View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by City Research Online

CITY
UNIVERSITY OF LONDON
EST 1894

City Research Online

City, University of London Institutional Repository

Citation: Asimit, V. ORCID: 0000-0002-7706-0066, Kyriakou, I. ORCID: 0000-0001-9592-596X, Santoni, S. ORCID: 0000-0002-5928-3901, Scognamiglio, S. and Zhu, R. ORCID: 0000-0002-9944-0369 (2021). Robust Classification via Support Vector Machines. .

This is the preprint version of the paper.

This version of the publication may differ from the final published version.

Permanent repository link: https://openaccess.city.ac.uk/id/eprint/25936/

Link to published version:

Copyright and reuse: City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.

City Research Online: http://openaccess.city.ac.uk/ publications@city.ac.uk

Robust Classification via Support Vector Machines

Vali Asimit, Ioannis Kyriakou, Simone Santoni, Salvatore Scognamiglio and Rui Zhu

Abstract—The loss function choice for any Support Vector Machine classifier has raised great interest in the literature due to the lack of robustness of the Hinge loss, which is the standard loss choice. In this paper, we plan to robustify the binary classifier by maintaining the overall advantages of the Hinge loss, rather than modifying this standard choice. We propose two robust classifiers under data uncertainty. The first is called *Single Perturbation SVM (SP-SVM)* and provides a constructive method by allowing a controlled perturbation to one feature of the data. The second method is called *Extreme Empirical Loss SVM (EEL-SVM)* and is based on a new empirical loss estimate, namely, the *Extreme Empirical Loss (EEL)*, that puts more emphasis on extreme violations of the classification hyper-plane, rather than taking the usual sample average with equal importance for all hyper-plane violations. Extensive numerical investigation reveals the advantages of the two robust classifiers on simulated data and well-known real datasets.

Index Terms—Support vector machine, robust classification, data uncertainty, extreme empirical loss.

1 Introduction

In many real-life problems, the assumption that input data are not affected by noise is unrealistic. In fact, both data recording and variable measurement are exposed to errors. For example, information asymmetry affects the accuracy of the recorded data points, whereas the design of a survey influences the answers to subjective questions. The presence of noise contamination in the data motivates the need for robust classification methods. For example, a robust classifier would enhance the classification error in the presence of data uncertainty with respect to the feature space and its label available information.

This paper aims to identify some robust formulations of the classical SVM that could extend to any other SVM-type classifier, such as twin support vector machines (TWSVM) [1].[1] We propose two robust formulations. First, we introduce a controlled perturbation to the data for a single feature. This allows the modeler to understand the sensitivity of the classifier vis-à-vis the uncertainty in the data that is induced by each feature from the given feature space. Our second robust formulation is based on a novel empirical loss estimate, namely, the Extreme Empirical Loss (EEL). More specifically, EEL emphasizes extreme violations of the classification hyper-plane rather than taking the usual sample average with equal importance for all hyper-plane violations. It is not surprising that EEL penalizes efficiently the classification violations as it provides more flexibility in modeling the permitted classification violations.

The paper is organized as follows. Section 2 provides the background for this study. Section 3 illustrates the two proposed robust SVM formulations. In Section 4, we report the results from our numerical experiments conducted over synthetic and popular real datasets. Section 5 concludes and presents recommendations that emerge from our analyses. More detailed analysis of theoretical results is deferred to the Appendix in Section 6.

2 Background and Problem Definition

The starting point is the training set that contains N instances and the associated labels, $\Big\{(\mathbf{x}_i, y_i),\ i=1,\ldots,N\Big\}$, which are observed values from an independent and identically distributed sample drawn from (X,Y), where $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ and $y_i \in \mathcal{Y}$. The binary classification reduces to $\mathcal{Y} := \{-1,1\}$, where $y_i = 1$ if $\mathbf{x}_i \in \mathcal{C}_{+1}$ and $y_i = -1$ if $\mathbf{x}_i \in \mathcal{C}_{-1}$. Our main objective is to construct an accurate (binary) classifier $c : \mathcal{X} \to \{-1,1\}$ which maximizes the probability that $c(\mathbf{x}_i) = y_i$.

The SVM aims to identify a separation hyper-plane $\mathbf{w}^T\phi(\mathbf{x}) + b$ that generates two parallel supporting hyper-planes as follows:

$$\mathbf{w}^T\phi(\mathbf{x}) + b = 1 \quad \text{and} \quad \mathbf{w}^T\phi(\mathbf{x}) + b = -1, \tag{2.1}$$

where $\phi(\cdot) : \Re$ is a real-valued function that transforms the feature space into a synthetic feature space that allows a linear hyper-plane separation of the data. The data are rarely perfectly separable and a compromise is made by allowing classification violations for non-separable data, also known as *soft margin SVM*, that is formulated as

$$\min_{\mathbf{w},b} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N} L\Big(1 - y_i\big(\mathbf{w}^T\phi(\mathbf{x}_i) + b\big)\Big), \tag{2.2}$$

where the first term aims to find the 'best' classifier by maximizing the distance between the hyper-planes in (2.1) and the second term means to penalize the classifier's violations measured by a carefully chosen loss function $L : \Re \to \Re_+$;

- *V. Asimit, I. Kyriakou and R. Zhu are with the Faculty of Actuarial Science and Insurance, Cass Business School, City, University of London, London EC1Y 8TZ, UK.*
- *S. Santoni is with the Faculty of Management, Cass Business School, City, University of London, London EC1Y 8TZ, UK.*
- *S. Scognamiglio is with the Department of Business and Quantitative Studies, University of Naples "Parthenope", Generale Parisi Street, 80132 Naples, Italy.*

[1] Hereafter, we will refer to the standard SVM representation only.

for details, see [2], [3]. The extant literature has dealt with several piecewise loss functions, such as:

i) *Hinge loss:* $L_H(u) := \max\{0, u\}$;
ii) *Truncated Hinge loss ($a \geq 1$):* $L_{TH}(u) := \min\{\max\{0, u\}, a\}$;
iii) *Pinball loss ($a \leq 0$):* $L_P(u) := \max\{au, u\}$;
iv) *Pinball loss with 'ϵ zone' ($\epsilon \geq 0$ and $a, b \leq 0$):* $L_{PEZ}(u) := \max\{0, u - \epsilon, au + b\}$;
v) *Truncated Pinball loss ($a \leq 0$ and $b \geq 0$):* $L_{TP}(u) := \max\{u, \min\{au, b\}\}$.

The simplicity of L_H is obvious as it reduces (2.2) to solving a convex *Linearly Constrained Quadratic Program (LCQP)*, which is the original SVM formulation as explained in [2], [3]; moreover, the Hinge loss is proved to be an upper bound of the classification error (see [4] and [5]), therefore is a pivotal loss choice. At the same time, the Hinge loss has been criticized for not being robust and sensitive to outliers, whereas the Truncated Hinge loss proposed by [6] overcomes this issue at the expense of computational complexity; this loss leads to non-convex optimization problems that rely on *Difference of Convex Functions Optimization Algorithm (DCA)* and suffers by scalability issues, nevertheless it behaves well under data uncertainty for small datasets. The two Pinball losses reduce to solving LCQP with many more linear equality constraints than the Hinge loss, but the Pinball loss seems to be more robust in noise sensitivity and stable when resampling (see [7]). Similar arguments have been used in [5] to justify the Truncated Pinball loss as a good choice when dealing with feature uncertainty, but it has the same drawback with the Truncated Hinge loss, i.e., the DCA algorithm is the designated solver, which is efficiently implemented only for relative small datasets.

Non-piecewise linear loss functions also appear various forms. The *Least Square loss*, i.e., $L_{LS}(u) := u^2$, is considered in [8] via a different LCQP formulation than the Hinge loss case. The *Correntropy loss* is defined in [9]; various compositions of this with other loss functions can be investigated, e.g., [10] for the SVM case and [11] for the imbalanced TWSVM case. One could understand the advantages of non-linear convex loss functions for other classification methods in [12], where once again the Hinge loss is shown to be the tightest margin-based upper bound of the mis-classification loss for many classification problems. Further, it is numerically shown that this property does not suffice to think of the Hinge loss as the universally best choice to measure mis-classification. Therefore, the classification efficiency depends on the loss choice for most of the classification methods, and this is true also for SVM. Strictly convex loss functions are argued in [13] to possess statistical properties when studying mis-classification. It is worth noting that binary classification has been mainly considered so far in the paper, and multi-class classification requires careful attention on the loss function and not only; e.g., [6] shows that a version of the Pinball loss with 'ϵ zone' can help when dealing with outliers in classification methods with at least three classes.

