LETTER **Communicated by Xingye Qiao**

Robust Support Vector Machines for Classification with Nonconvex and Smooth Losses

Yunlong Feng
yunlong.feng@esat.kuleuven.be
Yuning Yang
yuning.yang@esat.kuleuven.be
Department of Electrical Engineering, ESAT-STADIUS, KU Leuven, 3000 Leuven, Belgium

Xiaolin Huang
xiaolinhuang@sjtu.edu.cn
Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, 200400 Shanghai, China

Siamak Mehrkanoon
Smehrkan@waterloo.ca
Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada

Johan A. K. Suykens
johan.suykens@esat.kuleuven.be
Department of Electrical Engineering, ESAT-STADIUS, KU Leuven, 3000 Leuven, Belgium

This letter addresses the robustness problem when learning a large margin classifier in the presence of label noise. In our study, we achieve this purpose by proposing robustified large margin support vector machines. The robustness of the proposed robust support vector classifiers (RSVC), which is interpreted from a weighted viewpoint in this work, is due to the use of nonconvex classification losses. Besides the robustness, we also show that the proposed RSCV is simultaneously smooth, which again benefits from using smooth classification losses. The idea of proposing RSVC comes from M-estimation in statistics since the proposed robust and smooth classification losses can be taken as one-sided cost functions in robust statistics. Its Fisher consistency property and generalization ability are also investigated. Besides the robustness and smoothness, another nice property of RSVC lies in the fact that its solution can be obtained by solving weighted squared hinge loss–based support vector machine problems iteratively. We further show that in each iteration, it is a quadratic programming problem in its dual space

Neural Computation **28, 1217–1247** (2016)
doi:10.1162/NECO_a_00837

and can be solved by using state-of-the-art methods. We thus propose an iteratively reweighted type algorithm and provide a constructive proof of its convergence to a stationary point. Effectiveness of the proposed classifiers is verified on both artificial and real data sets.

1 Introduction and Motivation

Over the past two decades, support vector machines for classification (SVC) have become prevalent tools in analyzing categorical data owing to their significant empirical successes in applications and also being amenable to theoretical analysis. The development of SVC has also fostered the development of statistical learning theory.

The key to SVC is to find a hyperplane (classifier) by introducing hard margins for separable data and soft margins for linearly nonseparable data, the purpose of which is to separate data as far as possible from the hyperplane. To deal with the nonlinear case, one applies the kernel trick in SVC and seeks the hyperplane in the feature space. The hyperplane learned from SVC that is based on the hinge loss also depends on the instances that are misclassified. However, in real-world applications, it may be the case that the real data set contains outliers. Here "outliers" refers to the instances that are far away "from the pattern set by the majority of the data" (Hampel, Ronchetti, Rousseeuw, & Stahel, 2011) and are "often very hard to identify in high-dimensional data sets due to the curse of dimensionality" (Steinwart & Christmann, 2008). Therefore, the fact that misclassified instances contribute to the hyperplane together with the contaminated data makes the learned classifier unreliable. To illustrate this, we carry out a toy example with the artificial Two Moons data set.

In the left panel of Figure 1, the two-dimensional data set and a classifier trained by support vector machine with the squared hinge loss (L2-SVM) are plotted, where the data set is not contaminated. It can be seen from Figure 1 that in this case, an ideal classifier can be obtained to separate the two classes perfectly. To show the influence of outliers, we then flip 10% of its labels. The contaminated data set and the obtained classifier by using the same method are plotted in the right panel of Figure 1. We can see that in this case, the obtained classifier has many wiggles, and obviously outliers have a significant influence on the resulting classifier.

1.1 Robustification of Support Vector Machine for Classification.

Outliers in the context of supervised learning have two implications. The first is data points with extreme explanatory variables—the so-called leverage points. The second implication of outliers is data points with extreme response variables, which are also of interest in this study. Outliers added in the right panel of Figure 1 belong to the second case. From Figure 1, we see that outliers can ruin the resulting classifier. In light of this, various

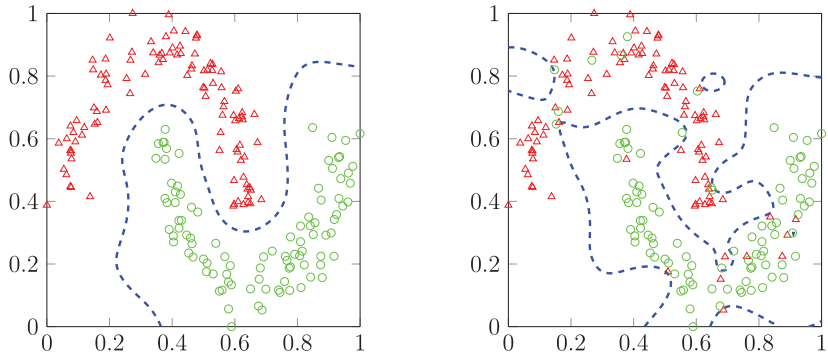

Figure 1: (Left) Plots of the Two Moons data set and the classifier trained by L2-SVM. The data set is not contaminated by outliers. (Right) Plots of the contaminated Two Moons data set and the classifier trained by L2-SVM with a gaussian kernel. The regularization parameter and the kernel bandwidth are tuned on a tuning set. The data set is contaminated by outliers, with 10% of its labels being flipped.

robust classification methods have been proposed to mitigate their effect. Roughly, there are three main approaches to dealing with outliers in classification problems: the data cleaning approach, the robust algorithm approach, and the robust model approach (Frénay & Verleysen, 2014).

In our study, we are mainly interested in the robust model approach to classification problems. One of the main strategies of introducing robustness to a classification model is applying a robust classification loss. A variety of studies in this line can be found in the literature. Here, we mention only several of them. For example, Shen, Tseng, Zhang, and Wong (2003) proposed a family of truncated nonconvex loss functions and applied them to the binary classification problem. By decomposing a nonconvex loss into the difference of two convex loss functions, Krause and Singer (2004) also addressed the robust classification problem. Wu and Liu (2007), Collobert, Sinz, Weston, and Bottou (2006), and Huang, Shi, and Suykens (2014) studied support vector machines on the basis of the truncated hinge loss and showed that the truncation operation could bring robustness and also sparser classifiers. Masnadi-Shirazi and Vasconcelos (2009) discussed the design of loss functions in classification problems and introduced a new robust classification loss. Park and Liu (2011) suggested using a truncated logistic loss to obtain robust probabilistic classifiers. Some special loss criteria were considered in Takeda, Fujiwara, and Kanamori (2014) and Kanamori, Fujiwara, and Takeda (2014) to produce robust classifiers. We notice that most of these methods are nonconvex. This is because researchers have realized that the enhanced robustness of learning machines that are based on nonconvex loss functions can be obtained from a breakdown viewpoint in the context

of regression (Kanamori et al., 2014) as well as classification (Shen et al., 2003; Reid & Williamson, 2010; Long & Servedio, 2010). Moreover, it has been also shown that nonconvexity approaches can provide scalability advantages over convexity (Collobert et al., 2006) in support vector machines for classification.

1.2 Smooth Support Vector Machines for Classification. Besides the robustness of support vector machines for classification, another important property in practice that one usually expects is its smoothness. It is obvious that due to using the nonsmooth hinge loss, SVC is not smooth. Therefore, more frequently, it is trained from its dual. Chapelle (2007) proposed training SVC in the primal by smoothing the nonsmooth hinge loss. In fact, in the literature of learning for classification, various smoothing techniques have been applied to SVC. For instance, Lee and Mangasarian (2001) proposed smooth SVM by replacing the hinge loss with a smooth enough loss. Wang, Zhu, and Zou (2007) studied a support vector classification problem by using a smooth variant of the hinge loss.

We also notice that a truncation operation applied to the loss function can deliver robustness to the model in classification problems. However, this operation also results in nondifferentiable loss functions, such as truncated hinge loss and truncated logistic loss. To the best of our knowledge, support vector machines for classification that are simultaneously robust and smooth have not been frequently employed. This is in fact our main reason for conducting this study.

1.3 Our Approach and Contributions. In this letter, we aim to design a large margin support vector classifier that is simultaneously robust and smooth. Our approach and contributions can be summarized as follows:

- We introduce a family of robust and smooth classification loss functions. Based on these classification loss functions, we then propose robust support vector machines for classification (RSVC) in reproducing kernel Hilbert space (RKHS).
- We show that RSVC has some connections with the weighted L2-SVM. Moreover, we show that solving RSVC can be done by solving an iteratively reweighted L2-SVM. The weighted L2-SVM can be efficiently solved in the dual via quadratic programming and also can be solved in the primal easily due to its smoothness.
- We interpret the robustness of RSVC from a weighted viewpoint and also study its Fisher consistency property and generalization ability.
- We provide an iteratively reweighted algorithm to solve RSVC and also prove its convergence. To our knowledge, we are the first to provide results that study the convergence of iteratively reweighted algorithms with noneven loss functions.

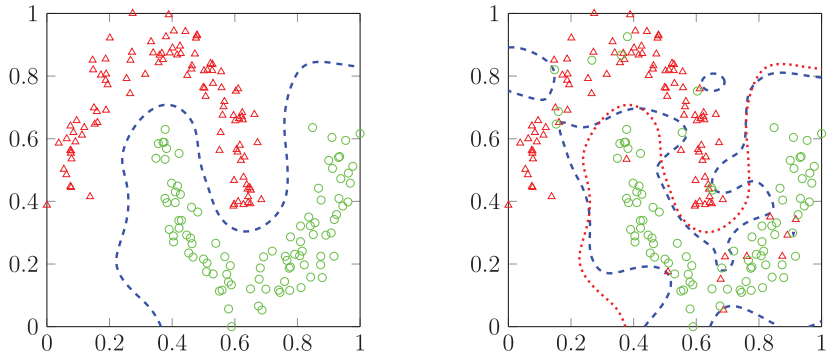

Figure 2: (Left) Plots of the Two Moons data set and a classifier trained by L2-SVM. The data set is not contaminated by outliers. (Right) Plots of the contaminated Two Moons data set and two classifiers. The data set is contaminated by outliers with 10% of its labels being flipped. The dashed blue curve is the classifier trained from L2-SVM. The dotted red curve stands for the classifier obtained from the proposed approach—RSVC.

To give a brief preview of the effectiveness of our approach, we again perform an artificial simulation on the contaminated Two Moons data set. The classifier obtained from RSVC is plotted in the right panel of Figure 2 with the dotted red curve. The dashed blue curve in the right panel is again obtained from L2-SVM, which is plotted for comparison. The left panel of Figure 2 is the same as that in Figure 1 and is plotted for comparison. It is easy to see from Figure 2 that the proposed robust classifier can be resistant to outliers and performs as well as the one in the left panel of Figure 2.

