Computers & Operations Research 170 (2024) 106755

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

Distributionally robust chance-constrained kernel-based support vector machine

Fengming Lin [a,*], Shu-Cherng Fang [a], Xiaolei Fang [a], Zheming Gao [b]

[a] Edward P. Fitts Department of Industrial and Systems Engineering, North Carolina State University, Raleigh, NC 27606, USA
[b] College of Information Science and Engineering, Northeastern University, Shenyang, Liaoning 110819, China

ARTICLE INFO

Keywords:
Uncertainty
Support vector machine
Distributionally robust optimization
Data-driven approach
ADMM

ABSTRACT

Support vector machine (SVM) is a powerful model for supervised learning. This article addresses the nonlinear binary classification problem using kernel-based SVM with uncertainty involved in the input data specified by the first- and second-order moments. To achieve a robust classifier with small probabilities of misclassification, we investigate a distributionally robust chance-constrained kernel-based SVM model. Since the moment information in the original problem becomes unclear/unavailable in the feature space via kernel transformation, we develop a data-driven approach utilizing empirical moments to provide a second-order cone programming (SOCP) reformulation for efficient computation. To speed up the required computations for solving large-size problems in higher dimensional space and/or with more sampling points involved in estimating empirical moments, we further design an alternating direction multipliers-based algorithm for fast computations. Extensive computational results support the effectiveness and efficiency of the proposed model and solution method. Results on public benchmark datasets without any moment information indicate that the proposed approach still works and, surprisingly, outperforms some commonly used state-of-the-art kernel-based SVM models.

1. Introduction

Support vector machines (SVMs) have been extensively studied and widely used for data classification. SVM aims to find a maximum-margin hyperplane that separates the data points into different classes (Cortes and Vapnik, 1995). When the datasets are non-linearly separable, a feature map is usually adopted to lift the data points to a higher dimensional feature space where it is more likely to be linearly separable. This is typically achieved by using kernel tricks, which yield the kernel-based SVM, a powerful tool for nonlinear classification (Vapnik, 1999; Carrizosa and Morales, 2013). Recently, kernel-free nonlinear SVMs have also been studied and shown attractive performance (Luo et al., 2016; Gao et al., 2021).

The success of standard SVMs relies on the assumption that the input data pertaining to the classification task are known exactly. However, real-world data often involve uncertainties due to imprecise data collection and inaccurate measurements during data gathering. Failing to acknowledge these uncertainties may result in significant classification performance degeneration (Goldfarb and Iyengar, 2003). To address this challenge, robust optimization-based SVM models are developed for applications whose data points are fluctuating within an uncertain set, specified by the norm uncertainty (Trafalis and Gilbert, 2006), ellipsoidal uncertainty (Bhattacharyya et al., 2004), and others (Bertsimas et al., 2019; Singla et al., 2020). In general, these robust models tend to be on the conservative side since they ignore the hidden distribution information embedded in the data sets.

To handle the uncertainties characterized by data distributions, models incorporating chance constraints are often utilized to ensure a minimal probability of misclassification for SVM models. Chance-constrained problems are usually challenging to solve and often require time-consuming Monte Carlo approximations. For example, Peng et al. (2023) proposed a Monte Carlo-based approximation method that empirically estimates the actual distribution using training data. It is noteworthy that finding an accurate estimation of the true distribution is challenging. Also, finding a well-estimated distribution may still be susceptible to the "optimizer's curse" (Kuhn et al., 2019), which leads to unsatisfactory performance. In an effort to address these issues, researchers consider using distributionally robust optimization (DRO) to hedge against distributional uncertainty (Lin et al., 2022).

Rather than relying on a single estimate of the distribution, DRO characterizes the distributional uncertainty by adopting the ambiguity set, which consists of a collection of distributions based on given prior information on the uncertainty. Distance-based DRO models have

* Corresponding author.
E-mail addresses: flin6@ncsu.edu (F. Lin), fang@ncsu.edu (S.-C. Fang), xfang8@ncsu.edu (X. Fang), gaozheming@ise.neu.edu.cn (Z. Gao).

https://doi.org/10.1016/j.cor.2024.106755
Received 19 May 2023; Received in revised form 24 June 2024; Accepted 25 June 2024
Available online 3 July 2024

been studied for handling different machine learning tasks (Duchi and Namkoong, 2019, 2021; Staib and Jegelka, 2019). Recently, the Wasserstein DRO models have been rigorously investigated for machine learning tasks (Kuhn et al., 2019; Liu et al., 2022). Specifically, the SVM models with Wasserstein ambiguity sets were investigated in Lee and Mehrotra (2015), Shafieezadeh-Abadeh et al. (2019). Another major way to characterize the ambiguity sets can be constructed with generalized moment conditions. Multiple DRO-based classification models (Lanckriet et al., 2001; Wang and Pardalos, 2014) have been proposed by utilizing the moment information. This paper specifically aims to explore the efficacy of distributionally robust chance-constrained (DRC) SVM for solving binary classification problems with uncertain input data points specified by the first- and second-order moments information. In this context, DRC linear SVM models have been well studied (Ben-Tal et al., 2011; Wang et al., 2018; Khanjani-Shiraz et al., 2023; Faccini et al., 2022). In particular, Chebyshev inequality (Marshall and Olkin, 1960) was used in Wang et al. (2018), Khanjani-Shiraz et al. (2023) to yield a second-order cone programming (SOCP) problem whose solution is guaranteed to satisfy the distributionally robust chance constraints. Moreover, Ben-Tal et al. (2011) employed the Bernstein bounding scheme to develop a less conservative SOCP reformulation. Such DRC SVM models with tractable SOCP reformulations/approximations were also applied to classification tasks involving missing values in the data (Shivaswamy et al., 2006), as well as unsupervised classification (Huang et al., 2013) where a kernelized formulation was also discussed. With the promising performance of the DRC SVM models, we intend to conduct a study of kernel-based DRC SVM models for nonlinear classification.

The nature of DRO brings challenges for implementation, especially when applied to solve large-scale problems (Cheramin et al., 2022). In recent research, the alternating direction multiplier method (ADMM) (Boyd et al., 2011) has been frequently utilized to solve DRO models. Li et al. (2019), Jiajin (2021) investigated the performance of multiple ADMM variants for solving Wasserstein distributionally robust logistic regression models. Ohmori (Ohmori, 2021) solved large-scale ϕ-divergence based DRO models with a distributed optimization algorithm that use consensus ADMM. In addition, the ADMM-based algorithms have also been applied to implement the DRO models in real-world applications such as integrated transmission-distribution systems (Zhai et al., 2022), power plants operations (Esfahani et al., 2024) and energy trading (Mohseni and Pishvaee, 2023; Zhang et al., 2023). In this paper, we design a fast ADMM algorithm to efficiently implement the corresponding SOCP reformulations of the proposed model.

The research presented in this paper contributes to the study of DRC kernel-based SVMs, with a specific focus on binary classification involving uncertain input data characterized by first- and second-order moments. The primary contributions are summarized as the following:

- We propose a DRC kernel-based SVM model for providing a robust classifier for nonlinear classification with stochastic input using the first- and second-order moments information.
- We develop a data-driven approach utilizing an assured empirical moment estimation in the higher dimensional feature space to provide a tractable SOCP reformulation for solving the proposed DRC kernel-based SVM model. This approach addresses the difficulty of missing corresponding moment information in the feature space.
- We design an ADMM-based algorithm for our data-driven SOCP, which significantly improves the computational efficiency compared to using commercial solvers, especially for large-scale problems.
- Computational experiments support the effectiveness of the proposed DRC kernel-based model and the computation efficiency of the proposed solution method. Results on public benchmark data sets without any given moment information show the promising performance of the proposed model over classical kernel-based SVMs.

The rest of the paper is organized as follows. Section 2 presents the proposed distributionally robust chance-constrained SVM model using kernel tricks for nonlinear classification with uncertain input data specified by the first- and second-order moments. An SOCP reformulation based on a data-driven approach using assured empirical moments for tractable computation is also included. Section 3 proposes an ADMM-based algorithm for solving the corresponding SOCP reformulation with fast computations. Extensive computational experiments reported in Section 4 evaluate the performance of the proposed DRC kernel-based model and validate the efficiency of the proposed solution method. Finally, conclusions and future works are provided in Section 5.

2. Distributionally robust chance-constrained kernel-based SVM

To address the problem of classifying nonlinearly separable data with uncertain input data points with only the first- and second-order moments being known, this section presents a distributionally robust chance-constrained kernel-based support vector machine model, denoted as DRCKSVM. Section 2.1 proposes the DRCKSVM model and provides an equivalent SOCP reformulation relying on the moments in the higher dimensional feature space. To address the challenge that the transformed moments are difficult to know exactly, Section 2.2 proposes a data-driven approach that employs the empirical moment estimation with assured quality to implement the DRCKSVM model.

2.1. DRC kernel-based SVM model

Given a set of N data points with n attributes $\{(\tilde{\boldsymbol{x}}^i, y^i)|\tilde{\boldsymbol{x}}^i \in \mathbb{R}^n,\ y^i \in \{-1,1\}, i=1,\dots,N\}$, we assume that each input data point is a random variable, i.e., $\tilde{\boldsymbol{x}}^i \sim F_i$, where F_i is a probability measure on $(\Xi_i, \mathcal{F}_i)$, for a given outcome space Ξ_i and its σ-algebra $\mathcal{F}_i \subseteq 2^{\Xi_i}$. That is, $F_i : \mathcal{F}_i \to \mathbb{R}$, and $F_i \in \mathcal{M}(\Xi_i, \mathcal{F}_i)$, the space of all probability measures defined on $(\Xi_i, \mathcal{F}_i)$. Let F_i be mutually independent for $i = 1,\dots,N$. Assume that the true distribution F_i is unknown, but its first two moments are known a priori, that is, mean $\boldsymbol{\mu}^i \triangleq \mathbb{E}_{F_i}[\tilde{\boldsymbol{x}}^i]$ and covariance matrix $\boldsymbol{\Sigma}^i \triangleq \mathbb{E}_{F_i}[(\tilde{\boldsymbol{x}}^i - \mathbb{E}_{F_i}[\tilde{\boldsymbol{x}}^i])(\tilde{\boldsymbol{x}}^i - \mathbb{E}_{F_i}[\tilde{\boldsymbol{x}}^i])^{\mathrm{T}}]$. We consider that F_i belongs to an ambiguous distribution family $\mathcal{P}_i$ defined by the two moments as

$$\mathcal{P}_i(\tilde{\boldsymbol{x}}^i; \boldsymbol{\mu}^i, \boldsymbol{\Sigma}^i) \triangleq \left\{ F_i \in \mathcal{M}(\Xi_i, \mathcal{F}_i) \;\middle|\; \begin{array}{l} \mathbb{P}_{F_i}(\tilde{\boldsymbol{x}}^i \in \Xi_i) = 1, \\ \mathbb{E}_{F_i}[\tilde{\boldsymbol{x}}^i] = \boldsymbol{\mu}^i, \\ \mathbb{E}_{F_i}[(\tilde{\boldsymbol{x}}^i - \boldsymbol{\mu}^i)(\tilde{\boldsymbol{x}}^i - \boldsymbol{\mu}^i)^{\mathrm{T}}] = \boldsymbol{\Sigma}^i \end{array} \right\}. \tag{1}$$

For nonlinearly separable datasets, a nonlinear transformation $\phi(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^d$ is first applied to map each data point $\tilde{\boldsymbol{x}}^i$ from the original space $\mathbb{R}^n$ to a higher-dimensional feature space $\mathbb{R}^d$, where $d \geq n$. To ensure the probability of misclassification under all possible distributions to be no larger than ϵ $(0 < \epsilon < 1)$, we can consider the following distributionally robust chance-constrained kernel-based SVM model:

$$\begin{array}{ll} \min & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i \\ s.t. & \sup_{F_i \in \mathcal{P}_i} \mathbb{P}_{F_i}\left\{ y_i\left(\boldsymbol{w}^{\mathrm{T}}\phi(\tilde{\boldsymbol{x}}^i) + b\right) \leq 1 - \xi_i \right\} \leq \epsilon,\ i = 1,\dots,N, \\ & \boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}_+^N, \end{array} \tag{DRCKSVM}$$

where $C > 0$ is a given parameter. If the first- and second-order moments of the mapped data $\phi(\tilde{\boldsymbol{x}}^i) \in \mathbb{R}^d$ are known exactly, denoted as mean $\boldsymbol{\mu}_\phi^i \in \mathbb{R}^d$ and covariance $\boldsymbol{\Sigma}_\phi^i \in \mathbb{S}_{++}^d$, respectively, the (DRCKSVM) model can be equivalently reformulated as an SOCP problem (following directly from Shivaswamy et al., 2006; Ben-Tal et al., 2011; Wang et al., 2018):

$$\begin{array}{ll} \min & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i \\ s.t. & y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_\phi^i + b\right) \geq 1 - \xi_i + \tau(\epsilon)\|(\boldsymbol{\Sigma}_\phi^i)^{\frac{1}{2}}\boldsymbol{w}\|_2,\ i = 1,\dots,N, \\ & \boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}_+^N, \end{array} \tag{DRCKSVM$_{SOCP}$}$$

where $\tau(\epsilon) = \sqrt{\frac{1-\epsilon}{\epsilon}}$. Notice that the constraints in (DRCKSVM$_{SOCP}$) can be explained from a geometric viewpoint here. Assume that $\boldsymbol{z} \in \mathbb{R}^d$ takes values within an ellipsoid with the center $\boldsymbol{\mu}_\phi^i$, metric $\boldsymbol{\Sigma}_\phi^i$, and radius r, i.e.,

$$\begin{aligned}\boldsymbol{z} \in \mathcal{E}\left(\boldsymbol{\mu}_\phi^i, \boldsymbol{\Sigma}_\phi^i, r\right) &\triangleq \left\{\boldsymbol{z} \in \mathbb{R}^d \mid (\boldsymbol{z}-\boldsymbol{\mu}_\phi^i)^{\mathrm{T}}(\boldsymbol{\Sigma}_\phi^i)^{-1}(\boldsymbol{z}-\boldsymbol{\mu}_\phi^i) \le r^2\right\} \\ &= \left\{\boldsymbol{z} \in \mathbb{R}^d \mid \boldsymbol{z} = \boldsymbol{\mu}_\phi^i + r(\boldsymbol{\Sigma}_\phi^i)^{\frac{1}{2}}\boldsymbol{u}, \|\boldsymbol{u}\|_2 \le 1\right\}.\end{aligned}$$

Then, for each i, the constraint in the (DRCKSVM$_{SOCP}$) model becomes

$$y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{z}^i + b\right) \ge 1 - \xi_i, \ \forall\ \boldsymbol{z}^i \in \mathcal{E}\left(\boldsymbol{\mu}_\phi^i, \boldsymbol{\Sigma}_\phi^i, \tau(\epsilon)\right). \tag{2}$$

This equivalency implies that the chance constraints in (DRCKSVM) turn out to be separating the ellipsoids $\mathcal{E}\left(\boldsymbol{\mu}_\phi^i, \boldsymbol{\Sigma}_\phi^i, \tau(\epsilon)\right)$ in the feature space, for $i = 1, \ldots, N$. Now we consider its dual problem.

