EURO Journal on Computational Optimization 13 (2025) 100115

Contents lists available at ScienceDirect

EURO Journal on Computational Optimization

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

A robust twin parametric margin support vector machine for multiclass classification

Renato De Leone [a], Francesca Maggioni [b,*], Andrea Spinelli [b]

[a] School of Science and Technology, University of Camerino, Via Madonna delle Carceri 9, Camerino 62032, Italy
[b] Department of Management, Information and Production Engineering, University of Bergamo, Viale G. Marconi 5, Dalmine 24044, Italy

HIGHLIGHTS

- A novel Twin Parametric Margin Support Vector Machine model is proposed.
- The model is designed to tackle multiclass classification tasks.
- The model is safeguarded against uncertainty via robust optimization techniques.
- Both linear and nonlinear classifiers are considered.
- The effectiveness of the proposed robust model is validated on real-world datasets.

ARTICLE INFO

Keywords:
Machine learning
Support vector machine
Robust optimization
Multiclass classification

ABSTRACT

In this paper, we introduce novel Twin Parametric Margin Support Vector Machine (TPMSVM) models designed to address multiclass classification tasks under feature uncertainty. To handle data perturbations, we construct bounded-by-norm uncertainty sets around each training observation and derive the robust counterparts of the deterministic models using robust optimization techniques. To capture complex data structures, we explore both linear and kernel-induced classifiers, providing computationally tractable reformulations of the resulting robust models. Additionally, we propose two alternatives for the final decision function, enhancing models' flexibility. Finally, we validate the effectiveness of the proposed robust multiclass TPMSVM methodology on real-world datasets, showing the good performance of the approach in the presence of uncertainty.

1. Introduction

Supervised classification is one of the most extensively studied tasks in *Machine Learning* (ML) thanks to its wide variety of application fields (see [1]). Although deep learning algorithms and neural networks currently represent the state-of-the-art paradigms in supervised classification, such methodologies do not always guarantee strong predictive accuracies, particularly when applied to tabular data (see [2]). Additionally, these techniques tend to be inefficient for low-dimensional inputs because of their high degree of overparametrization (see [3]). For this reason, the design of innovative data-driven approaches for addressing supervised classification tasks remains a significant field of research (see [4]).

According to [5], two optimization-based approaches to supervised classification can be identified in the literature. The first relies on classification rules derived from mathematical programming models. These techniques aim to minimize misclassification error by optimizing observable criteria. This approach, primarily developed within the Operations Research literature, can be seen as an

* Corresponding author.
Email address: francesca.maggioni@unibg.it (F. Maggioni).

https://doi.org/10.1016/j.ejco.2025.100115

Available online 11 September 2025

extension of the Fisher's *Linear Discriminant Function* (see [6]). The second group of methods, rooted in statistical learning theory, focuses on developing predictive algorithms with guaranteed generalization ability. These classifiers are constructed by optimizing sample accuracy measures associated with distribution-free bounds on misclassification probabilities. This line of research has led to the development of *Support Vector Machines* (SVMs, see [7]).

Classical SVMs consist of identifying the best classifier in the form of a hyperplane or a kernel-induced decision boundary that geometrically separates two sets of labelled training data using a structural risk minimization principle (see [8]). Due to their simplicity and efficiency, SVMs have received strong attention in the ML literature (see [9]). Significant methodological developments have been devised (see, for instance, [10–15]), making SVMs one of the most powerful tools for supervised classification (see [5]). The range of applications includes finance (see [16,17]), scheduling (see [18]), and business analytics (see [19,20]), to name a few.

Among all the possible SVM variants, in this paper, we focus on the *Twin Parametric Margin Support Vector Machine* (TPMSVM, see [21]). According to this method, two nonparallel classifiers, one for each class, are identified such that the training observations of the other class are as far as possible from the opposite classifier. The main advantage of this approach lies in its reduced computational complexity since each of the two classifiers is the solution of a small-sized optimization model. Empirical evidence suggests that TPMSVM achieves superior predictive accuracy compared to other SVM-type techniques (see [21,22]).

Optimization approaches for supervised classification, including SVMs, were traditionally designed to tackle binary classification tasks. However, many real-world applications involve multi-group problems (see [23]), requiring the development of specific methods to address them (see [24]). Typically, these problems are decomposed into a finite sequence of binary classification tasks, whose solutions are finally combined into an aggregate decision function (see [25]). Depending on how the decomposition and the subsequent reconstruction are performed, various approaches have been proposed (see [26]). Nevertheless, multiclass classification problems remain less explored in the ML literature due to their higher computational complexity (see [27]). For this reason, developing new algorithms for multiclass SVMs is considered a promising research area (see [5]).

In supervised classification methods, input data are supposed to be known exactly when training the models. However, real-world observations are often subject to noise and perturbations due to errors in the data collection process or to limited precision of the measurement instruments. In recent years, various techniques have been explored to address uncertainty in classification problems. Among these, *Robust Optimization* (RO) is one of the most widely studied paradigms in the ML literature (see [28]). RO techniques protect the optimization model against the worst possible realizations of the random parameters within a prescribed uncertainty set. Different uncertainty sets lead to distinct solutions, depending on the level of conservatism and the severity of the perturbation. Common choices to define the uncertainty sets often employ ℓ_p-norms (see [29,30]). When RO techniques are applied, the predictive performance of supervised classification algorithms is enhanced, especially in the case of SVMs (see [31,32]). Therefore, designing new robust models for SVMs represents a relevant research direction.

In this paper, we present novel TPMSVM-type models aimed at separating multiple classes of data under feature uncertainty. The formulation builds upon the work of [21] and introduces two key innovations. First, we handle multi-group classification problems instead of two-group tasks. Second, we apply robust optimization techniques to protect the multiclass models against uncertainty. Specifically, we consider bounded-by-ℓ_p-norm uncertainty sets around each training observation and derive the robust counterpart of the deterministic approach. In addition, we provide computationally tractable reformulations of the robust models in the form of *Second Order Cone Programming* (SOCP) models. To improve the generalization capability of the proposal, all the results are derived using both linear and kernel-induced classifiers. To the best of our knowledge, this is the first work in the ML literature to introduce a robust TPMSVM approach for addressing multiclass classification tasks using linear and kernel-induced classifiers under bounded-by-ℓ_p-norm uncertainty sets.

The main contributions of this paper are four-fold and can be summarized as follows:

- to propose new multiclass TPMSVM-type models with both linear and kernel-induced classifiers;
- to formulate the robust counterparts of the deterministic multiclass models using bounded-by-ℓ_p-norm uncertainty sets;
- to derive computationally tractable reformulations of the robust multiclass models as SOCP models for typical choices of the ℓ_p-norm;
- to provide numerical experiments on real-world datasets with the aim of evaluating the performance of the proposed approach and showing the advantages of explicitly accounting for uncertainty in the novel TPMSVM formulations.

The remainder of the paper is organized as follows. Section 2 reviews the existing literature on the problem. Section 3 introduces basic facts about binary TPMSVM model. In Section 4, the deterministic multiclass models are designed, while in Section 5 the robust counterparts are presented. Section 6 reports computational results to evaluate the accuracy of the proposed formulations. Finally, in Section 7 conclusions and future works are discussed.

Throughout the paper, all vectors are column vectors, unless transposed to row vectors by the superscript "$^\top$". For $p \in [1, \infty]$ and $a \in \mathbb{R}^n$, $\|a\|_p$ is the ℓ_p-norm of a. The dot product in an inner product space $\mathcal{H}$ will be denoted by $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. If $\mathcal{H} = \mathbb{R}^n$ and $a, b \in \mathbb{R}^n$, $a^\top b$ is equivalent to $\langle a, b \rangle_{\mathcal{H}}$. If $\mathcal{A}$ is a set, $|\mathcal{A}|$ represents its cardinality. In addition, we denote by e_n the column vector of ones in $\mathbb{R}^n$.

2. Literature review

Classical SVM was first introduced in [7] with the aim of solving classification problems involving two linearly separable sets. The generalization of the linear approach was proposed in [33] by including nonlinear decision boundaries induced by kernel functions. Cases of non perfectly separable training data were considered in [11], where a vector of slack variables was introduced into the SVM

model. The resulting formulation seeks a trade-off between the maximization of the margin, as in the classical SVM approach of [7], and the minimization of an empirical risk related to the slack variables.

When the classification problem involves more than two groups, solution strategies in the SVM literature fall into two categories: *all-together* methods (see [34]) and *decomposition-reconstruction* methods (see [25]). The former considers all training data at the same time within a single optimization model to derive a unified classifier (see [24,35,36]). In contrast, the latter decomposes the problem into a sequence of classification tasks, solving each independently and then combining the resulting SVM-classifiers into an aggregate multiclass decision function. Within this paradigm, various formulations have been proposed. In the *One-Versus-All* strategy (OVA, see [37]), a classifier is constructed for each class aiming to separate data points belonging to that class from all others. In the *One-Versus-One* approach (OVO, see [38]), a binary classification problem is solved for each pair of classes. Conversely, in the *One-Versus-One-Versus-Rest* strategy (OVOVR, see [39]), each subproblem focuses on the separation of a pair of classes together with all the remaining samples by means of two classifiers. These classifiers are close to their respective class, while being as far as possible from the other. At the same time, all remaining points are restricted to a region between the two classifiers. Decomposition-reconstruction methods are generally considered the most effective for multiclass classification problems (see [40]), especially due to the high computational complexity of all-together methods when handling large datasets (see [26]). However, efforts have been made in the literature to overcome these limitations and to unify existing all-together methods (see [41]).

To avoid low classification accuracy when training data are affected by perturbations, optimization under uncertainty techniques are employed within the SVM context. Specifically, *Robust Optimization* (RO, see [42]), *Distributionally Robust Optimization* (DRO, see [43]) and *Chance-Constrained Programming* (CCP, see [44]) are some of the most extensively studied approaches. Robust formulations of standard classification methods, including SVM, are provided in [29]. Methodological advancements and applications of RO techniques to SVMs are discussed in [31,32,45–47]. Recently, an adjustable RO approach for SVM under uncertainty has been proposed in [48]. Within the multiclass framework, in [49] a RO model for SVM is derived through piecewise-linear functions, robustifying the approach of [35] in the case of ellipsoidal uncertainty sets. Finally, [50–52] investigate the integration of CCP and DRO techniques into linear and kernel-induced SVM models, respectively.

Up to this point, we have outlined the general framework of SVMs, including approaches for multiclass classification problems and focusing on optimization techniques for uncertain data. In the following, we discuss SVM variants that are related to the current proposal. We start with the *ν-Support Vector Classification* (ν-SVC) approach, designed in [15]. Compared with the classical SVM presented in [7], a positive parameter ν is introduced in the risk minimization function to bound the fractions of support vectors and misclassification errors. Relying on this approach, in [53] a *parametric margin* model (the par-ν-SVM) is formulated to deal with cases of heteroscedastic noise. Rather than dealing with parallel hyperplanes as in [11], the *TWin Support Vector Machine* (TWSVM) considers a pair of nonparallel classifiers as solutions of two small-sized SVM-type models (see [12]). Consequently, the computational complexity of TWSVM is much reduced compared with the SVM approach of [11]. Due to its favourable performance, especially when handling large datasets, many variants of the TWSVM have been devised in the ML literature: *Least Squares TWSVM* (LS-TWSVM, see [54]), *Projection TWSVM* (P-TWSVM, see [55]), *Twin Parametric Margin SVM* (TPMSVM, see [21]), *Pinball loss TWSVM* (Pin-TWSVM, see [56]), *New Fuzzy TWSVM* (NFTWSVM, see [57]). For a comprehensive overview of recent developments on TWSVM the reader is referred to [58].

Among all the possible TWSVM alternatives, in this paper we focus on the TPMSVM. This formulation combines the contributions of the TWSVM and the par-ν-SVM. Specifically, the TPMSVM constructs two nonparallel classifiers, each of them determining the positive or negative parametric margin, by solving two small-sized optimization models. Therefore, this approach integrates the fast learning speed of the TWSVM and the flexible parametric margin of the par-ν-SVM. Alternative TPMSVM-based formulations are *Structural TPMSVM* (STPMSVM, see [59]), *Least Squares TPMSVM* (LSTPMSVM, see [60]), *Smooth TPMSVM* (STPMSVM, see [61]), *Centroid-based TPMSVM* (CTPSVM, see [62]), *Truncated Pinball Loss TPMSVM* (TPin-TSVM, see [63]).

Similar to SVM, TWSVM and its variants were originally designed for binary classification. To address multiclass classification problems, both all-together and decomposition-reconstruction methods have been explored in the TWSVM literature (see [26,40, 63–65]). In addition to these strategies, we mention the *Directed Acyclic Graph TWSVM* (DAG TWSVM, see [66]), the *Binary Tree TWSVM* (BT TWSVM, see [67]), and the *Multiple Birth SVM* (MBSVM, see [68]). A comprehensive survey on multiclass formulations specifically designed for TWSVM can be found in [26].

In the context of optimization under uncertainty approaches for TWSVM, various techniques have been investigated. Within the RO framework, in [69] a *Robust Minimum Class Variance* model (RMCV-TWSVM) is introduced. To handle data uncertainty, a pair of class variance-covariance matrices is considered, with uncertainty sets defined according to the Frobenius norm. In [70], two nonparallel classifiers are proposed in the case of ellipsoidal uncertainty sets. The corresponding model, called R-TWSVM, is then reformulated as an SOCP model. Instead of convex hulls to represent the training patterns, the *Robust NonParallel SVM* (RNPSVM, see [71]) and the TWin SOCP-SVM (see [72]) consider ellipsoids defined by the first two moments of the class distributions. The models are formulated through CCP techniques, leading to robust SOCP models. A similar CCP approach is employed in [73] for the case of twin multiclass SVM (TWin-KSOCP). Recently, in [74] an improved CCP version of RNPSVM, named IRNPSVM, has been designed to control the number of missing data. Regarding techniques for multiclass classification problems, [22] presents a robust TPMSVM model with an application to vehicle emissions. To the best of our knowledge, this is the first contribution to handle multi-group classification problems using a robust TPMSVM-based methodology. However, [22] considers only linear classifiers and spherical uncertainty sets, without addressing cases with general kernel functions or bounded-by-ℓ_p-norm uncertainty sets.

All the approaches discussed so far on the TWSVM and its variants are schematically reported in Fig. 1 and listed in Table 1.