This paper is organized as follows. In section 2, we revisit classification losses and their robust or smooth variants. We then propose a family of robust and smooth classification losses that we use to formulate RSVC and study its connection with L2-SVM. Section 3 studies the properties of RSVC including its Fisher consistency, generalization ability, and robustness. Algorithms and related convergence analysis are given in section 4. In section 5, we illustrate the iteratively reweighted procedure of RSVC step by step with a toy example and then validate RSVC with UCI benchmark data sets. We conclude in section 6.

2 Proposed Robust Support Vector Machine for Classification

In this section, we present formulations of the proposed robust support vector classifiers. As noted, the robustness comes from the nonconvex and smooth classification loss functions. To better illustrate our method, we first revisit frequently employed classification loss functions and their

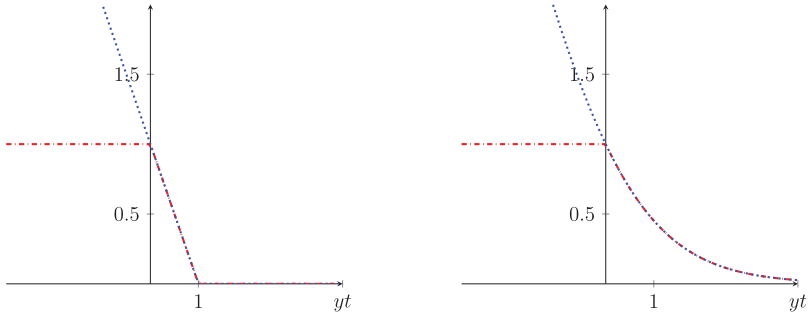

Figure 3: (Left) Plots of the hinge loss (dotted curve) and the truncated hinge loss (dotted-dashed curve). (Right) Plots of the logistic loss (dotted curve) and the truncated logistic loss (dotted-dashed curve).

robust or smooth variants. We then introduce a family of robust and smooth classification losses that we use to formulate the proposed robust classifiers.

2.1 Loss Functions for Classification and Their Robust or Smooth Variants. In regression problems, loss functions are used to measure the goodness of fit. As a binary-valued regression problem, probably the most intuitive loss function for classification is the misclassification loss. Mathematically, let $\mathcal{X} \subset \mathbb{R}^d$ be the input space, $\mathcal{Y} = \{-1, +1\}$ be the output space, and $f : \mathcal{X} \to \mathcal{Y}$ be a binary-valued classifier. For any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the misclassification loss ϕ_0 is defined as

$$\phi_0(y, f(x)) = \begin{cases} 0, & \text{if } yf(x) \geq 0, \\ 1, & \text{if } yf(x) < 0. \end{cases}$$

This misclassification loss ϕ_0 is nonconvex and not continuous. In the statistical machine learning literature, various classification loss functions have been proposed to serve as continuous and convex surrogates of the misclassification loss. Here we mention two typical classes of such surrogate classification losses. The first class is loss functions for producing probabilistic classifiers—for example, the logistic loss. The second class is margin-based classification loss functions, typical examples of which include hinge loss, Huberized hinge loss (Chapelle, 2007; Wang et al., 2007), squared hinge loss, and least squares loss.

A direct approach that delivers robustness to the model is applying the truncation operation to the classification losses. For instance, in the machine learning literature, the truncated hinge loss (Wu & Liu, 2007; Huang et al., 2014) and the truncated logistic loss (Park & Liu, 2011) have been proposed and drawn much attention (Wu & Liu, 2007; Brooks, 2011; Huang et al., 2014). The two truncated loss functions are plotted in Figure 3.

In Figure 3, the truncation operation leads to nonsmooth loss functions. However, some effort has been made to smoothen the hinge loss. A typical smoothened hinge loss is called Huberized hinge loss (Chapelle, 2007; Wang et al., 2007), which again is not robust to outliers due to its convexity and so penalizes the wrong misclassified instances at least linearly. As mentioned in section 1, support vector machines that are based on simultaneously robust and smooth classification losses have not been well developed. We therefore proceed by introducing a family of robust and smooth margin-based classification losses.

2.2 A Family of Nonconvex and Smooth Classification Losses. We first recall the following definition on margin-based classification loss from Steinwart and Christmann (2008):

Definition 1. *A loss function $\phi : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}^+$ is said to be a margin-based classification loss if there exists a representing function $\varphi : \mathbb{R} \to \mathbb{R}^+$ such that*

$$\phi(y,t) = \varphi(yt), \quad y \in \mathcal{Y},\ t \in \mathbb{R}, \quad \varphi'(0) \text{ exists and } \varphi'(0) < 0.$$

For any $t \in \mathbb{R}$, let us denote $t_+ := \max\{t, 0\}$. In this letter, we are interested in margin-based classification loss functions that satisfy the following assumptions:

Assumption 1. Suppose that ϕ is a margin-based classification loss that satisfies the following conditions:

1. There exists a representing function $\varphi : \mathbb{R}^+ \to \mathbb{R}^+$ such that $\phi(y,t) := \varphi((1-yt)_+)$ and $\varphi(0) = 0$.
2. φ is nondecreasing and continuously differentiable.
3. $\lim_{s \to +\infty} \varphi'(s) = 0$.
4. $\psi(0) := \lim_{s \to 0^+} \psi(s)$ exists and is finite, where $\psi(s) := \varphi'(s)/s$.

In definition 1, the first condition implies that the classification loss does not penalize instances that are correctly classified. Conditions 2 and 3 ensure that the penalization on the misclassified instances grows sufficiently slowly. As will be shown later, condition 4 is set to ensure that a certain relation between RSVC and L2-SVM holds. It should be noted that loss functions that satisfy the above assumption are nonconvex. Two typical classification losses that satisfy assumption 1 are as follows:

Example 1. A first example of the robust and smooth classification loss ℓ_σ that satisfies assumption 1 is

$$\ell_\sigma(y,t) = \sigma^2(1 - \exp(-(1-yt)_+^2/\sigma^2)), \quad y \in \mathcal{Y}, t \in \mathbb{R},$$

where $\sigma > 0$ is a scale parameter.

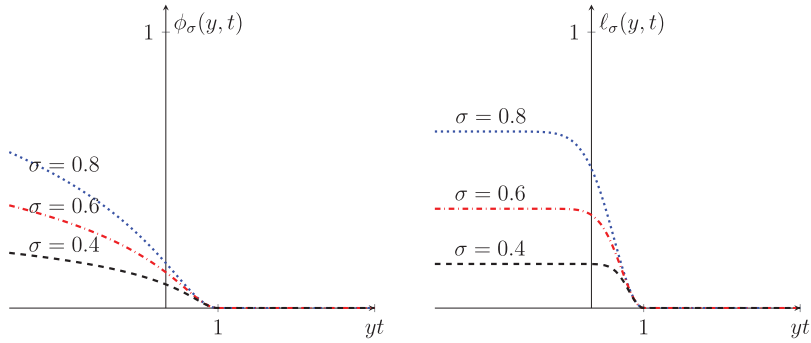

Figure 4: (Left) Plots of the loss function $\phi_\sigma(y,t)$ in example 2 with respect to yt for different σ values: $\sigma = 0.4$ (dashed curve), $\sigma = 0.6$ (dotted-dashed curve), and $\sigma = 0.8$ (dotted curve). (Right) Plots of the loss function $\ell_\sigma(y,t)$ in example 1 with respect to yt for different σ values: $\sigma = 0.4$ (dashed curve), $\sigma = 0.6$ (dotted-dashed curve), and $\sigma = 0.8$ (dotted curve).

Example 2. A second example of the robust and smooth classification loss ϕ_σ that satisfies assumption 1 is

$$\phi_\sigma(y,t) = \sigma^2 \log(1 + (1-yt)_+^2/\sigma^2), \ y \in \mathcal{Y}, t \in \mathbb{R},$$

where $\sigma > 0$ is a scale parameter.

In the above two loss functions, the positive scale parameter σ controls the influence of the residual $1 - yt$ for any $y \in \mathcal{Y}$ and $t \in \mathbb{R}$. Plots of the above two classification loss functions with different σ values are provided in Figure 4. It is easy to see from that figure that the larger σ is, the more the ℓ_σ loss penalizes the residual $1 - yt$. Moreover, the Taylor expansion of ℓ_σ with respect to yt shows that when σ is sufficiently large, there holds

$$\ell_\sigma(y,t) \approx (1-yt)_+^2, \quad y \in \mathcal{Y}, \ t \in \mathbb{R}.$$

Therefore, when σ goes to infinity, the ℓ_σ loss tends to the squared hinge loss, which is frequently employed for pursuing margin-based classifiers. Similar observations can be made for the classification loss ϕ_σ.

2.3 Formulations of the Proposed Robust Support Vector Classifiers. We now formulate the proposed robust support vector classifiers. We start with assuming that $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^m$ is a set of independent and identically distributed realizations of (X, Y) that takes values in $\mathcal{X} \times \mathcal{Y}$ with $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} = \{-1, +1\}$. Let ϕ be a margin-based classification loss that satisfies assumption 1.

We first consider the linear classifier f with $f(x) = \theta^\top x + b, x \in \mathcal{X}, \theta \in \mathbb{R}^d$ and $b \in \mathbb{R}$. Based on the above assumptions and notations, the proposed robust support vector classifier can be obtained by solving the following minimization problem,

$$(\theta_{\mathbf{z}},\ b_{\mathbf{z}}) = \operatorname*{argmin}_{\theta \in \mathbb{R}^d, b \in \mathbb{R}} \frac{1}{m} \sum_{i=1}^{m} \phi(y_i, \theta^\top x_i + b) + \lambda \|\theta\|_2^2, \tag{2.1}$$

where $\lambda > 0$ is a regularization parameter.

The nonlinear classifier is obtained through the kernel mapping. To this end, let us assume that $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a Mercer kernel and $\mathcal{H}_\mathcal{K}$ is a reproducing kernel Hilbert space induced by $\mathcal{K}$, which is the closure of the linear span of the set of functions $\{\mathcal{K}_x := \mathcal{K}(x, \cdot) : x \in \mathcal{X}\}$ with the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_\mathcal{K}} = \langle \cdot, \cdot \rangle_\mathcal{K}$ satisfying $\langle \mathcal{K}_x, \mathcal{K}_y \rangle$. Denote $\overline{\mathcal{H}}_\mathcal{K} = \mathcal{H}_\mathcal{K} + \mathbb{R}$. Then the proposed kernel-based robust support vector machine for classification can be formulated as

$$f_{\mathbf{z}} = \operatorname*{argmin}_{f \in \overline{\mathcal{H}}_\mathcal{K}} \frac{1}{m} \sum_{i=1}^{m} \phi(y_i, f(x_i)) + \lambda \|f\|_\mathcal{K}^2. \tag{2.2}$$

Noticing that the minimization problem, equation 2.2, works in an RKHS, and the penalty term is strictly monotonically increasing with respect to $\|f\|_\mathcal{K}$, one can then apply the representer theorem (Schölkopf, Herbrich, & Smola, 2001) to the optimization problem, equation 2.2. Consequently, the optimization problem, equation 2.2, can be reduced to a finite-dimensional optimization problem, and the solution $f_{\mathbf{z}}$ takes the form

$$f_{\mathbf{z}}(x) = \sum_{i=1}^{m} \alpha_{\mathbf{z},i} \mathcal{K}(x_i, x) + b_{\mathbf{z}},\ \ \alpha_{\mathbf{z},i} \in \mathbb{R},\ b_{\mathbf{z}} \in \mathbb{R},\ \ \forall x \in \mathcal{X},$$

where $\boldsymbol{\alpha}_{\mathbf{z}} = (\alpha_{\mathbf{z},1}, \ldots, \alpha_{\mathbf{z},m})^\top$. In what follows, without specification, our discussions will be concentrated on the kernel-based machine (see equation 2.2), which also apply to its linear counterpart, equation 2.1.

