UNIVERSITY OF PAVIA

Doctor of Philosophy in

APPLIED ECONOMICS AND MANAGEMENT

XXXVI Cycle

Optimization Under Uncertainty: Applications to Machine Learning and Waste Management

---

Co-ordinator:

Prof. Giovanna Magnani

Prof. Alberto Gaggero

Supervisor:

Prof. Francesca Maggioni

Candidate:

Andrea Spinelli

Candidate ID: 490820

Academic Year: 2023/2024

A Caterina e Adriana,
che hanno dato sapore alla matematica

Acknowledgments

First and foremost I would like to thank my PhD supervisor, Prof. Francesca Maggioni, for her valuable guidance and patience during my PhD study. I want to thank her for motivating me and for all the opportunities she has offered during these years. Parts of this thesis are the result of joint works with Prof. Renato De Leone, Prof. Ana Póvoa, Prof. Tânia Ramos, and Prof. Daniele Vigo. I would like to thank all of them for their collaborative efforts and valuable suggestions that have enriched this work. A sincere thanks to Prof. Ana Póvoa and Prof. Tânia Ramos who welcomed me and allowed to carry out this project during the research stay in Lisbon.

I want to extend the appreciation to all the colleagues I had the pleasure to meet these years. A special acknowledgment to Raquel and Erika for all their advice and for making me feel at home.

I am sincerely grateful to my parents, Carmen e Antonio, who have always pushed me on this journey. Thanks for always being there and looking after me all the time. Thanks for providing me with unwavering support and continuous encouragement throughout my years of study.

I would also like to give special thanks my grandma, Gianna, who has taken care of me and she is still protecting me. I hope she is proud of these results and achievements.

I would like to express my gratitude to Adriana and Caterina, to which the thesis is dedicated. Without their inspiration, I would not be the person I am today.

I am obliged to Giacomo for standing by my side during these years. I would have not undertaken this journey without his encouragement.

Finally, I thank my friends of a lifetime, for all the laughs and adventures we have shared, and for still being here.

Preface

In recent years, the evolution of Artificial Intelligence, and in particular of *Machine Learning* (ML) methods, has led to several advances in a variety of application fields. The goal of ML is to create autonomous systems able to learn from past experience in the form of input observations. Given the massive availability of data, ML is currently a field of strategic importance, whose success can be mostly attributed to research at the interface of computer science, statistics, and *Operational Research* (OR) ([59]).

A second challenging area that has recently gained a noticeable attention both from academia and companies is sustainability. Nowadays, all the decisions that are taken at strategical, tactical and operational levels need to account for environmental impacts ([7, 141]). Under the framework of sustainability, waste management is a growing research area since it involves a wide variety of technical, institutional and economical factors. In such a complex context, OR techniques may help service providers and decision makers to implement cost-effective and sustainable plans.

In classic OR problems all parameters are assumed to be perfectly known when taking decisions. Nevertheless, in many practical cases some parameters may be revealed over time or subject to perturbations. For instance, real-world data used as input parameters in ML methods may be plagued by measurement errors or mistakes in the data collection process. Although the exact value of the parameters is not known, nevertheless decisions need to be taken. For this reason, optimization under uncertainty techniques have been devised in the OR literature. Depending on the degree of information about the uncertain parameters, different approaches have been explored. In this thesis, we focus our attention on *Stochastic Optimization* (SO, [19, 80, 134]) and *Robust Optimization* (RO, [9, 13, 158]).

SO assumes that the decision maker has complete knowledge about the underlying uncertainty in a probabilistic sense: the probability distribution of the uncertain parameters is known or can be empirically estimated from historical data or experts' opinions ([126]). Whenever the problem depends on a sequence of decisions over time, the uncertainty is modeled through scenario trees, discretizing the future outcome of the random parameters.

This may cause an increase of the computational complexity of the SO model, requiring the use of decomposition techniques or heuristic approaches.

RO, on the other hand, considers uncertainty in OR problems without the use of probability distributions. Indeed, uncertainty sets that contain all possible values of the uncertain parameters are constructed and the corresponding optimization model looks for a solution which is optimal for all realizations inside the uncertainty sets ([14]). RO techniques prevent the optimization model against the worst possible realization of the uncertain parameters within the prescribed uncertainty sets. Unfortunately, this may cause an excess of conservatism.

In this thesis, we apply Robust Optimization and Stochastic Optimization techniques to Machine Learning methods (Chapters 1 to 3) and waste collection problems (Chapter 4). The common feature of these applications is the occurrence of uncertainty in some relevant parameters. We show that taking into account uncertainty in such a kind of problems provides better accuracies and efficient policies when compared to the deterministic counterpart problems.

The thesis is structured as follows.

In Chapter 1, we tackle the problem of binary classification by extending the *Support Vector Machine* (SVM) approach of [90] with nonlinear classifiers. In order to take into account uncertainty in the training observations, we construct bounded-by-norm uncertainty sets around each sample. By means of RO techniques, we derive the robust counterpart of the deterministic model. Extensive numerical experiments show the advantages of the proposed formulation with respect to classic methods in the extant ML literature.

In Chapter 2, we study a classification problem where the classifying categories are more than two. We extend the *Twin Parametric Margin SVM* (TPMSVM) formulation of [116] to the context of multiclass classification. We consider linear and nonlinear classifiers, and propose two alternatives for the decision function. Then, the approach is robustified through RO strategies, leading to tractable optimization problems. Finally, the deterministic and the robust models are tested on multiclass real-world datasets.

Chapter 3 presents an application of the techniques developed in Chapters 1 and 2 to a vehicles classification task. We consider the problem of predicting vehicles smog rating score on the basis of different characteristics. Binary and multiclass robust approaches with spherical uncertainty sets are explored. The models are validated on synthetic and real-world datasets, and the results are compared with different approaches in the literature.

Chapter 4 considers the problem of waste collection with stochastic accumulation rate. We propose a multi-stage mixed-integer SO model to solve a waste collection inventory

routing problem. To cope with the computational complexity of the problem, we apply the rolling horizon approach, providing a worst-case analysis on its performance. Given the availability of real-world data, the numerical experiments are designed to measure the impact of stochasticity and evaluate the performance of the rolling horizon approach. We finally report some managerial insights.

To conclude, in Chapter 5 we outline comprehensive conclusions and provide a discussion about further developments of the topics addressed within the thesis.

UNIVERSITY OF PAVIA

Abstract

Optimization Under Uncertainty: Applications to Machine Learning and Waste Management

Andrea Spinelli

In this thesis, we deal with optimization problems affected by uncertainty. The first class of problems we analyze aims at separating sets of data points by means of linear and nonlinear classifiers. The classification task is performed according to variants of the *Support Vector Machine* (SVM) and the uncertainty in real-world data is handled by means of *Robust Optimization* (RO) techniques. In the case of binary classification, we start by formulating a novel SVM-type model with nonlinear classifiers and perfectly known data points. Secondly, to prevent low accuracies in the classification process due to data perturbations, we construct bounded-by-norm uncertainty sets around the samples. Then, we derive the robust counterpart of the deterministic model thanks to RO strategies. To tackle the problem of multiclass classification, we design a new multiclass *Twin Parametric Margin SVM* (TPMSVM). We consider the cases of both linear and kernel-induced boundaries and propose two alternatives for the final decision function. Data perturbations are then included in the model and RO techniques are applied to prevent the TPMSVM against the worst possible realization of the uncertainty. All the aforementioned approaches are tested on real-world datasets, showing the advantages of explicitly considering the uncertainty versus deterministic approaches. The second problem we analyze is related to waste collection. Within this application, uncertainty lies in the waste accumulation rate of the network bins. Since information on the empirical distribution of the uncertainty is available, *Stochastic Optimization* (SO) techniques are applied. We model the waste collection problem as a multi-stage stochastic inventory routing problem, where the decisions are related to the selection of bins to be visited and the corresponding visiting sequence in a predefined time horizon. Given the computational complexity of the model, we solve it through a rolling horizon heuristic approach, and carry out computational experiments on real-data instances. The impact of stochasticity on waste generation is examined through stochastic measures, and the performance of the rolling horizon approach is evaluated. Finally, we discuss some managerial insights.

Contents

List of Figures

List of Tables

Notation

In Chapters 1 to 3, we consider the following notation:

$\mathbb{R}$	Set of real numbers
$\mathbb{R}^+$	Set of nonnegative real numbers
$\mathbb{R}_0^+$	Set of positive real numbers
$a = [a_1, \ldots, a_n] \in \mathbb{R}^n$	Column vector with n components
$a^\top$	Transpose of vector a
$x^{(i)}$	Training data points
$y^{(i)}$	Label of training data points
$\mathcal{H}$	Feature space
$\langle \cdot, \cdot \rangle_{\mathcal{H}}$	Inner product of $\mathcal{H}$
$\|z\|_{\mathcal{H}}$	$\mathcal{H}$-norm of z, $\|z\|_{\mathcal{H}} = \sqrt{\langle z, z \rangle_{\mathcal{H}}}$
$\phi : \mathbb{R}^n \to \mathcal{H}$	Feature map
$k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$	Kernel function, $k(x^{(i)}, x^{(j)}) = \langle \phi(x^{(i)}), \phi(x^{(j)}) \rangle_{\mathcal{H}}$
K	Gram matrix, $K_{ij} = k(x^{(i)}, x^{(j)})$
$\|a\|_p$	ℓ_p-norm of vector a, with $p \in [1, \infty]$
e_n	Column vector of ones in $\mathbb{R}^n$
$\mathbb{1}(c)$	Indicator function: 1 if $c > 0$, 0 otherwise
$\operatorname{sign}(c)$	Signum function: 1 if $c > 0$, -1 if $c < 0$, 0 otherwise
$\mathcal{U}(x)$	Uncertainty set centered in x
$\|\mathcal{S}\|$	Cardinality of set $\mathcal{S}$

Chapter 1

A Novel Robust Optimization Model for Nonlinear Support Vector Machine

Authors: Francesca Maggioni[1] and Andrea Spinelli[2].

Keywords: Machine Learning; Nonlinear Support Vector Machine; Robust Optimization.

This chapter is under first revision in *European Journal of Operational Research*.
Manuscript Reference Number: EJOR-D-23-02279.

[1]Department of Management, Information and Production Engineering, University of Bergamo, Viale G. Marconi 5, Dalmine 24044, Italy. francesca.maggioni@unibg.it
[2]Department of Management, Information and Production Engineering, University of Bergamo, Viale G. Marconi 5, Dalmine 24044, Italy. andrea.spinelli@unibg.it

1.1 Introduction

Support Vector Machine (SVM) is one of the main supervised *Machine Learning* (ML) techniques deployed for classification and regression purposes. Within the *Operational Research* (OR) domain, supervised ML methods are designed to support better decision-making. To this end, a plethora of methodologies have been devised and applied to various OR fields ([39]). In particular, combinatorial optimization ([10, 156]), customer churn prediction ([36, 102, 142]), banking ([46, 87, 163]) and maritime industry ([110, 125]). Introduced in [150], SVM has outperformed most other ML systems, due to its simplicity and better performances. For these reasons, it has been applied in many practical research fields, such as finance ([94, 144]), chemistry ([88, 106]), medicine ([96, 152]), and vehicle smog rating classification ([40, 99]), to name a few.

Hard Margin-SVM (HM-SVM) is the original SVM approach formulated in [150], consisting in finding a hyperplane classifying data into two classes, such that the margin, i.e. the distance from the hyperplane to the nearest point of each class, is maximized. The underlying hypothesis of the HM-SVM is that training data can always be linearly separated, such that no observation is misclassified. To overcome the assumption of linear separability, in [38] the *Soft Margin*-SVM (SM-SVM) is proposed. Within this approach, the optimal hyperplane seeks a trade-off between the maximization of the margin and the minimization of the training error of misclassification.

In order to improve the accuracy of the method, several SVM variants have been devised in the literature. Specifically, in this chapter we focus our attention on the one presented in [90]. According to this approach, data are firstly separated by means of two parallel hyperplanes and then the optimal hyperplane is searched in the strip between them, such that the total number of misclassified points is minimized.

Nevertheless, data points may not be always separable using linear classifiers, disrupting the reliability of the solution. In [22], the extension of the linear SVM is introduced, by considering nonlinear transformation of the data. This approach considers the use of kernel function to embed data points in a higher-dimensional space, without increasing the computational complexity of the problem. Several variants of this technique have been proposed, by considering different properties of the problem (see for example [11, 20, 29, 44, 45, 47, 69, 71, 73, 104, 116, 130, 161]).

For the methods mentioned above, all data points are implicitly assumed to be known exactly. However, in real-world observations this condition may not be always true. Indeed, measurement errors during data collection, random perturbations, presence of noise

and other forms of uncertainty may corrupt the quality of input values, resulting in worsening performances of the algorithm. In recent years different techniques have been investigated with the aim of facing uncertainty in ML methods. Among them, *Robust Optimization* (RO) is one of the main paradigm to protect optimization models against uncertainty (see for example [9, 13, 158]). RO assumes that all possible realizations of the uncertain parameter belong to a prescribed uncertainty set. The corresponding robust model is then obtained by optimizing against the worst-case realization of the parameter across the entire uncertainty set ([14]).

In this chapter, we present a novel SVM-variant aiming at generating two nonlinear decision boundaries such that all the points of a class lie on a specific side of the separators. The optimal classifier is finally searched in the region between them such that the misclassification error is minimized. Given the uncertain nature of real-world observations, we derive a robust counterpart of the deterministic model, by considering different kernel functions and uncertainty sets.

The main contributions of the chapter can be summarized as follows:

- To extend the linear SVM approach of [90] to the nonlinear case, by considering different kernel functions;
- To formulate a robust SVM with bounded-by-ℓ_p-norm uncertainty sets model with nonlinear classifiers;
- To derive bounds on the radii of the uncertainty sets in the input and feature spaces when considering nonlinear classifiers;
- To provide extensive numerical experiments based on real-world datasets with the aim of evaluating the performances of the proposed models.

The remainder of the chapter is organized as follows. Section 1.2 reviews the existing literature on the problem. In Section 1.3, the notation is introduced, along with a brief discussion on selected SVM-type problems. In Section 1.4, the novel deterministic model with nonlinear classifier is introduced. Section 1.5 considers the robust version extension along with the construction of uncertainty sets. In Section 1.6, the computational results are shown. Finally, Section 1.7 concludes the chapter and discusses future works.

1.2 Literature review

SVM is introduced as a pattern recognition technique in [150] for the case of optimal hyperplane and with separable classes. The generalization of the linear approach to the

nonlinear case is proposed in [22], where input vectors are first compared by means of a distance measure and then mapped to a higher-dimensional space (the so-called *feature space*) via a nonlinear transformation. The main drawback of this approach is that training data points are considered separable. In [149] the shortcoming is overcome by relaxing the condition of perfect separability. Indeed, a soft margin error vector is introduced, and the corresponding optimal separation hypersurface maximizes the margin for the correctly classified vectors and minimizes the magnitude of the soft margin error.

The approach presented so far has been applied to other nonlinear SVM variants, leading to alternative formulations. In [104] a kernel-induced decision boundary is derived by considering either a quadratic or piecewise-linear objective function. The corresponding model turns to be convex and is applied in [86] to extract relevant features of breast cancer patients. In [130] the formulation of *ν-Support Vector Classification* (ν-SVC) is proposed for both linear and nonlinear classifiers. This class of algorithm differs from the classical SVM paradigm of [149] since it involves a new parameter ν in the objective function, controlling the fraction of support vectors. In [71] the *TWin Support Vector Machine* (TWSVM) is designed. Contrary to standard SVM, TWSVM determines two nonparallel hyperplanes by solving two small-sized SVM-type problems. In this stream of research, [116] combines the TWSVM with a flexible parametric margin model, deriving the *Twin Parametric Margin Support Vector Machine* (TPMSVM). More recently, in [20] the classical ℓ_2-norm problem has been extended to the more general case of ℓ_p-norm with $p > 1$. Second order cone formulations for the resulting dual and primal problems are then derived. The problem of feature selection in nonlinear SVM is explored in [73]. The authors propose a method based on a min-max optimization problem, embedding a trade-off between model complexity and classification accuracy.

All the aforementioned papers consider only deterministic SVM models, whose underlying hypothesis is that training data points are perfectly known. Unfortunately, in many real-world applications data are plagued by uncertainty caused by corruption or measurement errors. However, the classification algorithm should perform appropriately even after such perturbations ([137]). *Robust Optimization* (RO) is one of the main paradigm to tackle the problem of dealing with uncertain parameters. Depending on the degree of information about data, different uncertainty sets may be constructed. Within the field of RO applied to linear SVM, in [16] hyperellipsoids around data points are considered, leading to a *Second-Order Cone Programming* (SOCP) formulation. A tractable robust counterpart of the classical SVM approach of [38] is derived in [14]. In particular, the authors robustify the soft margin SVM model against feature uncertainty by considering

bounded-by-norm additive perturbations in the training data. In [48] the binary classification problem under feature uncertainty is formulated with uncertainty sets in the form of hyperrectangles and hyperellipsoids around input data. With the same choices of uncertainty sets, in [53] a RO model of the linear SVM variant presented in [90] is proposed. The reader is referred to [154] for a survey on linear SVM under uncertainty.

As far as it concerns RO techniques applied to nonlinear SVM, different approaches exist in literature. In [15] and [8] the kernel matrix is assumed to be affected by uncertainty, due to feature perturbations in the input data. A decomposition of the kernel matrix as a combination of positive semidefinite matrices with bounded-by-ℓ_p-norm coefficients is proposed. The main limitation of this approach is that the functional form of the kernel matrices is typically unknown. Thus, it is not obvious how to characterize the elements in the uncertainty set, unless by using a sampling procedure. In [17] and in [147] data points in the input space are subject to uncertain but bounded-by-ℓ_p-norm perturbations. Robustified models are derived for both linear and nonlinear classifiers. In the latter case, when data are mapped to the feature space, an additive and unknown perturbation is introduced too. The robustification of the nonlinear SVM problem leads to a tractable SOCP formulation. A related work on bounded uncertainty sets is [158]. In [146] the stability of linear and quadratic programming SVMs with bounded noise in the input space is investigated by using linear and nonlinear discriminant functions. Polyhedral uncertainty sets are considered in [56], [58] and [74], based on the nonlinear classifier of [104].

RO techniques are applied to other SVM-type problems too. In [119] a robust TWSVM classifier is proposed, by considering data uncertainty in the variance matrices of the two classes. In [124] the robust counterpart of TWSVM is derived. For the nonlinear case, only Gaussian kernel and ellipsoidal uncertainty sets around data points are considered, resulting in SOCP formulation. In the following chapter, the robust and multiclass extension of the TPMSVM is provided ([41]). A complete survey on recent developments on TWSVM models can be found in [143].

When information on the probability distribution of the training data are available, RO is combined with *Chance-Constrained Programming* (CCP) and *Distributionally Robust Optimization* (DRO) ([72, 77]). The *Minimax Probability Machine* (MPM) is the first robust approach in the SVM context that minimizes the worst-case probability of misclassification ([85]). In [102] the MPMs are extended and applied to the robust profit-driven churn prediction. Within the MPM framework, the use of Cobb-Douglas function for maximizing the expected class accuracies under a worst-case distribution setting is

proposed in [101]. The problem of robust feature selection with CCP is explored in [92] by using difference of convex functions. Within the multiclass context, in [91] a robust formulation for multiclass classification via TWSVM is proposed. As far as it concern DRO methods applied to SVM, we mention the work of [53] where a moment-based distributionally robust formulation of the [90] approach is designed. Finally, a combination of CCP and DRO techniques applied to linear SVMs with uncertain data is explored in [78, 153].

All the approaches discussed so far are listed in Table 1.1. For a comprehensive review of RO in the field of SVM the reader is referred to [137].

The contribution of this chapter differs from the literature described above in several aspects. First of all, we present a novel optimization model with nonlinear classifiers, extending the approach of [90]. Secondly, we consider general bounded-by-ℓ_p-norm uncertainty sets around each observation, deriving closed-form expressions of the bounds in the feature space for some of typically used kernel function in ML literature. Furthermore, we derive the robust counterpart of the deterministic approach, protecting the model against data uncertainty.

1.3 Background and notation

In this section, we briefly recall some deterministic SVM-type models for pattern classification: the linear *Soft Margin*-SVM (SM-SVM, [149]), the SVM *Formulation* of the approach proposed in [90], and the *Generalized*-SVM (G-SVM, [104]) for nonlinear classification.

1.3.1 A selected review of SVM models

Let $\{x^{(i)}, y^{(i)}\}_{i=1}^m$ be the set of training data points, where $x^{(i)} \in \mathbb{R}^n$ is the vector of features, and $y^{(i)} \in \{-1, +1\}$ is the label representing the class to which the i-th data point belongs. In particular, we denote by $\mathcal{A}$ and $\mathcal{B}$ the class of *positive* (label "+1") and *negative* (label "−1") data points, respectively.

The Soft Margin Support Vector Machine (SM-SVM)

The *Soft Margin*-SVM approach (SM-SVM), firstly introduced in [38], finds the best separating hyperplane $H := (w, \gamma)$ defined by the equation $w^\top x = \gamma$, where $w \in \mathbb{R}^n$ and $\gamma \in \mathbb{R}$,

	SVM		Type of Robust Methodology						
	Linear classifier	Nonlinear classifier	Box RO	Ellipsoidal RO	Polyhedral RO	Bounded-by-norm RO	Matrix RO	Chance-Constrained	Distributionally RO
Vapnik & Chervonenkis (1974), [150]	✓								
Boser et al. (1992), [22]		✓							
Vapnik (1995), [149]	✓	✓							
Mangasarian (1998), [104]		✓							
Lee et al. (2000), [86]		✓							
Schölkopf et al. (2000), [130]	✓	✓							
Fung et al. (2002), [58]	✓	✓			✓				
Lanckriet et al. (2002), [85]	✓	✓		✓				✓	✓
El Ghaoui et al. (2003), [48]	✓		✓	✓					
Bhattacharyya (2004), [16]	✓			✓					
Bi & Zhang (2005), [17]	✓	✓				✓			
Trafalis & Gilbert (2006), [147]	✓	✓				✓			
Jayadeva et al. (2007), [71]	✓	✓							
Liu & Potra (2009), [90]	✓								
Xu et al. (2009), [158]	✓	✓				✓			
Bhadra et al. (2010), [15]		✓					✓	✓	
Trafalis & Alwazzi (2010), [146]	✓	✓				✓			
Peng (2011), [116]	✓	✓							
Ben-Tal et al. (2012), [8]		✓				✓	✓		
Ju & Tian (2012), [74]	✓	✓			✓				
Peng & Xu (2013), [119]	✓	✓					✓		
Qi et al. (2013), [124]	✓	✓		✓					
Fan et al. (2014), [56]	✓	✓			✓				
López et al. (2017), [91]	✓	✓						✓	
López et al. (2018), [92]	✓	✓						✓	
Wang et al. (2018), [153]	✓							✓	✓
Bertsimas et al. (2019), [14]	✓					✓			
Blanco et al. (2020), [20]	✓	✓							
Maldonado et al. (2020), [102]	✓	✓						✓	
Jiménez Cordero et al. (2021), [73]		✓							
Faccini et al. (2022), [53]	✓		✓	✓					✓
Maldonado et al. (2022), [101]	✓	✓						✓	
Khanjani-Shiraz et al. (2023), [78]	✓							✓	✓

Table 1.1: A selected SVM literature review. In the first row of the table the methodological contributions are listed in chronological order. Second and third rows specify the type of SVM classifier (linear or nonlinear). Finally, the RO methodologies employed in the articles are explored in rows four to ten.

as solution of the following ℓ_q-model, $q \geqslant 1$ (see [20]):

$$\begin{aligned} \min_{w,\gamma,\xi} \quad & \|w\|_q^q + \nu \sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad & y^{(i)}(w^\top x^{(i)} - \gamma) \geqslant 1 - \xi_i \qquad i = 1, \ldots, m \\ & \xi_i \geqslant 0 \qquad\qquad\qquad\qquad\quad i = 1, \ldots, m. \end{aligned} \tag{1.1}$$

The vector $\xi \in \mathbb{R}^m$ is the soft margin error vector and $\nu \geqslant 0$ is a regularization parameter, balancing the trade-off between the maximization of the margin (i.e. the minimization of $\|w\|_q^q$), and the minimization of the misclassification error. Indeed, data point $x^{(i)}$ is correctly classified by the separating hyperplane, i.e. it lies on the correct side of H, if $0 \leqslant \xi_i \leqslant 1$, otherwise is misclassified.

A new data point $x \in \mathbb{R}^n$ is classified as *positive* or *negative* depending on the decision function $\mathbb{1}\big(w^\top x - \gamma\big)$: if it is equal to 1, then x is assigned to class $\mathcal{A}$, otherwise to class $\mathcal{B}$.

The SVM Formulation of [90]

Instead of a single hyperplane as in the case of classical SM-SVM, in [90] a novel approach involving two parallel hyperplanes is proposed. The starting point of the formulation employs the solutions of model (1.1) with $q = 1$ to obtain the hyperplane $H_0 := (w, \gamma)$ and the soft margin error vector ξ. Then, H_0 is shifted in order to determine two parallel hyperplanes $H_\mathcal{A} := (w, \gamma - 1 + \omega_A)$ and $H_\mathcal{B} := (w, \gamma + 1 - \omega_B)$, where:

$$\omega_\mathcal{A} := \max_{i:x^{(i)} \in \mathcal{A}} \big\{\xi_i\big\}, \quad \omega_\mathcal{B} := \max_{i:x^{(i)} \in \mathcal{B}} \big\{\xi_i\big\}, \tag{1.2}$$

satisfying the following properties:

(P1) all points of class $\mathcal{A}$ lie on one halfspace of $H_\mathcal{A}$;

(P2) all points of class $\mathcal{B}$ lie on the opposite halfspace of $H_\mathcal{B}$;

(P3) the intersection of the convex hulls of $\mathcal{A}$ and $\mathcal{B}$ is contained in the region between $H_\mathcal{A}$ and $H_\mathcal{B}$.

Finally, the optimal separating hyperplane $H := (w, b)$ is determined such that is parallel to $H_\mathcal{A}$ and $H_\mathcal{B}$, lies in their strip, and the number of misclassified points is minimized. These conditions are satisfied finding the optimal parameter b, solution of the following

problem:

$$\begin{aligned} \min_{b} \quad & \sum_{i:x^{(i)}\in\mathcal{A}} \mathbb{1}\big(w^\top x^{(i)} - b\big) + \sum_{i:x^{(i)}\in\mathcal{B}} \mathbb{1}\big(b - w^\top x^{(i)}\big) \\ \text{s.t.} \quad & \gamma + 1 - \omega_{\mathcal{B}} \leqslant b \leqslant \gamma - 1 + \omega_{\mathcal{A}}. \end{aligned} \tag{1.3}$$

Similarly to SM-SVM, a new data point $x \in \mathbb{R}^n$ is classified in class $\mathcal{A}$ or $\mathcal{B}$ depending on the decision rule $\mathbb{1}\big(w^\top x - b\big)$.

The Generalized Support Vector Machine (G-SVM)

Data points coming from real-world measurements may not be always separable by means of an hyperplane and, even with *ad hoc* variants of linear SVM, the misclassification error may be significant. This observation motivates the idea of considering nonlinear kernel-induced decision boundaries (see [38]).

The concept behind this approach is the following: the training data points are mapped into a higher-dimensional space, where a separating hyperplane is constructed, yielding to a nonlinear decision hypersurface in $\mathbb{R}^n$. Specifically, a function $\phi(\cdot)$, usually referred as *feature map*, is introduced to map data from the *input space* $\mathbb{R}^n$ to a *feature space* $\mathcal{H}$, equipped with the dot product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$. Thus, model (1.1) in the feature space becomes:

$$\begin{aligned} \min_{\overline{w},\gamma,\xi} \quad & \|\overline{w}\|_{\mathcal{H}} + \nu \sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad & y^{(i)}(\langle \overline{w}, \phi(x^{(i)})\rangle_{\mathcal{H}} - \gamma) \geqslant 1 - \xi_i \qquad i = 1,\dots,m \\ & \xi_i \geqslant 0 \qquad\qquad\qquad\qquad\quad i = 1,\dots,m, \end{aligned} \tag{1.4}$$

where $\overline{w}$ refers to the vector defining the linear classifier in the feature space and the norm in $\mathcal{H}$ is induced by its inner product, i.e., for $z \in \mathcal{H}$, $\|z\|_{\mathcal{H}} := \sqrt{\langle z, z\rangle_{\mathcal{H}}}$.

The vector $\overline{w}$ can be decomposed as a finite linear combination of $\phi(x^{(j)})$, $j = 1,\dots,m$:

$$\overline{w} = \sum_{j=1}^{m} y^{(j)} u_j \phi(x^{(j)}), \tag{1.5}$$

for some coefficients $u_j \in \mathbb{R}$. Unfortunately, the expression of the mapping $\phi(\cdot)$ is usually unknown and $\mathcal{H}$ is potentially an infinite-dimensional space ([131]). For these reasons, model (1.4) cannot be solved in practice. To overcome this problem, a symmetric and positive semidefinite kernel $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is introduced to measure the similarity of two observations. Examples of kernel function typically used in ML literature are reported in Table 1.2. For a comprehensive overview, the reader is referred to [131].

Let K be the associate Gram matrix, i.e., $K_{ij} := k(x^{(i)}, x^{(j)}) = \langle \phi(x^{(i)}), \phi(x^{(j)})\rangle_{\mathcal{H}}$.

Kernel function	$k(x, x')$	Parameter
Homogeneous polynomial	$k(x, x') = (x^\top x')^d$	$d \in \mathbb{N}$
Inhomogeneous polynomial	$k(x, x') = (c + x^\top x')^d$	$c \in \mathbb{R}^+$, $d \in \mathbb{N}$
Gaussian Radial Basis Function (RBF)	$k(x, x') = \exp\left(-\frac{\|x - x'\|_2^2}{2\alpha^2}\right)$	$\alpha \in \mathbb{R}_0^+$
Sigmoid	$k(x, x') = \tanh(a\ x^\top x'$	$a \in \mathbb{R}$, $b \in \mathbb{R}$

Table 1.2: Examples of kernel functions. The first column reports the name of the functions. The second column provides their mathematical expressions. Finally, the third column contains the related relevant parameters.

The properties of $k(\cdot, \cdot)$ imply that K is a real, symmetric and positive semidefinite $m \times m$ matrix ([122]). Consequently, for all $i = 1, \dots, m$ the scalar product $\langle \overline{w}, \phi(x^{(i)}) \rangle_{\mathcal{H}}$ in the first set of constraints of model (1.4) can be formulated as:

$$\langle \overline{w}, \phi(x^{(i)}) \rangle_{\mathcal{H}} = \sum_{j=1}^{m} y^{(j)} u_j \langle \phi(x^{(j)}), \phi(x^{(i)}) \rangle_{\mathcal{H}} = \sum_{j=1}^{m} K_{ij} y^{(j)} u_j = K_i D u,$$

where D is a diagonal matrix with $D_{ii} := y^{(i)}$ and $u = [u_1, \dots, u_m]^\top$. Furthermore, the norm $\|\overline{w}\|_{\mathcal{H}}$ can be expressed in terms of matrices K and D as:

$$\|\overline{w}\|_{\mathcal{H}}^2 = \langle \overline{w}, \overline{w} \rangle_{\mathcal{H}} = \sum_{i=1}^{m} \sum_{j=1}^{m} y^{(i)} y^{(j)} u_i u_j k(x^{(i)}, x^{(j)}) = (Du)^\top K (Du).$$

Due to the structure of D, it holds that $\|Du\|_q = \|u\|_q$ for all $q \in [1, \infty]$, since:

$$\|Du\|_q = \left(\sum_{i=1}^{m} \left| y^{(i)} u_i \right|^q \right)^{\frac{1}{q}} = \left(\sum_{i=1}^{m} |u_i|^q \right)^{\frac{1}{q}} = \|u\|_q .$$

Besides, equality (1.5) states that vector $\overline{w}$ depends linearly on its coefficients u_j. Thus, as suggested in [86], in order to minimize $\|\overline{w}\|_{\mathcal{H}}$ we minimize the magnitude of $\|u\|_q$. Therefore, model (1.4) can be rewritten as:

$$\begin{aligned} \min_{u,\gamma,\xi} \quad & \|u\|_q^q + \nu \sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad & y^{(i)} \left(\sum_{j=1}^{m} K_{ij} y^{(j)} u_j - \gamma \right) \geqslant 1 - \xi_i \qquad i = 1, \dots, m \\ & \xi_i \geqslant 0 \qquad\qquad\qquad\qquad\qquad\quad i = 1, \dots, m, \end{aligned} \tag{1.6}$$

or, equivalently, in matrix form:

$$\begin{aligned} \min_{u,\gamma,\xi} \quad & \|u\|_q^q + \nu e_m^\top \xi \\ \text{s.t.} \quad & D(KDu - e_m\gamma) \geqslant e_m - \xi \\ & \xi \geqslant 0. \end{aligned} \tag{1.7}$$

Model (1.7) with $q = 1$ corresponds to the *Generalized*-SVM (G-SVM) presented in [104].

Within this context, the hyperplane in the feature space translates into a nonlinear separating decision boundary S in the input space, induced by the kernel function $k(\cdot, \cdot)$. Hypersurface S is defined implicitly by the following equation:

$$\sum_{i=1}^{m} k(x, x^{(i)}) y^{(i)} u_i = \gamma, \tag{1.8}$$

where $u \in \mathbb{R}^m$ and $\gamma \in \mathbb{R}$ are the solutions of model (1.6). Hereinafter, a kernel-induced decision boundary S in the input space satisfying equation (1.8) will be denoted by $S := (u, \gamma)$.

When a new data point $x \in \mathbb{R}^n$ occurs, it is classified either in class $\mathcal{A}$ or $\mathcal{B}$ according to whether the decision function:

$$\mathbb{1}\left(\sum_{i=1}^{m} k(x, x^{(i)}) y^{(i)} u_i - \gamma \right)$$

yields 1 or 0, respectively.

1.4 A novel approach for deterministic nonlinear SVM

In this section, we derive an extension of the [90] approach to the nonlinear case. Thus, nonlinear separating hypersurfaces in the input space, satisfying the aforementioned properties **(P1)**-**(P3)**, are considered.

First of all, by solving model (1.6), we find an initial decision boundary $S_0 := (u, \gamma)$ which induces a first nonlinear separation in the input space. The nonlinear hypersurface S_0 corresponds to a linear classifier H_0 in the feature space. As in [90], we set $q = 1$, corresponding to the maximization of the margin with respect to the ℓ_∞-norm. This choice provides a good compromise between structural risk minimization, related to the misclassification error, and parsimony since it automatically performs feature selection, by making zero nonrelevant components of the normal vector u (see [84, 93]). Moreover,

with $q = 1$, problem (1.6) reduces to a linear problem.

Then, as in (1.2), for each of the two classes, we compute the greatest misclassification error through the following formulas:

$$\omega_{\mathcal{A}} := \max_{i=1,\dots,m} (D\xi)_i \qquad \omega_{\mathcal{B}} := \max_{i=1,\dots,m} (-D\xi)_i. \tag{1.9}$$

Due to the structure of problem (1.6), the modulus of $-1+\omega_{\mathcal{A}}$ represents the distance of the farthest misclassified point of class $\mathcal{A}$ from H_0 in the feature space, and similarly for $1-\omega_{\mathcal{B}}$. However, it may happen that H_0 already classifies correctly all the data points of at least one of the two classes. Assume, without loss of generality, that it happens for class $\mathcal{A}$. This implies that $0 \leqslant \xi_i \leqslant 1$, for all i such that $x^{(i)} \in \mathcal{A}$. Thus, the modulus of $-1+\omega_{\mathcal{A}}$ is just the distance from the closest data points in $\mathcal{A}$ to the hyperplane H_0. Accordingly to the literature of SVM (see [38]), we call the points at distance $|-1+\omega_{\mathcal{A}}|$ and $|1-\omega_{\mathcal{B}}|$ the *support vectors* of class $\mathcal{A}$ and $\mathcal{B}$, respectively.

After the computation of $\omega_{\mathcal{A}}$ and $\omega_{\mathcal{B}}$, we shift H_0 by $-1+\omega_{\mathcal{A}}$ and $1-\omega_{\mathcal{B}}$ in the feature space, getting two parallel hyperplanes to H_0, namely $H_{\mathcal{A}}$ and $H_{\mathcal{B}}$, passing through the support vectors of the corresponding class. In the input space two nonlinear hypersurfaces $S_{\mathcal{A}}$ and $S_{\mathcal{B}}$ are derived, defined as $S_{\mathcal{A}} := (u, \gamma - 1 + \omega_{\mathcal{A}})$ and $S_{\mathcal{B}} := (u, \gamma + 1 - \omega_{\mathcal{B}})$, respectively, accordingly to equation (1.8). With this choice, properties **(P1)-(P2)** are satisfied by $H_{\mathcal{A}}$ and $H_{\mathcal{B}}$ in the feature space, and by $S_{\mathcal{A}}$ and $S_{\mathcal{B}}$ in the input space.

Finally, the optimal separating hypersurface $S := (u, b)$ is obtained. The parameter b is the solution of the following linear search procedure, aiming to minimize the overall number of misclassified points:

$$\begin{aligned} \min_{b} \quad & \sum_{i=1}^{m} \mathbb{1}\left(y^{(i)} b - y^{(i)} \sum_{j=1}^{m} K_{ij} y^{(j)} u_j \right) \\ \text{s.t.} \quad & \gamma + 1 - \omega_{\mathcal{B}} \leqslant b \leqslant \gamma - 1 + \omega_{\mathcal{A}}. \end{aligned} \tag{1.10}$$

The decision boundary S in the input space is induced by an hyperplane H in the feature space, lying in the strip between $H_{\mathcal{A}}$ and $H_{\mathcal{B}}$, and satisfying property **(P3)**.

Thus, a new observation $x \in \mathbb{R}^n$ is classified according to the decision function:

$$\mathbb{1}\left(\sum_{i=1}^{m} k(x, x^{(i)}) y^{(i)} u_i - b \right).$$

For the sake of clarity, all the steps of the approach discussed so far are schematically reported in Pseudocode 1.

Pseudocode 1 A novel approach for deterministic nonlinear SVM

Input: $\{x^{(i)}, y^{(i)}\}_{i=1}^{m}$, $\nu \geqslant 0$, $k(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$.

1: Calculate matrix $K_{ij} = k\big(x^{(i)}, x^{(j)}\big), i, j = 1, \ldots, m$ and the diagonal matrix of the labels $D_{ii} = y^{(i)}, i = 1, \ldots, m$.

2: Solve model (1.6) with $q = 1$.

3: Find the initial separating surface $S_0 = (u, \gamma)$, defined by equation (1.8).

4: Compute $\omega_{\mathcal{A}}$ and $\omega_{\mathcal{B}}$, according to formulas (1.9).

5: Shift S_0 to get the separating surface for each class, $S_{\mathcal{A}} = (u, \gamma - 1 + \omega_{\mathcal{A}})$ and $S_{\mathcal{B}} = (u, \gamma + 1 - \omega_{\mathcal{B}})$, defined by (1.8).

6: Solve model (1.10), obtaining the optimal parameter b.

Output: The optimal decision boundary $S = (u, b)$, defined by (1.8).

To conclude, we visualize in Figure 1.1 the interpretation of the novel approach, applied to a bidimensional dataset, in the case of Gaussian RBF kernel with $\alpha = 1.9$. The nonlinear decision boundaries are the contour lines of the implicit function (1.8).

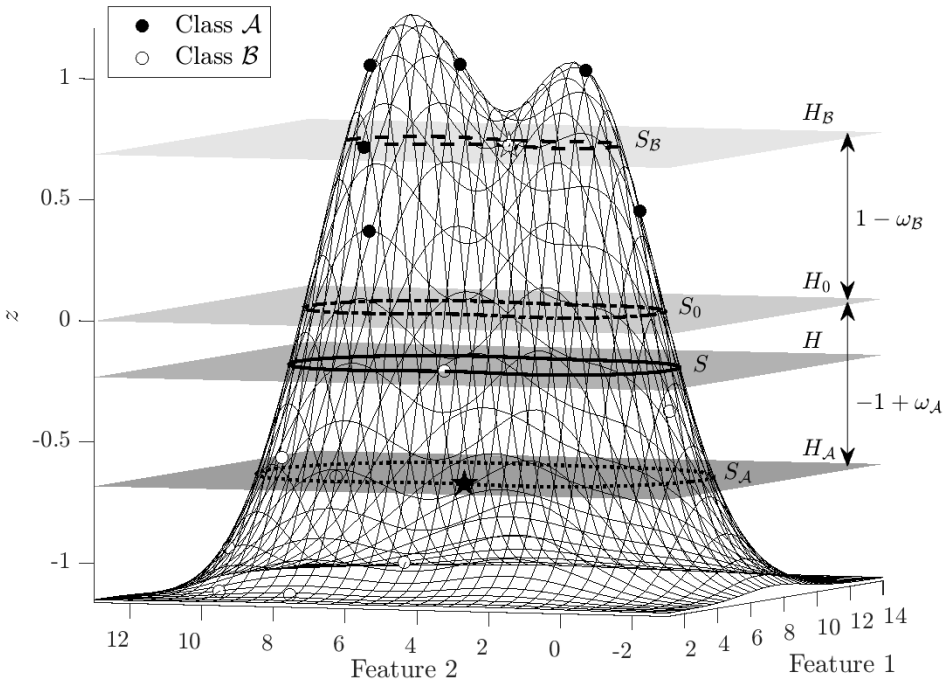


Figure 1.1: Graphical representation of the implicit function (1.8), in the case of Gaussian RBF kernel ($\alpha = 1.9$), along with the separating hyperplanes and decision boundaries. Parameter ν in the objective function of (1.6) has been set to 1. Support vectors are drawn as stars.

1.5 A robust model for nonlinear SVM

In this section, we derive the robust counterpart of the deterministic model introduced before, considering uncertainties in the input data. The uncertainties are taken into consideration when training the classifier by constructing an uncertainty set $\mathcal{U}(x^{(i)})$ around each observation $x^{(i)}$. Thus, the problem is to optimize against the worst-case realization across the entire uncertainty sets of all the observations ([14]). The uncertainty set can be defined in different ways, according to the degree of uncertainty that is considered in the model. Typically box, ellipsoidal and polyhedral uncertainty sets are considered because they lead to tractable optimization models (see [48, 56, 93], respectively).

The robust counterpart of the Liu and Potra linear model is derived in [53], where the uncertainty sets are modelled as box or ellipsoids. Unfortunately, in the nonlinear context, when data points $x^{(i)}$ are mapped in the feature space $\mathcal{H}$ via $\phi(\cdot)$, *a priori* control about the shape and the properties of the uncertainty set $\mathcal{U}\big(\phi(x^{(i)})\big)$ is not possible. In addition, a closed-form expression of $\phi(\cdot)$ is rarely available. Therefore, further assumptions when constructing $\mathcal{U}\big(\phi(x^{(i)})\big)$ are necessary.

The remainder of the section is organized as follows. In Subsection 1.5.1 uncertainty sets bounded by a general ℓ_p-norm are constructed by considering different kernel functions. Bounds on the radii of the uncertainty sets in the feature space are derived in Subsection 1.5.2. Finally, in Subsection 1.5.3 the robust counterpart of model (1.6) is formulated.

1.5.1 The construction of the uncertainty sets

We assume that each observation $x^{(i)}$ in the input space is subject to an additive and unknown perturbation vector $\sigma^{(i)}$. In addition, we assume that its ℓ_p-norm, with $p \in [1, \infty]$, can be bounded by a known nonnegative constant $\eta^{(i)}$. Therefore, the uncertainty set around $x^{(i)}$ in the input space has the following expression:

$$\mathcal{U}_p(x^{(i)}) := \left\{ x \in \mathbb{R}^n : x = x^{(i)} + \sigma^{(i)}, \|\sigma^{(i)}\|_p \leqslant \eta^{(i)} \right\}, \tag{1.11}$$

with $p \in [1, \infty]$. The nonnegative parameter $\eta^{(i)}$ calibrates the degree of conservatism. If $\eta^{(i)} = 0$, then $\sigma^{(i)}$ is the zero vector of $\mathbb{R}^n$ and $\mathcal{U}_p(x^{(i)})$ coincides with $x^{(i)}$. Different ℓ_p-norms lead to different geometrical properties of $\mathcal{U}_p(x^{(i)})$: ℓ_1-norm, ℓ_2-norm and ℓ_∞-norm yields to polyhedral, ellipsoidal and box uncertainty set, respectively.

According to equation (1.11), if x belongs to $\mathcal{U}_p(x^{(i)})$, then it can be written as $x^{(i)} + \sigma^{(i)}$. The application of the feature map $\phi(\cdot)$ implies that x will be projected onto the

feature space $\mathcal{H}$. Since $x = x^{(i)} + \sigma^{(i)}$, we argue that $\phi(x)$ results to be a perturbation of $\phi(x^{(i)})$, through a perturbation vector $\zeta^{(i)} \in \mathcal{H}$. Therefore:

$$\phi(x) = \phi(x^{(i)} + \sigma^{(i)}) = \phi(x^{(i)}) + \zeta^{(i)},$$

where the $\mathcal{H}$-norm of $\zeta^{(i)}$ is bounded a nonnegative constant $\delta^{(i)}$. The latter may be unknown but, in turn, depends on the known bound $\eta^{(i)}$ in the input space, i.e. $\delta^{(i)} = \delta^{(i)}\big(\eta^{(i)}\big)$. Specifically, if $\phi(\cdot)$ is associated to a kernel function $k(\cdot,\cdot)$, then it is possible to derive that ([158]):

$$\begin{aligned}
\left\|\zeta^{(i)}\right\|_{\mathcal{H}}^2 &= \left\|\phi(x) - \phi(x^{(i)})\right\|_{\mathcal{H}}^2 \\
&= \left\|\phi(x^{(i)} + \sigma^{(i)}) - \phi(x^{(i)})\right\|_{\mathcal{H}}^2 \\
&= \langle \phi(x^{(i)} + \sigma^{(i)}) - \phi(x^{(i)}), \phi(x^{(i)} + \sigma^{(i)}) - \phi(x^{(i)}) \rangle_{\mathcal{H}} \\
&= \langle \phi(x^{(i)} + \sigma^{(i)}), \phi(x^{(i)} + \sigma^{(i)}) \rangle_{\mathcal{H}} - 2\langle \phi(x^{(i)} + \sigma^{(i)}), \phi(x^{(i)} \rangle_{\mathcal{H}} + \langle \phi(x^{(i)}), \phi(x^{(i)}) \rangle_{\mathcal{H}} \\
&= k(x^{(i)} + \sigma^{(i)}, x^{(i)} + \sigma^{(i)}) - 2k(x^{(i)} + \sigma^{(i)}, x^{(i)}) + k(x^{(i)}, x^{(i)}).
\end{aligned}$$

The previous set of equalities holds regardless of the choice of $k(\cdot,\cdot)$. Interestingly, it can be noted that, if $\sigma^{(i)} = 0$, then the last right-hand side is equal to zero. This confirms the fact that if no uncertainty occurs in the input space, no uncertainty will occur in the feature space too. Thus, $\eta^{(i)} = 0$ implies $\delta^{(i)} = 0$.

Hence, by combining all the previous results, we model the uncertainty set around $\phi(x^{(i)})$ in the feature space as:

$$\mathcal{U}_{\mathcal{H}}\big(\phi(x^{(i)})\big) := \left\{ z \in \mathcal{H} : z = \phi(x^{(i)}) + \zeta^{(i)}, \|\zeta^{(i)}\|_{\mathcal{H}} \leqslant \delta^{(i)} \right\}. \tag{1.12}$$

In the case of homogeneous polynomial kernel, inhomogeneous polynomial kernel and Gaussian RBF kernel, it is possible to derive a closed-form expression for the bound $\delta^{(i)}$ in the feature space, knowing the bound $\eta^{(i)}$ in the input space.

1.5.2 Bounds on the uncertainty sets in the feature space

Let us now consider a symmetric and positive semidefinite kernel $k(\cdot,\cdot)$, whose corresponding feature map is $\phi(\cdot)$. In the following, we derive closed-form expressions for the bound $\delta^{(i)}$ by analysing separately the polynomial kernel and the Gaussian RBF kernel. Below, we provide the results and relegate the proofs to the Appendix A.1.

