two DWD implementations have been published thus far: a Matlab software (Marron, 2013) and an R package `DWD` (Huang et al., 2012). Both implementations are based on a Matlab SOCP solver `SDPT3`, which was developed by Tütüncü et al. (2003). We notice that the R package `DWD` can only compute the linear DWD.

Remark 2. To our best knowledge, the theoretical justification for the kernel DWD in Marron et al. (2007) is still unclear. The reason is likely due to the fact that the nonlinear extension is purely algorithmic. In fact, the Bayes risk consistency of DWD was proposed as an open research problem in Marron et al. (2007). The kernel DWD considered in this paper can be rigorously justified to have a universal Bayes risk consistency property; see details in section 4.2.

2.2.4 Generalized DWD

Marron et al. (2007) also attempted to replace the reciprocal in the DWD optimization problem (2.6) with the qth power ($q > 0$) of the inverse distances, and Hall et al. (2005) also used it as the original definition of DWD. We name the DWD with this new formulation the generalized DWD:

$$\begin{aligned} \min_{\omega_0,\boldsymbol{\omega}} \quad & \left[\sum_{i=1}^{n} \frac{1}{d_i^q} + c\sum_{i=1}^{n} \eta_i\right], \\ \text{subject to} \quad & d_i = y_i(\omega_0 + \boldsymbol{x}_i^T\boldsymbol{\omega}) + \eta_i \geq 0,\ \eta_i \geq 0,\ \forall i,\ \text{and } \boldsymbol{\omega}^T\boldsymbol{\omega} = 1, \end{aligned} \tag{2.10}$$

which degenerates to the standard DWD (2.6) when $q = 1$.

The first asymptotic theory for DWD and generalized DWD was given in Hall et al. (2005) who presented a novel geometric representation of the HDLSS data. Assuming $\boldsymbol{X}_1^+, \boldsymbol{X}_2^+, \ldots, \boldsymbol{X}_{n^+}^+$ are the data from the positive class and $\boldsymbol{X}_1^-, \boldsymbol{X}_2^-, \ldots, \boldsymbol{X}_{n^-}^-$ are from the negative class. Hall et al. (2005) stated that, when the sample size n is fixed and the dimension p goes to infinity, under some regularity conditions, there exist two constants l^+ and l^- such that for each pair of i and j,

$$p^{-1/2}||\boldsymbol{X}_i^+ - \boldsymbol{X}_j^+|| \xrightarrow{P} \sqrt{2}l^+, \text{ and } p^{-1/2}||\boldsymbol{X}_i^- - \boldsymbol{X}_j^-|| \xrightarrow{P} \sqrt{2}l^-,$$

as $p \to \infty$. This result was applied the results to study several classifiers including the SVM and the generalized DWD. For ease presentation let us consider the equal subgroup size case, i.e., $n_+ = n_- = n/2$. Hall et al. (2005) assumed that $p^{-1/2}||E\boldsymbol{X}^+ - E\boldsymbol{X}^-|| \to \mu$, as $p \to \infty$, The basic conclusion is that when μ is greater than a threshold that depends on l^+, l^-, n, the misclassification error converges to zero, and when μ is less than the same threshold, the misclassification error converges to 50%. For more details, see Theorem 1 and Theorem 2 in Hall et al. (2005). Ahn et al. (2007) further relaxed the assumptions thereof.

Remark 3. The generalized DWD has not been implemented yet because the SOCP transformation only works for the standard DWD ($q = 1$) (2.7), but its extension to handle the general cases is unclear if not impossible. That is why the current DWD literature only focuses on DWD with $q = 1$. In fact, the generalized DWD with $q \neq 1$ was proposed as an open research problem in Marron et al. (2007). The new algorithm proposed in this paper can easily solve the generalized DWD problem for any $q > 0$; see section 3.

3 A Novel Algorithm for DWD

Marron et al. (2007) originally solved the standard DWD by transforming (2.6) into a SOCP problem. This algorithm, however, cannot compute the generalized DWD (2.10) with $q \neq 1$. In this section, we propose an entirely different algorithm based on the majorization-minimization (MM) principle. Our new algorithm offers a unified solution to the standard DWD and the generalized DWD.

3.1 Generalized DWD loss

Our algorithm begins with a *loss + penalty* formulation of the DWD. Lemma 1 deploys the result. Note that the loss function also lays the foundation of the kernel DWD learning theory that will be discussed in section 4.

Lemma 1. *The generalized DWD classifier in* (2.10) *can be written as* $\text{sign}(\hat{\beta}_0 + \boldsymbol{x}_i^T\hat{\boldsymbol{\beta}})$*, where* $(\hat{\beta}_0, \hat{\boldsymbol{\beta}})$ *is computed from*

$$\min_{\beta_0,\boldsymbol{\beta}} \boldsymbol{C}(\beta_0, \boldsymbol{\beta}) \equiv \min_{\beta_0,\boldsymbol{\beta}} \left[\frac{1}{n} \sum_{i=1}^{n} V_q\left(y_i(\beta_0 + \boldsymbol{x}_i^T\boldsymbol{\beta})\right) + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta} \right], \tag{3.1}$$

for some λ*, where*

$$V_q(u) = \begin{cases} 1 - u, & \text{if } u \leq \dfrac{q}{q+1}, \\ \dfrac{1}{u^q}\dfrac{q^q}{(q+1)^{q+1}}, & \text{if } u > \dfrac{q}{q+1}. \end{cases} \tag{3.2}$$

Remark 4. The proof of Lemma 1 provides the one-to-one mapping between λ in (3.1) and c in (2.10). Write $(\hat{\beta}(\lambda)_0, \hat{\boldsymbol{\beta}}(\lambda))$ as the solution to (3.1). Define

$$c(\lambda) = \frac{(q+1)^{q+1}}{q^q} \|\hat{\boldsymbol{\beta}}(\lambda)\|^{q+1}.$$

Considering (2.10) using $c(\lambda)$,

$$\begin{aligned} (\hat{\omega_0}, \hat{\boldsymbol{\omega}}) = \underset{\omega_0,\boldsymbol{\omega}}{\text{argmin}} \quad & \left[\sum_{i=1}^{n} \frac{1}{d_i^q} + c(\lambda) \sum_{i=1}^{n} \eta_i \right], \\ \text{subject to} \quad & d_i = y_i(\omega_0 + \boldsymbol{x}_i^T\boldsymbol{\omega}) + \eta_i \geq 0, \ \eta_i \geq 0, \ \forall i, \text{ and } \boldsymbol{\omega}^T\boldsymbol{\omega} = 1, \end{aligned} \tag{3.3}$$

we have

$$\hat{\boldsymbol{\omega}} = \hat{\boldsymbol{\beta}}(\lambda)/\|\hat{\boldsymbol{\beta}}(\lambda)\| \text{ and } \hat{\omega}_0 = \hat{\beta}(\lambda)_0/\|\hat{\boldsymbol{\beta}}(\lambda)\|.$$

Note that $\text{sign}(\hat{\omega}_0 + \boldsymbol{x}_i^T\hat{\boldsymbol{\omega}}) = \text{sign}(\hat{\beta}(\lambda)_0 + \boldsymbol{x}_i^T\hat{\boldsymbol{\beta}}(\lambda))$, which means that the generalized DWD classifier defined by (3.3) is equivalent to the generalized DWD classifier defined by (3.1).

By Lemma 1, we call $V_q(\cdot)$ the generalized DWD loss. It can be visualized in Figure 3. We observe that the generalized DWD loss decreases as q increases and it approaches the SVM hinge loss function as $q \to \infty$. When $q = 1$, the generalized DWD loss becomes

$$V_1(u) = \begin{cases} 1 - u, & \text{if } u \leq 1/2, \\ 1/(4u), & \text{if } u > 1/2. \end{cases}$$

We notice that $V_1(u)$ has appeared in the literature (Qiao et al., 2010; Liu et al., 2011). In this work we give a unified treatment of all q values, not just $q = 1$.

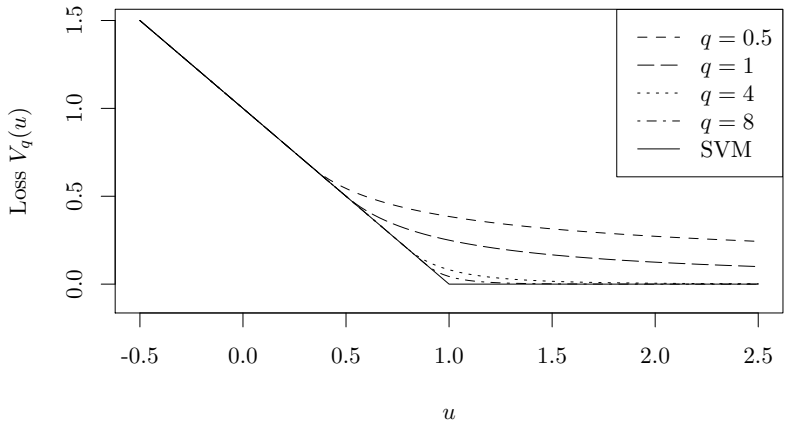

Figure 3. *Top to bottom are the DWD loss functions with $q = 0.5, 1, 4, 8$, and the SVM hinge loss.*

3.2 Derivation of the algorithm

We now show how to develop the new algorithm by using the MM principle (De Leeuw and Heiser, 1977; Lange et al., 2000; Hunter and Lange, 2004). Some recent successful applications of the MM principle can be seen in Hunter and Li (2005); Wu and Lange (2008); Zou and Li (2008); Zhou and Lange (2010); Yang and Zou (2013); Lange and Zhou

(2014), among others. The main idea of the MM principle is easy to understand. Suppose $\boldsymbol{\theta} = (\beta_0, \boldsymbol{\beta}^T)^T$ and we aim to minimize $\boldsymbol{C}(\boldsymbol{\theta})$, defined in (3.1). The MM principle finds a majorization function $\boldsymbol{D}(\boldsymbol{\theta}|\boldsymbol{\theta}_k)$ satisfying $\boldsymbol{C}(\boldsymbol{\theta}) < \boldsymbol{D}(\boldsymbol{\theta}|\boldsymbol{\theta}_k)$ for any $\boldsymbol{\theta} \neq \boldsymbol{\theta}_k$ and $\boldsymbol{C}(\boldsymbol{\theta}_k) = \boldsymbol{D}(\boldsymbol{\theta}_k|\boldsymbol{\theta}_k)$, and then we generate a sequence $\{\boldsymbol{C}(\boldsymbol{\theta}_k)\}_{k=1}^{\infty}$ by updating $\boldsymbol{\theta}_k$ via $\boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}_{k+1} = \operatorname{argmin}_{\boldsymbol{\theta}} \boldsymbol{D}(\boldsymbol{\theta}|\boldsymbol{\theta}_k)$.