2.4 Connection with L2-SVM. We now show that there is an interesting relation between RSVC and L2-SVM.

Proposition 1. *Let the loss function ϕ in equation 2.2 be a margin-based classification loss with the representing function φ and satisfy assumption 1. Then any stationary point of the minimization problem, equation 2.2, can be obtained by solving an iteratively reweighted L2-SVM.*

Proof. From the discussion in section 2.3, we know that one can apply the representer theorem to the optimization problem, equation 2.2. As a result, solving the minimization problem, equation 2.2, can be reduced to solving the following finite-dimensional minimization problem,

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^m, b\in\mathbb{R}} \frac{1}{m}\sum_{i=1}^{m}\phi(y_i, \mathcal{K}_i^\top\boldsymbol{\alpha}+b)+\lambda\boldsymbol{\alpha}^\top\mathcal{K}\boldsymbol{\alpha}.$$

Since φ is the representing function of ϕ, we know from assumption 1 that the previous formula can be rewritten as

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^m, b\in\mathbb{R}} \frac{1}{m}\sum_{i=1}^{m}\varphi((1-y_i\mathcal{K}_i^\top\boldsymbol{\alpha}-y_ib)_+)+\lambda\boldsymbol{\alpha}^\top\mathcal{K}\boldsymbol{\alpha},$$

where for $i=1,\ldots,m$, $\mathcal{K}_i=(\mathcal{K}(x_1,x_i),\ldots,\mathcal{K}(x_m,x_i))^\top$, and $\mathcal{K}=(\mathcal{K}_1,\ldots,\mathcal{K}_m)^\top$.

Note that the minimization problem is nonconvex because of the nonconvexity of φ. Therefore, in general, only a stationary point of the above minimization problem can be expected. Let $(\boldsymbol{\alpha}^\star, b^\star)$ be one of its stationary points and further denote

$$\mathcal{R}(\boldsymbol{\alpha}, b) := \frac{1}{m}\sum_{i=1}^{m}\varphi((1-y_i\mathcal{K}_i^\top\boldsymbol{\alpha}-y_ib)_+)+\lambda\boldsymbol{\alpha}^\top\mathcal{K}\boldsymbol{\alpha}.$$

Obviously the following two equations hold

$$\nabla_{\boldsymbol{\alpha}}\mathcal{R}(\boldsymbol{\alpha}^\star, b^\star)=0,$$
$$\nabla_b\mathcal{R}(\boldsymbol{\alpha}^\star, b^\star)=0.$$

After simple computations, we obtain the following equation system,

$$\begin{cases}\dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\omega_i(1-y_i\mathcal{K}_i^\top\boldsymbol{\alpha}^\star-y_ib^\star)_+y_i\mathcal{K}_i-\lambda\mathcal{K}\boldsymbol{\alpha}^\star=0,\\ \displaystyle\sum_{i=1}^{m}\omega_i(y_i-\mathcal{K}_i^\top\boldsymbol{\alpha}^\star-b^\star)=0,\end{cases} \tag{2.3}$$

where for $i=1,\ldots,m$, the weight ω_i is given by

$$\omega_i=\frac{1}{2}\psi((1-y_i\mathcal{K}_i^\top\boldsymbol{\alpha}^\star-y_ib^\star)_+).$$

For $i = 1, \ldots, m$, let us denote ω_i^k as the weight at the kth iteration with

$$\omega_i^k = \frac{1}{2}\psi((1 - y_i \mathcal{K}_i^\top \boldsymbol{\alpha}^k - y_i b^k)_+).$$

As will be proved in theorem 2 (see section 4.3), the solution to the equation system 2.3 can be obtained by solving the following iteratively reweighted L2-SVM problem,

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^m, b\in\mathbb{R}} \frac{1}{m}\sum_{i=1}^{m} \omega_i^k (y_i - \mathcal{K}_i^\top \boldsymbol{\alpha} - b)_+^2 + \lambda \boldsymbol{\alpha}^\top \mathcal{K} \boldsymbol{\alpha},$$

where ω_i^k is as above and updated in each iteration. The proof of proposition 1 is completed.

As a result of proposition 1, we see that solving the minimization problem in RSVC can be carried out by solving an iteratively reweighted L2-SVM problem. As we will show, this observation can be useful since it directly brings us a computational algorithm for solving RSVC. In what follows, we restrict our discussions to the loss ℓ_σ in example 1 for convenience. However, most of our observations also apply to other margin-based classification losses that satisfy assumption 1 (e.g., the ϕ_σ loss in example 2). It should be mentioned that in the literature, the iteratively reweighted technique has been applied in classification problems. For instance, by assigning instance-wise weights in each iteration, the influence of outliers can be weakened, as done in Suykens, De Brabanter, Lukas, and Vandewalle (2002). Similar techniques have been also employed in Wu and Liu (2013).

3 Properties of the Proposed Classifier

In this section, we investigate the properties of RSVC that include the Fisher consistent property, generalization ability, and robustness property.

Note that in RSVC, we use a generic classification loss ϕ as a surrogate of the misclassification ϕ_0. A classification loss ϕ is said to be Fisher consistent (Lin, 2004; Zhang, 2004; Bartlett, Jordan, & McAuliffe, 2006; Steinwart, 2005) (also termed as *classification calibrated*) if the classifier obtained by using the surrogate loss ϕ preserves the sign of the Bayes rule (Lin, 2002). The generalization ability of a classifier refers to the ability that it can generalize on future observations, a key property when assessing a learning machine. By applying statistical learning arguments, we show that RSVC is Fisher consistent and its generalization bounds can be established. An important motivation of investigating RSVC lies in its robustness. In this section, we show that RSVC has some connections with M-estimation in statistics, and we then interpret its robustness from a weighted viewpoint.

3.1 Fisher Consistency and Generalization Ability. To study the Fisher consistency and the generalization ability of RSVC, we need to introduce several notations. We assume that $\mathbf{z}$ is drawn from an unknown probability distribution ρ on $\mathcal{X} \times \mathcal{Y}$. Let $f : \mathcal{X} \to \mathbb{R}$ be any measurable function and $\text{sgn}(f(x))$ as the function that takes the value of 1 if $f(x) \geq 0$ and -1 otherwise, for any $x \in \mathcal{X}$. When f is taken as a classifier, the misclassification error is given by

$$\mathcal{R}(\text{sgn}(f)) = \mathbb{E}\phi_0(Y, \text{sgn}(f(X))),$$

where the expectation is taken over the joint distribution ρ. Denoting $\mathcal{M}$ as the function set of measurable functions from $\mathcal{X}$ to $\mathbb{R}$, the function that minimizes the misclassification error is Bayes' rule, which is denoted as

$$f_c = \underset{f \in \mathcal{M}}{\text{argmin}}\, \mathcal{R}(\text{sgn}(f)).$$

Therefore, Bayes' rule is essentially the optimal classifier when the underlying distribution is known. When the surrogate loss function ℓ_σ is used, the optimal classifier f_ρ^σ over the function class $\mathcal{M}$ is given by

$$f_\rho^\sigma = \underset{f \in \mathcal{M}}{\text{argmin}} \int_{\mathcal{X} \times \mathcal{Y}} \ell_\sigma(y, f(x)) d\rho.$$

When the classifier f_ρ^σ preserves the sign of f_c, we say that the classification loss ℓ_σ is Fisher consistent. In fact, recall that ℓ_σ is a margin-based classification loss and satisfies that $\ell_\sigma(y, t) = \varphi(yt)$ for all $y \in \mathcal{Y}$, $t \in \mathbb{R}$, and $\varphi'(0)$ exists with $\varphi'(0) < 0$. Following conclusions drawn in Lin (2004), we know that ℓ_σ is Fisher consistent.

We now move on to investigate the generalization ability of RSVC. Quantitatively, let us denote $\mathcal{E}_\mathbf{z}(f_\mathbf{z})$ and $\mathcal{E}(f_\mathbf{z})$ as the empirical risk and expected risk, respectively, that are defined as

$$\mathcal{E}_\mathbf{z}(f_\mathbf{z}) = \frac{1}{m} \sum_{i=1}^{m} \ell_\sigma(y_i, f_\mathbf{z}(x_i)), \text{ and } \mathcal{E}(f_\mathbf{z}) = \int_{\mathcal{X} \times \mathcal{Y}} \ell_\sigma(y, f_\mathbf{z}(x)) d\rho.$$

Then the generalization ability of RSCV can be cast as the convergence of $\mathcal{E}_\mathbf{z}(f_\mathbf{z})$ to $\mathcal{E}(f_\mathbf{z})$ with $f_\mathbf{z}$ produced by equation 2.2 when the sample size m tends to infinity. Following a learning theory analysis, we obtain:

Theorem 1. *Let ℓ_σ be a margin-based classification loss given in example 1. Let $f_\mathbf{z}$ be produced by RSVC, equation 2.2, that is associated with ℓ_σ. Then for any*

$0 < \delta < 1$, with confidence $1 - \delta$, there holds

$$\mathcal{E}(f_z) - \mathcal{E}_z(f_z) \leq \frac{4\sigma}{\sqrt{m\lambda}} + \sqrt{\frac{8\,ln(1/\delta)}{m}}.$$

Theorem 1 can be proved by applying results in Mendelson (2003) and Bartlett and Mendelson (2003); we leave the proof to the appendix. Improved convergence rates may be derived by employing advanced learning theory techniques, such as data-dependent complexity measurements (Cucker & Zhou, 2007; Steinwart & Christmann, 2008).

The generalization bound established in theorem 1 indicates the learnability of RSVC when the parameters λ and σ are properly chosen. It should be noted that the generalization bound established in theorem 1 shows that it is dependent with the scale parameter σ. In fact, a refined analysis shows that the larger σ is, the sharper the generalization bound in theorem 1 will be. On the other hand, as we show in section 2.2, ℓ_σ approaches the squared hinge loss when σ is large enough. Therefore, the larger σ is, the less robustness RSVC possesses. Extended discussions concerning this are detailed in the following section.