As discussed above, although robust optimization techniques have been applied to TWSVMs, there remains a gap in the literature concerning robust TPMSVM models. In this regard, the contributions of this paper differ from previous work in several aspects.

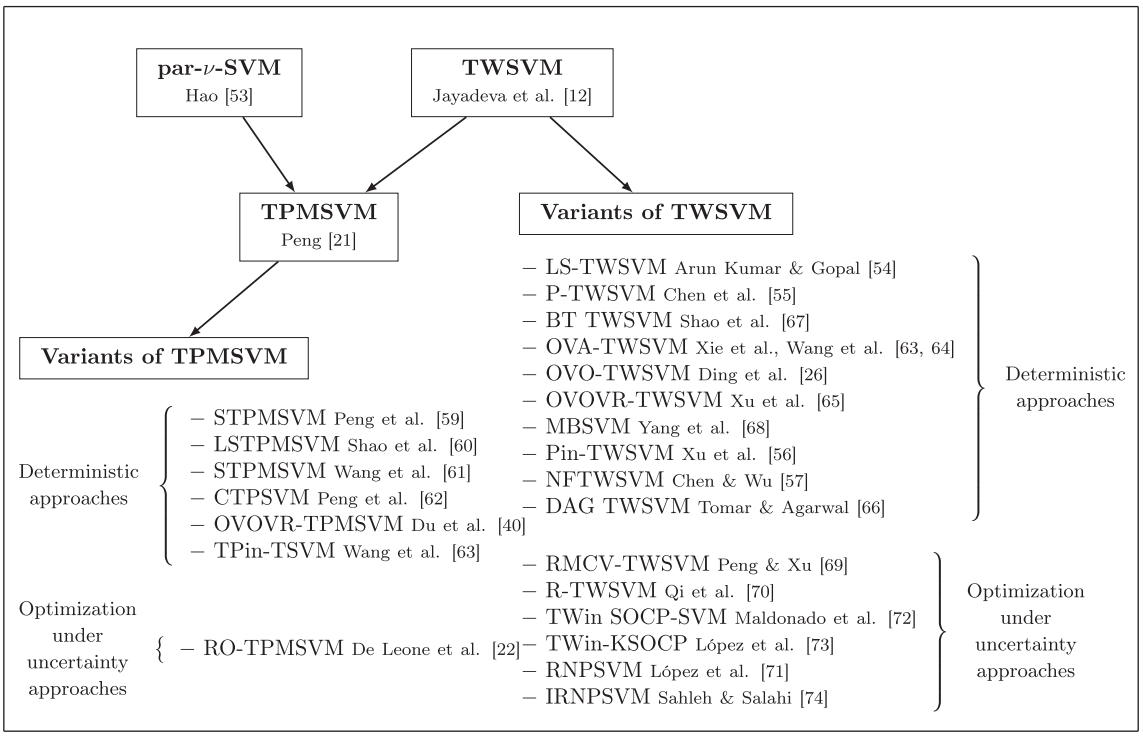


Fig. 1. Scheme of the selected TWSVM literature review. The models are distinguished in deterministic and optimization under uncertainty approaches.

First of all, we present novel TPMSVM-type models to address multiclass classification tasks, extending the binary approach of [21], both for linear and kernel-induced classifiers. Secondly, we consider bounded-by-ℓ_p-norm uncertainty sets around training observations, employing general kernel functions for nonlinear classification problems. Thirdly, we derive the robust counterpart of the deterministic multiclass TPMSVM formulations, protecting the models against feature uncertainty. Finally, we provide tractable reformulations for all robust models as SOCP models, offering clear advantages in terms of computational efficiency.

3. Prior work

In this section, we briefly recall the methods that are relevant to our proposal. Specifically, in Section 3.1 we focus on the linear TPMSVM approach, while in Section 3.2 we discuss its extension to the cases of nonlinear kernel-induced decision boundaries. Both formulations are designed for addressing binary classification tasks and provide an initial framework for our novel multiclass approach.

3.1. The binary TPMSVM for linear classification

Let $\{x^i, y_i\}_{i=1}^m$ be the set of training observations, where $x^i \in \mathbb{R}^n$ is the vector of features, and $y_i \in \{-1, +1\}$ is the label of the ith data point, denoting the class to which it belongs. We assume that each of the two categories is composed of m_- and m_+ observations, respectively, with $m_- + m_+ = m$. We denote by $X_- \in \mathbb{R}^{n\times m_-}$ and $X_+ \in \mathbb{R}^{n\times m_+}$ the matrices of the negative and positive samples, respectively, and $\mathcal{X}_-$ and $\mathcal{X}_+$ the corresponding index sets. For notational convenience, let $e_- := e_{m_-}$ and $e_+ := e_{m_+}$.

The binary TPMSVM approach for linear classification relies on two nonparallel hyperplanes H_+ and H_-, defined by the following equations:

$$H_+ : w_+^\top x + \theta_+ = 0 \qquad H_- : w_-^\top x + \theta_- = 0.$$

The normal vectors $w_+, w_- \in \mathbb{R}^n$ and the intercepts $\theta_+, \theta_- \in \mathbb{R}$ of H_+ and H_- are solutions of the following pair of *Quadratic Programming Problems* (QPPs):

$$\begin{aligned} \min_{w_+,\theta_+,\xi_+} \quad & \frac{1}{2}\|w_+\|_2^2 + \frac{\nu_+}{m_-} e_-^\top \left(X_-^\top w_+ + e_- \theta_+\right) + \frac{\alpha_+}{m_+} e_+^\top \xi_+ \\ \text{s.t.} \quad & X_+^\top w_+ + e_+ \theta_+ \geq -\xi_+ \\ & \xi_+ \geq 0, \end{aligned} \tag{1}$$

Table 1
A selected TWSVM literature review. In the first row of the table the contributions are listed in chronological order. Second and third rows specify the type of TWSVM (linear or nonlinear). Rows four and five consider binary versus multiclass classification. Finally, the optimization under uncertainty methodologies employed in the articles are explored in rows six to eight.

		Jayadeva et al. [12]	Arun Kumar & Gopal [54]	Chen et al. [55]	Peng [21]	Peng and Xu [69]	Peng et al. [59]	Qi et al. [70]	Shao et al. [60]	Shao et al. [67]	Wang et al. [61]	Xie et al. [64]	Xu et al. [65]	Yang et al. [68]	Peng et al. [62]	Maldonado et al. [72]	Maldonado et al. [73]	Xu et al. [56]	Chen & Wu [57]	López et al. [71]	Du et al. [40]	Wang et al. [63]	Sahleh & Salahi [74]	De Leone et al. [22]
TWSVM	Linear classifier	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Nonlinear classifier	✓	✓		✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Classification	Binary	✓	✓	✓	✓	✓	✓	✓	✓		✓				✓	✓		✓	✓	✓		✓	✓	
	Multiclass									✓		✓	✓	✓			✓			✓	✓			✓
Optimization under uncertainty approach	Ellipsoidal RO							✓																✓
	Matrix RO					✓																		
	Chance-Constrained															✓	✓			✓			✓	

and

$$\begin{aligned} \min_{w_-,\theta_-,\xi_-} \quad & \frac{1}{2}\|w_-\|_2^2 - \frac{\nu_-}{m_+} e_+^\top \left(X_+^\top w_- + e_+\theta_-\right) + \frac{\alpha_-}{m_-} e_-^\top \xi_- \\ \text{s.t.} \quad & X_-^\top w_- + e_-\theta_- \le \xi_- \\ & \xi_- \ge 0, \end{aligned} \tag{2}$$

where $\nu_+, \nu_- > 0$, $\alpha_+, \alpha_- > 0$ are regularization parameters, balancing the terms in the objective functions. The vectors $\xi_+ \in \mathbb{R}^{m_+}$, $\xi_- \in \mathbb{R}^{m_-}$ are slack vectors, associated with misclassified samples in each class (see [11]). As in the ν-SVC and par-ν-SVM (see [15,53]), the ratios ν_+/α_+ and ν_-/α_- control the fractions of support vectors and margin errors of each class.

The objective function of model (1) consists of three parts. The first term is related to the margin of the positive class, measured with respect to the ℓ_2-norm. The second term considers the projections of negative training points on H_+, ensuring that these observations are as far as possible from H_+. Finally, the third term is the empirical risk, accounting for the total number of misclassified positive samples. Similar considerations hold for the objective function of model (2).

The dual problems of models (1) and (2) are the following QPPs, respectively:

$$\begin{aligned} \max_{\lambda_+} \quad & -\frac{1}{2}\lambda_+^\top X_+^\top X_+ \lambda_+ + \frac{\nu_+}{m_-} e_-^\top X_-^\top X_+ \lambda_+ \\ \text{s.t.} \quad & e_+^\top \lambda_+ = \nu_+ \\ & 0 \le \lambda_+ \le \frac{\alpha_+}{m_+}, \end{aligned} \tag{3}$$

and

$$\begin{aligned} \max_{\lambda_-} \quad & -\frac{1}{2}\lambda_-^\top X_-^\top X_- \lambda_- + \frac{\nu_-}{m_+} e_+^\top X_+^\top X_- \lambda_- \\ \text{s.t.} \quad & e_-^\top \lambda_- = \nu_- \\ & 0 \le \lambda_- \le \frac{\alpha_-}{m_-}, \end{aligned} \tag{4}$$

where $\lambda_+ \in \mathbb{R}^{m_+}$ and $\lambda_- \in \mathbb{R}^{m_-}$ are the Lagrangian multiplier vectors for each class. Once (3) and (4) are solved, the optimal parameters (w_+, θ_+) and (w_-, θ_-) are computed through the *Karush–Kuhn–Tucker* (KKT) conditions as follows:

$$w_+ = X_+\lambda_+ - \frac{\nu_+}{m_-} X_- e_- \qquad \theta_+ = -\frac{1}{|\mathcal{N}_+|} \sum_{i \in \mathcal{N}_+} {x^i}^\top w_+, \tag{5}$$

and

$$w_- = \frac{\nu_-}{m_+} X_+ e_+ - X_-\lambda_- \qquad \theta_- = -\frac{1}{|\mathcal{N}_-|} \sum_{i \in \mathcal{N}_-} {x^i}^\top w_-, \tag{6}$$

with $\mathcal{N}_+$ the index set of positive training observations x^i, whose corresponding Lagrangian multiplier $\lambda_{+,i}$ satisfies $0 < \lambda_{+,i} < \alpha_+/m_+$. Similarly for $\mathcal{N}_-$.

After the computation of $w_+, w_-, \theta_+, \theta_-$, it is possible to classify each new observation $x \in \mathbb{R}^n$ as negative or positive according to the following decision function:

$$f_{\text{lin}}(x) := \text{sign}\left(\frac{w_+^\top x + \theta_+}{\|w_+\|_2} + \frac{w_-^\top x + \theta_-}{\|w_-\|_2} \right).$$

By way of illustration, in Fig. 2a we depict the hyperplanes H_- and H_+, together with the final classifier $f_{\text{lin}} = 0$, obtained by applying the linear TPMSVM to a bidimensional toy example. Misclassified points are represented as stars.

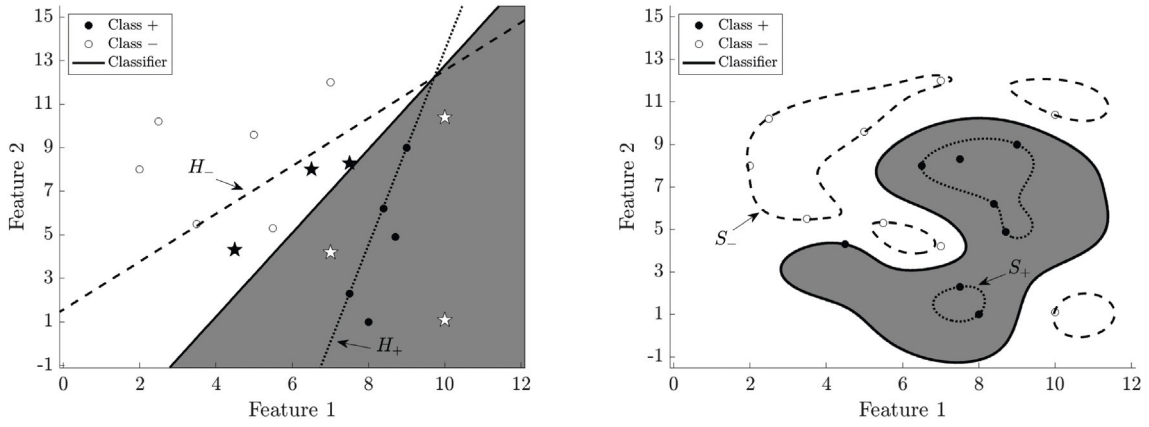
3.2. The binary TPMSVM for nonlinear classification

To increase the predictive power of the model, in [21] the nonlinear kernel-induced version of the TPMSVM approach is provided.

According to the classical procedure introduced in [11], the training input data are first mapped into an inner product space $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$ via a map $\varphi : \mathbb{R}^n \to \mathcal{H}$. The separating hyperplanes are then defined in the feature space $\mathcal{H}$ according to the following expressions:

$$\tilde{H}_+ : \langle \tilde{w}_+, \varphi(x)\rangle_{\mathcal{H}} + \tilde{\theta}_+ = 0 \qquad \tilde{H}_- : \langle \tilde{w}_-, \varphi(x)\rangle_{\mathcal{H}} + \tilde{\theta}_- = 0,$$

with $\tilde{w}_+, \tilde{w}_- \in \mathcal{H}$ and $\tilde{\theta}_+, \tilde{\theta}_- \in \mathbb{R}$.



(a) Linear binary TPMSVM (b) Nonlinear binary TPMSVM

Fig. 2. Linear and nonlinear classifiers for the case of binary TPMSVM. The hyperparameters are $\nu_+ = \nu_- = 0.5$ and $\alpha_+ = \alpha_- = 1$. In the nonlinear case (panel on the right), the Gaussian kernel with $\sigma = 1.5$ is considered. Misclassified points for each class are represented as stars.

Table 2
Examples of kernel functions. The first column reports the names of the functions. The second column provides their mathematical expressions. Finally, the third column contains their relevant parameters.