Lemma 1. *The Lagrange dual of* (DRCKSVM$_{SOCP}$) *is*

$$\begin{aligned}\min\quad & \tfrac{1}{2}\sum_{i=1}^N\sum_{j=1}^N \alpha_i y_i\left(\boldsymbol{\mu}_\phi^i + \tau(\epsilon)(\boldsymbol{\Sigma}_\phi^i)^{\frac{1}{2}}\boldsymbol{u}^i\right)^{\mathrm{T}}\left(\boldsymbol{\mu}_\phi^j + \tau(\epsilon)(\boldsymbol{\Sigma}_\phi^j)^{\frac{1}{2}}\boldsymbol{u}^j\right)\alpha_j y_j - \sum_{i=1}^N \alpha_i \\ s.t.\quad & \sum_{i=1}^N y_i\alpha_i = 0, \\ & \|\boldsymbol{u}^i\|_2 \le 1, \qquad i = 1, \ldots, N, \\ & 0 \le \alpha_i \le C, \qquad i = 1, \ldots, N, \\ & \boldsymbol{\alpha} \in \mathbb{R}^N, \boldsymbol{u}^i \in \mathbb{R}^d, \quad i = 1, \ldots, N.\end{aligned}$$

(Dual-DRCKSVM$_{SOCP}$)

Proof. The Lagrangian of (DRCKSVM$_{SOCP}$) is given by

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) =& \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^N \xi_i \\ & - \sum_{i=1}^N \alpha_i\left(y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_\phi^i + b) - 1 + \xi_i - \tau(\epsilon)\|(\boldsymbol{\Sigma}_\phi^i)^{\frac{1}{2}}\boldsymbol{w}\|_2\right) - \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\xi}.\end{aligned}$$

Note that for any $\boldsymbol{x} \in \mathbb{R}^n$, we have $\|\boldsymbol{x}\|_2 = \max_{\|\boldsymbol{y}\|_2 \le 1} \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x}$. Using this fact to eliminate the term $\|(\boldsymbol{\Sigma}_\phi^i)^{\frac{1}{2}}\boldsymbol{w}\|_2$ in $\mathcal{L}$, we have a modified Lagrangian

$$\begin{aligned}& \mathcal{L}_1(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{u}) \\ =\ & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^N \xi_i \\ & - \sum_{i=1}^N \alpha_i\left(y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_\phi^i + b) - 1 + \xi_i + \tau(\epsilon)y_i(\boldsymbol{u}^i)^{\mathrm{T}}(\boldsymbol{\Sigma}_\phi^i)^{\frac{1}{2}}\boldsymbol{w}\right) - \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\xi},\end{aligned}$$

and the relation $\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \max_{\|\boldsymbol{u}^i\|_2 \le 1, i=1,\ldots,N} \mathcal{L}_1(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{u})$. Here we use $-y_i\boldsymbol{u}^i$ in $\mathcal{L}_1$ in consideration of the future computation and it will not affect the result since $\boldsymbol{u}^i$ is an arbitrary vector satisfying $\|\boldsymbol{u}^i\|_2 \le 1$. The modified Lagrangian leads to an easier construction of the dual problem using the fact of

$$\max_{\boldsymbol{\alpha}\ge 0, \boldsymbol{\beta}\ge 0} \min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \max_{\boldsymbol{\alpha}\ge 0, \boldsymbol{\beta}\ge 0, \|\boldsymbol{u}^i\|_2 \le 1, \forall i} \min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \mathcal{L}_1(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{u}).$$

Taking partial derivatives of $\mathcal{L}_1$ with respect to $\boldsymbol{w}$, b, and $\boldsymbol{\xi}$, respectively, yields

$$\begin{aligned}\partial_{\boldsymbol{w}}\mathcal{L}_1 &= \boldsymbol{w} - \sum_{i=1}^N y_i\alpha_i\left(\boldsymbol{\mu}_\phi^i + \tau(\epsilon)((\boldsymbol{\Sigma}_\phi^i)^{\frac{1}{2}})^{\mathrm{T}}\boldsymbol{u}^i\right), \\ \partial_b\mathcal{L}_1 &= -\sum_{i=1}^N y_i\alpha_i, \\ \partial_{\boldsymbol{\xi}}\mathcal{L}_1 &= C\boldsymbol{e} - \boldsymbol{\alpha} - \boldsymbol{\beta}.\end{aligned}$$

Setting them to zero, we have

$$\begin{aligned}\boldsymbol{w} &= \sum_{i=1}^N y_i\alpha_i\left(\boldsymbol{\mu}_\phi^i + \tau(\epsilon)((\boldsymbol{\Sigma}_\phi^i)^{\frac{1}{2}})^{\mathrm{T}}\boldsymbol{u}^i\right), \\ \sum_{i=1}^N y_i\alpha_i &= 0, \\ 0 \le \boldsymbol{\alpha} &\le C\boldsymbol{e}.\end{aligned}$$

Substituting the above into $\mathcal{L}_1$, then the Lagrange dual problem can be derived accordingly. □

An interesting fact is that compared to the dual of the classical kernel-based SVM model (Wang, 2005), the uncertain dual model (Dual-DRCKSVM$_{SOCP}$) uses $\boldsymbol{\mu}_\phi^i + \tau(\epsilon)((\boldsymbol{\Sigma}_\phi^i)^{\frac{1}{2}})^{\mathrm{T}}\boldsymbol{u}^i$ with $\|\boldsymbol{u}^i\|_2 \le 1$ as the separation objectives, which represents the points in $\mathcal{E}(\boldsymbol{\mu}_\phi^i, \boldsymbol{\Sigma}_\phi^i, \tau(\epsilon))$, to construct the kernel matrix. Let $(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*)$ and $(\boldsymbol{\alpha}^*, \boldsymbol{u}^*)$ be the primal and dual optimal solutions, respectively. Then we can explore more from the KKT conditions that can be derived from Lemma 1:

$$\begin{aligned}& y_i\left((\boldsymbol{w}^*)^{\mathrm{T}}\boldsymbol{\mu}_\phi^i + b^*\right) \ge 1 - \xi_i + \tau(\epsilon)\|(\boldsymbol{\Sigma}_\phi^i)^{\frac{1}{2}}\boldsymbol{w}^*\|_2, \ i = 1, \ldots, N, \\ & \boldsymbol{w}^* = \sum_{i=1}^N y_i\alpha_i^*\left(\boldsymbol{\mu}_\phi^i + \tau(\epsilon)((\boldsymbol{\Sigma}_\phi^i)^{\frac{1}{2}})^{\mathrm{T}}(\boldsymbol{u}^i)^*\right), \ i = 1, \ldots, N, \\ & \sum_{i=1}^N y_i\alpha_i^* = 0, \\ & \alpha_i^*\left(y_i\left((\boldsymbol{w}^*)^{\mathrm{T}}\boldsymbol{\mu}_\phi^i + b^*\right) - 1 + \xi_i - \tau(\epsilon)\|(\boldsymbol{\Sigma}_\phi^i)^{\frac{1}{2}}\boldsymbol{w}^*\|_2\right) = 0, \ i = 1, \ldots, N, \\ & (C - \alpha_i^*)\xi_i = 0, \ \xi_i \ge 0, \ 0 \le \alpha_i^* \le C, \ i = 1, \ldots, N.\end{aligned} \tag{5}$$

The KKT conditions (5) of the problem provide some interesting insights:

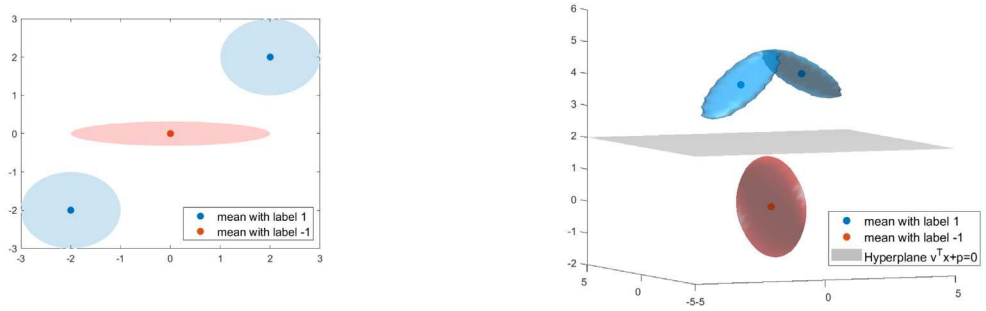
- The vector $\boldsymbol{w}^*$ is in the span of the points from the uncertainty ellipsoids $\mathcal{E}(\boldsymbol{\mu}_\phi^i, \boldsymbol{\Sigma}_\phi^i, \tau(\epsilon))$, $i = 1, \ldots, N$.
- The unit vector $\boldsymbol{u}^i$ that maximizes $(\boldsymbol{u}^i)^{\mathrm{T}}(\boldsymbol{\Sigma}_\phi^i)^{\frac{1}{2}}\boldsymbol{w}$ has the same directions as $(\boldsymbol{\Sigma}_\phi^i)^{\frac{1}{2}}\boldsymbol{w}$.
- Similarly as the support vector developed in the basic SVM models, we can define the support ellipsoid for (DRCKSVM$_{SOCP}$). In particular, the ellipsoid $\mathcal{E}(\boldsymbol{\mu}_\phi^i, \boldsymbol{\Sigma}_\phi^i, \tau(\epsilon))$ is a support ellipsoid when $\alpha_i^* \ne 0$.

Example 1. Consider a two-dimensional binary classification problem with three uncertain data points $\{\tilde{\boldsymbol{x}}^i \in \mathbb{R}^2, i = 1, 2, 3\}$ knowing the means $\boldsymbol{\mu}^1 = [2, 2]^{\mathrm{T}}$, $\boldsymbol{\mu}^2 = [-2, -2]^{\mathrm{T}}$, $\boldsymbol{\mu}^3 = [0, 0]^{\mathrm{T}}$, and covariance matrices $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \begin{bmatrix}0.5 & 0 \\ 0 & 0.5\end{bmatrix}$, $\boldsymbol{\Sigma}_3 = \begin{bmatrix}4 & 0 \\ 0 & 0.01\end{bmatrix}$, with labels $\{y_1 = y_2 = 1,\ y_3 = -1\}$. Note that for each i, $\tilde{x}_1^i$ and $\tilde{x}_2^i$ are independent. One can easily observe that it is not linearly separable from Fig. 1(a). We define a mapping $\phi(\tilde{x}_1, \tilde{x}_2) = [\tilde{x}_1, \tilde{x}_2, \tilde{x}_1\tilde{x}_2]^{\mathrm{T}} \in \mathbb{R}^3$. In this case, the mean and covariance matrix could be derived explicitly as $\mathbb{E}(\phi(\tilde{x}_1, \tilde{x}_2)) = [\mathbb{E}(\tilde{x}_1), \mathbb{E}(\tilde{x}_2), \mathbb{E}(\tilde{x}_1)\mathbb{E}(\tilde{x}_2)]^{\mathrm{T}}$, and

$$Cov(\phi(\tilde{x}_1, \tilde{x}_2)) = \begin{bmatrix} Var(\tilde{x}_1) & 0 & Var(\tilde{x}_1)\mathbb{E}(\tilde{x}_2) \\ 0 & Var(\tilde{x}_2) & Var(\tilde{x}_2)\mathbb{E}(\tilde{x}_1) \\ Var(\tilde{x}_1)\mathbb{E}(\tilde{x}_2) & Var(\tilde{x}_2)\mathbb{E}(\tilde{x}_1) & Var(\tilde{x}_1)\mathbb{E}(\tilde{x}_2^2) + Var(\tilde{x}_2)\mathbb{E}^2(\tilde{x}_1) \end{bmatrix}.$$

Then $\boldsymbol{\mu}_\phi^i = \mathbb{E}(\phi(\tilde{x}_1^i, \tilde{x}_2^i))$ and $\boldsymbol{\Sigma}_\phi^i = Cov(\phi(\tilde{x}_1^i, \tilde{x}_2^i))$ can be obtained accordingly, for $i = 1, 2, 3$. The (DRCKSVM) model can be applied to find a hyperplane separating ellipsoids in two classes by solving (DRCKSVM$_{SOCP}$), as shown in Fig. 1(b).

With strong assumptions on mean and covariance, we can explicitly show the classification result in the three-dimensional feature space for a simple two-dimensional example. However, even knowing ϕ, it is not easy to compute $\boldsymbol{\mu}_\phi^i$ and $\boldsymbol{\Sigma}_\phi^i$ of $\phi(\tilde{\boldsymbol{x}})$ directly based on the first two moments $\boldsymbol{\mu}^i$ and $\boldsymbol{\Sigma}^i$ for most cases, not to mention that the nonlinear mapping ϕ is usually defined implicitly and the mapped feature space could be in $\mathbb{R}^\infty$. The development of computationally tractable solution methods for solving (DRCKSVM) remains a crucial



(a) Data $\{(\boldsymbol{\mu}^i, \boldsymbol{\Sigma}^i)\}$ in $\mathbb{R}^2$. (b) Mapped data $\{(\boldsymbol{\mu}^i_\phi, \boldsymbol{\Sigma}^i_\phi)\}$ in $\mathbb{R}^3$.

Fig. 1. Geometric illustration of Example 1.

issue. In the following subsections, we propose a data-driven approach employing empirical moment estimation to address this issue.

2.2. Data-driven SOCP reformulation

Suppose that, for $i = 1, \ldots, N$, a batch of m_i independent extractions

$$S_i \triangleq \{\boldsymbol{x}^{i_j} \in \mathbb{R}^n, j = 1, \ldots, m_i\} \tag{6}$$

of the uncertain input $\tilde{\boldsymbol{x}}^i$ are available. Let $m = \sum_{i=1}^N m_i$ be the total number of samples such that $\{\boldsymbol{x}^s\}_{s=1}^m = \bigcup_{i=1}^N \{\boldsymbol{x}^{i_j}\}_{j=1}^{m_i}$. Based on the independence assumption among $\tilde{\boldsymbol{x}}^i$s, we will focus on $\tilde{\boldsymbol{x}}^i$ for each i in this section. The basic mechanism of the data-driven approach for solving (DRCKSVM$_{SOCP}$) is using $\phi(S_i) = \{\phi(\boldsymbol{x}^{i_j}) \in \mathbb{R}^d, j = 1, \ldots, m_i\}$ to estimate the moment information, $\boldsymbol{\mu}^i_\phi$ and $\boldsymbol{\Sigma}^i_\phi$. This raises a reliability concern about the empirical estimation. In other words, we need to know how close the sample mean based on $\phi(S_i)$ is to the true expectation $\boldsymbol{\mu}^i_\phi = \mathbb{E}[\phi(\tilde{\boldsymbol{x}}^i)] = \int_{\Xi_i} \phi(\tilde{\boldsymbol{x}}^i) dF_i$. We denote the sample mean by $\hat{\boldsymbol{\mu}}^{S_i}_\phi \triangleq \frac{1}{m_i}\sum_{j=1}^{m_i} \phi(\boldsymbol{x}^{i_j})$. There are two related results in the literature (Lemmas 2 and 3 below).

