Pontifícia Universidade Católica
do Rio de Janeiro

Tomás Frederico Maciel Gutierrez

PolieDRO: a novel analytics framework with non-parametric data-driven regularization

Tese de Doutorado

Thesis presented to the Programa de Pós–graduação em Engenharia de Produção, do Departamento de Engenharia Industrial da PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia de Produção.

Advisor : Prof. Davi Michel Valladão
Co-advisor: Prof. Bernardo Kulnig Pagnoncelli

Rio de Janeiro
September 2024

Pontifícia Universidade Católica
do Rio de Janeiro

Tomás Frederico Maciel Gutierrez

PolieDRO: a novel analytics framework with non-parametric data-driven regularization

Thesis presented to the Programa de Pós–graduação em Engenharia de Produção da PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia de Produção. Approved by the Examination Committee:

Prof. Davi Michel Valladão
Advisor
Departamento de Engenharia Industrial – PUC-Rio

Prof. Bernardo Kulnig Pagnoncelli
SKEMA Business School

Prof. Bruno Fânzeres dos Santos
Departamento de Engenharia Industrial – PUC-Rio

Prof. Tito Homem-de-Mello
Universidad Adolfo Ibáñez

Prof. Alexandre Street de Aguiar
Departamento de Engenharia Elétrica – PUC–Rio

Prof. Hamed Rahimian
Clemson University

Dr. Joaquim Masset Lacombe Dias Garcia
PSR Power Systems Research

Rio de Janeiro, September 25th, 2024

Tomás Frederico Maciel Gutierrez

Majored in Industrial Engineering and in Mathematics at the Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio). Holds a Master of Science in Industrial Engineering with an emphasis in Financial Engineering from the same institution. This thesis is part of his PhD in Industrial Engineering with an emphasis in Operations Research at PUC-Rio.

Bibliographic data

Maciel Gutierrez, Tomás Frederico

PolieDRO: a novel analytics framework with non-parametric data-driven regularization / Tomás Frederico Maciel Gutierrez; advisor: Davi Michel Valladão; co-advisor: Bernardo Kulnig Pagnoncelli. – 2024.

90 f: il. color. ; 30 cm

Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Industrial, 2024.

Inclui bibliografia

1. Engenharia Industrial – Teses. 2. Otimização Robusta a Distribuições. 3. Aprendizado de Máquina. 4. Otimização de Portfólio. 5. Análise Preditiva. 6. Análise Prescritiva. I. Valladão, Davi M.. II. Pagnoncelli, Bernardo K.. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Industrial. IV. Título.

CDD: 620.11

To my family

Acknowledgments

Acknowledgments will never fully reflect all the gratitude I would like to express. Therefore, this should be understood as an incomplete and imperfect attempt.

First of all, I thank my advisor, Davi Valladão. Over the years, with great mastery, he guided me along this path and constantly reminded me that I could take one step further. I also thank my co-advisor, Bernardo Pagnoncelli, for always offering an alternative yet rigorous perspective. I consider it a great privilege in my life to have had the opportunity to be mentored by both of them.

I also extend my gratitude to LAMPS for the environment it provided over the years, and I take this opportunity to thank all the individuals who passed through there. At the risk of being too general, I have deep gratitude for all the interactions I was fortunate to have in the lab. I feel this is a place that will be special for a long time to come.

I also thank Lamps Co, and once again I use the consolidation of the institution's name as a means of expressing gratitude to all its current and former members. I ask for permission, however, for particular acknowledgments to Professor Alexandre Street, Carol Freire, and Érica Telles.

I also thank Professor Bruno Fânzeres for his teachings over the years, as well as for his academic and personal partnership.

I am grateful to all the other members of the committee for their valuable comments on my work.

I thank the funding agencies, CAPES, CNPq, FAPERJ and VRAC – PUC-Rio, for their encouragement and support throughout my research.

To my parents, Miguel and Nazareth, my first and eternal mentors in this life. I thank them for their care, their love, and for awakening in me a curiosity that I hope will never cease. This work is nothing more than another reflection of the lessons they have given and continue to give me. Hope you feel proud.

To my brothers, Bruno and André, my great companions in life. Here's to many years of achievements to come, which will always be shared.

To my beautiful wife, Juliana, for her unconditional support throughout this journey. Your love, patience, joy, and admiration gave me strength I would not have had on my own. Hope you like it.

To my lifelong friends, each fundamental in their own way for this achievement.

Once again drawing on the collective, I thank all my family. This includes not only those already mentioned but also my aunts, uncles, cousins, grandfathers, and grandmothers. The greatest gift of my life was growing up and living with you all by my side.
Finally, I thank God.
This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Maciel Gutierrez, Tomás Frederico; Valladão, Davi M. (Advisor); Pagnoncelli, Bernardo K. (Co-Advisor). **PolieDRO: a novel analytics framework with non-parametric data-driven regularization**. Rio de Janeiro, 2024. 90p. Tese de Doutorado – Departamento de Engenharia Industrial, Pontifícia Universidade Católica do Rio de Janeiro.