We first expose some properties of the generalized DWD loss functions, which give rise to a quadratic majorization function of $\boldsymbol{C}(\boldsymbol{\theta})$. The generalized DWD loss is differentiable everywhere; its first-order derivative is given below,

$$V_q'(u) = \begin{cases} -1, & \text{if } u \leq \dfrac{q}{q+1}, \\ -\dfrac{1}{u^{q+1}}\left(\dfrac{q}{q+1}\right)^{q+1}, & \text{if } u > \dfrac{q}{q+1}. \end{cases} \tag{3.4}$$

Lemma 2. *The generalized DWD loss function $V_q(\cdot)$ has a Lipschitz continuous gradient,*

$$|V_q'(t) - V_q'(\tilde{t})| < M|t - \tilde{t}|, \tag{3.5}$$

which further implies a quadratic majorization function of $V_q(\cdot)$ such that

$$V_q(t) < V_q(\tilde{t}) + V_q'(\tilde{t})(t - \tilde{t}) + \frac{M}{2}(t - \tilde{t})^2 \tag{3.6}$$

for any $t \neq \tilde{t}$ and $M = (q+1)^2/q$.

Denote the current solution by $\tilde{\boldsymbol{\theta}} = (\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}^T)^T$ and the updated solution by $\boldsymbol{\theta} = (\beta_0, \boldsymbol{\beta}^T)^T$. We settle $\boldsymbol{C}(\boldsymbol{\theta}) = \boldsymbol{C}(\beta_0, \boldsymbol{\beta})$ and $\boldsymbol{D}(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}}) = \boldsymbol{D}(\beta_0, \boldsymbol{\beta})$ without abusing notations. We have that for any $(\beta_0, \boldsymbol{\beta}) \neq (\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$,

$$\begin{aligned} &\boldsymbol{C}(\beta_0, \boldsymbol{\beta}) \\ \equiv &\frac{1}{n}\sum_{i=1}^{n} V_q\left(y_i(\beta_0 + \boldsymbol{x}_i^T\boldsymbol{\beta})\right) + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta} \\ < &\frac{1}{n}\sum_{i=1}^{n} V_q\left(y_i(\tilde{\beta}_0 + \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}})\right) + \frac{1}{n}\sum_{i=1}^{n} V_q'\left(y_i(\tilde{\beta}_0 + \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}})\right)\left[y_i(\beta_0 - \tilde{\beta}_0) + y_i\boldsymbol{x}_i^T(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})\right] \\ &+ \frac{M}{2n}\sum_{i=1}^{n}\left[y_i(\beta_0 - \tilde{\beta}_0) + y_i\boldsymbol{x}_i^T(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})\right]^2 + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta} \\ \equiv &\boldsymbol{D}(\beta_0, \boldsymbol{\beta}). \end{aligned} \tag{3.7}$$

We now find the minimizer of $\boldsymbol{D}(\beta_0, \boldsymbol{\beta})$. The gradients of $\boldsymbol{D}(\beta_0, \boldsymbol{\beta})$ are given as follows:

$$\begin{aligned}
\partial \frac{\boldsymbol{D}(\beta_0, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} =& \frac{1}{n} \sum_{i=1}^{n} V_q' \left(y_i (\tilde{\beta}_0 + \boldsymbol{x}_i^T \tilde{\boldsymbol{\beta}}) \right) y_i \boldsymbol{x}_i + \frac{M}{n} \sum_{i=1}^{n} \left[(\beta_0 - \tilde{\beta}_0) + \boldsymbol{x}_i^T (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}) \right] \boldsymbol{x}_i + 2\lambda \boldsymbol{\beta} \\
=& \boldsymbol{X}^T \boldsymbol{z} + \frac{M}{n} (\beta_0 - \tilde{\beta}_0) \boldsymbol{X}^T \mathbf{1} + \frac{M}{n} \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^T (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}) + 2\lambda \boldsymbol{\beta} \\
=& \boldsymbol{X}^T \boldsymbol{z} + \frac{M}{n} (\beta_0 - \tilde{\beta}_0) \boldsymbol{X}^T \mathbf{1} + \left(\frac{M}{n} \boldsymbol{X}^T \boldsymbol{X} + 2\lambda \boldsymbol{I}_p \right) (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}) + 2\lambda \tilde{\boldsymbol{\beta}}, \qquad (3.8) \\
\partial \frac{\boldsymbol{D}(\beta_0, \boldsymbol{\beta})}{\partial \beta_0} =& \frac{1}{n} \sum_{i=1}^{n} V_q' \left(y_i (\tilde{\beta}_0 + \boldsymbol{x}_i^T \tilde{\boldsymbol{\beta}}) \right) y_i + \frac{M}{n} \sum_{i=1}^{n} \left[(\beta_0 - \tilde{\beta}_0) + \boldsymbol{x}_i^T (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}) \right] \\
=& \mathbf{1}^T \boldsymbol{z} + M(\beta_0 - \tilde{\beta}_0) + \frac{M}{n} \mathbf{1}^T \boldsymbol{X} (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}). \qquad (3.9)
\end{aligned}$$

where $\boldsymbol{X}$ is the $n \times p$ data matrix with the ith row $\boldsymbol{x}_i^T$, $\boldsymbol{z}$ is an $n \times 1$ vector with the ith element $y_i V_q'(y_i(\tilde{\beta}_0 + \boldsymbol{x}_i^T \tilde{\boldsymbol{\beta}}))/n$, and $\mathbf{1} \in \mathbb{R}^n$ is the vector of ones. Setting $[\partial \boldsymbol{D}(\beta_0, \boldsymbol{\beta})/\partial \beta_0, \partial \boldsymbol{D}(\beta_0, \boldsymbol{\beta})/\partial \boldsymbol{\beta}]$ to be zeros, we obtain the minimizer of $\boldsymbol{D}(\beta_0, \boldsymbol{\beta})$:

$$\begin{pmatrix} \beta_0 \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \tilde{\beta}_0 \\ \tilde{\boldsymbol{\beta}} \end{pmatrix} - \frac{n}{M} \begin{pmatrix} n & \mathbf{1}^T \boldsymbol{X} \\ \boldsymbol{X}^T \mathbf{1} & \boldsymbol{X}^T \boldsymbol{X} + \frac{2n\lambda}{M} \boldsymbol{I}_p \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{1}^T \boldsymbol{z} \\ \boldsymbol{X}^T \boldsymbol{z} + 2\lambda \tilde{\boldsymbol{\beta}} \end{pmatrix}. \qquad (3.10)$$

So far we have completed all the steps of the MM algorithm. Details are summarized in Algorithm 1.

We have implemented Algorithm 1 in an R package `kerndwd`, which is publicly available for download on CRAN.

3.3 Performance of the new algorithm

In this section, we show the superior computation performance of our R implementation, `kerndwd`, over the two existing implementations, the R package `DWD` (Huang et al., 2012) and the Matlab software (Marron, 2013). To avoid confusion, we henceforth use `OURS`, `HUANG`, and `MARRON` to denote `kerndwd`, `DWD`, and the Matlab implementation, respectively. Since `HUANG` is incapable of non-linear kernels and the generalized DWD with $q \neq 1$, we only attend to the linear DWD with q fixed to be one. All experiments were conducted on an Intel Core i5 M560 (2.67 GHz) processor.

For a fair comparison, we study the four numerical examples used in Marron et al. (2007), except for different sample sizes and dimensions. In each example, we generate a data set

Algorithm 1 Linear generalized DWD

Initialize $(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}^T)$
for each λ **do**
  Compute $\boldsymbol{P}^{-1}(\lambda)$:

$$\boldsymbol{P}^{-1}(\lambda) = \begin{pmatrix} n & \mathbf{1}^T\boldsymbol{X} \\ \boldsymbol{X}^T\mathbf{1} & \boldsymbol{X}^T\boldsymbol{X} + \frac{2n\lambda}{M}\boldsymbol{I}_p \end{pmatrix}^{-1}$$

  repeat
    Compute $\boldsymbol{z} = (z_1, \ldots, z_n)^T$: $z_i = y_i V_q'(y_i(\tilde{\beta}_0 + \boldsymbol{x}_i\tilde{\boldsymbol{\beta}}))/n$
    Compute:

$$\begin{pmatrix} \beta_0 \\ \boldsymbol{\beta} \end{pmatrix} \leftarrow \begin{pmatrix} \tilde{\beta}_0 \\ \tilde{\boldsymbol{\beta}} \end{pmatrix} - \frac{nq}{(q+1)^2}\boldsymbol{P}^{-1}(\lambda)\begin{pmatrix} \mathbf{1}^T\boldsymbol{z} \\ \boldsymbol{X}^T\boldsymbol{z} + 2\lambda\tilde{\boldsymbol{\beta}} \end{pmatrix}$$

    Set $(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}^T) = (\beta_0, \boldsymbol{\beta}^T)$
  until the convergence condition is met
end for

with sample size $n = 500$ and dimension $p = 50$. The responses are always binary; one half of the data have responses $+1$ and the other half have -1. Data in example 1 are generated from Gaussian distribution with means of $(\pm 2.2, 0, \ldots, 0)$ and an identity covariance for ± 1 classes respectively. Example 2 has 80% of data drawn as example 1 whereas the other 20% from Gaussian distributions with means of $(\pm 100, \pm 500, 0, \ldots, 0)$ for ± 1 classes. In example 3, 80% of the data are obtained as example 1 as well, while the means of the remaining 20% have the first coordinate replaced by ± 0.1 and one randomly chosen coordinate replaced by ± 100 for ± 1 classes. For example 4, at the first 25 coordinates, the data from -1 class are standard Gaussian and the data from $+1$ class are 11.09 times standard Gaussian; for both classes, the last 25 coordinates are just the squares of the first 25.

In each example, we fitted a linear DWD with five different tuning parameter values $\lambda = (0.01, 0.1, 1, 10, 100)$. After obtaining $(\hat{\beta}_0, \hat{\boldsymbol{\beta}})$, we computed $(\hat{\omega}_0, \hat{\boldsymbol{\omega}})$ and the constant c in (2.7) by using Remark 4. We then used `HUANG` and `MARRON` to compute their solutions. Note that in theory all three implementations should yield identical $(\hat{\omega}_0, \hat{\boldsymbol{\omega}})$. From table 1 we observe that `OURS` took remarkably less computation time than `HUANG` and `MARRON`. In example 1, for instance, `OURS` spent only 0.012 second on average to fit a DWD model, while `HUANG` used 14.525 seconds, and `MARRON` took 2.204 seconds, which were 1210 and 183 times

larger, respectively. In all four examples, the timings of OURS were 700 times above faster than the existing R implementation HUANG, and also more than 70 times faster than the Matlab implementation MARRON[3].