Lemma 2 (*Theorem 3 in Shawe and Taylor, 2003*). *For $i = 1, \ldots, N$, let $R_i = \sup_{\boldsymbol{x}^i \in \Xi_i} \|\phi(\boldsymbol{x}^i)\|_2$. Over the choice of S_i, we have*

$$\|\hat{\boldsymbol{\mu}}^{S_i}_\phi - \boldsymbol{\mu}^i_\phi\|_2 \le \frac{R_i}{\sqrt{m_i}}\left(2 + \sqrt{2\ln\frac{1}{\delta}}\right), \tag{7}$$

with a probability at least $1 - \delta$ $(0 < \delta < 1)$.

Next, consider the covariance matrix defined by $\boldsymbol{\Sigma}^i_\phi = \mathbb{E}[(\phi(\tilde{\boldsymbol{x}}^i) - \boldsymbol{\mu}^i_\phi)(\phi(\tilde{\boldsymbol{x}}^i) - \boldsymbol{\mu}^i_\phi)^{\mathrm{T}}]$. Let the empirical estimate of this quantity be

$$\hat{\boldsymbol{\Sigma}}^{S_i}_\phi \triangleq \frac{1}{m_i}\sum_{j=1}^{m_i}(\phi(\boldsymbol{x}^{i_j}) - \hat{\boldsymbol{\mu}}^{S_i}_\phi)(\phi(\boldsymbol{x}^{i_j}) - \hat{\boldsymbol{\mu}}^{S_i}_\phi)^{\mathrm{T}} = \frac{1}{m_i}\sum_{j=1}^{m_i}\phi(\boldsymbol{x}^{i_j})\phi(\boldsymbol{x}^{i_j})^{\mathrm{T}} - \hat{\boldsymbol{\mu}}^{S_i}_\phi(\hat{\boldsymbol{\mu}}^{S_i}_\phi)^{\mathrm{T}}.$$

Likewise, a comparable result can be extended to the covariance.

Lemma 3 (*Corollary 6 in Shawe and Taylor, 2003*). *For $i = 1, \ldots, N$, let $R_i = \sup_{\boldsymbol{x}^i \in \Xi_i} \|\phi(\boldsymbol{x}^i)\|_2$. Over the choice of S_i, we have*

$$\|\hat{\boldsymbol{\Sigma}}^{S_i}_\phi - \boldsymbol{\Sigma}^i_\phi\|_F \le \frac{2R_i^2}{\sqrt{m_i}}\left(2 + \sqrt{2\ln\frac{2}{\delta}}\right), \tag{8}$$

with a probability at least $1 - \delta$ $(0 < \delta < 1)$, provided that $m_i \ge \left(2 + \sqrt{2\ln(\frac{2}{\delta})}\right)^2$, where $\|\cdot\|_F$ is the Frobenius norm of matrices.

Lemmas 2 and 3 provide us with confidence regions for the sample mean and covariance containing the true mean and covariance matrix with a high probability.

Following this, we investigated the reliability of the chance constraints with the ambiguity set after incorporating these estimations. The obtained outcome will be used to create an ambiguity set that provides probabilistic assurances of the robustness of the data-driven solution with respect to the true distribution of the random vector.

Theorem 1. *For $i = 1, \ldots, N$, let $R_i = \sup_{\boldsymbol{x}^i \in \Xi_i} \|\phi(\boldsymbol{x}^i)\|_2$, $r_{1i} = \frac{R_i^2}{\sqrt{m_i}}(2 + \sqrt{2\ln\frac{2}{\delta}})$, and $r_{2i} = \frac{R_i^2}{\sqrt{m_i}}(2 + \sqrt{2\ln\frac{4}{\delta}})$. Over the choice of S_i, we have*

$$\sup_{F_i \in \mathcal{P}_i} \mathbb{P}_{F_i}\left\{y_i\left(\boldsymbol{w}^{\mathrm{T}}\phi(\tilde{\boldsymbol{x}}^i) + b\right) \le 1 - \xi_i\right\} \le \epsilon,$$

with a probability at least $1 - \delta$ $(0 < \delta < 1)$, provided that $m_i \ge (2 + \sqrt{2\ln\frac{4}{\delta}})^2$, and

$$y_i\left(\boldsymbol{w}^{\mathrm{T}}\hat{\boldsymbol{\mu}}^{S_i}_\phi + b\right) \ge 1 - \xi_i + \tau(\epsilon)\|(\hat{\boldsymbol{\Sigma}}^{S_i}_\phi + r_i\boldsymbol{I})^{\frac{1}{2}}\boldsymbol{w}\|_2,$$

where $r_i = \frac{r_{1i}}{\tau(\epsilon)} + r_{2i}$.

Proof. See Appendix A.2. □

With assured reliability of the empirical moment estimates, we then apply $\hat{\boldsymbol{\mu}}^{S_i}_\phi$ and $\hat{\boldsymbol{\Sigma}}^{S_i}_\phi$ to find an approximated and solvable model for (DRCKSVM$_{SOCP}$). A straightforward empirical approximated model employing the empirical estimations is

$$\begin{aligned} \min \quad & \tfrac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^N \xi_i \\ s.t. \quad & y_i\left(\boldsymbol{w}^{\mathrm{T}}\hat{\boldsymbol{\mu}}^{S_i}_\phi + b\right) \ge 1 - \xi_i + \tau(\epsilon)\|(\hat{\boldsymbol{\Sigma}}^{S_i}_\phi)^{\frac{1}{2}}\boldsymbol{w}\|_2, \ i = 1, \ldots, N, \\ & \boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}^N_+. \end{aligned} \tag{9}$$

Remark 2.1. Theorem 1 leads to a norm term $\|(\hat{\boldsymbol{\Sigma}}^{S_i}_\phi + r_i\boldsymbol{I})^{\frac{1}{2}}\boldsymbol{w}\|_2$, distinct from norm term $\|(\hat{\boldsymbol{\Sigma}}^{S_i}_\phi)^{\frac{1}{2}}\boldsymbol{w}\|_2$ in (9). Notice that r_i is determined by R_i, which is $\sup_{\boldsymbol{x}^i \in \Xi_i} \|\phi(\boldsymbol{x}^i)\|_2$. It is hard to calculate the value of R_i since for most kernels, the mapping ϕ might be implicit. We cannot solve the corresponding model directly. Moreover, the findings in Theorem 1 furnish a theoretical assurance that, under certain conditions, there exists a high probability of satisfying the original chance constraints when the data-driven empirical constraints are met. This insight remains valid without the diagonal matrix term $r_i\mathbf{I}$, providing us with valuable intuition.

Nonetheless, a persistent challenge arises due to the implicit nature of mapping ϕ for most kernel functions, making it impractical to directly utilize the empirical estimations-based model (9). The key to constructing a solvable data-driven model lies in obtaining parameters directly constructed from samples S_i defined by (6). Next, we leverage some theoretical results obtained in Section 2.1 and the inner product technique in kernel trick to address this issue.

According to the KKT conditions (5), the optimal $\boldsymbol{w}^*$ can be equivalently represented by a span of the points from the uncertainty

ellipsoids $\mathcal{E}(\boldsymbol{\mu}_\phi^i, \boldsymbol{\Sigma}_\phi^i, \tau(\epsilon))$, $i = 1, \ldots, N$, i.e., $\boldsymbol{w}^* = \sum_{i=1}^N y_i \alpha_i^* (\boldsymbol{\mu}_\phi^i + \tau(\epsilon)((\boldsymbol{\Sigma}_\phi^i)^{\frac{1}{2}})^{\mathrm{T}}(\boldsymbol{u}^i)^*)$. When the shape of the ellipsoid $\mathcal{E}(\boldsymbol{\mu}_\phi^i, \boldsymbol{\Sigma}_\phi^i, \tau(\epsilon))$ is determined by the covariance matrix $\boldsymbol{\Sigma}_\phi^i$, any point in this ellipsoid is in the span of the empirical points used in estimating the covariance matrix, since the eigenvectors of the covariance matrix span the entire ellipsoid. The eigenvectors of a covariance matrix are in the span of the empirical points from which the covariance matrix is estimated. Hence, we represent $\boldsymbol{\mu}_\phi^i + \tau(\epsilon)((\boldsymbol{\Sigma}_\phi^i)^{\frac{1}{2}})^{\mathrm{T}}(\boldsymbol{u}^i)^*$ as a linear combination of the empirical points S_i defined by (6), i.e., $\sum_{j=1}^{m_i} \alpha_{i_j}^0 \phi(\boldsymbol{x}^{i_j})$. Note that $\boldsymbol{w}$ is in the span of the training data points in the feature space, such that

$$\boldsymbol{w} = \sum_{i=1}^{N} y_i \alpha_i \sum_{j=1}^{m_i} \alpha_{i_j}^0 \phi(\boldsymbol{x}^{i_j}) = \boldsymbol{\phi}(\boldsymbol{X})\bar{\boldsymbol{Y}}\boldsymbol{v}, \tag{10}$$

where $\boldsymbol{v} = [v_1, \ldots, v_m]^{\mathrm{T}} \in \mathbb{R}^m$ is a rearranged dual variable of $\boldsymbol{\alpha}, \boldsymbol{\alpha}_{i_j}^0$ associated with all m reservations, $\boldsymbol{\phi}(\boldsymbol{X}) = [\phi(\boldsymbol{x}^{1_1}), \ldots, \phi(\boldsymbol{x}^{1_{m_1}}), \ldots, \phi(\boldsymbol{x}^{N_1}), \ldots, \phi(\boldsymbol{x}^{N_{m_N}})] \in \mathbb{R}^{d\times m}$, and $\bar{\boldsymbol{Y}} = diag(\sum_{i=1}^N y_i \boldsymbol{e}^i) \in \mathbb{R}^{m\times m}$ with $\boldsymbol{e}^i \in \mathbb{R}^m$ defined by

$$e_s^i = \begin{cases} 1, & \text{if } \boldsymbol{x}^s \text{ is a sample of } \tilde{\boldsymbol{x}}^i, \\ 0, & \text{otherwise,} \end{cases} \quad \text{for } s = 1, \ldots, m. \tag{11}$$

Define an m-dimensional kernel space and the kernel matrix $\bar{\boldsymbol{K}} \triangleq \boldsymbol{\phi}(\boldsymbol{X})^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{X}) \in \mathbb{R}^{m\times m}$ where $\bar{\boldsymbol{K}}_{sp} = \boldsymbol{\phi}(\boldsymbol{x}^s)^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}^p) = \kappa(\boldsymbol{x}^s, \boldsymbol{x}^p)$ for $s, p = 1, \ldots, m$, determined by a kernel function $\kappa : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$.

We then derive a solvable data-driven model for (9). We have $\|\boldsymbol{w}\|_2^2 = \boldsymbol{v}^{\mathrm{T}}\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}\bar{\boldsymbol{Y}}\boldsymbol{v}$. For $i = 1, \ldots, N$, we further have

$$\boldsymbol{w}^{\mathrm{T}}\hat{\boldsymbol{\mu}}_\phi^{S_i} = (\boldsymbol{\phi}(\boldsymbol{X})\bar{\boldsymbol{Y}}\boldsymbol{v})^{\mathrm{T}}\left(\frac{1}{m_i}\sum_{j=1}^{m_i}\phi(\boldsymbol{x}^{i_j})\right) = \boldsymbol{v}^{\mathrm{T}}\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}^i,$$

where $\bar{\boldsymbol{K}}^i = \frac{1}{m_i}\bar{\boldsymbol{K}}\boldsymbol{e}^i$. Similarly, we have

$$\begin{aligned} \boldsymbol{w}^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_\phi^{S_i}\boldsymbol{w} &= \boldsymbol{w}^{\mathrm{T}}\left(\frac{1}{m_i}\sum_{j=1}^{m_i}\left(\phi(\boldsymbol{x}^{i_j}) - \hat{\boldsymbol{\mu}}_\phi^{S_i}\right)\left(\phi(\boldsymbol{x}^{i_j}) - \hat{\boldsymbol{\mu}}_\phi^{S_i}\right)^{\mathrm{T}}\right)\boldsymbol{w} \\ &= \frac{1}{m_i}\sum_{j=1}^{m_i}\left(\boldsymbol{w}^{\mathrm{T}}(\phi(\boldsymbol{x}^{i_j}) - \hat{\boldsymbol{\mu}}_\phi^{S_i})\right)^2 \\ &= \frac{1}{m_i}\sum_{j=1}^{m_i}\left(\boldsymbol{v}^{\mathrm{T}}\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}^{i_j} - \boldsymbol{v}^{\mathrm{T}}\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}^i\right)^2 \\ &= \boldsymbol{v}^{\mathrm{T}}\left(\frac{1}{m_i}\sum_{j=1}^{m_i}(\bar{\boldsymbol{K}}^{i_j} - \bar{\boldsymbol{K}}^i)(\bar{\boldsymbol{K}}^{i_j} - \bar{\boldsymbol{K}}^i)^{\mathrm{T}}\right)\boldsymbol{v}, \end{aligned}$$

where for $i = 1, \ldots, N$, $j = 1, \ldots, m_i$, $\bar{\boldsymbol{K}}^{i_j} = \bar{\boldsymbol{K}}\boldsymbol{e}^{i_j}$, and $\boldsymbol{e}^{i_j} \in \mathbb{R}^m$ with

$$e_s^{i_j} = \begin{cases} 1, & \text{if } \boldsymbol{x}^s \text{ is the } j\text{th sample of } \tilde{\boldsymbol{x}}^i, \\ 0, & \text{otherwise,} \end{cases} \quad \text{for } s = 1, \ldots, m. \tag{12}$$

Denote $\boldsymbol{\Sigma}_K^i = \frac{1}{m_i}\sum_{j=1}^{m_i}(\bar{\boldsymbol{K}}^{i_j} - \bar{\boldsymbol{K}}^i)(\bar{\boldsymbol{K}}^{i_j} - \bar{\boldsymbol{K}}^i)^{\mathrm{T}} \in \mathbb{S}_+^m$. Then we have an approximated SOCP of the (DRCKSVM$_{SOCP}$) model as

$$\begin{aligned} \min \quad & \tfrac{1}{2}\boldsymbol{v}^{\mathrm{T}}\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}\bar{\boldsymbol{Y}}\boldsymbol{v} + C\sum_{i=1}^{N}\xi_i \\ s.t. \quad & y_i(\boldsymbol{v}^{\mathrm{T}}\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}^i + b) \geq 1 - \xi_i + \tau(\epsilon)\|(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}}\boldsymbol{v}\|_2, \ i = 1, \ldots, N, \\ & \boldsymbol{v} \in \mathbb{R}^m, b \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}_+^N. \end{aligned} \tag{DRCKSVM$_{aSOCP}$}$$

For an optimal solution $(\boldsymbol{v}^*, b^*, \boldsymbol{\xi}^*)$ of (DRCKSVM$_{aSOCP}$), the classification function is given by $f_\phi(\boldsymbol{x}; \boldsymbol{v}^*, b^*) = \mathrm{Sign}(\sum_{i=1}^N \frac{1}{m_i}\sum_{j=1}^{m_i} y_i v_{i_j}^* \kappa(\boldsymbol{x}^{i_j}, \boldsymbol{x}) + b^*)$.