3.2 Relating RSVC to M-Estimation. In the robust statistics literature, M-estimation refers to generalized maximum likelihood estimation, a general robust estimation method coined in Huber (1964). The M-estimator is usually defined to be a solution to a certain equation system obtained from the derivative of the likelihood objective function (Huber, 1964). Moreover, pursuing an M-estimator can be cast as finding a critical point of some objective functions.

More explicitly, an M-estimator of the parameter θ is the solution to the following minimization problem,

$$\min_{\theta} \sum_{i=1}^{m} L(e_i \mid \theta),$$

where e_i is residual and $L(\cdot)$ is a loss function that is nonnegative, symmetric, and nondecreasing. Frequently employed cost functions include least square loss and Huber's loss.

In this letter, the idea of investigating margin-based classification losses that satisfy assumption 1 comes from classical M-estimation. It is easy to see that the loss functions that satisfy assumption 1 can be taken as one-sided cost functions in M-estimation. Specifically, conditions 2 and 3 in assumption 1 ensure that penalization on the misclassified instances grows sufficiently slowly, akin to the redescending property of cost functions in redescending M-estimation in robust statistics (Andrews & Hampel, 2015).

To illustrate this, we consider the two loss functions ℓ_σ and ϕ_σ given in examples 1 and 2, respectively. In fact, the loss function ℓ_σ in example 1, a variant of which has been also investigated empirically in Singh, Pokharel, and Principe (2014), can be taken as a one-sided $\tilde{\ell}_\sigma$ loss where

$$\tilde{\ell}_\sigma(y,t) = \sigma^2(1 - \exp(-(y-t)^2/\sigma^2)), \ \ y \in \mathcal{Y}, t \in \mathbb{R}.$$

It is easy to see that the empirical risk minimization scheme based on the above $\tilde{\ell}_\sigma$ is an M-estimation. Some information-theoretic interpretation related to the loss function $\tilde{\ell}_\sigma$ can be found in Liu, Pokharel, and Principe (2007) and a learning theory analysis toward this loss is given in Feng, Huang, Shi, Yang, and Suykens (2015) recently. Regarding the ϕ_σ loss in example 2, it can be seen as a one-sided Cauchy loss $\tilde{\phi}_\sigma$ given by

$$\tilde{\phi}_\sigma(y,t) = \sigma^2 \log(1 + (1-yt)^2/\sigma^2), \ \ y \in \mathcal{Y}, t \in \mathbb{R}.$$

The $\tilde{\phi}_\sigma$ has been also applied to the compressed sensing (Suykens, Signoretto, & Argyriou, 2014) and tensor completion (Yang, Feng, & Suykens, in press) to enhance the robustness in estimation.

Besides the robustness property, another merit of M-estimation is that in most cases, an iteratively reweighted least squares algorithm can be performed to produce an M-estimator. As we show below, the proposed RSVC also inherits this property. Therefore, based on the above discussion, we see that RSVC can, in a sense, be taken as a one-sided M-estimation that is tailored for classification.

3.3 Robustness of RSVC from a Weighted Viewpoint. In section 3.2, we showed that RSVC has some connections with M-estimation in statistics and so may enjoy the robustness property. From the literature, we know that the robustness of a learning scheme can be quantitatively measured by using various robustness notions, such as influence function (Hampel, 1971), breakdown point (Donoho & Huber, 1983), and sensitivity curve (Hampel et al., 2011).

Note that RSVC is a kernel-based learning scheme, and solving RSVC amounts to finding a function in the reproducing kernel Hilbert space $\mathcal{H}_\mathcal{K}$. Within the kernel-based learning setup, the robustness property of learning machines has been investigated in Christmann and Steinwart (2004), Steinwart and Christmann (2008), Christmann, Van Messem, and Steinwart (2009), De Brabanter et al. (2009), and Debruyne, Christmann, Hubert, and Suykens (2010) for cases with convex loss functions. For instance, Christmann and Steinwart (2004) studied the robustness of kernel-based classification problem with a hinge loss. By introducing tools from functional analysis, they showed the robustness of learned classifier by proving the existence and boundedness of its influence function. Note, however, that

RSVC is nonconvex due to the use of a nonconvex loss function ϕ. In this case, there may be more than one local optimum of RSVC, and hence a quantitative robustness characterization is not easy to be obtained.

Instead of using quantitative robustness notions, we will show that RSVC also enjoys the robustness property from a weighted viewpoint by following steps in the proof of proposition 1. To this end, let us assume that the loss function in RSVC is the ℓ_σ loss given in example 1 and $(\hat{\boldsymbol{\alpha}}, \hat{b})$ is one of its stationary points. From the proof of proposition 1, we know that

$$\begin{cases} \dfrac{1}{m}\displaystyle\sum_{i=1}^{m} \mathcal{K}_i \exp\left(-\frac{(y_i - \mathcal{K}_i^\top \hat{\boldsymbol{\alpha}} - \hat{b})_+^2}{\sigma^2}\right)(y_i - \mathcal{K}_i^\top \hat{\boldsymbol{\alpha}} - \hat{b})_+ + \lambda \mathcal{K}\hat{\boldsymbol{\alpha}} = 0, \\ \displaystyle\sum_{i=1}^{m} (y_i - \mathcal{K}_i{}^\top \hat{\boldsymbol{\alpha}} - \hat{b})_+ = 0. \end{cases} \tag{3.1}$$

Here, again the solution to equation system 3.1 can be obtained by solving the following iteratively reweighted L2-SVM problem,

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^m, b\in\mathbb{R}} \frac{1}{m}\sum_{i=1}^{m} \exp(-(y_i - \mathcal{K}_i^\top \boldsymbol{\alpha}^k - b^k)_+^2/\sigma^2)(y_i - \mathcal{K}_i^\top \boldsymbol{\alpha} - b)_+^2 + \lambda \boldsymbol{\alpha}^\top \mathcal{K}\boldsymbol{\alpha},$$

where $(\boldsymbol{\alpha}^k, b^k)$ denotes the solution in the kth iteration and the quantity

$$\exp(-(y_i - \mathcal{K}_i^\top \boldsymbol{\alpha}^k - b^k)_+^2/\sigma^2), \text{ for } i = 1, \ldots, m,$$

stands for the weight that is updated in each iteration.

To see the robustness of RSVC, again we denote $\omega_i = \exp(-(y_i - \mathcal{K}_i^\top \hat{\boldsymbol{\alpha}} - \hat{b})_+^2/\sigma^2)$ for $i = 1, \ldots, m$ and $\hat{f}_{\mathbf{z}}(x_i) = \mathcal{K}_i^\top \hat{\boldsymbol{\alpha}} + \hat{b}$ as the obtained classifier from RSVC. Let us now consider misclassified instances: $\{x_i : y_i \hat{f}_{\mathbf{z}}(x_i) < 0\}$. The magnitude $|\hat{f}_{\mathbf{z}}(x_i)|$ can be interpreted as the extent that the learned label of x_i deviates from its input label y_i. The larger $|\hat{f}_{\mathbf{z}}(x_i)|$ is, the more likely that the observed instance pair (x_i, y_i) tends to be an outlier. However, from equation 3.1, we see that the value of ω_i decreases with an increase of $|\hat{f}_{\mathbf{z}}(x_i)|$ for misclassified instance x_i. That is, ℓ_σ can downweight the influence of instances that are far away from their labels. This explains the robustness of RSVC from a weighted viewpoint.

4 Computational Algorithm and Convergence Analysis

In this section, we are concerned with the computational aspects of RSVC.

Algorithm 1: Iteratively Reweighted Algorithm for Solving RSVC, **Equation** 2.2.

Input: data $\{(x_i, y_i)\}_{i=1}^m$, kernel matrix $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{m\times m}$, regularization parameter $\lambda > 0$, scale parameter $\sigma > 0$ and the initial guess $\boldsymbol{\alpha}^0 \in \mathbb{R}^m$, $b^0 \in \mathbb{R}$.
Output: the learned coefficient $\boldsymbol{\alpha}^{k+1} = (\alpha_1^{k+1}, \ldots, \alpha_m^{k+1})^\top$ and $b^{k+1} \in \mathbb{R}$.
while the stopping criterion is not satisfied **do**

- Compute $\boldsymbol{\alpha}^{k+1}$ and b^{k+1} by solving the following weighted L2-SVM problem:

$$(\boldsymbol{\alpha}^{k+1}, b^{k+1}) = \underset{\boldsymbol{\alpha}\in\mathbb{R}^m, b\in\mathbb{R}}{\operatorname{argmin}} \sum_{i=1}^{m} \omega_i^{k+1}(y_i - \boldsymbol{\mathcal{K}}_i^\top \boldsymbol{\alpha} - b)_+^2 + \lambda \boldsymbol{\alpha}^\top \boldsymbol{\mathcal{K}} \boldsymbol{\alpha},$$

where

$$\omega_i^{k+1} = \exp(-(y_i - \boldsymbol{\mathcal{K}}_i^\top \boldsymbol{\alpha}^k - b^k)_+^2/\sigma^2),\ i = 1, \ldots, m.$$

- Set $k := k + 1$.

end while

4.1 An Iteratively Reweighted Algorithm. From previous sections, we know that to solve RSVC, one can solve an iteratively reweighted L2-SVM. Therefore, the algorithm we propose is an iteratively reweighted one given in algorithm 1.

The convergence analysis of algorithm 1 will be provided in section 4.3. The nice aspect of the reduction from RSVC to an iteratively reweighted L2-SVM lies in the fact that in each iteration, the weighted L2-SVM subproblem is convex and can be efficiently implemented. Moreover, as shown in section 4.2, the dual of weighted L2-SVM is a quadratic programming and can be solved optimally with various off-the-shelf software packages, including Matlab quadprog and CVX (Grant & Boyd, 2014, 2008), as well as SMO-type algorithms (Platt, 1999). It should be also remarked that the proposed algorithm 1 is a direct benefit of the relation between RSVC and L2-SVM, as indicated in proposition 1. As an iteratively reweighted algorithm, it can be employed to interpret the robustness of RSVC as it downweights the influence of outliers in each iteration.

In algorithm 1, it is reduced to a quadratic programming in each iteration. However, thanks to the smoothness of RSVC, one may apply other first-order optimization algorithms to solve this problem in the primal as well as in the dual. It should be noted that algorithm 1 is proposed to solve the kernel-based RSVC, equation 2.2. In fact, when a linear RSVC, equation 2.1, is of interest, one may also solve RSVC easily in the primal by using many conventional algorithms (e.g., gradient descent), benefiting from its smoothness. In this case, one may consider the feature size and instance size of the observations to decide whether RSVC should be solved in the primal or the dual. However, this is not always the case when the primal problem is nonsmooth.