Proposition 1 (Polynomial kernel). Let $\mathcal{U}_p(x^{(i)})$ and $\mathcal{U}_{\mathcal{H}}\big(\phi(x^{(i)})\big)$ be the uncertainty

sets in the input and in the feature space as in (1.11) and (1.12), respectively, with $p \in [1, \infty]$. Consider the inhomogeneous polynomial kernel of degree $d \in \mathbb{N}$ and additive constant $c \geqslant 0$, with $\delta^{(i)} \equiv \delta_{d,c}^{(i)}$, and:

$$C = C(n,p) = \begin{cases} 1, & 1 \leqslant p \leqslant 2 \\ n^{\frac{p-2}{2p}}, & p > 2. \end{cases}$$

(i) If $d = 1$, then the bound in the feature space is:

$$\delta_{1,c}^{(i)} = C\eta^{(i)}. \tag{1.13}$$

(ii) If $d > 1$, then:

$$\delta_{d,c}^{(i)} = \sqrt{\left(\delta_{d,0}^{(i)}\right)^2 + \sum_{k=1}^{d-1} \binom{d}{k} c^k \left[\sum_{j=1}^{d-k} \binom{d-k}{j} \left\|x^{(i)}\right\|_2^{d-k-j} \left(C\eta^{(i)}\right)^j \right]^2}, \tag{1.14}$$

where $\delta_{d,0}^{(i)}$ is the bound for the corresponding homogeneous polynomial kernel:

$$\delta_{d,0}^{(i)} = \sum_{k=1}^{d} \binom{d}{k} \left\|x^{(i)}\right\|_2^{d-k} \left(C\eta^{(i)}\right)^k. \tag{1.15}$$

Notice that when $c = 0$, eq. (1.14) reduces to (1.15).

Proposition 2 (Gaussian RBF kernel). Let $\mathcal{U}_p(x^{(i)})$ and $\mathcal{U}_{\mathcal{H}}\big(\phi(x^{(i)})\big)$ be the uncertainty sets in the input and in the feature space as in (1.11) and (1.12), respectively, with $p \in [1, \infty]$. Consider the Gaussian RBF kernel with parameter $\alpha > 0$ and $\delta^{(i)} \equiv \delta_{\alpha}^{(i)}$. If:

$$C = C(n,p) = \begin{cases} 1, & 1 \leqslant p \leqslant 2 \\ n^{\frac{p-2}{2p}}, & p > 2, \end{cases}$$

then:

$$\delta_{\alpha}^{(i)} = \sqrt{2 - 2\exp\left(-\frac{\left(C\eta^{(i)}\right)^2}{2\alpha^2}\right)}. \tag{1.16}$$

We observe that Propositions 1-2 are consistent with Lemma 7 presented in [158]. However, in this chapter we specify the bound for particular kernels and extend the results for an uncertainty set bounded-by-ℓ_p-norm for a generic $p \in [1, \infty]$.

1.5.3 The robust model

Robustifying model (1.6) against the uncertainty set $\mathcal{U}_p(x^{(i)})$ yields the following optimization program:

$$\begin{aligned} \min_{u,\gamma,\xi} \quad & \|u\|_1 + \nu \sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad & y^{(i)} \sum_{j=1}^{m} k(x, x^{(j)}) y^{(j)} u_j \geqslant 1 - \xi_i + y^{(i)}\gamma \quad \forall x \in \mathcal{U}_p(x^{(i)}),\ i = 1, \dots, m \\ & \xi_i \geqslant 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad i = 1, \dots, m. \end{aligned} \tag{1.17}$$

Model (1.17) is intractable due to the infinite possibilities for choosing x in $\mathcal{U}_p(x^{(i)})$. However, a closed-form expression can be derived, as stated in the following theorem.

Theorem 1. Let $\mathcal{U}_p(x^{(i)})$ and $\mathcal{U}_{\mathcal{H}}\big(\phi(x^{(i)})\big)$ be the uncertainty sets in the input and in the feature space as in (1.11) and (1.12), respectively, with $p \in [1, \infty]$. The model (1.17) can be rewritten as:

$$\begin{aligned} \min_{u,\gamma,\xi} \quad & \|u\|_1 + \nu \sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad & y^{(i)} \sum_{j=1}^{m} K_{ij} y^{(j)} u_j - \delta^{(i)} \sum_{j=1}^{m} \sqrt{K_{jj}}\, |u_j| \geqslant 1 - \xi_i + y^{(i)}\gamma \quad i = 1, \dots, m \\ & \xi_i \geqslant 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad i = 1, \dots, m. \end{aligned} \tag{1.18}$$

Proof. The first set of constraints of model (1.17) is equivalent to:

$$\min_{x \in \mathcal{U}_p(x^{(i)})} \quad y^{(i)} \sum_{j=1}^{m} k(x, x^{(j)}) y^{(j)} u_j \geqslant 1 - \xi_i + y^{(i)}\gamma \qquad i = 1, \dots, m. \tag{1.19}$$

Due to the definition of $\mathcal{U}_p(x^{(i)})$, for all $i = 1, \dots, m$ the left-hand side of (1.19) can be re-stated as:

$$\begin{aligned} \min_{\sigma^{(i)}} \quad & y^{(i)} \sum_{j=1}^{m} k(x^{(i)} + \sigma^{(i)}, x^{(j)}) y^{(j)} u_j \\ \text{s.t.} \quad & \|\sigma^{(i)}\|_p \leqslant \eta^{(i)}. \end{aligned}$$

According to the definition of the kernel function and the assumption on $\mathcal{U}_{\mathcal{H}}\big(\phi(x^{(i)})\big)$, we have that:

$$k(x^{(i)} + \sigma^{(i)}, x^{(j)}) = \langle \phi(x^{(i)} + \sigma^{(i)}), \phi(x^{(j)}) \rangle_{\mathcal{H}} = \langle \phi(x^{(i)}) + \zeta^{(i)}, \phi(x^{(j)}) \rangle_{\mathcal{H}}.$$

Moreover, the linearity of the dot product in the feature space $\mathcal{H}$ implies that the model can be written as:

$$\begin{aligned} \min_{\zeta^{(i)}} \quad & y^{(i)} \sum_{j=1}^{m} \langle \zeta^{(i)}, \phi(x^{(j)}) \rangle_{\mathcal{H}} \; y^{(j)} u_j \\ \text{s.t.} \quad & \|\zeta^{(i)}\|_{\mathcal{H}} \leqslant \delta^{(i)}, \end{aligned} \tag{1.20}$$

where the term $y^{(i)} \sum_{j=1}^{m} \langle \phi(x^{(i)}), \phi(x^{(j)}) \rangle_{\mathcal{H}} \; y^{(j)} u_j$ is equivalent to $y^{(i)} \sum_{j=1}^{m} K_{ij} y^{(j)} u_j$. Being independent of $\zeta^{(i)}$, it is moved to the right-hand side of (1.19).

Then, by considering the modulus of the objective function of model (1.20), it can be bounded by $\sum_{j=1}^{m} \left| \langle \zeta^{(i)}, \phi(x^{(j)}) \rangle_{\mathcal{H}} \right| \cdot |u_j|$. By applying the Cauchy-Schwarz inequality in $\mathcal{H}$ and the boundedness condition on $\left\| \zeta^{(i)} \right\|_{\mathcal{H}}$, we get:

$$\left| \langle \zeta^{(i)}, \phi(x^{(j)}) \rangle_{\mathcal{H}} \right| \leqslant \left\| \zeta^{(i)} \right\|_{\mathcal{H}} \cdot \left\| \phi(x^{(j)}) \right\|_{\mathcal{H}} \leqslant \delta^{(i)} \cdot \sqrt{\langle \phi(x^{(j)}), \phi(x^{(j)}) \rangle_{\mathcal{H}}} = \delta^{(i)} \cdot \sqrt{K_{jj}}.$$

The value K_{jj} is nonnegative, due to the positive semidefiniteness of the Gram matrix K ([122]). Therefore, we obtain:

$$\left| y^{(i)} \sum_{j=1}^{m} \langle \zeta^{(i)}, \phi(x^{(j)}) \rangle_{\mathcal{H}} \; y^{(j)} u_j \right| \leqslant \delta^{(i)} \sum_{j=1}^{m} \sqrt{K_{jj}} \, |u_j| \, . \tag{1.21}$$

Thus, the objective value of model (1.20) is $-\delta^{(i)} \sum_{j=1}^{m} \sqrt{K_{jj}} \, |u_j|$, and substituting it in the first set of constraints of (1.17), the thesis follows. □

A similar result is derived in [147], where the robust counterpart of the deterministic model is written as a SOCP. However, in our contribution the robust problem (1.18) is a *Linear Programming* (LP) problem, with clearly advantages from a computational perspective.

When no uncertainty occurs in the data, $\delta^{(i)} = 0$ and model (1.18) reduces to model (1.6).

As in the deterministic case, once u, γ and ξ are obtained as solutions of model (1.18), then $\omega_{\mathcal{A}}$ and $\omega_{\mathcal{B}}$ are computed according to formulas (1.9). Finally, the optimal separating hypersurface $\mathcal{S} = (u, b)$ is derived, where parameter b is the optimal solution of the problem:

$$\begin{aligned} \min_{b} \quad & \sum_{i=1}^{m} \mathbb{1} \left[\left(y^{(i)} b - y^{(i)} \sum_{j=1}^{m} K_{ij} y^{(j)} u_j + \delta^{(i)} \sum_{j=1}^{m} \sqrt{K_{jj}} \, |u_j| \right)_i \right] \\ \text{s.t.} \quad & \gamma + 1 - \omega_{\mathcal{B}} \leqslant b \leqslant \gamma - 1 + \omega_{\mathcal{A}}. \end{aligned} \tag{1.22}$$

1.6 Computational results

In this section, we evaluate the performance of the deterministic model presented in Section 1.4 and its robust counterpart (1.17). The models have been implemented in MATLAB (v. 2021b) and solved using CVX (v. 2.2, see [64, 65]) and MOSEK solver (v. 9.1.9, see [112]). As far as it concerns the linear search problems (1.10) and (1.22), the interval $[\gamma+1-\omega_{\mathcal{B}}, \gamma-1+\omega_{\mathcal{A}}]$ has been split into 10^4 subintervals of equal length. The final solution is then given by the minimum value of all subproblems (see [53]). All computational experiments were run on a MacBookPro17.1 with a chip Apple M1 of 8 cores and 16 GB of RAM memory.

1.6.1 An illustrative example

For the sake of clarity, we start by considering a bidimensional toy example composed by 17 observations (see Figure 1.2). The parameter ν in the objective function of (1.6) and (1.18) has been set to 1, and the classification performed by Gaussian RBF kernel with $\alpha = 1.9$.

We illustrate in Figure 1.2 the results of the deterministic approach. The optimal classifier S is represented by a solid line, whereas hypersurfaces $S_{\mathcal{A}}$ and $S_{\mathcal{B}}$ are depicted as dotted and dashed line, respectively. The support vectors are drawn as stars. According to the nonlinear version of properties **(P1)**-**(P2)**, all the black points of class $\mathcal{A}$ lie inside the curve defined by $S_{\mathcal{A}}$, and all the white points of class $\mathcal{B}$ are outside $S_{\mathcal{B}}$. The optimal classifier S satisfies property **(P3)** since it is comprised in the region between $S_{\mathcal{A}}$ and $S_{\mathcal{B}}$, and minimizes the total number of misclassified points.

We depict in Figure 1.3 the kernel-induced decision boundaries of the robust model, considering the same toy example as before. The bound $\eta^{(i)}$ on the perturbation in the input space is set to 0.01 and 0.2 for data points in class $\mathcal{A}$ and $\mathcal{B}$, respectively. The model is trained for both spherical ($p = 2$, see Figure 1.3a) and box ($p = \infty$, see Figure 1.3b) uncertainty sets $\mathcal{U}_p(x^{(i)})$. Compared to Figure 1.2, the separating curves $S_{\mathcal{A}}$ and $S_{\mathcal{B}}$ still satisfy properties **(P1)**-**(P2)**, as well as S with property **(P3)**, but there are no support vectors since each data point is corrupted by uncertainties.

1.6.2 Real-world datasets

In order to test the performance of the proposed methodology on real-world data, we perform classification experiments on a selection of datasets taken from the UCI Machine Learning Repository (see [76]). The datasets are listed in the first column of Table 1.3,

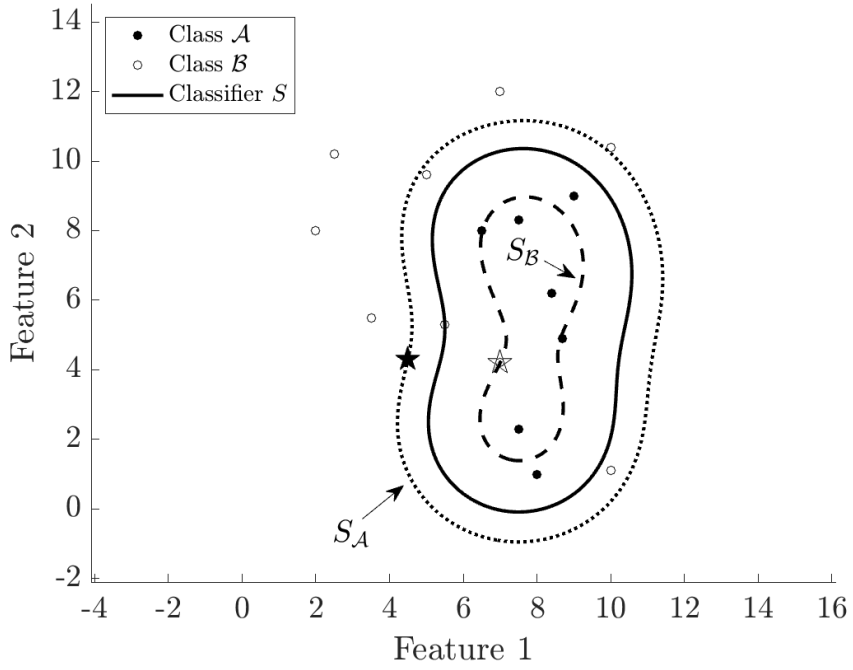
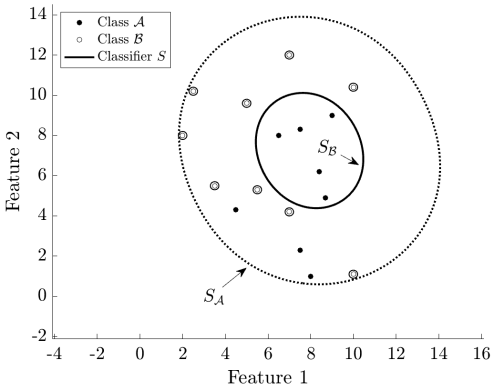
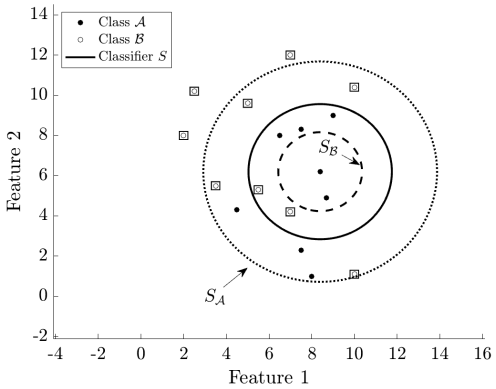


Figure 1.2: Separating hypersurfaces obtained with Gaussian RBF kernel ($\alpha = 1.9$) from the deterministic model. Support vectors are depicted as stars.



(a) Gaussian RBF kernel ($\alpha = 1.9$), $p = 2$



(b) Gaussian RBF kernel ($\alpha = 1.9$), $p = \infty$

Figure 1.3: Separating hypersurfaces obtained with Gaussian RBF kernel ($\alpha = 1.9$) from the robust model. The ℓ_p-norms defining the uncertainty set are $p = 2$ (on the left) and $p = \infty$ (on the right).

along with the corresponding number of features n and of observations m. For datasets with more than two classes we adopt the *one-versus-all* scheme, finding the optimal classifier separating the first class of points from the remaining ones.

Each dataset is split into two disjoint parts: the *training set*, composed by the β% of the observations, and the *testing set*, composed by the remaining $(1-\beta)$%. We account for three different values of β, leading to the following holdouts: 75%-25%, 50%-50%, and 25%-

75%. The partition is performed inline with the *proportional random sampling* strategy (see [34]), meaning that the original class balance in the entire dataset is maintained in both the training and testing set. Once the partition is complete, a kernel function $k(\cdot,\cdot)$ is chosen and the training set is used to train the deterministic classifier for different values of input parameter ν. Specifically, the deterministic formulation is solved on five logarithmically spaced values of ν between 10^{-3} and 10^{0}. The optimal classifier is chosen among the five candidates as the one minimizing the misclassification error on the training set. Finally, the out-of-sample misclassification error on the testing set is computed, as the ratio between the total number of misclassified points in the testing set and its cardinality. This procedure is repeated 96 times, parallelizing the code on the 8 cores of the working machine, in a *repeated holdout* fashion (see [79]). The results are then averaged.

As far as it concerns the kernel function $k(\cdot,\cdot)$, we test seven different alternatives: homogeneous linear ($d = 1$, $c = 0$), homogeneous quadratic ($d = 2$, $c = 0$), homogeneous cubic ($d = 3$, $c = 0$); inhomogeneous linear, inhomogeneous quadratic, inhomogeneous cubic; Gaussian RBF. The parameter α in the Gaussian RBF kernel is set as the maximum value of the standard deviation across features for the dataset under consideration. Similarly for the parameter c in the inhomogeneous polynomial kernels.

Potentially, the range of values in the datasets may vary widely across features, with different orders of magnitude. Since model (1.6) and its robust counterpart (1.18) are distance-based, this may result in giving high weights to specific attributes when classifying. For this reason, we apply pre-processing techniques of data transformation before training the models. Among all the possibilities we consider *min-max normalization* and *standardization*. For an overview on data pre-processing methods, the reader is referred to [68]. On one hand, in the min-max normalization the training dataset is linearly scaled feature-wise into the n-dimensional hypercube $[0,1]^n$, according to the formula:

$$x_j^{(i)\prime} := \frac{x_j^{(i)} - \min\limits_{l=1,\dots,m} x_j^{(l)}}{\max\limits_{l=1,\dots,m} x_j^{(l)} - \min\limits_{l=1,\dots,m} x_j^{(l)}} \qquad i = 1,\dots,m, \quad j = 1,\dots,n, \tag{1.23}$$

where $x_j^{(i)\prime}$ is the j-th transformed feature of observation i. On the other hand, in the standardization the values of a specific feature j are normalized based on its mean μ_j and standard deviation std_j, namely:

$$x_j^{(i)\prime} := \frac{x_j^{(i)} - \mu_j}{std_j} \qquad i = 1,\dots,m, \quad j = 1,\dots,n. \tag{1.24}$$

Among all the optimal deterministic classifiers found for each couple *data transformation-kernel function*, the best configuration is chosen as the one minimizing the overall misclassification error. Within this choice of *data transformation-kernel function*, the robust model is solved. The bounds $\eta^{(i)}$ on the perturbation vectors defining the uncertainty sets $\mathcal{U}_p(x^{(i)})$ are adjusted as:

$$\eta^{(i)} = \eta_{\mathcal{A}} := \rho_{\mathcal{A}} \max_{j=1,\dots,n} std_{j,\mathcal{A}} \qquad \forall i : x^{(i)} \in \mathcal{A}$$
$$\eta^{(i)} = \eta_{\mathcal{B}} := \rho_{\mathcal{B}} \max_{j=1,\dots,n} std_{j,\mathcal{B}} \qquad \forall i : x^{(i)} \in \mathcal{B},$$

where $\rho_{\mathcal{A}}$ is a nonnegative parameter allowing the user to tailor the degree of conservatism and $\max_{j=1,\dots,n} std_{j,\mathcal{A}}$ is the maximum standard deviation feature-wise for training points of class $\mathcal{A}$. Similarly for $\rho_{\mathcal{B}}$ and $\max_{j=1,\dots,n} std_{j,\mathcal{B}}$. For simplicity, we set $\rho_{\mathcal{A}} = \rho_{\mathcal{B}} = \rho$, and consider 7 logarithmically spaced values between 10^{-7} and 10^{-1}. As in the deterministic case, we average the out-of-sample testing errors for 96 random partitions of the dataset.

For each dataset, we report in Table 1.3 the best configuration *data transformation-kernel function*, along with the average out-of-sample testing errors and standard deviations for the deterministic and robust models. We consider the three main types of uncertainty set in the literature, defined respectively by ℓ_1-, ℓ_2- and ℓ_∞-norm. The listed results refer to the holdout 75% training set-25% testing set. Details are reported in the Appendix A.2 respectively in Tables A.1-A.3 for the deterministic model, and in Tables A.4-A.9 for the robust model.

We notice that all the considered robust formulations outperform the corresponding deterministic result. Specifically, in 4 out of 9 datasets the best results are achieved by the box robust formulation ($p = \infty$), followed by the ellipsoidal ($p = 2$, in 3 out of 9) and finally by the polyhedral ($p = 1$). Since box uncertainty sets are the most wide around data among the three, this implies that the proposed formulation benefits from a more conservative approach when treating uncertainties.

For the sake of completeness, we explore in details the performance of the proposed models when applied to the dataset "Parkinson". First of all, we discuss the results of the deterministic approach, with respect to both data transformation and kernel function. The out-of-sample testing errors for the holdout 75%-25% are depicted in Figure 1.4, while detailed results are reported in Table A.1 in the Appendix A.2. We note that the worst performances occur when no data transformations are applied (see the dash-dotted line in Figure 1.4). Conversely, min-max normalization (1.23) and standardization (1.24) provide good and comparable results: the best performance is achieved by the linear kernel on min-

Dataset $m \times n$	Data transformation	Kernel	Deterministic	Robust $p = 1$	 $p = 2$	 $p = \infty$
Arrhythmia 68×279	–	Gaussian RBF	$20.47\% \pm 0.07$	$\mathbf{19.12\% \pm 0.08}$	$19.30\% \pm 0.07$	$19.61\% \pm 0.07$
CPU time (s)			0.289	0.290	0.288	0.295
Parkinson 195×22	Min-max normalization	Hom. linear	$13.19\% \pm 0.03$	$12.98\% \pm 0.03$	$\mathbf{12.37\% \pm 0.03}$	$12.61\% \pm 0.04$
CPU time (s)			3.626	3.421	3.454	3.418
Heart Disease 297×13	Standardization	Inhom. linear	$17.48\% \pm 0.04$	$16.84\% \pm 0.04$	$17.53\% \pm 0.03$	$\mathbf{16.36\% \pm 0.04}$
CPU time (s)			12.253	11.602	11.477	11.417
Dermatology 358×34	–	Inhom. quadratic	$1.64\% \pm 0.02$	$1.65\% \pm 0.01$	$1.57\% \pm 0.01$	$\mathbf{0.55\% \pm 0.01}$
CPU time (s)			20.173	20.055	20.420	20.147
Climate Model Crashes 540×18	–	Hom. linear	$5.01\% \pm 0.02$	$4.47\% \pm 0.02$	$4.50\% \pm 0.01$	$\mathbf{4.34\% \pm 0.01}$
CPU time (s)			68.069	66.762	67.169	67.381
Breast Cancer Diagnostic 569×30	Min-max normalization	Inhom. quadratic	$3.02\% \pm 0.02$	$2.63\% \pm 0.01$	$2.65\% \pm 0.01$	$\mathbf{2.56\% \pm 0.01}$
CPU time (s)			77.786	77.968	78.267	77.543
Breast Cancer 683×9	Standardization	Hom. linear	$3.17\% \pm 0.01$	$\mathbf{2.97\% \pm 0.01}$	$3.07\% \pm 0.01$	$3.06\% \pm 0.01$
CPU time (s)			135.765	135.651	137.039	136.286
Blood Transfusion 748×4	Standardization	Inhom. cubic	$20.72\% \pm 0.02$	$20.60\% \pm 0.02$	$\mathbf{20.55\% \pm 0.02}$	$20.64\% \pm 0.02$
CPU time (s)			178.136	178.751	179.682	180.083
Mammographic Mass 830×5	Standardization	Inhom. quadratic	$15.71\% \pm 0.02$	$15.49\% \pm 0.02$	$\mathbf{15.42\% \pm 0.02}$	$15.54\% \pm 0.02$
CPU time (s)			241.205	241.810	242.614	241.929

Table 1.3: Average out-of-sample testing errors and standard deviations over 96 runs for the deterministic and robust models. Best results are highlighted. Holdout: 75% training set-25% testing set.

max normalized data (13.19%). Similar conclusions can be drawn for holdouts 50%-50% and 25%-75%, where in those cases the homogeneous quadratic kernel outperforms the others, still in the case of min-max normalized data (see Tables A.2-A.3 in the Appendix A.2).

In order to evaluate the performance of the robust model, we consider 60 logarithmically spaced values of ρ between 10^{-7} and 10^{-1}. The results are depicted in Figure 1.5. We notice that the increase of the value of β leads to better performances when considering the overall out-of-sample testing error (see Figure 1.5a), since more data points in the training set are available as input of the optimization model. In addition, when perturbations are included in the model, the performances improve with respect to the deterministic case. Indeed, the great majority of the points lies below the corresponding horizontal line, representing the out-of-sample testing error of the deterministic classifier. Interestingly, the increase of the uncertainty impacts differently on the two classes (see Figure 1.5b). It can be noted that points of class $\mathcal{A}$ benefit from including high perturbations in the model. On the contrary, points of class $\mathcal{B}$ are worsen classified when the level of corruption is high.

In addition, we compare the performance of our models with the results reported in

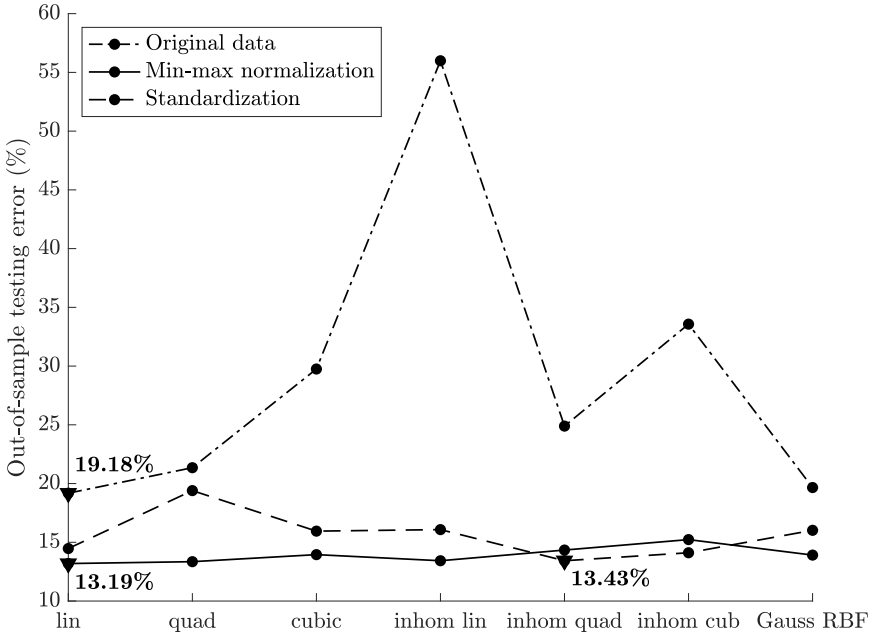
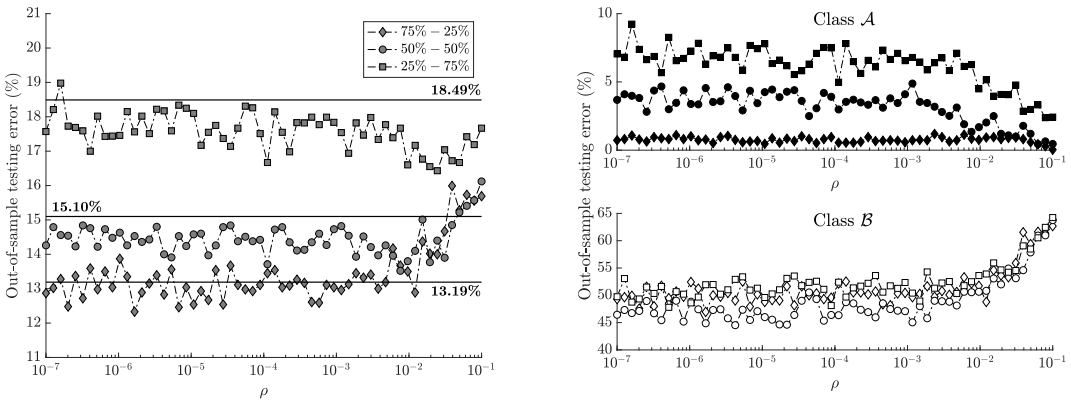


Figure 1.4: Out-of-sample testing error of the deterministic formulation applied to the dataset "Parkinson". Each triangle represents the lowest error for the corresponding data transformation technique. Holdout: 75% training set-25% testing set.



(a) Overall results.

(b) Results divided by class.

Figure 1.5: Out-of-sample testing error of the robust formulation applied to the dataset "Parkinson". Overall results are on the left, with the performance of the deterministic classifier depicted as horizontal line for each holdout. Results divided by class are on the right. The values of ρ are in logarithmic scale.

[53] and [14]. As shown in Table 1.4, in 6 out of 9 datasets the results of our deterministic classifier outperform the other methods. Consequently, the linear approach presented in [90] benefits from a generalization towards nonlinear classifier. Moreover, within the same 6 datasets, our robust formulation leads to even better accuracy, implying that it is

meaningful to consider uncertainties in the proposed SVM-type model.

Dataset	Deterministic			Robust		
	Table 1.3	[53]	[14]	Table 1.3	[53]	[14]
Arrhythmia	20.47%	25.65%	43.08%	**19.12%**	23.00%	29.23%
Parkinson	13.19%	14.13%	14.36%	**12.37%**	13.00%	16.41%
Heart Disease	17.48%	16.68%	**15.93%**	16.36%	16.20%	16.61%
Dermatology	1.64%	0.56%	3.38%	0.55%	**0.13%**	1.13%
Climate Model Crashes	5.01%	4.99%	5.00%	4.34%	4.34%	**4.07%**
Breast Cancer Diagnostic	3.02%	4.89%	6.49%	**2.56%**	3.89%	4.04%
Breast Cancer	3.17%	3.49%	5.00%	**2.97%**	3.12%	4.26%
Blood Transfusion	20.72%	23.49%	23.62%	**20.55%**	22.55%	23.62%
Mammographic Mass	15.71%	–	18.07%	**15.42%**	–	19.28%

Table 1.4: Out-of-sample testing error comparison among deterministic and robust results of Table 1.3, data from [53] and [14]. For each approach and dataset, the best result is underlined. The lowest out-of-sample testing error within a dataset is in bold.

From Table 1.3 it can be noticed that the choice of the best data transformation method strongly depends on the dataset. In order to guide the final user among the three possible techniques, we report in Table A.10 in the Appendix A.2 summary statistics on the 9 datasets. Specifically, for each feature we compute the mean and the corresponding coefficient of variation, defined as the ratio between the standard deviation and the mean. In Table A.10 we list the minimum and the maximum values of the two considered indices for each dataset, along with the corresponding best data transformation. We argue that, whenever the values of the observations are close, and so the minimum and the maximum too, the best approach is to classify the original data without any transformation (see datasets "Arrhythmia", "Dermatology" and "Climate Model Crashes"). In the extreme case of the presence of some constant features, i.e., the minimum and the maximum values coincide, and thus the coefficient of variation is zero, formulas (1.23)-(1.24) cannot be applied since the denominator is equal to zero. This situation occurs with the dataset "Arrhythmia", where only original data can be classified. On the other hand, the min-max normalization is a suitable choice when the order of magnitude across the features varies a lot. For instance, in datasets "Parkinson" and "Breast cancer diagnostic" there are 7 and 5 orders of magnitude of difference between the minimum and the maximum value of the mean of the features. Finally, standardization is an appropriate method in all other cases, where no significant differences occur among the orders of magnitude of the features (see datasets "Heart Disease", "Breast Cancer", "Blood Transfusion" and "Mammographic Mass").

Furthermore, numerical results show that the computational time is significantly high

for datasets with a large number of observations, especially when considering 75% of the instances as training set (see Table A.1 in the Appendix A.2). The performing speed benefits from a reduction of β, even if at the cost of worsening the accuracy. Nevertheless, when datasets are equally split in training and testing set, the accuracy does not decrease significantly compared to the case 75%-25% (see Table A.2). A similar conclusion is valid for the robust model (see Tables A.4-A.7).

1.7 Conclusions

In this chapter, we have proposed a new optimization model for solving a binary classification task through SVM. From a methodological perspective, we have extended the technique studied in [90] to the nonlinear context through the introduction of a kernel function. Data are mapped from the input space to a higher-dimensional space and a final linear search procedure aiming to minimize the overall misclassification error is considered. Motivated by the uncertain nature of real-world data, we have adopted a RO approach by constructing around each input data an uncertainty set bounded-by-ℓ_p-norm, with $p \in [1, \infty]$. Perturbation propagates from the input space to the feature space through the kernel function. Therefore, we have derived closed-form expressions for the uncertainty sets in the feature space, extending the results present in the literature. Finally, we have derived the robust counterpart of the deterministic model in the case of nonlinear classifier. Both the deterministic and the robust formulation reduce to LP problem, with clear advantages in terms of computational efficiency. The proposed models have been tested on real-world datasets, considering different combinations of data transformations and kernel functions. The results outperform other linear SVM approaches in most cases, even in the deterministic framework. Overall, the model benefits from including uncertainty during the training process. The accuracy is affected by the choice of the kernel function and of the data transformation before training. Insights to guide the user in choosing the best configuration are finally drawn.

Future works will focus on handling uncertainties in the labels of input data. It would also be interesting to extend the proposed robust approach to other SVM-type models and in the case of multiclass classification. Finally, devising distributionally robust formulation with different classes of ambiguity sets merits further research too.

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". This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

Chapter 2

A Robust Twin Parametric Margin Support Vector Machine for Multiclass Classification

Authors: Renato De Leone[1], Francesca Maggioni[2], Andrea Spinelli[3].

Keywords: Machine Learning; Support Vector Machine; Robust Optimization; Multiclass Classification.

This chapter is under evaluation in *Computers & Operations Research*.
Manuscript Reference Number: CAOR-D-24-00314.

[1]School of Science and Technology, University of Camerino, Via Madonna delle Carceri 9, Camerino 62032, Italy. renato.deleone@unicam.it
[2]Department of Management, Information and Production Engineering, University of Bergamo, Viale G. Marconi 5, Dalmine 24044, Italy. francesca.maggioni@unibg.it
[3]Department of Management, Information and Production Engineering, University of Bergamo, Viale G. Marconi 5, Dalmine 24044, Italy. andrea.spinelli@unibg.it

2.1 Introduction

Binary pattern classification is one of the most studied *Machine Learning* (ML) tasks thanks to its wide variety of application. Despite deep learning is currently the state-of-the-art paradigm in binary classification, it does not guarantee a strong predictive accuracy when applied to tabular data ([66]). For this reason, the design of new ML approaches for classifying such a kind of data is an important ongoing research topic ([101]).

In this chapter, we consider and extend the *Twin Parametric Margin Support Vector Machine* (TPMSVM) presented in [116]. The reader is referred to Section 1.1 for an introduction on *Support Vector Machine* (SVM), one of the best-known ML tools for classification. Rather than dealing with a single classifier, in the TPMSVM two nonparallel classifiers are detected, one for each class, such that training observations of the other class are as far as possible from the opposite classifier. Computational experiments show that the TPMSVM achieves better predictive accuracy when compared to other SVM-type techniques. In addition, its computational complexity is reduced since each of the two classifiers is the solution of a small-sized optimization model.

SVM was originally designed to solve binary classification problems. However, in many applications classifying categories might be more than two ([30, 105]), leading to the design of *ad hoc* methods in the ML literature. Typically, such a kind of problems are decomposed into a sequence of binary classification tasks, whose solutions are finally pieced together in an aggregate decision function ([70]). Depending on how the decomposition and the following reconstruction are performed, different approaches have been proposed ([45]). Compared to binary pattern recognition, multiclass classification problems are more challenging and less explored in the literature ([120]).

To cope with uncertainty arising during the measurement of real-world observations, *Robust Optimization* (RO, [9]) techniques have been designed, preventing the model against the worst possible realization of the uncertain parameter (see Section 1.1 for an overview on RO). The application of RO methods usually translates to superior predictive performance of the classification process ([53, 98, 102]). Therefore, it is important to design novel RO models for improving the accuracy of ML procedures.

In this chapter, we propose a novel TPMSVM multiclass classification model under uncertainty. The main contributions of the chapter can be summarized as follows:

- To extend the binary TPMSVM approach to the context of multiclass classification with linear and nonlinear classifiers;
- To formulate robust counterparts of the deterministic models with bounded-by-ℓ_p-

norm uncertainty sets;

- To provide computational experiments based on real-world datasets to test the performance of the models, showing the advantages of explicitly considering uncertainty in the proposed formulation.

The remainder of the chapter is organized as follows. Section 2.2 reviews the existing literature on the problem. Section 2.3 introduces basic facts on binary TPMSVM model. In Section 2.4 the deterministic multiclass model is designed, while in Section 2.5 the robust counterpart is presented. Section 2.6 reports computational results to evaluate the accuracy of the proposed formulations. Finally, in Section 2.7 conclusions and future works are discussed.

2.2 Literature review

Starting from the seminal work of Vapnik and Chervonenkis ([150]) where SVM has been introduced for the first time, several alternative formulations have been proposed in the literature. In the following, we focus our attention on methods related to TPMSVM. The reader is referred to Section 1.2 for a general introduction to classical SVM techniques.

The TPMSVM ([116]) can be seen as a combination of the *parametric-ν-margin* model (par-ν-SVM, [69]) and of the *TWin Support Vector Machine* (TWSVM, [71]). The par-ν-SVM is based on the *ν-Support Vector Classification* (ν-SVC, [130]) where a positive parameter ν in the objective function bounds the fractions of supporting vectors and misclassification errors. With respect to ν-SVC, the par-ν-SVM approach is able to deal with heteroscedastic noise. On the other hand, the TWSVM considers two nonparallel classifiers as solutions of two small-sized SVM-problems. Consequently, the computational complexity of TWSVM is much reduced compared with the classical SVM. Due to its favourable performance, especially when handling large datasets, many variants of the TWSVM approach have been devised in the ML literature: *Least Squares TWSVM* (LS-TWSVM, [5]), *Projection TWSVM* (P-TWSVM, [35]), *Twin Parametric Margin SVM* (TPMSVM, [116]), *Pinball loss TWSVM* (Pin-TWSVM, [160]), *New Fuzzy TWSVM* (NFTWSVM, [33]). For a comprehensive overview on recent developments on TWSVM the reader is referred to [143].

As stated above, the TPMSVM is a variant of the TWSVM. Specifically, it aims at generating two nonparallel classifiers, each of them determining the positive or negative parametric margin of the separating classifier. Therefore, it 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, [118]), *Least Squares TPMSVM* (LSTPMSVM, [132]), *Smooth TPMSVM* (STPMSVM, [155]), *Centroid-based TPMSVM* (CTPSVM, [117]), *Truncated Pinball Loss TPMSVM* (TPin-TSVM, [151]).

All the approaches discussed so far consider the case of binary classification. To tackle the problem of multiclass classification, two main techniques have been formulated in the literature: *all-together* methods and *decomposition-reconstruction* methods ([70]). On the one hand, all training data points are considered at the same time in one large optimization model and the classifier is derived accordingly ([23, 161, 165]). On the other hand, the multiclass classification problem is decomposed into a sequence of binary classification tasks. Each subproblem is solved independently and the binary classifiers are finally combined into an aggregate multiclass decision function. Nowadays, decomposition-reconstruction methods are considered to be as the most effective to achieve multiclass separation ([47]), especially due to the high computational complexity of the all-together methods with large datasets ([45]).

Different formulations have been designed within the decomposition-reconstruction paradigm. In the *one-versus-all* strategy ([151, 157]), a classifier for each class is constructed such that it separates the data points inside the class from the samples outside the class. In the *one-versus-one* approach ([89]), only pairs of classes are considered, leading to an increased number of binary classifiers. In contrast, in the *one-versus-one-versus-rest* strategy ([4, 159]) all training samples are considered in constructing the classification rule. Indeed, each subproblem focuses on the separation of a pair of classes together with all the remaining samples by means of two hyperplanes. Each hyperplane is close to a class and as far as possible from the other, with all the remaining points restricted in a region between the two hyperplanes ([47]). Other decomposition-reconstruction approaches in the literature are *direct acyclic graph* ([123]), *all-versus-one* (MBSVM, [162]) and *binary tree SVM structure* (DTTSVM, [133]). A review on multiclass models specifically designed for TWSVM can be found in [45].

For the methods mentioned above, all data points are implicitly assumed to be known exactly. RO techniques prevent worsening the quality of the solution in the case of uncertainty in the training samples ([14, 158]). In the context of robust TWSVM, a *Robust Minimum Class Variance* model (RMCV-TWSVM) is proposed in [119]. Specifically, a pair of uncertain class variance matrices is considered with uncertainty sets defined according to the Frobenius norm. In [124] two nonparallel classifiers are proposed in the case of

ellipsoidal uncertainty sets (R-TWSVM). The corresponding model is then reformulated as a *Second Order Cone Programming* (SOCP) problem. Instead of convex hulls to represent the training patterns, [93] and [100] consider ellipsoids defined by the first two moments of the class distributions (RNPSVM and Twin SOCP-SVM, respectively). The robust problem is then formulated as a *Chance-Constrained* (CC) programming model ([134]) and the robust counterpart reduces to a SOCP formulation. The same CC approach has been applied in [91] for the case of twin multiclass SVM (Twin-KSOCP). Recently, in [129] an improved version of RNPSVM (called IRNPSVM) is proposed to reduce the number of missing data through a CC approach. Within the multiclass framework, in [165] a robust classification through piecewise-linear functions is derived, robustifying the approach of [23] in the case of ellipsoidal uncertainty set.

All the approaches discussed so far on the TWSVM and its variants are schematically reported in Figure 2.1 and listed in Table 2.1.

In this chapter, we present a novel TPMSVM-type robust model for multiclass classification. We consider both the cases of linear and kernel-induced decision boundaries. Given the uncertain nature of real-world observations, we derive robust counterparts of the nominal models by considering a general bounded-by-ℓ_p-norm uncertainty set around each input data. To assess the accuracy, we test the deterministic and robust methodologies on publicly available datasets. To the best of our knowledge, this is the first time that a robust multiclass formulation based on the TPMSVM is proposed.

2.3 Prior work

The methods that are relevant for our proposal, namely the linear TPMSVM (Section 2.3.1) and the nonlinear TPMSVM (Section 2.3.2) for binary classification are presented in this section.

2.3.1 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 i-th data point, denoting the class to which it belongs. We assume that each of the two categories is composed by 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 indices sets.

The TPMSVM approach considers two nonparallel hyperplanes H_+ and H_-, defined

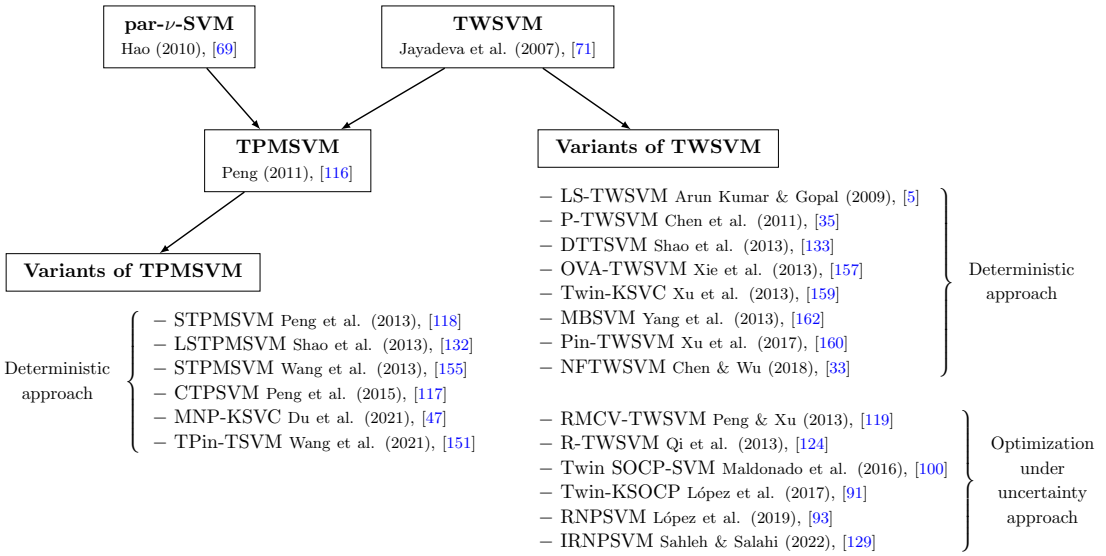


Figure 2.1: Scheme of the selected TWSVM literature review. The models are distinguished in deterministic and optimization under uncertainty approaches.

		Jayadeva et al. (2007), [71]	Arun Kumar & Gopal (2009), [5]	Chen et al. (2011), [35]	Peng (2011), [116]	Peng and Xu (2013), [119]	Peng et al. (2013), [118]	Qi et al. (2013), [124]	Shao et al. (2013), [132]	Shao et al. (2013), [133]	Wang et al. (2013), [155]	Xie et al. (2013), [157]	Xu et al. (2013), [159]	Yang et al. (2013), [162]	Peng et al. (2015), [117]	Maldonado et al. (2016), [100]	Maldonado et al. (2017), [91]	Xu et al. (2017), [160]	Chen & Wu (2018), [33]	Lopez et al. (2019), [93]	Du et al. (2021), [47]	Wang et al. (2021), [151]	Sahleh & Salahi (2022), [129]
TWSVM	Linear classifier	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Nonlinear classifier	✓	✓		✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Classification	Binary	✓	✓	✓	✓	✓	✓	✓	✓		✓				✓	✓		✓	✓	✓		✓	✓
	Multiclass									✓		✓	✓	✓			✓			✓	✓		
Optimization under uncertainty approach	Ellipsoidal RO							✓															
	Matrix RO					✓																	
	Chance-Constraints															✓	✓			✓			✓

Table 2.1: A selected TWSVM literature review. In the first row of the table the contributions are listed in chronological order. In the second and third rows, linear and nonlinear TWSVM classifiers are considered. Rows four and five deal with binary and multiclass classification. Finally, the optimization under uncertainty methodologies are explored in rows six to eight.

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 the solutions of a pair of *Quadratic Programming Problems* (QPPs):

$$\begin{aligned} \min_{w_+,\theta_+,\xi_+} \quad & \frac{1}{2}\|w_+\|_2^2 + \frac{\nu_+}{m_-} e_{m_-}^\top \left(X_-^\top w_+ + e_{m_-}\theta_+\right) + \frac{\alpha_+}{m_+} e_{m_+}^\top \xi_+ \\ \text{s.t.} \quad & X_+^\top w_+ + e_{m_+}\theta_+ \geqslant -\xi_+ \\ & \xi_+ \geqslant 0, \end{aligned} \tag{2.1}$$

and

$$\begin{aligned} \min_{w_-,\theta_-,\xi_-} \quad & \frac{1}{2}\|w_-\|_2^2 - \frac{\nu_-}{m_+} e_{m_+}^\top \left(X_+^\top w_- + e_{m_+}\theta_-\right) + \frac{\alpha_-}{m_-} e_{m_-}^\top \xi_- \\ \text{s.t.} \quad & X_-^\top w_- + e_{m_-}\theta_- \leqslant \xi_- \\ & \xi_- \geqslant 0, \end{aligned} \tag{2.2}$$

where $\nu_+, \nu_- > 0$, $\alpha_+, \alpha_- > 0$ are regularization parameters, balancing the terms in the objective functions, and $\xi_+ \in \mathbb{R}^{m_+}$, $\xi_- \in \mathbb{R}^{m_-}$ are slack vectors, associated with misclassified samples in each class ([38]).

The objective function of model (2.1) is composed by three parts. The first term is related to the margin for the positive class. The second term considers the projections of negative observations on H_+, requiring that the negative training points are as far as possible from H_+. Finally, the third term is the penalty function regarding the total number of misclassified positive samples. Similar observations can be made for the objective function of model (2.2).

As in [69] and [130], the ratios ν_+/α_+ and ν_-/α_- control the fractions of supporting vectors and margin errors in each class. Hence, ν_+ and ν_- cannot be greater than α_+ and α_-, respectively.