(DRCKSVM$_{aSOCP}$) provides a tractable formulation for solving (DRCKSVM$_{SOCP}$), which can be solved by commercial conic optimization solvers. The interior-point method for solving SOCP (Lobo et al., 1998) yields a worst-case complexity of $O(m^2(N+1)^{\frac{3}{2}})$ for (DRCKSVM$_{aSOCP}$). Note that it is often the case that $m = \sum_{i=1}^N m_i \gg N > n$. However, a good approximation usually requires a large sample size m, which significantly increases the computational complexity. We notice that although the authors discussed the kernel-based SVMs with moment information in the context of data with missing values and semi-supervised learning (Shivaswamy et al., 2006; Huang et al., 2013), they utilized optimization solvers to find solutions and avoided discussing the inevitable computational burden for data with large samples. In the next section, considering computational efficiency, we propose an ADMM-based algorithm to solve (DRCKSVM$_{aSOCP}$).

3. ADMM-based algorithm

This section develops an ADMM algorithm to provide fast computations for solving (DRCKSVM$_{aSOCP}$). The ADMM algorithm partitions a large optimization problem into several smaller sub-problems that are easier to solve (Boyd et al., 2011). In this section, we successfully derived the explicit solution for each sub-problem, thereby significantly enhancing computational efficiency.

We notice that there are 2-norm terms, $\|(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}}\boldsymbol{v}\|_2, i = 1, \ldots, N$, in (DRCKSVM$_{aSOCP}$). Since $\nabla_{\boldsymbol{v}}\|(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}}\boldsymbol{v}\|_2 = \boldsymbol{\Sigma}_K^i\boldsymbol{v}/\|(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}}\boldsymbol{v}\|_2$ whose denominator is a function of $\boldsymbol{v}$, it is hard to find explicit solutions if such a term is involved. According to the fact that $\|(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}}\boldsymbol{v}\|_2 = \max_{\|\boldsymbol{z}_i\|_2\leq 1}\boldsymbol{z}_i^{\mathrm{T}}(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}}\boldsymbol{v}$ for $\boldsymbol{z}_i \in \mathbb{R}^m$, we first rewrite (DRCKSVM$_{aSOCP}$) as

$$\begin{aligned} \min_{\boldsymbol{v},b,\boldsymbol{a},\boldsymbol{z}_i} \quad & \tfrac{1}{2}\boldsymbol{v}^{\mathrm{T}}\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}\bar{\boldsymbol{Y}}\boldsymbol{v} + C\sum_{i=1}^{N}(a_i)^+ + \sum_{i=1}^{N}\mathbb{1}_{\mathcal{A}}(\boldsymbol{z}_i) \\ s.t. \quad & \boldsymbol{a} = \boldsymbol{e}_N - \begin{bmatrix} \boldsymbol{Y}\boldsymbol{M} - \tau(\epsilon)\begin{bmatrix} \boldsymbol{z}_1^{\mathrm{T}}(\boldsymbol{\Sigma}_K^1)^{\frac{1}{2}} \\ \vdots \\ \boldsymbol{z}_N^{\mathrm{T}}(\boldsymbol{\Sigma}_K^N)^{\frac{1}{2}} \end{bmatrix} & \boldsymbol{Y}\boldsymbol{e}_N \end{bmatrix}\begin{bmatrix} \boldsymbol{v} \\ b \end{bmatrix}, \end{aligned} \tag{13}$$

where $\boldsymbol{M} = (\bar{\boldsymbol{Y}}[\bar{\boldsymbol{K}}^1, \ldots, \bar{\boldsymbol{K}}^N])^{\mathrm{T}} \in \mathbb{R}^{N\times m}$, $\boldsymbol{e}_N = (1, \ldots, 1)^{\mathrm{T}} \in \mathbb{R}^N$, and $\boldsymbol{Y}$ is a diagonal matrix of labels, i.e., $\boldsymbol{Y} = diag(y_1, \ldots, y_N)$. For each i, let $a_i^+ \triangleq \max\{0, a_i\}$ and $\mathbb{1}_{\mathcal{A}}(\boldsymbol{z}_i)$ be the indicator function of the convex set $\mathcal{A} \triangleq \{\boldsymbol{z} \in \mathbb{R}^m | \|\boldsymbol{z}\|_2 \leq 1\}$ defined by

$$\mathbb{1}_{\mathcal{A}}(\boldsymbol{z}_i) = \begin{cases} 0, & \text{if } \boldsymbol{z}_i \in \mathcal{A}, \\ \infty, & \text{otherwise.} \end{cases}$$

Please refer to Appendix A.3 for the detailed proof.

Also, let $\boldsymbol{H}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N) \triangleq \begin{bmatrix} \boldsymbol{Y}\boldsymbol{M} - \tau(\epsilon)\begin{bmatrix} \boldsymbol{z}_1^{\mathrm{T}}(\boldsymbol{\Sigma}_K^1)^{\frac{1}{2}} \\ \vdots \\ \boldsymbol{z}_N^{\mathrm{T}}(\boldsymbol{\Sigma}_K^N)^{\frac{1}{2}} \end{bmatrix} & \boldsymbol{Y}\boldsymbol{e}_N \end{bmatrix} \in \mathbb{R}^{N\times(m+1)}$, abbreviated by $\boldsymbol{H}(\boldsymbol{Z})$ for $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N)$. The augmented Lagrangian of (13) becomes

$$\begin{aligned} L(\boldsymbol{v}, b, \boldsymbol{a}, \boldsymbol{Z}, \boldsymbol{\beta}) = & \frac{1}{2}\boldsymbol{v}^{\mathrm{T}}\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}\bar{\boldsymbol{Y}}\boldsymbol{v} + C\sum_{i=1}^{N}(a_i)^+ + \sum_{i=1}^{N}\mathbb{1}_{\mathcal{A}}(\boldsymbol{z}_i) \\ & + \boldsymbol{\beta}^{\mathrm{T}}\left(\boldsymbol{H}(\boldsymbol{Z})\begin{bmatrix} \boldsymbol{v} \\ b \end{bmatrix} + \boldsymbol{a} - \boldsymbol{e}_N\right) + \frac{\rho}{2}\left\|\boldsymbol{H}(\boldsymbol{Z})\begin{bmatrix} \boldsymbol{v} \\ b \end{bmatrix} + \boldsymbol{a} - \boldsymbol{e}_N\right\|_2^2, \end{aligned} \tag{14}$$

where $\boldsymbol{\beta} \in \mathbb{R}^N$ is a Lagrangian multiplier, and $\rho > 0$ is the penalty parameter. Our ADMM optimizer solves the convex problem (13) by splitting it into $N + 3$ sub-problems with respect to variables $(\boldsymbol{v}, b)$, $(\boldsymbol{z}_i)_{i=1,\ldots,N}$, $\boldsymbol{a}$, and $\boldsymbol{\beta}$:

$$\begin{cases} (\boldsymbol{v}^{(t+1)}, b^{(t+1)}) & = \arg\min\limits_{\boldsymbol{v},b} \ L(\boldsymbol{v}, b, \boldsymbol{z}_1^{(t)}, \ldots, \boldsymbol{z}_N^{(t)}, \boldsymbol{a}^{(t)}, \boldsymbol{\beta}^{(t)}) \\ \boldsymbol{z}_1^{(t+1)} & = \arg\min\limits_{\boldsymbol{z}_1} \ L(\boldsymbol{v}^{(t+1)}, b^{(t+1)}, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_N^{(t)}, \boldsymbol{a}^{(t)}, \boldsymbol{\beta}^{(t)}) \\ & \cdots \\ \boldsymbol{z}_N^{(t+1)} & = \arg\min\limits_{\boldsymbol{z}_N} \ L(\boldsymbol{v}^{(t+1)}, b^{(t+1)}, \ \boldsymbol{z}_1^{(t+1)}, \ldots, \boldsymbol{z}_N, \boldsymbol{a}^{(t)}, \ \boldsymbol{\beta}^{(t)}) \\ \boldsymbol{a}^{(t+1)} & = \arg\min\limits_{\boldsymbol{a}} \ L(\boldsymbol{v}^{(t+1)}, b^{(t+1)}, \boldsymbol{z}_1^{(t+1)}, \ldots, \boldsymbol{z}_N^{(t+1)}, \boldsymbol{a}, \boldsymbol{\beta}^{(t)}) \\ \boldsymbol{\beta}^{(t+1)} & = \arg\min\limits_{\boldsymbol{\beta}} \ L(\boldsymbol{v}^{(t+1)}, b^{(t+1)}, \boldsymbol{z}_1^{(t+1)}, \ldots, \boldsymbol{z}_N^{(t+1)}, \boldsymbol{a}^{(t+1)}, \boldsymbol{\beta}), \end{cases} \tag{15}$$

where t denotes the iteration numbers. To find explicit solutions of $(\boldsymbol{v}^{(t+1)}, b^{(t+1)})$ in (15), we can solve a linear system with given $(\boldsymbol{z}_1^{(t)}, \ldots, \boldsymbol{z}_N^{(t)}, \boldsymbol{a}^{(t)}, \boldsymbol{\beta}^{(t)})$:

$$\nabla_{(\boldsymbol{v},b)} L(\boldsymbol{v}, b, \boldsymbol{Z}, \boldsymbol{a}, \boldsymbol{\beta}) = \left(\begin{bmatrix} \bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}\bar{\boldsymbol{Y}} & \\ & 0 \end{bmatrix} + \boldsymbol{H}(\boldsymbol{Z})^{\mathrm{T}} \boldsymbol{H}(\boldsymbol{Z}) \right) \begin{bmatrix} \boldsymbol{v} \\ b \end{bmatrix} + \boldsymbol{H}(\boldsymbol{Z})^{\mathrm{T}} (\boldsymbol{\beta} + \rho(\boldsymbol{a} - \boldsymbol{e}_N)) = 0 \tag{16}$$

$$\Rightarrow \begin{bmatrix} \boldsymbol{v} \\ b \end{bmatrix} = -\left(\begin{bmatrix} \bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}\bar{\boldsymbol{Y}} & \\ & 0 \end{bmatrix} + \boldsymbol{H}(\boldsymbol{Z})^{\mathrm{T}} \boldsymbol{H}(\boldsymbol{Z}) \right)^{-1} \boldsymbol{H}(\boldsymbol{Z})^{\mathrm{T}} (\boldsymbol{\beta} + \rho(\boldsymbol{a} - \boldsymbol{e}_N)).$$

Notice that the kernel matrices in real applications may suffer ill-conditioned problems. In general, we have $\mathrm{Rank}(\boldsymbol{H}(\boldsymbol{Z})^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{Z})) \leq N < m+1$. Consequently, computing the inverse matrix in (16) could be a computational issue. Note that a summation of vector outer products can be obtained if we expand $\boldsymbol{H}(\boldsymbol{Z})^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{Z})$ as follows:

$$\boldsymbol{H}(\boldsymbol{Z})^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{Z}) = \rho \sum_{i=1}^{N} \begin{bmatrix} y_i \bar{\boldsymbol{K}}_i - \tau(\epsilon) \boldsymbol{\Sigma}_{K_i}^{\frac{1}{2}} \boldsymbol{z}_i \\ y_i \end{bmatrix} [(y_i \bar{\boldsymbol{K}}_i - \tau(\epsilon) \boldsymbol{\Sigma}_{K_i}^{\frac{1}{2}} \boldsymbol{z}_i)^{\mathrm{T}} \quad y_i].$$

This special structure makes a significant contribution to the development of Algorithm , and the details are shown in Appendix B.1.

Algorithm 1 Inverse matrix for an ill-conditioned matrix with special structure.

Input: An ill-conditioned kernel matrix $\bar{\boldsymbol{K}}, \bar{\boldsymbol{Y}}, \bar{\boldsymbol{K}}_i, \boldsymbol{\Sigma}_{K_i}, \boldsymbol{z}_i, y_i, \tau(\epsilon)$, $i = 1, \ldots, N$, $\rho, \theta > 0$.

Output: $\left(\begin{bmatrix} \bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}\bar{\boldsymbol{Y}} & 0 \\ 0 & 0 \end{bmatrix} + \rho \sum_{i=1}^{N} \begin{bmatrix} y_i \bar{\boldsymbol{K}}_i - \tau(\epsilon) \boldsymbol{\Sigma}_{K_i}^{\frac{1}{2}} \boldsymbol{z}_i \\ y_i \end{bmatrix} [(y_i \bar{\boldsymbol{K}}_i - \tau(\epsilon) \boldsymbol{\Sigma}_{K_i}^{\frac{1}{2}} \boldsymbol{z}_i)^{\mathrm{T}} \quad y_i] \right)^{-1}$.

1: Set $\hat{\boldsymbol{K}} = \bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}\bar{\boldsymbol{Y}} + \theta \boldsymbol{I}_n \succ 0$.

2: Compute $\boldsymbol{\sigma}_i = y_i \bar{\boldsymbol{K}}_i - \tau(\epsilon) \boldsymbol{\Sigma}_{K_i}^{\frac{1}{2}} \boldsymbol{z}_i$, $i = 1, \ldots, N$.

3: Compute $\boldsymbol{A}_1 = \begin{bmatrix} \hat{\boldsymbol{K}}^{-1} & 0 \\ 0 & \frac{1}{\theta} \end{bmatrix} - \frac{1}{\boldsymbol{\sigma}_1^{\mathrm{T}} \hat{\boldsymbol{K}}^{-1} \boldsymbol{\sigma}_1 + \frac{1}{\theta} + \frac{1}{\rho}} \begin{bmatrix} \hat{\boldsymbol{K}}^{-1} \boldsymbol{\sigma}_1 \\ \frac{1}{\theta} y_1 \end{bmatrix} [(\hat{\boldsymbol{K}}^{-1} \boldsymbol{\sigma}_1)^{\mathrm{T}} \quad \frac{1}{\theta} y_1]$.

4: **for** $i = 2, \ldots, N$ **do**

5: $\quad g_i = [\boldsymbol{\sigma}_i^{\mathrm{T}} \quad y_i] \boldsymbol{A}_{i-1} \begin{bmatrix} \boldsymbol{\sigma}_i \\ y_i \end{bmatrix} + \frac{1}{\rho}$.

6: $\quad \boldsymbol{A}_i = \boldsymbol{A}_{i-1} - \frac{1}{g_i} \boldsymbol{A}_{i-1} \begin{bmatrix} \boldsymbol{\sigma}_i \\ y_i \end{bmatrix} [\boldsymbol{\sigma}_i^{\mathrm{T}} \quad y_i] \boldsymbol{A}_{i-1}$.