Table 1. *Timing comparisons among the R package* ***kerndwd*** *(denoted as* ***OURS****), the R package* ***DWD*** *(denoted as* ***HUANG****), and the Matlab implementation (denoted as* ***MARRON****). All the timings are averaged over 100 independent replicates.*

	Timing (in sec.)			Ratio	
	OURS	HUANG	MARRON	$\frac{t(\text{HUANG})}{t(\text{OURS})}$	$\frac{t(\text{MARRON})}{t(\text{OURS})}$
1	0.012	14.525	2.204	1210.8	183.7
2	0.024	18.018	2.411	750.8	100.5
3	0.028	26.918	2.076	961.4	74.1
4	0.020	21.536	2.264	1076.8	113.2

4 Kernel DWD in RKHS and Bayes Risk Consistency

4.1 Kernel DWD in RKHS

The kernel SVM can be derived by using the kernel trick or using the view of non-parametric function estimation in a reproducing kernel Hilbert space (RKHS). Much of the theoretical work on the kernel SVM is based on the RKHS formulation of SVMs. The derivation of the kernel SVM in a RKHS is given in Hastie et al. (2009). We take a similar approach to derive the kernel DWD, as our goal is to establish the kernel learning theory for DWD.

Consider $\mathcal{H}_K$, a reproducing kernel Hilbert space generated by the kernel function K. The Mercer's theorem ensures K to have an eigen-expansion $K(\boldsymbol{x}, \boldsymbol{x}') = \sum_{t=1}^{\infty} \gamma_t \phi_t(\boldsymbol{x}) \phi_t^T(\boldsymbol{x}')$, with $\gamma_t \geq 0$ and $\sum_{t=1}^{\infty} \gamma_t^2 < \infty$. Then the Hilbert space $\mathcal{H}_K$ is defined as the collection of functions $h(\boldsymbol{x}) = \sum_{t=1}^{\infty} \theta_t \phi_t(\boldsymbol{x})$, for any θ_t such that $\sum_{t=1}^{\infty} \theta_t^2/\gamma_t < \infty$, and the inner product is $\langle \sum_{t=1}^{\infty} \theta_t \phi_t(\boldsymbol{x}), \sum_{t'=1}^{\infty} \delta_{t'} \phi_{t'}(\boldsymbol{x}) \rangle_{\mathcal{H}_K} = \sum_{t=1}^{\infty} \theta_t \delta_t / \gamma_t$.

Given $\mathcal{H}_K$, let the non-linear DWD be written as $\text{sign}(\hat{\beta}_0 + \hat{h}(\boldsymbol{x}))$ where $(\hat{\beta}_0, \hat{h})$ is the

[3]We also checked the quality of the computed solutions by these different algorithms. In theory they should be identical. In practice, due to machine errors and implementations, they could be different. We found that in all examples our new algorithm gave better solutions in the sense that the objective function in (2.7) has the smallest value. HUANG and MARRON gave similar but slightly larger objective function values.

solution of

$$\min_{\substack{h\in\mathcal{H}_K\\ \beta_0\in\mathbb{R}}} \left[\frac{1}{n}\sum_{i=1}^{n} V_q\left(y_i(\beta_0+h(\boldsymbol{x}_i))\right)+\lambda||h||^2_{\mathcal{H}_K}\right], \tag{4.1}$$

where $V_q(\cdot)$ is the generalized DWD loss (3.2). The representer theorem concludes that the solution of (4.1) has a finite expansion based on $K(\boldsymbol{x},\boldsymbol{x}_i)$ (Wahba, 1990),

$$\hat{h}(\boldsymbol{x})=\sum_{i=1}^{n}\hat{\alpha}_i K(\boldsymbol{x},\boldsymbol{x}_i),$$

and thus

$$||\hat{h}||^2_{\mathcal{H}_K}=\sum_{i=1}^{n}\sum_{j=1}^{n}\hat{\alpha}_i\hat{\alpha}_j K(\boldsymbol{x}_i,\boldsymbol{x}_j).$$

Consequently, (4.1) can be paraphrased with matrix notation,

$$\min_{\beta_0,\boldsymbol{\alpha}} \boldsymbol{C}_K(\beta_0,\boldsymbol{\alpha})\equiv\min_{\beta_0,\boldsymbol{\alpha}}\left[\frac{1}{n}\sum_{i=1}^{n} V_q\left(y_i(\beta_0+\boldsymbol{K}_i^T\boldsymbol{\alpha})\right)+\lambda\boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{\alpha}\right], \tag{4.2}$$

where $\boldsymbol{K}$ is the kernel matrix with the (i,j)th element of $K(\boldsymbol{x}_i,\boldsymbol{x}_j)$ and $\boldsymbol{K}_i$ is the ith column of $\boldsymbol{K}$.

Remark 5. We can compare (4.2) to the kernel SVM (Hastie et al., 2009)

$$\min_{\beta_0,\boldsymbol{\alpha}}\left[\frac{1}{n}\sum_{i=1}^{n}\left[1-y_i(\beta_0+\boldsymbol{K}_i^T\boldsymbol{\alpha})\right]_+ +\lambda\boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{\alpha}\right], \tag{4.3}$$

where $[1-t]_+$ is the hinge loss underlying the SVM. As shown in Figure 3, the generalized DWD loss takes the hinge loss as its limit when $q\to\infty$. In general, the generalized DWD loss and the hinge loss look very similar, which suggests that the kernel DWD and the kernel SVM equipped with the same kernel have similar statistical behavior.

The procedure for deriving Algorithm 1 for the linear DWD can be directly adopted to derive an efficient algorithm for solving the kernel DWD. We obtain the majorization function $\boldsymbol{D}_K(\beta_0,\boldsymbol{\alpha})$,

$$\boldsymbol{D}_K(\beta_0,\boldsymbol{\alpha}) \;=\; \frac{1}{n}\sum_{i=1}^{n} V_q'\left(y_i(\tilde{\beta}_0+\boldsymbol{K}_i^T\tilde{\boldsymbol{\alpha}})\right)\left[y_i(\beta_0-\tilde{\beta}_0)+y_i\boldsymbol{K}_i^T(\boldsymbol{\alpha}-\tilde{\boldsymbol{\alpha}})\right]+\lambda\boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{\alpha}$$

Algorithm 2 Kernel DWD

1: Initialize $(\tilde{\beta}_0, \tilde{\boldsymbol{\alpha}}^T)$
2: **for** each λ **do**
3: Compute $\boldsymbol{P}^{-1}(\lambda)$:

$$\boldsymbol{P}^{-1}(\lambda) = \begin{pmatrix} n & \mathbf{1}^T\boldsymbol{K} \\ \boldsymbol{K}\mathbf{1} & \boldsymbol{K}\boldsymbol{K} + \frac{2nq\lambda}{(q+1)^2}\boldsymbol{K} \end{pmatrix}^{-1}$$

4: **repeat**
5: Compute $\boldsymbol{z} = (z_1, \ldots, z_n)^T$: $z_i = y_i V_q'(y_i(\tilde{\beta}_0 + \boldsymbol{K}_i\tilde{\boldsymbol{\alpha}}))/n$
6: Compute:

$$\begin{pmatrix} \beta_0 \\ \boldsymbol{\alpha} \end{pmatrix} \leftarrow \begin{pmatrix} \tilde{\beta}_0 \\ \tilde{\boldsymbol{\alpha}} \end{pmatrix} - \frac{nq}{(q+1)^2}\boldsymbol{P}^{-1}(\lambda)\begin{pmatrix} \mathbf{1}^T\boldsymbol{z} \\ \boldsymbol{K}\boldsymbol{z} + 2\lambda\boldsymbol{K}\tilde{\boldsymbol{\alpha}} \end{pmatrix}$$

7: Set $(\tilde{\beta}_0, \tilde{\boldsymbol{\alpha}}^T) = (\beta_0, \boldsymbol{\alpha}^T)$
8: **until** the convergence condition is met
9: **end for**

$$+\frac{M}{2n}\sum_{i=1}^{n}\left[y_i(\beta_0 - \tilde{\beta}_0) + y_i\boldsymbol{K}_i^T(\boldsymbol{\alpha} - \tilde{\boldsymbol{\alpha}})\right]^2 + \frac{1}{n}\sum_{i=1}^{n} V_q\left(y_i(\tilde{\beta}_0 + \boldsymbol{K}_i^T\tilde{\boldsymbol{\alpha}})\right)$$

and then find the minimizer of $\boldsymbol{D}_K(\beta_0, \boldsymbol{\alpha})$ which has a closed-form expression. We opt to omit the details here for space consideration. Algorithm 2 summarizes the entire algorithm for the kernel DWD.

4.2 Kernel learning theory

Lin (2002) formulated the kernel SVM as a non-parametric function estimation problem in a reproducing kernel Hilbert space and showed that the population minimizer of the SVM loss function is the Bayes rule, indicating that the SVM directly approximates the optimal Bayes classifier. Lin (2004) further coined a name "Fisher consistency" to describe such a result. The Vapnik-Chervonenkis (VC) analysis (Vapnik, 1998; Anthony and Bartlett, 1999) and the margin analysis (Bartlett and Shawe-Taylor, 1999; Shawe-Taylor and Cristianini, 2000) have been used to bound the expected classification error of the SVM. Zhang (2004) used the so-called leave-one-out analysis (Jaakkola and Haussler, 1999) to study a class of kernel machines. The exisiting theoretical work on the kernel SVM provides us a nice road map to study the kernel DWD. In this section we first elucidate the Fisher consistency (Lin,

2004) of the generalized kernel DWD, and we then establish the Bayes risk consistency of the kernel DWD when a universal kernel is employed.

Let $\eta(\boldsymbol{x})$ denote the conditional probability $P(Y = 1|\boldsymbol{X} = \boldsymbol{x})$. Under the 0-1 loss, the theoretical optimal Bayes rule is $f^\star(\boldsymbol{x}) = \text{sign}(\eta(\boldsymbol{x}) - 1/2)$. Assume $\eta(\boldsymbol{x})$ is a measurable function and $P(\eta(\boldsymbol{x}) = 1/2) = 0$ throughout.

Lemma 3. *The population minimizer of the expected generalized DWD loss* $E_{\boldsymbol{X}Y}[V_q(Yf(\boldsymbol{X}))]$ *is*

$$\tilde{f}(\boldsymbol{x}) = \frac{q}{q+1}\left[\left(\frac{\eta(\boldsymbol{x})}{1-\eta(\boldsymbol{x})}\right)^{\frac{1}{q+1}} \cdot I(\eta(\boldsymbol{x}) > 1/2) - \left(\frac{1-\eta(\boldsymbol{x})}{\eta(\boldsymbol{x})}\right)^{\frac{1}{q+1}} \cdot I(\eta(\boldsymbol{x}) < 1/2)\right], \quad (4.4)$$

where $I(\cdot)$ *is the indicator function. The population minimizer* $\tilde{f}(\boldsymbol{x})$ *has the same sign as* $\eta(\boldsymbol{x}) - 1/2$.