The dual models of (2.1) and (2.2) are the following QPPs, respectively:

$$\begin{aligned} \max_{\lambda_+} \quad & -\frac{1}{2}\lambda_+^\top X_+^\top X_+ \lambda_+ + \frac{\nu_+}{m_-} e_{m_-}^\top X_-^\top X_+ \lambda_+ \\ \text{s.t.} \quad & e_{m_+}^\top \lambda_+ = \nu_+ \\ & 0 \leqslant \lambda_+ \leqslant \frac{\alpha_+}{m_+}, \end{aligned} \tag{2.3}$$

and

$$\begin{aligned} \max_{\lambda_-} \quad & -\frac{1}{2}\lambda_-^\top X_-^\top X_- \lambda_- + \frac{\nu_-}{m_+} e_{m_+}^\top X_+^\top X_- \lambda_- \\ \text{s.t.} \quad & e_{m_-}^\top \lambda_- = \nu_- \\ & 0 \leqslant \lambda_- \leqslant \frac{\alpha_-}{m_-}, \end{aligned} \tag{2.4}$$

where $\lambda_+ \in \mathbb{R}^{m_+}$ and $\lambda_- \in \mathbb{R}^{m_-}$ are the Lagrangian multiplier vectors for each class. Once (2.3) and (2.4) are solved, by using the *Karush-Kuhn-Tucker* (KKT) conditions the optimal parameters (w_+, θ_+) and (w_-, θ_-) are computed as:

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

and

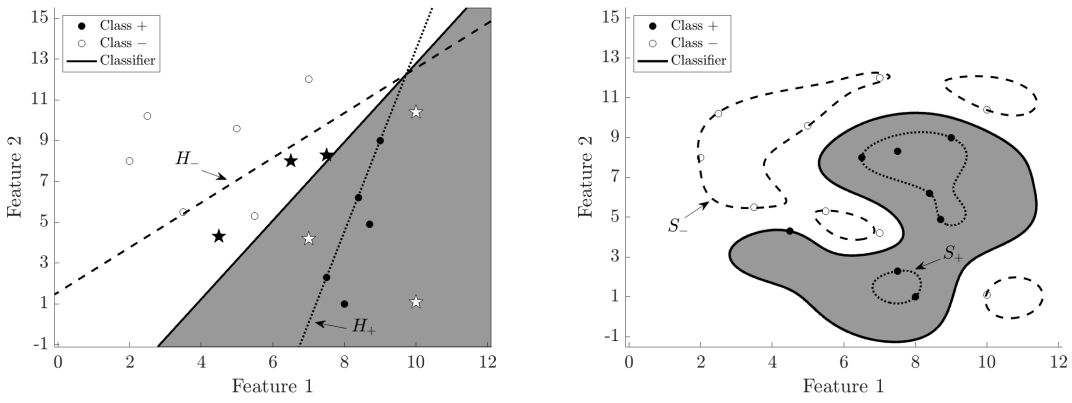
$$w_- = \frac{\nu_-}{m_+} X_+ e_{m_+} - X_-\lambda_- \qquad \theta_- = -\frac{1}{|\mathcal{N}_-|} \sum_{i \in \mathcal{N}_-} {x^{(i)}}^\top w_-,$$

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

Finally, a new observation $x \in \mathbb{R}^n$ is classified as negative or positive according to the following decision function:

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

In Figure 2.2a we depict the hyperplanes H_- and H_+, along with the classifier $f_{\text{lin}} = 0$ for a binary classification task with two features.



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

Figure 2.2: Linear and nonlinear classifiers for the case of binary TPMSVM. The parameters are $\nu_+ = \nu_- = 0.5$, $\alpha_+ = \alpha_- = 1$. In the nonlinear case, the Gaussian kernel with $\sigma = 1.5$ is considered. Misclassified points for each class are represented as stars.

2.3.2 The binary TPMSVM for nonlinear classification

To increase the predictive power of the model, allowing situations where training observations are not linearly separable, in [116] the nonlinear version of the TPMSVM approach is provided. According to the classical procedure of [38], input data points are mapped to a inner product space $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$ via a feature map $\phi : \mathbb{R}^n \to \mathcal{H}$.

Following this idea, the separating hyperplanes $\widetilde{H}_+$ and $\widetilde{H}_-$ are now defined in the feature space $\mathcal{H}$ as:

$$\widetilde{H}_+ : \langle \widetilde{w}_+, \phi(x)\rangle_{\mathcal{H}} + \theta_+ = 0 \qquad \widetilde{H}_- : \langle \widetilde{w}_-, \phi(x)\rangle_{\mathcal{H}} + \theta_- = 0,$$

with $\widetilde{w}_+, \widetilde{w}_- \in \mathcal{H}$ and $\theta_+, \theta_- \in \mathbb{R}$. Accordingly, models (2.1) and (2.2) are modified as:

$$\begin{aligned} \min_{\widetilde{w}_+,\theta_+,\xi_+} \quad & \frac{1}{2}\|\widetilde{w}_+\|^2_{\mathcal{H}} + \frac{\nu_+}{m_-}\sum_{i\in\mathcal{X}_-}\left(\langle \widetilde{w}_+, \phi(x^{(i)})\rangle_{\mathcal{H}} + \theta_+\right) + \frac{\alpha_+}{m_+}e_{m_+}^\top \xi_+ \\ \text{s.t.} \quad & \langle \widetilde{w}_+, \phi(x^{(i)})\rangle_{\mathcal{H}} + \theta_+ \geqslant -\xi_{+,i} \qquad i \in \mathcal{X}_+ \\ & \xi_+ \geqslant 0, \end{aligned} \tag{2.5}$$

and

$$\begin{aligned} \min_{\widetilde{w}_-,\theta_-,\xi_-} \quad & \frac{1}{2}\|\widetilde{w}_-\|^2_{\mathcal{H}} - \frac{\nu_-}{m_+}\sum_{i\in\mathcal{X}_+}\left(\langle \widetilde{w}_-, \phi(x^{(i)})\rangle_{\mathcal{H}} + \theta_-\right) + \frac{\alpha_-}{m_-}e_{m_-}^\top \xi_- \\ \text{s.t.} \quad & \langle \widetilde{w}_-, \phi(x^{(i)})\rangle_{\mathcal{H}} + \theta_- \leqslant \xi_{-,i} \qquad i \in \mathcal{X}_- \\ & \xi_- \geqslant 0, \end{aligned} \tag{2.6}$$

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

As mentioned in Chapter 1, a closed-form expression of the feature map $\phi(\cdot)$ is rarely available, and therefore models (2.5)-(2.6) are not solvable in practice ([73]). Nevertheless, it is possible to reformulate their duals by applying the so-called *kernel trick* ([38]). Indeed, a symmetric and positive semidefinite kernel function $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is introduced such that $k(x, x') := \langle \phi(x), \phi(x')\rangle_{\mathcal{H}}$, for all $x, x' \in \mathbb{R}^n$. Examples of kernels typically used in the ML literature are reported in Table 2.2. The reader is referred to [131] for a comprehensive overview on kernel functions.

Thus, the dual problems of models (2.5) and (2.6) can be reformulated in terms of $k(\cdot,\cdot)$ as:

$$\begin{aligned} \max_{\lambda_+} \quad & -\frac{1}{2}\lambda_+^\top K(X_+, X_+)\lambda_+ + \frac{\nu_+}{m_-}e_{m_-}^\top K(X_-, X_+)\lambda_+ \\ \text{s.t.} \quad & e_{m_+}^\top \lambda_+ = \nu_+ \\ & 0 \leqslant \lambda_+ \leqslant \frac{\alpha_+}{m_+}, \end{aligned} \tag{2.7}$$

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

Table 2.2: Examples of kernel functions. The first column reports the name of the kernel functions. The second column provides their mathematical expressions. Finally, the third column contains the related relevant parameters.

and

$$\begin{aligned} \max_{\lambda_-} \quad & -\frac{1}{2}\lambda_-^\top K(X_-, X_-)\lambda_- + \frac{\nu_-}{m_+} e_{m_+}^\top K(X_+, X_-)\lambda_- \\ \text{s.t.} \quad & e_{m_-}^\top \lambda_- = \nu_- \\ & 0 \leqslant \lambda_- \leqslant \frac{\alpha_-}{m_-}, \end{aligned} \tag{2.8}$$

where $K(X_+, X_+)$ is the 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 (2.7) and (2.8) are solved, the KKT conditions provide $(\widetilde{w}_+, \theta_+)$ and $(\widetilde{w}_-, \theta_-)$. Hyperplanes $\widetilde{H}_+$ and $\widetilde{H}_-$ in the feature space $\mathcal{H}$ correspond to kernel-induced 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:

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

In Figure 2.2b we depict the separating hypersurfaces S_+ and S_-, along with the classifier $f_{\text{nonlin}} = 0$, under a Gaussian kernel.

2.4 A novel multiclass TPMSVM-type model

In this section, we extend the binary TPMSVM approach to the case of multiclass classification both for linear (Section 2.4.1) and nonlinear (Section 2.4.2) decision boundaries. Among all the possible formulations, we adopt the *one-versus-all* strategy thanks to its reduced computational complexity and good accuracy ([45]).

In the following, we assume that C is the total number of classifying categories and, for each class $c = 1, \ldots, C$, the subscript $\cdot_{-c}$ will refer to points not in class c.

2.4.1 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 $X_c \in \mathbb{R}^{n \times m_c}$ represents all training data points of class c. Similarly, for matrix $X_{-c} \in \mathbb{R}^{n \times m_{-c}}$. The corresponding indices sets are $\mathcal{X}_c$ and $\mathcal{X}_{-c}$, respectively, and $\mathcal{X} := \mathcal{X}_c \cup \mathcal{X}_{-c}$.

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 the solutions of the following QPP:

$$\begin{aligned} \min_{w_c, \theta_c, \xi_c} \quad & \frac{1}{2} \|w_c\|_2^2 + \frac{\nu_c}{m_{-c}} e_{m_{-c}}^\top \left(X_{-c}^\top w_c + e_{m_{-c}} \theta_c \right) + \frac{\alpha_c}{m_c} e_{m_c}^\top \xi_c \\ \text{s.t.} \quad & X_c^\top w_c + e_{m_c} \theta_c \geqslant -\xi_c \\ & \xi_c \geqslant 0. \end{aligned} \tag{2.9}$$

Parameters $\nu_c > 0$, $\alpha_c > 0$ and slack vector $\xi_c \in \mathbb{R}^{m_c}$ have an equivalent interpretation of the corresponding ones in model (2.1).

By introducing the Lagrangian function of problem (2.9), the dual model is given by:

$$\begin{aligned} \max_{\lambda_c} \quad & -\frac{1}{2} \lambda_c^\top X_c^\top X_c \lambda_c + \frac{\nu_c}{m_{-c}} e_{m_{-c}}^\top X_{-c}^\top X_c \lambda_c \\ \text{s.t.} \quad & e_{m_c}^\top \lambda_c = \nu_c \\ & 0 \leqslant \lambda_c \leqslant \frac{\alpha_c}{m_c}, \end{aligned} \tag{2.10}$$

with optimal solutions derived according to the KKT conditions:

$$w_c = X_c \lambda_c - \frac{\nu_c}{m_{-c}} X_{-c} e_{m_{-c}} \qquad \theta_c = -\frac{1}{|\mathcal{N}_c|} \sum_{i \in \mathcal{N}_c} {x^{(i)}}^\top w_c,$$

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

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

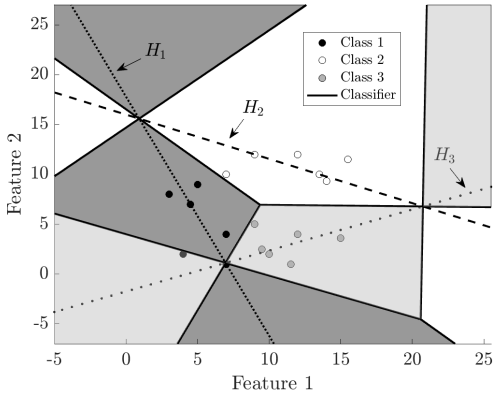
$$f_{\text{lin,min}}(x) := \arg\min_{c=1,\dots,C} \frac{\left| w_c^\top x + \theta_c \right|}{\|w_c\|_2} \tag{2.11}$$

and

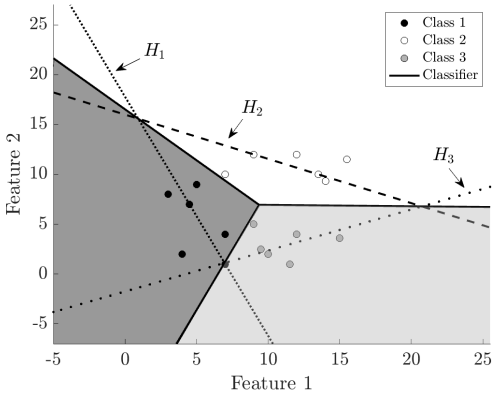
$$f_{\text{lin,max}}(x) := \arg\max_{c=1,\dots,C} \frac{w_c^\top x + \theta_c}{\|w_c\|_2}. \tag{2.12}$$

In Figures 2.3a-2.3b we consider a bidimensional toy example with three classes. We perform classification with both decision function (2.11) and (2.12). Hyperplanes H_1, H_2,

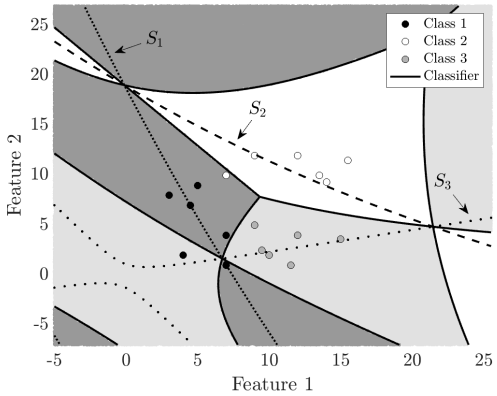
H_3 are depicted, along with the decision functions. Each of the three regions in black, white and grey corresponds to one of the three classes, according to equation (2.11) or (2.12).



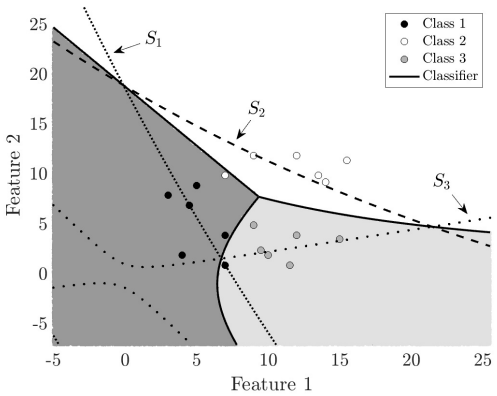
(a) Linear multiclass TPMSVM with argmin formula (2.11)



(b) Linear multiclass TPMSVM with argmax formula (2.12)



(c) Nonlinear multiclass TPMSVM with argmin formula (2.16)



(d) Nonlinear multiclass TPMSVM with argmax formula (2.17)

Figure 2.3: Linear and nonlinear classifiers for the case of three-classes TPMSVM. The parameters are $\nu_c = 0.5$, $\alpha_c = 1$ for $c = 1, 2, 3$. In the nonlinear case, the inhomogeneous polynomial kernel with $d = 2$ and $\gamma = 1.5$ is considered.

2.4.2 The multiclass TPMSVM for nonlinear classification

When dealing with nonlinear classifiers, in the feature space $\mathcal{H}$ model (2.9) becomes:

$$\begin{aligned} \min_{\widetilde{w}_c, \theta_c, \xi_c} \quad & \frac{1}{2} \|\widetilde{w}_c\|_{\mathcal{H}}^2 + \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} (\langle \widetilde{w}_c, \phi(x^{(i)}) \rangle_{\mathcal{H}} + \theta_c) + \frac{\alpha_c}{m_c} \sum_{i \in \mathcal{X}_c} \xi_{c,i} \\ \text{s.t.} \quad & \langle \widetilde{w}_c, \phi(x^{(i)}) \rangle_{\mathcal{H}} + \theta_c \geqslant -\xi_{c,i} \qquad i \in \mathcal{X}_c \\ & \xi_{c,i} \geqslant 0 \qquad i \in \mathcal{X}_c, \end{aligned} \tag{2.13}$$

which is intractable due to the non-availability of $\phi(\cdot)$. However, the kernel trick applied to model (2.10) leads to:

$$\begin{aligned} \max_{\lambda_c} \quad & -\frac{1}{2}\lambda_c^\top K(X_c, X_c)\lambda_c + \frac{\nu_c}{m_{-c}} e_{m_{-c}}^\top K(X_{-c}, X_c)\lambda_c \\ \text{s.t.} \quad & e_{m_c}^\top \lambda_c = \nu_c \\ & 0 \leqslant \lambda_c \leqslant \frac{\alpha_c}{m_c}, \end{aligned} \tag{2.14}$$

where $K(X_c, X_c)$ is the $m_c \times m_c$ matrix with entries $k(x^{(i)}, x^{(j)})$ for $i, j \in \mathcal{X}_c$. Similarly with $K(X_{-c}, X_c)$. The KKT conditions provide the optimal solutions $(\widetilde{w}_c, \theta_c)$ in terms of λ_c, namely:

$$\widetilde{w}_c = \sum_{i \in \mathcal{X}_c} \lambda_{c,i}\phi(x^{(i)}) - \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \phi(x^{(i)}) \qquad \theta_c = -\frac{1}{|\mathcal{N}_c|} \sum_{i \in \mathcal{N}_c} \langle \phi(x^{(i)}), \widetilde{w}_c \rangle_{\mathcal{H}}. \tag{2.15}$$

Within this case, the proposed decision functions are:

$$f_{\text{nonlin,min}}(x) := \arg\min_{c=1,\dots,C} \frac{|\langle \widetilde{w}_c, \phi(x) \rangle_{\mathcal{H}} + \theta_c|}{\|\widetilde{w}_c\|_{\mathcal{H}}} \tag{2.16}$$

and

$$f_{\text{nonlin,max}}(x) := \arg\max_{c=1,\dots,C} \frac{\langle \widetilde{w}_c, \phi(x) \rangle_{\mathcal{H}} + \theta_c}{\|\widetilde{w}_c\|_{\mathcal{H}}}, \tag{2.17}$$

where, for all $c = 1, \dots, C$,

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

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

and

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

In Figures 2.3c-2.3d we depict the results of model (2.14) when considering the same toy dataset of Figures 2.3a-2.3b and an inhomogeneous quadratic kernel.

2.5 The robust model

In this section, we derive the robust counterparts of models (2.9) and (2.13) by constructing proper uncertainty sets around data points. The deterministic approaches are then robustified by optimizing over worst-case realizations of the uncertain data in the uncertainty sets. Tractable reformulations are finally provided.

Section 2.5.1 considers the robust multiclass TPMSVM for linear classification, while Section 2.5.2 explores the robust kernel-induced decision boundaries.

2.5.1 The robust TPMSVM for linear multiclass classification

As in Section 1.5.1, we assume that each observation $x^{(i)} \in \mathbb{R}^n$ is subject to an unknown but bounded by ℓ_p-norm perburbation $\delta^{(i)} \in \mathbb{R}^n$, with $p \in [1, \infty]$. Specifically, the uncertainty set around $x^{(i)}$ has the following expression:

$$\mathcal{U}_p(x^{(i)}) := \left\{x \in \mathbb{R}^n | x = x^{(i)} + \delta^{(i)}, \left\|\delta^{(i)}\right\|_p \leqslant \varepsilon^{(i)}\right\}. \tag{2.18}$$

The value of the radius $\varepsilon^{(i)} \geqslant 0$ controls the degree of conservatism: when $\varepsilon^{(i)} = 0$, the uncertainty set $\mathcal{U}_p(x^{(i)})$ reduces to observation $x^{(i)}$.

Robustifying model (2.9) against the uncertainty set $\mathcal{U}_p(x^{(i)})$ yields the following optimization model:

$$\begin{aligned} \min_{w_c, \theta_c, \xi_c} \quad & \frac{1}{2}\|w_c\|_2^2 + \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \max_{\|\delta^{(i)}\|_p \leqslant \varepsilon^{(i)}} \left[\left(x^{(i)^\top} + \delta^{(i)^\top}\right) w_c + \theta_c\right] + \frac{\alpha_c}{m_c} \sum_{i \in \mathcal{X}_c} \xi_{c,i} \\ \text{s.t.} \quad & x^{(i)^\top} w_c + \theta_c \geqslant -\xi_{c,i} \qquad i \in \mathcal{X}_c, \forall x \in \mathcal{U}_p(x^{(i)}) \\ & \xi_{c,i} \geqslant 0 \qquad i \in \mathcal{X}_c. \end{aligned} \tag{2.19}$$

Since there exists infinite possibilities for choosing $x \in \mathcal{U}_p(x^{(i)})$, model (2.19) is intractable. In the following theorem, a tractable closed-form expression is derived.

Theorem 2. Let $\mathcal{U}_p(x^{(i)})$ be the uncertainty set as in (2.18), with $p \in [1, \infty]$. Let p' be the Hölder conjugate of p, namely $1/p + 1/p' = 1$. The robust counterpart of model (2.9) is:

$$\begin{aligned} \min_{w_c, \theta_c, \xi_c} \quad & \frac{1}{2}\|w_c\|_2^2 + \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \left(x^{(i)^\top} w_c + \varepsilon^{(i)} \|w_c\|_{p'}\right) + \nu_c \theta_c + \frac{\alpha_c}{m_c} \sum_{i \in \mathcal{X}_c} \xi_{c,i} \\ \text{s.t.} \quad & x^{(i)^\top} w_c + \theta_c - \varepsilon^{(i)} \|w_c\|_{p'} \geqslant -\xi_{c,i} \qquad i \in \mathcal{X}_c \\ & \xi_{c,i} \geqslant 0 \qquad i \in \mathcal{X}_c. \end{aligned} \tag{2.20}$$

Proof. Model (2.19) can be expressed as ([14]):

$$\begin{aligned}
\min_{w_c,\theta_c,\xi_c} \quad & \frac{1}{2}\|w_c\|_2^2 + \frac{\nu_c}{m_{-c}} \sum_{i\in\mathcal{X}_{-c}} \max_{\|\delta^{(i)}\|_p \leqslant \varepsilon^{(i)}} \left[\left(x^{(i)^\top} + \delta^{(i)^\top}\right) w_c + \theta_c \right] + \frac{\alpha_c}{m_c} \sum_{i\in\mathcal{X}_c} \xi_{c,i} \\
\text{s.t.} \quad & \min_{\|\delta^{(i)}\|_p \leqslant \varepsilon^{(i)}} \left[\left(x^{(i)^\top} + \delta^{(i)^\top}\right) w_c \right] + \theta_c \geqslant -\xi_{c,i} \qquad i \in \mathcal{X}_c \\
& \xi_{c,i} \geqslant 0 \qquad i \in \mathcal{X}_c.
\end{aligned} \tag{2.21}$$

The maximization term in the objective function corresponds to:

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

By definition of the dual norm ([128]), we get:

$$\max_{\|\delta^{(i)}\|_p \leqslant \varepsilon^{(i)}} \delta^{(i)^\top} w_c = \varepsilon^{(i)} \|w_c\|_{p'},$$

where p' is the Hölder conjugates of p. This implies that:

$$-\varepsilon^{(i)} \|w_c\|_{p'} \leqslant \delta^{(i)^\top} w_c \leqslant \varepsilon^{(i)} \|w_c\|_{p'}. \tag{2.22}$$

Consequently, the second term in the objective function of (2.21) corresponds to:

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

As far as it concerns the first set of constraints in (2.21), for all $i \in \mathcal{X}_c$ we have that:

$$\min_{\|\delta^{(i)}\|_p \leqslant \varepsilon^{(i)}} \left[\delta^{(i)^\top} w_c \right] \geqslant -\xi_{c,i} - \theta_c - x^{(i)^\top} w_c.$$

By considering the first inequality in (2.22), the previous minimization problem can be solved as:

$$x^{(i)^\top} w_c + \theta_c - \varepsilon^{(i)} \|w_c\|_{p'} \geqslant -\xi_{c,i}.$$

This concludes the proof. □

If no uncertainty occurs, $\varepsilon^{(i)} = 0$ for all $i \in \mathcal{X}$ and thus the robust model (2.20) reduces to the deterministic model (2.9). We notice that model (2.20) is a convex nonlinear optimization model due to the presence of the ℓ_2- and $\ell_{p'}$-norm of w_c. 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, $v_c \in \mathbb{R}$ ([124]), leading to:

$$\begin{aligned}
\min_{w_c,\theta_c,\xi_c,t_c,u_c,v_c} \quad & \frac{1}{2}(u_c - v_c) + \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \left({x^{(i)}}^\top w_c + \varepsilon^{(i)} \left\| w_c \right\|_{p'} \right) + \nu_c \theta_c + \frac{\alpha_c}{m_c} \sum_{i \in \mathcal{X}_c} \xi_{c,i} \\
\text{s.t.} \quad & {x^{(i)}}^\top w_c + \theta_c - \varepsilon^{(i)} \left\| w_c \right\|_{p'} \geqslant -\xi_{c,i} \qquad i \in \mathcal{X}_c \\
& t_c \geqslant \left\| w_c \right\|_2 \\
& u_c + v_c = 1 \\
& u_c \geqslant \sqrt{t_c^2 + v_c^2} \\
& \xi_{c,i} \geqslant 0 \qquad i \in \mathcal{X}_c.
\end{aligned} \tag{2.23}$$

In the cases of polyhedral ($p = 1$), spherical ($p = 2$) and box ($p = \infty$) uncertainty sets, model (2.23) reduces to a SOCP problem, as stated in the following result.

Corollary 1. Let $\mathcal{U}_p(x^{(i)})$ be the uncertainty set as in (2.18). Model (2.23) can be expressed as a SOCP problem in the following cases:

a) Case $p = 1$:

$$\begin{aligned}
\min_{w_c,\theta_c,\xi_c,t_c,u_c,v_c,s_c} \quad & \frac{1}{2}(u_c - v_c) + \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \left({x^{(i)}}^\top w_c + \varepsilon^{(i)} s_c \right) + \nu_c \theta_c + \frac{\alpha_c}{m_c} \sum_{i \in \mathcal{X}_c} \xi_{c,i} \\
\text{s.t.} \quad & {x^{(i)}}^\top w_c + \theta_c - \varepsilon^{(i)} s_c \geqslant -\xi_{c,i} \qquad i \in \mathcal{X}_c \\
& t_c \geqslant \left\| w_c \right\|_2 \\
& u_c + v_c = 1 \\
& u_c \geqslant \sqrt{t_c^2 + v_c^2} \\
& s_c \geqslant -w_{c,j} \qquad j = 1, \ldots, n \\
& s_c \geqslant w_{c,j} \qquad j = 1, \ldots, n \\
& s_c \geqslant 0 \\
& \xi_{c,i} \geqslant 0 \qquad i \in \mathcal{X}_c.
\end{aligned} \tag{2.24}$$

b) Case $p = 2$:

$$\begin{aligned}
\min_{w_c,\theta_c,\xi_c,t_c,u_c,v_c} \quad & \frac{1}{2}(u_c - v_c) + \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \left({x^{(i)}}^\top w_c + \varepsilon^{(i)} t_c \right) + \nu_c \theta_c + \frac{\alpha_c}{m_c} \sum_{i \in \mathcal{X}_c} \xi_{c,i} \\
\text{s.t.} \quad & {x^{(i)}}^\top w_c + \theta_c - \varepsilon^{(i)} t_c \geqslant -\xi_{c,i} \qquad i \in \mathcal{X}_c \\
& t_c \geqslant \left\| w_c \right\|_2 \\
& u_c + v_c = 1 \\
& u_c \geqslant \sqrt{t_c^2 + v_c^2} \\
& \xi_{c,i} \geqslant 0 \qquad i \in \mathcal{X}_c.
\end{aligned} \tag{2.25}$$

c) Case $p = \infty$:

$$\begin{aligned}
\min_{w_c,\theta_c,\xi_c,t_c,u_c,v_c,s_c} \quad & \frac{1}{2}(u_c - v_c) + \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \left(x^{(i)^\top} w_c + \varepsilon^{(i)} \sum_{j=1}^{n} s_{c,j} \right) + \nu_c \theta_c + \frac{\alpha_c}{m_c} \sum_{i \in \mathcal{X}_c} \xi_{c,i} \\
\text{s.t.} \quad & x^{(i)^\top} w_c + \theta_c - \varepsilon^{(i)} \sum_{j=1}^{n} s_{c,j} \geqslant -\xi_{c,i} \qquad i \in \mathcal{X}_c \\
& t_c \geqslant \|w_c\|_2 \\
& u_c + v_c = 1 \\
& u_c \geqslant \sqrt{t_c^2 + v_c^2} \\
& s_{c,j} \geqslant -w_{c,j} \qquad j = 1, \ldots, n \\
& s_{c,j} \geqslant w_{c,j} \qquad j = 1, \ldots, n \\
& s_{c,j} \geqslant 0 \qquad j = 1, \ldots, n \\
& \xi_{c,i} \geqslant 0 \qquad i \in \mathcal{X}_c.
\end{aligned} \tag{2.26}$$

Proof. a) If $p = 1$, then $p' = \infty$. By introducing an auxiliary variable $s_c \geqslant 0$ equal to $\|w_c\|_\infty$, and adding the constraints $s_c \geqslant -w_{c,j}$ and $s_c \geqslant w_{c,j}$ for all $j = 1, \ldots, n$, model (2.23) is equivalent to model (2.24).

b) If $p = 2$, then $p' = 2$, and so $\|w_c\|_{p'} = \|w_c\|_2 = t_c$. The equivalence between models (2.23) and (2.25) follows straightforwardly.

c) If $p = \infty$, then $p' = 1$. In this case model (2.23) can be rewritten as model (2.26) by introducing an auxiliary vector $s_c \in \mathbb{R}^n$ such that each component $s_{c,j}$ is equal to $|w_{c,j}|$ and adding the constraints $s_{c,j} \geqslant 0$, $s_{c,j} \geqslant -w_{c,j}$ and $s_{c,j} \geqslant w_{c,j}$ for all $j = 1, \ldots, n$.

□

For a general analysis of the case of $p \in (1, +\infty)$, the reader is referred to [20].

As in the deterministic case, once the optimal solutions (w_c, θ_c) are obtained for all $c = 1, \ldots, C$, the classification of a new observation is performed according to decision functions (2.11)-(2.12).

2.5.2 The robust TPMSVM for nonlinear multiclass classification

Similarly to Chapter 1 and [147], we model the uncertainty set in the feature space as follows:

$$\mathcal{U}_{\mathcal{H}}\left(\phi(x^{(i)})\right) := \left\{ z \in \mathcal{H} | z = \phi(x^{(i)}) + \widetilde{\delta}^{(i)}, \left\| \widetilde{\delta}^{(i)} \right\|_{\mathcal{H}} \leqslant \widetilde{\varepsilon}^{(i)} \right\}, \tag{2.27}$$

where the perturbation $\widetilde{\delta}^{(i)}$ belongs to $\mathcal{H}$ and its $\mathcal{H}$-norm is bounded by $\widetilde{\varepsilon}^{(i)} \geqslant 0$. The value of the constant $\widetilde{\varepsilon}_i$ may be unknown but depends on the bound ε_i for the corresponding uncertainty set $\mathcal{U}_p(x^{(i)})$ in the input space. Moreover, perturbation $\widetilde{\delta}^{(i)}$ arises in the feature space if and only if perturbation $\delta^{(i)}$ in the input space occurs. Thus, $\varepsilon^{(i)} = 0$ implies $\widetilde{\varepsilon}^{(i)} = 0$. Closed-form expressions of $\widetilde{\varepsilon}^{(i)}$ for kernel functions typically used in the ML literature are derived in Chapter 1.

We start by robustifying model (2.13) over the uncertainty set (2.27), obtaining the following optimization model:

$$\begin{aligned}
\min_{\widetilde{w}_c,\theta_c,\xi_c} \quad & \frac{1}{2}\|\widetilde{w}_c\|_{\mathcal{H}}^2 + \frac{\nu_c}{m_{-c}} \sum_{i\in\mathcal{X}_{-c}} \max_{\|\widetilde{\delta}^{(i)}\|_{\mathcal{H}} \leqslant \widetilde{\varepsilon}^{(i)}} \left[\langle \widetilde{w}_c, \phi(x^{(i)}) + \widetilde{\delta}^{(i)} \rangle_{\mathcal{H}} + \theta_c\right] + \frac{\alpha_c}{m_c} \sum_{i\in\mathcal{X}_c} \xi_{c,i} \\
\text{s.t.} \quad & \langle \widetilde{w}_c, z \rangle_{\mathcal{H}} + \theta_c \geqslant -\xi_{c,i} \qquad i \in \mathcal{X}_c, \forall z \in \mathcal{U}_{\mathcal{H}}\big(\phi(x^{(i)})\big) \\
& \xi_{c,i} \geqslant 0 \qquad i \in \mathcal{X}_c.
\end{aligned} \tag{2.28}$$

As in (2.19), model (2.28) is intractable. However, the following theorem holds.

Theorem 3. Let $\mathcal{U}_{\mathcal{H}}\big(\phi(x^{(i)})\big)$ be the uncertainty set as in (2.27). The robust counterpart of model (2.13) is:

$$\begin{aligned}
\min_{\beta_c,\theta_c,\xi_c} \quad & \frac{1}{2}\beta_c^\top K \beta_c + \frac{\nu_c}{m_{-c}} \sum_{i\in\mathcal{X}_{-c}} \left[\widetilde{\varepsilon}^{(i)} \sqrt{\beta_c^\top K \beta_c} + \sum_{j\in\mathcal{X}} \beta_{c,j} k(x^{(i)}, x^{(j)})\right] + \nu_c \theta_c + \frac{\alpha_c}{m_c} \sum_{i\in\mathcal{X}_c} \xi_{c,i} \\
\text{s.t.} \quad & \theta_c - \widetilde{\varepsilon}^{(i)} \sqrt{\beta_c^\top K \beta_c} + \sum_{j\in\mathcal{X}} \beta_{c,j} k(x^{(i)}, x^{(j)}) \geqslant -\xi_{c,i} \qquad i \in \mathcal{X}_c \\
& \beta_{c,i} = -\frac{\nu_c}{m_{-c}} \qquad i \in \mathcal{X}_{-c} \\
& \xi_{c,i} \geqslant 0 \qquad i \in \mathcal{X}_c.
\end{aligned} \tag{2.29}$$

Proof. Model (2.28) is equivalent to:

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

We notice that:

$$\max_{\|\widetilde{\delta}^{(i)}\|_{\mathcal{H}} \leqslant \widetilde{\varepsilon}^{(i)}} \left[\langle \widetilde{w}_c, \phi(x^{(i)}) + \widetilde{\delta}^{(i)} \rangle_{\mathcal{H}} + \theta_c\right] = \theta_c + \langle \widetilde{w}_c, \phi(x^{(i)}) \rangle_{\mathcal{H}} + \max_{\|\widetilde{\delta}^{(i)}\|_{\mathcal{H}} \leqslant \widetilde{\varepsilon}^{(i)}} \left[\langle \widetilde{w}_c, \widetilde{\delta}^{(i)} \rangle_{\mathcal{H}}\right].$$

Similarly it can be argued 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 (2.27), it holds that:

$$\begin{aligned}\left|\langle \widetilde{w}_c, \widetilde{\delta}^{(i)} \rangle_{\mathcal{H}}\right| &\leqslant \|\widetilde{w}_c\|_{\mathcal{H}} \left\|\widetilde{\delta}^{(i)}\right\|_{\mathcal{H}} \\ &\leqslant \|\widetilde{w}_c\|_{\mathcal{H}} \, \widetilde{\varepsilon}^{(i)}.\end{aligned}$$

Therefore, the solutions of the maximization and minimization problems are respectively:

$$\max_{\|\widetilde{\delta}^{(i)}\|_{\mathcal{H}} \leqslant \widetilde{\varepsilon}^{(i)}} \left[\langle \widetilde{w}_c, \phi(x^{(i)}) + \widetilde{\delta}^{(i)} \rangle_{\mathcal{H}} + \theta_c\right] = \theta_c + \langle \widetilde{w}_c, \phi(x^{(i)}) \rangle_{\mathcal{H}} + \widetilde{\varepsilon}^{(i)} \, \|\widetilde{w}_c\|_{\mathcal{H}}$$

and

$$\min_{\|\widetilde{\delta}^{(i)}\|_{\mathcal{H}} \leqslant \widetilde{\varepsilon}^{(i)}} \left[\langle \widetilde{w}_c, \phi(x^{(i)}) + \widetilde{\delta}^{(i)} \rangle_{\mathcal{H}} + \theta_c\right] = \theta_c + \langle \widetilde{w}_c, \phi(x^{(i)}) \rangle_{\mathcal{H}} - \widetilde{\varepsilon}^{(i)} \, \|\widetilde{w}_c\|_{\mathcal{H}} \, .$$

Consequently, model (2.28) is equivalent to:

$$\begin{aligned}\min_{\widetilde{w}_c, \theta_c, \xi_c} \quad & \frac{1}{2} \|\widetilde{w}_c\|_{\mathcal{H}}^2 + \frac{\nu_c}{m_{-c}} \sum_{i \in \mathcal{X}_{-c}} \left[\widetilde{\varepsilon}^{(i)} \, \|\widetilde{w}_c\|_{\mathcal{H}} + \langle \widetilde{w}_c, \phi(x^{(i)}) \rangle_{\mathcal{H}}\right] + \nu_c \theta_c + \frac{\alpha_c}{m_c} \sum_{i \in \mathcal{X}_c} \xi_{c,i} \\ \text{s.t.} \quad & \theta_c - \widetilde{\varepsilon}^{(i)} \, \|\widetilde{w}_c\|_{\mathcal{H}} + \langle \widetilde{w}_c, \phi(x^{(i)}) \rangle_{\mathcal{H}} \geqslant -\xi_{c,i} \qquad i \in \mathcal{X}_c \\ & \xi_{c,i} \geqslant 0 \qquad i \in \mathcal{X}_c.\end{aligned} \tag{2.30}$$

In the feature space, $\widetilde{w}_c$ can be decomposed as a linear combination of mapped input data by means of $\phi(\cdot)$. Specifically, by considering (2.15), it holds that:

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

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 $\widetilde{w}$ corresponds to:

$$\|\widetilde{w}_c\|_{\mathcal{H}}^2 = \langle \widetilde{w}_c, \widetilde{w}_c \rangle_{\mathcal{H}} = \sum_{i,j \in \mathcal{X}} \beta_{c,i} k(x^{(i)}, x^{(j)}) \beta_{c,j} = \beta_c^\top K \beta_c, \tag{2.31}$$

where K is the kernel matrix, i.e. $K_{ij} = K_{ji} = k(x^{(i)}, x^{(j)})$, with $i, j \in \mathcal{X}$, and the dot product is:

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

Thus, model (2.30) can be rewritten as:

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

This concludes the proof. □

Model (2.29) can be easily transformed in the following SOCP problem by introducing variables t_c, u_c, $v_c \in \mathbb{R}$:

$$\begin{aligned}
\min_{\beta_c,\theta_c,\xi_c,t_c,u_c,v_c} \quad & \frac{1}{2}(u_c - v_c) + \frac{\nu_c}{m_{-c}}\sum_{i\in\mathcal{X}_{-c}}\left[\widetilde{\varepsilon}^{(i)}t_c + \sum_{j\in\mathcal{X}}\beta_{c,j}k(x^{(i)},x^{(j)})\right] + \nu_c\theta_c + \frac{\alpha_c}{m_c}\sum_{i\in\mathcal{X}_c}\xi_{c,i} \\
\text{s.t.} \quad & \theta_c - \widetilde{\varepsilon}^{(i)}t_c + \sum_{j\in\mathcal{X}}\beta_{c,j}k(x^{(i)},x^{(j)}) \geqslant -\xi_{c,i} \qquad i\in\mathcal{X}_c \\
& \beta_{c,i} = -\frac{\nu_c}{m_{-c}} \qquad i\in\mathcal{X}_{-c} \\
& t_c \geqslant \sqrt{\beta_c^\top K\beta_c} \\
& u_c + v_c = 1 \\
& u_c \geqslant \sqrt{t_c^2 + v_c^2} \\
& \xi_{c,i} \geqslant 0 \qquad i\in\mathcal{X}_c.
\end{aligned} \tag{2.32}$$

Once β_c and θ_c are found, the classification task of a new observation $x \in \mathbb{R}^n$ is performed according to decision functions (2.16) and (2.17), where $\|\widetilde{w}_c\|_{\mathcal{H}}$ is computed as in (2.31) and the dot product is:

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

2.6 Experimental results

In this section, we investigate the performance of the proposed TPMSVM approaches for a multiclass classification task.

All models are implemented in MATLAB (version 2021b) and numerical results are obtained using CVX ([64, 65]) with the solver MOSEK (version 9.1.9, [112]). Computa-

tional experiments are run on a MacBookPro17.1 with a chip Apple M1 of 8 cores and 16 GB of RAM memory.

The section is structured as follows. A description of the benchmark datasets and the experimental setting are provided in Section 2.6.1. The performance of the deterministic approach is presented in Section 2.6.2. Finally, the results for the robust models are reported in Section 2.6.3.

2.6.1 Datasets and experimental setting

We consider three public-domain real-world multiclass datasets from the *UCI Machine Learning* repository (UCI, [76]) and from *Open Data Canada* (ODC, [115]). A description of the datasets can be found in Table 2.3. The first three columns report the dataset name, the source and the application field. The size m of the dataset and the number n of features are in the fourth and in the fifth columns, respectively. Finally, the last column reports the number of classifying categories.

Dataset	Source	Application field	Observations	Features	Classes
Iris	UCI	Life Sciences	150	4	3
Wine	UCI	Physical Sciences	178	13	3
Fuel Consumption Ratings	ODC	Transport	374	7	3

Table 2.3: Summary statistics of considered datasets.

In order to evaluate the performance of the proposed methodology, each dataset is randomly split into training set and testing set, with a proportion of 75%-25% of the total number of observations. The partition is performed according to the *proportional random sampling* strategy ([34]), implying that the original class balance in the entire dataset is maintained both in the training and in the testing set. In order to avoid imbalances among the orders of magnitude of the features, before the training phase each dataset is linearly scaled into the unit interval $[0, 1]$.

As far as it concerns hyperparameters in models (2.10) and (2.14), for simplicity we set $\nu_c = \nu$ and $\alpha_c = \alpha$ for all $c = 1, \dots, C$, and a grid search procedure ([164]) is applied to tune their values. Specifically, α is selected from the set $\{2^j | j = -6, -5, \dots, 5, 6\}$, whereas the value of ν/α is chosen from the set $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. In addition, the parameters γ for the inhomogeneous polynomial kernels and σ for the Gaussian kernel take value in $\{2^j | j = -4, -3, \dots, 3, 4\}$.

The best configuration of parameters is selected as the one maximizing the accuracy of the model on the training set. Finally, the model is tested on the testing set and the

corresponding accuracy is computed. In order to get stable results, for each hold-out 75%-25% the computational experiments are performed over 50 different combinations of training and testing set, and the results are then averaged.

2.6.2 Results for the deterministic TPMSVM models

Tables 2.4-2.5 report the results of deterministic models (2.9) and (2.14) in terms of percentage mean accuracy and standard deviation. For the kernel-induced decision classifiers, we consider five polynomial kernels (homogeneous quadratic and cubic; inhomogeneous linear, quadratic and cubic) and the Gaussian kernel. The CPU time for training the model is outlined below each result.

Deterministic model - argmin decision function								
Dataset		Kernel						
		Linear	Hom. quadratic	Hom. cubic	Inhom. linear	Inhom. quadratic	Inhom. cubic	Gaussian
Iris	Accuracy	91.78 ± 4.05	76.70 ± 4.69	85.95 ± 5.43	$\underline{\mathbf{92.22 \pm 4.11}}$	91.57 ± 3.85	88.27 ± 4.93	90.76 ± 4.79
	CPU time (s)	0.92	5.68	5.60	36.06	54.34	54.83	113.63
Wine	Accuracy	96.91 ± 2.28	96.23 ± 2.70	94.91 ± 2.74	$\underline{\mathbf{96.95 \pm 2.58}}$	96.18 ± 2.22	96.18 ± 2.57	39.45 ± 1.10
	CPU time (s)	0.84	18.42	18.49	91.77	167.29	169.93	353.39
Fuel	Accuracy	55.55 ± 4.77	50.75 ± 5.35	51.87 ± 5.27	54.90 ± 4.36	51.98 ± 5.46	54.15 ± 5.42	$\underline{\mathbf{71.91 \pm 4.19}}$
	CPU time (s)	1.07	27.95	28.63	109.94	262.62	263.74	753.60

Table 2.4: Detailed percentage results of average accuracy and standard deviation over 50 runs of the deterministic model. Classification is performed according to the argmin decision functions (2.11) and (2.16). The best result for the kernelized model (2.14) is underlined. Overall, the best result is in bold.

Deterministic model - argmax decision function								
Dataset		Kernel						
		Linear	Hom. quadratic	Hom. cubic	Inhom. linear	Inhom. quadratic	Inhom. cubic	Gaussian
Iris	Accuracy	73.30 ± 5.11	90.27 ± 3.86	89.08 ± 8.46	69.41 ± 2.86	90.70 ± 4.20	$\underline{\mathbf{93.84 \pm 4.05}}$	90.76 ± 4.79
	CPU time (s)	0.85	5.59	5.67	35.76	52.99	53.80	113.10
Wine	Accuracy	96.77 ± 2.35	96.09 ± 2.83	95.82 ± 2.84	$\underline{\mathbf{97.23 \pm 2.26}}$	96.18 ± 2.49	95.77 ± 2.60	39.41 ± 1.09
	CPU time (s)	0.84	18.23	18.58	90.42	169.06	168.93	352.94
Fuel	Accuracy	58.47 ± 3.68	57.01 ± 4.88	57.16 ± 4.20	56.32 ± 7.13	58.09 ± 4.92	57.98 ± 5.05	$\underline{\mathbf{68.47 \pm 2.77}}$
	CPU time (s)	0.95	27.95	28.07	109.31	264.03	263.11	750.86

Table 2.5: Detailed percentage results of average accuracy and standard deviation over 50 runs of the deterministic model. Classification is performed according to the argmax decision functions (2.12) and (2.17). The best result for the kernelized model (2.14) is underlined. Overall, the best result is in bold.

First of all, by comparing the third column of each table with the remaining ones, we notice that model (2.14) outperforms the linear classifier of model (2.9) in all the considered cases. Secondly, the choice of the decision function impacts on the predictiveness of the models, especially when considering the *Iris* dataset. Indeed, in the linear case the accuracy drops when passing from the argmin decision function (2.11) to the argmax one (2.12) (91.78% vs 73.30%). On the other hand, when the kernel is homogeneous quadratic or inhomogeneous cubic the best performance is attained with the argmax decision func-

tion (76.70% vs 90.27%, and 88.27% and 93.84%, respectively). By comparing the two tables, it can be argued that both the decision functions lead to comparable accuracy as best results. Finally, when passing from the linear classifier to nonlinear classifiers the CPU time increases significantly, especially when considering the Gaussian kernel. For this reason, the final user should look for a trade-off between accuracy and computational time when using the proposed methodology.

2.6.3 Results for the robust TPMSVM models

In order to test the robust approaches, we assume that the radius of the uncertainty set (2.18) in the input space is $\varepsilon^{(i)} = \varepsilon$ for all $i \in \mathcal{X}$. To make a sensitivity analysis, 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 computations for the case of robust linear classification are reported in Tables 2.6-2.7.

Robust model - linear classifier - argmin decision function										
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	92.81 ± 3.56	92.59 ± 3.85	$\mathbf{94.49 \pm 3.45}$	92.16 ± 3.79	92.81 ± 3.52	92.43 ± 4.19	92.43 ± 3.74	92.27 ± 3.28	79.24 ± 6.89
	CPU time (s)	1.12	1.07	1.06	1.13	1.05	1.03	1.18	1.09	1.09
Wine	Accuracy	96.86 ± 2.55	97.00 ± 2.13	96.68 ± 2.11	96.95 ± 2.13	96.73 ± 2.35	97.27 ± 2.20	97.14 ± 2.42	$\mathbf{97.41 \pm 1.89}$	94.14 ± 3.56
	CPU time (s)	1.19	1.08	1.09	1.05	1.04	1.04	1.08	1.10	1.12
Fuel	Accuracy	52.28 ± 3.78	51.85 ± 4.12	52.90 ± 3.34	51.57 ± 4.50	53.10 ± 4.94	$\mathbf{53.89 \pm 4.60}$	50.60 ± 4.82	51.81 ± 4.56	50.30 ± 3.93
	CPU time (s)	1.14	1.12	1.12	1.09	1.09	1.09	1.11	1.13	1.13

Table 2.6: Detailed percentage results of average accuracy and standard deviation over 50 runs of the robust model for linear classification. Classification is performed according to the argmin decision function (2.11). The best result for each ℓ_p-norm is underlined. Overall, the best result is in bold.

Robust model - linear classifier - argmax decision function										
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	69.95 ± 3.02	69.89 ± 2.89	69.41 ± 2.81	70.16 ± 3.18	69.84 ± 2.91	68.92 ± 2.80	70.05 ± 3.31	70.38 ± 3.66	$\mathbf{85.35 \pm 5.76}$
	CPU time (s)	1.09	1.08	1.07	1.13	1.05	1.03	1.08	1.09	1.08
Wine	Accuracy	97.09 ± 2.39	96.50 ± 2.52	96.64 ± 2.31	96.55 ± 2.44	96.86 ± 2.42	96.41 ± 2.52	$\mathbf{97.09 \pm 2.30}$	96.55 ± 2.35	96.27 ± 2.78
	CPU time (s)	1.09	1.10	1.09	1.06	1.05	1.05	1.10	1.10	1.10
Fuel	Accuracy	54.77 ± 5.40	52.45 ± 5.26	$\mathbf{60.86 \pm 3.59}$	54.71 ± 5.18	56.17 ± 5.52	59.81 ± 3.63	55.46 ± 5.89	57.29 ± 4.88	55.27 ± 5.80
	CPU time (s)	1.16	1.11	1.11	1.14	1.06	1.05	1.09	1.08	1.09

Table 2.7: Detailed percentage results of average accuracy and standard deviation over 50 runs of the robust model for linear classification. Classification is performed according to the argmax decision function (2.12). The best result for each ℓ_p-norm is underlined. Overall, the best result is in bold.