Kernel function	$k(x, x')$	Parameter
Homogeneous polynomial	$k(x,x') = (x^\top x')^d$	$d \in \mathbb{N}$
Inhomogeneous polynomial	$k(x,x') = (\gamma + x^\top x')^d$	$\gamma \geq 0,\ d \in \mathbb{N}$
Gaussian	$k(x,x') = \exp\left(-\frac{\|x-x'\|_2^2}{2\sigma^2}\right)$	$\sigma > 0$

Models (1) and (2) are modified accordingly as:

$$\begin{aligned} \min_{\tilde{w}_+, \tilde{\theta}_+, \xi_+} \quad & \frac{1}{2}\left\|\tilde{w}_+\right\|_{\mathcal{H}}^2 + \frac{\nu_+}{m_-}\sum_{i\in\mathcal{X}_-}\left(\langle \tilde{w}_+, \varphi(x^i)\rangle_{\mathcal{H}} + \tilde{\theta}_+\right) + \frac{\alpha_+}{m_+} e_+^\top \xi_+ \\ \text{s.t.} \quad & \langle \tilde{w}_+, \varphi(x^i)\rangle_{\mathcal{H}} + \tilde{\theta}_+ \geq -\xi_{+,i} \qquad i \in \mathcal{X}_+ \\ & \xi_+ \geq 0, \end{aligned} \tag{7}$$

and

$$\begin{aligned} \min_{\tilde{w}_-, \tilde{\theta}_-, \xi_-} \quad & \frac{1}{2}\left\|\tilde{w}_-\right\|_{\mathcal{H}}^2 - \frac{\nu_-}{m_+}\sum_{i\in\mathcal{X}_+}\left(\langle \tilde{w}_-, \varphi(x^i)\rangle_{\mathcal{H}} + \tilde{\theta}_-\right) + \frac{\alpha_-}{m_-} e_-^\top \xi_- \\ \text{s.t.} \quad & \langle \tilde{w}_-, \varphi(x^i)\rangle_{\mathcal{H}} + \tilde{\theta}_- \leq \xi_{-,i} \qquad i \in \mathcal{X}_- \\ & \xi_- \geq 0, \end{aligned} \tag{8}$$

where $\|\cdot\|_{\mathcal{H}}$ is the norm induced by the dot product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$, i.e., $\|z\|_{\mathcal{H}} := \sqrt{\langle z, z\rangle_{\mathcal{H}}}$, with $z \in \mathcal{H}$.

Unfortunately, since a closed-form expression of the feature map $\varphi(\cdot)$ is rarely available and the feature space $\mathcal{H}$ is potentially an infinite-dimensional space (see [75,76]), models (7) and (8) cannot be solved in practice (see [77]). To overcome this limitation, it is possible to reformulate their dual problems and efficiently solve them through the so-called *kernel trick* (see [11]). Specifically, a symmetric and positive semidefinite kernel function $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is introduced such that $k(x, x') := \langle \varphi(x), \varphi(x')\rangle_{\mathcal{H}}$, for all $x, x' \in \mathbb{R}^n$. Examples of kernel functions typically used in the ML literature are reported in Table 2. The reader is referred to [76] for a comprehensive overview of kernel functions.

Thus, the dual problems of models (7) and (8) can be reformulated in terms of the kernel function as follows:

$$\begin{aligned} \max_{\lambda_+} \quad & -\frac{1}{2}\lambda_+^\top K(X_+, X_+)\lambda_+ + \frac{\nu_+}{m_-} e_-^\top K(X_-, X_+)\lambda_+ \\ \text{s.t.} \quad & e_+^\top \lambda_+ = \nu_+ \\ & 0 \leq \lambda_+ \leq \frac{\alpha_+}{m_+}, \end{aligned} \tag{9}$$

and

$$\begin{aligned} \max_{\lambda_-} \quad & -\frac{1}{2}\lambda_-^\top K(X_-, X_-)\lambda_- + \frac{\nu_-}{m_+} e_+^\top K(X_+, X_-)\lambda_- \\ \text{s.t.} \quad & e_-^\top \lambda_- = \nu_- \\ & 0 \le \lambda_- \le \frac{\alpha_-}{m_-}, \end{aligned} \tag{10}$$

where $K(X_+, X_+)$ is the Gram matrix of the dot products $k(x^i, x^j)$ for $i, j \in \mathcal{X}_+$. Similarly with $K(X_+, X_-)$, $K(X_-, X_+)$ and $K(X_-, X_-)$.

As in the linear case, once problems (9) and (10) are solved, the KKT conditions provide $(\tilde{w}_+, \tilde{\theta}_+)$ and $(\tilde{w}_-, \tilde{\theta}_-)$, defining the hyperplanes $\tilde{H}_+$ and $\tilde{H}_-$ in the feature space $\mathcal{H}$. These hyperplanes, in turn, induce nonlinear decision boundaries S_+ and S_- in the input space $\mathbb{R}^n$.

Finally, the decision function in the case of binary TPMSVM with nonlinear classifiers is given as follows:

$$f_{\text{nonlin}}(x) := \text{sign}\left(\frac{\langle \tilde{w}_+, \varphi(x)\rangle_{\mathcal{H}} + \tilde{\theta}_+}{\|\tilde{w}_+\|_{\mathcal{H}}} + \frac{\langle \tilde{w}_-, \varphi(x)\rangle_{\mathcal{H}} + \tilde{\theta}_-}{\|\tilde{w}_+\|_{\mathcal{H}}} \right).$$

Fig. 2b shows the separating hypersurfaces S_+ and S_- induced by a Gaussian kernel when classifying the same bidimensional toy example as in Fig. 2a.

4. A novel multiclass TPMSVM-type model

In this section, we extend the binary TPMSVM approach recalled so far to address multiclass classification problems, considering both linear (Section 4.1) and nonlinear (Section 4.2) decision boundaries. Among all the possible multiclass strategies, we opt for the one-versus-all formulation due to its lower computational complexity and strong performance in terms of accuracy (see [26]).

In the following, we assume that C represents the total number of classes and, for each class $c = 1, \dots, C$, the subscript $\cdot_{-c}$ denotes the set of points not belonging to class c.

4.1. The multiclass TPMSVM for linear classification

Let $\{x^i, y_i\}_{i=1}^m$ be the set of training samples, with $y_i \in \{1, \dots, C\}$. For each class c, with $c = 1, \dots, C$, we denote by m_c the number of observations belonging to class c and $m_{-c} := m - m_c$. Matrix $X_c \in \mathbb{R}^{n \times m_c}$ comprises all training data points of class c, and correspondingly, matrix $X_{-c} \in \mathbb{R}^{n \times m_{-c}}$ includes the data for all the other classes. The associated index sets are $\mathcal{X}_c$ and $\mathcal{X}_{-c}$, respectively, and $\mathcal{X} := \mathcal{X}_c \cup \mathcal{X}_{-c}$. Let $e_c := e_{m_c}$ and $e_{-c} := e_{m_{-c}}$.

According to the one-versus-all strategy, for each class $c = 1, \dots, C$ we aim to find the best separating hyperplane H_c defined by equation $w_c^\top x + \theta_c = 0$, where $w_c \in \mathbb{R}^n$ and $\theta_c \in \mathbb{R}$ are solutions of the following QPP:

$$\begin{aligned} \min_{w_c, \theta_c, \xi_c} \quad & \frac{1}{2}\|w_c\|_2^2 + \frac{\nu_c}{m_{-c}} e_{-c}^\top \left(X_{-c}^\top w_c + e_{-c}\theta_c \right) + \frac{\alpha_c}{m_c} e_c^\top \xi_c \\ \text{s.t.} \quad & X_c^\top w_c + e_c \theta_c \ge -\xi_c \\ & \xi_c \ge 0. \end{aligned} \tag{11}$$

Parameters $\nu_c > 0$, $\alpha_c > 0$ and slack vector $\xi_c \in \mathbb{R}^{m_c}$ have an equivalent interpretation to the corresponding terms in model (1).

By introducing the Lagrangian function of problem (11), the KKT conditions lead to closed-form expressions for the normal vector w_c and the intercept θ_c of the hyperplane H_c, for all $c = 1, \dots, C$, following a similar approach to the one used to derive (5).

Once all the C hyperplanes have been determined, we propose two possible alternatives for the decision function:

$$f_{\text{lin,min}}(x) := \arg\min_{c=1,\dots,C} \frac{|w_c^\top x + \theta_c|}{\|w_c\|_2}, \tag{12}$$

and

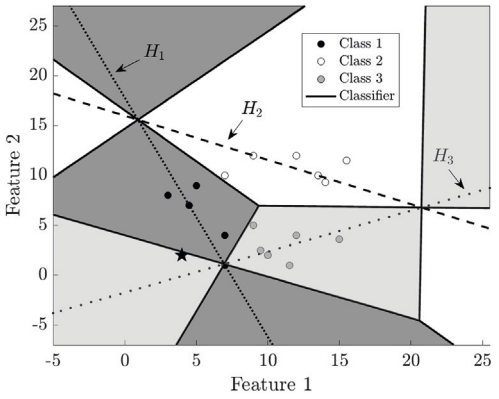
$$f_{\text{lin,max}}(x) := \arg\max_{c=1,\dots,C} \frac{w_c^\top x + \theta_c}{\|w_c\|_2}. \tag{13}$$

In Fig. 3a and b, we represent the results of the proposed linear methodology in the case of a multiclass classification task. We consider a bidimensional toy example with three classes (black, grey, and white points, respectively). The hyperplanes H_1, H_2, H_3 are depicted, along with the decision functions computed according to formulas (12) and (13).

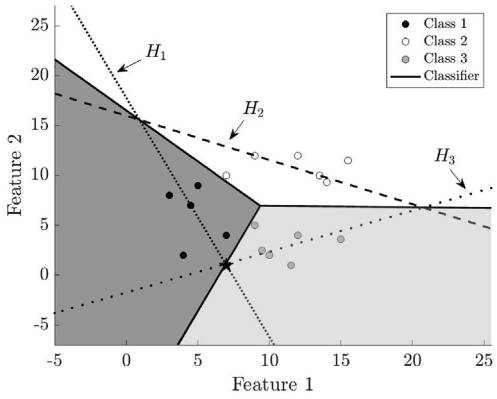
4.2. The multiclass TPMSVM for nonlinear classification

When dealing with nonlinear classifiers, model (11) is defined in the feature space $\mathcal{H}$, leading to the following formulation:

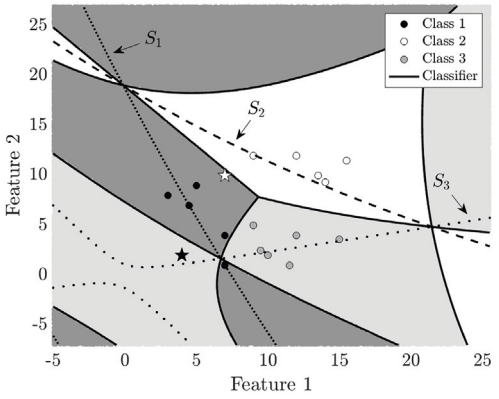
$$\begin{aligned} \min_{\tilde{w}_c, \tilde{\theta}_c, \xi_c} \quad & \frac{1}{2}\|\tilde{w}_c\|_{\mathcal{H}}^2 + \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \left(\langle \tilde{w}_c, \varphi(x^i)\rangle_{\mathcal{H}} + \tilde{\theta}_c \right) + \frac{\alpha_c}{m_c} \sum_{i \in \mathcal{X}_c} \xi_{c,i} \\ \text{s.t.} \quad & \langle \tilde{w}_c, \varphi(x^i)\rangle_{\mathcal{H}} + \tilde{\theta}_c \ge -\xi_{c,i} \qquad i \in \mathcal{X}_c \\ & \xi_{c,i} \ge 0 \qquad i \in \mathcal{X}_c. \end{aligned} \tag{14}$$



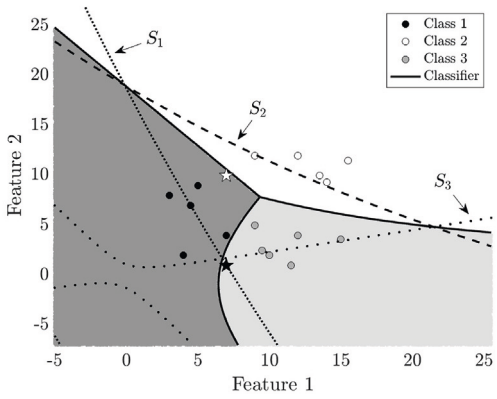
(a) Linear multiclass TPMSVM with argmin formula (12)



(b) Linear multiclass TPMSVM with argmax formula (13)



(c) Nonlinear multiclass TPMSVM with argmin formula (16)



(d) Nonlinear multiclass TPMSVM with argmax formula (17)

Fig. 3. Linear and nonlinear classifiers for the case of three-classes TPMSVM. The hyperparameters are $\nu_c = 0.5$ and $\alpha_c = 1$ for $c = 1, 2, 3$. In the nonlinear case (panels (c) and (d)), the inhomogeneous polynomial kernel with $d = 2$ and $\gamma = 1.5$ is considered. Misclassified points for each class are represented as stars.

As discussed in Section 3.2, the KKT conditions provide the optimal solutions $(\tilde{w}_c, \tilde{\theta}_c)$ in terms of the Lagrangian multiplier $\lambda_c \in \mathbb{R}^{m_c}$ as follows:

$$\tilde{w}_c = \sum_{i \in \mathcal{X}_c} \lambda_{c,i} \varphi(x^i) - \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \varphi(x^i) \qquad \tilde{\theta}_c = -\frac{1}{|\mathcal{N}_c|} \sum_{i \in \mathcal{N}_c} \langle \varphi(x^i), \tilde{w}_c \rangle_{\mathcal{H}}, \tag{15}$$

where $\mathcal{N}_c$ is the index set of observations x^i, with $i \in \mathcal{X}_c$, whose corresponding Lagrangian multiplier satisfies $0 < \lambda_{c,i} < \alpha_c / m_c$.