Fisher consistency is a property of the loss function. The interpretation is that the generalized DWD can approach Bayes rule with infinite many samples. We notice that Fisher consistency of $V_1(u)$ has been shown before (Qiao et al., 2010; Liu et al., 2011). In reality all classifiers are estimated from a finite sample. Thus, a more refined analysis of the actual DWD classifier is needed, and that is what we achieve in the following.

Following the convention in the literature, we absorb the intercept into h and present the kernel DWD as follows:

$$\hat{f}_n = \underset{f \in \mathcal{H}_K}{\text{argmin}} \left[\frac{1}{n}\sum_{i=1}^{n} V_q\left(y_i(f(\boldsymbol{x}_i)\right) + \lambda_n ||f||^2_{\mathcal{H}_K}\right]. \quad (4.5)$$

The ultimate goal is to show that the misclassification error of the kernel DWD approaches the Bayes error rate such that we can say the kernel DWD classifier works as well as the Bayes rule (asymptotically speaking). Following Zhang (2004), we derive the following lemma.

Lemma 4. *For a discrimination function* f, *we define* $R(f) = E_{\boldsymbol{X}Y}\left[Y \neq \text{sign}\left(f(\boldsymbol{X})\right)\right]$. *Assume that* $f^\star = \text{argmin}_f R(f)$ *is the Bayes rule and* $\hat{f}_n$ *is the solution of* (4.5), *then*

$$R(\hat{f}_n) - R(f^\star) \leq \frac{q+1}{q}(\varepsilon_A + \varepsilon_E), \quad (4.6)$$

where ε_A and ε_E are defined as follows and V_q is the generalized DWD loss,

$$\begin{aligned} \varepsilon_A &= \inf_{f \in \mathcal{H}_K} E_{\boldsymbol{X}Y}\Big[V_q(Yf(\boldsymbol{X}))\Big] - E_{\boldsymbol{X}Y}\Big[V_q\Big(Y\tilde{f}(\boldsymbol{X})\Big)\Big], \\ \varepsilon_E = \varepsilon_E(\hat{f}_n) &= E_{\boldsymbol{X}Y}\Big[V_q\Big(Y\hat{f}_n(\boldsymbol{X})\Big)\Big] - \inf_{f \in \mathcal{H}_K} E_{\boldsymbol{X}Y}\Big[V_q(Yf(\boldsymbol{X}))\Big]. \end{aligned} \tag{4.7}$$

In the above lemma $R(f^*)$ is the Bayes error rate and $R(\hat{f}_n)$ is the misclassification error of the kernel DWD applied to new data points. If $R(\hat{f}_n) \to R(f^\star)$, we say the classifier is Bayes risk consistent. Based on Lemma 4, it suffices to show that both ε_A and ε_E approach zero in order to demonstrate the Bayes risk consistency of the kernel DWD. Note that ε_A is deterministic and is called the approximation error. If the RKHS is rich enough then the approximation error can be made arbitrarily small. In the literature, the notation of *universal kernel* (Steinwart, 2001; Micchelli et al., 2006) has been proposed and studied. Suppose $\mathcal{X} \in \mathbb{R}^p$ is the compact input space of $\boldsymbol{X}$ and $C(\mathcal{X})$ is the space of all continuous functions $g : \mathcal{X} \to \mathbb{R}$. The kernel K is said to be *universal* if the function space $\mathcal{H}_K$ generated by K is dense in $\mathcal{C}(\mathcal{X})$, that is, for any positive ϵ and any function $g \in \mathcal{C}(\mathcal{X})$, there exists an $f \in \mathcal{H}_K$ such that $||f - g||_\infty < \epsilon$.

Theorem 1. *Suppose $\hat{f}_n$ is the solution of (4.5), $\mathcal{H}_K$ is induced by a universal kernel K, and the sample space $\mathcal{X}$ is compact. Then we have*

(1) $\varepsilon_A = 0$;

(2) Let $B = \sup_{\boldsymbol{x}} K(\boldsymbol{x}, \boldsymbol{x}) < \infty$. When $\lambda_n \to 0$ and $n\lambda_n \to \infty$, for any $\epsilon > 0$,

$$\lim_{n \to \infty} P\Big(\varepsilon_E(\hat{f}_n) > \epsilon\Big) = 0.$$

By (1) and (2) and (4.6) we have $R(\hat{f}_n) \to R(f^*)$ in probability.

The Gaussian kernel is universal and $B \leq 1$. Thus Theorem 1 says that the kernel DWD using the Gaussian kernel is Bayes risk consistent. This offers a theoretical explanation to the numerical results in Figure 1.

5 Real Data Analysis

In this section, we investigate the performance of `kerndwd` on four benchmark data sets: the BUPA liver disorder data, the Haberman's survival data, the Connectionist Bench (sonar, mines vs. rocks) data, and the vertebral column data. All the data sets were obtained from UCI Machine Learning Repository (Lichman, 2013).

For comparison purposes, we considered the SVM, the standard DWD ($q = 1$) and the generalized DWD models with $q = 0.5, 4, 8$. We computed all DWD models using our R package `kerndwd` and solved the SVM using the R package `kernlab` (Karatzoglou et al., 2004). We randomly split each data into a training and a test set with a ratio $2 : 1$. For each method using the linear kernel, we conducted a five-folder cross-validation on the training set to tune λ. For each method using Gaussian kernels, the pair of (σ, λ) was tuned by the five-folder cross-validation. We then fitted each model with the selected λ and evaluated its prediction accuracy on the test set.

Table 2 displays the average timing and mis-classification rates. We do not argue that either SVM or DWD outperforms the other; nevertheless, two models are highly comparable. SVM models work better on sonar and vertebral data, and DWD performs better on bupa and haberman data. For three out of the four data sets, the best method uses a Gaussian kernel, indicating that linear classifiers may not be adequate in such cases. In terms of timing, `kerndwd` runs faster than `kernlab` in all these examples. It is also interesting to see that DWD with $q = 0.5$ can work slightly better than DWD with $q = 1$ on bupa and haberman data, although the difference is not significant.

6 Discussion

In this paper we have developed a new algorithm for solving the linear generalized DWD and the kernel generalized DWD. Compared with the current state-of-the-art algorithm for solving the linear DWD, our new algorithm is easier to understand, more general, and much more efficient. DWD equipped with the new algorithm can be computationally more efficient than the SVM. We have established the statistical learning theory of the kernel generalized DWD, showing that the kernel DWD and the kernel SVM are comparable in theory. Our theoretical analysis and algorithm do not suggest DWD with $q = 1$ has any special merit compared to the other members in the generalized DWD family. Numerical examples further support our theoretical conclusions. DWD with $q = 1$ is called the standard DWD purely

Table 2. *The mis-classification rates and timings (in seconds) for four benchmark data sets. Each data set was split into a training and a test set. On the training set, the tuning parameters were selected by five-fold cross-validation and the models were fitted accordingly. The mis-classification rates were assessed on the test sets. All the timings include tuning parameters. For each dataset, the method with the best prediction accuracy is marked by black boxes.*

		Bupa $n = 345$, $p = 6$			Haberman $n = 305$, $p = 3$			Sonar $n = 208$, $p = 60$			Vertebral $n = 310$, $p = 6$		
		error (%)		time	error (%)		time	error (%)		time	error (%)		time
linear kernel	SVM	31.63	(0.50)	17.47	26.97	(0.53)	11.74	25.97	(0.66)	8.01	**14.83**	(0.42)	8.07
	DWD $q = 1$	34.82	(0.75)	0.05	26.71	(0.54)	0.03	25.65	(0.75)	0.30	16.76	(0.53)	0.07
	DWD $q = 0.5$	34.23	(0.72)	0.06	26.73	(0.53)	0.04	25.10	(0.72)	0.35	16.54	(0.51)	0.10
	DWD $q = 4$	35.08	(0.71)	0.05	26.69	(0.55)	0.03	26.00	(0.76)	0.32	16.54	(0.53)	0.06
	DWD $q = 8$	35.08	(0.76)	0.06	26.53	(0.56)	0.03	25.97	(0.71)	0.34	17.01	(0.53)	0.06
Gaussian kernel	SVM	32.23	(0.48)	6.57	27.92	(0.61)	6.00	**15.65**	(0.56)	8.96	16.50	(0.46)	6.07
	DWD $q = 1$	32.14	(0.63)	2.83	26.46	(0.57)	2.03	20.67	(0.76)	0.83	17.57	(0.49)	2.23
	DWD $q = 0.5$	**31.62**	(0.61)	2.80	**26.42**	(0.58)	2.06	21.42	(0.79)	0.84	17.59	(0.56)	2.27
	DWD $q = 4$	31.63	(0.61)	3.05	26.42	(0.57)	2.08	20.26	(0.76)	0.91	17.15	(0.50)	2.28
	DWD $q = 8$	32.07	(0.57)	3.28	26.53	(0.56)	2.21	20.00	(0.67)	0.98	16.93	(0.50)	2.39

due to the fact that it, not other generalized DWDs, can be solved by SOCP when the DWD idea was first proposed. Now with our new algorithm and theory, practitioners have the option to explore different DWD classifiers.

In the present paper we have considered the standard classification problem under the 0-1 loss. In many applications we may face the so-called non-standard classification problems. For example, observed data may be collected via biased sampling and/or we need to consider unequal costs for different types of mis-classification. Qiao et al. (2010) introduced a weighted DWD to handle the non-standard classification problem, which follows the treatment of the non-standard SVM in Lin et al. (2002). Qiao et al. (2010) defined the weighted DWD as follows,

$$\min_{\beta_0,\boldsymbol{\beta}} \left[\sum_{i=1}^{n} w(y_i) \left(\frac{1}{r_i} + c\xi_i \right) \right], \text{ subject to } r_i = y_i(\beta_0 + \boldsymbol{x}_i^T\boldsymbol{\beta}) + \xi_i \geq 0 \text{ and } \boldsymbol{\beta}^T\boldsymbol{\beta} = 1, \qquad (6.1)$$

which can be further generalized to the weighted kernel DWD:

$$\min_{\beta_0,\boldsymbol{\alpha}} \boldsymbol{C}_w(\beta_0, \boldsymbol{\alpha}) \equiv \min_{\beta_0,\boldsymbol{\alpha}} \left[\frac{1}{n} \sum_{i=1}^{n} w(y_i) V_q \left(y_i(\beta_0 + \boldsymbol{K}_i^T\boldsymbol{\alpha}) \right) + \lambda \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha} \right]. \qquad (6.2)$$

Qiao et al. (2010) gave the expressions for $w(y_i)$ for various non-standard classification problems. Qiao et al. (2010) solved the weighted DWD with $q = 1$ (6.1) based on the second-order-cone programming. The MM procedure for Algorithm 1 and Algorithm 2 can easily accommodate the weight factors $w(y_i)$'s to solve the weighted DWD and weighted kernel DWD. We have implemented the weighted DWD in the R package `kerndwd`.