A comparison among the results of Tables 2.6-2.7 with the third column of Tables 2.4-2.5 shows that in five out of six cases the robust model outperforms the corresponding deterministic formulation. We notice that the performance is particularly good with the argmax decision function (2.12) and boxed uncertainty set ($p = \infty$). Since such a kind of uncertainty set is the widest around the observation, opting for the most conservative

robust model increases the accuracy and prevents from the worst-case realizations of the uncertain data.

As far as it concerns the robust multiclass model (2.32) with nonlinear classifier, we consider as kernel function the one attaining the best accuracy in the deterministic setting (see the results in bold in Tables 2.4-2.5). The outcomes of the computations are shown in Tables 2.8-2.9.

Robust model - nonlinear classifier - argmin decision function												
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. linear	Accuracy	92.59 ± 3.66	92.05 ± 3.72	91.95 ± 3.56	92.38 ± 3.69	92.32 ± 3.68	92.43 ± 3.45	**92.59 ± 3.49**	91.68 ± 3.74	90.70 ± 5.63	
		CPU time (s)	49.66	44.49	46.38	48.61	47.39	47.27	47.36	44.73	44.49	
Wine	Inhom. linear	Accuracy	96.95 ± 2.58	96.86 ± 2.15	97.36 ± 1.97	96.95 ± 2.58	96.82 ± 2.20	**97.36 ± 1.86**	97.00 ± 2.57	96.50 ± 2.21	94.68 ± 3.17	
		CPU time (s)	99.14	94.82	97.70	98.99	97.76	97.59	99.41	98.19	96.83	
Fuel	Gaussian	Accuracy	59.01 ± 4.32	59.41 ± 2.97	38.98 ± 1.49	59.01 ± 4.32	59.41 ± 2.97	38.98 ± 1.49	**60.62 ± 3.20**	57.39 ± 4.67	37.50 ± 0.69	
		CPU time (s)	849.64	834.11	791.15	785.78	834.65	805.44	842.42	836.15	839.08	

Table 2.8: Detailed percentage results of average accuracy and standard deviation over 50 runs of the robust model for nonlinear classification (2.32). Classification is performed according to the argmin decision function (2.16). For each dataset, the kernel in the second column is chosen according to the corresponding best deterministic result of Table 2.4. The best result for each ℓ_p-norm is underlined. Overall, the best result is in bold.

Robust model - nonlinear classifier - argmax decision function												
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. cubic	Accuracy	94.43 ± 3.92	94.27 ± 3.77	**95.03 ± 3.29**	94.32 ± 3.91	94.22 ± 2.78	95.03 ± 3.64	94.00 ± 3.10	94.59 ± 2.89	83.24 ± 11.02	
		CPU time (s)	62.35	60.28	58.80	58.58	57.11	56.45	56.96	58.71	57.95	
Wine	Inhom. linear	Accuracy	96.82% ± 2.56	97.00% ± 2.18	96.73% ± 2.48	96.77% ± 2.25	**97.23% ± 2.07**	96.50% ± 2.57	96.82% ± 2.56	97.00% ± 2.18	96.09% ± 3.05	
		CPU time (s)	98.28	98.51	96.96	99.32	95.68	95.89	97.09	99.51	97.29	
Fuel	Gaussian	Accuracy	50.94 ± 5.20	**55.78 ± 4.47**	50.54 ± 5.51	50.94 ± 5.20	**55.78 ± 4.47**	50.54 ± 5.51	53.90 ± 6.64	53.76 ± 4.10	52.69 ± 6.14	
		CPU time (s)	848.21	833.59	832.41	807.30	791.32	788.50	786.92	840.70	794.83	

Table 2.9: Detailed percentage results of average accuracy and standard deviation over 50 runs of the robust model for nonlinear classification (2.32). Classification is performed according to the argmax decision function (2.16). For each dataset, the kernel in the second column is chosen according to the corresponding best deterministic result of Table 2.5. The best result for each ℓ_p-norm is underlined. Overall, the best result is in bold.

In four out of six cases the robust model provides better results when compared to the deterministic framework. On the other hand, when considering the *Fuel* dataset whose best kernel is the Gaussian one, the results worsen in the robust context. This is mainly due to the fact that this kind of kernel does not bear strong perturbation in data. Interestingly, the standard deviations over the 50 runs are in general lower when compared to the ones in the corresponding deterministic settings, meaning that the methodology is much more stable.

Finally, the CPU time of the robust approach, with both linear and nonlinear classifiers, is slightly higher than the one in the deterministic formulation. This is due to the fact that a SOCP problem is solved instead of a QPP.

2.7 Conclusions

In this chapter, we have proposed a novel method to address the problem of multiclass classification by adapting the TPMSVM approach of [116]. Both linear and kernel-induced decision boundaries have been considered. In order to protect the models against perturbations in the samples, we have constructed bounded-by-ℓ_p-norm uncertainty sets around each input data. This allows to increase the flexibility of the proposed methodology. Robust counterparts of the deterministic models have been derived both in the case of linear and nonlinear classifiers, leading to SOCP problems. To evaluate the accuracy of the proposed methodology, we have performed numerical results on real-world datasets, comparing deterministic and robust methods. Different ℓ_p-norms and levels of perturbations have been explored, as well as decision functions. The experimental analysis has shown that robust solutions provide higher accuracy when compared to deterministic classifiers.

Further research activities could be focused on robustifying the TPMSVM against uncertainties in the labels of the training set, by constructing a family of classifiers capable to cope with uncertainties both in the features and in the labels. The extension of the TPMSVM methodology to distributionally robust approaches should be investigated too.

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. This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

Chapter 3

An Application of Robust Support Vector Machine Approaches to Vehicles Smog Rating Classification

Authors: Renato De Leone[1], Francesca Maggioni[2] and Andrea Spinelli[3].

Keywords: Machine Learning; Support Vector Machine; Robust Optimization; Multiclass Classification; Carbon Emission

The first three subsections of the Experimental study of this chapter have been published in *Optimization in Green Sustainability and Ecological Transition (ODS 2023) - volume 12 of AIRO Springer Series.*

DOI: https://doi.org/10.1007/978-3-031-47686-0_19

The last subsection of the Experimental study of this chapter has been published in *Machine Learning, Optimization, and Data Science - volume 14506 of Lecture Notes in Computer Science.*

DOI: https://doi.org/10.1007/978-3-031-53966-4_22

[1]School of Science and Technology, University of Camerino, Via Madonna delle Carceri 9, Camerino 62032, Italy. renato.deleone@unicam.it

[2]Department of Management, Information and Production Engineering, University of Bergamo, Viale G. Marconi 5, Dalmine 24044, Italy. francesca.maggioni@unibg.it

[3]Department of Management, Information and Production Engineering, University of Bergamo, Viale G. Marconi 5, Dalmine 24044, Italy. andrea.spinelli@unibg.it

3.1 Introduction

Nowadays, sustainability is at the core of policy agendas and debates worldwide. According to the Sustainable Developments Goals 7 and 11 of the UN 2030 Agenda (see [148]) the global greenhouse gas emissions coming from transportations should be effectively reduced. This action can be performed through the promotion of electric and zero emission vehicles by means of new standards, fiscal incentives and improved consumer information.

To raise consumer awareness and promote green attitudes, in 1999 the European Union promulgated the "car labelling Directive" (see [52]) to inform consumers about the fuel consumption CO_2 emissions of new passenger cars. Similarly, in 2014 the United States Environmental Protection Agency introduced the *air pollution score* as a measurement to rate the amount of air pollution emitted by a vehicle (see [50]). Recently, in Canada fuel consumption tests have been performed on new vehicles by using a 5-cycle procedure by considering both city and highway conditions (see [115]). As in the US, a value ranging from 0 (worst) to 10 (best) is assigned to each new vehicle. Unfortunately, from a practical perspective, it is not reasonable to test every new vehicle to measure its fuel consumption. For these reasons, *Operational Research* and *Machine Learning* (ML) techniques should help policymakers providing new and sustainable solutions.

The aim of this study is to apply the robust *Support Vector Machine* (SVM) techniques presented in Chapters 1 and 2 to a classification task related to the pollution emitted by vehicles.

In the ML literature, such a kind of problem has been addressed in various works. In all of these studies, cars and trucks are classified on the basis of images properties or camera detections (see [57]). To the best of our knowledge, the approach proposed in this chapter is the first that considers the problem of classifying vehicles in terms of their emissions with a ML perspective and under data uncertainty.

In this chapter, we consider the problem of predicting vehicles smog rating on the basis of different characteristics of the vehicle such as engine size, number of cylinders, fuel consumption and CO_2 tailpipe emissions. We tackle the classification task by means of the novel robust SVM formulations from Chapters 1 and 2. Data uncertainty is explicitly handled within the models by means of spherical uncertainty sets centered around training observations. In the following, we provide numerical experiments on synthetic and real-world datasets with the aim of understanding the advantage of explicitly considering the uncertainty versus deterministic approaches in the considered application.

The chapter is organized as follows. Section 3.2 describes data and reports the numer-

ical experiments on both synthetic and real-world datasets. Conclusions of the work are summarized in Section 3.3.

Throughout the chapter we assume that each of the m vehicles in the dataset is described by a n-dimensional vector of features $x^{(i)}$, related to fuel consumptions and CO_2 emissions. A label $y^{(i)} \in \{1, \dots, C\}$ is attached to each vehicle, denoting the class to which it belongs, where C being the total number of classes. When dealing with robust models, we construct around each sample $x^{(i)}$ a spherical uncertainty set $\mathcal{U}_2(x^{(i)})$ defined as in (1.11) with $p = 2$, namely:

$$\mathcal{U}_2(x^{(i)}) := \left\{ x \in \mathbb{R}^n : x = x^{(i)} + \sigma^{(i)}, \|\sigma^{(i)}\|_2 \leqslant \eta^{(i)} \right\}. \tag{3.1}$$

3.2 Experimental study

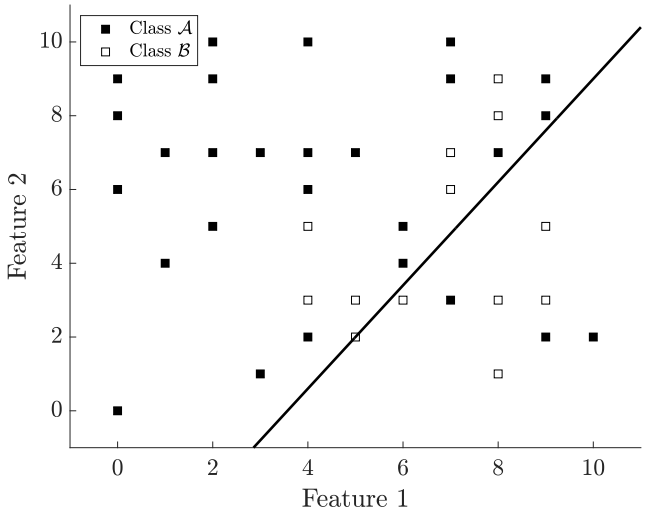
This section discusses the numerical experiments. In Subsection 3.2.1 we explore the performance of the models on a synthetic dataset. Then, we describe the features of the real-world dataset regarding vehicles emissions (Subsection 3.2.2). Finally, in Subsections 3.2.3-3.2.4 the performance of the models on the basis of classical statistical indicators are discussed.

All computational experiments are obtained using CVX (see [64, 65]) in MATLAB (v. 2021b) and solver MOSEK (v. 9.1.9, see [112]) on a MacBookPro17.1 with a chip Apple M1 and 16 GB of RAM.

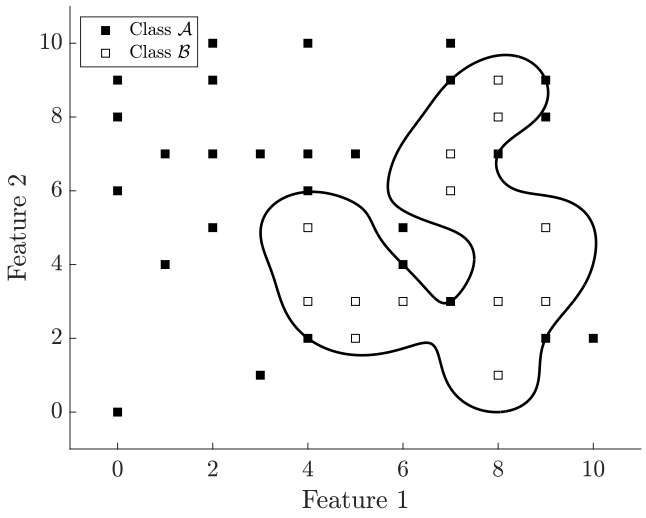
3.2.1 Synthetic dataset

We start by testing model (1.18) on an artificial example taken from [147]. The dataset is composed by 37 observations, belonging either to one class (13 observations) or to another class (24 observations), and characterized by two features. We set parameter $\nu = 10$ and for the Gaussian kernel $\alpha = 1$ as in [147]. In the uncertainty set (3.1) we take $\eta^{(i)} = \eta$ for all $i = 1, \dots, m$. We start by considering $\eta = 0$, namely the deterministic case. In this situation, the optimal separating surfaces are depicted in Figure 3.1.

When $\eta > 0$ data points are subject to perturbations. Specifically, as in [147] we study the evolution of the inverse of the margin $\|u\|_\infty$, when increasing the value of η. We consider one hundred evenly spaced values of η between 0 and $\eta_{\max}$, where $\eta_{\max}$ is equal to 0.1191 for the linear kernel and 0.0706 for the Gaussian kernel. Indeed, when the ℓ_2-norm of the perturbation is greater than $\eta_{\max}$, model (1.18) has as solutions $u = 0 \in \mathbb{R}^m$ and $\gamma = 0$. This implies that equation (1.8) is trivially satisfied, regardless of $x \in \mathbb{R}^n$.



(a) Linear kernel

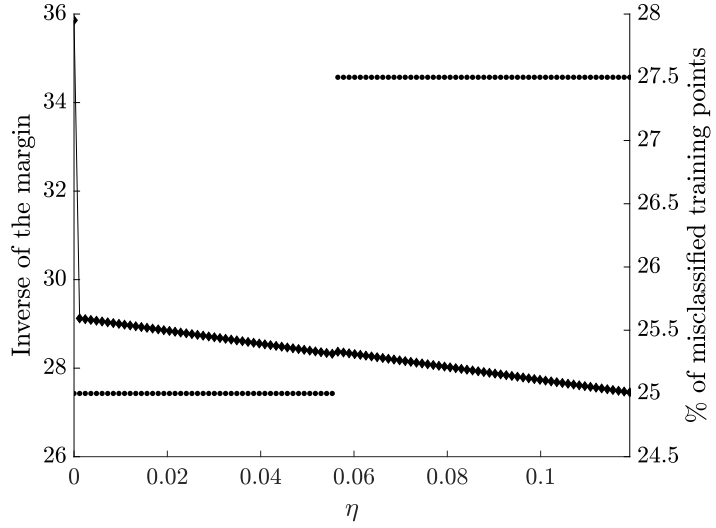


(b) Gaussian kernel

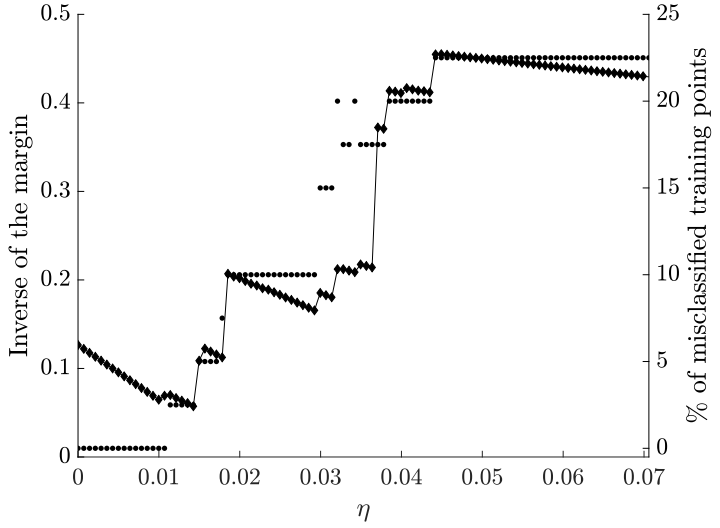
Figure 3.1: Optimal separating surfaces in the deterministic case (model (1.18)).

Therefore, no classification is induced. The results of the analysis are represented in Figure 3.2.

When $\eta \leqslant \eta_{\max}$, the quantity $\|u\|_\infty^{-1}$ shows a piecewise-linear trend, depicted as a solid line with diamonds. In particular, the Gaussian case (Figure 3.2b) shows a more fragmented evolution when compared to the linear case (Figure 3.2a). However, in both cases, the occurrence of a point of discontinuity in the evolution of $\|u\|_\infty^{-1}$ results in an increasing of the number of misclassified points, depicted in percentage as circles. In addition, there exists a value of η corresponding to the minimum of $\|u\|_\infty^{-1}$, respectively at $\eta = 0.1191$ and $\eta = 0.0143$. As in [147], we deduce that when a small uncertainty is



(a) Linear kernel



(b) Gaussian kernel

Figure 3.2: Plots of the inverse of the margin $\|u\|_\infty$ depicted as diamonds (left-hand scale) as function of η, with the corresponding percentage of misclassified training points (circles, right-hand scale) (model (1.18)).

considered, the robust classifier has a better generalized margin when compared to the deterministic case, by allowing the model to protect against uncertainty. For instance, in the linear case even the introduction of a small uncertainty implies a reduction in $\|u\|_\infty^{-1}$ (from 35.86 to 28.13), without affecting the accuracy (25% of misclassified training points). Finally, we notice that the order of magnitude of $\eta_{\max}$ is different between the linear and the Gaussian kernel. In particular, in the linear case a stronger uncertainty is allowed to be included in the model, but at the expense of predictive power: the misclassification error is higher when compared to the Gaussian kernel. Therefore we can conclude that

there exists a trade-off between the maximum level of allowed uncertainty and performance accuracy.

3.2.2 Real-world dataset description

Real-world data on the fuel consumption ratings on 374 different vehicles in the first months of 2023 are taken from [115]. Specifically, we focus our attention on the vehicles whose fuel type is "regular gasoline" because they are the most polluting. Each vehicle is described by 7 attributes: engine size (range $1.2 - 8$), number of cylinders (range $3 - 16$), fuel consumptions rating in city (range $4.4 - 30.3$), fuel consumptions rating in highway (range $4.4 - 20.9$), combined city-highway fuel consumptions rating (range $4.4 - 26.1$), tailpipe emissions of CO_2 for combined city-highway driving (range $104 - 608$), and tailpipe emissions of CO_2 rating (range $1 - 9$). Each vehicle is labelled according to the tailpipe emissions of smog-forming pollutants rate and assigned to one of the following categories: 3 (worst emissions), 5, 6, 7, 8 (best emissions).

The distribution of vehicles among the different classes is reported in Table 3.1.

Smog rating score	Class	Number of vehicles	Class distribution
3	1	15	4.01%
5	2	127	33.96%
6	3	83	22.19%
7	4	145	38.77%
8	5	4	1.07%

Table 3.1: Distribution of vehicles among classes in the considered datasets.

In the following, we report the performance of the models from Chapter 1 (Subsection 3.2.3) and Chapter 2 (Subsection 3.2.4) on the considered dataset. In both cases, the dataset has been divided into training set and testing set through a k_F-fold cross-validation technique. Further details on the procedure are provided below.

3.2.3 Model (1.18) validation

We start by considering model (1.18). Since this formulation performs a binary classification, we aggregate vehicles in classes 3-5-6 (bad smog rating, 60.16% of the vehicles) and vehicles in classes 7-8 (good smog rating, 39.84% of the vehicles).

For the cross-validation, we consider $k_F = 20$ folds. For each fold s, the classifier of model (1.18) has been trained on 356 points outside the subsample, and tested on the 18

points of the fold. Associated with fold s, the quality of the solution has been measured through the following indicators:

$$\mathrm{A}_s := \frac{\mathrm{TP}_s + \mathrm{TN}_s}{\mathrm{TP}_s + \mathrm{FP}_s + \mathrm{TN}_s + \mathrm{FN}_s}, \quad \mathrm{FPR}_s := \frac{\mathrm{FP}_s}{\mathrm{FP}_s + \mathrm{TN}_s}, \quad \mathrm{P}_s := \frac{\mathrm{TP}_s}{\mathrm{TP}_s + \mathrm{FP}_s},$$

where TP_s stands for true positive, TN_s for true negative, FP_s for false positive, and FN_s for false negative. Finally, the results are averaged, obtaining:

$$\text{Accuracy} := \frac{\sum_{s=1}^{20} \mathrm{A}_s}{20}, \quad \text{False Positive Rate} := \frac{\sum_{s=1}^{20} \mathrm{FPR}_s}{20}, \quad \text{Precision} := \frac{\sum_{s=1}^{20} \mathrm{P}_s}{20}.$$

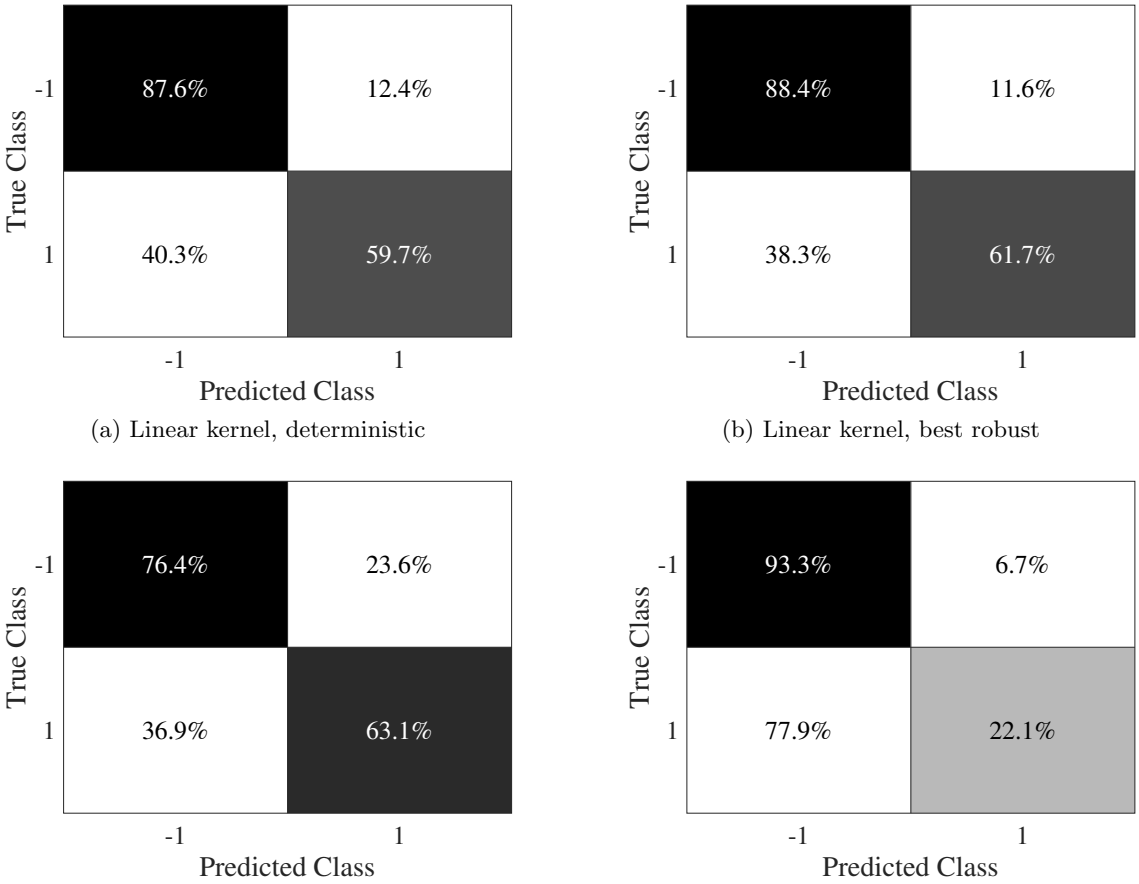
As in Subsection 3.2.1, we set $\nu = 10$ and for the Gaussian kernel $\alpha = 1$. For the robust model, as in [53] we set $\eta^{(i)} = \eta$ for all $i = 1, \ldots, m$, and considering three increasing levels η of uncertainty, namely 0.001, 0.005, 0.01. The results of the simulations are reported in Table 3.2.

	Linear kernel				Gaussian kernel			
	Deterministic	Robust			Deterministic	Robust		
η	0	0.001	0.005	0.01	0	0.001	0.005	0.01
Accuracy	76.43%	77.78%	**81.33%**	76.92%	71.10%	73.27%	72.91%	65.04%
False Positive Rate	12.46%	11.36%	15.04%	14.28%	23.33%	17.35%	11.59%	**6.59%**
Precision	77.38%	**78.64%**	74.52%	74.45%	71.24%	71.09%	75.15%	74.17%
CPU time (s)	9.72	9.50	9.57	9.68	9.54	9.56	9.43	9.37

Table 3.2: Performance of model (1.18) measured by different indicators. For each kernel and for each indicator, the best result is underlined. Overall, the best performance is highlighted in bold.

It can be noted that all indicators benefit from including uncertainties in the proposed formulations. The highest levels of accuracy (81.33%) and of precision (78.64%) are achieved by the linear kernel, respectively with $\eta = 0.005$ and $\eta = 0.001$. Conversely, with the Gaussian kernel and $\eta = 0.01$ the minimum false positive rate is attained (6.59%). It is worth noticing that, from a practical perspective, it is worse to classify a vehicle with "bad smog rating" in the class of vehicles with "good smog rating" than the opposite. Therefore, it is reasonable to consider a more conservative model with good accuracy and low false positive rate. Consequently and similarly to the artificial example in Section 3.2.1, a trade-off between the average performance of the model and its ability to protect against uncertainty needs to be taken into account.

In Figures 3.3 we report the confusion matrices of the average results both for the deterministic and "best robust" models in terms of false positive rate. In line with the previous reasoning, we focus our attention to the first row of each matrices, namely to the true and predicted *negative samples*. It can be noted that within the linear kernel (Figures 3.3a-3.3b) the increase of uncertainty causes a slight better performance (from 12.4% to 11.6% for the false positive samples). On the other hand, when the kernel is Gaussian,



(a) Linear kernel, deterministic

(b) Linear kernel, best robust

(c) Gaussian kernel, deterministic

(d) Gaussian kernel, best robust

Figure 3.3: Row-normalized confusion matrices. The "best robust" corresponds to the best performance in terms of false positive rate (model (1.18)).

the percentage of false positive drops significantly (from 23.6% to 6.7%), meaning that the model is protected against *type I error* (Figures 3.3c-3.3d).

As far as it concerns the computational time, no significant difference can be highlighted because they are all LP problems (see Table 3.2). In particular, each formulation is solved on average in 9.55 seconds.

3.2.4 Model (2.20) validation

Since in this case we deal with multiclass classification, it is not correct to refer to positive or negative observations. For this reason, we measure the performance of model (2.20) only in terms of the accuracy. Specifically, for each fold $s = 1, \dots, k_F$ and for each class $c = 1, \dots, C$, the quality of the solution has been measured by means of an in-class accuracy, defined as:

$$\mathrm{A}_c^s := \frac{\text{number of testing points correctly classified in class } c}{\text{number of testing points in class } c}.$$

In addition, for each fold s the overall performance of the classification process has been validated through an aggregate accuracy measure, computed as:

$$\mathrm{A}^s := \frac{\text{number of testing points correctly classified}}{\text{number of testing points}}.$$

Finally, the results are averaged:

$$\text{Accuracy for class } c := \frac{\sum_{s=1}^{k_F} \mathrm{A}_c^s}{k_F}, \qquad \text{Accuracy} := \frac{\sum_{s=1}^{k_F} \mathrm{A}^s}{k_F}.$$

Since class 5 contains only 4 samples (see Table 3.1), we set $k_F = 4$. As in [116], we normalize data such that the features locate in $[0, 1]$. We validate model (2.20) by comparing the results with the *One-Versus-Rest multiclass TWin Support Vector Machine* (OVR TWSVM) presented in [45]. The choice is motivated by the fact that both model (2.20) and OVR TWSVM are one-versus-all approaches, considering each class one at a time in the training process. For brevity's sake, regularization parameters are set to $\nu_c = \nu$ and $\alpha_c = \alpha$ for all $c = 1, \ldots, C$ and a grid search procedure is applied to tune their values. Specifically, α is selected from the set $\{2^j | j = -5, -4, \ldots, 4, 5\}$, and the value of ν/α is chosen from the set $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. The degree of perturbation $\eta^{(i)}$ in the uncertainty set (3.1) is assumed to be equal to η for all $i = 1, \ldots, m$. In particular, we consider $\eta = 0$, i.e. no perturbation, and three increasing levels of uncertainty, namely 0.001, 0.01 and 0.1. The final decision function is given by equation (2.11).

The results of the simulations are shown in Table 3.3.

	OVR TWSVM [45]	Deterministic model (2.20)	Robust model (2.20)		
η	0	0	0.001	0.01	0.1
Accuracy for class 1	0.00%	0.00%	0.00%	0.00%	0.00%
Accuracy for class 2	40.42%	51.99%	45.61%	**67.06**%	61.82%
Accuracy for class 3	7.50%	5.95%	**13.39**%	0.00%	0.00%
Accuracy for class 4	73.84%	73.03%	69.63%	75.11%	**77.46**%
Accuracy for class 5	50.00%	50.00%	50.00%	0.00%	0.00%
Accuracy	44.39%	47.85%	45.98%	**51.89**%	51.05%
Time (s)	0.513	0.717	0.719	0.697	0.662

Table 3.3: Performance of model (2.20) on real-world data on fuel consumption (see [115]). For each indicator, the best result is highlighted in bold.

First of all, we notice that our deterministic formulation (2.20) with $\eta = 0$ outperforms OVR TWSVM from [45] in terms of overall accuracy (47.85% vs 44.39%). Secondly, the majority of the indicators benefit from including uncertainties in the proposed formu-

lations. The overall accuracy across the different levels of perturbation is higher when compared to the deterministic case (47.85%), with the best result (51.89%) attained at $\eta = 0.01$. Nevertheless, due to the limited number of observations in classes 1 and 5 (see Table 3.1), both models are no longer able to predict the correct smog rating score in these classes. Specifically, class 5 does not gain additional accuracy improvement adding noise to the data. This is to be expected since in these cases specific techniques for handling rare events should be implemented (see [27, 67]). However, this is out of scope of the chapter. When considering the confusion matrices, both deterministic formulations predict 60.0% of the observations in class 1 to be in classes 3, 4 or 5, whereas the value decreases to 33.3% with the robust model (2.20) with $\eta = 0.01$. From a practical perspective, since class 1 is the most polluting, a robust approach may be suitable in order to protect the model against this type of misclassification error.

With the aim of evaluating the performance of our model, we aggregate data with bad smog rating (scores 3-5), and with good smog rating (scores 7-8). With this choice, the dataset is partitioned into three main categories: class $\tilde{1}$ (*bad emissions*), class $\tilde{2}$ (*medium emissions*), class $\tilde{3}$ (*good emissions*). The corresponding distribution of observations in each class is 37.97%, 22.19%, 39.84%, respectively.

The results of the computational experiments on the reduced dataset with 3 classes are shown in Table 3.4.

	OVR TWSVM [45]	Deterministic model (2.20)	Robust model (2.20)		
η	0	0	0.001	0.01	0.1
Accuracy for class $\tilde{1}$	61.81%	54.43%	50.81%	**64.90%**	61.81%
Accuracy for class $\tilde{2}$	10.77%	26.73%	**41.07%**	9.58%	4.76%
Accuracy for class $\tilde{3}$	70.63%	76.49%	69.81%	73.83%	**81.88%**
Accuracy	54.01%	56.67%	56.17%	56.15%	**57.21%**
Time (s)	0.321	0.424	0.411	0.436	0.424

Table 3.4: Performance of model (2.20) in the case of 3 classes. For each indicator, the best result is highlighted in bold.

The comparison between the results of OVR TWSVM from [45] and our formulation leads to the same conclusion as in the case of 5 classes, having an overall accuracy of 56.67% vs 54.01% in the deterministic case. Besides, the robust formulation allows to increase the overall accuracy level to 57.21% when $\eta = 0.1$. In addition, the predictive power of the models with 3 classes is always higher than with 5 classes, meaning that unbalanced data may disrupt the reliability of the solution. When including uncertainty, each class benefits at different extents: classes $\tilde{1}$ and $\tilde{3}$ withstand strong degrees of uncertainty ($\eta = 0.01$ or 0.1), whereas class $\tilde{2}$ takes advantages only with low perturbations ($\eta = 0.001$). It is

worth noticing that, from a practical perspective, it is worse to misclassify a vehicle with "bad emission"(class $\tilde{1}$) than the other cases. Therefore, it is reasonable to consider a strong robust model with $\eta = 0.01$ or 0.1 which attains a good accuracy for class $\tilde{1}$. Consequently, a trade-off between the performance of the model and its ability to protect against uncertainty needs to be taken into account.

As far as it concerns timing, the OVR TWSVM has a faster learning speed when compared to our formulation, but with a lower predictive power. Moreover, since both OVR TWSVM and model (2.20) are one-versus-all approaches, requiring to solve C optimization subproblems, the CPU time depends on the number of classes. Within our examples, when passing from 3 to 5 classes, an average increase of 62% of the learning time occurs.

3.3 Conclusions

This chapter presents novel optimization approaches to classify vehicles in terms of smog rating emissions under uncertainty as Support Vector Machine tasks. The techniques proposed in Chapters 1 and 2 help decision makers to rank passenger cars in terms of their pollution emissions. Given the uncertain nature of real-world data features, we formulate robust optimization models with spherical uncertainty sets around samples. The numerical results show the good performance of the proposed formulations, especially when including uncertainty, both in synthetic and real-world datasets.

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. This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

Chapter 4

A Rolling Horizon Heuristic Approach for a Multi-stage Stochastic Waste Collection Problem

Authors: Andrea Spinelli[1], Francesca Maggioni[2], Tânia Rodrigues Pereira Ramos[3], Ana Paula Barbosa-Póvoa[4] and Daniele Vigo[5].

Keywords: Routing; Waste Collection; Multi-stage Stochastic Programming; Rolling Horizon Approach.

This chapter is under first revision in *European Journal of Operational Research*.
Manuscript Reference Number: EJOR-D-23-01517.

[1]Department of Management, Information and Production Engineering, University of Bergamo, Viale G. Marconi 5, Dalmine 24044, Italy. andrea.spinelli@unibg.it
[2]Department of Management, Information and Production Engineering, University of Bergamo, Viale G. Marconi 5, Dalmine 24044, Italy. francesca.maggioni@unibg.it
[3]Centre for Management Studies (CEGIST), Instituto Superior Técnico, University of Lisbon, Av. Rovisco Pais 1, Lisbon 1049-001, Portugal. tania.p.ramos@tecnico.ulisboa.pt
[4]Centre for Management Studies (CEGIST), Instituto Superior Técnico, University of Lisbon, Av. Rovisco Pais 1, Lisbon 1049-001, Portugal. apovoa@tecnico.ulisboa.pt
[5]Department of Electrical, Electronic, and Information Engineering "G. Marconi", University of Bologna, Viale del Risorgimento 2, Bologna 40136, Italy. daniele.vigo@unibo.it

4.1 Introduction

In recent years, the importance of sustainable waste management processes has been recognized worldwide (see, for example, the new Circular Economy Action Plan [51]). These practices involve decisions at different levels (strategical, tactical, and operational), depending on the duration of the considered period (see [62] for a general survey), and combine different aspects. Among them, the efficiency of the waste collection operation of recyclable materials is a problem that needs to be addressed (see [18]). Traditionally, waste collection is based on static and pre-defined routes, based on average data about bins filling rate and then executed on a regular basis regardless of actual bin filling (see [42]). These practices may imply high rates of resources' inefficiencies, due to too early collection of not filled bins, or to poor service level because of too late collections.

The great majority of the literature on waste collection models considers deterministic formulations (see, for instance, [1, 42, 43]), where all the parameters are known when making decisions. Nevertheless, this assumption may not be true in all cases, as uncertainties exist namely on the traveling time as well as on the accumulation rate of waste in the bins (see [42]). In such a complex framework, *Stochastic Optimization* techniques (see [19]) may help service providers to implement cost-effective decision plans.

Motivated by the uncertain and dynamic nature of the waste accumulation, in this chapter we formulate a multi-stage linear mixed-integer stochastic optimization model for recyclable waste collection. The waste operator company is required to make decisions at tactical level in a mid-term time horizon, regarding waste bins selection over time and their combination into vehicle routes. The aim of the planning is to maximize the profit, given by the difference between the revenues from selling the collected waste and the transportation costs.

This type of problems are among the most challenging in the literature, combining stochasticity and discrete decisions. Exact solution methods are in general based on branch and bound type algorithms or branch and price methods. Since the size of stochastic optimization model grows exponentially with respect to the number of stages and of scenarios, heuristic algorithms are needed. On this purpose, we adopt the *rolling horizon* approach (see [32]), a classical heuristic for multi-stage stochastic problems. According to this technique, the model is decomposed into a sequence of subproblems defined over a reduced time horizon. The model is solved starting from the first time period and the value of the first-stage variables is captured. The procedure is then repeated starting from the second stage and so on until the end of the time horizon.

The proposed stochastic formulation is tested on instances of different sizes, based on real data, and the results are validated by means of classical stochastic measures.

The main contributions of this chapter can be summarized as follows:

- To develop a multi-stage stochastic optimization model for the waste collection inventory routing problem;
- To apply the rolling horizon approach to solve the model and to analyze its worst-case performance;
- To provide numerical experiments with the aim of:
  (1) validating the model in terms of *in-sample stability* (see [75]);
  (2) measuring the impact of uncertainty and the quality of the deterministic solution in a stochastic setting;
  (3) evaluating the performance of the rolling horizon approach in terms of optimal objective function value and reduction of CPU time;
  (4) testing the effectiveness of the proposed methodology on a real case study.

The remainder of the chapter is organized as follows. Section 4.2 reviews the existing literature on the problem. In Section 4.3, the waste collection problem is described and a multi-stage stochastic programming model is formulated. Section 4.4 describes the rolling horizon approach and provides a worst-case analysis on its performance. In Section 4.5, the computational results are shown and the managerial insights are discussed. Finally, Section 4.6 concludes the chapter.

4.2 Literature review

Waste collection problems are mostly modeled in the literature as *Vehicle Routing Problems* (VRPs), where a predefined set of bins to be collected is considered and routes are defined accordingly, by minimizing, for instance, the total travelled distance (see [145] for a comprehensive overview on VRPs). In recent years, such kind of problems have been widely studied and extended, in order to include different features. In [54], a *Capacitated Vehicle Routing Problem* (CVRP) in which garbage trucks have limited carrying capacity is studied; in [3], a *Periodic Vehicle Routing Problem* (PVRP) is designed such that visiting schedules on a given time horizon are associated with each container; in [127], a *Vehicle Routing Problem with Profits* (VRPP) is developed, where the profit comes from selling the collected waste to a recycling company.

Inventory Routing Problem (IRP) is an extension of VRP because it integrates inventory management and vehicle routing decisions over a medium or long-term planning period. In the classical IRP, three different decisions have to be made: when to restock the customers' inventories, how much product to deliver, and how to combine customers into vehicle routes. In the special case of waste collection, the flows are reversed because the aim of visiting is collecting rather than delivering and the decision on how much to collect is not important, because waste bins will always be fully emptied (see [109]). According to [103], IRP models in reverse logistics are mostly motivated by real case studies, by considering the collection of specific waste products: paper and cardboard ([43]), paper and glass ([21, 49]), white glass ([107]), waste vegetable oil ([1, 2, 25, 26]), infectious medical waste ([113]), components from end-of-life vehicles ([83]).

In the classical IRP models, all the involved parameters are treated as deterministic, i.e. they are assumed to be already known when taking decisions. Nevertheless, stochasticity may corrupt routing problems at different levels ([61]): stochastic customers, stochastic demand, and stochastic travel times. In the waste collection context, a high degree of uncertainty may affect waste production (i.e. the demand), that in general disrupts the reliability of the solution made by service providers through deterministic approaches. As an attempt to reduce uncertainty, in some areas waste containers are equipped with volumetric sensors that communicate their waste level to the waste manager. Basing on the transmitted real-time information, the collection is thus planned. In [127], the *Smart Waste Collection Routing Problem* (SWCRP) is introduced, where the sensors' usage is combined with optimization procedures to guarantee the maximization of the waste collected, at lowest transportation cost. In [55], a scheduling of weekly waste collection activities for multiple types of waste is derived, by considering sensors both on underground containers and inside garbage trucks. However, these contributions do not include uncertainty directly in the optimization model. A comprehensive survey on operational research applied to solid waste management with specific focuses on uncertainty can be found in [62].

Whenever the supplier has access to some information about the probability distribution of customer's demand, the IRP falls within the framework of *Stochastic Inventory Routing Problem* (SIRP). The reader is referred to [37] and [111] for an exhaustive analysis of the literature on IRP and particularly on SIRP. Stochastic programming, robust optimization and chance-constrained optimization ([113, 139, 140]) are some of the paradigms recently explored to cope with uncertainty in SIRP models. On the other hand, including stochasticity in IRP models increases dramatically the computational tractability.

Therefore, heuristic methods are needed, especially when solving large instances. In the following, we limit our attention to heuristics applied to the waste collection problem under uncertainty.

In [109], a simple heuristic is designed to identify containers that must be visited or may be visited, and combined in an efficient route. If it is convenient, additional containers are added and therefore emptied. In [3], a tabu search for waste collection with intermediate facilities is applied. Starting from an initial solution, in its neighborhood a new solution minimizing a penalized cost function is searched. If it is feasible and attains a better objective function value, then it is considered as the current best solution; otherwise, the search continues until a stopping criterion is met. In [114], the authors describe the zoning of a service territory as a stochastic periodic VRP with time windows, and they solve the problem by a metaheuristic. In their work the stochasticity lies in the accumulation rate of waste in each bin, and in the travel times. In [43], a dynamic approach to solve the SWCRP is proposed, by applying a combination between the rolling horizon approach and the relax-and-fix heuristic. Recently, in [42] a solution methodology for the SWCRP with workload concerns is explored. Similarly as before, to tackle the complexity of the model, a look-ahead heuristic in a first phase and either an optimisation-based approach or a hybrid metaheuristic approach in a second phase are considered. In [107], historical data and forecasting techniques are used to estimate the expected containers' filling rate over the planning horizon and to derive the distribution of the overflow probability. Then, an *Adaptive Large Neighborhood Search* (ALNS), with search guiding principle based on simulated annealing is applied, along with a rolling horizon approach.

In this work, we test the performance of a rolling horizon approach for the waste collection problem within the paradigm of SIRP. This approach has been extensively used in the literature (see [31] for a classified bibliography and [95] for the definition of rolling horizon measures). Among its applications to general transportation problems, we mention the works by [28, 97, 135]. In [12] a worst-case analysis of the rolling horizon approach for a stochastic multi-stage fixed charge transportation problem is provided.

4.3 Problem description and formulation

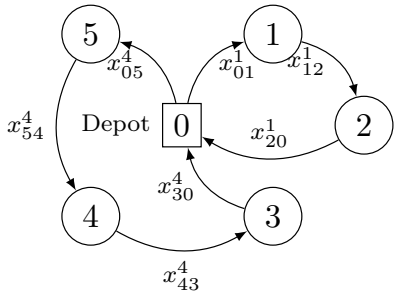
A company is responsible for the collection of a recyclable type of waste in a set of N locations (bins or containers) over a time horizon $\mathcal{T} = \{1, \dots, T\}$. The collection network is represented as a complete directed graph, defined on a set of vertices $\mathcal{I} = \{0, 1, \dots, N\}$ where 0 denotes the depot. Distances d_{ij} are associated with each arc $(i, j) \in \mathcal{I} \times \mathcal{I}$ in

the graph. The company needs to determine at stage $t = 1$ which waste bins have to be visited and the visiting sequence for all stages $t \in \mathcal{T}'' = \{2, \dots, T\}$. The choice has to be performed with the aim of maximizing the profit over the whole planning horizon $\mathcal{T}$, defined as the difference between the revenues from the selling of the collected waste and the transportation costs. The waste is sold at unit price R and the travelling cost per distance unit is fixed to C.

Each bin $i \in \mathcal{I}' = \{1, \dots, N\}$ has a fixed capacity E_i and in the first stage ($t = 1$) it is supposed to be filled at S_i^{init} percent of its volume. If we assume that the accumulation rate of waste $\{a_i^{(t)}\}_{t=1}^{T}$ of bin i is a random parameter evolving as a discrete-time stochastic process with support $[0, 1]$, then the information structure can be described in the form of a scenario tree. At each stage $t \in \mathcal{T}$, there is an ordered set $\mathcal{N}^t = \{1, \dots, n, \dots, n^t\}$ of nodes where a specific realization of the uncertain accumulation rate takes place. At the first stage it is associated a unique node $\mathcal{N}^1 = \{1\}$, i.e. the root, whereas the final n^T nodes are the leaves of the scenario tree. At stage $t \in \mathcal{T}''$, each node $n \in \mathcal{N}^t$ is connected to a unique node at stage $t - 1$, which is called parent (or ancestor) node $pa(n)$. A path through nodes from the root to a leaf is called scenario. At each stage $t \in \mathcal{T}$, each node $n \in \mathcal{N}^t$ has a probability π^n to occur, and $\sum_{n \in \mathcal{N}^t} \pi^n = 1$. We denote the accumulation rate for bin i at node $n \in \mathcal{N}^t$, $t \in \mathcal{T}''$ by a_i^n.

At each stage $t \in \mathcal{T}' = \{1, \dots, T - 1\}$, we define binary decision variables x_{ij}^t and y_i^t. The former is related to the activation of the arc (i, j) in period $t + 1$. Indeed, at each stage the model plans for the next stage, by reflecting what happens in practice for the scheduling of the resources in a waste collection company. If x_{ij}^t is equal to one, then the arc (i, j) will be traversed by a vehicle, with finite capacity Q. All the variables x_{ij}^t are defined on the whole graph. Indeed, we assume that, in the collection period, the vehicle starts at the depot, visits the selected bins and returns to the depot to discharge the waste. As far as it concerns the decision variables y_i^t, if bin $i \in \mathcal{I}'$ needs to be visited in period $t + 1$, then variable y_i^t is equal to one at stage t. After the realization of the accumulation rate, the amount of waste collected at bin i is denoted by w_i^n, for $n \in \mathcal{N}^t$, $t \in \mathcal{T}'$. In Figure 4.1 we provide an example of a planning for an horizon of six days.

At stage $t \in \mathcal{T}'$ and for nodes $n \in \mathcal{N}^t$, additional decision variables are f_{ij}^n representing the waste flow shipped through arc (i, j). We assume that the waste flow outgoing depot is zero. Finally, for all the time periods, we denote by u_i^n the accumulated amount of waste at bin i. By avoiding partial collection, when bin i is visited, u_i^n is null.



Stage t	$x^t_{ij}=1$	$y^t_i=1$	Visiting sequence
1	$x^1_{01}, x^1_{12}, x^1_{20}$	y^1_1, y^1_2	
2			$0,1,2,0$
3			
4	$x^4_{05}, x^4_{54}, x^4_{43}, x^4_{30}$	y^4_3, y^4_4, y^4_5	
5			$0,5,4,3,0$
6			

Figure 4.1: Example of a collection plan with 5 bins. On the left: collection routes for day 2 (bins 1, 2) and day 5 (bins 5, 4, 3). On the right: table with active binary decision variables x^t_{ij} and y^t_i and corresponding visiting sequence.

Moreover, we define the following notation.

Sets:

$\mathcal{I} = \{i : i = 0, 1, \ldots, N\}$: set of N waste bins and the depot, denoted by 0;

$\mathcal{I}' = \{i : i = 1, \ldots, N\}$: set of N waste bins (depot excluded);

$\mathcal{T} = \{t : t = 1, \ldots, T\}$: set of stages;

$\mathcal{T}' = \{t : t = 1, \ldots, T-1\}$: set of stages (last stage excluded);

$\mathcal{T}'' = \{t : t = 2, \ldots, T\}$: set of stages (first stage excluded);

$\mathcal{N}^1 = \{n : n = 1\}$: root node at stage 1;

$\mathcal{N}^t = \{n : n = 1, \ldots, n^t\}$: set of ordered nodes of the tree at stage $t \in \mathcal{T}$.