7: Return $\left(\begin{bmatrix} \bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}\bar{\boldsymbol{Y}} & 0 \\ 0 & 0 \end{bmatrix} + \rho \sum_{i=1}^{N} \begin{bmatrix} y_i \bar{\boldsymbol{K}}_i - \tau(\epsilon) \boldsymbol{\Sigma}_{K_i}^{\frac{1}{2}} \boldsymbol{z}_i \\ y_i \end{bmatrix} [(y_i \bar{\boldsymbol{K}}_i - \tau(\epsilon) \boldsymbol{\Sigma}_{K_i}^{\frac{1}{2}} \boldsymbol{z}_i)^{\mathrm{T}} \quad y_i] \right)^{-1}$
$= \boldsymbol{A}_N - \boldsymbol{A}_N (\boldsymbol{A}_N - \frac{1}{\theta} \boldsymbol{I})^{-1} \boldsymbol{A}_N$.

It is not straightforward to solve for $\boldsymbol{z}_i^{(t)}, i = 1, \ldots, N$. However, we give some closed-form solutions in Lemma 4.

Lemma 4. *At each iteration t, the optimal solution of*

$$\boldsymbol{z}_i^{(t+1)} = \arg\min_{\boldsymbol{z}_1} \quad L(\boldsymbol{v}^{(t+1)}, b^{(t+1)}, \boldsymbol{z}_1^{(t+1)}, \ldots, \boldsymbol{z}_i, \ldots, \boldsymbol{z}_N^{(t)}, \boldsymbol{a}^{(t)}, \boldsymbol{\beta}^{(t)}),$$

$i = 1, \ldots, N$, has a closed form of

$$\boldsymbol{z}_i^{(t+1)} = h(\beta_i^{(t)}, \boldsymbol{v}^{(t+1)}, b^{(t+1)}, a_i^{(t)}) \min \left\{ \frac{\|(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}} \boldsymbol{v}^{(t+1)}\|_2}{|h(\beta_i^{(t)}, \boldsymbol{v}^{(t+1)}, b^{(t+1)}, a_i^{(t)})|}, 1 \right\} \times \frac{(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}} \boldsymbol{v}^{(t+1)}}{\|(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}} \boldsymbol{v}^{(t+1)}\|_2^2}, \tag{17}$$

where $h(\beta_i^{(t)}, \boldsymbol{v}^{(t+1)}, b^{(t+1)}, a_i^{(t)}) = \frac{1}{\tau(\epsilon)} (\frac{\beta_i^{(t)}}{\rho} - 1 + y_i((\bar{\boldsymbol{K}}^i)^{\mathrm{T}} \boldsymbol{v}^{(t+1)} + b^{(t+1)}) + a_i^{(t)})$ for each i.

Proof. See Appendix A.1. □

Adopting the stopping criterion evaluated by the primal residual and solution errors (Boyd et al., 2011), the ADMM-based algorithm for solving (DRCKSVM$_{aSOCP}$) can be finalized as blow.

Algorithm 2 ADMM-based algorithm for (DRCKSVM$_{aSOCP}$)

Input: Data matrix $\boldsymbol{Y}$, $\bar{\boldsymbol{Y}}$, $\boldsymbol{M}$, $\boldsymbol{K}$, $\bar{\boldsymbol{K}}_i$, $\boldsymbol{\Sigma}_{K_i}, i = 1, \ldots, N$.
Preset parameters ρ, ϵ, and error thresholds ϵ_{res} and ϵ_{sol}.

Output: $\boldsymbol{v}^*, b^*$.

1: Initialize $t = 0$, $(\boldsymbol{v}^{(0)}, b^{(0)})$, $\boldsymbol{\beta}^{(0)}$, $\boldsymbol{z}_1^{(0)}, \ldots, \boldsymbol{z}_N^{(0)}$, and $a_1^{(0)}, \ldots, a_N^{(0)}$. Set the primal residual $\boldsymbol{r}^{(0)} = \boldsymbol{e}$, and $\epsilon(\boldsymbol{v}^{(0)}, b^{(0)}) = 1$.

2: **while** $\|\boldsymbol{r}^{(t)}\|_2 > \epsilon_{res}$ or $\epsilon(\boldsymbol{v}^{(t}, b^{(t)}) > \epsilon_{sol}$ **do**

3: $\quad$ **if** $\begin{bmatrix} \bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}\bar{\boldsymbol{Y}} & \\ & 0 \end{bmatrix} + \boldsymbol{H}(\boldsymbol{Z}^{(t)})^{\mathrm{T}} \boldsymbol{H}(\boldsymbol{Z}^{(t)})$ is ill-conditioned **then**

4: $\quad\quad$ Update $[\boldsymbol{v}^{(t+1)}; b^{(t+1)}]$ by Algorithm .

5: $\quad$ **else**

6: $\quad\quad$ Solve the following linear equation system for an update $[\boldsymbol{v}^{(t+1)}; b^{(t+1)}]$:

$$\begin{bmatrix} \boldsymbol{v}^{(t+1)} \\ b^{(t+1)} \end{bmatrix} = -\left(\begin{bmatrix} \bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}\bar{\boldsymbol{Y}} & \\ & 0 \end{bmatrix} + \boldsymbol{H}(\boldsymbol{Z}^{(t)})^{\mathrm{T}} \boldsymbol{H}(\boldsymbol{Z}^{(t)}) \right)^{-1} \boldsymbol{H}(\boldsymbol{Z}^{(t)})^{\mathrm{T}} (\boldsymbol{\beta}^{(t)} + \rho(\boldsymbol{a}^{(t)} - \boldsymbol{e}_N)).$$

7: $\quad$ Update $\boldsymbol{z}_i^{(t+1)}$ by (17) in Lemma 4, for $i = 1, \ldots, N$.

8: $\quad \boldsymbol{a}^{(t+1)} = S_{\frac{C}{\rho}} \left(\boldsymbol{e}_N - \boldsymbol{H}(\boldsymbol{Z}^{(t+1)}) \begin{bmatrix} \boldsymbol{v}^{(t+1)} \\ b^{(t+1)} \end{bmatrix} - \frac{1}{\rho} \boldsymbol{\beta}^{(t)} \right)$.

9: $\quad$ Update the primal residual

$$\boldsymbol{r}^{(t+1)} = \boldsymbol{e}_N - \boldsymbol{H}(\boldsymbol{Z}^{(t+1)}) \begin{bmatrix} \boldsymbol{v}^{(t+1)} \\ b^{(t+1)} \end{bmatrix} - \boldsymbol{a}^{(t+1)}.$$

10: $\quad \boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \rho \boldsymbol{r}^{(t+1)}$.

11: $\quad$ Update the solution error $\epsilon(\boldsymbol{v}^{(t+1)}, b^{(t+1)}) = \left\| \begin{bmatrix} \boldsymbol{v}^{(t+1)} \\ b^{(t+1)} \end{bmatrix} - \begin{bmatrix} \boldsymbol{v}^{(t)} \\ b^{(t)} \end{bmatrix} \right\|_2^2$.

12: $\quad t = t + 1$.

13: **return** $(\boldsymbol{v}^*, b^*)$.

The soft thresholding operator S in Step 4 is given by

$$S_\theta(\boldsymbol{\eta}) = \begin{bmatrix} S_\theta(\eta_1) \\ \vdots \\ S_\theta(\eta_N) \end{bmatrix}, \text{ where } S_\theta(\eta_i) = \begin{cases} \eta_i - \theta, & \eta_i > \theta, \\ 0, & 0 \leqslant \eta_i \leqslant \theta, \ \forall\, i, \\ \eta_i, & \eta_i < 0, \end{cases} \tag{18}$$

via solving $S_\theta(\eta_i) = \arg\min_{\eta_i} \{\theta \eta_i^+ + \frac{1}{2}\|\eta_i - \theta\|_2^2\}$. The proposed ADMM-based algorithm can also address the commonly seen linear cases studied in Shivaswamy et al. (2006), Ben-Tal et al. (2011), Huang et al. (2012), Wang et al. (2018).

4. Computational experiments

This section conducts computational experiments on the proposed (DRCKSVM) model and solution method. Section 4.1 validates the effectiveness of the (DRCKSVM) model via solving the proposed data-driven (DRCKSVM$_{aSOCP}$) reformulation, assuming that the first- and second-order moments information are given. Section 4.2 evaluates the effectiveness and efficiency of the proposed ADMM-based Algorithm Lemma 4 for solving (DRCKSVM$_{aSOCP}$). Section 4.3 compares the (DRCKSVM) model with state-of-the-art kernel-based SVM models for classifying public benchmark data sets without any moment information. In this section, all computational experiments were conducted using MATLAB (R2021a) software on a desktop equipped with Intel(R) Core(TM) i3-9100 CPU @ 3.60 GHz CPUs and 32 GB RAM.

4.1. Effectiveness of the proposed (DRCKSVM) model

This section aims to verify the effectiveness of the (DRCKSVM) model for robust classification. The proposed data-driven approach in Section 2.2 provides a (DRCKSVM$_{aSOCP}$) reformulation, which will be

Table 1
Synthetic data sets for effectiveness validation of (DRCKSVM).

Dataset		Syn_2d_20b	Syn_8d_40b	Syn_16d_100b
Training data	# feature (n)	2	8	16
	# input (batch) (N)	20	40	100
	probability distribution	Normal, Uniform, T-distribution		
	data-driven sample size/batch	5, 10, 50		
Testing data	probability distribution	Normal, Uniform, T-distribution		
	# repetitions	100		

solved by using the commercial solver Mosek[1] in this section. We will train (DRCKSVM$_{aSOCP}$) on synthetic data sets (see Table 1) to validate the effectiveness of robust classification for data with uncertainty specified by moment information. To validate that our distributionally robust models can hedge against distribution uncertainty, we did three groups of experiments using three different distribution functions to generate the training and testing data, including the Normal, Uniform, and T-distributions. For example, for the synthetic dataset Syn_2d_20b, we first generate 20 points (10 points labeled '1' and 10 points labeled '−1') as the mean vectors $\boldsymbol{\mu}^i \in \mathbb{R}^2$, $i = 1, \ldots, 20$, for each group. Then, we set the covariance matrix for each group. For the Normal-distributed group, we set $\boldsymbol{\Sigma}^i = [0.05\ 0; 0\ 0.05] \in \mathbb{R}^{2\times 2}$, $i = 1, \ldots, 20$; For the Uniform-distributed group, we set the interval to be $[\mu_j^i - 0.75, \mu_j^i + 0.75]$, for $j = 1, 2$, and $i = 1, \ldots, 20$; For the T-distributed group, we set $\boldsymbol{\Sigma}^i = [0.6\ 0; 0\ 0.6] \in \mathbb{R}^{2\times 2}$ with the degree of freedom of value 3, $i = 1, \ldots, 20$. Based on the provided information regarding the mean and covariance, we initially generate 5, 10, or 50 points for each batch i (indicated as 'data-driven sample size/batch' in Table 1) for $i = 1, \ldots, 20$. These points constitute our training samples, utilized to estimate the empirical mean and covariance in the kernel space, and are employed in our proposed data-driven (DRCKSVM$_{aSOCP}$) model. Subsequently, in each group, utilizing the same distributions, we generate 100 points for each batch i, for $i = 1, \ldots, 20$, to construct our testing sets.

Two commonly used nonlinear kernels are tested including the quadratic polynomial kernel, $\kappa_{quad}(\boldsymbol{x}^i, \boldsymbol{x}^j) = (\gamma_q + (\boldsymbol{x}^i)^{\mathrm{T}}\boldsymbol{x}^j)^2$, and the radial basis function (rbf) kernel, $\kappa_{rbf}(\boldsymbol{x}^i, \boldsymbol{x}^j) = \exp(-\|\boldsymbol{x}^i - \boldsymbol{x}^j\|_2^2/(2\gamma_r^2))$. The corresponding models are denoted as DRCKSVM$_{aSOCP}$-quad and DRCKSVM$_{aSOCP}$-rbf, respectively. The parameter C controls the trade-off between maximizing the margin and minimizing the misclassification loss and the parameters γ_q and γ_r define the kernel functions, as commonly adopted in most kernel-based SVM models. To show the influence of parameter ϵ on the proposed model, we plot the results of both DRCKSVM$_{aSOCP}$-quad and DRCKSVM$_{aSOCP}$-rbf models on a two-dimensional artificial data set Syn-2d-20b for different values of ϵ by fixing other parameters $C = 2^{10}$, $\gamma_q = 2^2$, and $\gamma_r = 2^{-2}$ (A set of parameters that performed well for most models after grid search). Fig. 2 shows the nonlinear classifiers, the training and testing data points, and Table 2 records the accuracy scores (Acc(%)) with mean and standard deviation (std), with respect to different values of ϵ. From Fig. 2 and Table 2, we have the following observations:

- The value of ϵ affects the classifiers learned based on the first- and second-order moments. When $\epsilon = 1.00$, (DRCKSVM) reduces to a deterministic model without using moment information. We can observe that utilizing moment information with $\epsilon = 0.10, 0.20$ will provide better classification accuracy.
- For different kernel functions, the best value of ϵ may vary. The performance of DRCKSVM$_{aSOCP}$-quad improves as ϵ decreases, whereas this behavior is not entirely observed for DRCKSVM$_{aSOCP}$-rbf.
- For simulated points coming from three different distributions, the classification results of the proposed model perform quite robustly. This validates that the (DRCKSVM) model can hedge against the distribution uncertainty.

Synthetic data sets with larger feature dimensions have been tested to investigate the proposed model further. And we also tested the effect of the sample size for training the data-driven-based (DRCKSVM$_{aSOCP}$) reformulation. Table 3 shows the results when we set $\epsilon = 0.20$ and test on normally distributed testing data points. Table 3 records the accuracy scores (Acc(%)) with mean and standard deviation (std), and average training CPU time (s).

Table 3 shows that both DRCKSVM$_{aSOCP}$-quad and DRCKSVM$_{aSOCP}$-rbf can provide high classification accuracy. This validates the effectiveness of the proposed model for robust classification. In addition, when the sample size increases, the mean of the classification accuracy increases and the standard deviation decreases, which indicates that increasing sample size indeed improves the reliability of the (DRCKSVM$_{aSOCP}$) reformulation. However, we notice that the corresponding training time increases a lot when the sample size increases. Using the commercial solver Mosek solving (DRCKSVM$_{aSOCP}$) costs drastic computational effort for large-scale data. In the next section, we shall validate the effectiveness and especially the efficiency of the proposed ADMM-based algorithm for solving (DRCKSVM$_{aSOCP}$).

4.2. The ADMM-based algorithm for fast computations

We have verified that the proposed (DRCKSVM) model effectively provides a robust classifier of high quality for input data points with uncertainty specified by moment information. However, solving its (DRCKSVM$_{aSOCP}$) reformulation using Mosek may invoke a huge computational burden when handling large-scale data sets. This section aims to validate the effectiveness and efficiency of the proposed ADMM-based Algorithm Lemma 4 for solving (DRCKSVM$_{aSOCP}$), focusing on classification accuracy and training time.