Within the nonlinear context, we propose as decision functions the following:

$$f_{\text{nonlin,min}}(x) := \underset{c=1,\dots,C}{\arg\min} \frac{\left| \langle \tilde{w}_c, \varphi(x) \rangle_{\mathcal{H}} + \tilde{\theta}_c \right|}{\left\| \tilde{w}_c \right\|_{\mathcal{H}}}, \tag{16}$$

and

$$f_{\text{nonlin,max}}(x) := \underset{c=1,\dots,C}{\arg\max} \frac{\langle \tilde{w}_c, \varphi(x) \rangle_{\mathcal{H}} + \tilde{\theta}_c}{\left\| \tilde{w}_c \right\|_{\mathcal{H}}}. \tag{17}$$

For all $c = 1, \dots, C$, each term in (16) and (17) is computed using the kernel trick, resulting in:

$$\langle \tilde{w}_c, \varphi(x) \rangle_{\mathcal{H}} = \lambda_c^\top K(X_c, x) - \frac{\nu_c}{m_{-c}} e_{-c}^\top K(X_{-c}, x),$$

$$\tilde{\theta}_c = -\frac{1}{|\mathcal{N}_c|} \sum_{i \in \mathcal{N}_c} \left[K(x^i, X_c) \lambda_c - \frac{\nu_c}{m_{-c}} K(x^i, X_{-c}) e_{-c} \right],$$

and

$$\|\tilde{w}_c\|_{\mathcal{H}}^2 = \langle \tilde{w}_c, \tilde{w}_c \rangle_{\mathcal{H}} = \lambda_c^\top K(X_c, X_c)\lambda_c - \frac{\nu_c}{m_{-c}} \lambda_c^\top K(X_c, X_{-c}) e_{-c} + \\ - \frac{\nu_c}{m_{-c}} e_{-c}^\top K(X_{-c}, X_c)\lambda_c + \frac{\nu_c^2}{m_{-c}^2} e_{-c}^\top K(X_{-c}, X_{-c}) e_{-c}.$$

The computational complexity of the binary TPMSVM is approximately $\mathcal{O}(m^3/4)$, as it involves solving two QPPs, each of size $m/2$ (see [21]). In the proposed multiclass extension, one QPP is solved for each of the C classes, both in the linear and kernelized settings. As a result, the overall computational complexity of our multiclass TPMSVM approach is $C \cdot \mathcal{O}(m^3/8)$.

In Fig. 3c and d, we illustrate the results of model (14), considering the same bidimensional toy dataset as in Fig. 3a and b. The hypersurfaces S_1, S_2, S_3 are induced by an inhomogeneous quadratic kernel, and the decision functions are computed according to formulas (16) and (17).

5. The robust multiclass TPMSVM-type model

In this section, we derive the robust counterpart formulations of models (11) and (14). Specifically, we start by constructing bounded-by-ℓ_p-norm uncertainty sets around training data points. The deterministic approaches presented in Section 4 are then robustified by optimizing over the worst-case realization of the uncertain data across the entire uncertainty sets. Tractable SOCP reformulations are finally provided.

Section 5.1 focuses on the robust extension of multiclass TPMSVM for linear classification, while Section 5.2 explores cases with kernel-induced decision boundaries.

5.1. The robust TPMSVM for linear multiclass classification

As in [32], in the construction of the uncertainty set we assume that each training observation $x^i \in \mathbb{R}^n$ is subject to an unknown bounded-by-ℓ_p-norm perturbation $\delta^i \in \mathbb{R}^n$, with $p \in [1, \infty]$. Therefore, the uncertainty set around x^i can be written as:

$$\mathcal{U}_p(x^i) := \left\{ x \in \mathbb{R}^n | x = x^i + \delta^i, \|\delta^i\|_p \le \varepsilon_i \right\}. \tag{18}$$

The radius $\varepsilon_i \ge 0$ controls the degree of conservatism: when $\varepsilon_i = 0$, the uncertainty set $\mathcal{U}_p(x^i)$ reduces to its center x^i.

Robustifying model (11) against the uncertainty set $\mathcal{U}_p(x^i)$ yields the following optimization model:

$$\begin{aligned} \min_{w_c, \theta_c, \xi_c} \quad & \frac{1}{2}\|w_c\|_2^2 + \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \max_{\|\delta^i\|_p \le \varepsilon_i} \left[(x^{i\top} + \delta^{i\top}) w_c + \theta_c \right] + \frac{\alpha_c}{m_c} \sum_{i \in \mathcal{X}_c} \xi_{c,i} \\ \text{s.t.} \quad & x^\top w_c + \theta_c \ge -\xi_{c,i} \qquad i \in \mathcal{X}_c, \forall x \in \mathcal{U}_p(x^i) \\ & \xi_{c,i} \ge 0 \qquad i \in \mathcal{X}_c. \end{aligned} \tag{19}$$

Since there exist infinite possibilities for choosing $x \in \mathcal{U}_p(x^i)$ in the first set of constraints, model (19) is not solvable in practice. However, a tractable closed-form reformulation is provided in the following theorem.

Theorem 1. *Let $\mathcal{U}_p(x^i)$ be the uncertainty set as in (18), with $p \in [1, \infty]$. Let p' be the Hölder conjugate of p, namely $1/p + 1/p' = 1$. Model (19) is equivalent to:*

$$\begin{aligned} \min_{w_c, \theta_c, \xi_c} \quad & \frac{1}{2}\|w_c\|_2^2 + \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \left(x^{i\top} w_c + \varepsilon_i \|w_c\|_{p'} \right) + \nu_c \theta_c + \frac{\alpha_c}{m_c} \sum_{i \in \mathcal{X}_c} \xi_{c,i} \\ s.t. \quad & x^{i\top} w_c + \theta_c - \varepsilon_i \|w_c\|_{p'} \ge -\xi_{c,i} \qquad i \in \mathcal{X}_c \\ & \xi_{c,i} \ge 0 \qquad i \in \mathcal{X}_c. \end{aligned} \tag{20}$$

Proof. Following the RO approach of [29], model (19) can be expressed as follows:

$$\begin{aligned} \min_{w_c, \theta_c, \xi_c} \quad & \frac{1}{2}\|w_c\|_2^2 + \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \max_{\|\delta^i\|_p \le \varepsilon_i} \left[(x^{i\top} + \delta^{i\top}) w_c + \theta_c \right] + \frac{\alpha_c}{m_c} \sum_{i \in \mathcal{X}_c} \xi_{c,i} \\ \text{s.t.} \quad & \min_{\|\delta^i\|_p \le \varepsilon_i} \left[(x^{i\top} + \delta^{i\top}) w_c \right] + \theta_c \ge -\xi_{c,i} \qquad i \in \mathcal{X}_c \\ & \xi_{c,i} \ge 0 \qquad i \in \mathcal{X}_c. \end{aligned} \tag{21}$$

The maximization term in the objective function corresponds to:

$$\frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \max_{\|\delta^i\|_p \le \varepsilon_i} \left[x^{i\top} w_c + \delta^{i\top} w_c + \theta_c \right] = \nu_c \theta_c + \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \left(x^{i\top} w_c + \max_{\|\delta^i\|_p \le \varepsilon_i} \left[\delta^{i\top} w_c \right] \right). \tag{22}$$

Similarly, the minimization term in the first set of constraints is equivalent to:

$$\min_{\|\delta^i\|_p \le \varepsilon_i} \left[x^{i\top} w_c + \delta^{i\top} w_c \right] = x^{i\top} w_c + \min_{\|\delta^i\|_p \le \varepsilon_i} \left[\delta^{i\top} w_c \right]. \tag{23}$$

We observe that the optimization problems in (22) and in (23) have the same feasible set, $\{\delta^i : \|\delta^i\|_p \leq \varepsilon_i\}$, and the same objective function, ${\delta^i}^\top w_c$. By applying the Hölder inequality (see [78]), we get:

$$-\varepsilon_i \|w_c\|_{p'} \leq {\delta^i}^\top w_c \leq \varepsilon_i \|w_c\|_{p'} \qquad \forall \delta^i : \|\delta^i\|_p \leq \varepsilon_i, \tag{24}$$

where p' is the Hölder conjugate of p. Consequently, the optimization problems in (22) and in (23) have the following optimal values:

$$\max_{\|\delta^i\|_p \leq \varepsilon_i} \left[{\delta^i}^\top w_c\right] = \varepsilon_i \|w_c\|_{p'} \quad \text{and} \quad \min_{\|\delta^i\|_p \leq \varepsilon_i} \left[{\delta^i}^\top w_c\right] = -\varepsilon_i \|w_c\|_{p'}. \tag{25}$$

Therefore, the second term in the objective function of (21) corresponds to:

$$\nu_c \theta_c + \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \left({x^i}^\top w_c + \varepsilon_i \|w_c\|_{p'} \right),$$

while the first set of constraints can be written as follows:

$${x^i}^\top w_c + \theta_c - \varepsilon_i \|w_c\|_{p'} \geq -\xi_{c,i}, \quad i \in \mathcal{X}_c.$$

This concludes the proof. □

If no uncertainty occurs in the training samples, $\varepsilon_i = 0$ for all $i \in \mathcal{X}$ and the robust model (20) reduces to the deterministic model (11). We notice that model (20) is a convex nonlinear optimization model due to the presence of the ℓ_2- and $\ell_{p'}$-norm of w_c. However, the quadratic term $\|w_c\|_2^2$ can be easily transformed from the objective function to the constraints by introducing auxiliary variables $t_c, u_c, v_c \in \mathbb{R}$ (see [70]), leading to:

$$\begin{aligned}
\min_{w_c, \theta_c, \xi_c, t_c, u_c, v_c} \quad & \frac{1}{2}(u_c - v_c) + \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \left({x^i}^\top w_c + \varepsilon_i \|w_c\|_{p'} \right) + \nu_c \theta_c + \frac{\alpha_c}{m_c} \sum_{i \in \mathcal{X}_c} \xi_{c,i} \\
\text{s.t.} \quad & {x^i}^\top w_c + \theta_c - \varepsilon_i \|w_c\|_{p'} \geq -\xi_{c,i} \qquad i \in \mathcal{X}_c \\
& t_c \geq \|w_c\|_2 \\
& u_c + v_c = 1 \\
& u_c \geq \sqrt{t_c^2 + v_c^2} \\
& \xi_{c,i} \geq 0 \qquad i \in \mathcal{X}_c.
\end{aligned} \tag{26}$$

Model (26) is still nonlinear, but for specific choices of the ℓ_p-norm used in the definition of the uncertainty set (18), it can be reformulated as an SOCP model, as stated in the following result.

Corollary 1. *Let $\mathcal{U}_p(x^i)$ be the uncertainty set as in (18). Model (26) can be expressed as an SOCP model in the following cases:*

(a) Case $p = 1$:

$$\begin{aligned}
\min_{w_c, \theta_c, \xi_c, t_c, u_c, v_c, s_c} \quad & \frac{1}{2}(u_c - v_c) + \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \left({x^i}^\top w_c + \varepsilon_i s_c \right) + \nu_c \theta_c + \frac{\alpha_c}{m_c} \sum_{i \in \mathcal{X}_c} \xi_{c,i} \\
s.t. \quad & {x^i}^\top w_c + \theta_c - \varepsilon_i s_c \geq -\xi_{c,i} \qquad i \in \mathcal{X}_c \\
& t_c \geq \|w_c\|_2 \\
& u_c + v_c = 1 \\
& u_c \geq \sqrt{t_c^2 + v_c^2} \\
& s_c \geq -w_{c,j} \qquad j = 1, \ldots, n \\
& s_c \geq w_{c,j} \qquad j = 1, \ldots, n \\
& s_c \geq 0 \\
& \xi_{c,i} \geq 0 \qquad i \in \mathcal{X}_c.
\end{aligned} \tag{27}$$

(b) Case $p = 2$:

$$\begin{aligned}
\min_{w_c,\theta_c,\xi_c,t_c,u_c,v_c} \quad & \frac{1}{2}(u_c - v_c) + \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \left(x^{i^\top} w_c + \varepsilon_i t_c \right) + \nu_c \theta_c + \frac{\alpha_c}{m_c} \sum_{i \in \mathcal{X}_c} \xi_{c,i} \\
s.t. \quad & x^{i^\top} w_c + \theta_c - \varepsilon_i t_c \geq -\xi_{c,i} \qquad i \in \mathcal{X}_c \\
& t_c \geq \|w_c\|_2 \\
& u_c + v_c = 1 \\
& u_c \geq \sqrt{t_c^2 + v_c^2} \\
& \xi_{c,i} \geq 0 \qquad i \in \mathcal{X}_c.
\end{aligned} \tag{28}$$

(c) Case $p = \infty$:

$$\begin{aligned}
\min_{w_c,\theta_c,\xi_c,t_c,u_c,v_c,s_c} \quad & \frac{1}{2}(u_c - v_c) + \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \left(x^{i^\top} w_c + \varepsilon_i \sum_{j=1}^{n} s_{c,j} \right) + \nu_c \theta_c + \frac{\alpha_c}{m_c} \sum_{i \in \mathcal{X}_c} \xi_{c,i} \\
s.t. \quad & x^{i^\top} w_c + \theta_c - \varepsilon_i \sum_{j=1}^{n} s_{c,j} \geq -\xi_{c,i} \qquad i \in \mathcal{X}_c \\
& t_c \geq \|w_c\|_2 \\
& u_c + v_c = 1 \\
& u_c \geq \sqrt{t_c^2 + v_c^2} \\
& s_{c,j} \geq -w_{c,j} \qquad j = 1, \dots, n \\
& s_{c,j} \geq w_{c,j} \qquad j = 1, \dots, n \\
& s_{c,j} \geq 0 \qquad j = 1, \dots, n \\
& \xi_{c,i} \geq 0 \qquad i \in \mathcal{X}_c.
\end{aligned} \tag{29}$$

Proof.

(a) If $p = 1$, then $p' = \infty$. By introducing an auxiliary variable $s_c \geq 0$ equal to $\|w_c\|_\infty$, and adding the constraints $s_c \geq -w_{c,j}$ and $s_c \geq w_{c,j}$ for all $j = 1, \dots, n$, model (26) is equivalent to model (27).
(b) If $p = 2$, then $p' = 2$, and so $\|w_c\|_{p'} = \|w_c\|_2 = t_c$. The equivalence between models (26) and (28) follows straightforwardly.
(c) If $p = \infty$, then $p' = 1$. In this case model (26) can be rewritten as model (29) by introducing an auxiliary vector $s_c \in \mathbb{R}^n$ such that each component $s_{c,j}$ is equal to $\left| w_{c,j} \right|$ and adding the constraints $s_{c,j} \geq 0$, $s_{c,j} \geq -w_{c,j}$ and $s_{c,j} \geq w_{c,j}$ for all $j = 1, \dots, n$. □

As in the deterministic setting (see Section 4.1), once the optimal solutions (w_c, θ_c) are obtained for all $c = 1, \dots, C$, the classification of each new observation is performed according to decision functions (12) and (13).