Deterministic parameters:

C: travelling cost per distance unit;

R: selling price of a recyclable material;

Q: vehicle capacity;

B: waste density;

M: Big-M number, i.e. a suitable large constant value;

d_{ij}: distance between $i \in \mathcal{I}$ and $j \in \mathcal{I}$;

S^{init}_i: percentage of waste on the total volume of bin $i \in \mathcal{I}'$ at the first stage;

E_i: capacity of bin $i \in \mathcal{I}'$;

$pa(n)$: parent of node $n \in \mathcal{N}^t$, $t \in \mathcal{T}''$.

Stochastic parameters:

a^n_i: uncertain accumulation rate of bin $i \in \mathcal{I}'$ at node $n \in \mathcal{N}^t$, $t \in \mathcal{T}''$ (percentage on the total volume of the bin);

π^n: probability of node $n \in \mathcal{N}^t$, $t \in \mathcal{T}$.

Decision variables:

$x^t_{ij} \in \{0, 1\}$: binary variable indicating if arc (i, j) is visited at time $t+1$, with $t \in \mathcal{T}'$ and for $i, j \in \mathcal{I}$, $i \neq j$;

$y_i^t \in \{0,1\}$: binary variable indicating if waste bin $i \in \mathcal{I}'$ is visited at time $t+1$, with $t \in \mathcal{T}'$;

$f_{ij}^n \in \mathbb{R}^+$: nonnegative variable representing the flow between $i \in \mathcal{I}'$ and $j \in \mathcal{I}$, $i \neq j$, for $n \in \mathcal{N}^t$, $t \in \mathcal{T}''$;

$w_i^n \in \mathbb{R}^+$: nonnegative variable representing the amount of waste collected at bin $i \in \mathcal{I}'$, for $n \in \mathcal{N}^t$, $t \in \mathcal{T}''$;

$u_i^n \in \mathbb{R}^+$: nonnegative variable representing the amount of waste at bin $i \in \mathcal{I}'$, for $n \in \mathcal{N}^t$, $t \in \mathcal{T}$.

We propose the following stochastic multi-stage mixed integer linear programming model $\mathcal{M}$:

$$\max \quad R \sum_{t \in \mathcal{T}''} \sum_{n \in \mathcal{N}^t} \pi^n \sum_{i \in \mathcal{I}'} w_i^n - C \sum_{t \in \mathcal{T}'} \sum_{\substack{i,j \in \mathcal{I} \\ i \neq j}} d_{ij} x_{ij}^t \tag{4.1}$$

$$\text{s.t.} \quad \sum_{\substack{j \in \mathcal{I} \\ j \neq i}} f_{ij}^n - \sum_{\substack{j \in \mathcal{I}' \\ j \neq i}} f_{ji}^n = w_i^n \qquad i \in \mathcal{I}', n \in \mathcal{N}^t, t \in \mathcal{T}'' \tag{4.2}$$

$$f_{ij}^n \leqslant (Q - E_j B a_j^n) x_{ij}^{t-1} \qquad i,j \in \mathcal{I}', i \neq j, n \in \mathcal{N}^t, t \in \mathcal{T}'' \tag{4.3}$$

$$f_{i0}^n \leqslant Q x_{i0}^{t-1} \qquad i \in \mathcal{I}', n \in \mathcal{N}^t, t \in \mathcal{T}'' \tag{4.4}$$

$$f_{ij}^n \leqslant Q - w_j^n \qquad i,j \in \mathcal{I}', i \neq j, n \in \mathcal{N}^t, t \in \mathcal{T}'' \tag{4.5}$$

$$f_{ij}^n \geqslant w_i^n - M(1 - x_{ij}^{t-1}) \qquad i \in \mathcal{I}', j \in \mathcal{I}, i \neq j, n \in \mathcal{N}^t, t \in \mathcal{T}'' \tag{4.6}$$

$$\sum_{\substack{j \in \mathcal{I} \\ j \neq i}} x_{ij}^t = y_i^t \qquad i \in \mathcal{I}', t \in \mathcal{T}' \tag{4.7}$$

$$\sum_{\substack{i \in \mathcal{I} \\ i \neq j}} x_{ij}^t = y_j^t \qquad j \in \mathcal{I}', t \in \mathcal{T}' \tag{4.8}$$

$$\sum_{i \in \mathcal{I}'} x_{i0}^t = \sum_{j \in \mathcal{I}'} x_{0j}^t \qquad t \in \mathcal{T}' \tag{4.9}$$

$$w_i^n \leqslant E_i B y_i^{t-1} \qquad i \in \mathcal{I}', n \in \mathcal{N}^t, t \in \mathcal{T}'' \tag{4.10}$$

$$u_i^n \leqslant M(1 - y_i^{t-1}) \qquad i \in \mathcal{I}', n \in \mathcal{N}^t, t \in \mathcal{T}'' \tag{4.11}$$

$$u_i^n = E_i B S_i^{init} \qquad i \in \mathcal{I}', n \in \mathcal{N}^1 \tag{4.12}$$

$$u_i^n = u_i^{pa(n)} + E_i B a_i^n - w_i^n \qquad i \in \mathcal{I}', n \in \mathcal{N}^t, t \in \mathcal{T}'' \tag{4.13}$$

$$u_i^{pa(n)} \leqslant \left(1 - a_i^n\right) E_i B \qquad i \in \mathcal{I}', n \in \mathcal{N}^t, t \in \mathcal{T}'' \tag{4.14}$$

$$x_{ij}^t \in \{0,1\} \qquad i,j \in \mathcal{I}, i \neq j, t \in \mathcal{T}' \tag{4.15}$$

$$y_i^t \in \{0,1\} \qquad i \in \mathcal{I}', t \in \mathcal{T}' \tag{4.16}$$

$$f_{ij}^n \geqslant 0 \qquad i \in \mathcal{I}', j \in \mathcal{I}, i \neq j, n \in \mathcal{N}^t, t \in \mathcal{T}'' \tag{4.17}$$

$$w_i^n \geqslant 0 \qquad i \in \mathcal{I}', n \in \mathcal{N}^t, t \in \mathcal{T}'' \tag{4.18}$$

$$u_i^n \geqslant 0 \qquad i \in \mathcal{I}', n \in \mathcal{N}^t, t \in \mathcal{T} \tag{4.19}$$

The objective function (4.1) is composed by the following terms: (i) the revenues from selling the expected collected waste and (ii) the transportation cost, depending on the routing plan and on the total travelled distance. Constraints (4.2) guarantee the flow balance at each waste bin i, for every node $n \in \mathcal{N}^t$ and for every period $t \in \mathcal{T}''$. Constraints (4.3) to (4.5) provide upper bounds on the flow variables f_{ij}^n, for each node $n \in \mathcal{N}^t$ at stage $t \in \mathcal{T}''$. Specifically, constraints (4.3) guarantee that if bins i and j are not connected, then the waste flow between them is zero; otherwise, its sum with the uncertain accumulation amount of waste at j cannot exceed the vehicle capacity. Similarly for constraints (4.4) as far as it concerns the flow between bin i and the depot, once the arc $(i, 0)$ is activated: the vehicle cannot transport to the depot more waste than its capacity. Finally, constraints (4.5) ensure that the sum of the waste flow between bins i and j and the amount of waste collected at bin j cannot exceed the vehicle capacity. Constraints (4.6) provide lower bounds on the flow variable f_{ij}^n such that if the vehicle travels from bin i to bin j or from bin i to the depot, with $n \in \mathcal{N}^t$ and $t \in \mathcal{T}''$, all of the accumulated amount of waste at bin i should be collected. Constraints (4.7) and (4.8) link together the decision variables x_{ij}^t and y_i^t for each stage $t \in \mathcal{T}'$ and ensure that, if bin i is visited, then there exists exactly one route reaching and one route leaving i; on the other hand, no visits at bin i imply no incoming edges to and no outgoing edges from i. Constraints (4.9) impose the depot's balance by enforcing that the numbers of incoming and outgoing edges are the same for every period $t \in \mathcal{T}'$. This means that, whether the vehicle performs a route starting from the depot, then it must return to the depot. Constraints (4.10) ensure that the collection amount w_i^n at bin i in node $n \in \mathcal{N}^t$, for $t \in \mathcal{T}''$ must be zero, unless the bin is visited. Constraints (4.11) guarantee that the amount of waste u_i^n at bin i at node $n \in \mathcal{N}^t$ and stage $t \in \mathcal{T}''$ must be zero if the bin is visited. Constraints (4.12) fix the initial amount of waste u_i^n at bin i at the root of the scenario tree. Constraints (4.13) update at every node $n \in \mathcal{N}^t$ and for every period $t \in \mathcal{T}''$ the amount of waste u_i^n at bin i by incorporating the uncertain accumulated amount of waste and, potentially, by subtracting the amount of collected waste w_i^n. Constraints (4.14) impose that no bins are allowed to overflow at each node $n \in \mathcal{N}^t$, $t \in \mathcal{T}''$. Finally, constraints from (4.15) to (4.19) define the decision variables of the problem. We denote by z^* the optimal expected profit of model $\mathcal{M}$.

4.3.1 A two-commodity flow model

In model $\mathcal{M}$, the distances between two locations are considered as asymmetric, i.e. in general $d_{ij} \neq d_{ji}$, for $i, j \in \mathcal{I}$. This assumption impacts not only the objective function (4.1), but also both the constraints (4.2) and (4.4)-(4.6) related to the flow variables f_{ij}^n,

and the degree constraints (4.7)-(4.9) on variables x_{ij}^t and y_i^t. This leads to an increase of the size of the model, due to a considerable number of inequality constraints.

In practical cases, however, distance d_{ij} and d_{ji} may not be significantly different and considering a symmetric distance matrix does not result in a considerable worsening of the solution. For this reason, we design an alternative version of model $\mathcal{M}$, denoted by $\mathcal{M}_{sym}$, based on the two-commodity flow formulation proposed in [6] and applied to a waste collection problem in [127]. Hence, a copy depot denoted by $N+1$ is introduced and each route is defined according to two paths: one from depot 0 to depot $N+1$, with variables f_{ij}^n representing the load of the vehicle, and one reversed, from depot $N+1$ to depot 0, with variables f_{ji}^n denoting the empty space of the vehicle (see Figure 4.2 for an illustrative example).

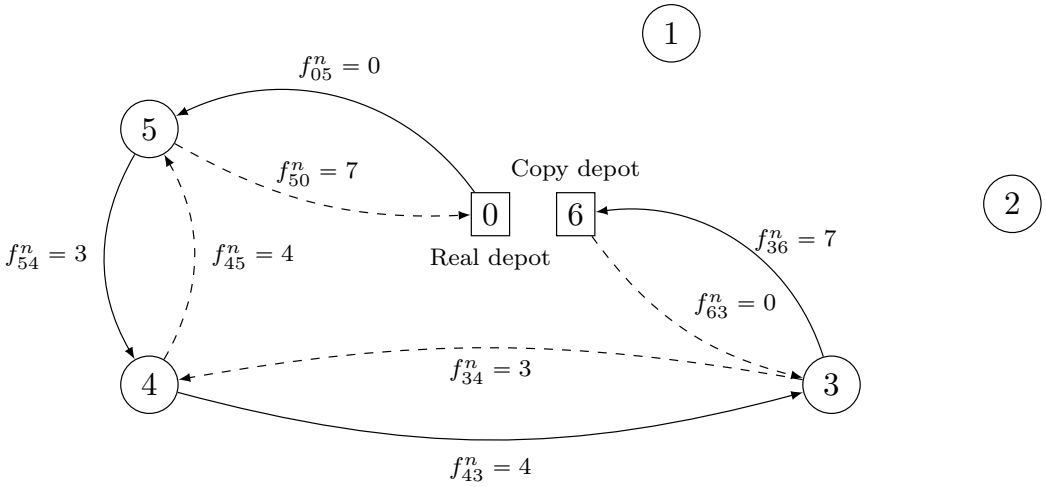


Figure 4.2: Representation of the two-commodity flow formulation on the same network of Figure 4.1. A copy depot (vertex 6) is introduced, and the truck capacity Q is set to 7. The solid lines represent the actual visiting sequence, starting from the real depot, with corresponding waste flows f_{ij}^n. The dashed lines are associated with the reverse flows f_{ji}^n, related to the empty space in the vehicle. Note that $f_{ij}^n + f_{ji}^n = Q$.

Each edge is therefore counted twice and the objective function (4.1) needs to be updated as:

$$\max \quad R \sum_{t \in \mathcal{T}''} \sum_{n \in \mathcal{N}^t} \pi^n \sum_{i \in \mathcal{I}'} w_i^n - \frac{C}{2} \sum_{t \in \mathcal{T}'} \sum_{\substack{i,j \in \mathcal{I} \\ i \neq j}} d_{ij} x_{ij}^t.$$

Constraints (4.2) are replaced by:

$$\sum_{\substack{j \in \mathcal{I} \\ j \neq i}} (f_{ij}^n - f_{ji}^n) = 2w_i^n \qquad i \in \mathcal{I}', n \in \mathcal{N}^t, t \in \mathcal{T}'',$$

since the two commodity flow formulation considers two flows passing through node i. In addition, constraints (4.4)-(4.6) are substituted by:

$$\sum_{i \in \mathcal{I}'} f^n_{iN+1} = \sum_{i \in \mathcal{I}'} w^n_i \qquad n \in \mathcal{N}^t, t \in \mathcal{T}'' \tag{4.20}$$

and

$$f^n_{ij} + f^n_{ji} = Q x^{t-1}_{ij} \qquad i, j \in \mathcal{I}, i \neq j, n \in \mathcal{N}^t, t \in \mathcal{T}''. \tag{4.21}$$

Constraints (4.20) ensure that the total inflow of the copy depot corresponds to the total amount of collected waste, whereas constraints (4.21) impose that, whenever an edge is traversed, the sum of the two traversing flows is equal to the capacity of the vehicle. Finally, the degree constraints (4.7)-(4.9) reduce to:

$$\sum_{\substack{i \in \mathcal{I} \\ i \neq j}} x^t_{ij} = 2y^t_j \qquad j \in \mathcal{I}', t \in \mathcal{T}'.$$

All the other constraints not mentioned remain unchanged when passing from model $\mathcal{M}$ to $\mathcal{M}_{sym}$. For the sake of completeness, the entire model formulation $\mathcal{M}_{sym}$ is reported in the Appendix B.1.

4.3.2 A polynomially solvable case

The proposed SIRP formulation is clearly NP-hard, since it can be reduced to the well-known NP-hard Travelling Salesman Problem (see [60]), whenever the time horizon is $\mathcal{T} = \{1, 2\}$, the selling price is null, i.e. $R = 0$, the capacity Q of the vehicle is infinite, and all the containers need to be visited at day 2 in order to avoid overflow.

On the other hand, the waste collection problem admits a polynomially solvable case whenever routing decisions are excluded from the problem. This will be addressed in the following proposition.

Proposition 3. If $C = 0$, then the optimal profit of model $\mathcal{M}$ is:

$$z^* = RB\left\{ \sum_{i \in \mathcal{I}'} E_i \left(S^{init}_i + \sum_{t \in \mathcal{T}''} \mathbb{E}\big[a^{(t)}_i\big] \right) \right\}, \tag{4.22}$$

where $\mathbb{E}\big[a^{(t)}_i\big]$ is the expected accumulation rate of waste at time $t \in \mathcal{T}''$ for bin $i \in \mathcal{I}'$.

Proof. We prove the proposition by induction on the time horizon T.

- (Base case) We consider the case of a two-stage problem ($T = 2$). Since $C = 0$ and

$\mathcal{T}'' = \{2\}$, profit (4.1) reduces to:

$$z = R \sum_{n \in \mathcal{N}^2} \pi^n \sum_{i \in \mathcal{I}'} w_i^n.$$

From constraints (4.12)-(4.13), it holds that:

$$w_i^n = E_i B S_i^{init} + E_i B a_i^n - u_i^n \qquad i \in \mathcal{I}', n \in \mathcal{N}^2,$$

which, substituting in the objective function, gives:

$$\begin{aligned} z &= R \sum_{n \in \mathcal{N}^2} \pi^n \sum_{i \in \mathcal{I}'} E_i B S_i^{init} + R \sum_{n \in \mathcal{N}^2} \pi^n \sum_{i \in \mathcal{I}'} E_i B a_i^n - R \sum_{n \in \mathcal{N}^2} \pi^n \sum_{i \in \mathcal{I}'} u_i^n \\ &= RB \sum_{i \in \mathcal{I}'} E_i (S_i^{init} + \mathbb{E}\big[a_i^{(2)}\big]) - R \sum_{n \in \mathcal{N}^2} \pi^n \sum_{i \in \mathcal{I}'} u_i^n, \end{aligned}$$

where we have applied $\sum_{n \in \mathcal{N}^2} \pi^n = 1$ and $\sum_{n \in \mathcal{N}^2} \pi^n a_i^n = \mathbb{E}\big[a_i^{(2)}\big]$. Moreover, we note that the objective function z is the difference of two nonnegative quantities, where the first one is constant. Thus:

$$\max z = RB \sum_{i \in \mathcal{I}'} E_i (S_i^{init} + \mathbb{E}\big[a_i^{(2)}\big]) - \min R \sum_{n \in \mathcal{N}^2} \pi^n \sum_{i \in \mathcal{I}'} u_i^n,$$

where the minimum of the second term is reached at $u_i^n = 0$, for all $n \in \mathcal{N}^2$, $i \in \mathcal{I}'$: the thesis is verified.

- (Inductive step) We assume that the thesis holds for a model with time horizon $T-1$. We need to prove that the thesis is verified for a model with time horizon T, whose objective function is:

$$R \sum_{t=2}^{T} \sum_{n \in \mathcal{N}^t} \pi^n \sum_{i \in \mathcal{I}'} w_i^n = R \sum_{t=2}^{T-1} \sum_{n \in \mathcal{N}^t} \pi^n \sum_{i \in \mathcal{I}'} w_i^n + R \sum_{n \in \mathcal{N}^T} \pi^n \sum_{i \in \mathcal{I}'} w_i^n.$$

Given the induction hypothesis, the optimal profit of the first addendum corresponds to $RB\left\{ \sum_{i \in \mathcal{I}'} E_i \left(S_i^{init} + \sum_{t=2}^{T-1} \mathbb{E}\big[a_i^{(t)}\big] \right) \right\}$. At stage T, from constraints (4.13), $w_i^n = E_i B a_i^n$, for all $i \in \mathcal{I}'$, $n \in \mathcal{N}^T$, since $u_i^{pa(n)} = 0$ for the induction hypothesis and $u_i^n = 0$ for the same reasoning of the base case. Consequently, we get:

$$z^* = RB\left\{ \sum_{i \in \mathcal{I}'} E_i \left(S_i^{init} + \sum_{t=2}^{T-1} \mathbb{E}\big[a_i^{(t)}\big] \right) \right\} + RB \sum_{i \in \mathcal{I}'} E_i \mathbb{E}\big[a_i^{(T)}\big],$$

which verifies the thesis. □

We conclude that parameters R and C have different roles: when $R = 0$ model $\mathcal{M}$ is NP-hard, whereas if $C = 0$ an optimal policy can be computed in $O(T)$ time. For this reason, given the computational complexity of the problem in the general case, heuristic methods are required. To cope with this issue, in the next section we consider the rolling horizon approach.

4.4 The rolling horizon approach and its worst-case analysis

One of the most classical heuristic algorithms for multi-stage stochastic programming models is the rolling horizon approach (see [31]): the multi-stage stochastic problem is decomposed in a sequence of subproblems with a fewer number W of consecutive periods (see [28]). This leads to a reduced computational effort because at each iteration of the algorithm the number of nodes considered in the scenario tree is fewer than the ones in the original multi-stage program. However, the quality of the solution may deteriorate since the time horizon is reduced and the solution may be suboptimal (see [12]). In the following we present the details of the approach.

First of all, we fix the reduced number W of consecutive period, with $1 \leqslant W < T - 1$. In the first iteration of the algorithm, the $(W + 1)$-stage stochastic programming model defined on $t = 1, \ldots, W + 1$ is solved, and the values of the first-stage decision variables x_{ij}^1 and y_i^1 and the second-stage variables w_i^n and u_i^n, for $n \in \mathcal{N}^2$, are stored. In the second iteration, the value of the inventory levels u_i^n for $n \in \mathcal{N}^2$ are fixed as the ones deduced from the first iteration. This is needed to keep track of the evolution of the process and to link two consecutive time periods. Then, the $(W + 1)$-stage stochastic programming model defined on $t = 2, \ldots, W + 2$ is solved and, as done before, the values of the second-stage decision variables x_{ij}^2 and y_i^2 and the third-stage variables w_i^n and u_i^n, for $n \in \mathcal{N}^3$, are stored. This process is repeated until the last iteration defined on stages $t = T - W, \ldots, T$ is performed. Then, a W-stage stochastic programming model defined on $t = T - W + 1, \ldots, T$ is solved and the same approach described above is applied. Next, a $(W - 1)$-stage stochastic programming model defined on $t = T - W + 2, \ldots, T$ is solved and the process is repeated until the last two-stage stochastic programming model defined on $t = T - 1, T$.

Once the $T - 1$ stochastic programming models have been solved, the variables x_{ij}^t, y_i^t for all $t \in \mathcal{T}'$ and w_i^n for all $n \in \mathcal{N}^t$ and $t \in \mathcal{T}''$ are obtained. The corresponding value of the objective function (4.1) is then computed, leading to $z^{RH,W}$.

Schematically, the algorithm can be represented as in Pseudocode 2.

Pseudocode 2 The rolling horizon approach for model $\mathcal{M}$

Input: T, $1 \leqslant W < T-1$

$k \leftarrow 1$, $l \leftarrow W+1$

$u_i^{+n*} \leftarrow u_i^1$, $n \in \mathcal{N}^t$, $t = k$

while $k \leqslant T - W$ **do**

  Solve $(W+1)$-stage SP on $t = k, \ldots, l$ with $u_i^{+n} \leftarrow u_i^{+n*}$, $n \in \mathcal{N}^t$, $t = k$

  Store x_{ij}^{t*}, y_i^{t*}, $t = k$ and u_i^{n*}, w_i^{n*}, $n \in \mathcal{N}^t$, $t = k+1$

  $k \leftarrow k+1$, $l \leftarrow l+1$

end while

$j \leftarrow 1$

while $k \leqslant T-1$ **do**

  Solve $(W+1-j)$-stage SP on $t = k, \ldots, T$ with $u_i^{+n} \leftarrow u_i^{+n*}$, $n \in \mathcal{N}^t$, $t = k$

  Store x_{ij}^{t*}, y_i^{t*}, $t = k$ and u_i^{n*}, w_i^{n*}, $n \in \mathcal{N}^t$, $t = k+1$

  $k \leftarrow k+1$, $j \leftarrow j+1$

end while

Return the corresponding value of the objective function (4.1).

We now perform a worst-case analysis of this approach. The following results hold true:

Theorem 4. If $C = 0$, then:

$$\frac{z^{RH,W}}{z^*} = 1,$$

for every choice of $W = 1, \ldots, T-2$.

Proof. Consider the case of $W = 1$, where $T-1$ two-stage stochastic optimization models have to be solved. Since all the subproblems do not share any overlapping period, denoting with $z_{t,t+1}^{RH,1}$ the optimal objective value on time period $t, t+1$, the optimal profit is:

$$\begin{aligned} z^{RH,1} &= z_{1,2}^{RH,1} + z_{2,3}^{RH,1} + \ldots + z_{T-1,T}^{RH,1} \\ &= RB\left\{ \sum_{i \in \mathcal{I}'} E_i \left(S_i^{init} + \mathbb{E}\big[a_i^{(2)}\big] + \mathbb{E}\big[a_i^{(3)}\big] + \mathbb{E}\big[a_i^{(T)}\big] \right) \right\}. \end{aligned}$$

The previous expression coincides with (4.22), and so the thesis is verified.

When considering a value $W > 1$, only the collecting variable w_i^n at the second stage of each subproblem are stored, implying that exclusively the accumulation rates at that

stage are considered in the optimal solution. This implies the thesis in a similar fashion as $W = 1$. □

On the other hand, when $R = 0$, the following result on the performance of the rolling horizon approach with $W = 1$ holds.

Theorem 5. There exists a class of instances such that $z^{RH,1} = -\infty$, even if model $\mathcal{M}$ is feasible.

Proof. Consider the following class of instances: initial amount of waste $S_i^{init} = 0$ for all $i \in \mathcal{I}'$; vehicle capacity $Q > \sum_{i \in \mathcal{I}'} E_i$; selling price $R = 0$.
For all $i \in \mathcal{I}'$, let $\alpha_i \in (0; 1)$, $\varepsilon_i \in (0; \alpha_i E_i]$, and the accumulation rate a_i^n be such that:

$$a_i^n = \begin{cases} 0 & \text{if } n \in \mathcal{N}^t,\ t \in \mathcal{T}' \cup \{T-2\} \\ \alpha_i E_i & \text{if } n \in \mathcal{N}^{T-1} \\ (1-\alpha_i)E_i + \varepsilon_i & \text{if } n \in \mathcal{N}^T. \end{cases} \tag{4.23}$$

The graph of a_i^n is depicted in Figure 4.3.

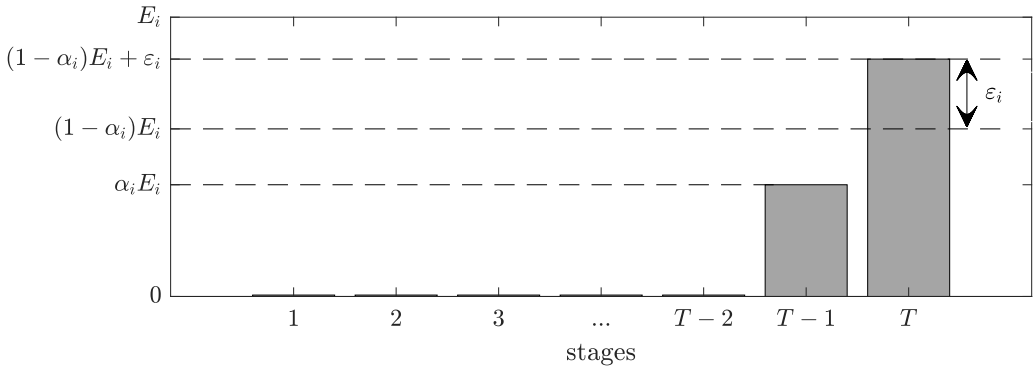


Figure 4.3: Graph of the accumulation rate (4.23).

Let us apply the rolling horizon approach with $W = 1$. This means that $T - 1$ two-stage stochastic programming models have to be solved. In the first $T - 3$ programs, all the decision variables are zero, since there is no waste to collect. Similarly for the model defined on stages $t = T - 2, T - 1$, with the exception of u_i^{T-1} which is equal to $\alpha_i E_i$. However, the last optimization model defined on stages $t = T - 1, T$ is infeasible because each container i will incur into overflowing, by violating constraints (4.14). This implies that $z^{RH,1} = -\infty$.

On the other hand, the optimal profit z^* of the multi-stage stochastic optimization model will be equal to $-C \sum_{\substack{i,j \in \mathcal{I} \\ i \neq j}} d_{ij} x_{ij}^{T-1}$, deriving from a collection on day T. □

Making the appropriate changes, a similar performance of the rolling horizon approach with $W = 1$ also holds within model $\mathcal{M}_{sym}$.

4.5 Computational results

In this section, we first describe the instances on which we perform numerical simulations (see Subsection 4.5.1). Subsection 4.5.2 compares the solutions of models $\mathcal{M}$ and $\mathcal{M}_{sym}$. In Subsection 4.5.3, the validation of model $\mathcal{M}$ with standard stochastic measures is provided, and the quality of the expected value solution is discussed. In Subsection 4.5.4, the performance of the rolling horizon approach is assessed. Then, the results on a large case study are presented in Subsection 4.5.5. Finally, managerial insights are provided in Subsection 4.5.6.

All computational experiments are obtained using GAMS 38.3.0 and solver Gurobi 9.5 on an Intel(R) Core(TM) i5-8500 64-bit machine, with 8 GB RAM and 3.00 GigaHertz processor. Unless otherwise specified, a runtime limit of 24h is imposed.

4.5.1 Data analysis

The data considered in this study are inspired by a real case problem provided by the industrial partner *ERSUC - Resíduos Sólidos do Centro, S.A.*, one of the main waste management companies in Portugal. The company operates in the Central Region of Portugal and owns a homogeneous fleet of vehicles based at two different depots, one near the city of *Aveiro* and the other close to *Coimbra*. The recyclables collection is performed independently for each type of waste (glass, paper/cardboard and plastic/metal).

The case study described in the following focuses on the collection of plastic/metal waste, related to packaging materials, around the suburban municipality of *Condeixa-a-Nova* in the district of *Coimbra*. The simulations we perform are inspired by real data provided by *ERSUC* on the filling rate of 121 waste bins between April and July 2019 (15 weeks). The data are gathered by the garbage collector only on the collection days (20 days in total). The working days of the company include all the days of a week, except Sunday. Therefore, the time horizon is $\mathcal{T} = \{1, \ldots, 6\}$.

We perform simulations on small and large instances. As far as it concerns the small cases, we generate a set of thirty instances, randomly drawn from the entire dataset of 121 bins, with a reduced number of bins. For simplicity, we denote each small instance on the basis of the coding scheme "*inst_draw_numbins*", where *draw* is an integer between 1 and 10 associated with the random draw, and *numbins* is the number of selected bins

(9, 10 or 11). In addition, we consider a large instance composed by 50 bins to simulate a real case study of waste collection, since fifty is the average number of bins in a collection route of the industrial partner.

The deterministic parameters of the model are shown in Table 4.1.

Parameter	Value	Source
C	1 €/km	*ERSUC*
R	0.30 €/kg	*Sociedade Ponto Verde*
Q	2000 kg	*ERSUC*
B	30 kg/m^3	*ERSUC*
M	10^5	-
d_{ij}, $i, j \in \mathcal{I}$	Actual road distance between i and j	*ERSUC* and *OpenRouteService*
S_i^{init}, $i \in \mathcal{I}'$	Initial percentage of waste on the total volume of bin i	*ERSUC*
E_i, $i \in \mathcal{I}'$	2.5 m^3	*ERSUC*

Table 4.1: Parameters values and sources.

As in [127], transportation cost C includes fuel consumption, maintenance of the vehicle and drivers' wages. The revenue parameter R is derived as follows: for each ton of packaging collected and sorted, the *Sociedade Ponto Verde* (the packaging waste regulator in Portugal) pays 545 €/ton to the waste collection company; since only the collection activity is being considered in this work, which corresponds to approximately the 55% of the total cost, the selling price R is adjusted to 0.30 €/kg.

We discuss now how we construct the random process $\{a_i^{(t)}\}_{t=1}^{T}$ of the daily accumulation rate, based on observations provided by the industrial partner. For each bin i, we have the historical data of the filling rate on collection days, that we denote by $\{p_i^{(t)}\}_{t=1}^{20}$. For all $i \in \mathcal{I}'$, we set S_i^{init} equal to $p_i^{(1)}$. We assume that, if t_1 and t_2 are two consecutive collection days, the increase (or decrease) of the filling rate of waste between t_1 and t_2 is constant. Once the waste collector visits bin i at time t_1, then she/he empties it. This implies that the daily accumulation rate of waste in bin i can be calculated as:

$$a_i^{(t)} = \frac{p_i^{(t_2)} - p_i^{(t_1)}}{t_2 - t_1} = \frac{p_i^{(t_2)}}{t_2 - t_1}, \qquad t = t_1 + 1, \ldots, t_2.$$

By following this procedure, for each bin $i \in \mathcal{I}'$, a complete trajectory of the stochastic process $\{a_i^{(t)}\}_{t=1}^{T}$ is obtained on a daily basis.

In the Appendix B.2 we report the scenario tree generation procedure we adopt, along with an in-sample stability analysis on the number of scenarios to be considered in the scenario tree (see the Appendix B.3). In the remainder of the chapter, we show the results obtained on a tree with 32 scenarios and 63 nodes.

4.5.2 A comparison of models $\mathcal{M}$ and $\mathcal{M}_{sym}$ solutions

Solving either model $\mathcal{M}$ or model $\mathcal{M}_{sym}$ to optimality on the whole dataset of 121 bins is not possible on our machine, given the high number of variables and constraints (see Table 4.3). When considering reduced instances, in small cases composed by 9, 10 or 11 bins, models $\mathcal{M}$ and $\mathcal{M}_{sym}$ provide the same policy in terms of bin selection, visiting schedule, routing and consequent weight of collected waste (see row 7 in Table 4.2). However, due to the assumption on the symmetric distance matrix, the profit with model $\mathcal{M}_{sym}$ is less accurate. On the other hand, on the large instance with 50 bins model $\mathcal{M}_{sym}$ outperforms model $\mathcal{M}$, since the optimality gap is much smaller (see Table 4.3). For this reason, in the following we will show results obtained with model $\mathcal{M}$ on small instances, whereas larger instances results rely on model $\mathcal{M}_{sym}$.

Number of bins	9 ($\mathcal{M}$)	9 ($\mathcal{M}_{sym}$)	10 ($\mathcal{M}$)	10 ($\mathcal{M}_{sym}$)	11 ($\mathcal{M}$)	11 ($\mathcal{M}_{sym}$)
Binary variables	495	595	600	710	715	835
Continuous variables	6705	7263	8070	8690	9559	10241
Equality constraints	1220	8052	1355	9546	1490	11164
Inequality constraints	16182	6138	19840	7440	23870	8866
Profit (€)	18.16	18.45	34.37	34.67	41.25	41.40
Weight of collected waste (kg)	478.15	478.15	513.83	513.83	612.48	612.48
Travelled distance (km)	125.28	124.99	119.78	119.46	142.50	142.25
Ratio weight/distance (kg/km)	3.80	3.80	4.32	4.33	4.31	4.32
CPU time (s)	1434.00	15.60	1382.00	32.80	3259.40	37.50

Table 4.2: Average results from solving models $\mathcal{M}$ and $\mathcal{M}_{sym}$ on small instances.

Number of bins	50 ($\mathcal{M}$)	50 ($\mathcal{M}_{sym}$)	121 ($\mathcal{M}$)	121 ($\mathcal{M}_{sym}$)
Binary variables	13000	13510	74415	75635
Continuous variables	164350	167450	930369	937871
Equality constraints	6755	170986	16340	946164
Inequality constraints	471200	161200	2738230	922746
Profit (€)	501.29	581.59	–	–
Weight of collected waste (kg)	2306.69	2585.16	–	–
Travelled distance (km)	190.72	193.96	–	–
Ratio weight/distance (kg/km)	12.09	13.33	–	–
CPU time (s)	86400 (20.82%)	86400 (2.77%)	OOM	OOM

Table 4.3: Average results from solving models $\mathcal{M}$ and $\mathcal{M}_{sym}$ on large instances. When the time limit is reached, the relative optimality gap in percentage is reported in brackets. OOM stands for "Out-Of-Memory".

4.5.3 The impact of uncertainty and the quality of the deterministic solution

The purpose of this section is twofold. Firstly, we discuss the importance of stochasticity in model $\mathcal{M}$ by comparing the stochastic formulation (i.e. the *Recourse Problem*, (RP)) with the perfect information case (the so-called *Wait and See* approach, (WS)) through the *Expected Value of Perfect Information* $(EVPI)$ (see [19]). Secondly, we show the benefits of taking into account stochasticity in model $\mathcal{M}$ with respect to its deterministic counterpart (the so-called *Expected Value* problem, (EV)), by considering the *Value of Stochastic Solution at stage* t (VSS^t), with $t \in \mathcal{T}'$ (see [95]). In this direction, we specify our analysis by including the *Multi-stage Loss Using the Skeleton Solution until stage* t $(MLUSS^t)$ and the *Multi-stage Loss of Upgrading the Deterministic Solution until stage* t $(MLUDS^t)$, with $t \in \mathcal{T}'$ (see [95]).

In the perfect information case, the realization of the accumulation rate of waste in all of the bins is known at the first stage. Then, the $\%EVPI$ is calculated according to the formula:

$$\%EVPI := (WS - RP)/RP.$$

Results are reported in the second column of Table 4.4, where we see that, on average, the $EVPI$ is 81% of the RP. This means that, for reaching perfect information on the accumulation rate, the decision maker would be ready to pay at most 81% of the profit. Detailed results for each instance are shown in the Appendix B.4.

Size	$\%EVPI$	$\%VSS^t$, $1 \leqslant t \leqslant 5$	$\%MLUSS^t$, $1 \leqslant t \leqslant 5$	$\%MLUDS^1$	$\%MLUDS^t$, $2 \leqslant t \leqslant 4$	$\%MLUDS^5$
9 bins	188%	∞	∞	8%	674%	681%
10 bins	36%	∞	∞	0%	175%	175%
11 bins	20%	∞	∞	0%	158%	158%
Average	81%	∞	∞	3%	336%	338%

Table 4.4: Summary results of stochastic measures $\%EVPI$, $\%VSS^t$, $\%MLUSS^t$, $\%MLUDS^t$, for $1 \leqslant t \leqslant 5$, expressed in percentage gap to the corresponding RP problem.

Furthermore, in a simpler approach the decision maker may replace the accumulation rate of waste by its expected value, and solve the deterministic EV program. In a multi-stage context, the $\%VSS^t$ measures the expected gain from solving the stochastic model $\mathcal{M}$ rather than its deterministic counterpart up to stage t, and it is calculated as:

$$\%VSS^t := (RP - EEV^t)/RP, \quad t = 1, \ldots, T-1.$$

EEV^t is the *Expected result of using the EV solution until stage* t and denotes the objective function value of the RP model where the decision variables x_{ij}^t and y_i^t on the

routing until stage t are fixed at optimal values obtained by solving the EV problem. In the great majority of the instances, the EEV^t problems are infeasible, due to the violation of the non-overflowing constraints (4.14). Thus, the corresponding $\%VSS^t$ is infinite, already at the first stage (see column 3 of Table 4.4). In fact, by taking the average on the accumulation rate, in the EV problem the data do not support a collection at stage 2, by postponing it later in the planning horizon. On the other hand, the RP model may require a collection, at least, on day 2, since the accumulation rate is significant for certain bins, but this is clearly in contradiction with the previous condition. The results discussed so far justify the adoption of a stochastic model compared to a deterministic setting when dealing with a problem of waste collection.

In the following, we further investigate if the deterministic solution still carries useful information for the stochastic case. To achieve this purpose, firstly we compute the *Multi-stage Expected Skeleton Solution Value at stage t* ($MESSV^t$), by solving the RP model having fixed at zero all the routing variables x_{ij}^t and y_i^t that are zero in the EV problem until stage t. This allows to test whether the deterministic model produces the correct non-zero variables. Once the $MESSV^t$ is computed, it is compared with RP by introducing the *%Multi-stage Loss Using Skeleton Solution until stage t* ($MLUSS^t$), expressed as:

$$\%MLUSS^t := (RP - MESSV^t)/RP, \quad t = 1, \ldots, T-1.$$

The results in Table 4.4 and in the Appendix B.4 show that $\%VSS^t$ coincides with $\%MLUSS^t$ for all $t = 1, \ldots, 5$, both in the case of infiniteness and finiteness of $\%VSS^t$. On one hand, whenever $\%VSS^t$ is infinite already at stage 1, the EV problem is not able to identify the overflowing bins at the first stage because average data do not support collection, resulting in an infeasibility of the $MESSV^t$ problem. On the other hand, if $\%VSS^t$ is finite (see the Appendix B.4), the model correctly selects the overflowing bins, and thus the great majority of x_{ij}^t variables are equal to zero at stage t. This implies straightforwardly that there is only one possible choice for the vehicle to visit the selected bins.

Finally, we carry out an analysis regarding the upgradeability of the expected value solution to become good, or optimal, in the stochastic setting. Specifically, we consider the EV solution $\bar{x}_{ij}^t$, $\bar{y}_i^t$ until stage t as a starting point in the RP model, by adding the constraints $x_{ij}^t \geqslant \bar{x}_{ij}^t$, for all $i, j \in \mathcal{I}$, and $y_i^t \geqslant \bar{y}_i^t$, for all $i \in \mathcal{I}'$ up to stage t. The corresponding optimal value is denoted as *Multi-stage Expected Input Value until stage t* ($MEIV^t$). From this measure, the *%Multi-stage Loss of Upgrading the Deterministic*

Solution until stage t ($\%MLUDS^t$) is defined as follows:

$$\%MLUDS^t := (RP - MEIV^t)/RP, \quad t = 1, \ldots, T-1.$$

As it is reported in Table 4.4, $\%MLUDS^1$ is close to zero on average. Indeed, only in *inst_4_9* (see the Appendix B.4) $\%MLUDS^1$ is strictly positive, whereas in all the other instances it is zero. On one hand, this situation is due to a collection later than stage 2 in the EV solution, with conditions $x_{ij}^1 \geqslant 0$, for all $i, j \in \mathcal{I}$, and $y_i^1 \geqslant 0$, for all $i \in \mathcal{I}'$, automatically satisfied by constraints (4.15)-(4.16) in the $MEIV^1$ problem. On the other hand, at stage 2 the EV problem imposes a collection on a subset of bins with respect to the RP problem and, thus, constraints (4.16) are themselves satisfied in the $MEIV^1$ problem.

The large values of $\%MLUDS^t$, with $t = 2, \ldots, 5$, depend on the fact that the corresponding $MEIV^t$ problems have collections at the two consecutive stages 1 and 2, due to the additional constraints on the EV solution at stage 2. Given the non-significant weight of waste in the bins because already emptied at the previous stage, the profits $MEIV^t$ turn to be negative, and $\%MLUDS^t$ very high.

From this analysis, it can be concluded that the deterministic solution may be taken as input in the stochastic model only in the first stage, whereas in the next stages the EV solution is no longer to be upgradeable.

4.5.4 Performance of the rolling horizon approach

In this section, we evaluate the performance of the rolling horizon approach described in Section 4.4. Since model $\mathcal{M}$ is NP-hard and with large instances obtaining the optimal solution may be challenging (see Table 4.3), we apply the rolling horizon approach over a reduced time horizon. Instead of solving a T-stage stochastic program, this heuristic algorithm requires to solve a sequence of $T-1$ subproblems over a reduced number of stages. In our case study, model $\mathcal{M}$ is a six-stage stochastic optimization program and, thus, the reduced number of periods W is an integer between 1 and 4.

In Figure 4.4 we depict with vertical bars the average performance of the rolling horizon approach in terms of percentage gap between the RP solution and the heuristic solution. As highlighted in Section 4.4, when $W = 1$ the rolling horizon approach may be in principle infinitely suboptimal: over the thirty instances, five of them exhibit infeasibility in the first two-stage problem with $W = 1$. On the other hand, when we consider the remaining twenty-five instances for which there is no infeasibility, on average the profit gap is 29.53%

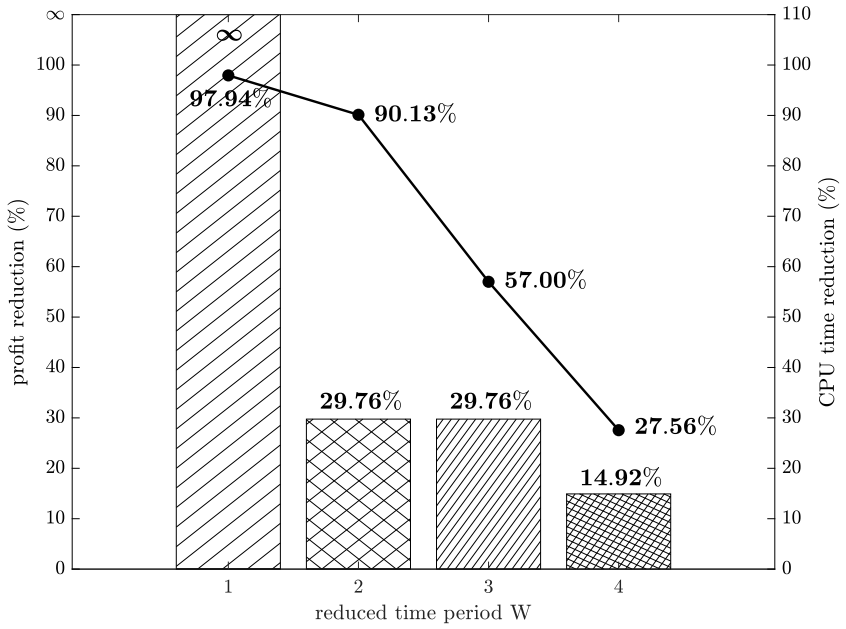


Figure 4.4: Performance of the rolling horizon approach. The vertical bars represent the profit percentage reduction when applying the rolling horizon approach (left-hand scale). The results show the average over the thirty instances. When $W = 1$, due to infeasibility, the reduction may be infinite. The solid line refers to the CPU time percentage reduction to solve at optimality with the rolling horizon approach, compared to the original six-stage program (right-hand scale).

when compared to the RP solution. Furthermore, we note that the result obtained with $W = 2$ and $W = 3$ (29.76% in both cases) is very similar to the one obtained for $W = 1$ when infeasibility does not occur. However, with $W = 2, 3$ no infeasibility issues arise in any instance. Finally, the performance improves when $W = 4$, with an average profit gap of 14.92%.

Regarding the computational time, we report as well in Figure 4.4 the results in terms of percentage reduction with respect to the six-stage model. We notice that, as W increases, the CPU time required to solve at optimality the five subproblems with the rolling horizon approach increases too. Specifically, when $W = 1$ and $W = 2$ the average savings are 97.94% and 90.13% of the computational time, respectively. If $W = 3$, even in the face of the same performance in terms of profit as $W = 2$, the CPU time is much larger, with a reduction of the 57.00%. Lastly, the great similarity of the profit between the RP model and the case with $W = 4$ requires a significant effort to be reached, with an average CPU time saving around 27.56%. Detailed results on the performance of the rolling horizon approach are presented in the Appendix.

From the previous analysis, we conclude that the rolling horizon approach is effective for the proposed six-stage model. As expected, the performance of the algorithm strongly

depends on the size of the reduced time horizon. If the decision maker requires a good accuracy in a short time, $W = 2$ is the best candidate. On the other hand, if she/he is willing to wait, $W = 4$ attains better results but with more computational time.

4.5.5 A real case study

In this section, we present the results of the simulations in a real case study. We consider a large instance composed by 50 bins randomly chosen from the original set of 121 waste containers.

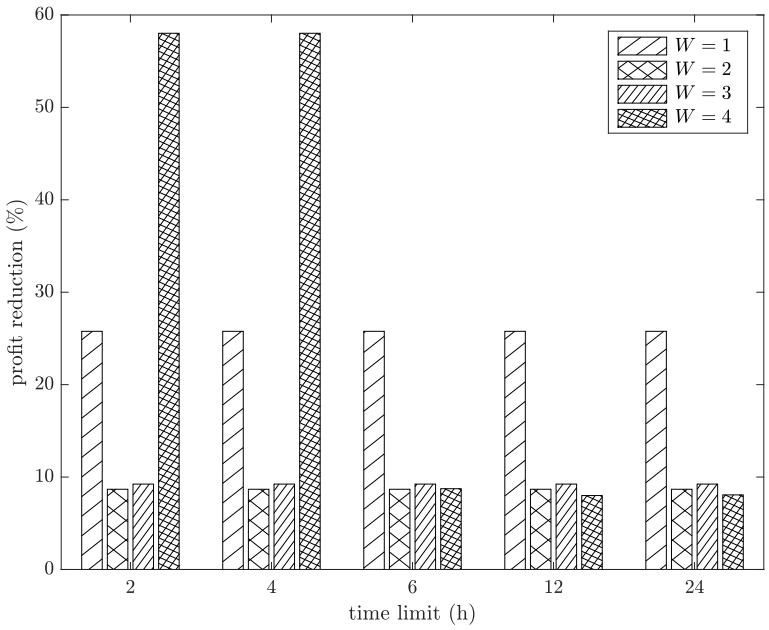


Figure 4.5: Performance of the rolling horizon approach for the instance with 50 bins in terms of profit reduction.

Given the high number of variables and of constraints (see Table 4.3), Gurobi is not able to solve at optimality model $\mathcal{M}$ within the time limit of 86400 seconds (one day), with a resulting relative optimality gap of 20.82%. However, when considering model $\mathcal{M}_{sym}$ with distances the average between d_{ij} and d_{ji}, the results are significantly better, with an optimality gap of 2.77% (see Table 4.3).

For this reason, we consider model $\mathcal{M}_{sym}$ for the study of the large real case instance. Particularly, we investigate how the rolling horizon approach performs in this situation, with a reduced time limit. Indeed, from a managerial perspective, the time limit of one day is excessively high. Thus, we fix a time limit TL of 2, 4, 6, 12, 24 hours on the whole algorithm and, then, we set a time limit TL_{sub} for each of the corresponding subproblem,

as $TL_{sub} = \frac{TL}{\#\ subproblems}$. Following the approach of [28], after solving a subproblem, if some time is left, we add the remaining time to the following subproblem to be solved.