4.2 Dual Formulation of Weighted L2-SVM. Weighted L2-SVM is solvable via quadratic programming. Therefore, to illustrate this point, we derive the dual formula of the weighted L2-SVM in this section.

Let $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m)^\top$ be a fixed weight vector. The primal of weighted L2-SVM can be expressed as

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \mathcal{J}_P(\boldsymbol{w}, \boldsymbol{\xi}) = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \frac{C}{2}\sum_{k=1}^{m} \mu_k \xi_k^2$$
$$\text{such that} \qquad y_k(\boldsymbol{w}^\top \boldsymbol{\phi}(x_k) + b) \geq 1 - \xi_k, \quad k = 1, \ldots, m, \tag{4.1}$$

where $C := \frac{1}{m\lambda}$, $\boldsymbol{\phi}(x)$ denotes the implicit feature map of x via a Mercer kernel $\mathcal{K}$ with $\mathcal{K}(x, x') = \langle \boldsymbol{\phi}(x), \boldsymbol{\phi}(x') \rangle$, for $x, x' \in \mathcal{X}$ and the classifier is assumed to take the form $y(x) = \operatorname{sgn}(\boldsymbol{w}^\top \boldsymbol{\phi}(x) + b)$ in the primal space.

Proposition 2. *Let the primal formulation of weighted L2-SVM be given in equation 4.1. The dual formulation of weighted L2-SVM can be written as*

$$\max_{\boldsymbol{\alpha}} \sum_{k=1}^{m} \alpha_k - \frac{1}{2}\sum_{k,l=1}^{m} \alpha_k \alpha_l y_k y_l \left(\mathcal{K}(x_k, x_l) + \frac{\delta_{kl}}{C\mu_k} \right)$$
$$\text{subject to} \qquad \sum_{k=1}^{m} \alpha_k y_k = 0, \quad \alpha_k \geq 0, \quad k = 1, \ldots, m, \tag{4.2}$$

where δ_{kl} is the Kronecker's delta function, which takes the value 1 for $k = l$ and the value 0 otherwise. Moreover, the offset b can be determined by

$$y_k \left(\sum_{l=1}^{m} \alpha_l y_l \left(\mathcal{K}(x_k, x_l) + \frac{\delta_{kl}}{C\mu_k} \right) + b \right) - 1 = 0, \quad \text{when } \alpha_k > 0,$$
$$\text{for } k = 1, \ldots, m.$$

Proof. Recalling the primal formula of weighted L2-SVM in equation 4.1 and introducing Lagrange multipliers $\alpha_k \geq 0$, $\nu_k \geq 0$ for $k = 1, \ldots, m$, the Lagrangian of equation 4.1 takes the following form:

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\nu}) = \mathcal{J}_P(\boldsymbol{w}, \boldsymbol{\xi}) - \sum_{k=1}^{m} \alpha_k (y_k(\boldsymbol{w}^\top \boldsymbol{\phi}(x_k) + b) - 1 + \xi_k). \tag{4.3}$$

The solution of weighted L2-SVM is

$$\max_{\boldsymbol{\alpha},\boldsymbol{\nu}} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\nu}).$$

The KKT conditions yield

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0 \quad \Rightarrow \quad \boldsymbol{w} = \displaystyle\sum_{l=1}^{m} \alpha_l y_l \boldsymbol{\phi}(x_l), \\ \dfrac{\partial \mathcal{L}}{\partial \xi_k} = 0 \quad \Rightarrow \quad C\mu_k \xi_k - \alpha_k = 0, \\ \dfrac{\partial \mathcal{L}}{\partial b} = 0 \quad \Rightarrow \quad \displaystyle\sum_{l=1}^{m} \alpha_l y_l = 0, \\ \quad \alpha_k (y_k (\boldsymbol{w}^\top \boldsymbol{\phi}(x_k) + b) - 1 + \xi_k) = 0, \quad k = 1, \ldots, m, \\ \quad y_k (\boldsymbol{w}^\top \boldsymbol{\phi}(x_k) + b) - 1 + \xi_k \geq 0, \qquad k = 1, \ldots, m, \\ \quad \alpha_k \geq 0, \qquad\qquad\qquad\qquad k = 1, \ldots, m. \end{cases}$$

Substituting them into formula 4.3, we obtain the following dual form of weighted L2-SVM:

$$\max_{\boldsymbol{\alpha}, \boldsymbol{v}} \sum_{k=1}^{m} \alpha_k - \frac{1}{2} \sum_{k,l=1}^{m} \alpha_k \alpha_l y_k y_l \mathcal{K}(x_k, x_l) - \frac{1}{2C} \sum_{k=1}^{m} \frac{\alpha_k^2}{\mu_k},$$

$$\text{such that} \quad \sum_{k=1}^{m} \alpha_k y_k = 0, \ \alpha_k \geq 0, \ k = 1, \ldots, m.$$

Introducing Kronecker's delta function, we obtain the desired dual formula, equation 4.2, for weighted L2-SVM. Concerning the offset b, from KKT conditions we know that

$$y_k (\boldsymbol{w}^\top \boldsymbol{\phi}(x_k) + b) - 1 + \xi_k = 0, \quad \text{when } \alpha_k > 0, \text{ for } k = 1, \ldots, m.$$

Therefore, the offset b can be computed by using the above equation for any training data point with $\alpha_k > 0$. This completes the proof of proposition 2.

4.3 Convergence Analysis. In this section, we provide the convergence analysis of algorithm 1, which is motivated by the idea in half-quadratic minimization methods (Geman & Yang, 1995; Nikolova & Ng, 2005). However, we note that the analysis concerning the convergence of the half-quadratic minimization methods cannot be tailored to our case due to the noneven property of the loss function ℓ_σ. We therefore introduce the following auxiliary lemma:

Lemma 1. *Let* $h(t) = \sigma^2(1 - exp(-(1-t)_+^2)/\sigma^2)$. *Then* h *can be expressed as*

$$h(t) = \inf_{\omega \in \mathbb{R}_+} \omega(1-t)_+^2 + \sigma^2 \varrho(\omega), \tag{4.4}$$

where

$$\varrho(\omega) = \begin{cases} 1, & \omega = 0, \\ 1 - \omega + \omega \log \omega, & 0 < \omega \leq 1, \\ 0, & \omega > 1. \end{cases} \tag{4.5}$$

Moreover, if we denote

$$\omega^\star = \operatorname*{argmin}_{\omega \in \mathbb{R}_+} \omega(1-t)_+^2 + \sigma^2 \varrho(\omega),$$

we then have

$$\omega^\star = exp(-(1-t)_+^2)/\sigma^2.$$

Proof. We note first that the continuous function ϱ is convex. To see this, we compute the first derivative of ϱ:

$$\varrho'(\omega) = \begin{cases} \ln \omega, & 0 < \omega \leq 1, \\ 0, & \omega > 1. \end{cases}$$

The nondecreasing property of ϱ' on the interval $(0, +\infty)$ reveals the convexity of ϱ.

To verify equation 4.4, we first consider the case when $t \in (-\infty, 1)$ when we have $h(t) = \sigma^2(1 - \exp(-(1-t)^2/\sigma^2))$. The minimum of the right-hand side of equation 4.4 must occur either at a stationary point ω_0 of $g(\omega)$ with

$$g(\omega) = \omega(1-t)^2 + \sigma^2 \varrho(\omega),$$

or at $\omega = 0$. With simple computations, we see that

$$g'(\omega) = \begin{cases} (1-t)^2 + \sigma^2 \ln \omega, & 0 < \omega \leq 1, \\ (1-t)^2, & \omega > 1. \end{cases}$$

As a result, $g'(\omega_0) = 0$ if and only if $\ln \omega_0 = -(1-t)^2/\sigma^2$, for any $t < 1$. As a result, we obtain

$$\omega_0 = \exp(-(1-t)^2/\sigma^2), \text{ for any } t < 1.$$

Moreover, from the definition of g, we know that

$$g(\omega_0) = \sigma^2(1 - \exp(-(1-t)^2)/\sigma^2) \text{ and } g(0) = \sigma^2.$$

Consequently, when $t \in (-\infty, 1)$, there holds

$$\inf_{\omega \in \mathbb{R}_+} g(\omega) = \sigma^2(1 - \exp(-(1-t)^2)/\sigma^2),$$

and the minimum is achieved at $\omega = \omega_0$.

Now let us discuss the case when $t \in [1, +\infty)$. From the definition of ϕ, we know that $h(t) = 0$. In fact, it is easy to see that $\inf_{\omega \geq 0} g(\omega) = 0$ at $\omega = 1$, and consequently we have verified equation 4.4 for $t \in [1, +\infty)$.

From the above discussions, we see that

$$\omega^\star = \exp(-(1-t)_+^2)/\sigma^2.$$

Recall that the representer theorem ensures that the minimization problem, equation 2.2, can be reduced to the following finite-dimensional minimization problem:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^m, b \in \mathbb{R}} \frac{1}{m} \sum_{i=1}^m \sigma^2(1 - \exp(-(y_i - \mathcal{K}_i^\top \boldsymbol{\alpha} - b)_+^2/\sigma^2)) + \lambda \boldsymbol{\alpha}^\top \mathcal{K} \boldsymbol{\alpha}.$$

From lemma 1, we know that the above minimization problem can be further rewritten as

$$\min_{\boldsymbol{\omega} \in \mathbb{R}_+^m, \boldsymbol{\alpha} \in \mathbb{R}^m, b \in \mathbb{R}} \frac{1}{m} \sum_{i=1}^m \omega_i (y_i - \mathcal{K}_i^\top \boldsymbol{\alpha} - b)_+^2 + \frac{\sigma^2}{m} \sum_{i=1}^m \varrho(\omega_i) + \lambda \boldsymbol{\alpha}^\top \mathcal{K} \boldsymbol{\alpha},$$

where the function ϱ is defined in equation 4.5. This observation enables us to prove the convergence of algorithm 1.