A desirable loss function property for a generic classification method is the Fisher consistency or classification calibration (see [13]). By definition, the loss function L is Fisher-consistent if

$$\underset{\{f:\, f:\mathcal{X}\to\Re\}}{\operatorname{argmin}} \mathbf{E}_{\mathcal{X},\mathcal{Y}} L\Big(1 - Y f(\mathbf{X})\Big) \tag{2.3}$$

leads to the Bayes classifier. In the context of binary SVM classification, one can show that (2.3) holds if

$$\underset{z\in\Re}{\operatorname{argmin}} \mathbf{E}_{\mathcal{Y}|\mathbf{x}} L\Big(1 - Yz\Big) \tag{2.4}$$
$$= \begin{cases} 1, & \text{if} \quad \Pr\big(Y = 1|\mathbf{x}\big) > \Pr\big(Y = -1|\mathbf{x}\big) \\ -1, & \text{if} \quad \Pr\big(Y = 1|\mathbf{x}\big) < \Pr\big(Y = -1|\mathbf{x}\big) \end{cases}$$

is true for all $\mathbf{x} \in \mathcal{X}$ (e.g., see Proposition 1 in [6]).

The Fisher consistency has been investigated intensively in the literature; we provide a concise review relatable to our aims. Theorem 3.1 in [12] shows that, if the global minimizer of (2.3) exists, then it has to be the same as the Bayes decision rule, which is valid for any classification method. In the binary SVM setting, Proposition 1 in [6] and Theorem 1 in [5] show that this property also holds for non-convex loss functions, where the first result covers a large set of truncated loss functions, whereas the second focuses on the Truncated Pinball loss. An early result of [14], namely, Lemma 3.1, proves that the Hinge loss is Fisher-consistent; also, Theorem 1 in [7] shows that the Pinball loss — a convex loss function — is Fisher-consistent as well. Our next result extends the Fisher-consistency to a general convex loss function L for the binary SVM case.

Theorem 2.1. *Assume that $L : \Re \to \Re_+$ is a convex loss function such that $L(0) = 0$. If $L(\cdot)$ is linear on $(0, 2 + \epsilon)$ for some $\epsilon > 0$, then L is Fisher-consistent.*

The proof of Theorem 2.1 can be found in Section 6.1. Therefore, we further consider in Section 3 only convex loss functions that require convex programming when solving (2.1), which should be — at least in theory — computationally feasible even for large-scale problems.

3 Robust SVM

The standard SVM classifier is intuitive and provides a transparent procedure that is very appealing to practitioners. As mentioned in the previous section, the SVM method has been criticized for being sensitive to outliers and can lead to non-robust classification criteria when dealing with data uncertainty. A substantial effort has been made in the recent literature to tackle this drawback and there are two main strands of research in this respect.

The first approach to robust SVM is the use of different loss functions, with some examples given in [11], [5], [6], [7], [8], [9], [10] where the model robustness is probed by data contamination concerning the features and/or labels. The second approach relaxes the constraints of the standard optimization problem and has (2.2) rewritten as a *Chance Constrained (CC)* instance and/or relies on *Robust optimization (RO)*. The two ideas are interrelated: the model assumptions are considered uncertain and this enlarges the feasibility set somehow. The CC choice is considered by [15], [16], [17], [18], [19], [20] where data uncertainty is assumed amongst all features. This, in turn, requires some less explainable choices of the probability uncertainty set, or relies on knowing the empirical covariance (feature) matrix, which is a notoriously problematic assumption as its estimate is often not a positive definite matrix. One then normally needs to create an uncertainty set with respect to the covariance matrix, or find a positive definite matrix that is 'sufficiently' close to the empirical covariance (feature) matrix under an ad hoc criterion. The RO choice is explored in [21], where minimizing a 'worst-case scenario' is chosen to acknowledge the data uncertainty. This approach has the same drawback of reliance on a well-behaved empirical covariance (feature) matrix, which is not practical. It should be noted that RO is widely accepted as a robust method with respect to the optimal objective function, but not necessarily with respect to its optimal solution (see [22]) though finding a robust optimal solution is the main aim of a robust SVM.

Before deploying our robust SVM proposals, we explain briefly the concept of robust statistics inaugurated in the seminal paper of [23]. It should be noted that [24] actually formalizes the notion of robustness, which is a continuity property of a generic estimator, where the breakdown point of the estimator is introduced. This concept becomes a quantitative measure of robustness in the statistical literature; for a thorough introduction to robust statistics, readers may refer to the comprehensive review of [25]. Simply stated, the question is by how much the estimate changes when the underlying unknown distribution is slightly perturbed. One way to think of this perturbation is to consider by how much the sample needs to change in order to significantly deviate from the observed sample estimate. Therefore, robust statistics are a good choice for detecting outliers and, as a result, make the estimation exercise more robust.

We are now ready to proceed with our two robust SVM formulations, namely, the *Single Perturbation SVM (SP-SVM)* and the *Extreme Empirical Loss SVM (EEL-SVM)*.

3.1 Single Perturbation SVM

Our first robust formulation is designed by introducing perturbation in the data without relying on the empirical covariance (feature) matrix, which usually brings together other sources of uncertainty that are very difficult to deal with. More specifically, we only allow one perturbation to the most influential feature in the spirit of statistical robustness, hoping for improvement of the classification error. The most influential feature can be chosen by the modeler when the domain knowledge provides credible evidence or via basic statistical evidence, e.g., the feature with the largest variation, skewness, etc. Alternatively, one may apply the SP-SVM method for each possible feature amongst those expected to have a hefty contribution to the overall sampling error, so that the method provides a constructive sensitivity analysis to the standard SVM classifier in the sense that the data uncertainty is not only localized, but also included in the classification decision.

It is well-known that solving (2.2) under the Hinge loss is equivalent to solving the following LCQP instance:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \ & \frac{1}{2}\mathbf{w}^T\mathbf{w}+C\sum_{i=1}^{N}\xi_i \qquad (3.1)\\ \text{s.t.}\ & y_i\Big(\mathbf{w}^T\phi(\mathbf{x}_i)+b\Big)\geq 1-\xi_i,\ \xi_i\geq 0,\ \ 1\leq i\leq N, \end{aligned}$$

where the tuning parameter satisfies $C > 0$.

As anticipated, we are interested in calibrating (3.1) in the presence of data uncertainty with respect to one feature, e.g., the k^{th} feature. Thus, the j^{th} entry of $\phi(\mathbf{x}_i)$, denoted by $\phi_j(\mathbf{x}_i)$ is deterministic for all $1 \leq i \leq N$ and $1 \leq j \neq k \leq d$, whereas the k^{th} feature is affected by an error term Z_{ik} and $\phi_k(\mathbf{x}_i)$ is replaced by $\phi_k(\mathbf{x}_i) + Z_{ik}$ for all $1 \leq i \leq N$. Moreover, each error term is defined on a probability space $(\Omega_{ik}, \mathcal{F}, P)$ with $\Omega_{ik} \subseteq \Re$. Therefore, (3.1) for a randomized k^{th} feature becomes

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \ & \frac{1}{2}\mathbf{w}^T\mathbf{w}+C\sum_{i=1}^{N}\xi_i \qquad (3.2)\\ \text{s.t.}\ & \Pr\bigg(y_i\Big(\mathbf{w}^T\phi(\mathbf{x}_i)+w_kZ_{ik}+b\Big)\geq 1-\xi_i\bigg)\geq\alpha,\\ & \xi_i\geq 0,\ 1\leq i\leq N, \end{aligned}$$

where $\alpha \in [0, 1]$ reflects the modeler's belief of the sensitivity of the k^{th} feature with respect to the classification decision; clearly, w_k is the k^{th} element of $\mathbf{w}$. This kind of probability-like constraint is also known as the *chance constraint* in the operations research literature.