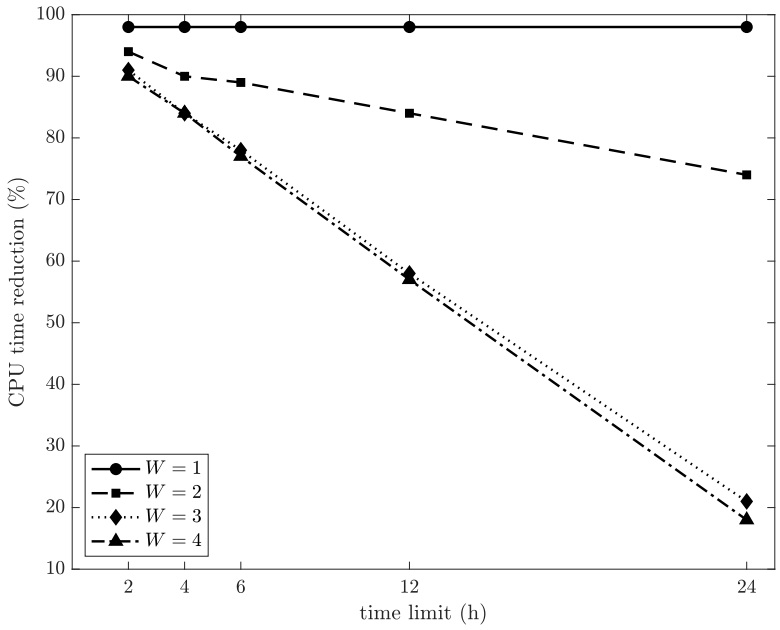


Figure 4.6: Performance of the rolling horizon approach for the instance with 50 bins in terms of CPU time reduction.

Figure 4.5 displays the percentage profit reduction, when applying the rolling horizon approach over a reduced time period W and with different time limits TL. The reduction is with respect to the feasible value obtained when solving the RP model. On one hand, we notice that, regardless of the time limit, the rolling horizon approach with $W = 1, 2, 3$ does not improve its performance. Specifically, when $W = 1$ the profit reduction is 25.77%, while for $W = 2$ and $W = 3$ the reduction is similar (8.68% and 9.24%, respectively). On the other hand, when the time limit is enlarged, the rolling horizon approach with $W = 4$ shows an enhancement on the results: from 58.02%, when the time limit is low (2 and 4 hours), to 8.07% with TL equal to 12 and 24 hours. In this case, the bad performances with low time limits are due to the large size of the first two subproblems, defined respectively on stages 1-5 and 2-6, which are difficult to solve in a short time (24 or 48 minutes).

Similarly to the analysis carried out on the small instances, in Figure 4.6 we depict the percentage CPU time reduction of the rolling horizon approach. When $W = 1$, the reduction is constant, independently of the time limit. Indeed, the five subproblems are solved in less time than the time limit. Next, if $W = 2$, the reduction is high with a time limit of 2 hours (93.78%) and it reaches a minimum with $TL = 24$ of 73.91%. Finally, the situations with $W = 3$ and $W = 4$ show the same behaviour, which is almost

linear. Indeed, in these cases, the time limit is always reached, because of the size of the subproblems.

By combining all the previous results, we conclude that it is worth applying the rolling horizon approach when solving large instances for the stochastic waste collection problem. The performance depends on the reduced time horizon and on the time limit set by the user, but the rolling horizon approach with $W = 2$ and time limit 2 hours is a good trade-off between accuracy and time savings. In Figure 4.7 we depict the route obtained with these choices of the parameters.

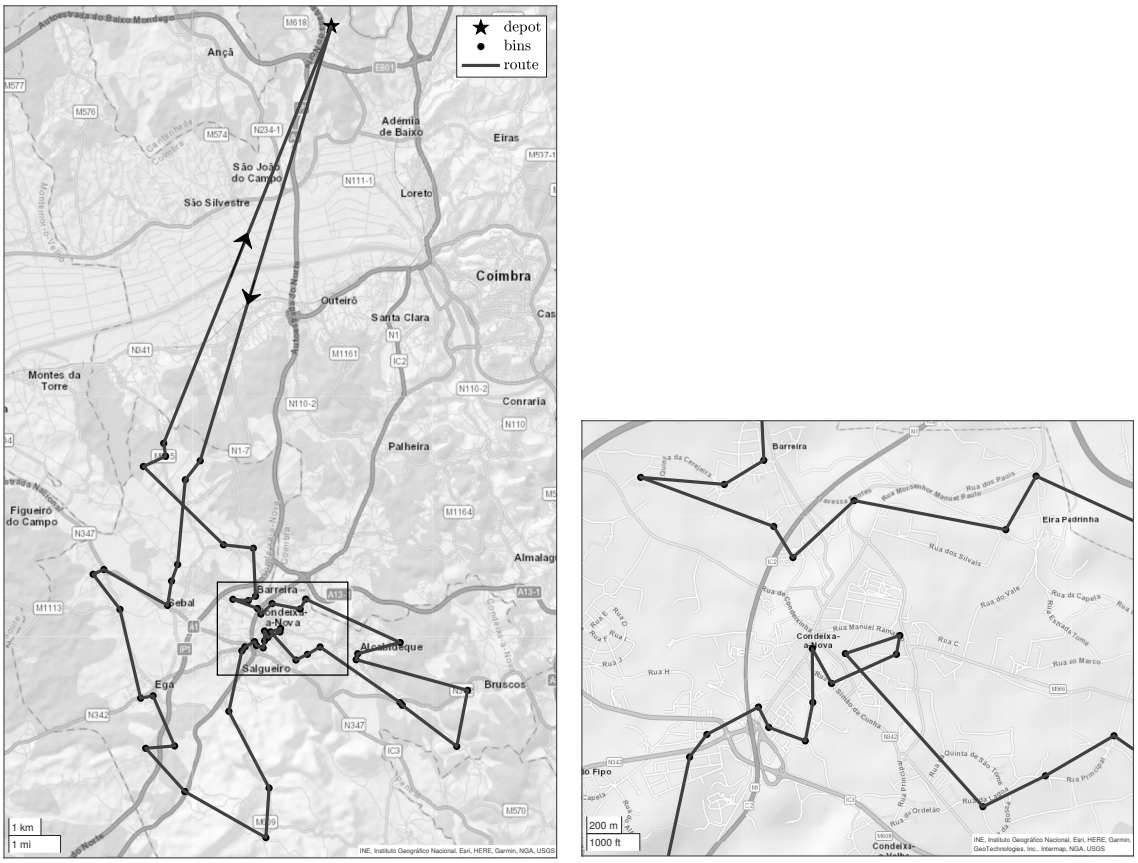


Figure 4.7: Route for the large case instance with 50 bins, obtained by applying the rolling horizon approach with $W = 2$ and runtime limit 2 hours. The route is performed on days 2 and 6 of the planning period, by visiting all the bins. In the picture on the right, a zoom on the area of *Condeixa-a-Nova* is depicted.

4.5.6 Managerial insights

We conclude our analysis by providing some managerial insights on the discussed problem.

First of all, from our previous analysis we note that the waste operator benefits from the application of the rolling horizon approach for a waste collection problem on a large instance because this technique provides accurate results in a short time. To support this

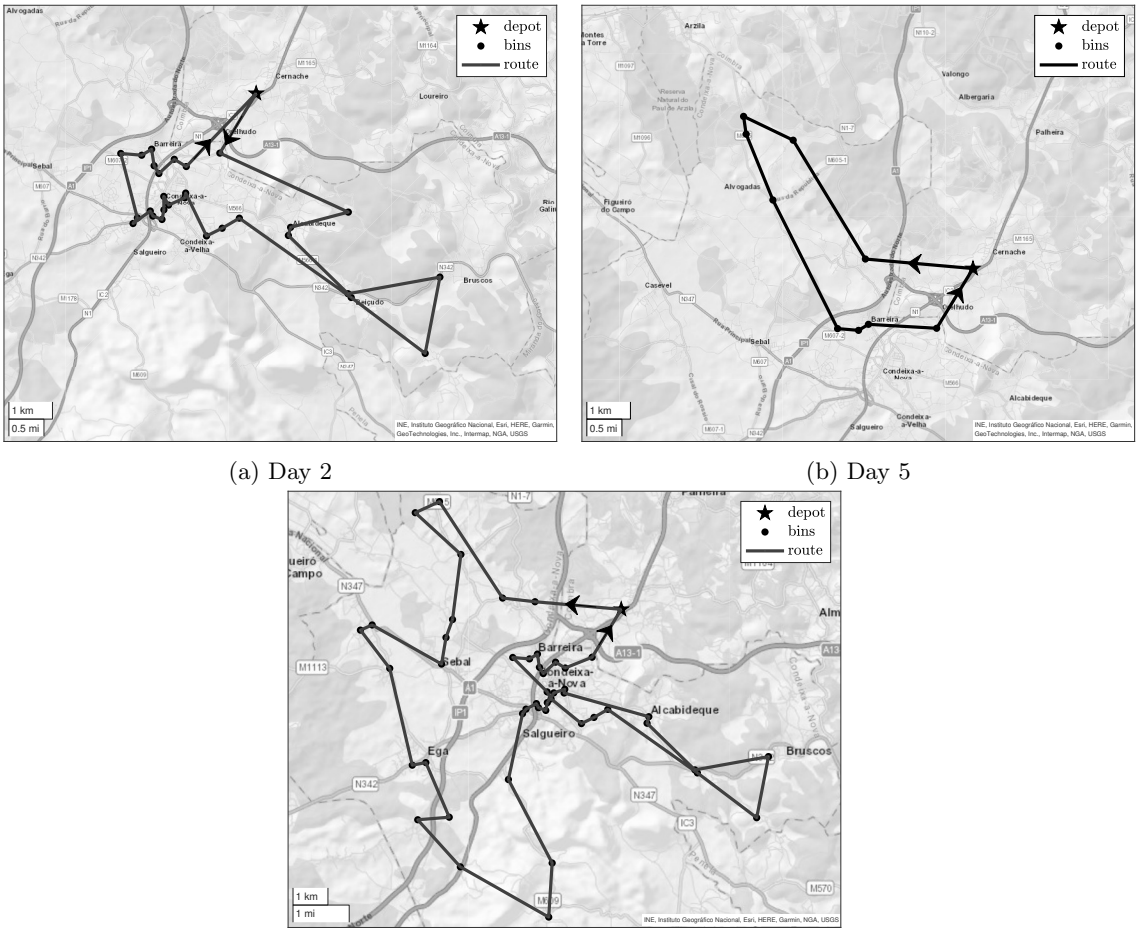
consideration, we report in Table 4.5 some key performance indicators of the problem, when considering a time limit of two hours for the rolling horizon approach.

W	1	2	3	4
Profit (€)	431.70	531.13	527.82	244.17
Total weight of collected waste (kg)	2573.31	2558.26	2564.08	1201.77
Total travelled distance (km)	340.29	236.35	241.41	116.36
Weight/distance (kg/km)	7.56	10.82	10.62	10.33
Profit reduction/CPU time reduction	0.262	0.093	0.102	0.643

Table 4.5: Key performance indicators for the real case instance of 50 bins, when applying the rolling horizon approach with a time limit of 2 hours.

We notice that the highest value of both the profit and of the ratio between the total weight of collected waste and the total travelled distance is reached in the case of $W = 2$. This implies that, from a managerial perspective, choosing a reduced time period of 2 leads to a more efficient and cost-effective planning when compared to the other cases. A similar conclusion can be drawn when considering the ratio between the profit reduction and the CPU time reduction, with respect to the original RP problem. Indeed, even in this case, the best value is attained when $W = 2$. The result confirms that this value of the reduced time period is a good trade-off between accuracy and time savings when solving models $\mathcal{M}$ and $\mathcal{M}_{sym}$ with the rolling horizon approach.

From a managerial point of view, one of the key feature of the model is the selection of the bins to be visited. In Figure 4.7 the same route is performed on days 2 and 6 of the planning period. All the bins are visited twice, but this is due to the very high distance between the depot and the bins. In Figure 4.8 we depict the results of a simulation applying the rolling horizon approach with $W = 2$, where the bins are the same as in the large instance described in Section 4.5.5, but with a closer depot. We notice that the collection is performed on three days (days 2, 5 and 6), with a different selection of bins, respectively 28, 9 and 47. The profit is increased by 15%, compared to the one reported in the third column of Table 4.5, due to the decrease of the total travelled distance (151.62 km vs 236.35 km). The total weight of collected waste remains almost unchanged (2545.94 kg vs 2558.26 kg). The waste manager may benefit from these results because they suggest that the opening of a new depot, closer to the bins, increases significantly the profit, since the routes are more selective and accurate.



(a) Day 2

(b) Day 5

(c) Day 6

Figure 4.8: Routes performed on days 2, 5, 6 of the planning period, with a closer depot to the fifty bins. The results are obtained by applying the rolling horizon approach with $W = 2$ and time limit 2 hours.

4.6 Conclusions

In this chapter, we have studied a Stochastic Inventory Routing Problem applied to waste collection of recyclable materials. For this problem we have proposed a multi-stage mixed integer stochastic programming formulation, with the aim of maximizing the total expected profit. Scenario trees of uncertain waste accumulation have been generated by means of conditional density estimation and dynamic stochastic approximation techniques, and validated in terms of in-sample stability. The impact of stochasticity in waste collection optimization problem has been investigated through standard stochastic measure, showing the benefits of the stochastic methodology when compared to the deterministic framework. We have proposed the rolling horizon as an heuristic methodology to face with the computational complexity of the model and performed a worst-case analysis. Moreover, we have carried out computational experiments showing that considering the

deterministic solution in a stochastic framework may be highly inappropriate. We have tested the performance of the rolling horizon approach on instances of different sizes, based on real data. We have found out that, if the reduced time horizon is properly chosen, the rolling horizon approach provides good quality results with limited computational effort. Finally, we have drawn managerial insights.

Future works will consider a stochastic programming model with real-time information provided by sensors installed in bins and/or in the garbage truck. In addition, the instances to test the model should be enlarged. This would make the formulation very challenging from a computational perspective. Thus, Benders' decomposition and column generation algorithms would be useful techniques to be tested.

Acknowledgments

Research activities of Andrea Spinelli were performed both at University of Bergamo (Italy) and at Instituto Superior Técnico (Portugal), thanks to the international mobility grant of University of Pavia, year 2022.

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, official website: https://ultraoptymal.unibg.it), by the Portuguese Foundation for Science and Technology (FCT) through the research projects 2022.04180.PTDC and UIDB/00097/2020, by European Union Next Generation EU - PRIN2022 (grant number 20223MHHA8), by H2020 TUPLES, and by USAF AFORS (grant number FA8655-21-1-7046).

Chapter 5

Conclusions

In this thesis, we presented models and applications of optimization under uncertainty approaches.

On the one hand, *Robust Optimization* (RO) techniques were applied to tackle problems of binary and multiclass classification through a *Machine Learning* (ML) perspective. Specifically, we extended two approaches of *Support Vector Machine* (SVM) by assuming that input data were plagued by uncertainties. The proposed RO models prevented the worst-case realizations of the uncertain data across prescribed uncertainty sets. This resulted in an increased predictive accuracy of the robust classifiers with respect to the deterministic ones. The performance of the proposed formulations were validated on various ML datasets. Finally, the robust SVM approaches were applied to a vehicles smog rating classification task.

On the other hand, *Stochastic Optimization* (SO) techniques were considered to handle a waste collection problem where the accumulation rate in the waste containers was assumed to be uncertain. To this extent, we formulated a multi-stage SO inventory routing problem with the aim of maximizing the total expected profit coming from collection activities. We faced the computational complexity of the problem by means of the rolling horizon heuristic. The impact of stochasticity on waste generation was analyzed through stochastic measures and the performance of the rolling horizon approach was evaluated on small and large instances inspired by a real case study.

Regarding future developments, several streams of research can originate from this work.

Starting from the robust SVM framework, first of all extend the approaches to handle uncertainties in the labels of input data. This should increase the generalization capability of the models. Additionally, in this thesis we have followed the classic RO approach

of including uncertainty during the training phase (see, for instance, [14]). It should be noteworthy to consider perturbations both in the training and in the testing sets. However, this choice would increase the complexity of the models and novel measures to quantify the accuracy have to be devised, since it is not obvious how to classify a *whole* uncertainty set in one class or another as opposed to the case of single data point. Further techniques should be used to speed up the approaches, especially in the phase of tuning parameters (see, for example, the Bayesian optimization in [138]). Finally, different methodologies should be applied to further robustify the models. For instance, *Distributionally Robust Optimization* (see [125]) with ambiguity sets defined by moments, phi-divergences or Wasserstein distance merits further research too.

As far as it concerns waste collection problems under uncertainty, recent advanced techniques deserve further study. To mention one, the *Distributionally Robust Chance-Constrained Capacitated Vehicle Routing Problem* (see [63]). Within this approach, the customer demand, i.e. the waste accumulation rate in the considered application, is assumed to follow a probability distribution that it is only partially known, and it imposes chance-constraints on the vehicle's capacity. A similar approach can be employed in treating other forms of uncertainty, for instance the travel time. Finally, to increase sustainability a fleet of electric or hybrid vehicles can be considered for the waste collection activity (see, for instance, [24, 108]). Compared to the case study analyzed in this thesis, the computational complexity grows as the stochastic nature of the energy consumption has to be taken into account.

Appendix A

Appendix to Chapter 1

A.1 Supplementary proofs

We first recall a lemma that will be useful to prove Propositions 1-2.

Lemma 1 (**Inequalities in ℓ_p-norm**). Let x be a vector in $\mathbb{R}^n$. If $1 \leqslant p \leqslant q \leqslant \infty$, then:

$$\|x\|_q \leqslant \|x\|_p \leqslant n^{\frac{1}{p}-\frac{1}{q}} \|x\|_q . \tag{A.1}$$

Proof. We consider the two inequalities separately, starting from $\|x\|_q \leqslant \|x\|_p$. First of all, if $x = 0$, then the inequality is obviously true. Otherwise, let $y \in \mathbb{R}^n$ such that $y_i := |x_i| / \|x\|_q$ for $i = 1, \ldots, n$. Therefore, $0 \leqslant y_i \leqslant 1$. Indeed:

$$\|x\|_q^q = \sum_{i=1}^{n} |x_i|^q \geqslant |x_i|^q ,$$

for all $i = 1, \ldots, n$ and thus $|x_i| / \|x\|_q \leqslant 1$. The hypothesis $p \leqslant q$ and the decreasing property of the exponential function with basis lower than one imply that:

$$y_i^p \geqslant y_i^q, \qquad i = 1, \ldots, n.$$

By summing we have:

$$\|y\|_p \geqslant \|y\|_q .$$

Finally, by definition of y we derive that:

$$\frac{\|x\|_p}{\|x\|_q} \geqslant \frac{\|x\|_q}{\|x\|_q} = 1,$$

from which the thesis follows.

On the other hand, to prove the second inequality we recall the Hölder inequality (see, for instance, [128]). Let a and b be in $\mathbb{R}^n$. If r and r' are conjugate exponents, i.e. $\frac{1}{r} + \frac{1}{r'} = 1$, with $1 \leqslant r, r' \leqslant \infty$, then:

$$\|ab\|_1 \leqslant \|a\|_r \cdot \|b\|_{r'},$$

or, equivalently:

$$\sum_{i=1}^{n} |a_i|\,|b_i| \leqslant \left(\sum_{i=1}^{n} |a_i|^r \right)^{\frac{1}{r}} \cdot \left(\sum_{i=1}^{n} |b_i|^{r'} \right)^{\frac{1}{r'}}. \tag{A.2}$$

First of all, we rewrite the ℓ_p-norm of x as:

$$\|x\|_p^p = \sum_{i=1}^{n} |x_i|^p = \sum_{i=1}^{n} |x_i|^p \cdot 1.$$

In the Hölder inequality (A.2), let $a = x$ and $b = e$ and consider as conjugate exponents $r = \frac{q}{p}$ and $r' = \frac{q}{q-p}$. Both r and r' are greater than or equal to 1 because, by hypothesis, $p \leqslant q$. Consequently, we can bound the ℓ_p-norm of x by:

$$\|x\|_p^p \leqslant \left(\sum_{i=1}^{n} \left(|x_i|^p \right)^{\frac{q}{p}} \right)^{\frac{p}{q}} \cdot \left(\sum_{i=1}^{n} 1^{\frac{q}{q-p}} \right)^{1-\frac{p}{q}} = \left(\sum_{i=1}^{n} |x_i|^q \right)^{\frac{p}{q}} n^{1-\frac{p}{q}} = \|x\|_q^p \, n^{1-\frac{p}{q}}.$$

Finally, the thesis follows by taking the p-th root of both sides of the inequality. □

A graphical representation of inequality (A.1) is depicted in Figure A.1.

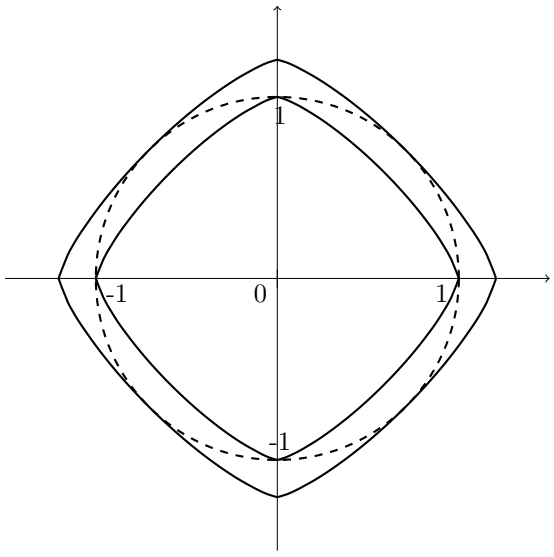


Figure A.1: Graphical representation of Lemma 1 in the case of $p = 1.3$, $q = 2$, $n = 2$. The dashed ℓ_2 unit ball lies between the $\ell_{1.3}$ unit ball and the $\ell_{1.3}$ ball with radius $2^{\frac{1}{1.3}-\frac{1}{2}} \approx 1.205$.

As special cases, Lemma 1 implies that, whenever $1 \leqslant p \leqslant 2$, then:

$$\|x\|_2 \leqslant \|x\|_p . \tag{A.3}$$

Conversely, if $p > 2$, then:

$$\|x\|_2 \leqslant n^{\frac{p-2}{2p}} \|x\|_p . \tag{A.4}$$

Thus, combining these results, we can write:

$$\|x\|_2 \leqslant C \|x\|_p ,$$

with:

$$C = C(n,p) = \begin{cases} 1, & 1 \leqslant p \leqslant 2 \\ n^{\frac{p-2}{2p}}, & p > 2. \end{cases} \tag{A.5}$$

Proof of Proposition 1

Proof. The $\mathcal{H}$-norm of the vector of perturbation $\zeta^{(i)}$ in the feature space can be expanded as:

$$\begin{aligned} \left\|\zeta^{(i)}\right\|_{\mathcal{H}}^2 &= \left\|\phi(x) - \phi(x^{(i)})\right\|_{\mathcal{H}}^2 \\ &= \left\|\phi(x^{(i)} + \sigma^{(i)}) - \phi(x^{(i)})\right\|_{\mathcal{H}}^2 \\ &= \langle \phi(x^{(i)} + \sigma^{(i)}) - \phi(x^{(i)}), \phi(x^{(i)} + \sigma^{(i)}) - \phi(x^{(i)}) \rangle_{\mathcal{H}} \\ &= \langle \phi(x^{(i)} + \sigma^{(i)}), \phi(x^{(i)} + \sigma^{(i)}) \rangle_{\mathcal{H}} - 2\langle \phi(x^{(i)} + \sigma^{(i)}), \phi(x^{(i)} \rangle_{\mathcal{H}} + \langle \phi(x^{(i)}), \phi(x^{(i)}) \rangle_H \\ &= k(x^{(i)} + \sigma^{(i)}, x^{(i)} + \sigma^{(i)}) - 2k(x^{(i)} + \sigma^{(i)}, x^{(i)}) + k(x^{(i)}, x^{(i)}). \end{aligned} \tag{A.6}$$

By definition of the inhomogeneous polynomial kernel of degree d, the last right-hand side of (A.6) becomes:

$$\begin{aligned} \left\|\zeta^{(i)}\right\|_{\mathcal{H}}^2 &= \left(\left\|x^{(i)} + \sigma^{(i)}\right\|_2^2 + c \right)^d - 2\left(\langle x^{(i)} + \sigma^{(i)}, x^{(i)} \rangle + c\right)^d + \left(\left\|x^{(i)}\right\|_2^2 + c \right)^d \\ &= \left(\left\|x^{(i)}\right\|_2^2 + \left\|\sigma^{(i)}\right\|_2^2 + 2\,\langle \sigma^{(i)}, x^{(i)} \rangle + c \right)^d - 2\left(\left\|x^{(i)}\right\|_2^2 + \langle \sigma^{(i)}, x^{(i)} \rangle + c \right)^d + \left(\left\|x^{(i)}\right\|_2^2 + c \right)^d . \end{aligned}$$

By applying the Cauchy-Schwarz inequality in $\mathbb{R}^n$ to the terms containing the dot product, the previous expression simplifies further, leading to:

$$\begin{aligned} \left\|\zeta^{(i)}\right\|_{\mathcal{H}}^2 &\leqslant \left(\left\|x^{(i)}\right\|_2^2 + \left\|\sigma^{(i)}\right\|_2^2 + 2\left\|\sigma^{(i)}\right\|_2 \left\|x^{(i)}\right\|_2 + c \right)^d - 2\left(\left\|x^{(i)}\right\|_2^2 + \left\|\sigma^{(i)}\right\|_2 \left\|x^{(i)}\right\|_2 + c \right)^d + \left(\left\|x^{(i)}\right\|_2^2 + c \right)^d \\ &= \left[\left(\left\|x^{(i)}\right\|_2 + \left\|\sigma^{(i)}\right\|_2 \right)^2 + c \right]^d - 2\left[\left\|x^{(i)}\right\|_2 \left(\left\|x^{(i)}\right\|_2 + \left\|\sigma^{(i)}\right\|_2 \right) + c \right]^d + \left(\left\|x^{(i)}\right\|_2^2 + c \right)^d . \end{aligned}$$

Applying the binomial expansion to three d-th powers implies that:

$$\left\|\zeta^{(i)}\right\|_{\mathcal{H}}^2 \leqslant \sum_{k=0}^{d}\binom{d}{k}c^k\left(\left\|x^{(i)}\right\|_2+\left\|\sigma^{(i)}\right\|_2\right)^{2(d-k)} - 2\sum_{k=0}^{d}\binom{d}{k}c^k\left\|x^{(i)}\right\|_2^{d-k}\left(\left\|x^{(i)}\right\|_2+\left\|\sigma^{(i)}\right\|_2\right)^{d-k} + \\ + \sum_{k=0}^{d}\binom{d}{k}c^k\left\|x^{(i)}\right\|_2^{2(d-k)}.$$

We now split all the three sums by considering separately the cases when $k=0$, $k=d$ and, then, all the intermediate cases. Firstly, let us call a_0 the addendum of the sum corresponding to $k=0$. Therefore:

$$\begin{aligned} a_0 &= \left(\left\|x^{(i)}\right\|_2+\left\|\sigma^{(i)}\right\|_2\right)^{2d} - 2\left\|x^{(i)}\right\|_2^{d}\left(\left\|x^{(i)}\right\|_2+\left\|\sigma^{(i)}\right\|_2\right)^{d} + \left\|x^{(i)}\right\|_2^{2d} \\ &= \left[\left(\left\|x^{(i)}\right\|_2+\left\|\sigma^{(i)}\right\|_2\right)^{d} - \left\|x^{(i)}\right\|_2^{d}\right]^2 \\ &= \left[\sum_{k=0}^{d}\binom{d}{k}\left\|x^{(i)}\right\|_2^{d-k}\left\|\sigma^{(i)}\right\|_2^{k} - \left\|x^{(i)}\right\|_2^{d}\right]^2 \\ &= \left[\sum_{k=1}^{d}\binom{d}{k}\left\|x^{(i)}\right\|_2^{d-k}\left\|\sigma^{(i)}\right\|_2^{k} + \left\|x^{(i)}\right\|_2^{d} - \left\|x^{(i)}\right\|_2^{d}\right]^2 = \left[\sum_{k=1}^{d}\binom{d}{k}\left\|x^{(i)}\right\|_2^{d-k}\left\|\sigma^{(i)}\right\|_2^{k}\right]^2. \end{aligned}$$

We notice that a_0 is the only addendum of the sum that does not contain c. This implies that a_0 is related to the bound $\delta_{d,0}^{(i)}$ for the homogeneous polynomial kernel.
Secondly, if $k=d$, we have no contribution because $c^d - 2c^d + c^d = 0$. Before considering the cases $k=1,\dots,d-1$, we now investigate what happens when the degree d is equal to 1. Here, the index k of the sums goes from 0 to 1, and therefore, as seen before:

$$\left\|\zeta^{(i)}\right\|_{\mathcal{H}}^2 \leqslant \left(\delta_{\text{hom}}^{(i)}\right)^2 = \left(C\eta^{(i)}\right)^2.$$

Hence, when $d=1$, then $\delta_{1,c}^{(i)} = C\eta^{(i)}$. Conversely, when $d>1$, we have all the addenda between $k=1$ and $k=d-1$. Thus, by combining all the three sums together we have:

$$\begin{aligned} \left\|\zeta^{(i)}\right\|_{\mathcal{H}}^2 &\leqslant a_0 + \sum_{k=1}^{d-1}\binom{d}{k}c^k\left[\left(\left\|x^{(i)}\right\|_2+\left\|\sigma^{(i)}\right\|_2\right)^{2(d-k)} - 2\left\|x^{(i)}\right\|_2^{d-k}\left(\left\|x^{(i)}\right\|_2+\left\|\sigma^{(i)}\right\|_2\right)^{d-k} + \left\|x^{(i)}\right\|_2^{2(d-k)}\right] \\ &= a_0 + \sum_{k=1}^{d-1}\binom{d}{k}c^k\left[\left(\left\|x^{(i)}\right\|_2+\left\|\sigma^{(i)}\right\|_2\right)^{d-k} - \left\|x^{(i)}\right\|_2^{d-k}\right]^2. \end{aligned}$$

Again, by applying the binomial expansion to the $(d-k)$-th power of $\left(\left\|x^{(i)}\right\|_2+\left\|\sigma^{(i)}\right\|_2\right)$

and by splitting the sum, we are able to simplify the last term. Hence:

$$\begin{aligned}\left\|\zeta^{(i)}\right\|_{\mathcal{H}}^2 &\leqslant a_0 + \sum_{k=1}^{d-1}\binom{d}{k}c^k\left[\sum_{j=0}^{d-k}\binom{d-k}{j}\left\|x^{(i)}\right\|_2^{d-k-j}\left\|\sigma^{(i)}\right\|_2^{j} - \left\|x^{(i)}\right\|_2^{d-k}\right]^2 \\ &= a_0 + \sum_{k=1}^{d-1}\binom{d}{k}c^k\left[\sum_{j=1}^{d-k}\binom{d-k}{j}\left\|x^{(i)}\right\|_2^{d-k-j}\left\|\sigma^{(i)}\right\|_2^{j}\right]^2.\end{aligned}$$

Therefore, by taking the square root:

$$\left\|\zeta^{(i)}\right\|_{\mathcal{H}} \leqslant \sqrt{a_0 + \sum_{k=1}^{d-1}\binom{d}{k}c^k\left[\sum_{j=1}^{d-k}\binom{d-k}{j}\left\|x^{(i)}\right\|_2^{d-k-j}\left\|\sigma^{(i)}\right\|_2^{j}\right]^2}.$$

According to inequalities (A.3)−(A.4) and to hypothesis $\left\|\sigma^{(i)}\right\|_p \leqslant \eta^{(i)}$, we obtain that:

$$\left\|\sigma^{(i)}\right\|_2 \leqslant \begin{cases} \left\|\sigma^{(i)}\right\|_p \leqslant \eta^{(i)}, & 1 \leqslant p \leqslant 2 \\ n^{\frac{p-2}{2p}}\left\|\sigma^{(i)}\right\|_p \leqslant n^{\frac{p-2}{2p}}\eta^{(i)}, & p > 2. \end{cases}$$

Finally, whenever $1 \leqslant p \leqslant 2$, we have that:

$$a_0 \leqslant \left[\sum_{k=1}^{d}\binom{d}{k}\left\|x^{(i)}\right\|_2^{d-k}\left\|\sigma^{(i)}\right\|_p^{k}\right]^2 \leqslant \left[\sum_{k=1}^{d}\binom{d}{k}\left\|x^{(i)}\right\|_2^{d-k}\left(\eta^{(i)}\right)^k\right]^2 = \left(\delta_{d,0}^{(i)}\right)^2,$$

and the second addendum in the square root can be bounded by:

$$\sum_{k=1}^{d-1}\binom{d}{k}c^k\left[\sum_{j=1}^{d-k}\binom{d-k}{j}\left\|x^{(i)}\right\|_2^{d-k-j}\left(\eta^{(i)}\right)^j\right]^2.$$

On the other hand, if $p > 2$, then:

$$a_0 \leqslant \left[\sum_{k=1}^{d}\binom{d}{k}\left\|x^{(i)}\right\|_2^{d-k}n^{\frac{k(p-2)}{2p}}\left\|\sigma^{(i)}\right\|_p^{k}\right]^2 \leqslant \left[\sum_{k=1}^{d}\binom{d}{k}\left\|x^{(i)}\right\|_2^{d-k}\left(n^{\frac{p-2}{2p}}\eta^{(i)}\right)^k\right]^2 = \left(\delta_{d,0}^{(i)}\right)^2,$$

and similarly the second addendum in the square root is always less than or equal to:

$$\sum_{k=1}^{d-1}\binom{d}{k}c^k\left[\sum_{j=1}^{d-k}\binom{d-k}{j}\left\|x^{(i)}\right\|_2^{d-k-j}\left(n^{\frac{p-2}{2p}}\eta^{(i)}\right)^j\right]^2.$$

□

Proof of Proposition 2

Proof. For all x in $\mathbb{R}^n$, we have that $k(x,x) = 1$ and, thus, equation (A.6) reduces to:

$$\left\|\zeta^{(i)}\right\|_{\mathcal{H}}^2 = 1 - 2\exp\left(-\frac{\left\|x^{(i)} + \sigma^{(i)} - x^{(i)}\right\|_2^2}{2\alpha^2}\right) + 1 = 2 - 2\exp\left(-\frac{\left\|\sigma^{(i)}\right\|_2^2}{2\alpha^2}\right).$$

Therefore:

$$\left\|\zeta^{(i)}\right\|_{\mathcal{H}} = \sqrt{2 - 2\exp\left(-\frac{\left\|\sigma^{(i)}\right\|_2^2}{2\alpha^2}\right)}.$$

The thesis follows by applying inequalities (A.3)−(A.4) and by considering the monotonicity of function $g(x) = -\exp(-x^2)$ when $x > 0$.

□

A.2 Supplementary results

Dataset	Data transformation	Kernel						
		Hom. linear	Hom. quadratic	Hom. cubic	Inhom. linear	Inhom. quadratic	Inhom. cubic	Gaussian RBF
Arrhythmia	–	21.94% ± 0.10	53.43% ± 0.20	–	21.88% ± 0.10	53.31% ± 0.20	–	**20.47% ± 0.07**
	CPU time (s)	0.298	0.297	–	0.309	0.290	–	0.289
	Min-max normalization	–	–	–	–	–	–	–
	CPU time (s)	–	–	–	–	–	–	–
	Standardization	–	–	–	–	–	–	–
	CPU time (s)	–	–	–	–	–	–	–
Parkinson	–	19.18% ± 0.10	21.35% ± 0.06	29.75% ± 0.15	55.99% ± 0.36	24.89% ± 0.12	33.57% ± 0.18	19.66% ± 0.03
	CPU time (s)	3.698	3.661	4.402	3.713	3.762	4.259	3.702
	Min-max normalization	**13.19% ± 0.03**	13.35% ± 0.04	13.95% ± 0.05	13.43% ± 0.05	14.34% ± 0.05	15.23% ± 0.04	13.91% ± 0.04
	CPU time (s)	3.626	3.657	3.636	3.731	3.754	3.656	3.629
	Standardization	14.47% ± 0.04	19.40% ± 0.06	15.95% ± 0.06	16.08% ± 0.06	13.43% ± 0.05	14.11% ± 0.05	16.02% ± 0.05
	CPU time (s)	3.621	3.542	3.600	3.640	3.628	3.673	3.576
Heart Disease	–	23.47% ± 0.09	27.79% ± 0.05	37.36% ± 0.08	35.28% ± 0.13	29.77% ± 0.08	40.37% ± 0.09	33.01% ± 0.05
	CPU time (s)	12.102	12.646	13.050	12.296	12.391	12.755	12.333
	Min-max normalization	18.57% ± 0.04	19.82% ± 0.04	22.75% ± 0.05	18.01% ± 0.04	19.75% ± 0.04	22.24% ± 0.05	31.84% ± 0.06
	CPU time (s)	12.115	12.167	12.050	12.124	12.061	12.162	12.749
	Standardization	19.00% ± 0.04	37.27% ± 0.06	23.56% ± 0.04	**17.48% ± 0.04**	27.48% ± 0.05	24.07% ± 0.04	47.49% ± 0.04
	CPU time (s)	12.162	12.100	12.203	12.253	12.532	12.123	11.597
Dermatology	–	5.41% ± 0.06	2.05% ± 0.01	3.03% ± 0.02	6.14% ± 0.07	**1.64% ± 0.02**	2.90% ± 0.02	7.81% ± 0.08
	CPU time (s)	20.584	20.032	19.969	20.094	20.091	20.033	20.246
	Min-max normalization	3.35% ± 0.03	2.84% ± 0.02	1.85% ± 0.01	3.34% ± 0.03	3.02% ± 0.02	1.95% ± 0.02	30.83% ± 0.01
	CPU time (s)	20.253	20.329	20.173	20.102	20.178	20.132	20.548
	Standardization	3.23% ± 0.03	5.59% ± 0.03	3.29% ± 0.02	3.86% ± 0.03	5.22% ± 0.03	3.35% ± 0.02	30.83% ± 0.01
	CPU time (s)	20.050	20.118	20.290	20.073	20.127	20.230	20.349
Climate Model Crashes	–	**5.01% ± 0.02**	6.04% ± 0.02	8.23% ± 0.02	5.25% ± 0.02	5.87% ± 0.02	7.64% ± 0.02	13.19% ± 0.03
	CPU time (s)	68.069	67.104	65.726	66.070	65.745	66.235	66.383
	Min-max normalization	5.08% ± 0.02	5.52% ± 0.02	7.78% ± 0.02	5.09% ± 0.02	5.87% ± 0.02	7.82% ± 0.03	13.50% ± 0.03
	CPU time (s)	68.296	68.102	69.397	68.228	68.510	69.750	70.330
	Standardization	5.20% ± 0.02	20.15% ± 0.03	11.54% ± 0.02	5.11% ± 0.02	15.73% ± 0.04	11.57% ± 0.03	13.81% ± 0.03
	CPU time (s)	67.022	66.851	66.544	65.792	65.528	65.046	69.635
Breast Cancer Diagnostic	–	10.69% ± 0.15	24.85% ± 0.22	–	16.06% ± 0.21	41.46% ± 0.23	–	8.58% ± 0.02
	CPU time (s)	76.706	77.126	–	76.718	78.570	–	80.493
	Min-max normalization	4.12% ± 0.03	3.15% ± 0.02	3.88% ± 0.02	4.39% ± 0.03	**3.02% ± 0.02**	5.80% ± 0.05	12.87% ± 0.05
	CPU time (s)	76.340	76.476	76.106	76.350	77.786	78.282	77.690
	Standardization	3.65% ± 0.02	17.72% ± 0.03	5.40% ± 0.02	3.88% ± 0.02	6.88% ± 0.02	4.92% ± 0.02	36.62% ± 0.01
	CPU time (s)	78.100	78.279	77.534	76.813	76.041	76.715	77.248
Breast Cancer	–	3.21% ± 0.01	7.02% ± 0.02	8.36% ± 0.07	3.39% ± 0.02	6.84% ± 0.02	11.58% ± 0.16	3.20% ± 0.01
	CPU time (s)	133.833	132.231	133.750	134.134	134.697	135.338	133.728
	Min-max normalization	4.06% ± 0.04	3.29% ± 0.01	4.20% ± 0.02	4.12% ± 0.02	4.43% ± 0.03	4.82% ± 0.02	3.20% ± 0.01
	CPU time (s)	135.390	135.382	135.109	137.616	136.871	134.736	136.484
	Standardization	**3.17% ± 0.01**	6.80% ± 0.03	5.88% ± 0.02	3.19% ± 0.01	6.21% ± 0.02	5.61% ± 0.02	3.88% ± 0.02
	CPU time (s)	135.765	135.553	136.774	135.623	134.514	135.597	137.221
Blood Transfusion	–	24.09% ± 0.01	26.57 ± 0.15	–	24.00% ± 0.01	27.35% ± 0.15	–	23.73% ± 0.01
	CPU time (s)	170.744	176.855	–	174.407	176.929	–	174.808
	Min-max normalization	23.82% ± 0.00	23.85% ± 0.01	23.73% ± 0.02	23.84% ± 0.00	23.92% ± 0.01	23.25% ± 0.01	23.53% ± 0.02
	CPU time (s)	176.273	178.011	178.221	177.326	179.052	175.440	176.455
	Standardization	23.85% ± 0.01	23.37% ± 0.01	22.00% ± 0.02	24.01% ± 0.01	20.97% ± 0.02	**20.72% ± 0.02**	21.09% ± 0.02
	CPU time (s)	178.088	178.107	177.398	177.692	179.141	178.136	176.627
Mammographic Mass	–	20.92% ± 0.07	15.85% ± 0.02	17.51% ± 0.05	17.12% ± 0.02	16.20% ± 0.02	28.47% ± 0.15	18.48% ± 0.02
	CPU time (s)	240.550	239.300	241.644	241.607	242.298	242.582	239.141
	Min-max normalization	26.22% ± 0.12	16.60% ± 0.02	16.09% ± 0.02	26.77% ± 0.13	16.04% ± 0.02	16.06% ± 0.02	17.25% ± 0.02
	CPU time (s)	241.645	240.950	239.648	241.134	241.525	239.143	241.536
	Standardization	19.49% ± 0.06	31.14% ± 0.05	18.82% ± 0.03	19.73% ± 0.08	**15.71% ± 0.02**	18.63% ± 0.02	18.29% ± 0.02
	CPU time (s)	239.300	236.677	239.877	238.003	241.205	242.163	240.254

Table A.1: Detailed results of average out-of-sample testing errors and standard deviations over 96 runs of the deterministic model. Holdout: 75% training set-25% testing set.

Dataset	Data transformation	Kernel						
		Hom. linear	Hom. quadratic	Hom. cubic	Inhom. linear	Inhom. quadratic	Inhom. cubic	Gaussian RBF
Arrhythmia	–	23.90% ± 0.06	54.23% ± 0.02	–	24.08% ± 0.05	51.72% ± 0.21	–	**23.59% ± 0.04**
	CPU time (s)	0.194	0.180	–	0.191	0.216	–	0.181
	Min-max normalization	–	–	–	–	–	–	–
	CPU time (s)	–	–	–	–	–	–	–
	Standardization	–	–	–	–	–	–	–
	CPU time (s)	–	–	–	–	–	–	–
Parkinson	–	19.58% ± 0.11	25.10% ± 0.06	40.05% ± 0.22	46.88% ± 0.27	23.70% ± 0.07	32.07% ± 0.17	19.97% ± 0.05
	CPU time (s)	1.283	1.230	1.290	1.211	1.330	1.382	1.264
	Min-max normalization	15.54% ± 0.06	**15.10% ± 0.04**	16.02% ± 0.04	15.43% ± 0.04	16.26% ± 0.04	16.13% ± 0.04	18.38% ± 0.04
	CPU time (s)	1.195	1.203	1.206	1.214	1.202	1.207	1.184
	Standardization	15.79% ± 0.04	22.71% ± 0.06	17.74% ± 0.05	17.62% ± 0.04	17.98% ± 0.04	17.71% ± 0.05	19.33% ± 0.05
	CPU time (s)	1.203	1.257	1.207	1.201	1.193	1.204	1.183
Heart Disease	–	23.00% ± 0.08	28.55% ± 0.04	37.81% ± 0.07	31.07% ± 0.13	30.36% ± 0.07	42.13% ± 0.09	34.45% ± 0.04
	CPU time (s)	4.132	4.199	4.732	4.193	4.223	4.821	4.045
	Min-max normalization	20.00% ± 0.03	21.40% ± 0.03	22.87% ± 0.04	20.11% ± 0.06	20.82% ± 0.03	22.64% ± 0.03	32.07% ± 0.06
	CPU time (s)	4.078	4.076	4.097	4.063	4.098	4.168	4.057
	Standardization	19.05% ± 0.04	37.16% ± 0.04	23.97% ± 0.03	**18.92% ± 0.03**	26.82% ± 0.04	23.99% ± 0.04	46.01% ± 0.04
	CPU time (s)	4.182	4.131	4.147	4.138	4.075	4.112	4.095
Dermatology	–	8.90% ± 0.08	2.01% ± 0.01	3.17% ± 0.02	12.08% ± 0.11	**1.96% ± 0.01**	3.34% ± 0.02	8.82% ± 0.08
	CPU time (s)	5.999	6.015	6.115	6.089	6.075	6.072	6.115
	Min-max normalization	4.35% ± 0.05	3.45% ± 0.02	2.55% ± 0.02	4.82% ± 0.06	3.93% ± 0.02	2.42% ± 0.01	30.97% ± 0.00
	CPU time (s)	6.019	6.159	6.049	6.077	6.090	6.021	6.137
	Standardization	4.16% ± 0.03	7.16% ± 0.02	4.19% ± 0.02	4.78% ± 0.03	5.74% ± 0.02	4.14% ± 0.02	30.97% ± 0.00
	CPU time (s)	6.092	6.127	6.258	6.137	6.146	6.101	6.101
Climate Model Crashes	–	5.56% ± 0.01	7.01% ± 0.02	8.16% ± 0.02	**5.35% ± 0.01**	7.44% ± 0.02	8.23% ± 0.02	13.42% ± 0.02
	CPU time (s)	20.032	20.018	20.056	20.035	20.051	20.145	19.856
	Min-max normalization	5.57% ± 0.01	7.21% ± 0.02	8.24% ± 0.02	5.42% ± 0.01	7.17% ± 0.02	8.29% ± 0.02	13.57% ± 0.02
	CPU time (s)	20.742	21.174	20.553	20.941	20.628	20.147	20.740
	Standardization	5.82% ± 0.01	20.01% ± 0.03	11.77% ± 0.02	6.40% ± 0.02	15.21% ± 0.03	11.55% ± 0.03	13.14% ± 0.02
	CPU time (s)	20.059	19.566	19.518	19.702	19.748	19.780	20.310
Breast Cancer Diagnostic	–	13.53% ± 0.18	27.06% ± 0.24	–	16.94% ± 0.23	34.81% ± 0.23	–	9.26% ± 0.02
	CPU time (s)	24.553	24.289	–	24.410	24.671	–	24.525
	Min-max normalization	6.21% ± 0.06	3.87% ± 0.03	4.43% ± 0.01	5.99% ± 0.05	**3.69% ± 0.02**	5.19% ± 0.03	20.68% ± 0.08
	CPU time (s)	24.449	24.405	24.600	24.671	24.237	24.634	23.636
	Standardization	4.17% ± 0.02	19.01% ± 0.02	5.67% ± 0.02	4.45% ± 0.03	7.43% ± 0.02	5.22% ± 0.01	37.21% ± 0.00
	CPU time (s)	24.472	24.526	24.855	24.664	22.988	23.080	23.791
Breast Cancer	–	4.54% ± 0.04	6.47% ± 0.02	12.57% ± 0.11	3.61% ± 0.02	6.72% ± 0.01	26.05% ± 0.25	3.84% ± 0.01
	CPU time (s)	39.279	39.238	40.161	38.794	40.341	39.618	39.730
	Min-max normalization	5.71% ± 0.06	**3.31% ± 0.01**	4.52% ± 0.01	10.05% ± 0.11	4.37% ± 0.02	4.99% ± 0.01	3.62% ± 0.01
	CPU time (s)	38.718	40.067	39.394	39.775	39.780	39.395	42.094
	Standardization	3.37% ± 0.01	7.69% ± 0.02	6.13% ± 0.01	3.75% ± 0.01	6.43% ± 0.01	5.97% ± 0.01	5.08% ± 0.02
	CPU time (s)	38.914	39.175	38.866	39.353	39.007	38.892	40.610
Blood Transfusion	–	23.81% ± 0.01	22.99% ± 0.01	–	23.68% ± 0.00	31.60% ± 0.19	–	25.40% ± 0.09
	CPU time (s)	49.452	50.652	–	51.609	54.469	–	51.579
	Min-max normalization	23.85% ± 0.00	23.84% ± 0.01	23.69% ± 0.01	23.81% ± 0.00	23.77% ± 0.01	23.59% ± 0.01	23.38% ± 0.01
	CPU time (s)	51.422	51.582	52.451	52.365	51.648	52.098	51.996
	Standardization	23.77% ± 0.01	23.77% ± 0, 01	22.52% ± 0.01	23.69% ± 0.00	21.98% ± 0.01	**21.86% ± 0.03**	22.07% ± 0.01
	CPU time (s)	51.396	52.654	53.843	52.609	52.676	52.658	52.006
Mammographic Mass	–	24.02% ± 0.10	17.28% ± 0.05	35.66% ± 0.16	25.55% ± 0.11	40.95% ± 0.14	46.42% ± 0.09	19.84 ± 0.02
	CPU time (s)	70.958	70.916	71.854	71.198	72.179	72.495	70.880
	Min-max normalization	21.74% ± 0.09	17.72% ± 0.02	**16.49% ± 0.02**	23.36% ± 0.11	19.71% ± 0.08	18.62% ± 0.08	17.94% ± 0.01
	CPU time (s)	71.468	71.291	71.426	71.415	71.274	73.143	71.589
	Standardization	20.08% ± 0.06	32.87% ± 0.07	19.86% ± 0.02	20.25% ± 0.06	16.56% ± 0.01	19.54% ± 0.02	18.84% ± 0.02
	CPU time (s)	70.693	72.523	72.951	71.156	71.861	71.803	71.269

Table A.2: Detailed results of average out-of-sample testing errors and standard deviations over 96 runs of the deterministic model. Holdout: 50% training set-50% testing set.