Theorem 2. *Let $\{(\boldsymbol{\alpha}^k, b^k)\}_{k \geq 1}$ be the sequence generated in algorithm 1. Then every limit point of $\{(\boldsymbol{\alpha}^k, b^k)\}_{k \geq 1}$ must be a stationary point of RSVC.*

Proof. For notation simplification, we denote

$$\mathcal{Q}(\boldsymbol{\alpha}, b, \boldsymbol{\omega}) = \frac{1}{m} \sum_{i=1}^m \omega_i (y_i - \mathcal{K}_i^\top \boldsymbol{\alpha} - b)_+^2 + \frac{\sigma^2}{m} \sum_{i=1}^m \varrho(\omega_i) + \lambda \boldsymbol{\alpha}^\top \mathcal{K} \boldsymbol{\alpha}.$$

From algorithm 1 and the above discussions, we know that

$$\boldsymbol{\omega}^{k+1} = \underset{\boldsymbol{\omega}\in\mathbb{R}^m_+}{\operatorname{argmin}} \mathcal{Q}(\boldsymbol{\alpha}^k, b^k, \boldsymbol{\omega}) \quad \text{and} \quad (\boldsymbol{\alpha}^{k+1}, b^{k+1}) = \underset{\boldsymbol{\alpha}\in\mathbb{R}^m, b\in\mathbb{R}}{\operatorname{argmin}} \mathcal{Q}(\boldsymbol{\alpha}, b, \boldsymbol{\omega}^{k+1}).$$

Due to the positive definiteness of $\mathcal{K}$, we know that $\mathcal{Q}(\boldsymbol{\alpha}, b, \boldsymbol{\omega})$ is coercive with respect to $\boldsymbol{\alpha}$. It is easy to see that, $\mathcal{Q}(\boldsymbol{\alpha}, b, \boldsymbol{\omega})$ is also coercive with respect to b. Therefore, the sequences $\{\boldsymbol{\alpha}^k\}_{k\geq 1}$ and $\{b^k\}_{k\geq 1}$ are bounded. Recalling that $\boldsymbol{\omega}^{k+1} = (\omega_1^{k+1}, \ldots, \omega_m^{k+1})^\top$ with

$$\begin{aligned}\omega_i^{k+1} &= \underset{\omega\in\mathbb{R}_+}{\operatorname{argmin}}\, \omega(y_i - \mathcal{K}_i^\top \boldsymbol{\alpha}^k - b^k)_+^2 + \sigma^2\varrho(\omega) \\ &= \exp(-(y_i - \mathcal{K}_i^\top \boldsymbol{\alpha}^k - b^k)_+^2/\sigma^2), \quad i = 1, \ldots, m,\end{aligned}$$

we also see the boundedness of $\{\boldsymbol{\omega}^k\}_{k\geq 1}$. As a result, the sequence $\{(\boldsymbol{\alpha}^k, b^k, \boldsymbol{\omega}^k)\}_{k\geq 1}$ has limit points.

Suppose that $\{(\boldsymbol{\alpha}^{k_l}, b^{k_l}, \boldsymbol{\omega}^{k_l})\}_{l\geq 1}$ is a sub-sequence that converges to $(\boldsymbol{\alpha}^\star, b^\star, \boldsymbol{\omega}^\star)$ as $l \to \infty$. We further denote

$$(\boldsymbol{\alpha}_{\boldsymbol{\omega}^\star}, b_{\boldsymbol{\omega}^\star}) = \underset{\boldsymbol{\alpha}\in\mathbb{R}^m, b\in\mathbb{R}}{\operatorname{argmin}} \mathcal{Q}(\boldsymbol{\alpha}, b, \boldsymbol{\omega}^\star), \text{ and } \boldsymbol{\omega}_{\boldsymbol{\alpha}^\star} = \underset{\boldsymbol{\omega}\in\mathbb{R}^m_+}{\operatorname{argmin}} \mathcal{Q}(\boldsymbol{\alpha}^\star, b^\star, \boldsymbol{\omega}).$$

This, in connection with the definitions of $\boldsymbol{\omega}^{k_l+1}$, $\boldsymbol{\alpha}^{k_l+1}$, and b^{k_l+1}, implies

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\alpha}^{k_l}, b^{k_l}, \boldsymbol{\omega}_{\boldsymbol{\alpha}^\star}) &\geq \mathcal{Q}(\boldsymbol{\alpha}^{k_l}, b^{k_l}, \boldsymbol{\omega}^{k_l+1}) \geq \mathcal{Q}(\boldsymbol{\alpha}^{k_l+1}, b^{k_l+1}, \boldsymbol{\omega}^{k_l+1}) \\ &\geq \mathcal{Q}(\boldsymbol{\alpha}^{k_{l+1}}, b^{k_{l+1}}, \boldsymbol{\omega}^{k_{l+1}}).\end{aligned}$$

The definition of $(\boldsymbol{\alpha}^{k_l}, b^{k_l})$ also tells us that there holds

$$\mathcal{Q}(\boldsymbol{\alpha}_{\boldsymbol{\omega}^\star}, b_{\boldsymbol{\omega}^\star}, \boldsymbol{\omega}^{k_l}) \geq \mathcal{Q}(\boldsymbol{\alpha}^{k_l}, b^{k_l}, \boldsymbol{\omega}^{k_l}).$$

By letting $l \to +\infty$ in the above two inequalities, we see that

$$\mathcal{Q}(\boldsymbol{\alpha}_{\boldsymbol{\omega}^\star}, b_{\boldsymbol{\omega}^\star}, \boldsymbol{\omega}^\star) \geq \mathcal{Q}(\boldsymbol{\alpha}^\star, b^\star, \boldsymbol{\omega}^\star) \quad \text{and} \quad \mathcal{Q}(\boldsymbol{\alpha}^\star, b^\star, \boldsymbol{\omega}_{\boldsymbol{\alpha}^\star}) \geq \mathcal{Q}(\boldsymbol{\alpha}^\star, b^\star, \boldsymbol{\omega}^\star).$$

From the definitions of $\boldsymbol{\alpha}_{\boldsymbol{\omega}^\star}$ and $\boldsymbol{\omega}_{\boldsymbol{\alpha}^\star}$, we have

$$(\boldsymbol{\alpha}^\star, b^\star) = \underset{\boldsymbol{\alpha}\in\mathbb{R}^m, b\in\mathbb{R}}{\operatorname{argmin}} \mathcal{Q}(\boldsymbol{\alpha}, b, \boldsymbol{\omega}^\star) \quad \text{and} \quad \boldsymbol{\omega}^\star = \underset{\boldsymbol{\omega}\in\mathbb{R}^m_+}{\operatorname{argmin}} \mathcal{Q}(\boldsymbol{\alpha}^\star, b^\star, \boldsymbol{\omega}).$$

Particularly, the fact that $(\boldsymbol{\alpha}^\star, b^\star) = \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^m, b\in\mathbb{R}} \mathcal{Q}(\boldsymbol{\alpha}, b, \boldsymbol{\omega}^\star)$ also implies

$$\nabla_{\boldsymbol{\alpha}} \mathcal{Q}(\boldsymbol{\alpha}^\star, b^\star, \boldsymbol{\omega}^\star) = 0, \text{ and } \nabla_b \mathcal{Q}(\boldsymbol{\alpha}^\star, b^\star, \boldsymbol{\omega}^\star) = 0,$$

which can be equivalently expressed as

$$\sum_{i=1}^{m} \mathcal{K}_i \exp(-(y_i - \mathcal{K}_i^\top \boldsymbol{\alpha}^\star - b^\star)_+^2/\sigma^2)(y_i - \mathcal{K}_i^\top \boldsymbol{\alpha}^\star - b^\star)_+ + \lambda \mathcal{K}\boldsymbol{\alpha}^\star = 0,$$

and

$$\sum_{i=1}^{m} \left(y_i - \mathcal{K}_i^\top \boldsymbol{\alpha}^\star - b^\star\right)_+ = 0.$$

This verifies that $(\boldsymbol{\alpha}^\star, b^\star)$ is a stationary point of RSVC. Thus, we have accomplished the proof of theorem 2.

Note that the above convergence analysis on algorithm 1 is conducted with respect to the loss function ℓ_σ and relies on lemma 1. We remark that when the loss function ϕ_σ in example 2 is employed, its convergence to a stationary point can be also proved via a similar auxiliary lemma given in section A.2.

5 Experimental Results

We present more experimental results in this section to show the effectiveness of RSVC by applying algorithm 1. In our experiments, each subproblem in algorithm 1 is a weighted L2-SVM, and we use quadprog function of Matlab to solve this quadratic programming problem in its dual. The solution to L2-SVM is chosen as the initial guess of algorithm 1. The stopping criterion is $\|(\boldsymbol{\alpha}^{k+1}, b^{k+1}) - (\boldsymbol{\alpha}^k, b^k)\|_2 \le 10^{-4}$. The maximum iteration number is set to 100. All numerical computations are implemented on an Intel i7-3770 CPU desktop computer with 16 GB of RAM. The supporting software is Matlab R2013a.

5.1 An Illustrative Example. As shown in algorithm 1, solving RSVC is carried out by solving an iteratively reweighted L2-SVM. In this section, we implement an artificial simulation by using the Two Moons data set again to illustrate this iteration process. This two-dimensional data set contains a training set and a test set, each of size 200 and binary labeled. Its plot is given in the left panel of Figure 1, which is scaled to $[0, 1] \times [0, 1]$. To compare L2-SVM and RSVC, we randomly flip 10% of the labels of the data set and use the test set for validation. The plot of this contaminated data set is given in the right panel of Figure 1. In this experiment, the gaussian

kernel $\mathcal{K}(x, x') = \exp(-\frac{\|x-x'\|^2}{2h^2})$ and the loss function ℓ_σ are used. Three parameters—the scale parameter σ, the regularization parameter C, and the bandwidth of the gaussian kernel h—are tuned on the test set.

We then train two classifiers with this contaminated data set by applying L2-SVM and RSVC, respectively. To see the influence of the weights that are applied to outliers, we plot the classifier obtained after each iteration in Figure 5. The red dotted curve in each panel presents the classifier obtained from RSVC at the kth step, where the number k is marked at the bottom-left corner of each panel. The loosely dashed black curve in each panel stands for the classifier trained by L2-SVM with the uncontaminated Two Moons data set, which is taken as the ground truth.

From Figure 5, it can be observed that the influence of the outliers on the output classifier is weakened after each iteration. Moreover, the algorithm converges fast since the obtained classifier after the ninth iteration is very close to the ground truth as shown in the bottom-right panel.

5.2 Experiments on UCI Data Sets. We now validate the effectiveness of RSVC on UCI data sets (Bache & Lichman, 2013): Monk's Problem Data Set (Monks1, Monks2, and Monks3), Spect Heart Data Set (Spect), Haberman's Survival Data Set (Haber), Breast Cancer Wisconsin Dataset (Breast), Pima Indians Diabetes Data Set (Pima), and Ionosphere Data Set (Ionosphere).

In our experiments, we conduct empirical comparisons on the classification accuracy and computation time between RSVC and L2-SVM considering that RSVC can be taken as robustified L2-SVM. We also carry out numerical comparisons between the classifier obtained from RSVC and that trained from robust but nonsmooth support vector machines. A well-known example is the truncated hinge loss–based support vector classifier (TSVC) proposed in Wu and Liu (2007), which is solved by using the concave-convex procedure (CCCP) proposed in Collobert et al. (2006).