We first test the synthetic datasets used in Section 4.1, as shown in Table 3, i.e., Syn_2d_20b, Syn_8d_40b, and Syn_16d_100b, each with 50 samples per batch. The same hyperparameters are used for each dataset as in Section 4.1. Table 4 records the accuracy scores (Acc(%)) with mean and standard deviation (std), and the average training CPU time (s) obtained by Mosek and ADMM-based Algorithm Lemma 4, respectively.

From Table 4, we can observe that the mean accuracy scores obtained by Mosek are slightly better than those achieved by the proposed Algorithm Lemma 4. These results align with the assertion in Hong et al. (2015) that the ADMM algorithm, when solving optimization problems with nonconvex bilinear constraints, may converge to a locally optimal saddle point close to the global optimal solution. Overall, the proposed ADMM-based algorithm for solving (DRCKSVM$_{aSOCP}$) provides a classifier of comparable quality to that of Mosek. The important observation is that the average training CPU time required by the ADMM-based algorithm reduces in significant orders compared to the time used by Mosek. The proposed ADMM-based algorithm indeed provides a fast algorithm for solving (DRCKSVM$_{aSOCP}$).

[1] Mosek is a state-of-the-art interior-point optimizer for conic problems. https://www.mosek.com/.

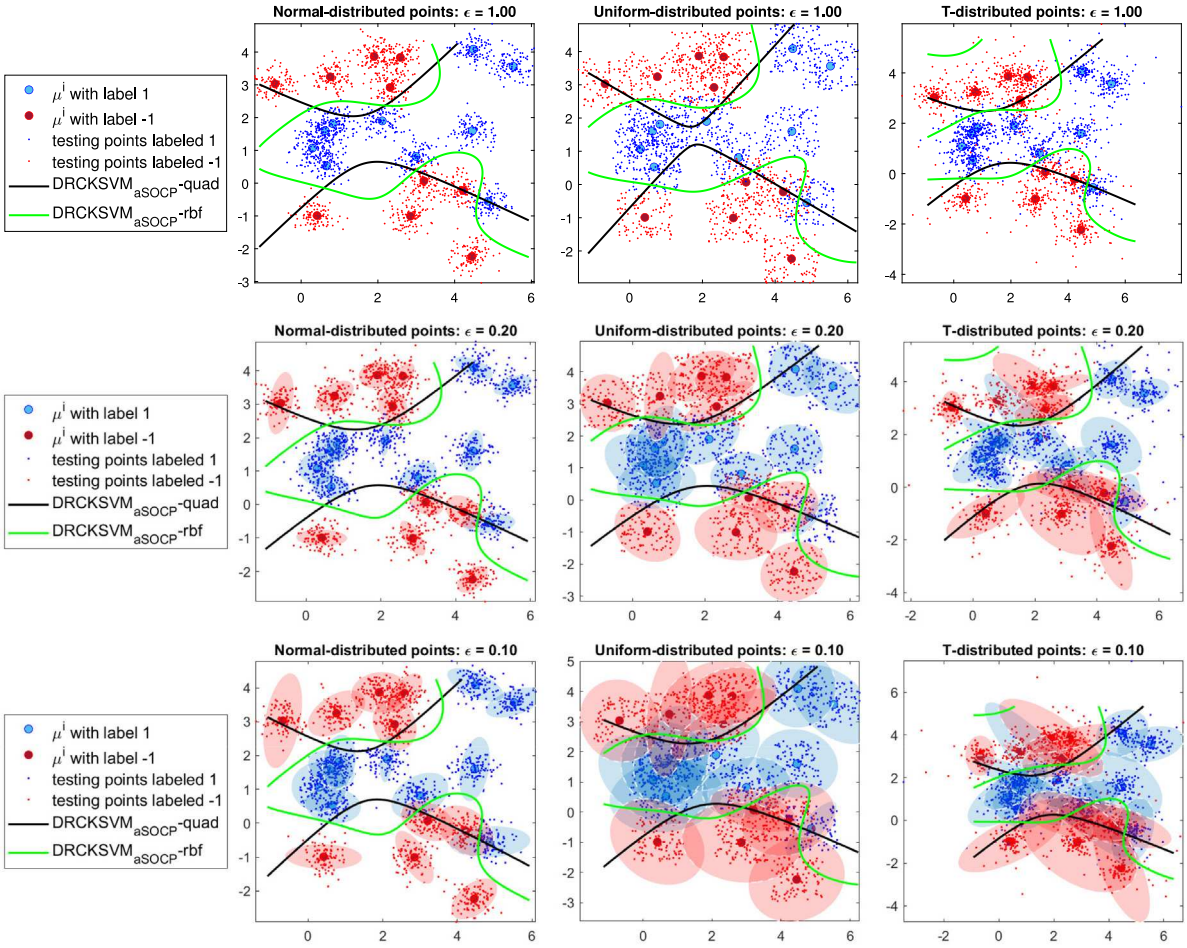


Fig. 2. Results on the data set Syn-2d-20b with different ϵ using 10 samples per batch for training.

Table 2
Results on the data set Syn-2d-20b with different ϵ using 10 samples per batch for training.

Testing sample		Normal		Uniform		T-distribution	
		Acc(%)		Acc(%)		Acc(%)	
	ϵ	mean	std	mean	std	mean	std
DRCKSVM$_{aSOCP}$-quad	1.00	87.70	5.15	81.95	8.20	85.70	6.18
	0.20	88.45	4.02	83.55	7.75	84.00	5.57
	0.10	**89.85**	5.67	**84.20**	6.31	**86.30**	6.12
DRCKSVM$_{aSOCP}$-rbf	1.00	96.70	5.73	90.55	6.21	92.55	6.45
	0.20	97.40	4.38	91.50	6.30	**93.65**	6.03
	0.10	**98.00**	4.94	**91.85**	5.73	91.70	5.01

Table 3
Results on the synthetic data with different training sample sizes per batch with $\epsilon = 0.20$.

	# Samples /batch	Syn_2d_20b			Syn_8d_40b			Syn_16d_100b		
		Acc(%)		CPU(s)	Acc(%)		CPU(s)	Acc(%)		CPU(s)
		mean	std		mean	std		mean	std	
DRCKSVM$_{aSOCP}$-quad	5	84.20	7.16	0.99	94.79	3.47	3.38	92.43	2.44	77.90
	10	85.90	6.52	5.25	96.15	2.96	12.14	96.81	1.66	338.12
	50	**88.80**	**5.40**	35.97	**98.37**	**2.03**	399.38	**98.01**	**1.32**	10 215.04
DRCKSVM$_{aSOCP}$-rbf	5	90.40	6.46	0.82	97.12	2.63	3.51	96.77	1.67	80.21
	10	95.20	4.28	4.13	97.53	2.43	12.89	97.20	1.55	342.25
	50	**96.32**	**4.12**	34.99	**98.36**	**2.05**	432.92	**98.86**	**1.05**	10 142.65

Table 4
Mosek vs. ADMM-based Algorithm Lemma 4 for solving (DRCKSVM$_{aSOCP}$) on synthetic data.

		Syn_2d_20b			Syn_8d_40b			Syn_16d_100b		
		Acc(%)		CPU(s)	Acc(%)		CPU(s)	Acc(%)		CPU(s)
		mean	std		mean	std		mean	std	
DRCKSVM$_{aSOCP}$-quad	Mosek	88.80	5.40	35.97	98.37	2.03	399.38	98.01	1.32	10215.04
	ADMM	86.00	2.02	5.12	97.59	7.71	8.75	97.12	7.02	94.01
DRCKSVM$_{aSOCP}$-rbf	Mosek	96.32	4.12	34.99	98.36	2.05	432.92	98.86	1.05	10142.65
	ADMM	95.70	5.31	2.11	97.43	4.45	11.32	97.06	3.75	92.15

Table 5
Mosek vs. ADMM-based Algorithm Lemma 4 for solving (DRCKSVM$_{aSOCP}$) on benchmark data.

		Sonar			Ionosphere			WIBC		
		Acc(%)		CPU(s)	Acc(%)		CPU(s)	Acc(%)		CPU(s)
		mean	std		mean	std		mean	std	
DRCKSVM$_{aSOCP}$-quad	Mosek	82.14	7.27	6.76	91.46	3.17	50.57	98.16	3.68	969.36
	ADMM	81.36	3.65	0.11	90.04	2.33	0.34	97.57	2.12	12.82
DRCKSVM$_{aSOCP}$-rbf	Mosek	83.73*	7.30	5.16	95.37	4.71	30.76	97.44	1.99	252.42
	ADMM	86.75	3.98	0.41	93.24	2.58	1.67	96.70	1.30	13.97

* Lower accuracy by Mosek due to the ill-conditioned matrix involved.

Table 6
Public benchmark data sets.

Data set	Sonar	Liver	Inosphere	WIBC	German	Car_evaluate	Heart	Cod_RNA
# Features	60	6	34	9	20	6	15	8
# Samples	208	319	351	666	1000	1594	3658	59 535

We also tested several benchmark data sets drawn from the UCI databases.[2] For each data set, we have utilized all points as training samples to calculate the estimated mean and covariance required by the proposed model, and then used all the original data points as testing samples. For example, there are 208 samples in the Sonar data set which is denoted as $\{\boldsymbol{x}^s, s = 1, \ldots, 208\}$. We assume that these 208 points are instances of 8 uncertain distributions, $\{\boldsymbol{x}^s, s = 1, \ldots, 208\} = \cup_{i=1}^{8}\{\boldsymbol{x}^{ij}, j = 1, \ldots, m_i\}$. The values of clusters are determined by the K-means clustering method. Then, the empirical mean and covariance calculated from these 208 training points, denoted as $\{\bar{\boldsymbol{K}}^i \in \mathbb{R}^{208}, \boldsymbol{\Sigma}_K^i \in \mathbb{S}_+^{208},\ i = 1, 2, \ldots, 8\}$ will be utilized to train the DRCKSVM$_{aSOCP}$ model. The dataset Ionosphere has 351 points with 34 features, and we treat them as 10 uncertain inputs for calculating the first- and second-order moments. The dataset WIBC has 208 points with 60 features, and we treat them as 8 uncertain inputs for calculating the first- and second-order moments. For benchmark datasets, the cross-validation and grid methods are adopted to select the best parameters of C, ϵ, and $\gamma_{q,r}$ from the ranges of $C \in \{2^{-1}, 2^1, \ldots, 2^{14}\}$, $\epsilon \in \{0.1, 0.2, \ldots, 1\}$, and $\gamma_q, \gamma_r \in \{2^{-5}, 2^{-4}, \ldots, 2^5\}$, respectively.

Table 5 records the same measures used in Table 4 and shows similar results that support the effectiveness and efficiency of the proposed ADMM-based algorithm for solving (DRCKSVM$_{aSOCP}$). This advantage of computational efficiency becomes more significant when the sample size increases. Moreover, the results of the Sonar dataset show that the ADMM-based algorithm may outperform Mosek when involving ill-conditioned matrices. Unlike the synthetic datasets we have tested, the benchmark datasets do not assume that all points come from an underlying distribution. The satisfying performance on the benchmark datasets using (DRCKSVM$_{aSOCP}$) shows the potential of our (DRCKSVM) model for classifying general data without moment information, and we shall conduct additional computational experiments in the next section.

4.3. Performance on public data sets without moment information

The results in the previous section indicate the effectiveness and promising efficiency of the proposed ADMM-based algorithm for solving (DRCKSVM$_{aSOCP}$). In this section, we conduct computational experiments to compare it with several state-of-the-art kernel-based SVMs using some commonly seen public benchmark data sets listed in Table 6 without assuming any stochastic uncertainty and moment information.

For all tests, the cross-validation and grid methods are adopted to select the best parameters of C, ϵ, γ_q, and γ_r from the ranges of $C \in \{2^{-1}, 2^1, \ldots, 2^{14}\}$, $\epsilon \in \{0.1, 0.2, \ldots, 1\}$, and $\gamma_q, \gamma_r \in \{2^{-5}, 2^{-4}, \ldots, 2^5\}$, respectively. For the data sets with a sample size larger than 600 (the last 5 data sets in Table 6), we select a small ratio of the data set as the training data set. All test results are based on the best-selected parameters. The standard kernel-based SVM models, including SVM-quad and SVM-rbf, are implemented by LIBSVM (Chang and Lin, 2011). To apply (DRCKSVM), we first use the K-means clustering algorithm (Vassilvitskii and Arthur, 2006) to partition each dataset into K clusters, where each cluster can be treated as a random sample set of an uncertain input and used to calculate the desired mean and covariance. Note that K cannot be too small or too large; large K values lead to suboptimal classification performance and excessively long computation times. Let $K = \min\{8, 10\% \times N_{train}\}$, where N_{train} is the sample size of the training points, and the values 8 and 10% are user-defined based on the size of the data set. Then the corresponding DRCKSVM$_{aSOCP}$-quad and DRCKSVM$_{aSOCP}$-rbf are solved using the ADMM-based Algorithm Lemma 4 for robust classifications. Table 7 reports the average accuracy scores (%) and Table 8 reports the average training CPU time (s).

[2] The UCI Machine Learning Repository is a collection of databases that are used by the machine learning community. https://archive.ics.uci.edu/ml/datasets.php.

Table 7
Average accuracy scores (%) tested on public benchmark data sets.

Data set	Sonar	Liver	Ionosphere	WIBC	German	Car_evaluate	Heart	Cod_RNA
SVM-quad	81.95	73.98	91.10	94.88	70.62	94.90	84.76	92.98
SVM-rbf	83.14	**76.49**	**95.01**	95.97	71.00	**95.98**	84.76	81.46
DRCKSVM$_{aSOCP}$-quad	81.36	73.05	90.04	**97.57**	70.50	85.81	84.74	**94.59**
DRCKSVM$_{aSOCP}$-rbf	**86.75**	75.99	93.24	96.70	**72.20**	93.39	**84.83**	86.07

Table 8
Average CPU time (s) tested on public benchmark data sets.

Data set	Sonar	Liver	Ionosphere	WIBC	German	Car_evaluate	Heart	Cod_RNA
SVM-quad	0.01	1.08	0.01	0.26	0.03	0.08	0.03	7.80
SVM-rbf	0.01	0.39	0.01	0.01	0.03	0.06	0.01	3.04
DRCKSVM$_{aSOCP}$-quad	0.11	0.91	0.34	0.01	0.03	0.05	0.02	7.07
DRCKSVM$_{aSOCP}$-rbf	0.41	1.72	1.67	0.02	0.09	0.10	0.04	13.64

Based on the accuracy score in Table 7 and CPU time in Table 8, we can observe that the proposed (DRCKSVM) models with quadratic polynomial kernel and rbf kernel produce more accurate classifications than the commonly used kernel-based SVM models without considering uncertainty for most datasets. It indicates that utilizing moment information hidden in the data can help improve the generalization of the classifier for a better prediction.