For any given tuple (i, k), $F_{ik}(\cdot) := \Pr(Z_{ik} \leq \cdot)$ is defined on Ω_{ik} and is known as the *cumulative distribution function (cdf)* of Z_{ik}. Further, two generalized inverse functions are defined as follows:

$$\begin{aligned} F_{ik}^{-1}(t) &:= \inf\big\{x\in\Re:\ F_{ik}(x)\geq t\big\},\\ F_{ik}^{-1+}(t) &:= \sup\big\{x\in\Re:\ F_{ik}(x)\leq t\big\}, \end{aligned}$$

for all $t \in [0, 1]$, where $\inf\emptyset = \infty$ and $\sup\emptyset = -\infty$ hold by convention. It is not difficult to show that

$$t\leq\Pr\big(Z_{ik}\leq x\big)\Leftrightarrow F_{ik}^{-1}(t)\leq x,\ \ x\in\Re \text{ and } t\in[0,1]$$

and

$$\Pr\big(Z_{ik}<x\big)\leq t\Leftrightarrow x\leq F_{ik}^{-1+}(t),\ \ x\in\Re \text{ and } t\in[0,1].$$

Therefore, the chance constraint of (3.2) is equivalent to

$$\begin{aligned} & y_i\Big(\mathbf{w}^T\phi(\mathbf{x}_i)+b\Big)+y_iw_kF_{ik}^{-1+}(1-\alpha)\geq 1-\xi_i \qquad (3.3)\\ & y_i\Big(\mathbf{w}^T\phi(\mathbf{x}_i)+b\Big)+y_iw_kF_{ik}^{-1}(\alpha)\geq 1-\xi_i \end{aligned}$$

whenever $y_iw_k \geq 0$ and $y_iw_k < 0$, respectively. Without imposing any restriction on F_{ik}, the conditional constraint from (3.3) makes (3.2) a mixed integer programming optimization problem that is difficult to solve efficiently as the inequality constraints depend on the sign of w_k. The next set of conditions on the error terms enable us to efficiently solve (3.2).

Assumption 3.1. *For a given integer* $1 \leq k \leq d$ *and* $\alpha \in [0, 1]$, $F_{ik}^{-1}(\alpha) + F_{ik}^{-1+}(1-\alpha) = 0$ *holds.*

Clearly, any error term defined on $\Omega_{ik} = \big(-\omega_{ik}, \omega_{ik}\big)$ with $0 < \omega_{ik} \leq \infty$ such that its cdf is continuous, increasing and $F_{ik}(\cdot) + F_{ik}(-\cdot) = 1$ in a neighborhood of $F_{ik}^{-1+}(\alpha)$ satisfies the conditions stated in Assumption 3.1. Symmetric continuous cdfs, such as the Gaussian, Student's t or any other member of the elliptical family of distributions centered at 0 (for details, see [26]), satisfy these conditions and, implicitly, Assumption 3.1. The elliptical family is a vast set of distributions that are commonly used in data science, econometric and statistical error modeling, therefore such choices are plausible for our SP-SVM classifier.

Under Assumption 3.1, (3.3) is equivalent to

$$y_i\Big(\mathbf{w}^T\phi(\mathbf{x}_i)+b\Big) - |w_k|F_{ik}^{-1}(\alpha) \geq 1-\xi_i,$$

and, in turn, the instance (3.2) is equivalent to solving

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i\Big(\mathbf{w}^T\phi(\mathbf{x}_i)+b\Big) \geq 1-\xi_i, \\ & y_i\Big(\mathbf{w}^T\phi(\mathbf{x}_i)+b\Big) - y_i w_k a_{ik} \geq 1-\xi_i, \\ & y_i\Big(\mathbf{w}^T\phi(\mathbf{x}_i)+b\Big) + y_i w_k a_{ik} \geq 1-\xi_i, \\ & \xi_i \geq 0, \;\; 1 \leq i \leq N, \end{aligned} \tag{3.4}$$

provided that $a_{ik} := F_{ik}^{-1}(\alpha) \geq 0$. It is evident that (3.4) can be solved efficiently; in addition, (3.1) and (3.4) have the same level of computational complexity and the two instances are identical when $a_{ik} = 0$, but (3.4) is always more conservative than (3.1).

The solution of (3.4) is relegated to Section 6.2, but before leaving this section, we provide a practical recommendation regarding a 'reasonable' choice for parameter a_{ik}. One choice would be to assume Gaussian errors with zero mean and variance given by the sampling error estimate, i.e.,

$$\hat{a}_{ik} = q_{\alpha,G}^{-1}\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\big(x_{ik}-\bar{x}_{kl}\big)^2} \text{ and } \bar{x}_{kl} := \frac{1}{N}\sum_{i=1}^{N} x_{ik},$$

where $q_{\alpha,G}$ is the α-Normal quantile. It might be argued that Gaussian errors are unable to capture the level of data uncertainty and that more heavy-tailed errors, such as from Student's t or any other elliptical distribution, are more appropriate; in such a case, we can simply replace $q_{\alpha,G}$ by the α quantile of the distribution of choice.

3.2 Extreme Empirical Loss SVM

Our second robust formulation is conceptually different from all the other attempts to robustify the SVM classifier. Before detailing our model, we note that the soft margin SVM from (2.2) can be rewritten as

$$\min_{\mathbf{w},b} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\widehat{E}\Big[L\Big(1-Y\big(\mathbf{w}^T\phi(\mathbf{X})+b\big)\Big)\Big], \tag{3.5}$$

where the second term acts as the empirical estimate of the penalty associated with the classifier's violations; it is given by the average deviation measured via the loss function L. According to our knowledge, all SVM classifiers focus on changing the choice of L, but the penalty term is always based on taking the usual sample average with equal importance for all hyper-plane violations. The choice of loss function can influence the borderline decisions where examples can be classified either way, and a clever loss choice can reduce the mis-classification error. The empirical estimate of the overall violation induced by a classifier is also influenced by the way we perceive these individual violations, and a neutral approach is to average these deviations based on equal weights. Our EEL-SVM aims to focus more on the large deviations that may considerably perturb the classification decision in the presence of data uncertainty. To this end, we place more weight to the larger violations via a novel empirical loss estimate, namely, the *Extreme Empirical Loss (EEL)*, which is formulated as

$$\min_{z} z + \frac{1}{N(1-\alpha)}\sum_{i=1}^{N}\max\big\{\zeta_i - z, 0\big\}, \tag{3.6}$$

where $\zeta_i = L\Big(1-y_i\big(\mathbf{w}^T\phi(\mathbf{x}_i)+b\big)\Big)$, $1 \leq i \leq N$, are the individual violation penalties. Note that (3.6) is the empirical estimate of the *Conditional Value-at-Risk at level α ($CVaR_\alpha$)* of the classifier's violation, i.e.,

$$\widehat{CVaR}_\alpha\bigg(L\Big(1-Y\big(\mathbf{w}^T\phi(\mathbf{X})+b\big)\Big)\bigg);$$

for details, see the seminal paper of [27] which introduces $CVaR$ as a risk management measure in the insurance and banking sectors. The parameter $0 \leq \alpha < 1$ represents the caution level chosen by the modeler; the higher the value of α, the more conservative the EEL. This is made obvious by noting that (3.6) reduces to

$$\frac{1}{r}\sum_{i=1}^{N}\zeta_{i,N} \text{ if } \alpha = 1-\frac{r}{N},\; 1 \leq r \leq N$$

for any integer r, where $\zeta_{1,N} \geq \zeta_{2,N} \geq \ldots \geq \zeta_{N,N}$ are the upper order statistics of the sample $\big\{\zeta_i; 1 \leq i \leq N\big\}$. Clearly, the least conservative EEL is attained when $\alpha = 0$, and becomes the sample average $\frac{1}{N}\sum_{i=1}^{N}\zeta_i$.

In summary, EEL aims to penalize more the larger deviations so that the separation hyper-planes are expected to be less sensitive to noisy features, which should make the classification decision more robust. Similarly to SP-SVM, EEL-SVM is not specifically designed to robustify SVM classifiers with noisy labels, thus our numerical investigations in Section 4 only include data with noisy features rather than adding label perturbations.