Appendix: technical proofs

Proof of Lemma 1

Write $v_i = y_i(\omega_0 + \boldsymbol{x}_i^T\boldsymbol{\omega})$ and $G(\eta_i) = 1/(v_i + \eta_i)^q + c\eta_i$. The objective function of (2.10) can be written as $\sum_{i=1}^n G(\eta_i)$. We next minimize (2.10) over η_i for every fixed i by computing the first-order and the second-order derivatives of $G(\eta_i)$:

$$G'(\eta_i) = -\frac{q}{(v_i+\eta_i)^{q+1}} + c = 0 \Rightarrow v_i + \eta_i = \left(\frac{q}{c}\right)^{\frac{1}{q+1}},$$
$$G''(\eta_i) = \frac{q(q+1)}{(v_i+\eta_i)^{q+2}} > 0.$$

If $v_i > (\frac{q}{c})^{\frac{1}{q+1}}$, then $G'(\eta_i) > 0$ for all $\eta_i \geq 0$, and $\eta_i^\star = 0$ is the minimizer. If $v_i \leq (\frac{q}{c})^{\frac{1}{q+1}}$, then $\eta_i^\star = (\frac{q}{c})^{\frac{1}{q+1}} - v_i$ is the minimizer as $G'(\eta^\star) = 0$ and $G''(\eta^\star) > 0$.

By plugging in the minimizer $\eta_i^\star$ into $\sum_{i=1}^n G(\eta_i)$, we obtain

$$\min_{\omega_0,\boldsymbol{\omega}} \sum_{i=1}^n \tilde{V}_q\left(y_i(\omega_0 + \boldsymbol{x}_i^T\boldsymbol{\omega})\right), \text{ subject to } \boldsymbol{\omega}^T\boldsymbol{\omega} = 1, \tag{6.3}$$

where

$$\tilde{V}_q(v) = \begin{cases} \left(\frac{q}{c}\right)^{-\frac{q}{q+1}} + c\left(\frac{q}{c}\right)^{\frac{1}{q+1}} - cv, & \text{if } v \leq \left(\frac{q}{c}\right)^{\frac{1}{q+1}}, \\ \frac{1}{v^q}, & \text{if } v > \left(\frac{q}{c}\right)^{\frac{1}{q+1}}. \end{cases}$$

We now simplify (6.3). Suppose $t = (\frac{q}{q+1})(\frac{q}{c})^{-\frac{1}{q+1}}$ and $t_1 = (\frac{1}{q+1})(\frac{q}{c})^{\frac{q}{q+1}}$. We define $V_q(u) = t_1 \cdot \tilde{V}_q(u/t)$ for each q,

$$V_q(u) = \begin{cases} 1 - u, & \text{if } u \leq \dfrac{q}{q+1}, \\ \dfrac{1}{u^q}\dfrac{q^q}{(q+1)^{q+1}}, & \text{if } u > \dfrac{q}{q+1}. \end{cases}$$

By setting $\beta_0 = t \cdot \omega_0$ and $\boldsymbol{\beta} = t \cdot \boldsymbol{\omega}$, we find that (6.3) becomes

$$\min_{\beta_0, \boldsymbol{\beta}} \sum_{i=1}^{n} V_q \left(y_i(\beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}) \right), \text{ subject to } \boldsymbol{\beta}^T \boldsymbol{\beta} = t^2,$$

which can be further transformed to (3.1) with λ and t one-to-one correspondent.

Proof of Lemma 2

We first prove (3.5). We observe that $0 < V_q''(u) = \frac{1}{u^{q+2}} \frac{q^{q+1}}{(q+1)^q} < \frac{(q+1)^2}{q}$, for any $u > \frac{q}{q+1}$. Also $V_q'(u)$ is continuous on $[\frac{q}{q+1}, \infty)$ and differentiable on $(\frac{q}{q+1}, \infty)$.

If both u_1 and $u_2 > \frac{q}{q+1}$, then the mean value theorem implies that there exists $u^{\star\star} > \frac{q}{q+1}$, such that,

$$\frac{|V_q'(u_1) - V_q'(u_2)|}{|u_1 - u_2|} = |V_q''(u^{\star\star})| < \frac{(q+1)^2}{q}. \tag{6.4}$$

If $u_1 > \frac{q}{q+1}$ and $u_2 \leq \frac{q}{q+1}$, then $V_q'(u_2) = V_q' \left(\frac{q}{q+1} \right) = -1$. The mean value theorem implies that there exists $u^{\star\star} > \frac{q}{q+1}$ satisfying

$$\frac{|V_q'(u_1) - V_q'(u_2)|}{|u_1 - u_2|} \leq \frac{|V_q'(u_1) - V_q'(\frac{q}{q+1})|}{|u_1 - \frac{q}{q+1}|} = |V_q''(u^{\star\star})| < \frac{(q+1)^2}{q}. \tag{6.5}$$

If both u_1 and $u_2 \leq \frac{q}{q+1}$, $V_q'(u_1) = V_q'(u_2) = -1$. It is trivial that

$$\frac{|V_q'(u_1) - V_q'(u_2)|}{|u_1 - u_2|} = 0 < \frac{(q+1)^2}{q}. \tag{6.6}$$

By (6.4), (6.5), and (6.6), we prove (3.5).

We now prove (3.6). Let $\nu(a) \equiv \frac{(q+1)^2}{2q} a^2 - V_q(a)$. From (3.5), it is not hard to show that $\nu'(a) = \frac{(q+1)^2}{q} a - V_q'(a)$ is strictly increasing. Therefore $\nu(a)$ is a strictly convex function, and its first-order condition, $\nu(t) > \nu(\tilde{t}) + \nu'(\tilde{t})(t - \tilde{t})$, verifies (3.6) directly.

Proof of Lemma 3

Given that $\eta(\boldsymbol{x}) = P(Y = 1|\boldsymbol{X} = \boldsymbol{x})$, we have that $E_{\boldsymbol{X}Y}\left[V_q(Yf(\boldsymbol{X}))\right] \equiv E_{\boldsymbol{X}}\zeta(f(\boldsymbol{X}))$:

$$\begin{aligned}\zeta(f(\boldsymbol{x})) &\equiv \eta(\boldsymbol{x})V_q(f(\boldsymbol{x})) + [1-\eta(\boldsymbol{x})]V_q(-f(\boldsymbol{x})) \\ &= \begin{cases} \eta(\boldsymbol{x})\dfrac{1}{f(\boldsymbol{x})^q}\dfrac{q^q}{(q+1)^{q+1}} + [1-\eta(\boldsymbol{x})][1+f(\boldsymbol{x})], & \text{if } f(\boldsymbol{x}) > \dfrac{q}{q+1}, \\ \eta(\boldsymbol{x})[1-f(\boldsymbol{x})] + [1-\eta(\boldsymbol{x})][1+f(\boldsymbol{x})], & \text{if } -\dfrac{q}{q+1} \le f(\boldsymbol{x}) \le \dfrac{q}{q+1}, \\ \eta(\boldsymbol{x})[1-f(\boldsymbol{x})] + [1-\eta(\boldsymbol{x})]\dfrac{1}{[-f(\boldsymbol{x})]^q}\dfrac{q^q}{(q+1)^{q+1}}, & \text{if } f(\boldsymbol{x}) < -\dfrac{q}{q+1}. \end{cases}\end{aligned}$$

For each given $\boldsymbol{x}$, we take both $f(\boldsymbol{x})$ and $\eta(\boldsymbol{x})$ as scalars and hereby write them as f and η respectively. We then take $\zeta(f) = \zeta(f(\boldsymbol{x}))$ as a function of f and compute the derivative with respect to f:

$$\frac{\partial\zeta(f)}{\partial f} = \begin{cases} -\eta\dfrac{1}{f^{q+1}}\dfrac{q^{q+1}}{(q+1)^{q+1}} + 1 - \eta, & \text{if } f > \dfrac{q}{q+1}, \\ 1 - 2\eta, & \text{if } -\dfrac{q}{q+1} \le f \le \dfrac{q}{q+1}, \\ -\eta + (1-\eta)\dfrac{1}{(-f)^{q+1}}\dfrac{q^{q+1}}{(q+1)^{q+1}}, & \text{if } f < -\dfrac{q}{q+1}. \end{cases}$$

We see that (1) when $\eta > 0.5$, $\partial\zeta(f)/\partial f = 0$ only when $f = \tilde{f} \equiv \frac{q}{q+1}\left(\frac{\eta}{1-\eta}\right)^{\frac{1}{q+1}}$, and (2) when $\eta < 0.5$, $\partial\zeta(f)/\partial f = 0$ only when $f = \tilde{f} \equiv -\frac{q}{q+1}\left(\frac{1-\eta}{\eta}\right)^{\frac{1}{q+1}}$. For these two cases, we also observe that

$$\begin{cases} \partial\zeta(f)/\partial f < 0, \text{ if } f < \tilde{f}, \\ \partial\zeta(f)/\partial f > 0, \text{ if } f > \tilde{f}, \end{cases} \tag{6.7}$$

which follows that $\tilde{f}$ is the minimizer of $\zeta(f)$.