5. Conclusions

In this paper, we have studied a distributionally robust chance-constrained kernel-based SVM model for addressing binary classification tasks involving uncertain input data characterized by first- and second-order moments. Notice that the robust chance-constrained kernel-based SVM model is generally intractable even with true moments of the mapped data in the feature space, not to mention that the true moments are hard to obtain. Drawing on theoretical insights from our proposed (DRCKSVM) model, we have introduced a data-driven approach utilizing the empirical moments and kernel tricks to formulate a computationally efficient (DRCKSVM$_{aSOCP}$) reformulation. Real-world applications often necessitate solving large-scale SOCP reformulations. To address this, we have analyzed the problem's structural characteristics and developed an efficient ADMM algorithm tailored for solving (DRCKSVM$_{aSOCP}$).

Our computational experiments have validated that the proposed model can effectively find a robust classifier for data with uncertain inputs specified by the first- and second-order moments information. Results on both synthetic and benchmark datasets clearly support the effectiveness and efficiency of the ADMM-based algorithm. For the commonly seen public data sets without moment information, the proposed method utilizing the hidden moment information has shown better performance over other state-of-the-art kernel-based SVM models.

This study reveals an interesting fact that incorporating a reasonable amount of uncertainty during the training phase may significantly improve the predictive power of a resulting classifier. Future research can explore several interesting directions. Real-world machine learning problems frequently face data containing missing values in the inputs (Pelckmans et al., 2005; Shivaswamy et al., 2006). The proposed (DRCKSVM) model in this paper has utilized the moment information of inputs to classify possible realizations of inputs which may include the ones with missing variables. Extending the work in this paper may provide a method for resolving problems facing missing values. In addition, this paper has focused on data with input uncertainty assuming the output labels are known while, in practice, labels may be not accurately or not completely recorded. Robust optimization has been applied to this problem with promising performance (Bertsimas et al., 2019). Employing a distributionally robust optimization method for data involving uncertain labels can be an interesting direction for future research.

CRediT authorship contribution statement

Fengming Lin: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Shu-Cherng Fang:** Writing – review & editing, Validation, Supervision, Methodology, Formal analysis. **Xiaolei Fang:** Writing – review & editing, Validation, Supervision, Resources, Funding acquisition. **Zheming Gao:** Writing – review & editing, Funding acquisition, Data curation.

Data availability

All the data used in this research is collected from the public data repositories (UCI, Kaggle, etc.).

Acknowledgments

This work has been sponsored by the National Science Foundation, USA CNS-2229245 and by the National Natural Science Foundation of China under Grant 72201052.

Appendix A. Proofs

A.1. Proof of Lemma 4

Proof. For each $i = 1, \dots, N$, $L(\boldsymbol{z}_i; \boldsymbol{v}, b, \boldsymbol{z}_j, \boldsymbol{a}, \boldsymbol{\beta})$ given $(\boldsymbol{v}, b, \boldsymbol{z}_j, \boldsymbol{a}, \boldsymbol{\beta}), j \neq i$ is convex respect to $\boldsymbol{z}_i$. We derive that

$$\nabla_{\boldsymbol{z}_i} L(\boldsymbol{z}_i; \boldsymbol{v}, b, \boldsymbol{z}_j, \boldsymbol{a}, \boldsymbol{\beta}) = (\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}} \boldsymbol{v} \left(\rho(y_i(\bar{\boldsymbol{K}}^i)^{\mathrm{T}} \boldsymbol{v} + b) + a_i - 1 + \beta_i - \rho\tau(\epsilon)((\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}} \boldsymbol{v})^{\mathrm{T}} \boldsymbol{z}_i \right). \tag{A.1}$$

Note $(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}} \boldsymbol{v} \neq 0$ since $\boldsymbol{\Sigma}_K^i \succ 0$ and WLOG, the decision variable $\boldsymbol{v} \neq 0$. Thus, let $\boldsymbol{z}_i^*$ satisfying $\nabla_{\boldsymbol{z}_i^*} L(\boldsymbol{z}_i; \boldsymbol{v}, b, \boldsymbol{z}_j, \boldsymbol{a}, \boldsymbol{\beta}) = 0$. We have

$$((\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}} \boldsymbol{v})^{\mathrm{T}} \boldsymbol{z}_i^* = \frac{1}{\tau(\epsilon)}(y_i(\bar{\boldsymbol{K}}^i)^{\mathrm{T}} \boldsymbol{v} + b) + a_i - 1 + \frac{1}{\rho\tau(\epsilon)}\beta_i.$$

The solution may not be unique but an optimal direction of $\boldsymbol{z}_i^*$ can be found by KKT conditions of the original problem (DRCKSVM$_{aSOCP}$). The KKT conditions give an insight that the $\boldsymbol{z}_i$ maximizing $((\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}} \boldsymbol{v})^{\mathrm{T}} \boldsymbol{z}_i$ has the same direction with $(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}} \boldsymbol{v}$. Thus, we have

$$\boldsymbol{z}_i^* = \left(\frac{1}{\tau(\epsilon)}(y_i(\bar{\boldsymbol{K}}^i)^{\mathrm{T}} \boldsymbol{v} + b) + a_i - 1 + \frac{1}{\rho\tau(\epsilon)}\beta_i \right) \frac{(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}} \boldsymbol{v}}{\|(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}} \boldsymbol{v}\|_2^2}.$$

Let $h(\beta_i, \boldsymbol{v}, b, a_i) \triangleq \frac{1}{\tau(\epsilon)}(y_i((\bar{\boldsymbol{K}}^i)^{\mathrm{T}} \boldsymbol{v} + b) + a_i - 1) + \frac{1}{\rho\tau(\epsilon)}\beta_i$. Then $\boldsymbol{z}_i^* = h(\beta_i, \boldsymbol{v}, b, a_i) \frac{(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}} \boldsymbol{v}}{\|(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}} \boldsymbol{v}\|_2^2}$. Then the optimal solution of $\operatorname{argmin}_{\boldsymbol{z}_i} L(\boldsymbol{z}_i;$

$\boldsymbol{v}, b, \boldsymbol{z}_j, \boldsymbol{a}, \boldsymbol{\beta}$) can be derived as the projection onto $\mathcal{A}$. We have

$$\Pi_{\mathcal{A}}(\boldsymbol{z}_i^*) = \begin{cases} \boldsymbol{z}_i^*, & \text{if } |h(\beta_i, \boldsymbol{v}, b, a_i)| \leqslant \|(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}}\boldsymbol{v}\|_2, \\ \frac{\|(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}}\boldsymbol{v}\|_2}{|h(\beta_i, \boldsymbol{v}, b, a_i)|}\boldsymbol{z}_i^*, & \text{otherwise.} \end{cases}$$

Then Lemma 4 can be proved accordingly. □

A.2. Proof of Theorem 1

Proof. By Lemmas 2 and 3 and the fact of $(1-\frac{\delta}{2})^2 \geq 1-\delta$, with probability at least $1-\delta$, we have

$$\|\hat{\boldsymbol{\mu}}_\phi^{S_i} - \boldsymbol{\mu}_\phi^i\| \leq r_{1i}, \text{ and } \|\hat{\boldsymbol{\Sigma}}_\phi^{S_i} - \boldsymbol{\Sigma}_\phi^i\|_F \leq r_{2i},$$

provided that $m_i \geq (2+\sqrt{2\ln\frac{4}{\delta}})^2$. For the distributionally robust chance constraints, $\sup_{F_i\in\mathcal{P}_{F_i}} \mathbb{P}_{F_i}\left\{y_i\left(\boldsymbol{w}^{\mathrm{T}}\phi(\tilde{\boldsymbol{x}}^i)+b\right) \leq 1-\xi_i\right\} \leq \epsilon$, we have an equivalent SOC constraint $y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_\phi^i + b) \geq 1-\xi_i+\tau(\epsilon)\|(\boldsymbol{\Sigma}_\phi^i)^{\frac{1}{2}}\boldsymbol{w}\|_2$ by (DRCKSVM$_{SOCP}$). The statement of the theorem then follows below:

$$\begin{aligned}
& \tau(\epsilon)\|(\boldsymbol{\Sigma}_\phi^i)^{\frac{1}{2}}\boldsymbol{w}\|_2 + y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_\phi^i + b) \\
= \;& \tau(\epsilon)\|(\boldsymbol{\Sigma}_\phi^i - \hat{\boldsymbol{\Sigma}}_\phi^{S_i} + \hat{\boldsymbol{\Sigma}}_\phi^{S_i})^{\frac{1}{2}}\boldsymbol{w}\|_2 + y_i(\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{\mu}_\phi^i - \hat{\boldsymbol{\mu}}_\phi^{S_i}) + \boldsymbol{w}^{\mathrm{T}}\hat{\boldsymbol{\mu}}_\phi^{S_i} + b) \\
\leq \;& \tau(\epsilon)\sqrt{\boldsymbol{w}^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_\phi^{S_i}\boldsymbol{w} + \|\boldsymbol{\Sigma}_\phi^i - \hat{\boldsymbol{\Sigma}}_\phi^{S_i}\|_F\|\boldsymbol{w}\boldsymbol{w}^{\mathrm{T}}\|_F} + y_i(\boldsymbol{w}^{\mathrm{T}}\hat{\boldsymbol{\mu}}_\phi^{S_i} + b) \\
& + y_i\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{\mu}_\phi^i - \hat{\boldsymbol{\mu}}_\phi^{S_i}) \\
\leq \;& \tau(\epsilon)\sqrt{\boldsymbol{w}^{\mathrm{T}}(\hat{\boldsymbol{\Sigma}}_\phi^{S_i} + r_{2i}\boldsymbol{I})\boldsymbol{w}} + y_i(\boldsymbol{w}^{\mathrm{T}}\hat{\boldsymbol{\mu}}_\phi^{S_i} + b) + y_i\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{\mu}_\phi^i - \hat{\boldsymbol{\mu}}_\phi^{S_i}) \\
= \;& \tau(\epsilon)\|(\hat{\boldsymbol{\Sigma}}_\phi^{S_i} + r_{2i}\boldsymbol{I})^{\frac{1}{2}}\boldsymbol{w}\|_2 + y_i(\boldsymbol{w}^{\mathrm{T}}\hat{\boldsymbol{\mu}}_\phi^{S_i} + b) + y_i\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{\mu}_\phi^i - \hat{\boldsymbol{\mu}}_\phi^{S_i}),
\end{aligned}$$

and $|y_i\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{\mu}_\phi^i - \hat{\boldsymbol{\mu}}_\phi^{S_i})| \leq \|r_{1i}\boldsymbol{I}\boldsymbol{w}\|_2$. □

A.3. Proof of equivalence between DRCKSVM$_{aSOCP}$ and Problem (13)

Proof. Recall the proposed model (DRCKSVM$_{aSOCP}$) and it is equivalent to

$$\min_{\boldsymbol{v},b} \quad \frac{1}{2}\boldsymbol{v}^{\mathrm{T}}\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}\bar{\boldsymbol{Y}}\boldsymbol{v} + C\sum_{i=1}^N(1 - y_i(\boldsymbol{v}^{\mathrm{T}}\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}^i + b) + \tau(\epsilon)\|(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}}\boldsymbol{v}\|_2)^+, \tag{A.2}$$

where $x^+ \triangleq \max\{0, x\}$ for any $x \in \mathbb{R}$. Let $a_i = 1 - y_i(\boldsymbol{v}^{\mathrm{T}}\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}^i + b) + \tau(\epsilon)\|(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}}\boldsymbol{v}\|_2$, for $i = 1, \ldots, N$. Then the model (A.2) becomes

$$\begin{aligned}
\min_{\boldsymbol{v},b,\boldsymbol{a}} \quad & \frac{1}{2}\boldsymbol{v}^{\mathrm{T}}\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}\bar{\boldsymbol{Y}}\boldsymbol{v} + C\sum_{i=1}^N(a_i)^+ \\
s.t. \quad & a_i = 1 - y_i(\boldsymbol{v}^{\mathrm{T}}\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}^i + b) + \tau(\epsilon)\|(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}}\boldsymbol{v}\|_2, \; i = 1, \ldots, N.
\end{aligned} \tag{A.3}$$

Note that for $i = 1, \ldots, N$, for any $\boldsymbol{z}_i \in \mathbb{R}^m$, we have $\|(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}}\boldsymbol{v}\|_2 = \max_{\|\boldsymbol{z}_i\|_2\leq 1}(\boldsymbol{z}_i)^{\mathrm{T}}(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}}\boldsymbol{v}$. We then derive the constraints in (A.3) as

$$\begin{aligned}
a_i &= 1 - y_i(\boldsymbol{v}^{\mathrm{T}}\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}^i + b) + \tau(\epsilon)\max_{\|\boldsymbol{z}_i\|_2\leq 1}(\boldsymbol{z}_i)^{\mathrm{T}}(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}}\boldsymbol{v} \\
&= 1 - \left[y_i(\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}^i)^{\mathrm{T}} - \tau(\epsilon)\max_{\|\boldsymbol{z}_i\|_2\leq 1}(\boldsymbol{z}_i)^{\mathrm{T}}(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}} \quad y_i\right]\begin{bmatrix}\boldsymbol{v}\\ b\end{bmatrix}.
\end{aligned}$$

Let $\mathbb{1}_{\mathcal{A}}(\boldsymbol{z}_i)$ be the indicator function of the convex set $\mathcal{A} \triangleq \{\boldsymbol{z} \in \mathbb{R}^m | \|\boldsymbol{z}\|_2 \leq 1\}$ defined by

$$\mathbb{1}_{\mathcal{A}}(\boldsymbol{z}_i) = \begin{cases} 0, & \text{if } \boldsymbol{z}_i \in \mathcal{A}, \\ \infty, & \text{otherwise.} \end{cases}$$

Then the model (A.3) can be rewritten as

$$\begin{aligned}
\min_{\boldsymbol{v},b,\boldsymbol{a},\boldsymbol{z}_i} \quad & \frac{1}{2}\boldsymbol{v}^{\mathrm{T}}\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}\bar{\boldsymbol{Y}}\boldsymbol{v} + C\sum_{i=1}^N(a_i)^+ + \sum_{i=1}^N\mathbb{1}_{\mathcal{A}}(\boldsymbol{z}_i) \\
s.t. \quad & a_i = 1 - \left[y_i(\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}^i)^{\mathrm{T}} - \tau(\epsilon)(\boldsymbol{z}_i)^{\mathrm{T}}(\boldsymbol{\Sigma}_K^i)^{\frac{1}{2}} \quad y_i\right]\begin{bmatrix}\boldsymbol{v}\\ b\end{bmatrix}, \; i = 1, \ldots, N.
\end{aligned} \tag{A.4}$$

Let $\boldsymbol{M} = (\bar{\boldsymbol{Y}}[\bar{\boldsymbol{K}}^1, \ldots, \bar{\boldsymbol{K}}^N])^{\mathrm{T}} \in \mathbb{R}^{N\times m}$, $\boldsymbol{e}_N = (1, \ldots, 1)^{\mathrm{T}} \in \mathbb{R}^N$, and $\boldsymbol{Y}$ be a diagonal matrix of labels, i.e., $\boldsymbol{Y} = diag(y_1, \ldots, y_N)$. Then we may rewrite model (A.4) into the matrix format and receive (13). □