By keeping in mind (2.2) and (3.6), the EEL-SVM formulation reduces to solving the instance

$$\begin{aligned} \min_{\mathbf{w},b,z,\boldsymbol{\xi}} \quad & \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + Dz + \frac{D}{N(1-\alpha)}\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & \xi_i + z \geq L\Big(1-y_i\big(\mathbf{w}^T\phi(\mathbf{x}_i)+b\big)\Big), \\ & \xi_i \geq 0, \;\; 1 \leq i \leq N, \end{aligned}$$

for any loss function L, while the Hinge loss choice EEL-SVM becomes another convex LCQP instance:

$$\begin{aligned} \min_{\mathbf{w},b,z,\boldsymbol{\xi}} \quad & \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + Dz + \frac{D}{N(1-\alpha)}\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i\big(\mathbf{w}^T\phi(\mathbf{x}_i)+b\big) + z \geq 1-\xi_i, \\ & \xi_i + z \geq 0,\; \xi_i \geq 0, \;\; 1 \leq i \leq N. \end{aligned} \tag{3.7}$$

One may derive similar formulations for any other loss function and easily write the convex LCQP formulations for Pinball loss and Pinball loss with 'ϵ zone'. Non-convex loss functions, such as Truncated Hinge and Truncated Pinball, require bespoke DCA solvers, but such details are beyond the scope of this paper. The simplified solution of (3.7) is given in Section 6.3 via the usual duality arguments.

4 Numerical Experiments

In this section, we carry out numerical experiments to validate our SP-SVM and EEL-SVM classifiers and compare them in terms of accuracy and robustness resilience with three SVM-like competitors: the standard SVM (C-SVM) from [2], the Pinball SVM (pin-SVM) from [7] and the Truncated Pinball SVM ($\overline{pin}$-SVM) from [5]. A comprehensive comparison amongst C-SVM, pin-SVM and $\overline{pin}$-SVM and, therefore, the synthetic datasets in Section 4.1 follow the setting from [7] and [5]. Section 4.2 further compares these classifiers for various widely investigated real-life data. The code is available at https://github.com/salvatorescognamiglio/SPsvm_EELsvm.

4.1 Synthetic Data

The first set of numerical experiments aims to compare the classification performance of SP-SVM and EEL-SVM with C-SVM, pin-SVM and $\overline{pin}$-SVM for a simple synthetic dataset discussed in [7] and [5]. In Section 4.1.1 no data uncertainty is added, whereas in Section 4.1.2 we compare the classifiers when controlled data contamination is introduced in order to understand the degree of robustness of each method.

The non-contaminated data are simulated based on a Gaussian bivariate model for which a linear classification boundary is known. The nested simulation requires the labels to be simulated from a Bernoulli random variable B with probability of 'success' $p = 0.5$; therefore, we simulate N random variates from this distribution, where the sample size N is the total number of examples from the two classes, '-1' and '1'; we consider $N \in \{100, 200\}$. The features $\{\mathbf{x}_i\}_{i=1}^N$ are then randomly generated as follows:

$$\mathbf{X}_i \,|\, B{=}1 \sim \mathcal{N}\big(\boldsymbol{\mu}, \boldsymbol{\Sigma}\big) \text{ and } \mathbf{X}_i \,|\, B{=}{-1} \sim \mathcal{N}\big(-\boldsymbol{\mu}, \boldsymbol{\Sigma}\big), \quad (4.1)$$

where $\boldsymbol{\mu} = [0.5, -3]^T$ and $\boldsymbol{\Sigma} = \mathrm{diag}(0.2, 3)$. Note that these synthetic data have an analytic Bayes classifier given by $x_2 = m_0 x_1 + q_0$, where $m_0 = 2.5$ and $q_0 = 0$. Our aim is to estimate m and q for all SVM classifiers and compare them with the Bayes classifier that defines the ideal classification boundary. Note that the SP-SVM training is performed by considering data uncertainty only with respect to the second feature, which has a higher theoretical standard deviation than the first feature, but the sample standard deviation estimates should follow the same pattern.

4.1.1 Synthetic Non-contaminated Data

We first simulate various samples from the nested model and conduct 10-fold cross-validation to tune the parameters associated with each classifier.

The SP-SVM and EEL-SVM require tuning of the parameter α, the pin-SVM requires tuning of τ, whereas τ and s must both be tuned for the $\overline{pin}$-SVM. The 10-fold cross-validation is run on the following parameter spaces:

- $\alpha \in \mathcal{A}_{SP} = \{0.50, 0.51, \ldots, 0.60\}$ for SP-SVM;
- $\alpha \in \mathcal{A}_{EEL} = \{0, 0.01, 0.02\}$ for EEL-SVM;
- $\tau \in \mathcal{T}_{pin} = \{0.1, 0.2, \ldots, 1\}$ for pin-SVM;
- $(\tau, s) \in \mathcal{T}_{\overline{pin}} \times \mathcal{S} = \{0, 0.25, 0.75\} \times \{0, 0.25, 0.5, 1\}$ for $\overline{pin}$-SVM.

For fairness of comparison, we have defined parameter spaces with similar cardinalities. Only for the EEL-SVM, we opt for a smaller set. We also choose the same penalty value for all methods setting $C = 100$ for C-SVM, SP-SVM, pin-SVM and $\overline{pin}$-SVM and $D = 100 \times N$ for EEL-SVM.

The validity of all the SVM models is checked by simulating 100 independent samples of size N, and then the linear decision rule is computed, i.e., (m_i, q_i) for all $1 \leq i \leq 100$. Each classifier is fairly compared with the Bayes classification boundary via the distance

$$d_j = |\bar{m}_j - m_0|\, \widehat{\sigma}_{m_j} + |\bar{q}_j - q_0|\, \widehat{\sigma}_{q_j}, \qquad (4.2)$$

for $j \in \{$SP-SVM, EEL-SVM, pin-SVM, $\overline{pin}$-SVM, C-SVM$\}$, where $\bar{m}_j$ $(\bar{q}_j)$ and $\widehat{\sigma}_{m_j}$ $(\widehat{\sigma}_{q_j})$ are, respectively, the mean and standard deviation estimates of m (q) based on the 100 point estimates of m (q). The summary results are reported in Table 4.1, where we observe that there is no clear ranking among the methods under study when non-contaminated data are considered.

TABLE 4.1
Distance (4.2) between various SVM classifiers and Bayes classifier for non-contaminated synthetic data

	$N = 100$	$N = 200$
C-SVM	0.3763	**0.0185**
pin-SVM	0.1132	0.0337
$\overline{pin}$-SVM	0.2407	0.0188
SP-SVM	**0.1014**	0.2166
EEL-SVM	0.3477	0.0397

4.1.2 Synthetic Contaminated Data

We now test the robustness by comparing the performance of all the classifiers for synthetic data randomly generated from the same nested model, but now with a percentage $r \in [0, 1]$ of them contaminated, i.e., only $(1 - r) \times N$ examples are generated according to (4.1). The contaminated data can be generated in different ways. One possibility is to generate random numbers around a 'central' point from the theoretical separation hyper-plane; our focal point is $(0, 0)$ as in [7] and [5]. The features, i.e., $\mathbf{x}_i$'s, are random vectors generated from three elliptical distributions centered at $(0, 0)$, namely, a bivariate Normal $\mathcal{N}\big(\mathbf{0}, \boldsymbol{\Sigma}_c\big)$ and two bivariate Student's $t\big(\mathbf{0}, \boldsymbol{\Sigma}_c, g\big)$ with $g \in \{5, 1\}$ degrees of freedom, where the covariance matrix is

$$\boldsymbol{\Sigma}_c = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}.$$

The labels, i.e., y_i's, are randomly generated from the Bernoulli distribution with a probability of 'success' of 0.5. Note that the linear separation makes the contamination equally, on average, distributed on both sides of the hyper-plane; moreover, the negative correlation from $\boldsymbol{\Sigma}_c$ is deliberate so that the contaminated points are more likely to be wrongly labeled. A pictorial representation of our contamination model and the classification rules is provided in Figure 4.1 via 10-fold cross-validation; the plot shows a sample of size $N = 200$ with a contamination rate $r = 5\%$ generated from the three elliptical distributions and the green points signify the contaminated data.