Proof of Lemma 4

As $\tilde{f}(\boldsymbol{x})$ was defined in (4.4), we see that for each $\boldsymbol{x}$,

$$\begin{aligned}\zeta\left(\tilde{f}(\boldsymbol{x})\right) &\equiv \eta(\boldsymbol{x})V_q\left(\tilde{f}(\boldsymbol{x})\right) + [1-\eta(\boldsymbol{x})]V_q\left(-\tilde{f}(\boldsymbol{x})\right) \\ &= \begin{cases} \eta(\boldsymbol{x}) + [1-\eta(\boldsymbol{x})]^{\frac{1}{q+1}}\eta(\boldsymbol{x})^{\frac{q}{q+1}}, & \text{if } \eta(\boldsymbol{x}) \le 1/2, \\ 1 - \eta(\boldsymbol{x}) + \eta(\boldsymbol{x})^{\frac{1}{q+1}}[1-\eta(\boldsymbol{x})]^{\frac{q}{q+1}}, & \text{if } \eta(\boldsymbol{x}) > 1/2, \end{cases} \\ &= \frac{1}{2}\Big(1 - |2\eta(\boldsymbol{x}) - 1|\Big) + \frac{1}{2}\Big(1 + |2\eta(\boldsymbol{x}) - 1|\Big)^{\frac{1}{q+1}}\Big(1 - |2\eta(\boldsymbol{x}) - 1|\Big)^{\frac{q}{q+1}}.\end{aligned}$$

For $a \in [0,1]$, we define $\gamma(a)$ and compute its first-order derivative as follows,

$$\begin{aligned}
\gamma(a) &\equiv 1 - \frac{1}{2}(1-a) - \frac{1}{2}(1+a)^{\frac{1}{q+1}}(1-a)^{\frac{q}{q+1}} - \frac{q}{q+1}a, \\
\gamma'(a) &= \frac{1}{2} - \frac{1}{2(q+1)}\left(\frac{1-a}{1+a}\right)^{\frac{q}{q+1}} + \frac{q}{2(q+1)}\left(\frac{1+a}{1-a}\right)^{\frac{1}{q+1}} - \frac{q}{q+1} \\
&= \left[\frac{1}{2(q+1)} - \frac{1}{2(q+1)}\left(\frac{1-a}{1+a}\right)^{\frac{q}{q+1}}\right] + \left[\frac{q}{2(q+1)} + \frac{q}{2(q+1)}\left(\frac{1+a}{1-a}\right)^{\frac{1}{q+1}} - \frac{q}{q+1}\right] \geq 0.
\end{aligned}$$

Hence for each $a \in [0,1]$, $\gamma(a) \geq \gamma(0) = 0$. For each $\boldsymbol{x}$, let $a = |2\eta(\boldsymbol{x}) - 1|$ and we see that

$$1 - \zeta\left(\tilde{f}(\boldsymbol{x})\right) \geq \frac{q}{q+1}|2\eta(\boldsymbol{x}) - 1|.$$

By $R(f) = E_{\boldsymbol{X}Y}[Y \neq \operatorname{sign}(f(\boldsymbol{X})] = E_{\{\boldsymbol{X}:f(\boldsymbol{X})\geq 0\}}[1-\eta(\boldsymbol{X})] + E_{\{\boldsymbol{X}:f(\boldsymbol{X})\leq 0\}}\eta(\boldsymbol{X})$, we obtain

$$\begin{aligned}
R(\hat{f}_n) - R(f^\star) &= E_{\{\boldsymbol{X}:\hat{f}_n(\boldsymbol{X})\geq 0,\ f^\star(\boldsymbol{X})<0\}}[1-2\eta(\boldsymbol{X})] + E_{\{\boldsymbol{X}:\hat{f}_n(\boldsymbol{X})\leq 0,\ f^\star(\boldsymbol{X})>0\}}[2\eta(\boldsymbol{X})-1] \\
&\leq E_{\{\boldsymbol{X}:\hat{f}_n(\boldsymbol{X})f^\star(\boldsymbol{X})\leq 0\}}|2\eta(\boldsymbol{X})-1| \\
&\leq \frac{q+1}{q}E_{\{\boldsymbol{X}:\hat{f}_n(\boldsymbol{X})f^\star(\boldsymbol{X})\leq 0\}}\left[1-\zeta\left(\tilde{f}(\boldsymbol{X})\right)\right].
\end{aligned} \tag{6.8}$$

Since $f^\star(\boldsymbol{X})$ and $\tilde{f}(\boldsymbol{X})$ share the same sign, $\hat{f}_n(\boldsymbol{X})f^\star(\boldsymbol{X}) \leq 0$ implies that $\hat{f}_n(\boldsymbol{X})\tilde{f}(\boldsymbol{X}) \leq 0$. When $\hat{f}_n(\boldsymbol{X})\tilde{f}(\boldsymbol{X}) \leq 0$, 0 is between $\hat{f}_n(\boldsymbol{X})$ and $\tilde{f}(\boldsymbol{X})$, and thus (6.7) indicates that $\zeta(\tilde{f}(\boldsymbol{X})) \leq \zeta(0) = 1 \leq \zeta(\hat{f}_n(\boldsymbol{X}))$. From (6.8), we conclude that

$$\begin{aligned}
R(\hat{f}_n) - R(f^\star) &\leq \frac{q+1}{q}E_{\{\boldsymbol{X}:\hat{f}_n(\boldsymbol{X})f^\star(\boldsymbol{X})\leq 0\}}\left[\zeta\left(\hat{f}_n(\boldsymbol{X})\right) - \zeta\left(\tilde{f}(\boldsymbol{X})\right)\right] \\
&\leq \frac{q+1}{q}E_{\boldsymbol{X}}\left[\zeta\left(\hat{f}_n(\boldsymbol{X})\right) - \zeta\left(\tilde{f}(\boldsymbol{X})\right)\right] \\
&= \frac{q+1}{q}E_{\boldsymbol{X}Y}\left[V_q\left(Y\hat{f}_n(\boldsymbol{X})\right) - V_q\left(Y\tilde{f}(\boldsymbol{X})\right)\right] \\
&= \frac{q+1}{q}(\varepsilon_A + \varepsilon_E).
\end{aligned}$$

Proof of Theorem 1

Part (1). We first show that when $\mathcal{H}_K$ is induced by a universal kernel, the approximation error $\varepsilon_A = 0$. By definition, we need to show that for any $\epsilon > 0$, there exists $f_\epsilon \in \mathcal{H}_K$ such

that

$$\left| E_{\boldsymbol{X}Y} V_q \left(Y f_\epsilon(\boldsymbol{X}) \right) - E_{\boldsymbol{X}Y} V_q \left(Y \tilde{f}(\boldsymbol{X}) \right) \right| < \epsilon. \tag{6.9}$$

We first use truncation to consider a truncated version of $\tilde{f}$. For any given $\delta \in (0, 0.5)$, we define

$$f_\delta(\boldsymbol{X}) = \begin{cases} \frac{q}{q+1} \left(\frac{1-\delta}{\delta} \right)^{\frac{1}{q+1}}, & \text{if } \eta(\boldsymbol{X}) > 1-\delta, \\ \tilde{f}(\boldsymbol{X}), & \text{if } -\delta \le \eta(\boldsymbol{X}) \le 1-\delta, \\ -\frac{q}{q+1} \left(\frac{\delta}{1-\delta} \right)^{\frac{1}{q+1}}, & \text{if } \eta(\boldsymbol{X}) < \delta. \end{cases}$$

We have that

$$0 \le E_{\boldsymbol{X}Y} V_q \left(Y f_\delta(\boldsymbol{X}) \right) - E_{\boldsymbol{X}Y} V_q \left(Y \tilde{f}(\boldsymbol{X}) \right) = \kappa_+ + \kappa_-,$$

where

$$\begin{aligned} \kappa_+ =& E_{\boldsymbol{X}:\eta(\boldsymbol{X})>1-\delta} \left[\eta(\boldsymbol{X}) V_q(f_\delta(\boldsymbol{X})) + (1-\eta(\boldsymbol{X})) V_q(-f_\delta(\boldsymbol{X})) \right] \\ & - E_{\boldsymbol{X}:\eta(\boldsymbol{X})>1-\delta} \left[\eta(\boldsymbol{X}) V_q \left(\tilde{f}(\boldsymbol{X}) \right) + (1-\eta(\boldsymbol{X})) V_q \left(-\tilde{f}(\boldsymbol{X}) \right) \right], \\ \kappa_- =& E_{\boldsymbol{X}:\eta(\boldsymbol{X})<\delta} \left[\eta(\boldsymbol{X}) V_q(f_\delta(\boldsymbol{X})) + (1-\eta(\boldsymbol{X})) V_q(-f_\delta(\boldsymbol{X})) \right] \\ & - E_{\boldsymbol{X}:\eta(\boldsymbol{X})<\delta} \left[\eta(\boldsymbol{X}) V_q \left(\tilde{f}(\boldsymbol{X}) \right) + (1-\eta(\boldsymbol{X})) V_q \left(-\tilde{f}(\boldsymbol{X}) \right) \right]. \end{aligned}$$

Since $V_q(f_\delta(\boldsymbol{X})) < V_q(-f_\delta(\boldsymbol{X}))$ when $\eta(\boldsymbol{X}) > 1-\delta$,

$$\begin{aligned} \kappa_+ <& E_{\boldsymbol{X}:\eta(\boldsymbol{X})>1-\delta} \left[(1-\delta) V_q(f_\delta(\boldsymbol{X})) + \delta V_q(-f_\delta(\boldsymbol{X})) \right] \\ & - E_{\boldsymbol{X}:\eta(\boldsymbol{X})>1-\delta} \left[\eta(\boldsymbol{X}) V_q \left(\tilde{f}(\boldsymbol{X}) \right) + (1-\eta(\boldsymbol{X})) V_q \left(-\tilde{f}(\boldsymbol{X}) \right) \right] \\ =& \left[\delta + (1-\delta)^{\frac{1}{q+1}} \delta^{\frac{q}{q+1}} \right] - E_{\boldsymbol{X}:\eta(\boldsymbol{X})>1-\delta} \left[1 - \eta(\boldsymbol{X}) + \eta(\boldsymbol{X})^{\frac{1}{q+1}} (1-\eta(\boldsymbol{X}))^{\frac{q}{q+1}} \right]. \end{aligned}$$

We notice that $(1-a) + a^{\frac{1}{q+1}} (1-a)^{\frac{q}{q+1}}$ is a continuous function in terms of $a \in (0,1)$. Since $\eta(\boldsymbol{X}) > 1-\delta$ implies that $|\eta(\boldsymbol{X}) - (1-\delta)| < \delta$, we conclude that for any given $\epsilon > 0$, there exists a sufficiently small δ such that $\kappa_+ < \epsilon/6$. We can also obtain $\kappa_- < \epsilon/6$ in the same spirit. Therefore,

$$0 \le E_{\boldsymbol{X}Y} V_q \left(Y f_\delta(\boldsymbol{X}) \right) - E_{\boldsymbol{X}Y} V_q \left(Y \tilde{f}(\boldsymbol{X}) \right) \le \kappa_+ + \kappa_- < \epsilon/3. \tag{6.10}$$

By Lusin's Theorem, there exists a continuous function $\varrho(\boldsymbol{X})$ such that $P(\varrho(\boldsymbol{X}) \neq$