Appendix B. Auxiliary information

B.1. The inverse of the sum of matrices

Lemma 5 (*Miller, 1981*). *If $\boldsymbol{A}$ and $\boldsymbol{A}+\boldsymbol{B}$ are invertible, and $\boldsymbol{B}$ has rank 1, then let $g = trace(\boldsymbol{B}\boldsymbol{A}^{-1})$. Then $g \neq -1$ and*

$$(\boldsymbol{A}+\boldsymbol{B})^{-1} = \boldsymbol{A}^{-1} - \frac{1}{g+1}\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{A}^{-1}.$$

In steps of Algorithm , solving a linear system needs to find the inverse matrix of

$$\begin{aligned}
\boldsymbol{D}_N &\triangleq \begin{bmatrix}\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}\bar{\boldsymbol{Y}} & 0\\ 0 & 0\end{bmatrix} + \rho\sum_{i=1}^N\begin{bmatrix} y_i\bar{\boldsymbol{K}}_i - \tau(\epsilon)\boldsymbol{\Sigma}_{K_i}^{\frac{1}{2}}\boldsymbol{z}_i \\ y_i\end{bmatrix}[(y_i\bar{\boldsymbol{K}}_i - \tau(\epsilon)\boldsymbol{\Sigma}_{K_i}^{\frac{1}{2}}\boldsymbol{z}_i)^{\mathrm{T}} \quad y_i] \\
&= \begin{bmatrix}\bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}\bar{\boldsymbol{Y}} + \theta\boldsymbol{I}_n & 0\\ 0 & \theta\end{bmatrix} \\
&\quad + \rho\sum_{i=1}^N\begin{bmatrix} y_i\boldsymbol{\mu}^i - \tau(\epsilon)\boldsymbol{\Sigma}_{K_i}^{\frac{1}{2}}\boldsymbol{z}_i \\ y_i\end{bmatrix}[y_i(\boldsymbol{\mu}^i)^{\mathrm{T}} - \tau(\epsilon)\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{\Sigma}_{K_i}^{\frac{1}{2}} \quad y_i] - \theta\boldsymbol{I}_{n+1} \\
&= \left(\begin{bmatrix}\hat{\boldsymbol{K}} & 0\\ 0 & \theta\end{bmatrix} + \rho\sum_{i=1}^N\begin{bmatrix}\boldsymbol{\sigma}^i\\ y_i\end{bmatrix}[(\boldsymbol{\sigma}^i)^{\mathrm{T}} \quad y_i]\right) - \theta\boldsymbol{I}_{n+1},
\end{aligned}$$

where $\hat{\boldsymbol{K}} = \bar{\boldsymbol{Y}}\bar{\boldsymbol{K}}\bar{\boldsymbol{Y}} + \theta\boldsymbol{I}_n$, and $\boldsymbol{\sigma}^i = y_i\boldsymbol{\mu}^i - \tau(\epsilon)\boldsymbol{\Sigma}_{K_i}^{\frac{1}{2}}\boldsymbol{z}_i \in \mathbb{R}^n$. Applying Woodbury matrix identity, we have

$$\begin{aligned}
\boldsymbol{D}_N^{-1} &= \left(\left(\begin{bmatrix}\hat{\boldsymbol{K}} & 0\\ 0 & \theta\end{bmatrix} + \rho\sum_{i=1}^N\begin{bmatrix}\boldsymbol{\sigma}^i\\ y_i\end{bmatrix}[(\boldsymbol{\sigma}^i)^{\mathrm{T}} \quad y_i]\right) - \theta\boldsymbol{I}_{n+1}\right)^{-1} \\
&= (\boldsymbol{A}_N - \theta\boldsymbol{I}_{n+1})^{-1} \\
&= \boldsymbol{A}_N^{-1} - \boldsymbol{A}_N^{-1}(\boldsymbol{A}_N^{-1} - \tfrac{1}{\theta}\boldsymbol{I}_{n+1})^{-1}\boldsymbol{A}_N^{-1},
\end{aligned} \tag{B.1}$$

where $\boldsymbol{A}_N \triangleq \begin{bmatrix}\hat{\boldsymbol{K}} & 0\\ 0 & \theta\end{bmatrix} + \rho\sum_{i=1}^N\begin{bmatrix}\boldsymbol{\sigma}^i\\ y_i\end{bmatrix}[(\boldsymbol{\sigma}^i)^{\mathrm{T}} \quad y_i]$ has inverse because a proper choice of θ can ensure the positive definiteness of the matrix. By Lemma 5, we have

$$\begin{aligned}
\boldsymbol{A}_N^{-1} &= \left(\left(\begin{bmatrix}\hat{\boldsymbol{K}} & 0\\ 0 & \theta\end{bmatrix} + \rho\sum_{i=1}^{N-1}\begin{bmatrix}\boldsymbol{\sigma}^i\\ y_i\end{bmatrix}[(\boldsymbol{\sigma}^i)^{\mathrm{T}} \quad y_i]\right) + \rho\begin{bmatrix}\boldsymbol{\sigma}_N\\ y_N\end{bmatrix}[\boldsymbol{\sigma}_N^{\mathrm{T}} \quad y_N]\right)^{-1} \\
&= (\boldsymbol{A}_{N-1} + \rho\begin{bmatrix}\boldsymbol{\sigma}_N\\ y_N\end{bmatrix}[\boldsymbol{\sigma}_N^{\mathrm{T}} \quad y_N])^{-1} \\
&= \boldsymbol{A}_{N-1}^{-1} - \frac{1}{[\boldsymbol{\sigma}_N^{\mathrm{T}} \quad y_N]\boldsymbol{A}_{N-1}^{-1}\begin{bmatrix}\boldsymbol{\sigma}_N\\ y_N\end{bmatrix} + 1/\rho}\boldsymbol{A}_{N-1}^{-1}\begin{bmatrix}\boldsymbol{\sigma}_N\\ y_N\end{bmatrix}[\boldsymbol{\sigma}_N^{\mathrm{T}} \quad y_N]\boldsymbol{A}_{N-1}^{-1}.
\end{aligned} \tag{B.2}$$

When $N = 1$, we have

$$\begin{aligned}
\boldsymbol{A}_1^{-1} &= (\begin{bmatrix}\hat{\boldsymbol{K}} & 0\\ 0 & \theta\end{bmatrix} + \rho\begin{bmatrix}\boldsymbol{\sigma}_1\\ y_1\end{bmatrix}[\boldsymbol{\sigma}_1^{\mathrm{T}} \quad y_1])^{-1} \\
&= \begin{bmatrix}\hat{\boldsymbol{K}}^{-1} & 0\\ 0 & \frac{1}{\theta}\end{bmatrix} - \frac{1}{\boldsymbol{\sigma}_1^{\mathrm{T}}\hat{\boldsymbol{K}}\boldsymbol{\sigma}_1 + \theta + 1/\rho}\begin{bmatrix}\hat{\boldsymbol{K}}^{-1}\boldsymbol{\sigma}_1\\ \frac{1}{\theta}y_1\end{bmatrix}\begin{bmatrix}\boldsymbol{\sigma}_1^{\mathrm{T}}\hat{\boldsymbol{K}}^{-1} & \frac{1}{\theta}y_1\end{bmatrix}
\end{aligned}$$

and consequently, $\boldsymbol{A}_N^{-1}$ and $\boldsymbol{D}_N^{-1}$ can be calculated by (B.2) and (B.1).

References

Ben-Tal, A., Bhadra, S., Bhattacharyya, C., Nath, J.S., 2011. Chance constrained uncertain classification via robust optimization. Math. Program. 127 (1), 145–173.

Bertsimas, D., Dunn, J., Pawlowski, C., Zhuo, Y.D., 2019. Robust classification. INFORMS J. Optim. 1 (1), 2–34.

Bhattacharyya, C., Grate, L., Jordan, M.I., Ghaoui, L.E., Mian, I.S., 2004. Robust sparse hyperplane classifiers: Application to uncertain molecular profiling data. J. Comput. Biol. 11 (6), 1073–1089.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends® Mach. Learn. 3 (1), 1–122.

Carrizosa, E., Morales, D.R., 2013. Supervised classification and mathematical optimization. Comput. Oper. Res. 40 (1), 150–165.
Chang, C.-C., Lin, C.-J., 2011. LIBSVM: A library for support vector machines. ACM Trans. Intell. Syst. Technol. 2 (3), 1–27.
Cheramin, M., Cheng, J., Jiang, R., Pan, K., 2022. Computationally efficient approximations for distributionally robust optimization under moment and wasserstein ambiguity. INFORMS J. Comput. 34 (3), 1768–1794.
Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20 (3), 273–297.
Duchi, J., Namkoong, H., 2019. Variance-based regularization with convex objectives. J. Mach. Learn. Res. 20 (68), 1–55.
Duchi, J.C., Namkoong, H., 2021. Learning models with uniform performance via distributionally robust optimization. Ann. Statist. 49 (3), 1378–1406.
Esfahani, M., Alizadeh, A., Amjady, N., Kamwa, I., 2024. A distributed VPP-integrated co-optimization framework for energy scheduling, frequency regulation, and voltage support using data-driven distributionally robust optimization with wasserstein metric. Appl. Energy 361, 122883.
Faccini, D., Maggioni, F., Potra, F.A., 2022. Robust and distributionally robust optimization models for linear support vector machine. Comput. Oper. Res. 147, 105930.
Gao, Z., Fang, S.-C., Luo, J., Medhin, N., 2021. A kernel-free double well potential support vector machine with applications. European J. Oper. Res. 290 (1), 248–262.
Goldfarb, D., Iyengar, G., 2003. Robust convex quadratically constrained programs. Math. Program. 97 (3), 495–515.
Hong, M., Luo, Z.-Q., Razaviyayn, M., 2015. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. arXiv:1410.1390.
Huang, G., Song, S., Gupta, J.N., Wu, C., 2013. A second order cone programming approach for semi-supervised learning. Pattern Recognit. 46 (12), 3548–3558.
Huang, G., Song, S., Wu, C., You, K., 2012. Robust support vector regression for uncertain input and output data. IEEE Trans. Neural Netw. Learn. Syst. 23 (11), 1690–1700.
Jiajin, L., 2021. Efficient and Provable Algorithms for Wasserstein Distributionally Robust Optimization in Machine Learning (Ph.D. thesis). The Chinese University of Hong Kong (Hong Kong).
Khanjani-Shiraz, R., Babapour-Azar, A., Hosseini-Nodeh, Z., Pardalos, P.M., 2023. Distributionally robust joint chance-constrained support vector machines. Optim. Lett. 17 (2), 299–332.
Kuhn, D., Esfahani, P.M., Nguyen, V.A., Shafieezadeh-Abadeh, S., 2019. Wasserstein distributionally robust optimization: Theory and applications in machine learning. In: Operations Research & Management Science in the Age of Analytics. Informs, pp. 130–166.
Lanckriet, G., Ghaoui, L., Bhattacharyya, C., Jordan, M., 2001. Minimax probability machine. Adv. Sing Syst. 14.
Lee, C., Mehrotra, S., 2015. A distributionally-robust approach for finding support vector machines. Available from Optimization Online.
Li, J., Huang, S., So, A.M.-C., 2019. A first-order algorithmic framework for distributionally robust logistic regression. Adv. Neural Inf. Process. Syst. 32.
Lin, F., Fang, X., Gao, Z., 2022. Distributionally robust optimization: A review on theory and applications. Numer. Algebr., Control Optim. 12 (1), 159–212.
Liu, J., Wu, J., Li, B., Cui, P., 2022. Distributionally robust optimization with data geometry. Adv. Neural Inf. Process. Syst. 35, 33689–33701.
Lobo, M.S., Vandenberghe, L., Boyd, S., Lebret, H., 1998. Applications of second-order cone programming. Linear Algebra Appl. 284 (1–3), 193–228.
Luo, J., Fang, S.-C., Deng, Z., Guo, X., 2016. Soft quadratic surface support vector machine for binary classification. Asia-Pac. J. Oper. Res. 33 (6), 1650046.
Marshall, A.W., Olkin, I., 1960. Multivariate Chebyshev inequalities. Ann. Math. Stat. 1001–1014.
Miller, K.S., 1981. On the inverse of the sum of matrices. Math. Mag. 54 (2), 67–72.
Mohseni, S., Pishvaee, M.S., 2023. Energy trading and scheduling in networked microgrids using fuzzy bargaining game theory and distributionally robust optimization. Appl. Energy 350, 121748.
Ohmori, S., 2021. Consensus distributionally robust optimization with phi-divergence. IEEE Access 9, 92204–92213.
Pelckmans, K., De Brabanter, J., Suykens, J.A., De Moor, B., 2005. Handling missing values in support vector machine classifiers. Neural Netw. 18 (5–6), 684–692.
Peng, S., Gianpiero, C., Zhihua, A.-Z., 2023. Chance constrained conic-segmentation support vector machine with uncertain data. Ann. Math. Artif. Intell. 1–23.
Shafieezadeh-Abadeh, S., Kuhn, D., Esfahani, P.M., 2019. Regularization via mass transportation. J. Mach. Learn. Res. 20 (103), 1–68.
Shawe, J., Taylor, N.C., 2003. Estimating the moments of a random vector. In: Proceedings of GRETSI 2003 Conference, I: 47j52.
Shivaswamy, P.K., Bhattacharyya, C., Smola, A.J., 2006. Second order cone programming approaches for handling missing and uncertain data. J. Mach. Learn. Res. 7 (47), 1283–1314.
Singla, M., Ghosh, D., Shukla, K., 2020. A survey of robust optimization based machine learning with special reference to support vector machines. Int. J. Mach. Learn. Cybern. 11 (7), 1359–1385.
Staib, M., Jegelka, S., 2019. Distributionally robust optimization and generalization in kernel methods. Adv. Neural Inf. Process. Syst. 32.
Trafalis, T.B., Gilbert, R.C., 2006. Robust classification and regression using support vector machines. European J. Oper. Res. 173 (3), 893–909.
Vapnik, V.N., 1999. An overview of statistical learning theory. IEEE Trans. Neural Netw. 10 (5), 988–999.
Vassilvitskii, S., Arthur, D., 2006. K-means++: The advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 1027–1035.
Wang, L., 2005. Support vector machines: Theory and applications. In: Machine Learning and Its Applications: Advanced Lectures. Springer, Berlin, Heidelberg.
Wang, X., Fan, N., Pardalos, P.M., 2018. Robust chance-constrained support vector machines with second-order moment information. Ann. Oper. Res. 263 (1), 45–68.
Wang, X., Pardalos, P.M., 2014. A survey of support vector machines with uncertainties. Ann. Data Sci. 1 (3–4), 293–309.
Zhai, J., Jiang, Y., Shi, Y., Jones, C.N., Zhang, X.-P., 2022. Distributionally robust joint chance-constrained dispatch for integrated transmission-distribution systems via distributed optimization. IEEE Trans. Smart Grid 13 (3), 2132–2147.
Zhang, X., Ge, S., Liu, H., Zhou, Y., He, X., Xu, Z., 2023. Distributionally robust optimization for peer-to-peer energy trading considering data-driven ambiguity sets. Appl. Energy 331, 120436.