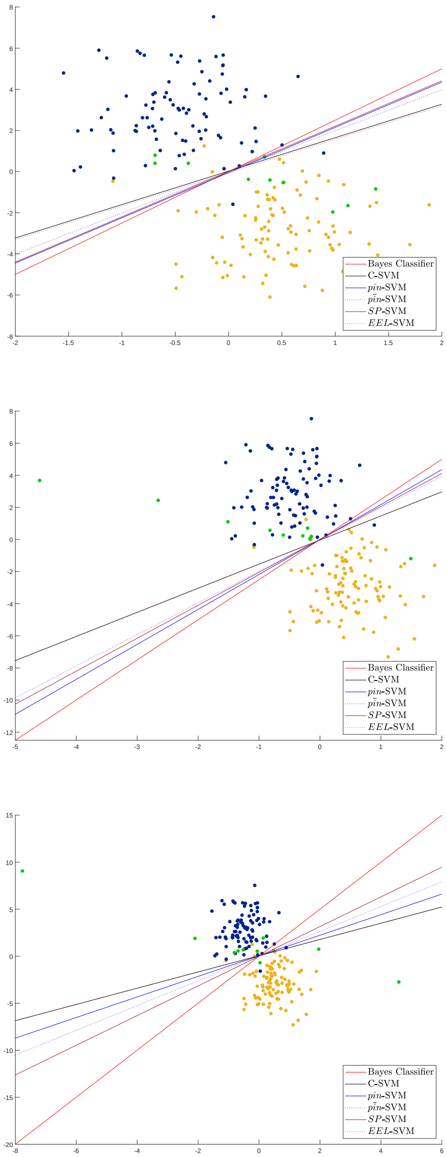

Fig. 4.1. Classification boundaries of all five SVM classifiers for Normal (top), Student's t with 5 degrees of freedom (middle) and Student's t with 1 degree of freedom (bottom) contaminated data.

TABLE 4.2
Distance (4.2) between various SVM classifiers and Bayes classifier for contaminated synthetic data.

	Normal distribution			
	$r = 0.05$		$r = 0.10$	
	$N = 100$	$N = 200$	$N = 100$	$N = 200$
C-SVM	0.8397	0.6623	1.1455	0.8675
pin-SVM	**0.0704**	**0.1880**	**0.2955**	0.2996
$\overline{pin}$-SVM	0.4659	0.3727	0.7951	0.4502
SP-SVM	0.2305	0.2994	0.5938	**0.2813**
EEL-SVM	0.8431	0.6788	1.1682	0.8785
	Student's t distribution 5df			
	$r = 0.05$		$r = 0.10$	
	$N = 100$	$N = 200$	$N = 100$	$N = 200$
C-SVM	1.1520	0.6795	1.4983	0.9863
pin-SVM	**0.2966**	**0.1710**	**0.6861**	**0.3322**
$\overline{pin}$-SVM	0.6318	0.3945	1.0209	0.6642
SP-SVM	0.7405	0.3754	0.8801	0.4539
EEL-SVM	1.1560	0.6895	1.5077	1.0025
	Student's t distribution 1df			
	$r = 0.05$		$r = 0.10$	
	$N = 100$	$N = 200$	$N = 100$	$N = 200$
C-SVM	1.8189	2.0358	2.6941	3.2549
pin-SVM	**1.1983**	1.6466	2.0853	2.1472
$\overline{pin}$-SVM	1.3674	1.3883	1.9844	2.5569
SP-SVM	1.2384	**1.2675**	**1.8261**	**1.8463**
EEL-SVM	1.8558	2.0280	2.6788	3.2757

The actual numerical experiments are once again resampled 100 times in order to somehow eliminate the effect of sampling error. For each sample, we conduct the 10-fold cross-validation to tune the additional parameters and we compute the linear decision rule. The performances are measured via the distance (4.2). The summary of this analysis is exhibited in Table 4.2 for samples of size $N \in \{100, 200\}$ and contamination ratio $r \in \{0.05, 0.10\}$. Note that the tuning parameters are calibrated as in Section 4.1.1, the same hyper-parameter spaces are adopted for the tuning of SP-SVM, pin-SVM and $\overline{pin}$-SVM, whereas for EEL-SVM we extend $\mathcal{A}_{EEL}$ as follows:

$$\mathcal{A}_{EEL} := \begin{cases} \{0, 0.01, \ldots, 0.05\} & \text{if } r = 0.05 \\ \{0, 0.01, \ldots, 0.10\} & \text{if } r = 0.10 \end{cases}.$$

Table 4.2 reveals some interesting evidence. As expected, the performances of all methods deteriorate when the data are strongly contaminated. The distance from the Bayes classifier increases with the contamination ratio r. In addition, the performances worsen when we move from the Normal to Student's t distribution with 5 degrees of freedom, and further deteriorate when the degrees of freedom decrease to 1. Comparing the methods' performances, we find that the SP-SVM and pin-SVM are more robust than the others (see boldfaced reports). In almost all noise scenarios these two have a smaller distance from the Bayes classifier, whereas the $\overline{pin}$-SVM appears to be competitive in only a few cases. The ranking between the SP-SVM and pin-SVM depends on the noise scenario under consideration. When the noisy points are sampled from the Normal or Student's t distribution with 5 degrees of freedom, the pin-SVM presents a marginally better performance than the SP-SVM. This changes when we assume a fat-tailed distribution for the noise, where we observe the SP-SVM to be producing the best results.

Further, the computational time ratios are measured for each SVM classifier as compared to the C-SVM and are reported in Table 4.3. We observe that the EEL-SVM is the fastest method as marked bold. The $\overline{pin}$-SVM, SP-SVM and pin-SVM exhibit similar running times amongst them. However, the training of the $\overline{pin}$-SVM seems to be very time-consuming when the data are contaminated with noisy points sampled from Student's t distribution with 1 degree of freedom; possibly, in these scenarios, the DCA-based solver converges slowly due to strong data contamination. Furthermore, it is noteworthy that the EEL-SVM, pin-SVM and SP-SVM have different running times, although their training involves optimization problems with the same number of variables ($3 \times N$).

TABLE 4.3
Computational time ratios of various SVM classifiers as compared to the C-SVM classifier for contaminated synthetic data.

	Normal distribution			
	$r = 0.05$		$r = 0.10$	
	$N = 100$	$N = 200$	$N = 100$	$N = 200$
pin-SVM	16.0960	37.6800	17.0162	38.7557
$\overline{pin}$-SVM	44.8931	32.1208	52.1791	36.5134
SP-SVM	12.5230	32.5378	13.6473	32.3102
EEL-SVM	**7.0742**	**15.2258**	**7.6822**	**16.0050**

	Student's t distribution 5df			
	$r = 0.05$		$r = 0.10$	
	$N = 100$	$N = 200$	$N = 100$	$N = 200$
pin-SVM	12.6435	39.1753	18.0470	45.5606
$\overline{pin}$-SVM	27.3567	29.1628	51.1566	38.9943
SP-SVM	10.9345	33.9614	13.7961	39.5174
EEL-SVM	**5.2442**	**16.5176**	**8.2361**	**19.1090**

	Student's t distribution 1df			
	$r = 0.05$		$r = 0.10$	
	$N = 100$	$N = 200$	$N = 100$	$N = 200$
pin-SVM	10.6909	37.5669	10.6909	38.2959
$\overline{pin}$-SVM	41.1003	49.6065	41.1003	67.6386
SP-SVM	8.9840	33.7658	8.9840	33.3778
EEL-SVM	**4.8769**	**15.6297**	**4.8769**	**16.1950**

4.2 Real Data Analysis

In this section, we test the performance of the EEL-SVM and SP-SVM classifiers based on real data. We perform some experiments on eight well-known real-world datasets available from the LIBSVM [28] and UCI depository[2]; their details are given in Table 4.4. It should be noted that all data have features rescaled to $[-1, 1]$. Moreover, the analysis is carried out over the original data and contaminated data. Data perturbations are introduced via the MATLAB R2019a function `awgn` with different *Signal Noise Ratios (SNR)*; perturbations are separately introduced 10 times for each dataset before training, and the average classification accuracy is reported so that the sampling error from the random generation of the noise is, to a certain extent, removed.

A random choice of the training set containing $2/3$ of a given dataset is made, with the remaining examples used for testing. All SVM methods rely on the *Radial Basis Function (RBF)* kernel chosen to calibrate the lack of linearity in the data. Once again, the SP-SVM classifier is applied for each dataset by assuming that feature uncertainty is mainly embedded within the feature with the largest standard deviation.

The classifiers' parameters are all tuned via 10-fold cross-validation. The kernel parameter γ and penalty parameter C are tuned by allowing

$$\gamma, C \in \{2^{-9}, 2^{-8}, \ldots, 2^{8}, 2^{9}\}$$

for the C-SVM, and

$$C \in \{2^{-5}, 2^{-3}, 2^{-1}, 2^{0}, 2^{1}, 2^{3}, 2^{5}\},$$
$$\gamma \in \{2^{-7}, 2^{-5}, 2^{-3}, 2^{-1}, 2^{0}, 2^{1}\}$$

for all the other SVM classifiers. Further, we consider the following values' ranges for the additional parameters:

TABLE 4.4
Specifications of all eight datasets.