SOCP models are typically solved using specialized algorithms. One of the most widely used is the *primal-dual interior-point method*, whose computational complexity is bounded by $\mathcal{O}\left(N^{7/2} \log(1/\eta)\right)$, where N denotes the total number of variables and η is the duality gap threshold that determines the solution accuracy (see [79]). For the robust models in problems (27)–(29), the corresponding problem sizes are $N = n + m_c + 5$, $N = n + m_c + 4$, and $N = 2n + m_c + 4$, respectively. Since an SOCP model is solved independently for each class, the overall time complexity scales with the number of classes C, yielding a bound on the total computational complexity of $C \cdot \mathcal{O}\left(N^{7/2} \log(1/\eta)\right)$ for each formulation. Note that, since $m_c \leq m$, we can substitute m_c with m in each expression of N to obtain a valid upper bound on the total complexity.

5.2. The robust TPMSVM for nonlinear multiclass classification

Let $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be a symmetric and positive semidefinite kernel, with feature map $\varphi : \mathcal{H} \to \mathbb{R}$. Similarly to [30,32], for each training observation $x^i \in \mathbb{R}^n$, we model the uncertainty set centered at $\varphi(x^i) \in \mathcal{H}$ as follows:

$$\mathcal{U}_{\mathcal{H}}\left(\varphi(x^i)\right) := \left\{ z \in \mathcal{H} | z = \varphi(x^i) + \tilde{\delta}^i, \left\|\tilde{\delta}\right\|_{\mathcal{H}}^i \leq \tilde{\varepsilon}_i \right\}, \tag{30}$$

where the perturbation $\tilde{\delta}^i$ belongs to $\mathcal{H}$ and its $\mathcal{H}$-norm is bounded by a constant $\tilde{\varepsilon}_i \geq 0$. The value of $\tilde{\varepsilon}_i$ depends on the radius ε_i of the corresponding uncertainty set $\mathcal{U}_p(x^i)$ in the input space (18). Closed-form expressions for $\tilde{\varepsilon}_i$ for kernel functions $k(\cdot, \cdot)$ typically used in the ML literature are derived in [32]. Additionally, a perturbation $\tilde{\delta}^i$ in the feature space arises if and only if a perturbation δ^i occurs in the input space. Consequently, $\varepsilon_i = 0$ implies $\tilde{\varepsilon}_i = 0$.

As in Section 5.1, we start by robustifying model (14) over the uncertainty set (30), obtaining the following optimization model:

$$\begin{aligned} \min_{\tilde{w}_c,\tilde{\theta}_c,\xi_c} \quad & \frac{1}{2}\|\tilde{w}_c\|_{\mathcal{H}}^2 + \frac{\nu_c}{m_{-c}} \sum_{i\in\mathcal{X}_{-c}} \max_{\|\tilde{\delta}^i\|_{\mathcal{H}}\le\tilde{\varepsilon}_i} \left[\langle \tilde{w}_c, \varphi(x^i)+\tilde{\delta}^i\rangle_{\mathcal{H}} + \tilde{\theta}_c\right] + \frac{\alpha_c}{m_c}\sum_{i\in\mathcal{X}_c}\xi_{c,i} \\ \text{s.t.} \quad & \langle \tilde{w}_c, z\rangle_{\mathcal{H}} + \tilde{\theta}_c \ge -\xi_{c,i} \qquad i\in\mathcal{X}_c, \forall z\in\mathcal{U}_{\mathcal{H}}\left(\varphi(x^i)\right) \\ & \xi_{c,i}\ge 0 \qquad i\in\mathcal{X}_c. \end{aligned} \tag{31}$$

Model (31) is intractable in practice, due to the infinite possible choices of $z\in\mathcal{U}_{\mathcal{H}}\left(\varphi(x^i)\right)$ in the first set of constraints. Nevertheless, it can be reformulated into a tractable form, as stated in the following theorem.

Theorem 2. *Let $\mathcal{U}_{\mathcal{H}}\left(\varphi(x^i)\right)$ be the uncertainty set as in (30). Model (31) is equivalent to:*

$$\begin{aligned} \min_{\beta_c,\tilde{\theta}_c,\xi_c} \quad & \frac{1}{2}\beta_c^\top K\beta_c + \frac{\nu_c}{m_{-c}}\sum_{i\in\mathcal{X}_{-c}}\left[\tilde{\varepsilon}_i\sqrt{\beta_c^\top K\beta_c} + \sum_{j\in\mathcal{X}}\beta_{c,j}k(x^i,x^j)\right] + \nu_c\tilde{\theta}_c + \frac{\alpha_c}{m_c}\sum_{i\in\mathcal{X}_c}\xi_{c,i} \\ s.t. \quad & \tilde{\theta}_c - \tilde{\varepsilon}_i\sqrt{\beta_c^\top K\beta_c} + \sum_{j\in\mathcal{X}}\beta_{c,j}k(x^i,x^j) \ge -\xi_{c,i} \qquad i\in\mathcal{X}_c \\ & \beta_{c,i} = -\frac{\nu_c}{m_{-c}} \qquad i\in\mathcal{X}_{-c} \\ & \xi_{c,i}\ge 0 \qquad i\in\mathcal{X}_c. \end{aligned} \tag{32}$$

Proof. Optimizing with respect to the worst-case realization across the uncertainty set (30) yields the following equivalent reformulation of model (31):

$$\begin{aligned} \min_{\tilde{w}_c,\tilde{\theta}_c,\xi_c} \quad & \frac{1}{2}\|\tilde{w}_c\|_{\mathcal{H}}^2 + \frac{\nu_c}{m_{-c}} \sum_{i\in\mathcal{X}_{-c}} \max_{\|\tilde{\delta}^i\|_{\mathcal{H}}\le\tilde{\varepsilon}_i} \left[\langle \tilde{w}_c, \varphi(x^i)+\tilde{\delta}^i\rangle_{\mathcal{H}} + \tilde{\theta}_c\right] + \frac{\alpha_c}{m_c}\sum_{i\in\mathcal{X}_c}\xi_{c,i} \\ \text{s.t.} \quad & \min_{\|\tilde{\delta}^i\|_{\mathcal{H}}\le\tilde{\varepsilon}_i} \left[\langle \tilde{w}_c, \varphi(x^i)+\tilde{\delta}^i\rangle_{\mathcal{H}} + \tilde{\theta}_c\right] \ge -\xi_{c,i} \qquad i\in\mathcal{X}_c \\ & \xi_{c,i}\ge 0 \qquad i\in\mathcal{X}_c. \end{aligned} \tag{33}$$

The maximization problem in the second term of the objective function in (33) can be rewritten as follows:

$$\max_{\|\tilde{\delta}^i\|_{\mathcal{H}}\le\tilde{\varepsilon}_i} \left[\langle \tilde{w}_c, \varphi(x^i)+\tilde{\delta}^i\rangle_{\mathcal{H}} + \tilde{\theta}_c\right] = \tilde{\theta}_c + \langle \tilde{w}_c, \varphi(x^i)\rangle_{\mathcal{H}} + \max_{\|\tilde{\delta}^i\|_{\mathcal{H}}\le\tilde{\varepsilon}_i}\left[\langle \tilde{w}_c, \tilde{\delta}^i\rangle_{\mathcal{H}}\right].$$

The same argument holds for the minimization problem in the first set of constraints. By applying the Cauchy–Schwarz inequality in $\mathcal{H}$ and the structure of the uncertainty set (30), we obtain:

$$\left|\langle \tilde{w}_c, \tilde{\delta}^i\rangle_{\mathcal{H}}\right| \le \|\tilde{w}_c\|_{\mathcal{H}}\|\tilde{\delta}^i\|_{\mathcal{H}} \le \|\tilde{w}_c\|_{\mathcal{H}}\tilde{\varepsilon}_i \qquad \forall\tilde{\delta}^i : \|\tilde{\delta}^i\|_{\mathcal{H}}\le\tilde{\varepsilon}_i.$$

Therefore, the optimal values of the maximization and minimization problems described above are respectively:

$$\max_{\|\tilde{\delta}^i\|_{\mathcal{H}}\le\tilde{\varepsilon}_i} \left[\langle \tilde{w}_c, \varphi(x^i)+\tilde{\delta}^i\rangle_{\mathcal{H}} + \tilde{\theta}_c\right] = \tilde{\theta}_c + \langle \tilde{w}_c, \varphi(x^i)\rangle_{\mathcal{H}} + \tilde{\varepsilon}_i\|\tilde{w}_c\|_{\mathcal{H}}$$

and

$$\min_{\|\tilde{\delta}^i\|_{\mathcal{H}}\le\tilde{\varepsilon}_i} \left[\langle \tilde{w}_c, \varphi(x^i)+\tilde{\delta}^i\rangle_{\mathcal{H}} + \tilde{\theta}_c\right] = \tilde{\theta}_c + \langle \tilde{w}_c, \varphi(x^i)\rangle_{\mathcal{H}} - \tilde{\varepsilon}_i\|\tilde{w}_c\|_{\mathcal{H}}.$$

Consequently, model (33) is equivalent to:

$$\begin{aligned} \min_{\tilde{w}_c,\tilde{\theta}_c,\xi_c} \quad & \frac{1}{2}\|\tilde{w}_c\|_{\mathcal{H}}^2 + \frac{\nu_c}{m_{-c}}\sum_{i\in\mathcal{X}_{-c}}\left[\tilde{\varepsilon}_i\|\tilde{w}_c\|_{\mathcal{H}} + \langle \tilde{w}_c, \varphi(x^i)\rangle_{\mathcal{H}}\right] + \nu_c\tilde{\theta}_c + \frac{\alpha_c}{m_c}\sum_{i\in\mathcal{X}_c}\xi_{c,i} \\ \text{s.t.} \quad & \tilde{\theta}_c - \tilde{\varepsilon}_i\|\tilde{w}_c\|_{\mathcal{H}} + \langle \tilde{w}_c, \varphi(x^i)\rangle_{\mathcal{H}} \ge -\xi_{c,i} \qquad i\in\mathcal{X}_c \\ & \xi_{c,i}\ge 0 \qquad i\in\mathcal{X}_c. \end{aligned} \tag{34}$$

In the feature space $\mathcal{H}$, vector $\tilde{w}_c$ can be decomposed as a linear combination of mapped input data by means of $\varphi(\cdot)$ (see [11]). Specifically, by considering Eq. (15), it holds that:

$$\tilde{w}_c = \sum_{i\in\mathcal{X}_c}\lambda_{c,i}\varphi(x^i) - \frac{\nu_c}{m_{-c}}\sum_{i\in\mathcal{X}_{-c}}\varphi(x^i) = \sum_{i\in\mathcal{X}}\beta_{c,i}\varphi(x^i),$$

where $\beta_c\in\mathbb{R}^m$ and $\beta_{c,i} = -\nu_c/m_{-c}$, for all $i\in\mathcal{X}_{-c}$. Therefore, the squared norm of $\tilde{w}_c$ in the objective function of (34) corresponds to:

$$\|\tilde{w}_c\|_{\mathcal{H}}^2 = \langle \tilde{w}_c, \tilde{w}_c\rangle_{\mathcal{H}} = \sum_{i,j\in\mathcal{X}}\beta_{c,i}k(x^i,x^j)\beta_{c,j} = \beta_c^\top K\beta_c, \tag{35}$$

where K is the Gram matrix of the entire training set, i.e., $K_{ij} = K_{ji} = k(x^i, x^j)$, with $i, j \in \mathcal{X}$. For all $i \in \mathcal{X}_c$, the dot product $\langle \tilde{w}_c, \varphi(x^i)\rangle_{\mathcal{H}}$ in the objective function and in the first set of constraints of (34) can be rewritten as:

$$\langle \tilde{w}_c, \varphi(x^i)\rangle_{\mathcal{H}} = \sum_{j\in\mathcal{X}} \beta_{c,j} k(x^i, x^j). \tag{36}$$

Thus, model (34) is equivalent to:

$$\begin{aligned}
\min_{\beta_c,\tilde{\theta}_c,\xi_c} \quad & \frac{1}{2}\beta_c^\top K\beta_c + \frac{\nu_c}{m_{-c}} \sum_{i\in\mathcal{X}_{-c}} \left[\tilde{\varepsilon}_i \sqrt{\beta_c^\top K\beta_c} + \sum_{j\in\mathcal{X}} \beta_{c,j} k(x^i, x^j)\right] + \nu_c\tilde{\theta}_c + \frac{\alpha_c}{m_c} \sum_{i\in\mathcal{X}_c} \xi_{c,i} \\
\text{s.t.} \quad & \tilde{\theta}_c - \tilde{\varepsilon}_i \sqrt{\beta_c^\top K\beta_c} + \sum_{j\in\mathcal{X}} \beta_{c,j} k(x^i, x^j) \geq -\xi_{c,i} \qquad i \in \mathcal{X}_c \\
& \beta_{c,i} = -\frac{\nu_c}{m_{-c}} \qquad i \in \mathcal{X}_{-c} \\
& \xi_{c,i} \geq 0 \qquad i \in \mathcal{X}_c.
\end{aligned}$$

This concludes the proof. □

Model (32) can be reformulated as the following SOCP model by introducing variables $t_c, u_c, v_c \in \mathbb{R}$:

$$\begin{aligned}
\min_{\beta_c,\tilde{\theta}_c,\xi_c,t_c,u_c,v_c} \quad & \frac{1}{2}(u_c - v_c) + \frac{\nu_c}{m_{-c}} \sum_{i\in\mathcal{X}_{-c}} \left[\tilde{\varepsilon}_i t_c + \sum_{j\in\mathcal{X}} \beta_{c,j} k(x^i, x^j)\right] + \nu_c\tilde{\theta}_c + \frac{\alpha_c}{m_c} \sum_{i\in\mathcal{X}_c} \xi_{c,i} \\
\text{s.t.} \quad & \tilde{\theta}_c - \tilde{\varepsilon}_i t_c + \sum_{j\in\mathcal{X}} \beta_{c,j} k(x^i, x^j) \geq -\xi_{c,i} \qquad i \in \mathcal{X}_c \\
& \beta_{c,i} = -\frac{\nu_c}{m_{-c}} \qquad i \in \mathcal{X}_{-c} \\
& t_c \geq \sqrt{\beta_c^\top K\beta_c} \\
& u_c + v_c = 1 \\
& u_c \geq \sqrt{t_c^2 + v_c^2} \\
& \xi_{c,i} \geq 0 \qquad i \in \mathcal{X}_c.
\end{aligned} \tag{37}$$

Once β_c and $\tilde{\theta}_c$ are determined as solutions of (37), the classification of a new observation $x \in \mathbb{R}^n$ is performed using the decision functions (16) and (17). In these functions, the $\mathcal{H}$-norm of $\tilde{w}_c$ is computed as in (35), while the dot product $\langle \tilde{w}_c, \varphi(x)\rangle_{\mathcal{H}}$ is calculated similarly to (36), i.e.:

$$\langle \tilde{w}_c, \varphi(x)\rangle_{\mathcal{H}} = \sum_{i\in\mathcal{X}} \beta_{c,i} \langle \varphi(x^i), \varphi(x)\rangle_{\mathcal{H}} = \sum_{i\in\mathcal{X}} \beta_{c,i} k(x^i, x).$$

Using the same complexity analysis outlined in Section 5.1 for robust SOCP models with linear classifiers, we conclude that a valid upper bound on the total computational complexity of problem (37) is $C \cdot \mathcal{O}\left((2\,m+4)^{7/2} \log(1/\eta)\right)$.