In our experiments, the variables in each data set are scaled into the interval $[0, 1]$. For each data set, we randomly split it into training, tuning, and test sets, each of which consists of 60%, 20%, and 20% instances of the data set, respectively.[1] As in the toy example, we use the gaussian kernel and the loss function ℓ_σ. The parameters C, h, and σ are tuned on the tuning set. To show the robustness of RSVC, we randomly flip the labels of the training set with different levels: 0%, 10%, and 20%. The classification accuracy on test set that is averaged over 20 repetitions is reported in Table 1. We also record the averaged computation time of RSVC and TSVC over 20 repetitions in Table 2.

From experimental results in Tables 1 and 2, we make the following observations:

[1]The Monk's Problem data set and the Spect Heart data set consist of a training set and a test set. In our experiment, we merge the two sets before splitting.

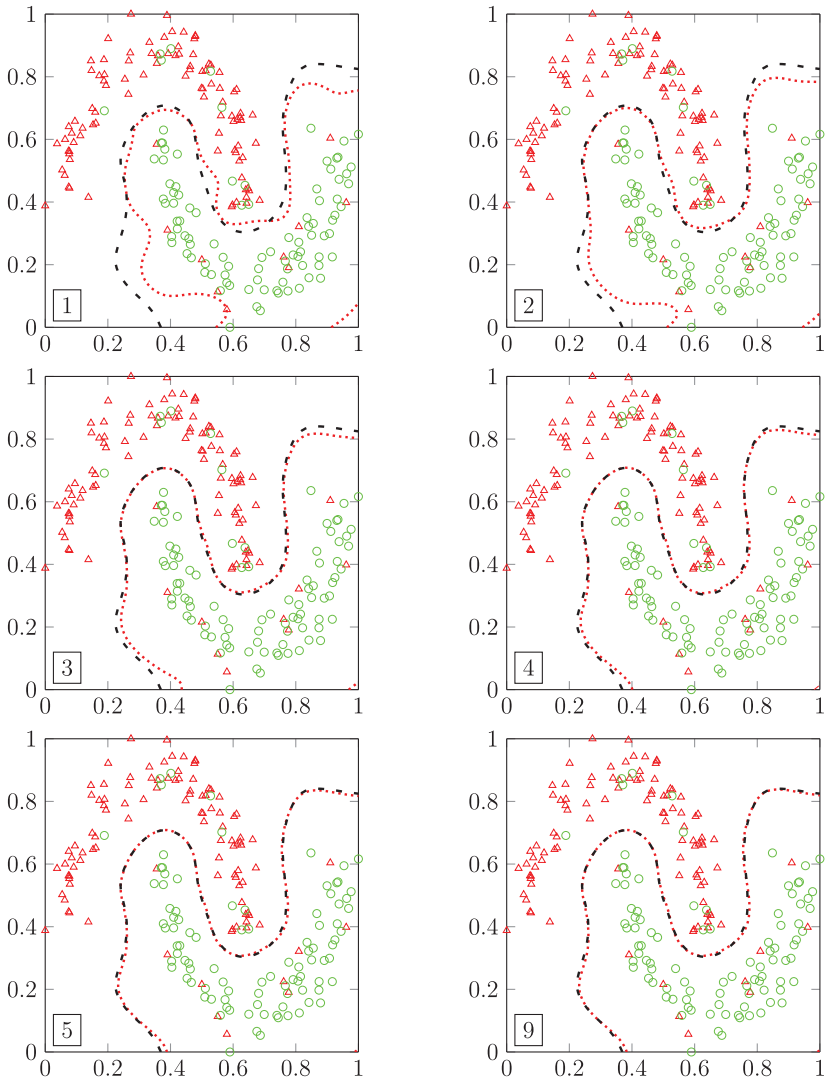

Figure 5: Plots of the classifiers trained by RSVC in each iteration. For each panel, the red dotted curve presents the classifier obtained by solving RSVC after the kth iteration, where the number k is marked at the bottom-left corner of each panel. The loosely dashed black curve in each panel is the classifier trained by L2-SVM on the uncontaminated Two Moons data set and is plotted for comparison.

Table 1: Classification Accuracy on Test Data of UCI Data Sets (%).

	0%			10%			20%		
	L2-SVM	RSVC	TSVC	L2-SVM	RSVC	TSVC	L2-SVM	RSVC	TSVC
Monks1	**95.80**	95.70	91.80	83.50	**89.95**	88.95	78.65	**88.15**	82.70
Monks2	90.17	89.83	**90.25**	81.42	81.83	**82.13**	75.00	**76.21**	74.17
Monks3	95.14	**95.68**	95.50	95.18	**97.09**	96.73	91.45	**95.14**	95.00
Spect	83.43	**83.66**	82.91	82.76	**83.51**	83.51	83.28	**83.58**	83.06
Haber	70.15	**70.45**	69.92	70.15	**70.76**	70.53	72.20	**73.64**	72.88
Breast	**97.10**	96.85	96.81	95.76	**96.39**	96.05	94.92	**95.80**	95.63
Pima	**75.65**	75.45	75.58	75.39	**75.45**	74.12	75.58	75.75	**75.88**
Ionosphere	**94.20**	93.95	93.00	91.25	91.65	**92.70**	91.10	91.30	**91.90**

Note: The best results from the compared methods are in bold.

Table 2: Computation Time on Test Data of UCI Data Sets (in seconds).

	0%		10%		20%	
	RSVC	TSVC	RSVC	TSVC	RSVC	TSVC
Monks1	0.5637	**0.3859**	**0.5135**	0.6032	**0.4912**	0.7329
Monks2	0.3708	**0.1649**	0.6430	**0.3870**	**0.4619**	0.4876
Monks3	0.3444	**0.3112**	**0.3124**	0.4499	**0.6114**	0.8149
Spect	**0.0672**	0.1043	**0.0824**	0.1004	**0.0396**	0.0739
Haber	**0.0984**	0.5695	**0.0798**	0.8318	**0.1003**	0.5608
Breast	**0.2864**	0.3382	**0.1420**	0.3936	**0.1650**	0.4530
Pima	**0.3782**	2.5918	**0.5136**	2.7937	**0.4638**	2.7533
Ionosphere	0.1023	**0.0812**	**0.0642**	0.0930	**0.0901**	0.1116

Note: The best results from the compared methods are in bold.

- Concerning classification accuracy, we see from Table 1 that in the absence of flipped labels, the three classifiers L2-SVM, RSVC, and TSVC perform comparably. More precisely, L2-SVM and RSVC give slightly more accurate classification results than TSVC.
- Concerning classification accuracy, Table 1 tells us that in the presence of flipped labels, RSVC always outperforms L2-SVM. We also see that RSVC can give more accurate classification results than TSVC.
- Concerning computation time, Table 2 shows that in the absence of flipped labels, the computational costs of RSVC and TSVC are comparable, although RSVC performs far better than TSVC for some specific cases.
- Concerning computation time, Table 2 also tells us that in the presence of flipped labels, RSVC can outperform TSVC. Moreover, for some cases, the computational cost of RSVC can be significantly lower than that of TSVC, especially when the size of the training data gets large, according to our experimental experience.

The above observations drawn from the experimental results indicate that:

- RSVC is more robust than L2-SVM and also performs comparably to L2-SVM in the absence of flipped labels. In this respect, RSVC can be a better choice if robustness is a concern in classification problems over L2-SVM;
- RSVC can outperform the well-known TSVC in terms of classification accuracy and computation time and could be a better choice, especially when computational cost is an important concern.

Here we remark that in comparison to L2-SVM, the enhancement of the robustness of RSVC is at the expense of computational cost since we solve RSVC by solving a reweighted L2-SVM iteratively. However, as shown in Figure 5, algorithm 1 converges within only a few iterations, so the computation time for solving RSVC is still acceptable.

6 Conclusion

In this letter, we studied the large margin classification problem in the presence of label noise. This purpose was achieved by introducing new robust classification losses. Besides the robustness, another property of the proposed method is that its associated optimization problem is smooth, which again benefits from the use of smooth losses. Studies were then conducted on the proposed robust classifier following two lines. First, we examined its Fisher consistency property and generalization ability, and we also interpreted its robustness from a weighted viewpoint. Second, we were also concerned with its computational aspects. We showed that the proposed classifier had some close connections with L2-SVM. For instance, it can be obtained by solving iteratively reweighted L2-SVM problems. An iteratively reweighted algorithm was then proposed and its convergence analysis was provided. Simulation studies with toy examples and real data sets suggest that in comparison to L2-SVM, the proposed method can be a better choice in real-world applications, especially when robustness is a major concern.

Appendix A

A.1 Proof of Theorem 1. To prove Theorem 1, we introduce the notion of Rademacher complexity, which measures the complexity of a function class.

Definition 2 (Rademacher Complexity). *Let $\rho_{\mathcal{X}}$ be the marginal distribution of ρ on $\mathcal{X}$. Let $\boldsymbol{X} := \{x_1, \ldots, x_n\}$ be drawn i.i.d. from $\rho_{\mathcal{X}}$ and $\mathcal{F}$ be a class of*

functions mapping from $\mathcal{X}$ to $\mathbb{R}$. Then the empirical Rademacher complexity of $\mathcal{F}$ is defined as

$$\hat{R}_n(\mathcal{F}) := \mathbb{E}\left[\sup_{f\in\mathcal{F}}\left|\frac{2}{n}\sum_{i=1}^{n}\varepsilon_i f(x_i)\right| \middle| \boldsymbol{X}\right],$$

where $\varepsilon_1, \ldots, \varepsilon_n$ are independent uniform $\{\pm 1\}$-valued random variables. Then the Rademacher complexity of $\mathcal{F}$ is $R_n(\mathcal{F}) = \mathbb{E}\hat{R}_n(\mathcal{F})$.