Dataset	Data transformation	Kernel						
		Hom. linear	Hom. quadratic	Hom. cubic	Inhom. linear	Inhom. quadratic	Inhom. cubic	Gaussian RBF
Arrhythmia	–	28.66% ± 0.07	53.45% ± 0.20	–	**27.31% ± 0.07**	53.84% ± 0.20	–	28.29% ± 0.02
	CPU time (s)	0.142	0.165	–	0.144	0.148	–	0.153
	Min-max normalization	–	–	–	–	–	–	–
	CPU time (s)	–	–	–	–	–	–	–
	Standardization	–	–	–	–	–	–	–
	CPU time (s)	–	–	–	–	–	–	–
Parkinson	–	26.58% ± 0.17	25.55% ± 0.05	45.19% ± 0.24	37.54% ± 0.25	25.85% ± 0.08	45.44% ± 0.25	21.48% ± 0.03
	CPU time (s)	0.293	0.303	0.530	0.301	0.285	0.596	0.295
	Min-max normalization	18.91% ± 0.05	**18.49% ± 0.04**	20.92% ± 0.05	19.98% ± 0.06	19.66% ± 0.07	21.08% ± 0.06	21.88% ± 0.04
	CPU time (s)	0.311	0.286	0.281	0.291	0.294	0.301	0.291
	Standardization	20.03% ± 0.04	30.18% ± 0.05	23.10% ± 0.06	19.73% ± 0.04	23.47% ± 0.06	22.85% ± 0.05	23.78% ± 0.05
	CPU time (s)	0.286	0.292	0.278	0.277	0.289	0.301	0.289
Heart Disease	–	25.58% ± 0.08	28.73% ± 0.04	42.38% ± 0.08	28.03% ± 0.10	29.50% ± 0.06	46.02% ± 0.08	36.61% ± 0.04
	CPU time (s)	0.656	0.648	1.496	0.703	0.682	1.863	0.613
	Min-max normalization	22.00% ± 0.05	23.18% ± 0.03	22.76% ± 0.03	**21.85% ± 0.05**	23.33% ± 0.04	23.07% ± 0.03	38.86% ± 0.07
	CPU time (s)	0.638	0.640	0.663	0.625	0.628	0.640	0.634
	Standardization	22.48% ± 0.04	39.13% ± 0.04	25.48% ± 0.04	22.94% ± 0.06	28.38% ± 0.03	25.67% ± 0.04	45.74% ± 0.03
	CPU time (s)	0.640	0.653	0.701	0.637	0.629	0.623	0.629
Dermatology	–	13.17% ± 0.11	3.13% ± 0.02	4.19% ± 0.03	13.88% ± 0.11	3.14% ± 0.02	4.18% ± 0.04	10.01% ± 0.04
	CPU time (s)	0.971	0.981	0.963	0.956	0.951	0.965	0.974
	Min-max normalization	10.32% ± 0.11	5.81% ± 0.05	3.65% ± 0.02	9.03% ± 0.10	5.66% ± 0.05	**3.01% ± 0.02**	30.97% ± 0.00
	CPU time (s)	1.006	1.100	0.958	0.960	0.960	0.977	0.957
	Standardization	6.74% ± 0.04	10.54% ± 0.03	7.35% ± 0.03	7.97% ± 0.05	9.52% ± 0.03	7.05% ± 0.03	30.97% ± 0.00
	CPU time (s)	0.968	0.962	0.957	0.953	0.972	0.967	0.955
Climate Model Crashes	–	7.28% ± 0.01	10.51% ± 0.02	10.47% ± 0.02	**7.15% ± 0.01**	10.59% ± 0.03	11.18% ± 0.03	14.34% ± 0.02
	CPU time (s)	2.804	2.715	2.686	2.671	2.653	2.662	2.811
	Min-max normalization	7.20% ± 0.01	10.54% ± 0.02	10.80% ± 0.03	7.27% ± 0.01	10.77% ± 0.03	10.66% ± 0.03	14.14% ± 0.02
	CPU time (s)	2.847	2.827	2.840	2.848	2.865	2.856	2.839
	Standardization	10.04% ± 0.03	19.74% ± 0.05	12.58% ± 0.04	9.99% ± 0.03	14.36% ± 0.04	12.53% ± 0.03	13.51% ± 0.02
	CPU time (s)	2.874	2.823	2.759	2.844	2.853	2.814	2.827
Breast Cancer Diagnostic	–	17.25% ± 0.19	39.75% ± 0.24	–	20.31% ± 0.23	28.03% ± 0.23	–	11.21% ± 0.04
	CPU time (s)	3.498	3.515	–	3.256	3.419	–	3.376
	Min-max normalization	8.62% ± 0.07	6.29% ± 0.05	5.98% ± 0.02	8.89% ± 0.08	5.87% ± 0.04	6.43% ± 0.03	33.15% ± 0.06
	CPU time (s)	3.439	3.249	3.285	3.289	3.250	3.255	3.395
	Standardization	5.11% ± 0.02	22.85% ± 0.03	6.37% ± 0.02	**5.02% ± 0.02**	10.49% ± 0.02	6.17% ± 0.02	37.32% ± 0.00
	CPU time (s)	3.287	3.309	3.317	3.278	3.381	3.302	3.278
Breast Cancer	–	7.05% ± 0.06	6.58% ± 0.02	21.30% ± 0.14	6.56% ± 0.06	6.73% ± 0.02	21.95% ± 0.20	5.00% ± 0.02
	CPU time (s)	5.392	5.318	5.406	5.440	5.490	5.506	5.511
	Min-max normalization	8.62% ± 0.09	**4.47% ± 0.02**	5.92% ± 0.02	11.70% ± 0.11	5.01% ± 0.04	5.83% ± 0.02	5.00% ± 0.02
	CPU time (s)	5.507	5.536	5.574	5.420	5.463	5.522	5.505
	Standardization	5.12% ± 0.05	9.45% ± 0.02	6.36% ± 0.02	4.64% ± 0.03	7.33% ± 0.02	6.18% ± 0.02	6.01% ± 0.02
	CPU time (s)	5.526	5.423	5.399	5.413	5.418	5.422	5.565
Blood Transfusion	–	23.69% ± 0.00	23.55% ± 0.01	–	23.69% ± 0.01	42.69% ± 0.25	–	23.96% ± 0.01
	CPU time (s)	7.214	7.618	–	7.342	7.622	–	6.838
	Min-max normalization	23.85% ± 0.01	23.75% ± 0.01	23.69% ± 0.01	23.77% ± 0.00	23.68% ± 0.00	23.68% ± 0.01	23.53% ± 0.01
	CPU time (s)	7.319	7.430	7.141	7.398	7.122	7.144	6.665
	Standardization	23.75% ± 0.01	23.63% ± 0.00	**23.32% ± 0.01**	23.72% ± 0.00	23.37% ± 0.05	26.03% ± 0.09	23.35% ± 0.01
	CPU time (s)	7.357	7.145	7.222	7.183	7.084	7.344	6.737
Mammographic Mass	–	28.35% ± 0.13	18.83% ± 0.05	37.40% ± 0.14	30.20% ± 0.14	36.33% ± 0.15	39.55% ± 0.15	22.21% ± 0.03
	CPU time (s)	9.024	9.013	9.166	9.206	9.152	9.166	8.964
	Min-max normalization	22.21% ± 0.09	19.02% ± 0.04	**17.68% ± 0.02**	24.31% ± 0.11	19.56% ± 0.05	20.38% ± 0.08	19.39% ± 0.02
	CPU time (s)	9.068	9.086	9.116	9.019	9.009	9.021	8.953
	Standardization	19.48% ± 0.04	32.89% ± 0.09	21.98% ± 0.04	21.02% ± 0.06	19.21% ± 0.04	24.04% ± 0.07	20.13% ± 0.02
	CPU time (s)	9.114	9.125	9.184	9.101	9.152	9.250	9.114

Table A.3: Detailed results of average out-of-sample testing errors and standard deviations over 96 runs of the deterministic model. Holdout: 25% training set-75% testing set.

Dataset	Data transformation	Kernel	ρ	Robust		
				$p = 1$	$p = 2$	$p = \infty$
Arrhythmia	–	Gaussian RBF	10^{-7}	$19.18\% \pm 0.07$	$19.42\% \pm 0.07$	$19.67\% \pm 0.07$
			10^{-6}	$19.30\% \pm 0.06$	$19.61\% \pm 0.07$	$20.40\% \pm 0.08$
			10^{-5}	$20.47\% \pm 0.07$	$19.91\% \pm 0.07$	$19.73\% \pm 0.06$
			10^{-4}	$20.10\% \pm 0.07$	$19.55\% \pm 0.07$	$19.61\% \pm 0.07$
			10^{-3}	$\mathbf{19.12\% \pm 0.08}$	$19.30\% \pm 0.07$	$23.28\% \pm 0.06$
			10^{-2}	$19.30\% \pm 0.07$	$20.83\% \pm 0.08$	$29.41\% \pm 0.00$
			10^{-1}	$29.41\% \pm 0.00$	$29.41\% \pm 0.00$	$29.41\% \pm 0.00$
			CPU time (s)	0.290	0.288	0.295
Parkinson	Min-max normalization	Hom. linear	10^{-7}	$12.98\% \pm 0.03$	$12.87\% \pm 0.03$	$13.02\% \pm 0.04$
			10^{-6}	$13.50\% \pm 0.04$	$13.02\% \pm 0.04$	$12.80\% \pm 0.04$
			10^{-5}	$13.04\% \pm 0.04$	$13.28\% \pm 0.04$	$12.61\% \pm 0.04$
			10^{-4}	$13.93\% \pm 0.03$	$\mathbf{12.37\% \pm 0.03}$	$12.72\% \pm 0.03$
			10^{-3}	$13.54\% \pm 0.03$	$13.32\% \pm 0.04$	$13.48\% \pm 0.04$
			10^{-2}	$12.98\% \pm 0.03$	$13.17\% \pm 0.04$	$15.15\% \pm 0.04$
			10^{-1}	$15.28\% \pm 0.03$	$15.58\% \pm 0.03$	$25.00\% \pm 0.00$
			CPU time (s)	3.421	3.454	3.418
Heart disease	Standardization	Inhom. linear	10^{-7}	$16.84\% \pm 0.04$	$17.53\% \pm 0.04$	$16.84\% \pm 0.04$
			10^{-6}	$17.53\% \pm 0.04$	$17.72\% \pm 0.04$	$17.53\% \pm 0.04$
			10^{-5}	$17.37\% \pm 0.04$	$18.26\% \pm 0.03$	$17.38\% \pm 0.04$
			10^{-4}	$17.75\% \pm 0.04$	$18.27\% \pm 0.04$	$17.64\% \pm 0.04$
			10^{-3}	$17.13\% \pm 0.04$	$18.43\% \pm 0.04$	$17.12\% \pm 0.04$
			10^{-2}	$17.10\% \pm 0.04$	$17.92\% \pm 0.04$	$\mathbf{16.36\% \pm 0.04}$
			10^{-1}	$16.98\% \pm 0.04$	$17.53\% \pm 0.03$	$16.37\% \pm 0.04$
			CPU time (s)	11.602	11.477	11.417
Dermatology	–	Inhom. quadratic	10^{-7}	$1.65\% \pm 0.01$	$1.71\% \pm 0.01$	$1.72\% \pm 0.01$
			10^{-6}	$1.78\% \pm 0.01$	$1.80\% \pm 0.02$	$1.79\% \pm 0.01$
			10^{-5}	$1.73\% \pm 0.02$	$1.57\% \pm 0.01$	$1.76\% \pm 0.01$
			10^{-4}	$11.06\% \pm 0.04$	$0.39\% \pm 0.04$	$1.28\% \pm 0.01$
			10^{-3}	$30.93\% \pm 0.01$	$30.93\% \pm 0.01$	$\mathbf{0.55\% \pm 0.01}$
			10^{-2}	$30.91\% \pm 0.01$	$30.86\% \pm 0.01$	$30.89\% \pm 0.01$
			10^{-1}	$38.06\% \pm 0.21$	$32.33\% \pm 0.10$	$30.92\% \pm 0.01$
			CPU time (s)	20.055	20.420	20.147
Climate Model Crashes	–	Hom. linear	10^{-7}	$4.74\% \pm 0.02$	$4.51\% \pm 0.01$	$4.60\% \pm 0.02$
			10^{-6}	$4.70\% \pm 0.02$	$4.88\% \pm 0.01$	$4.93\% \pm 0.02$
			10^{-5}	$4.52\% \pm 0.02$	$4.56\% \pm 0.01$	$4.71\% \pm 0.02$
			10^{-4}	$4.86\% \pm 0.02$	$4.78\% \pm 0.02$	$4.85\% \pm 0.01$
			10^{-3}	$4.47\% \pm 0.02$	$4.71\% \pm 0.01$	$\mathbf{4.34\% \pm 0.01}$
			10^{-2}	$4.67\% \pm 0.01$	$4.50\% \pm 0.01$	$4.81\% \pm 0.02$
			10^{-1}	$8.46\% \pm 0.00$	$8.52\% \pm 0.00$	$8.47\% \pm 0.00$
			CPU time (s)	66.762	67.169	67.381

Table A.4: Average out-of-sample testing errors and standard deviations over 96 runs of the robust model. Holdout: 75% training set-25% testing set.

Dataset	Data transformation	Kernel	ρ	Robust		
				$p = 1$	$p = 2$	$p = \infty$
Breast Cancer Diagnostic	Min-max normalization	Inhom. quadratic	10^{-7}	$2.80\% \pm 0.01$	$2.65\% \pm 0.01$	$2.80\% \pm 0.01$
			10^{-6}	$2.96\% \pm 0.02$	$2.70\% \pm 0.01$	$2.96\% \pm 0.02$
			10^{-5}	$2.63\% \pm 0.01$	$2.99\% \pm 0.01$	$2.66\% \pm 0.01$
			10^{-4}	$2.88\% \pm 0.01$	$2.88\% \pm 0.01$	$\mathbf{2.56\% \pm 0.01}$
			10^{-3}	$2.91\% \pm 0.01$	$3.19\% \pm 0.01$	$9.76\% \pm 0.03$
			10^{-2}	$37.32\% \pm 0.00$	$37.32\% \pm 0.00$	$37.32\% \pm 0.00$
			10^{-1}	$37.32\% \pm 0.00$	$37.32\% \pm 0.00$	$37.32\% \pm 0.00$
			CPU time (s)	77.968	78.267	77.543
Breast Cancer	Standardization	Hom. linear	10^{-7}	$3.20\% \pm 0.01$	$3.24\% \pm 0.01$	$3.17\% \pm 0.01$
			10^{-6}	$3.16\% \pm 0.01$	$3.26\% \pm 0.01$	$3.17\% \pm 0.01$
			10^{-5}	$\mathbf{2.97\% \pm 0.01}$	$3.32\% \pm 0.01$	$3.14\% \pm 0.01$
			10^{-4}	$3.23\% \pm 0.01$	$3.50\% \pm 0.01$	$3.20\% \pm 0.01$
			10^{-3}	$3.11\% \pm 0.01$	$3.07\% \pm 0.01$	$3.21\% \pm 0.01$
			10^{-2}	$3.33\% \pm 0.01$	$3.19\% \pm 0.01$	$3.08\% \pm 0.01$
			10^{-1}	$3.07\% \pm 0.01$	$3.32\% \pm 0.01$	$3.06\% \pm 0.01$
			CPU time (s)	135.651	137.039	136.286
Blood Transfusion	Standardization	Inhom. cubic	10^{-7}	$20.60\% \pm 0.02$	$\mathbf{20.55\% \pm 0.02}$	$20.64\% \pm 0.02$
			10^{-6}	$20.72\% \pm 0.02$	$20.80\% \pm 0.02$	$20.77\% \pm 0.02$
			10^{-5}	$21.26\% \pm 0.02$	$20.97\% \pm 0.02$	$22.49\% \pm 0.02$
			10^{-4}	$23.88\% \pm 0.00$	$23.85\% \pm 0.00$	$23.79\% \pm 0.00$
			10^{-3}	$23.80\% \pm 0.00$	$24.57\% \pm 0.08$	$26.18\% \pm 0.13$
			10^{-2}	$26.19\% \pm 0.13$	$30.94\% \pm 0.22$	$38.88\% \pm 0.31$
			10^{-1}	$61.12\% \pm 0.38$	$57.13\% \pm 0.38$	$56.37\% \pm 0.38$
			CPU time (s)	178.751	179.682	180.083
Mammographic Mass	Standardization	Inhom. quadratic	10^{-7}	$15.71\% \pm 0.02$	$\mathbf{15.42\% \pm 0.02}$	$15.54\% \pm 0.02$
			10^{-6}	$15.57\% \pm 0.02$	$15.46\% \pm 0.03$	$15.74\% \pm 0.03$
			10^{-5}	$15.49\% \pm 0.02$	$16.16\% \pm 0.03$	$15.66\% \pm 0.02$
			10^{-4}	$15.91\% \pm 0.02$	$16.16\% \pm 0.03$	$18.81\% \pm 0.02$
			10^{-3}	$48.54\% \pm 0.00$	$48.56\% \pm 0.00$	$48.56\% \pm 0.00$
			10^{-2}	$48.57\% \pm 0.00$	$48.53\% \pm 0.00$	$48.53\% \pm 0.00$
			10^{-1}	$48.56\% \pm 0.00$	$48.54\% \pm 0.00$	$48.54\% \pm 0.00$
			CPU time (s)	241.810	242.614	241.929

Table A.5: Average out-of-sample testing errors and standard deviations over 96 runs of the robust model. Holdout: 75% training set-25% testing set (continued).

Dataset	Data transformation	Kernel	ρ	Robust		
				$p = 1$	$p = 2$	$p = \infty$
Arrhythmia	–	Gaussian RBF	10^{-7}	$24.66\% \pm 0.04$	$23.99\% \pm 0.05$	$\mathbf{23.44\% \pm 0.04}$
			10^{-6}	$24.11\% \pm 0.05$	$24.63\% \pm 0.04$	$23.77\% \pm 0.05$
			10^{-5}	$23.81\% \pm 0.05$	$24.08\% \pm 0.04$	$24.11\% \pm 0.05$
			10^{-4}	$23.77\% \pm 0.05$	$23.74\% \pm 0.05$	$24.33\% \pm 0.05$
			10^{-3}	$24.51\% \pm 0.04$	$24.60\% \pm 0.05$	$26.10\% \pm 0.04$
			10^{-2}	$24.36\% \pm 0.04$	$23.77\% \pm 0.05$	$29.41\% \pm 0.00$
			10^{-1}	$29.41\% \pm 0.00$	$29.41\% \pm 0.00$	$29.41\% \pm 0.00$
			CPU time (s)	0.191	0.195	0.196
Parkinson	Min-max normalization	Hom. quadratic	10^{-7}	$13.92\% \pm 0.03$	$14.89\% \pm 0.04$	$14.51\% \pm 0.03$
			10^{-6}	$14.85\% \pm 0.03$	$14.45\% \pm 0.03$	$14.25\% \pm 0.03$
			10^{-5}	$14.52\% \pm 0.03$	$14.45\% \pm 0.03$	$14.45\% \pm 0.03$
			10^{-4}	$14.28\% \pm 0.04$	$14.28\% \pm 0.03$	$14.33\% \pm 0.03$
			10^{-3}	$14.84\% \pm 0.03$	$14.41\% \pm 0.03$	$13.85\% \pm 0.03$
			10^{-2}	$\mathbf{13.84\% \pm 0.03}$	$13.86\% \pm 0.03$	$15.01\% \pm 0.03$
			10^{-1}	$15.38\% \pm 0.02$	$15.70\% \pm 0.02$	$24.74\% \pm 0.00$
			CPU time (s)	1.195	1.217	1.224
Heart disease	Standardization	Inhom. linear	10^{-7}	$18.38\% \pm 0.03$	$18.21\% \pm 0.02$	$18.21\% \pm 0.02$
			10^{-6}	$18.18\% \pm 0.03$	$18.53\% \pm 0.03$	$18.53\% \pm 0.03$
			10^{-5}	$17.98\% \pm 0.03$	$18.17\% \pm 0.03$	$18.17\% \pm 0.03$
			10^{-4}	$18.29\% \pm 0.03$	$18.82\% \pm 0.03$	$18.78\% \pm 0.03$
			10^{-3}	$18.88\% \pm 0.03$	$18.19\% \pm 0.03$	$18.19\% \pm 0.03$
			10^{-2}	$18.92\% \pm 0.03$	$18.22\% \pm 0.03$	$18.05\% \pm 0.03$
			10^{-1}	$17.34\% \pm 0.02$	$17.65\% \pm 0.02$	$\mathbf{17.29\% \pm 0.02}$
			CPU time (s)	3.686	3.795	3.766
Dermatology	–	Inhom. quadratic	10^{-7}	$1.97\% \pm 0.01$	$2.19\% \pm 0.01$	$1.97\% \pm 0.01$
			10^{-6}	$1.93\% \pm 0.01$	$1.96\% \pm 0.01$	$1.93\% \pm 0.01$
			10^{-5}	$1.98\% \pm 0.01$	$2.38\% \pm 0.01$	$1.94\% \pm 0.01$
			10^{-4}	$2.04\% \pm 0.01$	$2.12\% \pm 0.01$	$1.71\% \pm 0.01$
			10^{-3}	$1.55\% \pm 0.01$	$1.40\% \pm 0.01$	$0.73\% \pm 0.01$
			10^{-2}	$0.62\% \pm 0.01$	$\mathbf{0.51\% \pm 0.01}$	$31.00\% \pm 0.00$
			10^{-1}	$31.02\% \pm 0.00$	$30.98\% \pm 0.00$	$31.02\% \pm 0.00$
			CPU time (s)	6.156	6.178	6.200
Climate Model Crashes	–	Inhom. linear	10^{-7}	$5.27\% \pm 0.01$	$5.23\% \pm 0.01$	$5.23\% \pm 0.01$
			10^{-6}	$5.23\% \pm 0.01$	$5.27\% \pm 0.01$	$5.27\% \pm 0.01$
			10^{-5}	$5.35\% \pm 0.01$	$5.35\% \pm 0.01$	$5.34\% \pm 0.01$
			10^{-4}	$5.46\% \pm 0.01$	$5.28\% \pm 0.01$	$5.23\% \pm 0.01$
			10^{-3}	$5.43\% \pm 0.01$	$\mathbf{5.21\% \pm 0.01}$	$5.41\% \pm 0.01$
			10^{-2}	$5.46\% \pm 0.01$	$5.44\% \pm 0.01$	$6.30\% \pm 0.01$
			10^{-1}	$8.51\% \pm 0.00$	$8.52\% \pm 0.00$	$8.52\% \pm 0.00$
			CPU time (s)	19.874	20.420	19.868

Table A.6: Average out-of-sample testing errors and standard deviations over 96 runs of the robust model. Holdout: 50% training set-50% testing set.

Dataset	Data transformation	Kernel	ρ	Robust		
				$p = 1$	$p = 2$	$p = \infty$
Breast Cancer Diagnostic	Min-max normalization	Inhom. quadratic	10^{-7}	$3.06\% \pm 0.01$	$3.19\% \pm 0.01$	$3.06\% \pm 0.01$
			10^{-6}	$3.19\% \pm 0.01$	$3.19\% \pm 0.01$	$3.18\% \pm 0.01$
			10^{-5}	$2.87\% \pm 0.01$	$3.17\% \pm 0.01$	$\mathbf{2.86\% \pm 0.01}$
			10^{-4}	$3.26\% \pm 0.01$	$2.99\% \pm 0.01$	$3.21\% \pm 0.01$
			10^{-3}	$2.90\% \pm 0.01$	$3.29\% \pm 0.01$	$5.67\% \pm 0.01$
			10^{-2}	$11.14\% \pm 0.03$	$10.74\% \pm 0.03$	$37.32\% \pm 0.00$
			10^{-1}	$37.32\% \pm 0.00$	$37.32\% \pm 0.00$	$37.32\% \pm 0.00$
			CPU time (s)	23.844	24.039	24.074
Breast Cancer	Min-max normalization	Hom. quadratic	10^{-7}	$3.32\% \pm 0.01$	$3.32\% \pm 0.01$	$3.32\% \pm 0.01$
			10^{-6}	$3.22\% \pm 0.01$	$3.22\% \pm 0.01$	$3.22\% \pm 0.01$
			10^{-5}	$3.36\% \pm 0.01$	$3.36\% \pm 0.01$	$3.36\% \pm 0.01$
			10^{-4}	$3.27\% \pm 0.01$	$3.27\% \pm 0.01$	$3.23\% \pm 0.01$
			10^{-3}	$3.29\% \pm 0.01$	$3.29\% \pm 0.01$	$3.26\% \pm 0.01$
			10^{-2}	$3.24\% \pm 0.01$	$3.24\% \pm 0.01$	$3.16\% \pm 0.01$
			10^{-1}	$3.09\% \pm 0.01$	$3.09\% \pm 0.01$	$\mathbf{2.91\% \pm 0.01}$
			CPU time (s)	40.660	40.554	41.035
Blood Transfusion	Standardization	Inhom. cubic	10^{-7}	$21.61\% \pm 0.01$	$21.47\% \pm 0.01$	$21.46\% \pm 0.02$
			10^{-6}	$21.54\% \pm 0.02$	$21.48\% \pm 0.02$	$\mathbf{21.33\% \pm 0.02}$
			10^{-5}	$21.63\% \pm 0.01$	$21.63\% \pm 0.01$	$22.09\% \pm 0.01$
			10^{-4}	$23.69\% \pm 0.00$	$23.67\% \pm 0.00$	$23.80\% \pm 0.00$
			10^{-3}	$23.80\% \pm 0.00$	$23.80\% \pm 0.00$	$23.80\% \pm 0.00$
			10^{-2}	$25.38\% \pm 0.11$	$25.38\% \pm 0.11$	$30.94\% \pm 0.22$
			10^{-1}	$47.61\% \pm 0.36$	$47.61\% \pm 0.36$	$52.37\% \pm 0.37$
			CPU time (s)	52.918	52.915	52.598
Mammographic Mass	Min-max normalization	Hom. cubic	10^{-7}	$16.58\% \pm 0.02$	$16.31\% \pm 0.02$	$16.58\% \pm 0.02$
			10^{-6}	$16.46\% \pm 0.01$	$\mathbf{16.15\% \pm 0.01}$	$16.46\% \pm 0.01$
			10^{-5}	$16.51\% \pm 0.02$	$16.67\% \pm 0.01$	$16.54\% \pm 0.02$
			10^{-4}	$16.45\% \pm 0.02$	$16.39\% \pm 0.01$	$16.54\% \pm 0.01$
			10^{-3}	$17.34\% \pm 0.02$	$16.84\% \pm 0.02$	$17.86\% \pm 0.02$
			10^{-2}	$18.05\% \pm 0.02$	$18.30\% \pm 0.02$	$18.87\% \pm 0.02$
			10^{-1}	$19.86\% \pm 0.01$	$19.93\% \pm 0.01$	$19.50\% \pm 0.01$
			CPU time (s)	71.626	71.648	71.730

Table A.7: Average out-of-sample testing errors and standard deviations over 96 runs of the robust model. Holdout: 50% training set-50% testing set (continued).

Dataset	Data transformation	Kernel	ρ	Robust		
				$p = 1$	$p = 2$	$p = \infty$
Arrhythmia	–	Inhom. linear	10^{-7}	**26.70% ± 0.07**	27.47% ± 0.06	28.82% ± 0.06
			10^{-6}	28.00% ± 0.06	27.21% ± 0.06	28.15% ± 0.07
			10^{-5}	28.66% ± 0.06	28.29% ± 0.06	28.10% ± 0.06
			10^{-4}	28.78% ± 0.06	28.04% ± 0.06	27.84% ± 0.07
			10^{-3}	27.41% ± 0.06	28.68% ± 0.07	32.29% ± 0.06
			10^{-2}	31.56% ± 0.07	30.43% ± 0.07	29.41% ± 0.00
			10^{-1}	29.45% ± 0.00	29.45% ± 0.00	29.41% ± 0.00
			CPU time (s)	0.151	0.161	0.142
Parkinson	Min-max normalization	Hom. quadratic	10^{-7}	18.87% ± 0.04	18.00% ± 0.04	17.73% ± 0.04
			10^{-6}	18.37% ± 0.04	17.23% ± 0.04	18.12% ± 0.04
			10^{-5}	18.77% ± 0.04	17.84% ± 0.04	18.15% ± 0.04
			10^{-4}	17.65% ± 0.04	17.18% ± 0.04	17.62% ± 0.04
			10^{-3}	17.43% ± 0.04	17.46% ± 0.04	17.93% ± 0.04
			10^{-2}	**16.96% ± 0.04**	17.18% ± 0.04	17.07% ± 0.03
			10^{-1}	17.04% ± 0.03	17.22% ± 0.03	24.66% ± 0.00
			CPU time (s)	0.301	0.304	0.303
Heart disease	Min-max normalization	Inhom. linear	10^{-7}	19.99% ± 0.03	20.39% ± 0.03	20.97% ± 0.03
			10^{-6}	20.93% ± 0.03	20.59% ± 0.03	21.17% ± 0.03
			10^{-5}	20.91% ± 0.03	20.88% ± 0.03	20.97% ± 0.03
			10^{-4}	20.51% ± 0.03	20.25% ± 0.03	20.49% ± 0.03
			10^{-3}	20.65% ± 0.03	20.51% ± 0.02	20.31% ± 0.02
			10^{-2}	21.08% ± 0.03	19.64% ± 0.03	19.75% ± 0.02
			10^{-1}	19.98% ± 0.02	19.89% ± 0.02	**19.47% ± 0.02**
			CPU time (s)	0.643	0.640	0.649
Dermatology	Min-max normalization	Inhom. cubic	10^{-7}	2.45% ± 0.02	2.31% ± 0.02	2.11% ± 0.02
			10^{-6}	2.06% ± 0.02	2.29% ± 0.02	2.19% ± 0.02
			10^{-5}	2.46% ± 0.02	2.32% ± 0.02	**2.04% ± 0.01**
			10^{-4}	2.46% ± 0.02	2.13% ± 0.01	2.12% ± 0.02
			10^{-3}	2.23% ± 0.02	2.30% ± 0.01	25.62% ± 0.08
			10^{-2}	30.97% ± 0.00	30.97% ± 0.00	30.97% ± 0.00
			10^{-1}	30.97% ± 0.00	30.97% ± 0.00	30.97% ± 0.00
			CPU time (s)	1.015	1.028	1.050
Climate Model Crashes	–	Inhom. linear	10^{-7}	7.15% ± 0.01	7.14% ± 0.01	7.40% ± 0.02
			10^{-6}	7.19% ± 0.01	7.20% ± 0.01	7.11% ± 0.01
			10^{-5}	7.27% ± 0.01	**6.98% ± 0.01**	7.15% ± 0.01
			10^{-4}	7.27% ± 0.01	7.16% ± 0.01	7.29% ± 0.01
			10^{-3}	7.10% ± 0.01	7.07% ± 0.01	6.98% ± 0.01
			10^{-2}	7.17% ± 0.01	7.12% ± 0.01	7.71% ± 0.01
			10^{-1}	8.50% ± 0.00	8.49% ± 0.00	8.52% ± 0.00
			CPU time (s)	2.776	2.847	2.769

Table A.8: Average out-of-sample testing errors and standard deviations over 96 runs of the robust model. Holdout: 25% training set-75% testing set.

Dataset	Data transformation	Kernel	ρ	Robust		
				$p = 1$	$p = 2$	$p = \infty$
Breast Cancer Diagnostic	Standardization	Inhom. linear	10^{-7}	$4.78\% \pm 0.01$	$4.81\% \pm 0.01$	$4.60\% \pm 0.01$
			10^{-6}	$4.84\% \pm 0.01$	$4.74\% \pm 0.02$	$4.94\% \pm 0.01$
			10^{-5}	$4.65\% \pm 0.01$	$4.85\% \pm 0.01$	$4.76\% \pm 0.01$
			10^{-4}	$4.86\% \pm 0.01$	$4.86\% \pm 0.01$	$4.82\% \pm 0.01$
			10^{-3}	$4.89\% \pm 0.01$	$4.76\% \pm 0.01$	$4.79\% \pm 0.01$
			10^{-2}	$4.22\% \pm 0.01$	$4.72\% \pm 0.02$	$3.91\% \pm 0.01$
			10^{-1}	$\mathbf{3.68\% \pm 0.01}$	$3.74\% \pm 0.01$	$4.91\% \pm 0.01$
			CPU time (s)	3.242	3.271	3.231
Breast Cancer	Min-max normalization	Hom. quadratic	10^{-7}	$3.81\% \pm 0.01$	$3.73\% \pm 0.01$	$3.59\% \pm 0.01$
			10^{-6}	$3.77\% \pm 0.01$	$3.83\% \pm 0.01$	$3.65\% \pm 0.01$
			10^{-5}	$3.63\% \pm 0.01$	$3.65\% \pm 0.01$	$3.66\% \pm 0.01$
			10^{-4}	$3.57\% \pm 0.01$	$3.69\% \pm 0.01$	$3.53\% \pm 0.01$
			10^{-3}	$3.84\% \pm 0.01$	$3.97\% \pm 0.01$	$3.76\% \pm 0.01$
			10^{-2}	$3.37\% \pm 0.01$	$3.46\% \pm 0.01$	$3.25\% \pm 0.01$
			10^{-1}	$3.18\% \pm 0.01$	$3.15\% \pm 0.01$	$\mathbf{2.90\% \pm 0.00}$
			CPU time (s)	5.301	5362	5.309
Blood Transfusion	Standardization	Hom. cubic	10^{-7}	$23.21\% \pm 0.01$	$23.20\% \pm 0.02$	$23.25\% \pm 0.01$
			10^{-6}	$23.41\% \pm 0.01$	$23.28\% \pm 0.01$	$23.34\% \pm 0.01$
			10^{-5}	$23.36\% \pm 0.01$	$23.47\% \pm 0.02$	$23.19\% \pm 0.01$
			10^{-4}	$23.24\% \pm 0.01$	$23.45\% \pm 0.01$	$23.44\% \pm 0.01$
			10^{-3}	$\mathbf{23.08\% \pm 0.01}$	$23.15\% \pm 0.01$	$23.12\% \pm 0.01$
			10^{-2}	$23.34\% \pm 0.01$	$23.26\% \pm 0.02$	$23.54\% \pm 0.00$
			10^{-1}	$23.61\% \pm 0.00$	$23.58\% \pm 0.00$	$23.69\% \pm 0.00$
			CPU time (s)	6.952	6.904	6.936
Mammographic Mass	Min-max normalization	Hom. cubic	10^{-7}	$17.62\% \pm 0.01$	$17.83\% \pm 0.02$	$17.84\% \pm 0.02$
			10^{-6}	$17.99\% \pm 0.01$	$17.61\% \pm 0.01$	$\mathbf{17.59\% \pm 0.01}$
			10^{-5}	$17.62\% \pm 0.02$	$17.98\% \pm 0.01$	$17.97\% \pm 0.02$
			10^{-4}	$17.69\% \pm 0.01$	$17.62\% \pm 0.01$	$17.79\% \pm 0.02$
			10^{-3}	$17.83\% \pm 0.01$	$18.06\% \pm 0.01$	$18.22\% \pm 0.01$
			10^{-2}	$18.58\% \pm 0.01$	$18.60\% \pm 0.01$	$19.19\% \pm 0.01$
			10^{-1}	$19.82\% \pm 0.01$	$19.79\% \pm 0.01$	$19.64\% \pm 0.01$
			CPU time (s)	9.039	9.169	9.280

Table A.9: Average out-of-sample testing errors and standard deviations over 96 runs of the robust model. Holdout: 25% training set-75% testing set (continued).

Dataset	Data transformation		Mean value of features	CV of features
Arrhythmia	–	Min	2.23×10^{2}	0
		Max	6.20×10^{2}	5.29×10^{-1}
Parkinson	Min-max normalization	Min	4.40×10^{-5}	7.71×10^{-2}
		Max	1.97×10^{2}	1.63×10^{0}
Heart Disease	Standardization	Min	1.45×10^{-1}	1.35×10^{-1}
		Max	2.47×10^{2}	2.43×10^{0}
Dermatology	–	Min	1.06×10^{-1}	3.20×10^{-1}
		Max	3.63×10^{1}	4.29×10^{0}
Climate Model Crashes	–	Min	5.00×10^{-1}	5.78×10^{-1}
		Max	5.00×10^{-1}	5.78×10^{-1}
Breast Cancer Diagnostic	Min-max normalization	Min	3.79×10^{-3}	1.12×10^{-1}
		Max	8.81×10^{2}	1.13×10^{0}
Breast Cancer	Standardization	Min	1.59×10^{0}	6.41×10^{-1}
		Max	4.39×10^{0}	1.09×10^{0}
Blood Transfusion	Standardization	Min	5.51×10^{0}	7.11×10^{-1}
		Max	1.38×10^{3}	1.06×10^{0}
Mammographic Mass	Standardization	Min	2.78×10^{0}	1.59×10^{-1}
		Max	5.58×10^{1}	5.57×10^{-1}

Table A.10: Minimum and maximum values for the mean and the coefficient of variation (CV) computed feature-wise. The data transformation refers to the best choice when classifying the holdout 75%-25% with the deterministic model.

Appendix B

Appendix to Chapter 5

B.1 Multi-stage stochastic model $\mathcal{M}_{sym}$ with a two-commodity flow formulation

Sets:

$\mathcal{I} = \{i : i = 0, 1, \ldots, N, N+1\}$: set of N waste bins and the real depot 0 and the copy depot $N+1$;

$\mathcal{I}' = \{i : i = 1, \ldots, N\}$: set of N waste bins (depots excluded);

$\mathcal{T} = \{t : t = 1, \ldots, T\}$: set of stages;

$\mathcal{T}' = \{t : t = 1, \ldots, T-1\}$: set of stages (last stage excluded);

$\mathcal{T}'' = \{t : t = 2, \ldots, T\}$: set of stages (first stage excluded);

$\mathcal{N}^1 = \{n : n = 1\}$: root node at stage 1;

$\mathcal{N}^t = \{n : n = 1, \ldots, n^t\}$: set of ordered nodes of the tree at stage $t \in \mathcal{T}$.

Deterministic parameters:

C: travelling cost per distance unit;

R: selling price of a recyclable material;

Q: vehicle capacity;

B: waste density;

M: Big-M number;

d_{ij}: distance between $i \in \mathcal{I}$ and $j \in \mathcal{I}$;

S_i^{init}: percentage of waste on the total volume of bin $i \in \mathcal{I}'$ at the first stage;

E_i: capacity of bin $i \in \mathcal{I}'$;

$pa(n)$: parent of node $n \in \mathcal{N}^t$, $t \in \mathcal{T}''$.

Stochastic parameters:

a_i^n: uncertain accumulation rate of bin $i \in \mathcal{I}'$ at node $n \in \mathcal{N}^t$, $t \in \mathcal{T}''$;

π^n: probability of node $n \in \mathcal{N}^t$, $t \in \mathcal{T}$.

Decision variables:

$x_{ij}^t \in \{0,1\}$: binary variable indicating if arc (i,j) is visited at time $t+1$, with $t \in \mathcal{T}'$ and for $i,j \in \mathcal{I}$, $i \neq j$;

$y_i^t \in \{0,1\}$: binary variable indicating if waste bin $i \in \mathcal{I}'$ is visited at time $t+1$, with $t \in \mathcal{T}'$;

$f_{ij}^n \in \mathbb{R}^+$: nonnegative variable representing the flow between $i \in \mathcal{I}'$ and $j \in \mathcal{I}$, $i \neq j$, for $n \in \mathcal{N}^t$, $t \in \mathcal{T}''$;

$w_i^n \in \mathbb{R}^+$: nonnegative variable representing the amount of waste collected at waste bin $i \in \mathcal{I}'$, for $n \in \mathcal{N}^t$, $t \in \mathcal{T}''$;

$u_i^n \in \mathbb{R}^+$: nonnegative variable representing the amount of waste at waste bin $i \in \mathcal{I}'$, for $n \in \mathcal{N}^t$, $t \in \mathcal{T}$.

Model $\mathcal{M}_{sym}$:

$$
\begin{aligned}
\max \quad & R \sum_{t \in \mathcal{T}''} \sum_{n \in \mathcal{N}^t} \pi^n \sum_{i \in \mathcal{I}'} w_i^n - \frac{C}{2} \sum_{t \in \mathcal{T}'} \sum_{\substack{i,j \in \mathcal{I} \\ i \neq j}} d_{ij} x_{ij}^t \\
\text{s.t.} \quad & \sum_{\substack{j \in \mathcal{I} \\ j \neq i}} (f_{ij}^n - f_{ji}^n) = 2w_i^n \qquad i \in \mathcal{I}', n \in \mathcal{N}^t, t \in \mathcal{T}'' \\
& \sum_{i \in \mathcal{I}'} f_{iN+1}^n = \sum_{i \in \mathcal{I}'} w_i^n \qquad n \in \mathcal{N}^t, t \in \mathcal{T}'' \\
& f_{ij}^n + f_{ji}^n = Q x_{ij}^{t-1} \qquad i,j \in \mathcal{I}, i \neq j, n \in \mathcal{N}^t, t \in \mathcal{T}'' \\
& f_{ij}^n \leqslant (Q - E_j B a_j^n) x_{ij}^{t-1} \qquad i,j \in \mathcal{I}', i \neq j, n \in \mathcal{N}^t, t \in \mathcal{T}'' \\
& \sum_{\substack{i \in \mathcal{I} \\ i \neq j}} x_{ij}^t = 2y_j^t \qquad j \in \mathcal{I}', t \in \mathcal{T}' \\
& w_i^n \leqslant E_i B y_i^{t-1} \qquad i \in \mathcal{I}', n \in \mathcal{N}^t, t \in \mathcal{T}'' \\
& u_i^n \leqslant M(1 - y_i^{t-1}) \qquad i \in \mathcal{I}', n \in \mathcal{N}^t, t \in \mathcal{T}'' \\
& u_i^n = E_i B S_i^{init} \qquad i \in \mathcal{I}', n \in \mathcal{N}^1 \\
& u_i^n = u_i^{pa(n)} + E_i B a_i^n - w_i^n \qquad i \in \mathcal{I}', n \in \mathcal{N}^t, t \in \mathcal{T}'' \\
& u_i^{pa(n)} \leqslant \left(1 - a_i^n\right) E_i B \qquad i \in \mathcal{I}', n \in \mathcal{N}^t, t \in \mathcal{T}'' \\
& x_{ij}^t \in \{0,1\} \qquad i,j \in \mathcal{I}, i \neq j, t \in \mathcal{T}' \\
& y_i^t \in \{0,1\} \qquad i \in \mathcal{I}', t \in \mathcal{T}' \\
& f_{ij}^n \geqslant 0 \qquad i \in \mathcal{I}', j \in \mathcal{I}, i \neq j, n \in \mathcal{N}^t, t \in \mathcal{T}'' \\
& w_i^n \geqslant 0 \qquad i \in \mathcal{I}', n \in \mathcal{N}^t, t \in \mathcal{T}'' \\
& u_i^n \geqslant 0 \qquad i \in \mathcal{I}', n \in \mathcal{N}^t, t \in \mathcal{T}
\end{aligned}
$$

B.2 Scenario tree generation

In this section, we discuss how to generate scenario trees to describe the problem uncertainty. We adopt the methodology proposed by Kirui et al. in [81], which are based on the works of Pflug and Pichler (see [121] for details).

Since only a limited number of trajectories of the accumulation rate is available from historical data, new and additional samples are needed to be generated, even if the true distribution of the accumulation rate is not known. However, it can be estimated by a non-parametric kernel density technique discussed in the following.

Let $\big(a_{i,o}^{(1)},\dots,a_{i,o}^{(t)},\dots,a_{i,o}^{(T)}\big)$ be the vector denoting the accumulation rate of bin i for week of observation o, with $o=1,\dots,N_o$. Let $k(\cdot)$ be a kernel function and $\big(p_1,\dots,p_o,\dots,p_{N_o}\big)$ be a N_o-dimensional vector of positive weights such that $\sum_{o=1}^{N_o} p_o = 1$. Let α be a random number drawn from the uniform distribution $\mathcal{U}\big(0,1\big)$. At stage $t=1,\dots,T$, a new sample $\widehat{a}_i^{(t)}$ of the accumulation rate of bin i is given by:

$$\widehat{a}_i^{(t)} = a_{i,o^*}^{(t)} + h^{(t)} \cdot K^{(t)},$$

where:

- o^* is an index between 1 and N_o such that $\sum_{o=1}^{o^*-1} p_o < \alpha \leqslant \sum_{o=1}^{o^*} p_o$;
- $h^{(t)}$ is the bandwidth, computed according to the Silverman's rule of thumb (see [136]), namely $h^{(t)} = \sigma^{(t)} \cdot N_o^{-\frac{1}{m^{(t)}+4}}$, being $\sigma^{(t)}$ the standard deviation of data at stage t and $m^{(t)}$ the dimension of the process at stage t;
- $K^{(t)}$ is a random value sampled from the kernel distribution $k(\cdot)$ at stage t.

Before computing a new sample at stage $t+1$, each weight p_o is updated according to the formula $p_o \cdot (h^{(t)})^{-m^{(t+1)}} \cdot k\big(\frac{\widehat{a}_i^{(t)} - a_{i,o}^{(t)}}{h^{(t)}}\big)$, and then normalized. Further, a random number α is drawn anew.
Using this procedure, the conditional density $g_i^{(t+1)}$ of the accumulation rate of bin i at stage $t+1$, given $\widehat{a}_i^{(1)},\dots,\widehat{a}_i^{(t)}$, can be estimated by:

$$\widehat{g}_i^{(t+1)}\big(\widehat{a}_i^{(t+1)} \big| \widehat{a}_i^{(1)},\dots,\widehat{a}_i^{(t)}\big) = \sum_{o=1}^{N_o} p_o \cdot (h^{(N_o)})^{-m^{(t+1)}} \cdot k\left(\frac{\widehat{a}_i^{(t+1)} - a_{i,o}^{(t+1)}}{h^{N_o}}\right).$$

Within this approach, every new trajectory starts at $\widehat{a}_i^{(1)}$, and new samples $\widehat{a}_i^{(t+1)}$ are generated according to the density $\widehat{g}_i^{(t+1)}$, for $t=1,\dots,T-1$. At the end of the procedure at stage T, a new trajectory $(\widehat{a}_i^{(1)},\dots,\widehat{a}_i^{(T)})$ has been generated from the initial data.

We set $a_{i,o}^{(1)} = 0 = \widehat{a}_i^{(1)}$ for all $i = 1, \ldots, N$, $o = 1, \ldots, N_o$ because no increase of waste at the first stage of the time horizon is assumed, and $m^{(t)} = N$ for all $t = 1, \ldots, T$ since, at each node, the dimension of the state corresponds to the total number of bins. Furthermore, as suggested in [81], the kernel $k(\cdot)$ is set to be logistic. Figure B.1 shows one hundred trajectories of the accumulation rate in six different bins, generated according to the conditional density estimation process described so far.

Secondly, we apply a dynamic stochastic approximation algorithm to generate a candidate scenario tree (see [121] for details). Starting from an initial guess of a tree with a prescribed branching structure, at every iteration of the procedure a new sample path is generated according to the conditional density estimation process discussed above. The algorithm finds one possible sequence of nodes in the scenario tree whose distance between the states of those nodes and the generated sample is minimal. Thus, the states of those nodes are updated with the values of the generated sample and the others remain unchanged. Then, the algorithm calculates the conditional probabilities to reach each node of the tree starting from its root, and it stops when all the iterations, whose number is decided in advance, have been performed.