6. Experimental results

In this section, we evaluate the performance of the deterministic TPMSVM models presented in Section 4 and their robust counterparts derived in Section 5 on a selection of five public-domain multiclass datasets. All models were implemented in MATLAB (v. 2024 b) and solved using CVX (see [80,81]) with MOSEK as solver (v. 9.1.9, see [82]). All computational experiments were run on a MacBookPro17.1 with an Apple M1 chip of 8 cores and 16 GB of RAM. The codes were parallelized across the machine's cores using the MATLAB Parallel Computing Toolbox.

This section is organized as follows. In Section 6.1, we provide a description of the benchmark datasets and the experimental setting. The performance of the deterministic approaches is presented in Section 6.2, while the results for the robust models are discussed in Section 6.3.

6.1. Datasets and experimental setting

We considered five public-domain real-world multiclass datasets from the *UCI Machine Learning* repository (UCI, see [83]) and from *Open Data Canada* (ODC, see [84]). Relevant details of the datasets are summarized in Table 3.

To assess the performance of the proposed methodology, each dataset was randomly divided into training and testing sets, with a proportion of 75 %–25 % of the total number of observations. The partition follows the *proportional random sampling* strategy (see [85]), thereby preserving the original class distribution both in the training and in the testing set. To avoid imbalances among the orders of magnitude of the features, each dataset was linearly scaled into the n-dimensional hypercube $[0, 1]^n$. For kernel-based classifiers, we considered five polynomial kernels (homogeneous quadratic and cubic; inhomogeneous linear, quadratic and cubic) as well as the Gaussian kernel (see Table 2).

Table 3
Summary statistics of considered datasets.

Dataset	Source	Subject area	Observations (m)	Features (n)	Classes (C)
Iris	UCI	Biology	150	4	3
Wine	UCI	Physics and chemistry	178	13	3
Glass identification	UCI	Physics and chemistry	214	9	6
Fuel consumption ratings	ODC	Transport	374	7	3
Car evaluation	UCI	Other	1728	6	4

Table 4
Detailed percentage results of average accuracy and standard deviation over 50 runs of the deterministic models (11) and (14). Classification is performed according to the argmin and argmax decision functions, respectively in (a) and (b). The best result for the kernelized model is underlined. Overall, the best result is in bold.

Dataset		Linear	Kernel					
			Hom. quadratic	Hom. cubic	Inhom. linear	Inhom. quadratic	Inhom. cubic	Gaussian
(a) Deterministic models – argmin decision functions (12) and (16).								
Iris	Accuracy	**92.81 ± 4.29**	78.54 ± 7.67	85.68 ± 5.58	92.38 ± 3.89	91.35 ± 5.09	88.00 ± 5.46	87.78 ± 5.78
	CPU time (s)	1.61	1.19	1.22	10.61	10.20	10.32	10.63
Wine	Accuracy	97.00 ± 2.08	96.59 ± 2.98	95.23 ± 3.22	96.64 ± 2.57	**97.41 ± 2.43**	95.91 ± 3.47	39.45 ± 1.10
	CPU time (s)	1.50	1.22	1.11	10.04	9.98	10.00	10.20
Glass	Accuracy	38.49 ± 7.17	41.81 ± 5.75	44.26 ± 5.22	36.94 ± 6.97	36.34 ± 6.95	44.38 ± 6.39	**61.17 ± 5.23**
	CPU time (s)	3.07	2.35	2.25	19.77	19.61	19.87	20.51
Fuel	Accuracy	52.73 ± 5.35	50.47 ± 4.29	51.68 ± 5.78	55.66 ± 4.67	52.62 ± 5.31	53.42 ± 5.48	**70.02 ± 4.84**
	CPU time (s)	1.68	1.24	1.14	9.67	9.92	10.84	11.17
Car	Accuracy	73.15 ± 2.00	77.71 ± 2.15	79.07 ± 1.88	76.62 ± 1.99	78.14 ± 1.85	**79.12 ± 2.20**	69.96 ± 0.10
	CPU time (s)	2.06	2.17	3.05	20.42	20.23	28.85	46.70
(b) Deterministic models – argmax decision functions (13) and (17).								
Dataset		Linear	Kernel					
			Hom. quadratic	Hom. cubic	Inhom. linear	Inhom. quadratic	Inhom. cubic	Gaussian
Iris	Accuracy	70.22 ± 2.47	94.59 ± 4.01	92.65 ± 6.23	76.92 ± 5.60	**95.30 ± 3.53**	94.49 ± 3.54	87.62 ± 4.64
	CPU time (s)	1.59	1.15	1.15	10.07	10.15	10.29	10.56
Wine	Accuracy	**96.55 ± 3.44**	96.36 ± 3.15	95.55 ± 3.47	96.41 ± 2.39	95.95 ± 2.53	95.82 ± 2.36	39.45 ± 1.10
	CPU time (s)	1.57	1.13	1.13	10.14	10.28	10.26	10.53
Glass	Accuracy	46.42 ± 6.40	45.13 ± 7.11	47.28 ± 6.99	41.17 ± 5.29	45.74 ± 7.09	49.02 ± 6.65	**61.58 ± 5.21**
	CPU time (s)	3.05	2.29	2.26	19.69	20.16	19.51	20.16
Fuel	Accuracy	54.92 ± 5.90	56.86 ± 4.90	56.65 ± 6.08	57.57 ± 5.20	57.72 ± 5.27	58.37 ± 4.23	**66.60 ± 4.53**
	CPU time (s)	1.69	1.23	1.13	10.03	10.18	10.81	11.28
Car	Accuracy	75.49 ± 1.64	81.50 ± 1.71	81.82 ± 1.38	77.23 ± 1.44	82.25 ± 1.52	**84.07 ± 1.92**	69.96 ± 0.10
	CPU time (s)	2.06	2.20	3.13	19.16	19.58	28.78	47.11

As far as it concerns hyperparameters, for simplicity we set $\nu_c = \nu$ and $\alpha_c = \alpha$ for all $c = 1, \dots, C$. A grid-search procedure was then conducted to tune their values. Specifically, α was selected from the set $\{2^j | j = -6, -5, \dots, 5, 6\}$, whereas the value of ν/α was chosen from the set $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. Additionally, the parameters γ for the inhomogeneous polynomial kernels and σ for the Gaussian kernel were selected from $\{2^j | j = -4, -3, \dots, 3, 4\}$.

The best configuration of parameters was identified as the one maximizing the accuracy on the training set. The performance of the model was then evaluated on the testing set and the corresponding accuracy was computed. To ensure stability of the results, for each hold-out 75 %–25 % the computational experiments were performed over 50 random combinations of training and testing sets, and the results were finally averaged.

6.2. Results for the deterministic multiclass TPMSVM models

Table 4a and b report the results of deterministic models (11) and (14) in terms of percentage average accuracy and standard deviation. The CPU time for training each model is outlined below the results.

First of all, by comparing the third column of Table 4a and b with the remaining ones, we observe that the best kernel-induced classifier consistently outperforms the separating hyperplane obtained from model (11) across the majority of the tasks. Indeed, nonlinear classifiers provide better accuracy regardless of the decision function used, showing the strength of kernelized approaches. Even in the few cases where the linear classifier achieves slightly better accuracy (see the *Iris* dataset with the argmin decision function and the *Wine* dataset with the argmax decision function), the accuracy differences between the results are negligible (0.43 % and 0.14 %, respectively).

Secondly, the choice of the decision function plays an important role in the performance of the models, particularly for the *Iris* dataset. In the linear case, the accuracy drops from 92.81 % (see Table 4a) using the argmin decision function to 70.22 % (see Table 4b) with the argmax decision function. This highlights that the argmin decision function (12) is better suited for solving linear classification

Table 5
Detailed percentage results of average accuracy and standard deviation over 50 runs of the robust model (20) with linear classifier. Classification is performed according to the argmin and argmax decision functions, respectively in (a) and (b). The best result for each ℓ_p-norm is underlined. Overall, the best result is in bold.

Dataset	ℓ_p-norm	$p=1$			$p=2$			$p=\infty$		
	ε	10^{-3}	10^{-2}	10^{-1}	10^{-3}	10^{-2}	10^{-1}	10^{-3}	10^{-2}	10^{-1}
(a) Robust model – linear classifier – argmin decision function (12).										
Iris	Accuracy	92.70 ± 3.79	93.89 ± 3.93	**95.35 ± 3.32**	93.19 ± 3.46	93.14 ± 3.38	92.49 ± 4.24	91.95 ± 3.72	92.38 ± 4.29	78.59 ± 5.93
	CPU time (s)	1.54	1.46	1.45	1.41	1.40	1.40	1.46	1.46	1.45
Wine	Accuracy	97.36 ± 2.52	**97.59 ± 1.92**	96.64 ± 2.84	97.14 ± 2.19	97.23 ± 2.12	96.59 ± 2.12	96.64 ± 2.40	96.91 ± 2.09	93.86 ± 3.16
	CPU time (s)	1.44	1.46	1.46	1.42	1.42	1.46	1.48	1.46	1.45
Glass	Accuracy	39.32 ± 6.06	38.15 ± 6.62	34.00 ± 7.14	38.64 ± 7.32	37.85 ± 6.92	35.25 ± 6.82	37.92 ± 6.68	37.96 ± 7.18	**39.92 ± 6.06**
	CPU time (s)	3.04	2.91	2.85	2.92	2.83	2.85	2.96	2.93	2.97
Fuel	Accuracy	51.66 ± 5.30	52.65 ± 4.95	52.90 ± 4.62	51.48 ± 4.48	**55.40 ± 5.59**	53.94 ± 4.55	51.48 ± 3.77	53.05 ± 4.08	51.01 ± 3.71
	CPU time (s)	1.51	1.44	1.47	1.41	1.41	1.39	1.47	1.44	1.44
Car	Accuracy	71.99 ± 1.96	72.00 ± 2.19	71.73 ± 1.97	**72.55 ± 1.88**	72.50 ± 2.18	70.25 ± 2.16	72.12 ± 2.00	70.67 ± 2.40	58.74 ± 2.48
	CPU time (s)	1.97	1.96	2.01	1.94	1.94	1.94	2.02	2.01	2.02
(b) Robust model – linear classifier – argmax decision function (13).										
Dataset	ℓ_p-norm	$p=1$			$p=2$			$p=\infty$		
	ε	10^{-3}	10^{-2}	10^{-1}	10^{-3}	10^{-2}	10^{-1}	10^{-3}	10^{-2}	10^{-1}
Iris	Accuracy	69.57 ± 2.88	70.11 ± 3.01	69.24 ± 2.72	70.65 ± 2.94	70.38 ± 3.78	69.78 ± 2.98	70.54 ± 3.24	69.57 ± 2.83	**84.38 ± 4.83**
	CPU time (s)	1.57	1.47	1.52	1.43	1.41	1.41	1.48	1.47	1.49
Wine	Accuracy	97.00 ± 2.27	96.82 ± 2.05	96.68 ± 2.80	97.09 ± 2.06	96.36 ± 2.30	**97.14 ± 2.29**	96.32 ± 2.29	97.00 ± 2.45	96.27 ± 2.55
	CPU time (s)	1.48	1.54	1.46	1.42	1.42	1.40	1.50	1.49	1.51
Glass	Accuracy	45.13 ± 6.13	**47.85 ± 7.05**	45.47 ± 5.88	46.83 ± 7.39	45.89 ± 7.65	44.75 ± 5.30	46.53 ± 7.78	47.09 ± 6.22	42.87 ± 4.36
	CPU time (s)	2.92	2.89	2.95	2.82	2.84	2.82	2.93	2.95	2.94
Fuel	Accuracy	56.34 ± 6.04	53.57 ± 4.15	**60.65 ± 3.48**	55.46 ± 6.05	57.53 ± 4.13	60.26 ± 3.22	54.30 ± 5.04	57.12 ± 5.58	54.00 ± 5.14
	CPU time (s)	1.49	1.46	1.43	1.39	1.39	1.40	1.43	1.43	1.44
Car	Accuracy	75.11 ± 1.79	75.53 ± 1.53	76.22 ± 1.33	75.71 ± 1.35	75.81 ± 1.36	77.72 ± 1.75	75.48 ± 1.74	75.66 ± 1.82	**79.42 ± 1.45**
	CPU time (s)	2.04	2.05	2.04	1.94	1.97	1.97	2.05	2.07	2.05

tasks within this dataset. Conversely, in the nonlinear setting, the argmax decision function (17) yields the best results for certain kernels: accuracy improves from 78.54 % to 94.59 % with the homogeneous quadratic kernel (+16.05 %), and from 85.68 % to 92.65 % with the homogeneous cubic kernel (+6.97 %). For the *Wine* dataset, both decision functions lead to comparable performance across all kernels, suggesting flexibility in choice. Finally, for the remaining datasets (*Glass*, *Fuel*, and *Car*) the argmax decision function generally delivers superior accuracy.

In terms of computational efficiency, kernel-induced models, and particularly the ones with the Gaussian kernel, require significantly more CPU time compared to linear classifiers. Therefore, we can conclude that there exists a trade-off between accuracy and performance speed. The final user must balance these factors based on their priorities: faster results with good accuracy (linear classifiers) or longer runtimes with better predictive performance (nonlinear classifiers).

6.3. Results for the robust multiclass TPMSVM models

To evaluate the performance of the robust approaches, we assume a constant radius for the uncertainty set (18) across all observations, i.e., $\varepsilon_i = \varepsilon$ for all $i \in \mathcal{X}$. We consider three increasing levels of perturbation ($\varepsilon = 10^{-3}, 10^{-2}, 10^{-1}$) and three different ℓ_p-norms ($p = 1, 2, \infty$).