Proof of Theorem 1. Recalling that $f_{\mathbf{z}}$ is produced by equation 2.2 with the regularization parameter λ, we have

$$\lambda \|f_{\mathbf{z}}^{\phi}\|_{\mathcal{K}}^2 \leq \sigma^2,$$

which yields $\|f_{\mathbf{z}}^{\phi}\|_{\mathcal{K}} \leq \sigma/\sqrt{\lambda}$. On the other hand, due to the Lipschitz continuity property of ϕ, we can apply theorem 8 in Bartlett and Mendelson (2003) and see that for any $0 < \delta < 1$, there holds

$$\mathcal{E}(f_{\mathbf{z}}^{\phi}) - \mathcal{E}_{\mathbf{z}}(f_{\mathbf{z}}^{\phi}) \leq R_m(\mathcal{H}) + \sqrt{8\ln(1/\delta)/m},$$

where the function set $\mathcal{H}$ is defined as

$$\mathcal{H} := \{h \mid h(x,y) := \phi(yf(x)) - \phi(0),\ f \in \mathcal{H}_{\mathcal{K}},\ \|f\|_{\mathcal{K}} \leq \sigma/\sqrt{\lambda},\ (x,y) \in \mathcal{X}\times\mathcal{Y}\}.$$

Again, the Lipschitz continuity of ϕ, together with theorem 12 in Bartlett and Mendelson (2003) yields that

$$R_m(\mathcal{H}) \leq 2R_m(\mathcal{F}_\lambda) \quad \text{with}$$
$$\mathcal{F}_\lambda = \left\{ f \,\middle|\, f \in \mathcal{H}_{\mathcal{K}},\ \|f\|_{\mathcal{K}} \leq \sigma\sqrt{\log(1+\sigma^{-2})/\lambda} \right\}.$$

It follows from Mendelson (2003) that

$$R_m(\mathcal{F}_\lambda) \leq \frac{2\sigma}{\sqrt{m\lambda}}.$$

Combining the above estimates, we see that for any $0 < \delta < 1$, there holds

$$\mathcal{E}(f_{\mathbf{z}}^{\phi}) - \mathcal{E}_{\mathbf{z}}(f_{\mathbf{z}}^{\phi}) \leq \frac{4\sigma}{\sqrt{m\lambda}} + \sqrt{\frac{8\ln(1/\delta)}{m}}.$$

A.2 An Auxiliary Lemma. The following lemma can be proved analogous to lemma 1:

Lemma 2. *Let* $h(t) = \sigma^2 log(1 + (1-t)_+^2/\sigma^2)$. *Then h can be expressed as*

$$h(t) = \inf_{\omega \in \mathbb{R}_+} \omega(1-t)_+^2 + \sigma^2 \varrho(\omega), \tag{A.1}$$

where

$$\varrho(\omega) = \begin{cases} +\infty, & \omega = 0, \\ \omega - log\, \omega, & 0 < \omega \leq 1, \\ 0, & \omega > 1. \end{cases} \tag{A.2}$$

Moreover, denoting

$$\omega^\star = \underset{\omega \in \mathbb{R}_+}{argmin}\, \omega(1-t)_+^2 + \sigma^2 \varrho(\omega),$$

we then have

$$\omega^\star = (1 + (1-t)_+^2/\sigma^2)^{-1}.$$

Acknowledgments

We thank the editor and the reviewers for their insightful comments and helpful suggestions that helped to improve the quality of this letter. Y.F. thanks Bertrand Gauthier for his helpful suggestions that improved the presentation of figures and notations in the letter. The research leading to these results received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC AdG A-DATADRIVE-B (290923). This letter reflects only our views: The EU is not responsible for any use that may be made of the information in it. The research leading to these results received funds from the following sources: Research Council KUL: GOA/10/09 MaNet, CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants; Flemish Government: FWO: PhD/Postdoc grants, projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity); IWT: PhD/Postdoc grants, projects: SBO POM (100031); iMinds Medical Information Technologies SBO 2014; Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012-2017). J.S. is a professor at KU Leuven, Belgium.

References

Andrews, D. F., & Hampel, F. R. (2015). *Robust estimates of location: Survey and advances*. Princeton, NJ: Princeton University Press.

Bache, K., & Lichman, M. (2013). *UCI machine learning repository*. http://archive.ics.uci.edu/ml

Bartlett, P. L., Jordan, M. I., & McAuliffe, J. D. (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, *101*(473), 138–156.

Bartlett, P. L., & Mendelson, S. (2003). Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, *3*, 463–482.

Brooks, P. J. (2011). Support vector machines with the ramp loss and the hard margin loss. *Operations Research*, *59*(2), 467–479.

Chapelle, O. (2007). Training a support vector machine in the primal. *Neural Computation*, *19*(5), 1155–1178.

Christmann, A., & Steinwart, I. (2004). On robustness properties of convex risk minimization methods for pattern recognition. *Journal of Machine Learning Research*, *5*, 1007–1034.

Christmann, A., Van Messem, A., & Steinwart, I. (2009). On consistency and robustness properties of support vector machines for heavy-tailed distributions. *Statistics and Its Interface*, *2*, 311–327.

Collobert, R., Sinz, F., Weston, J., & Bottou, L. (2006). Trading convexity for scalability. In *Proceedings of the 23rd International Conference on Machine Learning* (pp. 201–208). New York: ACM.

Cucker, F., & Zhou, D.-X. (2007). *Learning theory: An approximation theory viewpoint*. Cambridge: Cambridge University Press.

De Brabanter, K., Pelckmans, K., De Brabanter, J., Debruyne, M., Suykens, J. A. K., Hubert, M., & De Moor, B. (2009). Robustness of kernel based regression: A comparison of iterative weighting schemes. In *Artificial Neural Networks—ICANN 2009* (pp. 100–110). New York: Springer.

Debruyne, M., Christmann, A., Hubert, A., & Suykens, J. A. K. (2010). Robustness of reweighted least squares kernel based regression. *Journal of Multivariate Analysis*, *101*(2), 447–463.

Donoho, D. L., & Huber, P. J. (1983). The notion of breakdown point. In P. J. Bickel, K. Doksum, & J. L. Hodges (Eds.), *A festschrift for Erich L. Lehmann* (pp. 157–184). Belmont, CA: Wadsworth.

Feng, Y., Huang, X., Shi, L., Yang, Y., & Suykens, J. A. K. (2015). Learning with the maximum correntropy criterion induced losses for regression. *Journal of Machine Learning Research*, *16*, 993–1034.

Frénay, B., & Verleysen, M. (2014). Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(5), 845–869.

Geman, D., & Yang, C. (1995). Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing*, *4*(7), 932–946.

Grant, M., & Boyd, S. (2008). Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, & H. Kimura (Eds.), *Recent advances in learning and control*, Lecture Notes in Control and Information Sciences (pp. 95–110). New York: Springer-Verlag. http://stanford.edu/ boyd/graph_dcp.html

Grant, M., & Boyd, S. (2014). *CVX: Matlab software for disciplined convex programming, version 2.1*. http://cvxr.com/cvx

Hampel, F. R. (1971). A general qualitative definition of robustness. *Annals of Mathematical Statistics*, *42*(6), 1887–1896.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., & Stahel, W. A. (2011). *Robust statistics: The approach based on influence functions*. Hoboken, NJ: Wiley.

Huang, X., Shi, L., & Suykens, J. A. K. (2014). Ramp loss linear programming support vector machine. *Journal of Machine Learning Research*, *15*, 2185–2211.

Huber, P. J. (1964). Robust estimation of a location parameter. *Annals of Mathematical Statistics*, *35*(1), 73–101.

Kanamori, T., Fujiwara, S., & Takeda, A. (2014). *Breakdown point of robust support vector machine*. arXiv:1409.0934

Krause, N., & Singer, Y. (2004). Leveraging the margin more carefully. In *Proceedings of the 21st International Conference on Machine Learning* (p. 63). New York: ACM.

Lee, Y.-J., & Mangasarian, O. L. (2001). SSVM: A smooth support vector machine for classification. *Computational Optimization and Applications*, *20*(1), 5–22.

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 and Probability Letters*, *68*(1), 73–82.

Liu, W., Pokharel, P. P., & Principe, J. C. (2007). Correntropy: Properties and applications in non-gaussian signal processing. *IEEE Transactions on Signal Processing*, *55*(11), 5286–5298.

Long, P. M., & Servedio, R. A. (2010). Random classification noise defeats all convex potential boosters. *Machine Learning*, *78*(3), 287–304.

Masnadi-Shirazi, H., & Vasconcelos, N. (2009). On the design of loss functions for classification: Theory, robustness to outliers, and savageboost. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems, 21* (pp. 1049–1056). Cambridge, MA: MIT Press.

Mendelson, S. (2003). A few notes on statistical learning theory. In O. Bousquet, U. von Luxburg, & G. Rötsch (Eds.), *Advanced lectures on machine learning* (pp. 1–40). New York: Springer.

Nikolova, M., & Ng, M. K. (2005). Analysis of half-quadratic minimization methods for signal and image recovery. *SIAM Journal on Scientific Computing*, *27*(3), 937–966.

Park, S. Y., & Liu, Y. (2011). Robust penalized logistic regression with truncated loss functions. *Canadian Journal of Statistics*, *39*(2), 300–323.

Platt, J. (1999). Fast training of support vector machines using sequential minial optimization. In B. Schölkopf, C. J. C. Burges, & A. J. Smola (Eds.), *Advances in kernel methods—Support vector learning* (pp. 185–208). Cambridge, MA: MIT Press.

Reid, M. D., & Williamson, R. C. (2010). Composite binary losses. *Journal of Machine Learning Research*, *11*, 2387–2422.

Schölkopf, B., Herbrich, R., & Smola, A. J. (2001). A generalized representer theorem. In *Proceedings of the 14th Annual Conference on Computational Learning Theory* (pp. 416–426). New York: Springer.

Shen, X., Tseng, G. C., Zhang, X., & Wong, W. H. (2003). On ψ-learning. *Journal of the American Statistical Association*, *98*(463), 724–734.

Singh, A., Pokharel, R., & Principe, J. C. (2014). The C-loss function for pattern classification. *Pattern Recognition*, *47*(1), 441–453.

Steinwart, I. (2005). Consistency of support vector machines and other regularized kernel classifiers. *IEEE Transactions on Information Theory*, *51*(1), 128–142.

Steinwart, I., & Christmann, A. (2008). *Support vector machines*. New York: Springer.

Suykens, J. A. K., De Brabanter, J., Lukas, L., & Vandewalle, J. (2002). Weighted least squares support vector machines: Robustness and sparse approximation. *Neurocomputing*, *48*(1), 85–105.

Suykens, J. A. K., Signoretto, M., & Argyriou, A. (Eds.). (2014). *Regularization, optimization, kernels, and support vector machines*. Boca Raton, FL: CRC Press.

Takeda, A., Fujiwara, S., & Kanamori, T. (2014). Extended robust support vector machine based on financial risk minimization. *Neural Computation*, *26*(11), 2541–2569.

Wang, L., Zhu, J., & Zou, H. (2007). Hybrid Huberized support vector machines for microarray classification. In *Proceedings of the 24th International Conference on Machine Learning* (pp. 983–990). New York: ACM.

Wu, Y., & Liu, Y. (2007). Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association*, *102*(479), 974–983.

Wu, Y., & Liu, Y. (2013). Adaptively weighted large margin classifiers. *Journal of Computational and Graphical Statistics*, *22*(2), 416–432.

Yang, Y., Feng, Y., & Suykens, J. A. K. (in press). Robust low rank tensor recovery with regularized redescending M-estimator. *IEEE Transactions on Neural Networks and Learning Systems*.

Zhang, T. (2004). Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, *32*(1), 56–85.

Received September 25, 2015; accepted January 27, 2016.

This article has been cited by:

1. Hong Chen, Yulong Wang. 2016. Kernel-based sparse regression with the correntropy-induced loss. *Applied and Computational Harmonic Analysis* . [CrossRef]