$f_\delta(\boldsymbol{X})) \leq \epsilon(q+1)/(6q)$. Notice that $\sup_{\boldsymbol{X}} |f_\delta(\boldsymbol{X})| \leq q/(q+1)$. Define

$$\tau(\boldsymbol{X}) = \begin{cases} \varrho(\boldsymbol{X}), & \text{if } |\varrho(\boldsymbol{X})| \leq \dfrac{q}{q+1}, \\ \dfrac{q}{q+1} \cdot \dfrac{\varrho(\boldsymbol{X})}{|\varrho(\boldsymbol{X})|}, & \text{if } |\varrho(\boldsymbol{X})| > \dfrac{q}{q+1}, \end{cases}$$

then $P(\tau(\boldsymbol{X}) \neq f_\delta(\boldsymbol{X})) \leq \epsilon(q+1)/(6q)$ as well. Hence

$$\begin{aligned} \left| E_{\boldsymbol{X}Y} V_q\left(Y f_\delta(\boldsymbol{X})\right) - E_{\boldsymbol{X}Y} V_q\left(Y \tau(\boldsymbol{X})\right) \right| \leq & \quad E_{\boldsymbol{X}} |f_\delta(\boldsymbol{X}) - \tau(\boldsymbol{X})| \\ = & \quad E_{\{\boldsymbol{X}:\tau(\boldsymbol{X}) \neq f_\delta(\boldsymbol{X})\}} |f_\delta(\boldsymbol{X}) - \tau(\boldsymbol{X})| \\ \leq & \quad \frac{2q}{q+1} \cdot \frac{\epsilon(q+1)}{6q} = \epsilon/3, \end{aligned}$$

where the first inequality comes from the fact that $V_q(u)$ is Lipschitz continuous, i.e.,

$$|V_q(u_1) - V_q(u_2)| \leq |u_1 - u_2|, \ \forall u_1, u_2 \in \mathbb{R}.$$

Notice that $\tau(\boldsymbol{X})$ is also continuous. The definition of the universal kernel implies the existence of a function $f_\epsilon \in \mathcal{H}_K$ such that

$$\left| E_{\boldsymbol{X}Y} V_q\left(Y f_\epsilon(\boldsymbol{X})\right) - E_{\boldsymbol{X}Y} V_q\left(Y \tau(\boldsymbol{X})\right) \right| < \sup_{\boldsymbol{X}} |f_\epsilon(\boldsymbol{X}) - \tau(\boldsymbol{X})| < \epsilon/3. \tag{6.11}$$

By combining (6.10), (6.11), and (6.11) we obtain (6.9).

Part (2). In this part we bound the estimation error $\varepsilon_E(\hat{f}_n)$. Note that RKHS has the following reproducing property (Wahba, 1990; Hastie et al., 2009):

$$\begin{aligned} &\langle K(\boldsymbol{x}_i, \boldsymbol{x}), f(\boldsymbol{x}) \rangle_{\mathcal{H}_K} = f(\boldsymbol{x}_i), \\ &\langle K(\boldsymbol{x}_i, \boldsymbol{x}), K(\boldsymbol{x}_j, \boldsymbol{x}) \rangle_{\mathcal{H}_K} = K(\boldsymbol{x}_i, \boldsymbol{x}_j). \end{aligned} \tag{6.12}$$

Fix any $\epsilon > 0$. By the KKT condition of (4.5) and the representor theorem, we have

$$\frac{1}{n} \sum_{i=1}^{n} V_q'\left(y_i \hat{f}_n(\boldsymbol{x}_i)\right) y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + 2\lambda_n \hat{f}_n(\boldsymbol{x}) = 0. \tag{6.13}$$

We define $\hat{f}^{[k]}$ as the solution of (4.5) when the kth observation is excluded from the training

data, i.e.,

$$\hat{f}^{[k]} = \underset{f \in \mathcal{H}_K}{\operatorname{argmin}} \left[\frac{1}{n} \sum_{i=1, i \neq k}^{n} V_q \left(y_i (f(\boldsymbol{x}_i)) + \lambda_n ||f||_{\mathcal{H}_K}^2 \right] . \right. \tag{6.14}$$

By the definition of $\hat{f}^{[k]}$ and the convexity of V_q, we have

$$\begin{aligned} 0 \leq & \frac{1}{n} \sum_{i=1, i \neq k}^{n} V_q \left(y_i \hat{f}_n(\boldsymbol{x}_i) \right) + \lambda_n ||\hat{f}_n||_{\mathcal{H}_K}^2 - \frac{1}{n} \sum_{i=1, i \neq k}^{n} V_q \left(y_i \hat{f}^{[k]}(\boldsymbol{x}_i) \right) - \lambda_n ||\hat{f}^{[k]}||_{\mathcal{H}_K}^2 \\ \leq & - \frac{1}{n} \sum_{i=1, i \neq k}^{n} V_q' \left(y_i \hat{f}_n(\boldsymbol{x}_i) \right) y_i \left(\hat{f}^{[k]}(\boldsymbol{x}_i) - \hat{f}_n(\boldsymbol{x}_i) \right) + \lambda_n ||\hat{f}_n||_{\mathcal{H}_K}^2 - \lambda_n ||\hat{f}^{[k]}||_{\mathcal{H}_K}^2 . \end{aligned}$$

By the reproducing property, we further have

$$\begin{aligned} 0 \leq & - \frac{1}{n} \sum_{i=1, i \neq k}^{n} V_q' \left(y_i \hat{f}_n(\boldsymbol{x}_i) \right) y_i \left\langle K(\boldsymbol{x}_i, \boldsymbol{x}), \hat{f}^{[k]}(\boldsymbol{x}) - \hat{f}_n(\boldsymbol{x}) \right\rangle_{\mathcal{H}_K} + \lambda_n ||\hat{f}_n||_{\mathcal{H}_K}^2 - \lambda_n ||\hat{f}^{[k]}||_{\mathcal{H}_K}^2 \\ = & - \frac{1}{n} \sum_{i=1, i \neq k}^{n} V_q' \left(y_i \hat{f}_n(\boldsymbol{x}_i) \right) y_i \left\langle K(\boldsymbol{x}_i, \boldsymbol{x}), \hat{f}^{[k]}(\boldsymbol{x}) - \hat{f}_n(\boldsymbol{x}) \right\rangle_{\mathcal{H}_K} \\ & - 2\lambda_n \left\langle \hat{f}_n(\boldsymbol{x}), \hat{f}^{[k]}(\boldsymbol{x}) - \hat{f}_n(\boldsymbol{x}) \right\rangle_{\mathcal{H}_K} - \lambda_n ||\hat{f}^{[k]} - \hat{f}_n||_{\mathcal{H}_K}^2 \\ = & \frac{1}{n} V_q' \left(y_k \hat{f}_n(\boldsymbol{x}_k) \right) y_k \left\langle K(\boldsymbol{x}_k, \boldsymbol{x}), \hat{f}^{[k]}(\boldsymbol{x}) - \hat{f}_n(\boldsymbol{x}) \right\rangle_{\mathcal{H}_K} - \lambda_n ||\hat{f}^{[k]} - \hat{f}_n||_{\mathcal{H}_K}^2 , \end{aligned}$$

where the equality in the end holds by (6.13). Thus, by Cauchy-Schwartz inequality,

$$\begin{aligned} & n\lambda_n ||\hat{f}^{[k]} - \hat{f}_n||_{\mathcal{H}_K}^2 \leq V_q' \left(y_k \hat{f}_n(\boldsymbol{x}_k) \right) y_k \left\langle K(\boldsymbol{x}_k, \boldsymbol{x}), \hat{f}^{[k]}(\boldsymbol{x}) - \hat{f}_n(\boldsymbol{x}) \right\rangle_{\mathcal{H}_K} \\ \leq & \left| V_q' \left(y_k \hat{f}_n(\boldsymbol{x}_k) \right) \right| ||K(\boldsymbol{x}_k, \boldsymbol{x})||_{\mathcal{H}_K} ||\hat{f}^{[k]} - \hat{f}_n||_{\mathcal{H}_K} \leq \sqrt{K(\boldsymbol{x}_k, \boldsymbol{x}_k)} \cdot ||\hat{f}^{[k]} - \hat{f}_n||_{\mathcal{H}_K} , \end{aligned}$$

which implies

$$||\hat{f}^{[k]} - \hat{f}_n||_{\mathcal{H}_K} \leq \frac{\sqrt{B}}{n\lambda_n},$$

where $B = \sup_{\boldsymbol{x}} K(\boldsymbol{x}, \boldsymbol{x})$. By the reproducing property, we have

$$\begin{aligned} |\hat{f}^{[k]}(\boldsymbol{x}_k) - \hat{f}_n(\boldsymbol{x}_k)|^2 &= \left(\langle K(\boldsymbol{x}_i, \boldsymbol{x}_k), \hat{f}^{[k]}(\boldsymbol{x}_i) - \hat{f}_n(\boldsymbol{x}_i) \rangle_{\mathcal{H}_K} \right)^2 \\ & \leq K(\boldsymbol{x}_k, \boldsymbol{x}_k) ||\hat{f}^{[k]} - \hat{f}_n||_{\mathcal{H}_K}^2 \leq B \left(\frac{\sqrt{B}}{n\lambda_n} \right)^2 . \end{aligned}$$

By the Lipschitz continuity of the DWD loss, we obtain that for each $k = 1, \ldots, n$,

$$V_q\left(y_k \hat{f}^{[k]}(\boldsymbol{x}_k)\right) - V_q\left(y_k \hat{f}_n(\boldsymbol{x}_k)\right) \leq \ |\hat{f}^{[k]}(\boldsymbol{x}_k) - \hat{f}_n(\boldsymbol{x}_k)| \leq \frac{B}{n\lambda_n},$$

and therefore,

$$\frac{1}{n}\sum_{k=1}^{n} V_q\left(y_k \hat{f}^{[k]}(\boldsymbol{x}_k)\right) \leq \frac{1}{n}\sum_{k=1}^{n} V_q\left(y_k \hat{f}_n(\boldsymbol{x}_k)\right) + \frac{B}{n\lambda_n}. \tag{6.15}$$

Let $f_\epsilon^* \in \mathcal{H}_K$ such that

$$E_{\boldsymbol{X}Y} V_q\left(Y f_\epsilon^*(\boldsymbol{X})\right) \leq \inf_{f \in \mathcal{H}_K} E_{\boldsymbol{X}Y} V_q\left(Y f(\boldsymbol{X})\right) + \epsilon/3. \tag{6.16}$$

By definition of $\hat{f}_n$, we have

$$\frac{1}{n}\sum_{k=1}^{n} V_q\left(y_k \hat{f}_n(\boldsymbol{x}_k)\right) + \lambda_n ||\hat{f}_n||_{\mathcal{H}_K}^2 \leq \frac{1}{n}\sum_{k=1}^{n} V_q\left(y_k f_\epsilon^*(\boldsymbol{x}_k)\right) + \lambda_n ||f_\epsilon^*||_{\mathcal{H}_K}^2. \tag{6.17}$$