	Data	# features	# train	# test
(I)	Fourclass	2	580	282
(II)	Diabetes	8	520	248
(III)	Breast cancer	10	460	223
(IV)	Australian	14	470	220
(V)	Statlog	13	180	90
(VI)	Customer	7	300	140
(VII)	Trial	17	520	252
(VIII)	Banknote	4	920	452

- $\alpha \in \mathcal{A}_{SP} = \{0.50, 0.51, \ldots, 0.56, 0.58, 0.60\}$ for SP-SVM;
- $\alpha \in \mathcal{A}_{EEL} = \{0, 0.05, 0.10, \ldots, 0.3\}$ for EEL-SVM;
- $\tau \in \mathcal{T}_{pin} = \{0.1, 0.2, \ldots, 0.8, 1\}$ for pin-SVM;
- $(\tau, s) \in \mathcal{T}_{\overline{pin}} \times \mathcal{S} = \{0, 0.25, 0.75\} \times \{0, 0.5, 1\}$ for $\overline{pin}$-SVM.

Finally, we should note that a computational budget of around 370 parameter combinations is imposed on almost all cases; only for the EEL-SVM fewer parameter combinations are tested (294 combinations).

TABLE 4.5
Classification accuracy of all five SVM classifiers for each dataset.

Data	SNR	C-SVM	pin-SVM	$\overline{pin}$-SVM	SP-SVM	EEL-SVM
(I)	NA	99.29	99.29	**99.65**	**99.65**	**99.65**
	10	99.65	99.65	99.57	99.61	**99.75**
	5	99.65	99.65	99.68	99.54	**99.72**
	1	99.54	99.61	99.61	99.50	**99.65**
(II)	NA	77.02	79.84	79.44	**80.24**	78.63
	10	76.98	76.49	78.87	**79.64**	77.26
	5	76.69	76.57	77.62	**78.02**	76.45
	1	76.49	**77.70**	77.14	77.66	74.96
(III)	NA	93.72	93.27	93.27	**94.62**	93.72
	10	93.90	**94.75**	94.26	93.32	94.44
	5	93.86	**94.57**	94.04	94.13	94.08
	1	93.81	93.86	93.41	93.86	**94.04**
(IV)	NA	88.64	88.18	**89.55**	88.18	89.09
	10	**85.82**	85.23	85.32	85.45	85.32
	5	**80.68**	80.50	80.64	80.59	78.45
	1	76.59	75.86	**77.55**	76.14	76.23
(V)	NA	82.22	82.22	82.22	**83.33**	78.89
	10	80.22	80.56	81.22	80.22	**82.44**
	5	80.00	79.33	79.11	79.33	**81.33**
	1	78.67	78.22	**79.56**	78.44	76.89
(VI)	NA	92.14	91.43	92.14	92.14	**92.86**
	10	92.86	92.50	**93.36**	92.71	93.21
	5	92.93	92.86	91.29	**93.07**	**93.07**
	1	92.57	**92.93**	90.14	92.57	92.86
(VII)	NA	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	10	99.56	**99.80**	99.29	99.76	99.60
	5	94.72	94.60	94.52	**94.84**	94.52
	1	88.13	88.13	**88.41**	88.29	88.21
(VIII)	NA	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	10	99.73	99.71	99.78	99.76	**99.89**
	5	**99.38**	99.18	**99.38**	99.25	99.36
	1	97.94	97.92	**98.01**	97.94	97.83

Table 4.5 presents the classification performance for all five SVM-like methods applied to the eight real-world datasets (reported as NA) and their various contaminated variants with SNR values $\{1, 5, 10\}$ — a smaller SNR value means a higher degree of data contamination — and the highest accuracy amongst the five classifiers is marked bold. The EEL-SVM achieves good results at this stage with best performance in $12/32$ scenarios investigated; both the SP-SVM and $\overline{pin}$-SVM produce the best accuracy in $10/32$

[2] for details, see https://archive.ics.uci.edu/ml/index.php

cases, whereas the pin-SVM and C-SVM appear to be less accurate with best performances in only $7/32$ and $5/32$ scenarios, respectively. Overall, the EEL-SVM and SP-SVM methods have performed well on the real datasets considered.

5 Conclusions and Recommendations

This paper addresses the problem of binary classification in the data uncertainty context. Two robust SVM-style classification algorithms are developed and discussed: the SP-SVM and EEL-SVM. A large set of numerical experiments has been conducted to check their effectiveness on synthetic and real-world datasets, both with and without noise contamination. Their performances have been compared against the classical C-SVM and two well-known robust SVM formulations from the literature, the pin-SVM and $\overline{pin}$-SVM.

Our results highlight that both our newly proposed methods are promising alternatives to those currently available in the literature. The SP-SVM achieved good results in all our experiments, especially on synthetic data with strong noise contamination. The EEL-SVM did not produce high-quality performances in the synthetic experiments, but it appeared an accurate classification tool on real-world datasets. Besides, the training of these two methods involves optimisation problems that can be solved efficiently in less time than the other SVM methods.

Future research will focus on the development of a new method that combines the SP-SVM and EEL-SVM paradigms but also extend these to multiclass classification problems.

6 Appendix

6.1 Proof of Theorem 2.1

It is sufficient to show that (2.4) holds when $p > q$ and $p < q$, where

$$p := \Pr\big(Y = 1|\mathbf{x}\big) \quad \text{and} \quad q := \Pr\big(Y = -1|\mathbf{x}\big),$$

that is,

$$\begin{aligned} \operatorname*{argmin}_{z\in\Re} \mathbf{E}_{\mathcal{Y}|\mathbf{x}} L\Big(1-Yz\Big) &= \operatorname*{argmin}_{z\in\Re} pL(1-z) + qL(1+z) \\ &= \begin{cases} 1, & \text{if} \quad p > q, \\ -1, & \text{if} \quad p < q. \end{cases} \end{aligned} \tag{6.1}$$

Note, first, that $L(1 \pm \cdot)$ are compositions with affine mappings of the convex function L, therefore the objective function in (6.1) is convex as well. Moreover, the left and right derivatives of L exist as the loss function is convex.

Assume first that $p > q$. The left and right derivatives at 1 of the objective function in (6.1) are $-pL'(0^+) + qL'(2^-)$ and $-pL'(0^-) + qL'(2^+)$, respectively. Clearly,

$$-pL'(0^+) + qL'(2^-) = L'(0^+)(q-p) \le 0$$

is true as L is linear on $(0, 2+\epsilon)$ for some $\epsilon > 0$. Further,

$$-pL'(0^-) + qL'(2^+) \ge 0$$

also holds due to the fact that $L'(0^-) \le 0 \le L'(2^+)$, which is a consequence of the convexity of L that attains its global minimum at 0. Thus, the global minimum of the convex function in (6.1) is attained at 1 whenever $p > q$.

Assume now that $p < q$. Similarly, the left and right derivatives at -1 of the objective function in (6.1) are $-pL'(2^+) + qL'(0^-)$ and $-pL'(2^-) + qL'(0^+)$, respectively. Clearly, $-pL'(2^+) + qL'(0^-) \le 0$ holds as $L'(0^-) \le 0 \le L'(2^+)$ and L is convex that attains its global minimum at 0. Further, $-pL'(2^-) + qL'(0^+) = L'(0^+)(q-p) \ge 0$ is true as L is linear on $(0, 2+\epsilon)$ for some $\epsilon > 0$. Thus, the global minimum of the convex function in (6.1) is attained at -1 whenever $p < q$. The proof is now complete.