The scenario tree generation procedure described so far has been implemented in Julia, relying on the package ScenTrees.jl (see [82]). The number of iterations for the stochastic approximation process has been set to 10000.

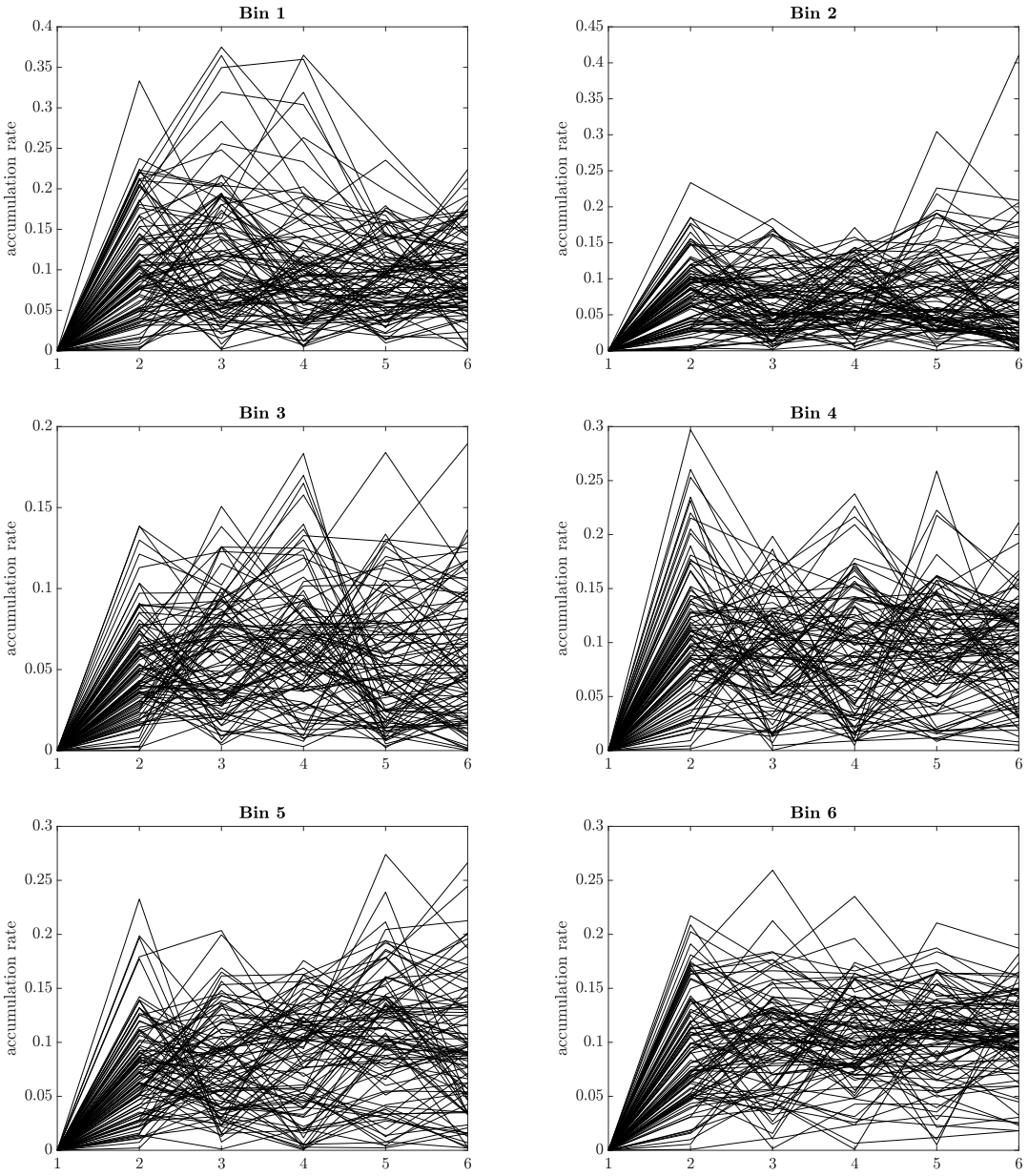


Figure B.1: For each of the six bins, one hundred trajectories on the accumulation rate of waste generated from historical data through the conditional density estimation process are depicted. The stages are represented on the horizontal axis.

B.3 In-sample stability

In this section, we carry out an in-sample stability analysis (see [75]).

In Table B.1 we report average results obtained by solving model $\mathcal{M}$ over five runs on *inst_9_1*, with increasing size of the scenario tree. Box-plots of objective function and of weight of collected waste are depicted in Figure B.2.

Scenarios	Branching structure	Profit (€)	Weight of waste (kg)	Distance (km)	CPU time (s)	Multistage distance
32	[1 2 2 2 2 2]	7.43	267.58	72.84	55.22	0.063
72	[1 3 3 2 2 2]	7.61	268.18	72.84	166.24	0.046
162	[1 3 3 3 3 2]	7.42	267.52	72.84	1165.00	0.034
324	[1 4 3 3 3 3]	7.63	268.22	72.84	7109.18	0.027
576	[1 4 4 4 3 3]	7.50	267.81	72.84	51928.64	0.020
1024	[1 4 4 4 4 4]	Not solved to optimality within 24 hours				0.016

Table B.1: Average results on the in-sample stability analysis over five runs on scenario trees with increasing size. The results are drawn from model $\mathcal{M}$ on *inst_9_1*.

Since various indicators (profit, weight of collected waste, total travelled distance) do not vary significantly when increasing the size of the tree, we conclude that the methodology we applied to generate scenario trees is stable even with small trees. Besides, the multistage distance (see the last column of Table B.1), is throughout close to zero, due to the minimization of the distance in the dynamic stochastic approximation algorithm. On the other hand, the computational time increases considerably, when increasing the size of the tree.

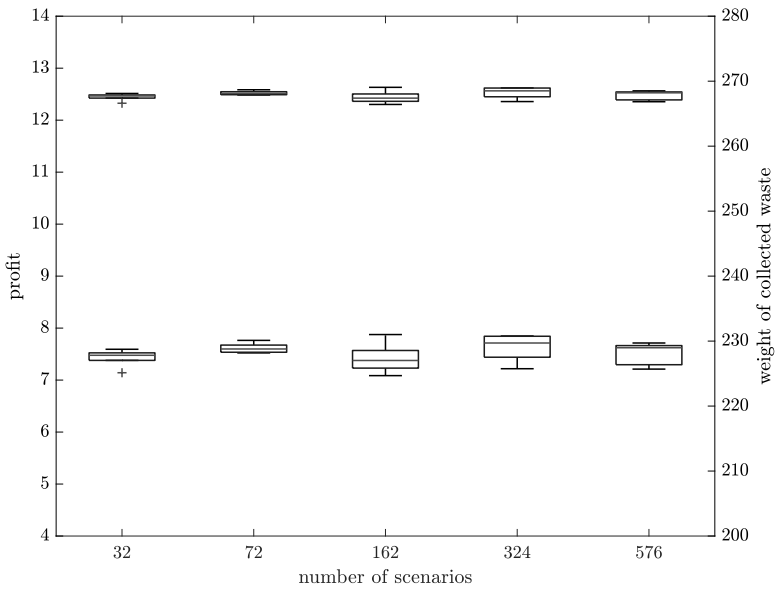


Figure B.2: Box-plots of objective function value (below, left-hand scale) and of weight of collected waste (above, right-hand scale) over 5 runs of scenario trees with increasing cardinality.

For all of these reasons, we decide to consider a scenario tree of size $S = 32$, with 63 nodes. In Figure B.3 we depict six binary scenario trees of six different bins with the corresponding probability distributions generated from the dynamic stochastic approximation algorithm.

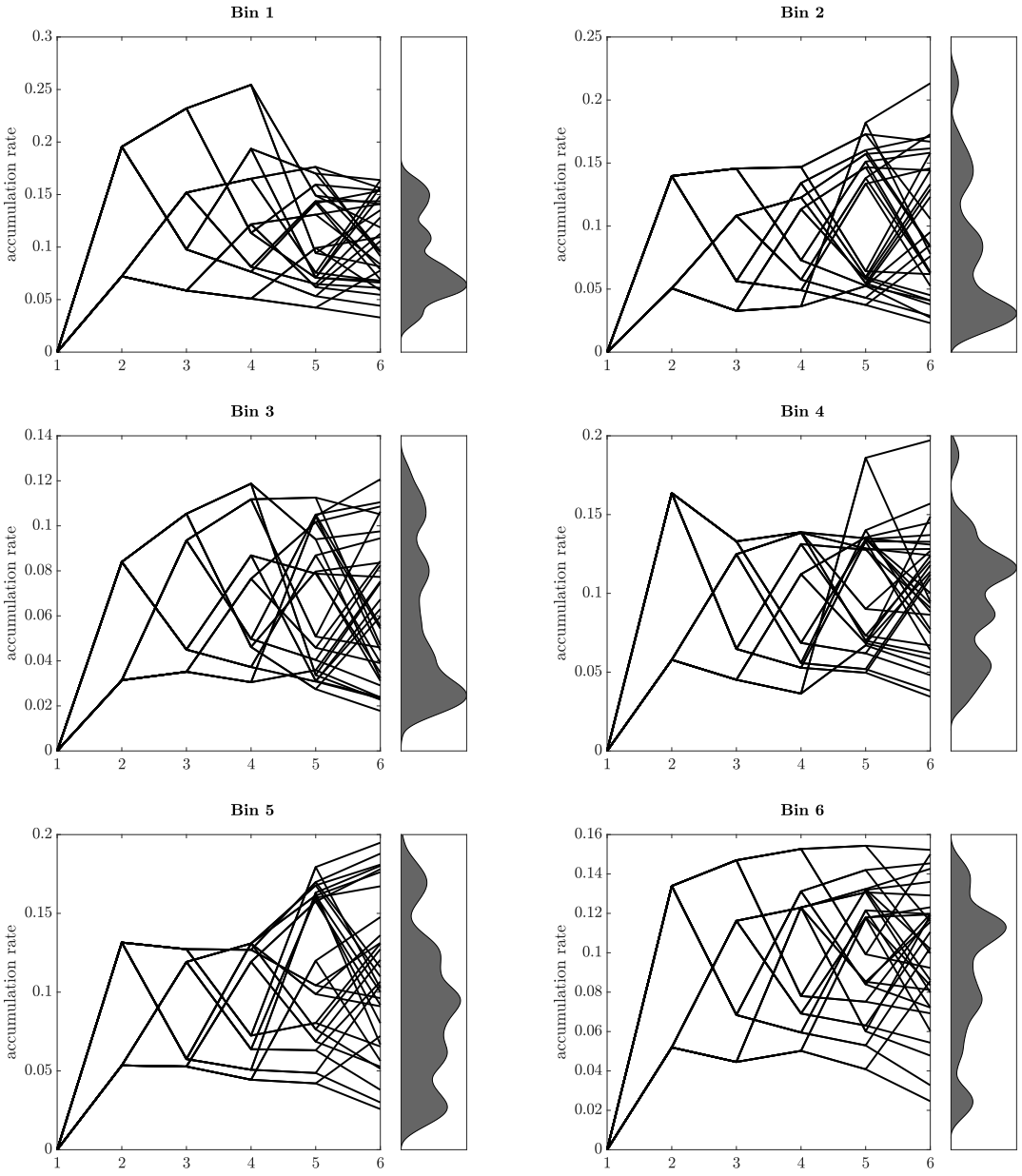


Figure B.3: Examples of six-stages scenario trees of the accumulation rate of waste in six different bins. The corresponding probability distribution is depicted on the right of each plot.

B.4 Stochastic measures (detailed results for small instances)

	inst_1_9	*inst_2_9*	*inst_3_9*	*inst_4_9*	*inst_5_9*	*inst_6_9*	*inst_7_9*	*inst_8_9*	*inst_9_9*	*inst_10_9*
RP	9.36	11.92	31.58	32.66	2.48	4.27	30.76	22.90	2.72	32.96
EV	23.79	14.43	38.23	38.58	17.24	18.10	36.85	45.80	18.60	42.82
WS	17.63	17.46	45.50	37.94	16.40	15.88	35.98	25.40	24.04	45.68
$\%EVPI$	88%	46%	44%	16%	562%	272%	17%	11%	783%	39%
$\%VSS^1$	∞	∞	∞	77%	∞	∞	∞	∞	∞	∞
$\%VSS^2$	∞	∞	∞	77%	∞	∞	∞	∞	∞	∞
$\%VSS^3$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%VSS^4$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%VSS^5$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%MLUSS^1$	∞	∞	∞	77%	∞	∞	∞	∞	∞	∞
$\%MLUSS^2$	∞	∞	∞	77%	∞	∞	∞	∞	∞	∞
$\%MLUSS^3$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%MLUSS^4$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%MLUSS^5$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%MLUDS^1$	0%	0%	0%	77%	0%	0%	0%	0%	0%	0%
$\%MLUDS^2$	546%	500%	179%	77%	1992%	1131%	174%	235%	1760%	150%
$\%MLUDS^3$	546%	500%	179%	77%	1992%	1131%	174%	235%	1760%	150%
$\%MLUDS^4$	546%	500%	179%	77%	1992%	1131%	174%	235%	1760%	150%
$\%MLUDS^5$	546%	500%	179%	147%	1992%	1131%	174%	235%	1760%	150%

Table B.2: Detailed results of RP, EV, WS and of stochastic measures $\%EVPI$, $\%VSS^t$, $\%MLUSS^t$, $\%MLUDS^t$, for $1 \leqslant t \leqslant 5$. The values in percentage denote the gap with respect to the corresponding RP problem. The results refer to the instances with 9 bins.

	inst_1_10	*inst_2_10*	*inst_3_10*	*inst_4_10*	*inst_5_10*	*inst_6_10*	*inst_7_10*	*inst_8_10*	*inst_9_10*	*inst_10_10*
RP	14.57	25.97	53.88	54.09	16.41	32.07	41.28	35.71	33.16	36.59
EV	32.06	28.48	53.88	58.82	22.50	32.07	48.12	40.65	40.45	36.59
WS	34.42	32.40	57.08	63.13	29.61	35.38	47.91	49.40	43.67	38.10
$\%EVPI$	136%	25%	6%	17%	80%	10%	16%	38%	32%	4%
$\%VSS^1$	∞	∞	∞	∞	∞	∞	0%	∞	∞	∞
$\%VSS^2$	∞	∞	∞	∞	∞	∞	55%	∞	∞	∞
$\%VSS^3$	∞	∞	∞	∞	∞	∞	55%	∞	∞	∞
$\%VSS^4$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%VSS^5$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%MLUSS^1$	∞	∞	∞	∞	∞	∞	0%	∞	∞	∞
$\%MLUSS^2$	∞	∞	∞	∞	∞	∞	55%	∞	∞	∞
$\%MLUSS^3$	∞	∞	∞	∞	∞	∞	55%	∞	∞	∞
$\%MLUSS^4$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%MLUSS^5$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%MLUDS^1$	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
$\%MLUDS^2$	316%	229%	95%	87%	348%	160%	55%	153%	157%	147%
$\%MLUDS^3$	316%	229%	95%	87%	348%	160%	55%	153%	157%	147%
$\%MLUDS^4$	316%	229%	95%	87%	348%	160%	55%	153%	157%	147%
$\%MLUDS^5$	316%	229%	95%	87%	348%	160%	55%	153%	157%	147%

Table B.3: Detailed results of RP, EV, WS and of stochastic measures $\%EVPI$, $\%VSS^t$, $\%MLUSS^t$, $\%MLUDS^t$, for $1 \leqslant t \leqslant 5$. The values in percentage denote the gap with respect to the corresponding RP problem. The results refer to the instances with 10 bins.

	inst_1_11	*inst_2_11*	*inst_3_11*	*inst_4_11*	*inst_5_11*	*inst_6_11*	*inst_7_11*	*inst_8_11*	*inst_9_11*	*inst_10_11*
RP	30.83	38.46	64.24	33.12	46.72	50.21	60.99	29.87	15.73	42.31
EV	32.38	41.03	66.18	33.12	49.84	53.15	61.94	34.10	26.57	52.39
WS	40.32	41.96	65.57	46.96	51.60	52.02	62.36	36.42	25.93	48.32
$\%EVPI$	31%	9%	2%	42%	10%	4%	2%	22%	65%	14%
$\%VSS^1$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%VSS^2$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%VSS^3$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%VSS^4$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%VSS^5$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%MLUSS^1$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%MLUSS^2$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%MLUSS^3$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%MLUSS^4$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%MLUSS^5$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$\%MLUDS^1$	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
$\%MLUDS^2$	200%	155%	100%	156%	129%	124%	93%	166%	326%	131%
$\%MLUDS^3$	200%	155%	100%	156%	129%	124%	93%	166%	326%	131%
$\%MLUDS^4$	200%	155%	100%	156%	129%	124%	93%	166%	326%	131%
$\%MLUDS^5$	200%	155%	100%	156%	129%	124%	93%	166%	326%	131%

Table B.4: Detailed results of RP, EV, WS and of stochastic measures $\%EVPI$, $\%VSS^t$, $\%MLUSS^t$, $\%MLUDS^t$, for $1 \leqslant t \leqslant 5$. The values in percentage denote the gap with respect to the corresponding RP problem. The results refer to the instances with 11 bins.

B.5 Performance of the rolling horizon approach (detailed results for small instances)

	inst_1_9	*inst_2_9*	*inst_3_9*	*inst_4_9*	*inst_5_9*	*inst_6_9*	*inst_7_9*	*inst_8_9*	*inst_9_9*	*inst_10_9*
W	**Profit reduction (%)**									
1	11%	∞	37%	8%	74%	0%	32%	∞	0%	7%
2	11%	95%	37%	8%	74%	0%	32%	54%	0%	7%
3	11%	95%	37%	8%	74%	0%	32%	54%	0%	7%
4	11%	0%	37%	8%	74%	0%	32%	0%	0%	7%
W	**Computational time reduction (%)**									
1	94%	99%	99%	94%	99%	99%	100%	95%	100%	100%
2	85%	76%	96%	81%	98%	98%	99%	79%	100%	97%
3	67%	49%	90%	49%	83%	96%	96%	49%	99%	91%
4	40%	−9%	75%	5%	85%	79%	94%	−36%	98%	67%

Table B.5: Detailed results on the performance of the rolling horizon approach, in terms of reduction of the profit and of the CPU time when compared to the RP problem. The results refer to the instances with 9 bins.

	inst_1_10	*inst_2_10*	*inst_3_10*	*inst_4_10*	*inst_5_10*	*inst_6_10*	*inst_7_10*	*inst_8_10*	*inst_9_10*	*inst_10_10*
W	**Profit reduction (%)**									
1	0%	∞	59%	4%	68%	15%	∞	26%	17%	∞
2	0%	50%	20%	4%	68%	15%	55%	26%	17%	36%
3	0%	50%	20%	4%	68%	15%	55%	26%	17%	36%
4	0%	0%	0%	4%	68%	36%	0%	26%	17%	0%
W	**Computational time reduction (%)**									
1	99%	87%	100%	99%	99%	100%	100%	100%	99%	100%
2	95%	56%	97%	91%	97%	99%	97%	95%	94%	99%
3	89%	0%	91%	79%	86%	94%	89%	86%	84%	97%
4	78%	−82%	75%	41%	28%	37%	86%	42%	−151%	82%

Table B.6: Detailed results on the performance of the rolling horizon approach, in terms of reduction of the profit and of the CPU time when compared to the RP problem. The results refer to the instances with 10 bins.

	inst_1_11	*inst_2_11*	*inst_3_11*	*inst_4_11*	*inst_5_11*	*inst_6_11*	*inst_7_11*	*inst_8_11*	*inst_9_11*	*inst_10_11*
W	**Profit reduction (%)**									
1	49%	93%	28%	46%	11%	26%	54%	16%	48%	10%
2	49%	34%	28%	46%	11%	26%	15%	16%	48%	10%
3	49%	34%	28%	46%	11%	26%	15%	16%	48%	10%
4	0%	0%	0%	54%	0%	0%	0%	16%	48%	10%
W	**Computational time reduction (%)**									
1	100%	98%	98%	84%	100%	100%	100%	98%	99%	100%
2	99%	85%	84%	30%	96%	99%	96%	91%	97%	99%
3	95%	−490%	61%	−23%	91%	98%	69%	55%	93%	96%
4	82%	−141%	−36%	−161%	80%	93%	66%	10%	10%	90%

Table B.7: Detailed results on the performance of the rolling horizon approach, in terms of reduction of the profit and of the CPU time when compared to the RP problem. The results refer to the instances with 11 bins.

Bibliography

[1] D. Aksen, O. Kaya, F. S. Salman, and Y. Akça. Selective and periodic inventory routing problem for waste vegetable oil collection. *Optimization Letters*, 6:1063–1080, 2012.

[2] D. Aksen, O. Kaya, F. Sibel Salman, and O. Tüncel. An adaptive large neighborhood search algorithm for a selective and periodic inventory routing problem. *European Journal of Operational Research*, 239(2):413–426, 2014.

[3] E. Angelelli and M. G. Speranza. The periodic vehicle routing problem with intermediate facilities. *European Journal of Operational Research*, 137(2):233–247, 2002.

[4] C. Angulo, X. Parra, and A. Català. K-svcr. a support vector machine for multi-class classification. *Neurocomputing*, 55:57–77, 2003.

[5] M. Arun Kumar and M. Gopal. Least squares twin support vector machines for pattern classification. *Expert Systems with Applications*, 36(4):7535–7543, 2009.

[6] R. Baldacci, E. Hadjiconstantinou, and A. Mingozzi. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Operations Research*, 52(5):723–738, 2004.

[7] A. P. Barbosa-Póvoa, C. da Silva, and A. Carvalho. Opportunities and challenges in sustainable supply chain: An operations research perspective. *European Journal of Operational Research*, 268(2):399–431, 2018.

[8] A. Ben-Tal, S. Bhadra, C. Bhattacharyya, and A. Nemirovski. Efficient methods for robust classification under uncertainty in kernel matrices. *Journal of Machine Learning Research*, 13:2923–2954, 2012.

[9] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. Robust optimization. Princeton University Press, 2009.

[10] Y. Bengio, A. Lodi, and A. Prouvost. Machine learning for combinatorial optimization: A methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405–421, 2021.

[11] K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods & Software*, 1:23–34, 1992.

[12] L. Bertazzi and F. Maggioni. A stochastic multi-stage fixed charge transportation problem: worst-case analysis of the rolling horizon approach. *European Journal of Operations Research*, 267:555–569, 2018.

[13] D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM review*, 53:464–501, 2011.

[14] D. Bertsimas, J. Dunn, C. Pawlowski, and Y. D. Zhuo. Robust classification. *INFORMS Journal of Optimization*, 1:2–34, 2019.

[15] S. Bhadra, S. Bhattacharya, C. Bhattacharyya, and A. Ben-Tal. Robust formulations for handling uncertainty in kernel matrices. *Proceedings for the 27th International Conference on Machine Learning*, pages 71–78, 2010.

[16] C. Bhattacharyya. Robust classification of noisy data using second order cone programming approach. In *International Conference on Intelligent Sensing and Information Processing, 2004*, pages 433–438, 2004.

[17] J. Bi and T. Zhang. Support vector classification with input data uncertainty. In *Advances in neural information processing systems*, pages 161–168, 2005.

[18] X. Bing, J. M. Bloemhof, T. R. P. Ramos, A. P. Barbosa-Póvoa, C. Y. Wong, and J. G. van der Vorst. Research challenges in municipal solid waste logistics management. *Waste Management*, 48:584–592, 2016.

[19] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer & Science Business Media, 2011.

[20] V. Blanco, J. Puerto, and A. M. Rodríguez-Chía. On lp-support vector machines and multidimensional kernels. *Journal of Machine Learning Research*, 21:1–29, 2020.

[21] M. B. Bogh, H. Mikkelsen, and S. Wøhlk. Collection of recyclables from cubes a case study. *Socio-Economic Planning Sciences*, 48(2):127–134, 2014.

[22] B. E. Boser, I. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual Workshop of Computational Learning Theory*, 5:144–152, 1992.

[23] E. J. Bredensteiner and K. P. Bennett. Multicategory classification by support vector machines. *Computational Optimization and Applications*, 12:53–79, 1999.

[24] M. Bruglieri, S. Mancini, F. Pezzella, and O. Pisacane. A path-based solution approach for the green vehicle routing problem. *Computers & Operations Research*, 103:109–122, 2019.

[25] L. E. Cárdenas-Barrón, J. L. González-Velarde, G. Treviño-Garza, and D. Garza-Nuñez. Heuristic algorithm based on reduce and optimize approach for a selective and periodic inventory routing problem in a waste vegetable oil collection environment. *International Journal of Production Economics*, 211:44–59, 2019.

[26] L. E. Cárdenas-Barrón and R. A. Melo. A fast and effective mip-based heuristic for a selective and periodic inventory routing problem in reverse logistics. *Omega*, 103:102394, 2021.

[27] A. Carreño, I. Inza, and J. A. Lozano. Analyzing rare event, anomaly, novelty and outlier detection terms under the supervised classification framework. *Artificial Intelligence Review*, 53:3575 – 3594, 2020.

[28] R. Cavagnini, L. Bertazzi, and F. Maggioni. A rolling horizon approach for a multi-stage stochastic fixed-charge transportation problem with transshipment. *European Journal of Operations Research*, 301:912–922, 2022.

[29] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 408:189–215, 2020.

[30] F. F. Chamasemani and Y. P. Singh. Multi-class support vector machine (svm) classifiers – an application in hypothyroid detection and classification. In *2011 Sixth International Conference on Bio-Inspired Computing: Theories and Applications*, 2011.

[31] S. Chand, V. Hsu, and S. Sethi. Forecast, solution, and rolling horizons in operations management problems: A classified bibliography. *Manufacturing & Service Operations Management*, 4(1):25–43, 2002.

[32] S. Chand, V. N. Hsu, and S. Sethi. Forecast, solution, and rolling horizons in operations management problems: A classified bibliography. *Manufacturing & Service Operations Management*, 4:25–43, 2002.

[33] S.-G. Chen and X. Wu. A new fuzzy twin support vector machine for pattern classification. *International Journal of Machine Learning and Cybernetics*, 9(9):1553–1564, 2018.

[34] T. Y. Chen, T. H. Tse, and Y.-T. Yu. Proportional sampling strategy: a compendium and some insights. *The Journal of Systems and Software*, 58(1):65–81, 2001.

[35] X. Chen, J. Yang, Q. Ye, and J. Liang. Recursive projection twin support vector machine via within-class variance minimization. *Pattern Recognition*, 44(10):2643–2655, 2011.

[36] Z.-Y. Chen, Z.-P. Fan, and M. Sun. A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. *European Journal of Operational Research*, 223(2):461–472, 2012.

[37] L. C. Coelho, J.-F. Cordeau, and G. Laporte. Thirty years of inventory routing. *Transportation Science*, 48(1):1–19, 2014.

[38] C. Cortes and V. N. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[39] K. W. De Bock, K. Coussement, A. D. Caigny, R. Słowiński, B. Baesens, R. N. Boute, T.-M. Choi, D. Delen, M. Kraus, S. Lessmann, S. Maldonado, D. Martens, M. Óskarsdóttir, C. Vairetti, W. Verbeke, and R. Weber. Explainable ai for operational research: A defining framework, methods, applications, and a research agenda. *European Journal of Operational Research*, in press, 2023.

[40] R. De Leone, F. Maggioni, and A. Spinelli. A multiclass robust twin parametric margin support vector machine with an application tovehicles emissions. In G. Nicosia, V. Ojha, E. La Malfa, G. La Malfa, P. M. Pardalos, and R. Umeton, editors, *Machine Learning, Optimization, and Data Science*, volume 14506 of *Lecture Notes in Computer Science*, pages 299–310, Cham, 2024. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-53966-4_22.

[41] R. De Leone, F. Maggioni, and A. Spinelli. A robust twin parametric margin support vector machine for multiclass classification. *Under review in Computers & Operations Research*, 2024. http://arxiv.org/abs/2306.06213.

[42] C. S. de Morais, D. R. R. Jorge, A. R. Aguiar, A. P. Barbosa-Póvoa, A. P. Antunes, and T. R. P. Ramos. A solution methodology for a smart waste collection routing problem with workload concerns: computational and managerial insights from a real case study. *International Journal of Systems Science: Operations & Logistics*, pages 1–31, 2022.

[43] C. S. de Morais, T. R. P. Ramos, and A. P. Barbosa-Póvoa. Dynamic approaches to solve the smart waste collection routing problem. In M. J. Alves, J. P. Almeida, J. F. Oliveira, and A. A. Pinto, editors, *Operational Research*, pages 173–188, 2019.

[44] S. Ding and X. Hua. Recursive least squares projection twin support vector machines for nonlinear classification. *Neurocomputing*, 130:3–9, 2014. Track on Intelligent Computing and Applications Complex Learning in Connectionist Networks.

[45] S. Ding, X. Zhao, J. Zhang, X. Zhang, and Y. Xue. A review on multi-class twsvm. *Artificial Intelligence Review*, 52:775–801, 2019.

[46] M. Doumpos, C. Zopounidis, D. Gounopoulos, E. Platanakis, and W. Zhang. Operational research and artificial intelligence methods in banking. *European Journal of Operational Research*, 306(1):1–16, 2023.

[47] S.-W. Du, M.-C. Zhang, P. Chen, H.-F. Sun, W.-J. Chen, and Y.-H. Shao. A multiclass nonparallel parametric-margin support vector machine. *Information*, 12(12):515–533, 2021.

[48] L. El Ghaoui, G. R. G. Lanckriet, G. Natsoulis, et al. Robust classification with interval data. In *Computer Science Division, University of California Berkeley*, 2003.

[49] M. Elbek and S. Wøhlk. A variable neighborhood search for the multi-period collection of recyclable materials. *European Journal of Operational Research*, 249(2):540–550, 2016.

[50] Environmental Protection Agency, 2014. Federal Register, 79(81), 23414-23886.

[51] European Commission. A new Circular Economy Action Plan, 2020. Accessed on November 7, 2022.

[52] European Parliament and the Council, 1999. Directive 19994/94/EC.

[53] D. Faccini, F. Maggioni, and F. A. Potra. Robust and distributionally robust optimization models for linear support vector machine. *Computers and Operations Research*, 147:105930, 2022.

[54] M. Faccio, A. Persona, and G. Zanin. Waste collection multi objective model with real time traceability data. *Waste Management*, 31:2391–2405, 2011.

[55] E. Fadda, L. Gobbato, G. Perboli, M. Rosano, and R. Tadei. Waste collection in urban areas: A case study. *Interfaces*, 48(4):307–322, 2018.

[56] N. Fan, E. Sadeghi, and P. M. Pardalos. Robust support vector machines with polyhedral uncertainty of the input data. In *Learning and Intelligent Optimization. International Conference on Learning and Intelligent Optimization*, pages 291–305. Springer-Verlag, 2014.

[57] A. Farid, F. Hussain, K. Khan, M. Shahzad, U. Khan, and Z. Mahmood. A fast and accurate real-time vehicle detection method using deep learning for unconstrained environments. *Applied Sciences*, 13(5):30–59, 2023.

[58] G. Fung, O. L. Mangasarian, and J. W. Shavlik. Knowledge-based support vector machine classifiers. In *NIPS*, pages 521–528, 2002.

[59] C. Gambella, B. Ghaddar, and J. Naoum-Sawaya. Optimization problems for machine learning: A survey. *European Journal of Operational Research*, 290(3):807–828, 2021.

[60] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[61] M. Gendreau, G. Laporte, and R. Séguin. Stochastic vehicle routing. *European Journal of Operational Research*, 88(1):3–12, 1996.

[62] G. Ghiani, D. Laganà, E. Manni, R. Musmanno, and D. Vigo. Operations research in solid waste management: A survey of strategic and tactical issues. *Computers & Operations Research*, 44:22–32, 2014.

[63] S. Ghosal and W. Wiesemann. The distributionally robust chance-constrained vehicle routing problem. *Operations Research*, 68:716–732, 2020.

[64] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. http://stanford.edu/~boyd/graph_dcp.html.

[65] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx, Mar. 2014.

[66] B. R. Gunnarsson, S. vanden Broucke, B. Baesens, M. Óskarsdóttir, and W. Lemahieu. Deep learning for credit scoring: Do or don't? *European Journal of Operational Research*, 295(1):292–305, 2021.

[67] H. Guo, Y. Li, J. S. Shang, M. Gu, Y. Huang, and G. Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73(1):220–239, 2017.

[68] J. Han, M. Kamber, and J. Pei. *Data mining: concepts and techniques - 3rd edition*. Morgan Kaufmann, 2011.

[69] P.-Y. Hao. New support vector algorithms with parametric insensitive/margin model. *Neural networks : the official journal of the International Neural Network Society*, 23(1):60–73, 2010.

[70] C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

[71] Jayadeva, R. Khemchandani, and S. Chandra. Twin support vector machines for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):905–910, 2007.

[72] J. Jiang and S. Peng. Mathematical programs with distributionally robust chance constraints: Statistical robustness, discretization and reformulation. *European Journal of Operational Research*, 313(2):616–627, 2024.

[73] A. Jiménez-Cordero, J. M. Morales, and S. Pineda. A novel embedded min-max approach for feature selection in nonlinear support vector machine classification. *European Journal of Operational Research*, 293:24–35, 2021.

[74] X. Ju and Y. Tian. Knowledge-based support vector machine classifiers via nearest points. *Procedia Computer Science*, 9:1240–1248, 2012.

[75] M. Kaut and S. W. Wallace. Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization*, 3(2):257–271, 2007.

[76] M. Kelly, R. Longjohn, and K. Nottingham. UCI machine learning repository, 2023. http://archive.ics.uci.edu/ml.

[77] S. S. Ketkov. A study of distributionally robust mixed-integer programming with wasserstein metric: on the value of incomplete data. *European Journal of Operational Research*, 313(2):602–615, 2024.

[78] R. Khanjani-Shiraz, A. Babapour-Azar, Z. Hosseini-Nodeh, and P. M. Pardalos. Distributionally robust joint chance-constrained support vector machines. *Optimization Letters*, 17:299–332, 2023.

[79] J.-H. Kim. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational Statistics and Data Analysis*, 53:3735–3745, 2009.

[80] A. J. King and S. W. Wallace. *Modeling with Stochastic Programming*. Springer, New York, NY, 2012.

[81] K. Kirui, G. C. Pflug, and A. Pichler. New algorithms and fast implementations to approximate stochastic processses. 2020. http://arxiv.org/abs/2012.01185.

[82] K. Kirui, A. Pichler, and G. Ch. Pflug. Scentrees.jl: A Julia package for generating scenario trees and scenario lattices for multistage stochastic programming. *Journal of Open Source Software*, 5:1912, 2020.

[83] H. Krikke, I. le Blanc, M. van Krieken, and H. Fleuren. Low-frequency collection of materials disassembled from end-of-life vehicles: On the value of on-line monitoring in optimizing route planning. *International Journal of Production Economics*, 111(2):209–228, 2008.

[84] M. Labbé, L. I. Martínez-Merino, and A. M. Rodríguez-Chía. Mixed integer linear programming for feature selection in support vector machine. *Discrete Applied Mathematics*, 261:276–304, 2019.

[85] G. R. G. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan. A robust minimax approach to classification. *Journal of Machine Learning Research*, 3:555–582, 2002.

[86] Y.-J. Lee, O. L. Mangasarian, and W. H. Wolberg. Breast cancer survival and chemotherapy: a support vector machine analysis. *Discrete mathematical problems with medical applications*, 55:1–10, 2000.

[87] S. Lessmann, B. Baesens, H.-V. Seow, and L. C. Thomas. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 247(1):124–136, 2015.

[88] H. Li, Y. Liang, and Q. Xu. Support vector machines and its applications in chemistry. *Chemometrics and Intelligent Laboratory Systems*, 95(2):188–198, 2009.

[89] Z. Li, S. Tang, and S. Yan. Multi-class svm classifier based on pairwise coupling. In S.-W. Lee and A. Verri, editors, *Pattern Recognition with Support Vector Machines*, pages 321–333, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[90] X. Liu and F. A. Potra. Pattern separation and prediction via linear and semidefinite programming. *Studies in Informatics and Control*, 18(1):71–82, 2009.

[91] J. López, S. Maldonado, and M. Carrasco. A robust formulation for twin multiclass support vector machine. *Applied Intelligence*, 47:1031–1043, 2017.

[92] J. López, S. Maldonado, and M. Carrasco. Double regularization methods for robust feature selection and svm classification via dc programming. *Information Sciences*, 429:377–389, 2018.

[93] J. López, S. Maldonado, and M. Carrasco. Robust nonparallel support vector machines via second-order cone programming. *Neurocomputing*, 364:227–238, 2019.

[94] J. Luo, X. Yan, and Y. Tian. Unsupervised quadratic surface support vector machine with application to credit risk assessment. *European Journal of Operational Research*, 280(3):1008–1017, 2020.

[95] F. Maggioni, E. Allevi, and M. Bertocchi. Bounds in multistage linear stochastic programming. *Journal of Optimization Theory and Applications*, 163:200–229, 2014.

[96] F. Maggioni, D. Faccini, F. Gheza, F. Manelli, and G. Bonetti. Machine learning based classification models for covid-19 patients. In R. Aringhieri, F. Maggioni, E. Lanzarone, M. Reuter-Oppermann, G. Righini, and M. T. Vespucci, editors, *Operations Research for Health Care in Red Zone*, pages 35–46, Cham, 2023. Springer International Publishing.

[97] F. Maggioni, M. Kaut, and L. Bertazzi. Stochastic optimization models for a single-sink transportation problem. *Computational Management Science*, 6(2):251–267, 2009.

[98] F. Maggioni and A. Spinelli. A novel robust optimization model for nonlinear support vector machine. *Under review in European Journal of Operational Research*, 2024. http://arxiv.org/abs/2306.06223.

[99] F. Maggioni and A. Spinelli. A robust nonlinear support vector machine approach for vehicles smog rating classification. In M. Bruglieri, P. Festa, G. Macrina, and O. Pisacane, editors, *Optimization in Green Sustainability and Ecological Transition*, AIRO Springer Series. Springer Cham, 2024. https://doi.org/10.1007/978-3-031-47686-0_19.

[100] S. Maldonado, J. López, and M. Carrasco. A second-order cone programming formulation for twin support vector machines. *Applied Intelligence*, 45:265–276, 2016.

[101] S. Maldonado, J. López, and M. Carrasco. The cobb-douglas learning machine. *Pattern Recognition*, 128:108701, 2022.

[102] S. Maldonado, J. López, and C. Vairetti. Profit-based churn prediction based on minimax probability machines. *European Journal of Operational Research*, 284(1):273–284, 2020.

[103] K. T. Malladi and T. Sowlati. Sustainability aspects in inventory routing problem: A review of new trends in the literature. *Journal of Cleaner Production*, 197:804–814, 2018.

[104] O. L. Mangasarian. Generalized support vector machines. In *Advances in Large Margin Classifiers*, pages 135–146. MIT Press, 1998.

[105] S. Maqsood and R. Damaševičius. Multiclass skin lesion localization and classification using deep learning based features fusion and selection framework for smart healthcare. *Neural Networks*, 160:238–258, 2023.

[106] E. Marcelli and 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, and R. Umeton, editors, *Machine Learning, Optimization, and Data Science*, pages 1–11, Cham, 2020. Springer International Publishing.

[107] I. Markov, M. Bierlaire, J.-F. Cordeau, Y. Maknoon, and S. Varone. Waste collection inventory routing with non-stationary stochastic demands. *Computers and Operations Research*, 113:104798, 2020.

[108] M. A. Masmoudi, L. C. Coelho, and E. Demir. Plug-in hybrid electric refuse vehicle routing problem for waste collection. *Transportation Research Part E: Logistics and Transportation Review*, 166:102875, 2022.

[109] M. Mes, M. Schutten, and A. P. Rivera. Inventory routing for dynamic waste collection. *Waste Management*, 34(9):1564–1576, 2014.

[110] C. Mi, J. Wang, W. Mi, Y. Huang, Z. Zhang, Y. Yang, J. Jiang, and P. Octavian. Research on regional clustering and two-stage svm method for container truck recognition. *Discrete and Continuous Dynamical Systems - S*, 12(4-5):1117–1133, 2019.

[111] N. H. Moin and S. Salhi. Inventory routing problems: a logistical overview. *Journal of the Operational Research Society*, 58(9):1185–1194, 2007.

[112] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 9.1*, 2019. http://docs.mosek.com/9.1/toolbox/index.html.

[113] P. C. Nolz, N. Absi, and D. Feillet. A stochastic inventory routing problem for infectious medical waste collection. *Networks*, 63(1):82–95, 2014.

[114] T. Nuortio, J. Kytöjoki, H. Niska, and O. Bräysy. Improved route planning and scheduling of waste collection and transport. *Expert Systems with Applications*, 30:223–32, 2006.

[115] Open Data - Government of Canada, 2023. http://open.canada.ca/en/open-data, Accessed on 05.03.2023.

[116] X. Peng. Tpmsvm: A novel twin parametric-margin support vector machine for pattern recognition. *Pattern Recognition*, 44(10):2678–2692, 2011.

[117] X. Peng, L. Kong, and D. Chen. Improvements on twin parametric-margin support vector machine. *Neurocomputing*, 151:857–863, 2015.

[118] X. Peng, Y. Wang, and D. Xu. Structural twin parametric-margin support vector machine for binary classification. *Knowledge-Based Systems*, 49:63–72, 2013.

[119] X. Peng and D. Xu. Robust minimum class variance twin support vector machine classifier. *Neural Computing and Applications*, 22:999–1011, 2013.

[120] Y. Peng, G. Kou, G. Wang, and Y. Shi. Famcdm: A fusion approach of mcdm methods to rank multiclass classification algorithms. *Omega*, 39(6):677–689, 2011.

[121] G. C. Pflug and A. Pichler. From empirical observations to tree models for stochastic optimization: convergence properties. *SIAM Journal on Optimization*, 26:1715–1740, 2016.

[122] V. Piccialli and M. Sciandrone. Nonlinear optimization and support vector machines. *4OR - A Quarterly Journal of Operations Research*, 16:111–149, 2018.

[123] J. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin dags for multiclass classification. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 547–553. MIT Press, 1999.

[124] Z. Qi, Y. Tian, and Y. Shi. Robust twin support vector machine for pattern classification. *Pattern Recognition*, 46(1):305–316, 2013.

[125] R. Raeesi, N. Sahebjamnia, and S. A. Mansouri. The synergistic effect of operational research and big data analytics in greening container terminal operations: A review and future directions. *European Journal of Operational Research*, 310(3):943–973, 2023.

[126] H. Rahimian and S. Mehrotra. Frameworks and results in distributionally robust optimization. *Open Journal of Mathematical Optimization*, 3:1–85, 2022.

[127] T. R. P. Ramos, C. S. de Morais, and A. P. Barbosa-Póvoa. The smart waste collection routing problem: Alternative operational management approaches. *Expert Systems With Applications*, 103:146–158, 2018.

[128] W. Rudin. *Real and complex analysis*. McGraw-Hill, 1987.

[129] A. Sahleh and M. Salahi. Improved robust nonparallel support vector machines. *International Journal of Data Science and Analytics*, 2022.

[130] B. Schölkopf, A. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, 2000.

[131] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, regularization, optimization, and beyond*. MIT press, 2001.

[132] Y. Shao, Z. Wang, W.-J. Chen, and N. Deng. Least squares twin parametric-margin support vector machine for classification. *Applied Intelligence*, 39(3):451 – 464, 2013.

[133] Y.-H. Shao, W.-J. Chen, W.-B. Huang, Z.-M. Yang, and N.-Y. Deng. The best separating decision tree twin support vector machine for multi-class classification. *Procedia Computer Science*, 17:1032–1038, 2013.

[134] A. Shapiro, D. Dentcheva, and A. Ruszczynski. *Lectures on Stochastic Programming: Modeling and Theory, Third Edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2021.

[135] Q. Shen, F. Chu, and H. Chen. A lagrangian relaxation approach for a multi-mode inventory routing problem with transshipment in crude oil transportation. *Computers & Chemical Engineering*, 35(10):2113–2123, 2011.

[136] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC Press, London/Boca Raton, 1998.

[137] M. Singla, D. Ghosh, and K. K. Shukla. A survey of robust optimization based machine learning with special reference to support vector machines. *International Journal of Machine Learning and Cybernetics*, 11:1359–1385, 2020.

[138] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[139] O. Solyal, J.-F. Cordeau, and G. Laporte. Robust inventory routing under demand uncertainty. *Transportation Science*, 46(3):327–340, 2012.

[140] M. Soysal, J. M. Bloemhof-Ruwaard, R. Haijema, and J. G. van der Vorst. Modeling a green inventory routing problem for perishable products with horizontal collaboration. *Computers & Operations Research*, 89:168–182, 2018.

[141] A. Spinelli, F. Maggioni, T. R. P. Ramos, A. P. Barbosa-Póvoa, and D. Vigo. A rolling horizon heuristic approach for a multi-stage stochastic waste collection problem. *Under review in European Journal of Operational Research*, 2024. http://arxiv.org/abs/2405.14499.

[142] M. Szelag and R. Słowiński. Explaining and predicting customer churn by monotonic rules induced from ordinal data. *European Journal of Operational Research*, in press, 2023.

[143] M. Tanveer, T. Rajani, R. Rastogi, and Y. Shao. Comprehensive review on twin support vector machines. *Annals of Operations Research*, pages 1–46, 2022.

[144] F. E. Tay and L. Cao. Application of support vector machines in financial time series forecasting. *Omega*, 29(4):309–317, 2001.

[145] P. Toth and D. Vigo. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, 2002.

[146] T. B. Trafalis and S. A. Alwazzi. Support vector machine classification with noisy data: a second order cone programming approach. *International Journal of General Systems*, 39:757–781, 2010.

[147] T. B. Trafalis and R. C. Gilbert. Robust classification and regression using support vector machines. *European Journal of Operational Research*, 173:893–909, 2006.

[148] United Nations. Transforming our world: the 2030 agenda for sustainable development, 2015. http://wedocs.unep.org/20.500.11822/9814.

[149] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag, 1995.

[150] V. N. Vapnik and A. Y. Chervonenkis. *Theory of Pattern Recognition*. Nauka, Moscow, 1974.

[151] H. Wang, Y. Xu, and Z. Zhou. Twin-parametric margin support vector machine with truncated pinball loss. *Neural Computing and Applications*, 33(8):3781–3798, 2021.

[152] H. Wang, B. Zheng, S. W. Yoon, and H. S. Ko. A support vector machine-based ensemble algorithm for breast cancer diagnosis. *European Journal of Operational Research*, 267(2):687–699, 2018.

[153] X. Wang, N. Fan, and P. M. Pardalos. Robust chance-constrained support vector machines with second-order moment information. *Annals of Operations Research*, 263:45–68, 2018.

[154] X. Wang and P. M. Pardalos. A survey of support vector machines with uncertainties. *Annals of Data Science*, 1:293–309, 2014.

[155] Z. Wang, Y.-H. Shao, and T.-R. Wu. A ga-based model selection for smooth twin parametric-margin support vector machine. *Pattern Recognition*, 46(8):2267–2277, 2013.

[156] Z. Wei, J.-K. Hao, J. Ren, and F. Glover. Responsive strategic oscillation for solving the disjunctively constrained knapsack problem. *European Journal of Operational Research*, 309(3):993–1009, 2023.

[157] J. Xie, K. S. Hone, W. Xie, X. Gao, Y. Shi, and X. Liu. Extending twin support vector machine classifier for multi-category classification problems. *Intelligent Data Analysis*, 17:649–664, 2013.

[158] H. Xu, C. Caramanis, and S. Mannor. Robustness and regularization of support vector machines. *Journal of Machine Learning Research*, 10:1485–1510, 2009.

[159] Y. Xu, R. Guo, and L. Wang. A twin multi-class classification support vector machine. *Cognitive Computation*, 5(4):580–588, 2013.

[160] Y. Xu, Z. Yang, and X. Pan. A novel twin support-vector machine with pinball loss. *IEEE Transactions on Neural Networks and Learning Systems*, 28(2):359–370, 2017.

[161] Y. Yajima. Linear programming approaches for multicategory support vector machines. *European Journal of Operational Research*, 162(2):514–531, 2005.

[162] Z. Yang, Y. Shao, and X.-S. Zhang. Multiple birth support vector machine for multi-class classification. *Neural Computing and Applications*, 22:153–161, 2013.

[163] X. Yao, J. Crook, and G. Andreeva. Enhancing two-stage modelling methodology for loss given default with support vector machines. *European Journal of Operational Research*, 263(2):679–689, 2017.

[164] T. Yu and H. Zhu. Hyper-parameter optimization: A review of algorithms and applications, 2020.

[165] P. Zhong and M. Fukushima. Second-order cone programming formulations for robust multiclass classification. *Neural Computation*, 19:258–282, 2007.