The results of the numerical experiments for the robust linear classification tasks are presented in Table 5a and b.

A comparison between these results and those in the third column of Table 4a and b shows that in nine out of ten cases the robust model outperforms the corresponding deterministic formulation. This is particularly evident when using the argmax decision function (13). For instance, considering the *Iris* dataset with $p = \infty$ and $\varepsilon = 10^{-1}$, the robust linear model achieves an accuracy of 84.38 % (see Table 5b), compared to 70.22 % for the deterministic counterpart (see Table 4b). A significant improvement is observed for the *Car* dataset too (79.42 % vs 75.49 %).

Regarding the choice of the ℓ_p-norm used to define the uncertainty set, the results show that no single norm consistently yields the best performance across all tasks. Specifically, the ℓ_1-norm leads to the highest accuracy in four cases, while the ℓ_2- and ℓ_∞-norms are the best in three cases each. This variability suggests that the effectiveness of a particular norm is problem-dependent and may be influenced by the structure of the data.

In addition to the choice of norm, the perturbation level ε also plays a crucial role in model performance. As ε increases from 10^{-3} to 10^{-1}, accuracy varies depending on the dataset and the choice of ℓ_p-norm. The general trend indicates that increasing the perturbation level can be beneficial for the model. In fact, in only one out of ten cases the best accuracy is achieved at the lowest perturbation level ($\varepsilon = 10^{-3}$, see the *Car* dataset in Table 5a), while in six out of ten cases the best results are obtained with the highest level of perturbation ($\varepsilon = 10^{-1}$). This suggests that introducing a controlled level of uncertainty can improve predictive performance. Particularly, a significant improvement is observed for the *Iris* dataset with the argmax decision function and ℓ_∞-norm (see Table 5b),

Table 6
Detailed percentage results of average accuracy and standard deviation over 50 runs of the robust model (37) with nonlinear classifier. Classification is performed according to the argmin and argmax decision functions, respectively in (a) and (b). For each dataset, the kernel in the second column is chosen according to the corresponding best deterministic result of Table 4a and b. The best result for each ℓ_p-norm is underlined. Overall, the best result is in bold.

Dataset	Kernel	ℓ_p-norm	$p=1$			$p=2$			$p=\infty$		
		ε	10^{-3}	10^{-2}	10^{-1}	10^{-3}	10^{-2}	10^{-1}	10^{-3}	10^{-2}	10^{-1}
(a) Robust model – nonlinear classifier – argmin decision function (16).											
Iris	Inhom. linear	Accuracy	92.97 ± 3.98	92.86 ± 4.15	90.32 ± 6.93	**93.30 ± 3.79**	92.27 ± 3.54	89.24 ± 7.96	91.46 ± 3.76	92.76 ± 3.64	86.76 ± 10.98
		CPU time (s)	13.09	12.70	13.05	12.90	13.51	13.08	13.09	13.04	12.83
Wine	Inhom. quadratic	Accuracy	96.14 ± 2.61	96.86 ± 2.94	96.05 ± 2.75	96.00 ± 2.89	**96.91 ± 2.86**	96.45 ± 2.39	96.45 ± 2.64	96.64 ± 2.84	7.09 ± 3.14
		CPU time (s)	101.51	101.51	99.93	100.82	97.19	99.03	101.68	106.09	99.37
Glass	Gaussian	Accuracy	**39.86 ± 8.22**	37.26 ± 3.46	36.79 ± 4.16	36.56 ± 3.34	37.26 ± 3.46	36.79 ± 4.16	37.26 ± 3.87	37.74 ± 3.51	37.74 ± 3.34
		CPU time (s)	2598.09	2810.71	2869.73	2772.89	2893.72	2871.30	2701.93	2717.65	2877.96
Fuel	Gaussian	Accuracy	**62.99 ± 5.69**	62.75 ± 5.96	40.26 ± 2.63	60.08 ± 3.87	60.35 ± 2.90	39.52 ± 1.70	61.56 ± 2.68	59.01 ± 3.65	37.63 ± 0.81
		CPU time (s)	5341.74	5163.59	4985.68	4885.93	5186.56	4843.79	5083.51	5019.50	5197.60
Car	Inhom. cubic	Accuracy	76.24 ± 1.95	**76.59 ± 2.18**	67.29 ± 2.20	75.72 ± 2.27	75.33 ± 2.30	60.45 ± 2.60	75.74 ± 2.01	74.61 ± 2.26	42.17 ± 5.02
		CPU time (s)	2978.02	2979.47	2984.12	2954.32	2902.26	2909.32	3018.96	3033.63	2950.35
(b) Robust model – nonlinear classifier – argmax decision function (17).											
Dataset	Kernel	ℓ_p-norm	$p=1$			$p=2$			$p=\infty$		
		ε	10^{-3}	10^{-2}	10^{-1}	10^{-3}	10^{-2}	10^{-1}	10^{-3}	10^{-2}	10^{-1}
Iris	Inhom. quadratic	Accuracy	93.78 ± 3.75	95.46 ± 3.84	93.51 ± 3.98	94.27 ± 3.81	94.97 ± 3.86	95.14 ± 3.94	94.43 ± 3.47	**95.46 ± 3.34**	94.32 ± 3.83
		CPU time (s)	13.27	13.39	13.38	13.24	13.29	13.42	13.38	13.36	13.38
Wine	Inhom. linear	Accuracy	**97.23 ± 2.40**	96.23 ± 2.85	96.82 ± 2.05	96.68 ± 2.44	95.91 ± 2.60	96.50 ± 2.48	96.32 ± 2.24	96.50 ± 2.52	96.82 ± 2.60
		CPU time (s)	102.90	102.28	98.78	101.24	98.27	96.16	100.98	104.46	99.43
Glass	Gaussian	Accuracy	**38.92 ± 4.39**	35.14 ± 6.29	35.85 ± 5.43	37.74 ± 3.77	35.38 ± 3.74	30.42 ± 8.64	36.32 ± 5.02	35.14 ± 4.39	29.48 ± 9.02
		CPU time (s)	2583.94	2803.87	2856.64	2753.53	2884.25	2869.89	2697.41	2705.17	2855.15
Fuel	Gaussian	Accuracy	56.82 ± 6.64	55.10 ± 5.26	49.03 ± 5.11	**57.80 ± 5.10**	55.91 ± 4.60	52.15 ± 5.60	54.44 ± 6.99	54.03 ± 4.48	53.23 ± 6.43
		CPU time (s)	5161.05	5102.44	4914.07	4832.09	4981.42	4875.20	5015.18	4988.35	5209.15
Car	Inhom. cubic	Accuracy	85.10 ± 0.97	86.26 ± 1.25	87.24 ± 1.16	85.24 ± 1.85	85.58 ± 1.69	**87.56 ± 2.02**	84.98 ± 1.78	86.21 ± 1.57	82.29 ± 0.98
		CPU time (s)	3060.21	3033.12	2984.90	2930.46	2941.68	2875.33	3026.86	3035.45	2981.57

Table 7
Percentage average accuracy comparison between deterministic and robust results obtained from the SVM formulations proposed in this work and in [32]. For each formulation, the best result is underlined. Overall, the best result is in bold.

Dataset	Deterministic formulations				Robust formulations			
	Linear classifiers		Nonlinear classifiers		Linear classifiers		Nonlinear classifiers	
	Table 4a and b	[32]	Table 4a and b	[32]	Table 5a and b	[32]	Table 6a and b	[32]
Iris	92.81	88.92	95.30	94.59	95.35	78.05	**95.46**	94.76
Wine	97.00	96.00	97.41	97.14	**97.59**	96.32	97.23	96.09
Glass	46.42	46.98	61.58	65.13	47.85	34.75	39.86	**65.96**
Fuel	54.92	56.62	**70.02**	58.30	60.65	69.16	62.99	57.53
Car	75.49	75.42	84.07	**96.00**	79.42	69.99	87.56	69.02

where increasing ϵ leads to a substantial gain in accuracy: from 70.54 % at $\epsilon = 10^{-3}$ to 84.38 % at $\epsilon = 10^{-1}$. On the other hand, overly conservative settings may deteriorate performance in some specific cases, as observed for the *Car* dataset (see Table 5a), where too large perturbations lead to a decline in accuracy. All these findings highlight the importance of carefully selecting the perturbation level and an appropriate norm to balance robustness and predictive performance, potentially validated through empirical tuning.

In the case of the robust multiclass model (37) with nonlinear classifiers, we selected as kernel function the one that achieved the highest accuracy in the deterministic setting (see Table 4a and b). The results of the numerical experiments are reported in Table 6a and b.

The robust model generally yields an improved accuracy over the deterministic setting, particularly when using the argmax decision function (see Table 6b). For example, in the *Wine* dataset, the robust model with an inhomogeneous linear kernel achieves an accuracy of 97.23 %, compared to 96.41 % in the deterministic case. In general, the robust approach provides an accuracy that is either superior or comparable to the deterministic counterpart, especially when polynomial kernels are used, as observed in the *Iris*, *Wine*, and *Car* datasets. Conversely, when using the Gaussian kernel (see the *Glass* and *Fuel* datasets), the performance tends to decline in the robust setting. This reduction is mainly due to the Gaussian kernel's sensitivity to data perturbations, which can negatively impact the decision boundary under strong uncertainty. It is worth noting that the best performances in the robust context are most frequently obtained using the ℓ_1-norm and a small perturbation level ($\epsilon = 10^{-3}$), suggesting a clear preference for these settings across the considered datasets.

In terms of CPU time, the robust approach with linear classifiers exhibits comparable computational times to the deterministic case. In contrast, when using kernel-based classifiers, the CPU time increases significantly, particularly for the Gaussian kernel (see the *Glass* and *Wine* datasets) and for datasets with a large number of observations (see the *Car* dataset). This increase in computational time for larger datasets is consistent with the complexity analysis discussed at the end of Section 5.2.

Finally, in Table 7, we compare the performance of our proposed method with the deterministic and robust SVM formulations presented in [32], considering both linear and nonlinear classifiers. We observe that in three out of five cases (see the *Iris*, *Wine*, and *Glass* datasets) the robust formulations outperform their deterministic counterparts, emphasizing the benefit of incorporating uncertainty in SVM models. Furthermore, in three out of five datasets (see the *Iris*, *Wine*, and *Fuel* datasets), our approach achieves the highest accuracy overall, demonstrating its competitiveness with state-of-the-art robust SVM alternatives.

7. Conclusions

In this paper, we have introduced novel Twin Parametric Margin Support Vector Machine (TPMSVM) models to address multiclass classification tasks under data uncertainty. From a methodological perspective, we have extended the TPMSVM techniques proposed in [21] to handle multiclass settings, employing both linear and kernel-induced decision boundaries. As final decision functions, we have proposed two alternatives based on the argmin and argmax formulations.

Given the uncertain nature of real-world data, we have adopted a Robust Optimization approach by constructing a bounded-by-ℓ_p-norm uncertainty set around each input data. This strategy aims to protect the deterministic models against uncertainties and prevent the worsening of the classification's performance. Then, we have derived robust counterparts of the deterministic models, and provided tractable reformulations in the form of Second Order Cone Programming models.

To validate the effectiveness of our proposal, we have evaluated the multiclass models on real-world datasets, considering different combinations of kernels and decision functions, while exploring various ℓ_p-norms and levels of perturbations within the robust setting. The results of our experimental analysis show that robust solutions consistently yield higher accuracy compared to their deterministic counterparts, particularly when dealing with linear and polynomial kernels, but at the cost of increased computational time.

Regarding future research, various streams can originate from this work. First of all, extending the robust TPMSVM framework to account for uncertainties in the labels of input data could enhance the model's generalization capabilities. Secondly, different optimization under uncertainty methodologies could be explored to further robustify the TPMSVM approach. In particular, Chance-Constrained Programming and Distributionally Robust Optimization could represent effective strategies for enhancing robustness in multiclass classification under uncertainty. Thirdly, improving the interpretability and explainability of robust TPMSVM models, particularly when applied in sensitive domains such as medicine, merits additional investigation in future works. As robustness can make the models more complex and opaque, developing tools to analyze feature relevance under uncertainty and trace the effect of perturbations could offer valuable insights to the final users. Finally, from a computational perspective, two extensions are worth

exploring. On the one hand, adopting alternative hyperparameter tuning techniques, such as Bayesian optimization, could lead to improved performance and computational efficiency. On the other hand, when working with large datasets, splitting the data into three subsets (training, tuning, and testing sets) rather than only two (training and testing sets), as suggested in [86], could help mitigate overfitting and provide a more reliable assessment of model performance.

CRediT authorship contribution statement

Renato De Leone: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Resources, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Francesca Maggioni:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Andrea Spinelli:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. The author Francesca Maggioni is an Associate Editor for EURO Journal on Computational Optimization and was not involved in the editorial review or the decision to publish this article.

Acknowledgements

This work has been supported by “ULTRA OPTYMAL – Urban Logistics and sustainable TRAnsportation: OPtimization under uncertainTY and MAchine Learning”, a PRIN2020 project funded by the Italian University and Research Ministry (grant number 20207C8T9M).