6.2 Solution Identification to (3.4)

Let $\phi_j(\mathbf{x}_i)$ be the j^{th} element of $\phi(\mathbf{x}_i)$. Denote by $\phi_1(\mathbf{x}_i)$ and $\phi_2(\mathbf{x}_i)$ two vectors with their j^{th} elements given by $\phi_{1j}(\mathbf{x}_i) = \phi_j(\mathbf{x}_i) - a_{ik}I_{j=k}$ and $\phi_{2j}(\mathbf{x}_i) = \phi_j(\mathbf{x}_i) + a_{ik}I_{j=k}$ for all $1 \le i \le N$ and $1 \le j \le d$, where I_A is the indicator of set A that takes the values 1 or 0 if A is true or false, respectively. Thus, (3.4) can be written as

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i\Big(\mathbf{w}^T\phi(\mathbf{x}_i) + b\Big) \ge 1 - \xi_i, \\ & y_i\Big(\mathbf{w}^T\phi_1(\mathbf{x}_i) + b\Big) \ge 1 - \xi_i, \\ & y_i\Big(\mathbf{w}^T\phi_2(\mathbf{x}_i) + b\Big) \ge 1 - \xi_i, \\ & \xi_i \ge 0, \quad 1 \le i \le N. \end{aligned} \tag{6.2}$$

It should be noted that the above is a convex optimization problem that has only affine constraints, therefore the strong duality holds. The Lagrangian of (6.2) is given by

$$\begin{aligned} & H\big(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}\big) \\ &:= \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\delta_i\xi_i \\ &\quad - \sum_{i=1}^{N}\alpha_i\Big(y_i\Big(\mathbf{w}^T\phi(\mathbf{x}_i) + b\Big) - 1 + \xi_i\Big) \\ &\quad - \sum_{i=1}^{N}\beta_i\Big(y_i\Big(\mathbf{w}^T\phi_1(\mathbf{x}_i) + b\Big) - 1 + \xi_i\Big) \\ &\quad - \sum_{i=1}^{N}\gamma_i\Big(y_i\Big(\mathbf{w}^T\phi_2(\mathbf{x}_i) + b\Big) - 1 + \xi_i\Big), \end{aligned} \tag{6.3}$$

where the dual variables satisfy $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta} \ge \mathbf{0}$. By virtue of the strong duality property, we first construct the dual formulation of (6.2) by imposing:

$$\begin{aligned} \mathbf{0} &= \frac{\partial H}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N}\Big(\alpha_i y_i \phi(\mathbf{x}_i) + \beta_i y_i \phi_1(\mathbf{x}_i) + \gamma_i y_i \phi_2(\mathbf{x}_i)\Big), \\ 0 &= \frac{\partial H}{\partial b} = -\mathbf{y}^T\boldsymbol{\alpha} - \mathbf{y}^T\boldsymbol{\beta} - \mathbf{y}^T\boldsymbol{\gamma}, \\ \mathbf{0} &= \frac{\partial H}{\partial \boldsymbol{\xi}} = C\mathbf{1} - \boldsymbol{\alpha} - \boldsymbol{\beta} - \boldsymbol{\gamma} - \boldsymbol{\delta}. \end{aligned}$$

Therefore, the dual of (6.2) is given by

$$\begin{array}{ll} \max\limits_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta}} & -\frac{1}{2}\big[\boldsymbol{\alpha}\ \boldsymbol{\beta}\ \boldsymbol{\gamma}\big]^T\ \mathbf{T}\ \big[\boldsymbol{\alpha}\ \boldsymbol{\beta}\ \boldsymbol{\gamma}\big] \\ & \qquad +\mathbf{1}^T\boldsymbol{\alpha} + \mathbf{1}^T\boldsymbol{\beta} + \mathbf{1}^T\boldsymbol{\gamma} \\ \text{s.t.} & \boldsymbol{\alpha} \geq \mathbf{0}, \boldsymbol{\beta} \geq \mathbf{0}, \boldsymbol{\gamma} \geq \mathbf{0}, \boldsymbol{\delta} \geq \mathbf{0}, \\ & \boldsymbol{\alpha} + \boldsymbol{\beta} + \boldsymbol{\gamma} + \boldsymbol{\delta} = C\mathbf{1}, \\ & \mathbf{y}^T\boldsymbol{\alpha} + \mathbf{y}^T\boldsymbol{\beta} + \mathbf{y}^T\boldsymbol{\gamma} = 0, \end{array} \tag{6.4}$$

where the block matrix **T** is given by

$$\mathbf{T} = \left[\begin{array}{c|c|c} \mathbf{T}^{\phi,\phi} & \mathbf{T}^{\phi,\phi_1} & \mathbf{T}^{\phi,\phi_2} \\ \hline \mathbf{T}^{\phi_1,\phi} & \mathbf{T}^{\phi_1,\phi_1} & \mathbf{T}^{\phi_1,\phi_2} \\ \hline \mathbf{T}^{\phi_2,\phi} & \mathbf{T}^{\phi_2,\phi_1} & \mathbf{T}^{\phi_2,\phi_2} \end{array}\right]$$

with $\mathbf{T}^{\varphi_1,\varphi_2}$ an $N \times N$ matrix with the $(i,j)^{th}$ entry given by $y_i\varphi_1^T(\mathbf{x}_i)\varphi_2(\mathbf{x}_i)y_j$ for all $\varphi_1, \varphi_2 \in \{\phi, \phi_1, \phi_2\}$ and $1 \leq i, j \leq N$. Clearly, (6.4) is equivalent to solving the LCQP:

$$\begin{array}{ll} \min\limits_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}} & \frac{1}{2}\big[\boldsymbol{\alpha}\ \boldsymbol{\beta}\ \boldsymbol{\gamma}\big]^T\ \mathbf{T}\ \big[\boldsymbol{\alpha}\ \boldsymbol{\beta}\ \boldsymbol{\gamma}\big] - \mathbf{1}^T\boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\beta} - \mathbf{1}^T\boldsymbol{\gamma} \\ \text{s.t.} & \mathbf{0} \leq \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \\ & \boldsymbol{\alpha} + \boldsymbol{\beta} + \boldsymbol{\gamma} \leq C\mathbf{1}, \\ & \mathbf{y}^T\boldsymbol{\alpha} + \mathbf{y}^T\boldsymbol{\beta} + \mathbf{y}^T\boldsymbol{\gamma} = 0. \end{array} \tag{6.5}$$

Let $\big(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*, \boldsymbol{\gamma}^*\big)$ be an optimal solution of (6.5), which helps in finding an optimal solution of the primal instance in (3.4), which, in turn, gives the classification rule identified by $\mathbf{w}^*$ and b^* that are due to be found. Clearly,

$$\mathbf{w}^* := \sum_{i=1}^N \Big(\alpha_i^* y_i \phi(\mathbf{x}_i) + \beta_i^* y_i \phi_1(\mathbf{x}_i) + \gamma_i^* y_i \phi_2(\mathbf{x}_i)\Big).$$

The choice of b^* is possible by considering the complementary slackness conditions related to (6.3); a sensible estimate of b^* is

$$\widehat{b^*} := \frac{\widehat{b_l^+}}{|\mathcal{S}_l|},$$

where $|\cdot|$ represents the set cardinality, $\mathcal{S}_l$ is the set with the largest set cardinality among

$$\begin{aligned} \mathcal{S}_0 &:= \big\{1 \leq i \leq N:\ \alpha_i^*(C - \alpha_i - \beta_i - \gamma_i) > 0\big\}, \\ \mathcal{S}_1 &:= \big\{1 \leq i \leq N:\ \beta_i^*(C - \alpha_i - \beta_i - \gamma_i) > 0\big\}, \\ \mathcal{S}_2 &:= \big\{1 \leq i \leq N:\ \gamma_i^*(C - \alpha_i - \beta_i - \gamma_i) > 0\big\} \end{aligned}$$

and

$$\begin{aligned} \widehat{b_l^+} := \sum_{j \in \mathcal{S}_l} y_j &- \sum_{j \in \mathcal{S}_l}\sum_{i=1}^N \alpha_i^* y_i \phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j) \\ &- \sum_{j \in \mathcal{S}_l}\sum_{i=1}^N \beta_i^* y_i \phi_1^T(\mathbf{x}_i)\phi(\mathbf{x}_j) \\ &- \sum_{j \in \mathcal{S}_l}\sum_{i=1}^N \gamma_i^* y_i \phi_2^T(\mathbf{x}_i)\phi(\mathbf{x}_j). \end{aligned}$$