Since each data point in $\boldsymbol{T}_n = \{(\boldsymbol{x}_k, y_k)\}_{k=1}^n$ is drawn from the same distribution, we have

$$E_{\boldsymbol{T}_n}\left[\frac{1}{n}\sum_{k=1}^{n} V_q\left(y_k \hat{f}^{[k]}(\boldsymbol{x}_k)\right)\right] = \frac{1}{n}\sum_{k=1}^{n} E_{\boldsymbol{T}_n} V_q\left(y_k \hat{f}^{[k]}(\boldsymbol{x}_k)\right) = E_{\boldsymbol{T}_{n-1}} E_{\boldsymbol{X}Y} V_q\left(Y \hat{f}_{n-1}(\boldsymbol{X})\right). \tag{6.18}$$

By combining (6.15)–(6.18) we have

$$E_{\boldsymbol{T}_{n-1}} E_{\boldsymbol{X}Y} V_q\left(Y \hat{f}_{n-1}(\boldsymbol{X})\right) \leq \inf_{f \in \mathcal{H}_K} E_{\boldsymbol{X}Y} V_q\left(Y f(\boldsymbol{X})\right) + \lambda_n ||f_\epsilon^*||_{\mathcal{H}_K}^2 + \frac{B}{n\lambda_n} + \frac{\epsilon}{3}. \tag{6.19}$$

By the choice of λ_n, we see that there exits N_ϵ such that when $n > N_\epsilon$ we have $\lambda_n < \epsilon/(3||f_\epsilon^*||_{\mathcal{H}_K}^2)$, $n\lambda_n > 3B/\epsilon$, and hence

$$E_{\boldsymbol{T}_{n-1}}\left[E_{\boldsymbol{X}Y} V_q\left(Y \hat{f}_{n-1}(\boldsymbol{X})\right)\right] \leq \inf_{f \in \mathcal{H}_K} E_{\boldsymbol{X}Y} V_q\left(Y f(\boldsymbol{X})\right) + \epsilon.$$

Because ϵ is arbitrary and $E_{\boldsymbol{T}_{n-1}}[E_{\boldsymbol{X}Y} V_q(Y \hat{f}_{n-1}(\boldsymbol{X}))] \geq \inf_{f \in \mathcal{H}_K} E_{\boldsymbol{X}Y} V_q\left(Y f(\boldsymbol{X})\right)$, we have $\lim_{n\to\infty} E_{\boldsymbol{T}_{n-1}}[E_{\boldsymbol{X}Y} V_q(Y \hat{f}_{n-1}(\boldsymbol{X}))] = \inf_{f \in \mathcal{H}_K} E_{\boldsymbol{X}Y} V_q\left(Y f(\boldsymbol{X})\right)$, which equivalently indicates that $\lim_{n\to\infty} E_{\boldsymbol{T}_n} \varepsilon_E(\hat{f}_n) = 0$. Since $\varepsilon_E(\hat{f}_n) \geq 0$, then by Markov inequality, we prove part (2).

References

Ahn, J., Marron, J.S., Muller, K., and Chi, Y. (2007), "The high-dimension, low-sample-size geometric representation holds under mild conditions," *Biometrika*, 94(3), 760–766.

Ahn, J. and Marron, J.S. (2010), "The maximal data-piling direction for discrimination," *Biometrika*, 97(1), 254–259.

Aizerman, A., Braverman, E., and Rozoner, L. (1964), "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and remote control*, 25, 821–837.

Alizadeh, F. and Goldfarb, D. (2004), "Second-order cone programming," *Mathematical Programming, Series B*, 95(1), 3–51.

Anthony, M. and Bartlett, P. (1999), *Neural Network Learning: Theoretical Foundations.*, Cambridge University Press, Cambridge.

Bartlett, P. and Shawe-Taylor, J. (1999), "Generalization performance of support vector machines and other pattern classifiers", *Advances in Kernel Methods–Support Vector Learning*, 43–54.

Boyd, S. and Vandenberghe, L. (2004), *Convex Optimization*, Cambridge University Press, Cambridge.

Breiman, L. (2001), "Random forests," *Machine Learning*, 45(1), 5–32.

De Leeuw, J. and Heiser, W. (1977), "Convergence of correction matrix algorithms for multidimensional scaling", 735–752.

Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014), "Do we need hundreds of classifiers to solve real world classification problems?" *The Journal of Machine Learning Research*, 15, 3133–3181.

Freund, Y. and Schapire, R. (1997), "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, 55(1), 119–139.

Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. (2007), "Pathwise coordinate optimization," *The Annals of Applied Statistics*, 1(2), 302–332.

Girosi, F., Jones, M., and Poggio, T. (1995), "Regularization theory and neural networks architectures," *Neural Computation*, 7(2), 219–269.

Hall, P., Marron, J.S., and Neeman, A. (2005), "Geometric representation of high dimensions, low sample size data," *Journal of the Royal Statistical Society, Series B*, 67(3), 427–444.

Hastie, T., Tibshirani, R., and Friedman, J. (2009), *The Elements of Statistical Learning: Prediction, Inference, and Data Mining*, 2nd edition, Springer-Verlag, New York.

Huang, H., Liu, Y., Du. Y., Perou, C., Hayes, D., Todd, M., and Marron, J.S. (2013), "Multiclass distance-weighted discrimination," *Journal of Computational and Graphical Statistics*, 22(4), 953–969.

Huang, H., Lu, X., Liu, Y., Haaland, P., and Marron, J.S. (2012), "R/DWD: distance-weighted discrimination for classification, visualization and batch adjustment," *Bioinformatics*, 28(8), 1182–1183.

Hunter, D. and Lange, K. (2004), "A tutorial on MM algorithms," *The American Statistician*, 58(1), 30–37.

Hunter, D. and Li, R. (2005), "Variable selection using MM algorithms," *The Annals of Statistics*, 33(4), 1617–1642.

Jaakkola, T. and Haussler, D. (1999), "Probabilistic kernel regression models," *Proceedings of the 1999 Conference on AI and Statistics*, 126, 00–04.

Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A.(2004), "kernlab – An S4 Package for Kernel Methods in R," *Journal of Statistical Software*, 11(9), 1–20.

Lange, K., Hunter, D., and Yang, I. (2000), "Optimization transfer using surrogate objective functions," *Journal of Computational and Graphical Statistics*, 9(1), 1–20.

Lange, K. and Zhou, H. (2014), "MM algorithms for geometric and signomial programming," *Mathematical Programming*, 143(1-2), 339–356.

Lichman, M. (2013), "UCI Machine Learning Repository," http://archive.ics.uci.edu/ml, Irvine, CA: University of California, School of Information and Computer Science.

Lin, Y., Lee, Y., and Wahba, G. (2002), "Support vector machines for classification in nonstandard situations," *Machine Learning*, 46, 191–202.

Lin, Y. (2002), "Support vector machines and the Bayes rule in classification," *Data Mining and Knowledge Discovery*, 6(3), 259–275.

Lin, Y. (2004), "A note on margin-based loss functions in classification," *Statistics & Probability Letters*, 68(1), 73–82.

Liu, Y., Zhang, H., and Wu, Y. (2011), "Hard or soft classification? Large-margin unified machines," *Journal of American Statistical Association*, 106(493), 166–177.

Marron, J.S., Todd, M., and Ahn, J. (2007), "Distance weighted discrimination," *Journal of American Statistical Association*, 102(480), 1267–1271.

Marron, J.S. (2013), "Smoothing, functional data analysis, and distance weighted discrimination software," http://www.unc.edu/~marron/marron_software.html.

Marron, J.S. (2015), "Distance-weighted discrimination," *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(2), 109–114.

Micchelli, C., Xu, Y., and Zhang, H. (2006), "Universal kernels," *Journal of Machine Learning Research*, 7, 2651–2667.

Qiao, X., Zhang, H., Liu, Y., Todd, M., Marron, J.S. (2010), "Weighted distance weighted discrimination and its asymptotic properties," *Journal of American Statistical Association*, 105(489), 401–414.

Qiao, X. and Zhang, L. (2015a), "Distance-weighted support vector machine," *Statistics and Its Interface*, 8(3), 331–345.

Qiao, X. and Zhang, L. (2015b), "Flexible high-dimensional classification machines and their asymptotic properties," *Journal of Machine Learning Research*, forthcoming.

Shawe-Taylor, J. and Cristianini, N. (2000), "Margin distribution and soft margin", *Advances in Kernel Methods–Support Vector Learning*, 349–358.

Steinwart, I. (2001), "On the influence of the kernel on the consistency of support vector machines," *Journal of Machine Learning Research*, 2, 67–93.

Tütüncü R., Toh, K., Todd, M. (2003), "Solving semidefinite-quadratic-linear programs using SDPT3," *Mathematical Programming*, 95(2), 189–217.

Vapnik, V. (1995), *The Nature of Statistical Learning Theory*, Springer-Verlag, New York.

Vapnik, V. (1998), *Statisitcal Learning Theory*, Wiley, New York.

Wahba, G. (1990), *Spline Models for Observational Data*, 59, SIAM.

Wahba, G., Gu, C., Wang, Y., and Campbell, R. (1994), "Soft classification, aka risk estimation, via penalized log likelihood and smoothing spline analysis of variance," In *Santa fe Institute Studies in the Sciences of Complexity-Proceeding Vol*, 20, Addison-Wesley Publishing CO, 331–331.

Wahba, G. (1999), "Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV," *Advances in Kernel Methods-Support Vector Learning*, 6, 69–87.

Wang, B. and Zou, H. (2015), "Sparse distance weighted discrimination," *Journal of Computational and Graphical Statistics*, forthcoming.

Wu, T.T. and Lange, K. (2008), "Coordinate descent algorithms for lasso penalized regression," *The Annals of Applied Statistics*, 2(1), 224–244.

Yang, Y. and Zou, H. (2013), "An efficient algorithm for computing the HHSVM and its generalizations," *Journal of Computational and Graphical Statistics*, 22(2), 396–415.

Zhang, T. (2004), "Statistical behavior and consistency of classification methods based on convex risk minimization," *The Annals of Statistics*, 32(1), 56–134.

Zhou, H. and Lange, K. (2010), "MM algorithms for some discrete multivariate distributions," *Journal of Computational and Graphical Statistics*, 19(3), 645–665.

Zhu, J. and Hastie, T. (2005), "Kernel logistic regression and the import vector machine," *Journal of Computational and Graphical Statistics*, 14(1), 185–205.

Zou, H. and Li, R. (2008), "One-step sparse estimates in nonconcave penalized likelihood models," *The Annals of Statistics*, 36(4), 1509-1533.