This study was also carried out within the MOST – Sustainable Mobility National Research Center and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.4 – D.D. 1033 17/06/2022, CN00000023), Spoke 5 “Light Vehicle and Active Mobility” and by the PNRR MUR project ECS_00000041-VITALITY-CUP J13C22000430001, Spoke 6. This manuscript reflects only the authors’ views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

[1] T. Dias, P. Amaral, A classification method based on a cloud of spheres, Eur. J. Comput. Optim. 11 (2023) 100077.
[2] B.R. Gunnarsson, S. Vanden Broucke, B. Baesens, M. Óskarsdóttir, W. Lemahieu, Deep learning for credit scoring: do or don’t?, Eur. J. Oper. Res. 295 (1) (2021) 292–305.
[3] S. Cipolla, J. Gondzio, Training very large scale nonlinear svms using alternating direction method of multipliers coupled with the hierarchically semi-separable kernel approximations, Eur. J. Comput. Optim. 10 (2022) 100046.
[4] S. Maldonado, J. López, M. Carrasco, The Cobb-Douglas learning machine, Pattern Recognit. 128 (2022) 108701.
[5] A. Pedro Duarte Silva, Optimization approaches to supervised classification, Eur. J. Oper. Res. 261 (2) (2017) 772–788.
[6] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugen. 7 (2) (1936) 179–188.
[7] V.N. Vapnik, A.Y. Chervonenkis, Theory of Pattern Recognition, Nauka, Moscow, 1974.
[8] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, 1995.
[9] W.E.I. Jiang, S. Siddiqui, Hyper-parameter optimization for support vector machines using stochastic gradient descent and dual coordinate descent, Eur. J. Comput. Optim. 8 (1) (2020) 85–101.
[10] E. Carrizosa, A. Nogales-Gómez, D. Romero Morales, Clustering categories in support vector machines, Omega 66 (2017) 28–37.
[11] C. Cortes, V.N. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.
[12] Jayadeva, R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, IEEE Trans. Pattern Anal. Mach. Intell. 29 (5) (2007) 905–910.
[13] X. Liu, F.A. Potra, Pattern separation and prediction via linear and semidefinite programming, Stud. Inf. Control 18 (1) (2009) 71–82.
[14] E. Marcelli, R. De Leone, Multi-kernel covariance terms in multi-output support vector machines, in: G. Nicosia, V. Ojha, E. La Malfa, G. Jansen, V. Sciacca, P. Pardalos, G. Giuffrida, R. Umeton (Eds.), Machine Learning, Optimization, and Data Science, Springer International Publishing, Cham, 2020, pp. 1–11.
[15] B. Schölkopf, A. Smola, R.C. Williamson, P.L. Bartlett, New support vector algorithms, Neural Comput. 12 (5) (2000) 1207–1245.
[16] R. De Leone, Support vector regression for time series analysis, in: Operations Research, Springer, 2010.
[17] D. Islip, R.H. Kwon, S. Kim, Integration of support vector machines and mean-variance optimization for capital allocation, Eur. J. Oper. Res. 322 (3) (2025) 1045–1058.
[18] D. Golmohammadi, L. Zhao, D. Dreyfus, Using machine learning techniques to reduce uncertainty for outpatient appointment scheduling practices in outpatient clinics, Omega 120 (2023) 102907.
[19] S. Maldonado, G. Domínguez, D. Olaya, W. Verbeke, Profit-driven churn prediction for the mutual fund industry: a multisegment approach, Omega 100 (2021) 102380.
[20] S. Maldonado, J. López, C. Vairetti, Profit-based churn prediction based on minimax probability machines, Eur. J. Oper. Res. 284 (1) (2020) 273–284.
[21] X. Peng, Tpmsvm: a novel twin parametric-margin support vector machine for pattern recognition, Pattern Recognit. 44 (10) (2011) 2678–2692.
[22] R. De Leone, F. Maggioni, A. Spinelli, A multiclass robust twin parametric margin support vector machine with an application to vehicles emissions, in: G. Nicosia, V. Ojha, E. La Malfa, G. La Malfa, P.M. Pardalos, R. Umeton (Eds.), Machine Learning, Optimization, and Data Science, Vol. 14506 of Lecture Notes in Computer Science, Springer Nature Switzerland, Cham, 2024, pp. 299–310, https://doi.org/10.1007/978-3-031-53966-4_22.
[23] S. Maqsood, R. Damaševičius, Multiclass skin lesion localization and classification using deep learning based features fusion and selection framework for smart healthcare, Neural Netw. 160 (2023) 238–258.
[24] J. Weston, C. Watkins, Support vector machines for multi-class pattern recognition, in: M. Verleysen (Ed.), Proceedings of the Seventh European Symposium On Artificial Neural Networks (ESANN), 1999, pp. 219–224.
[25] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Netw. 13 (2) (2002) 415–425.
[26] S. Ding, X. Zhao, J. Zhang, X. Zhang, Y. Xue, A review on multi-class twsvm, Artif. Intell. Rev. 52 (2019) 775–801.
[27] Y. Peng, G. Kou, G. Wang, Y. Shi, FAMCDM: a fusion approach of MCDM methods to rank multiclass classification algorithms, Omega 39 (6) (2011) 677–689.
[28] A. Ben-Tal, L. El Ghaoui, A. Nemirovski, Robust Optimization, Princeton University Press, 2009.
[29] D. Bertsimas, J. Dunn, C. Pawlowski, Y.D. Zhuo, Robust classification, INFORMS J. Optim. 1 (2019) 2–34.

[30] T.B. Trafalis, R.C. Gilbert, Robust classification and regression using support vector machines, Eur. J. Oper. Res. 173 (2006) 893–909.
[31] D. Faccini, F. Maggioni, F.A. Potra, Robust and distributionally robust optimization models for linear support vector machine, Comput. Oper. Res. 147 (2022) 105930.
[32] F. Maggioni, A. Spinelli, A novel robust optimization model for nonlinear support vector machine, Eur. J. Oper. Res. 322 (1) (2025) 237–253, https://doi.org/10.1016/j.ejor.2024.12.014.
[33] B.E. Boser, I. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, Proc. Fifth Annu. Workshop Comput. Learn. Theory 5 (1992) 144–152.
[34] K. Crammer, Y. Singer, On the algorithmic implementation of multiclass kernel-based vector machines, J. Mach. Learn. Res. 2 (2001) 265–292.
[35] E.J. Bredensteiner, K.P. Bennett, Multicategory classification by support vector machines, Comput. Optim. Appl. 12 (1999) 53–79.
[36] Y. Yajima, Linear programming approaches for multicategory support vector machines, Eur. J. Oper. Res. 162 (2) (2005) 514–531.
[37] R.M. Rifkin, A. Klautau, In defense of one-vs-all classification, J. Mach. Learn. Res. 5 (2004) 101–141.
[38] Z. Li, S. Tang, S. Yan, Multi-class svm classifier based on pairwise coupling, in: S.-W. Lee, A. Verri (Eds.), Pattern Recognition with Support Vector Machines, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 321–333.
[39] C. Angulo, X. Parra, A. Català, K-svcr. A support vector machine for multi-class classification, Neurocomputing 55 (2003) 57–77.
[40] S.-W. Du, M.-C. Zhang, P. Chen, H.-F. Sun, W.-J. Chen, Y.-H. Shao, A multiclass nonparallel parametric-margin support vector machine, Information 12 (12) (2021) 515–533.
[41] Ü. Doğan, T. Glasmachers, C. Igel, A unified view on multi-class support vector classification, J. Mach. Learn. Res. 17 (2016) 1–32.
[42] H. Xu, C. Caramanis, S. Mannor, Robustness and regularization of support vector machines, J. Mach. Learn. Res. 10 (2009) 1485–1510.
[43] H. Rahimian, S. Mehrotra, Frameworks and results in distributionally robust optimization, Open J. Math. Optim. 3 (2022) 1–85.
[44] A. Shapiro, D. Dentcheva, A. Ruszczynski, Lectures on Stochastic Programming - Modeling and Theory, MOS-SIAM Series on Optimization, 2009.
[45] F. Maggioni, D. Faccini, F. Gheza, F. Manelli, G. Bonetti, Machine learning based classification models for COVID-19 patients, in: R. Aringhieri, F. Maggioni, E. Lanzarone, M. Reuter-Oppermann, G. Righini, M.T. Vespucci (Eds.), Operations Research for Health Care in Red Zone, Springer International Publishing, Cham, 2023, pp. 35–46.
[46] F. Maggioni, A. Spinelli, A robust nonlinear support vector machine approach for vehicles smog rating classification, in: M. Bruglieri, P. Festa, G. Macrina, O. Pisacane (Eds.), Optimization in Green Sustainability and Ecological Transition, AIRO Springer Series, Springer, Cham, 2024, https://doi.org/10.1007/978-3-031-47686-0_19.
[47] M. Piazza, A. Spinelli, F. Maggioni, M. Bedoni, E. Messina, A robust support vector machine approach for raman data classification, Decis. Anal. J. (2025) 100595, https://doi.org/10.1016/j.dajour.2025.100595.
[48] F. Hooshmand, F. Seilsepour, S.A. MirHassani, Adjustable robust optimization approach for svm under uncertainty, Omega 131 (2025) 103206.
[49] P. Zhong, M. Fukushima, Second-order cone programming formulations for robust multiclass classification, Neural Comput. 19 (2007) 258–282.
[50] R. Khanjani-Shiraz, A. Babapour-Azar, Z. Hosseini-Nodeh, P.M. Pardalos, Distributionally robust joint chance-constrained support vector machines, Optim. Lett. 17 (2023) 299–332.
[51] F. Lin, S.-C. Fang, X. Fang, Z. Gao, Distributionally robust chance-constrained kernel-based support vector machine, Comput. Oper. Res. 170 (2024) 106755.
[52] F. Lin, S.-C. Fang, X. Fang, Z. Gao, J. Luo, A distributionally robust chance-constrained kernel-free quadratic surface support vector machine, Eur. J. Oper. Res. 316 (1) (2024) 46–60.
[53] P.-Y. Hao, New support vector algorithms with parametric insensitive/margin model, Neural Netw. 23 (1) (2010) 60–73.
[54] M. Arun Kumar, M. Gopal, Least squares twin support vector machines for pattern classification, Expert Syst. Appl. 36 (4) (2009) 7535–7543.
[55] X. Chen, J. Yang, Q. Ye, J. Liang, Recursive projection twin support vector machine via within-class variance minimization, Pattern Recognit. 44 (10) (2011) 2643–2655.
[56] Y. Xu, Z. Yang, X. Pan, A novel twin support-vector machine with pinball loss, IEEE Trans. Neural Netw. Learn. Syst. 28 (2) (2017) 359–370.
[57] S.-G. Chen, X. Wu, A new fuzzy twin support vector machine for pattern classification, Int. J. Mach. Learn. Cybern. 9 (9) (2018) 1553–1564.
[58] M. Tanveer, T. Rajani, R. Rastogi, Y. Shao, Comprehensive review on twin support vector machines, Ann. Oper. Res. 310 (2) (2022) 1–46.
[59] X. Peng, Y. Wang, D. Xu, Structural twin parametric-margin support vector machine for binary classification, Knowl.-Based Syst. 49 (2013) 63–72.
[60] Y. Shao, Z. Wang, W.-J. Chen, N. Deng, Least squares twin parametric-margin support vector machine for classification, Appl. Intell. 39 (3) (2013) 451–464.
[61] Z. Wang, Y.-H. Shao, T.-R. Wu, A GA-based model selection for smooth twin parametric-margin support vector machine, Pattern Recognit. 46 (8) (2013) 2267–2277.
[62] X. Peng, L. Kong, D. Chen, Improvements on twin parametric-margin support vector machine, Neurocomputing 151 (2015) 857–863.
[63] H. Wang, Y. Xu, Z. Zhou, Twin-parametric margin support vector machine with truncated pinball loss, Neural Comput. Appl. 33 (8) (2021) 3781–3798.
[64] J. Xie, K.S. Hone, W. Xie, X. Gao, Y. Shi, X. Liu, Extending twin support vector machine classifier for multi-category classification problems, Intell. Data Anal. 17 (2013) 649–664.
[65] Y. Xu, R. Guo, L. Wang, A twin multi-class classification support vector machine, Cogn. Comput. 5 (4) (2013) 580–588.
[66] D. Tomar, S. Agarwal, Direct acyclic graph based multi-class twin support vector machine for pattern classification, in: Proceedings of the Second ACM IKDD Conference on Data Sciences, CODS '15, ACM, 2015, pp. 80–85.
[67] Y.-H. Shao, W.-J. Chen, W.-B. Huang, Z.-M. Yang, N.-Y. Deng, The best separating decision tree twin support vector machine for multi-class classification, Proc. Comput. Sci. 17 (2013) 1032–1038.
[68] Z. Yang, Y. Shao, X.-S. Zhang, Multiple birth support vector machine for multi-class classification, Neural Comput. Appl. 22 (2013) 153–161.
[69] X. Peng, D. Xu, Robust minimum class variance twin support vector machine classifier, Neural Comput. Appl. 22 (2013) 999–1011.
[70] Z. Qi, Y. Tian, Y. Shi, Robust twin support vector machine for pattern classification, Pattern Recognit. 46 (1) (2013) 305–316.
[71] J. López, S. Maldonado, M. Carrasco, Robust nonparallel support vector machines via second-order cone programming, Neurocomputing 364 (2019) 227–238.
[72] S. Maldonado, J. López, M. Carrasco, A second-order cone programming formulation for twin support vector machines, Appl. Intell. 45 (2016) 265–276.
[73] J. López, S. Maldonado, M. Carrasco, A robust formulation for twin multiclass support vector machine, Appl. Intell. 47 (2017) 1031–1043.
[74] A. Sahleh, M. Salahi, Improved robust nonparallel support vector machines, Int. J. Data Sci. Anal. 1 (2022) 1–14.
[75] V. Piccialli, M. Sciandrone, Nonlinear optimization and support vector machines, 4OR - Q. J. Oper. Res. 16 (2018) 111–149.
[76] B. Schölkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, 2001.
[77] A. Jiménez-Cordero, J.M. Morales, S. Pineda, A novel embedded min-max approach for feature selection in nonlinear support vector machine classification, Eur. J. Oper. Res. 293 (2021) 24–35.
[78] W. Rudin, Real and Complex Analysis, McGraw-Hill, 1987.
[79] S.J. Wright, Primal-Dual Interior-Point Methods, SIAM, 1997.
[80] M. Grant, S. Boyd, Graph implementations for nonsmooth convex programs, in: V. Blondel; S. Boyd, H. Kimura (Eds.), Recent Advances in Learning and Control, Lecture Notes in Control and Information Sciences, Springer-Verlag Limited, 2008, pp. 95–110, http://stanford.edu/boyd/graph_dcp.html.
[81] M. Grant, S. Boyd, CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx (March 2014).
[82] MOSEK ApS, The MOSEK optimization toolbox for MATLAB manual. Version 9.1, 2019. http://docs.mosek.com/9.1/toolbox/index.html.
[83] M. Kelly, R. Longjohn, K. Nottingham, UCI machine learning repository, 2023. http://archive.ics.uci.edu/ml.
[84] Open Data - Government of Canada, 2023. https://open.canada.ca/en/open-data (Accessed on March 5, 2023).
[85] T.Y. Chen, T.H. Tse, Y.-T. Yu, Proportional sampling strategy: a compendium and some insights, J. Syst. Softw. 58 (1) (2001) 65–81.
[86] B. Bischl, M. Binder, M. Lang, T. Pielok, J. Richter, S. Coors, J. Thomas, T. Ullmann, M. Becker, A. Boulesteix, D. Deng, M.T. Lindauer, Hyperparameter optimization: foundations, algorithms, best practices, and open challenges, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 13 (2023).