6.3 Solution Identification to (3.7)

The derivations in this section are quite similar to those in Section 6.2, thus we provide only the main steps. We first note that the convex optimization problem (3.7) has only affine constraints, therefore the strong duality holds. Now, the Lagrangian of (3.7) is given by

$$\begin{aligned} &H\big(\mathbf{w}, b, \boldsymbol{\xi}, z, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\big) \qquad (6.6) \\ &\quad := \frac{1}{2}\mathbf{w}^T\mathbf{w} + Dz + \sum_{i=1}^N (D_1 - \beta_i - \gamma_i)\xi_i - z\sum_{i=1}^N \beta_i \\ &\qquad - \sum_{i=1}^N \alpha_i\Big(y_i\Big(\mathbf{w}^T\phi(\mathbf{x}_i) + b\Big) + z - 1 + \xi_i\Big), \end{aligned}$$

where $D_1 := \frac{D}{N(1-\alpha)}$; moreover, the dual variables satisfy $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \geq \mathbf{0}$. On account of the strong duality property, we only need to solve the dual of (3.7), thus we set

$$\begin{aligned} &\mathbf{0} = \frac{\partial H}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^N \alpha_i y_i \phi(\mathbf{x}_i), \quad 0 = \frac{\partial H}{\partial b} = -\mathbf{y}^T\boldsymbol{\alpha}, \\ &\mathbf{0} = \frac{\partial H}{\partial \boldsymbol{\xi}} = D_1\mathbf{1} - \boldsymbol{\alpha} - \boldsymbol{\beta} - \boldsymbol{\gamma}, \quad 0 = \frac{\partial H}{\partial z} = D - \mathbf{1}^T\boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\beta}. \end{aligned}$$

One may show that the dual of (3.7) is equivalent to solving the LCQP:

$$\begin{array}{ll} \min\limits_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}} & \frac{1}{2}\boldsymbol{\alpha}\ \mathbf{T}^{\phi,\phi}\ \boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\alpha} \\ \text{s.t.} & \mathbf{0} \leq \boldsymbol{\alpha},\ \mathbf{0} \leq \boldsymbol{\beta},\ \mathbf{0} \leq \boldsymbol{\gamma}, \\ & \boldsymbol{\alpha} + \boldsymbol{\beta} + \boldsymbol{\gamma} = D_1\mathbf{1}, \\ & \mathbf{y}^T\boldsymbol{\alpha} = 0, \\ & \mathbf{1}^T\boldsymbol{\alpha} + \mathbf{1}^T\boldsymbol{\beta} = D, \end{array} \tag{6.7}$$

where $\mathbf{T}^{\phi,\phi}$ is as defined in Section 6.2.

Let $\big(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*, \boldsymbol{\gamma}^*\big)$ be an optimal solution of (6.7). Then, (3.7) is solved with

$$\mathbf{w}^* = \sum_{i=1}^N \alpha_i^* y_i \phi(\mathbf{x}_i).$$

Finally, the bias term b^* can be estimated as

$$\big(\widehat{b^*}, \widehat{z^*}\big) := \begin{cases} \big(\widehat{b^{*1}}, 0\big) & \text{if} \quad |\mathcal{S}_4| \leq |\mathcal{S}_3|, \\ \big(\widehat{b^{*2}}, 0\big) & \text{if} \quad |\mathcal{S}_4| > |\mathcal{S}_3|, \end{cases}$$

where

$$\widehat{b^{*1}} = \frac{\widehat{b_3^+}}{|\mathcal{S}_3|}, \quad \widehat{b_3^+} = \sum_{j \in \mathcal{S}_3} y_j - \sum_{j \in \mathcal{S}_3}\sum_{i=1}^N \alpha_i^* y_i \phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j),$$

$$\widehat{b^{*2}} = \frac{\widehat{b_4^+}}{|\mathcal{S}_4|}, \quad \widehat{b_4^+} = \sum_{j \in \mathcal{S}_4} y_j - \sum_{j \in \mathcal{S}_4}\sum_{i=1}^N \alpha_i^* y_i \phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j)$$

and

$$\begin{aligned} \mathcal{S}_3 &:= \big\{1 \leq i \leq N:\ \alpha_i^*\beta_i^*\gamma_i^* > 0\big\} \\ \mathcal{S}_4 &:= \big\{1 \leq i \leq N:\ \alpha_i^*\beta_i^* > 0, \gamma_i^* = 0\big\}. \end{aligned}$$

References

[1] R. K. Jayadeva and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 29, no. 5, pp. 905–910, 2007.

[2] C. Cortes and V. N. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[3] V. N. Vapnik, *The nature of statistical learning theory, second edition*. Springer, New York, 2000.

[4] T. Zhang, "Statistical analysis of some multi-category large margin classification methods," *Journal of Machine Learning Research*, vol. 5, no. Oct, pp. 1225–1251, 2004.

[5] X. Shen, L. Niu, Z. Qi, and Y. Tian, “Support vector machine classifier with truncated pinball loss,” *Pattern Recognition*, vol. 68, pp. 199–210, 2017.

[6] Y. Wu and Y. Liu, “Robust truncated hinge loss support vector machines,” *Journal of the American Statistical Association*, vol. 102, no. 479, pp. 974–983, 2007.

[7] X. Huang, L. Shi, and J. A. Suykens, “Support vector machine classifier with pinball loss,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 984–997, 2014.

[8] J. A. Suykens and J. Vandewalle, “Least squares support vector machine classifiers,” *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[9] A. Singh, R. Pokharel, and J. Principe, “The c-loss function for pattern classification,” *Pattern Recognition*, vol. 47, no. 1, pp. 441–453, 2014.

[10] G. Xu, Z. Cao, B.-G. Hu, and J. C. Principe, “Robust support vector machines based on the rescaled hinge loss function,” *Pattern Recognition*, vol. 63, pp. 139–148, 2017.

[11] L. W. Huang, Y. H. Shao, J. Zhang, Y. T. Zhao, and J. Y. Teng, “Robust rescaled hinge loss twin support vector machine for imbalanced noisy classification,” *IEEE Access*, vol. 7, pp. 65 390–65 404, 2019.

[12] Y. Lin, “A note on margin-based loss functions in classification,” *Statistics & Probability Letters*, vol. 68, no. 1, pp. 73–82, 2004.

[13] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, “Convexity, classification, and risk bounds,” *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 138–156, 2006.

[14] Y. Lin, “Support vector machines and the bayes rule in classification,” *Data Mining and Knowledge Discovery*, vol. 6, no. 3, pp. 259–275, 2002.

[15] J. Bi and T. Zhang, “Support vector classification with input data uncertainty,” in *Advances in Neural Information Processing Systems*, 2005, pp. 161–168.

[16] G. Lanckriet, L. El Ghaoui, C. Bhattacharyya, and M. Jordan, “A robust minimax approach to classification,” *Journal of Machine Learning Research*, vol. 3, pp. 555–582, 2002.

[17] P. Shivaswamy, C. Bhattacharyya, and A. Smola, “Robust support vector regression for uncertain input and output data,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 11, pp. 1690–1700, 2012.

[18] J. Nath and C. Bhattacharyya, “Maximum margin classifiers with specific false positive and false negative error rates,” *Proceedings of the 2007 SIAM International Conference on Data Mining*, pp. 35–46, 2007.

[19] G. Huang, S. Song, J. Gupta, and C. Wu, “A second order cone programming approach for semi-supervised learning,” *Pattern Recognition*, vol. 46, no. 12, pp. 3548–3558, 2013.

[20] G. Huang, S. Song, C. Wu, and K. You, “Robust support vector regression for uncertain input and output data,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 11, pp. 1690–1700, 2012.

[21] H. Xu, C. Caramanis, and S. Mannor, “Robustness and regularization of support vector machines,” *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1485–1510, 2009.

[22] A. Asimit, V. Bignozzi, K. Cheung, J. Hu, and E. Kim, “Robust and pareto optimality of insurance contracts,” *European Journal of Operational Research*, vol. 262, no. 2, pp. 720—732, 2017.

[23] P. J. Huber, “Robust estimation of a location parameter,” *The Annals of Mathematical Statistics*, pp. 73–101, 1964.

[24] F. R. Hampel, “A general qualitative definition of robustness,” *The Annals of Mathematical Statistics*, pp. 1887–1896, 1971.

[25] P. J. Huber and E. Ronchetti, *Robust Statistics, second edition*. John Wiley & Sons, New Jersey, 2009.

[26] K. Fang, S. Kotz, and K. Ng, *Symmetric multivariate and related distributions*. Chapman and Hall, London, 1990.

[27] R. Rockafellar and S. Uryasev, “Optimization of conditional value-at-risk,” *Journal of Risk*, vol. 2, no. 3, pp. 21—42, 2000.

[28] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.