PolieDRO is a novel analytics framework with applications to both predictive and prescriptive realms. It harnesses the power and flexibility of Data-Driven Distributionally Robust Optimization (DRO) to circumvent the need for regularization hyperparameters, while extracting structure from the underlying data.

In the field of predictive modeling, recent literature shows that traditional machine learning methods such as SVM and (square-root) LASSO can be written as Wasserstein-based DRO problems. Inspired by those results we propose a hyperparameter-free ambiguity set that explores the polyhedral structure of data-driven convex hulls, generating computationally tractable regression and classification methods for any convex loss function. Numerical results based on 100 real-world databases and an extensive experiment with synthetically generated data show that our methods consistently outperform their traditional counterparts.

In the prescriptive realm, we develop a portfolio optimization model that employs the DRO approach simultaneously at the risk and return levels. Applying this model to real financial data spanning several decades, we achieve consistent superior performance compared to a benchmark.

Keywords

Distributionally Robust Optimization; Machine Learning; Portfolio Optimization; Predictive Analytics; Prescriptive Analytic.

Resumo

Maciel Gutierrez, Tomás Frederico; Valladão, Davi M.; Pagnoncelli, Bernardo K.. **PolieDRO: um novo framework analítico com regularização não paramétrica orientada por dados**. Rio de Janeiro, 2024. 90p. Tese de Doutorado – Departamento de Engenharia Industrial, Pontifícia Universidade Católica do Rio de Janeiro.

PolieDRO é um novo framework com aplicações tanto no âmbito preditivo quanto prescritivo. Ela aproveita o poder e a flexibilidade da Otimização Robusta a Distribuições (DRO) orientada por dados para evitar a necessidade de hiperparâmetros de regularização, ao mesmo tempo em que extrai estrutura dos dados subjacentes.

No âmbito preditivo, a literatura recente mostra que métodos tradicionais de aprendizado de máquina, como SVM e (square-root) LASSO, podem ser formulados como problemas de DRO baseados em métricas de distância de Wasserstein. Inspirados por esses resultados, propomos um conjunto de ambiguidades sem hiperparâmetros que explora a estrutura poliédrica de invólucros convexos orientados por dados, gerando métodos de regressão e classificação computacionalmente viáveis para qualquer função de perda convexa. Resultados numéricos baseados em 100 bancos de dados do mundo real e um extenso experimento com dados gerados sinteticamente mostram que nossos métodos superam consistentemente seus equivalentes tradicionais.

No âmbito prescritivo, desenvolvemos um modelo de otimização de portfólio no qual a abordagem DRO é empregada simultaneamente nos níveis de risco e retorno. Aplicando este modelo a dados financeiros reais que abrangem várias décadas, alcançamos um desempenho consistentemente superior em comparação com um benchmark tradicional.

Palavras-chave

Otimização Robusta a Distribuições; Aprendizado de Máquina; Otimização de Portfólio; Análise Preditiva; Análise Prescritiva.

Table of contents

List of figures

List of tables

List of algorithms

Alis Grave Nil

Com asas nada é pesado, *PUC-Rio.*

1
Introduction

In this thesis, we introduce a novel analytical framework that utilizes Data-Driven Distributionally Robust Optimization (DRO) for both predictive and prescriptive tasks. Our approach centers on our newly proposed data-driven ambiguity set, characterized by properties that enable a finite reformulation of distributionally robust optimization problems that were previously intractable.

We begin by examining the traditional formulation of a DRO problem, such as the one shown in Equation 1-1:

$$\min_{\beta \in \mathcal{B}} \left\{ \sup_{P \in \mathcal{P}} \mathbb{E}_P[h(W;\beta)] \right\}, \tag{1-1}$$

where $h(W, \beta)$ is a cost function, the decision variable $\beta \in \mathcal{B} \subset \mathbb{R}^d$ represents the vector of model coefficients and W is a random vector with probability distribution P. The set $\mathcal{B}$ encompasses all feasible constraints on β, while $\mathcal{P}$ denotes the ambiguity set, or Distributional Uncertainty Set (DUS), which encapsulates the potential distributions considered in the optimization process.

Building on this foundation, we develop a data-driven method to construct an ambiguity set that allows for a computationally tractable reformulation of the DRO problem. This reformulation underpins the proposed framework, termed PolieDRO, which is applicable to both predictive and prescriptive tasks under certain conditions. In Chapter 2, we explore predictive applications within the field of Machine Learning. Through the PolieDRO framework, we introduce novel methods for commonly used loss functions in both classification and regression tasks. These new models obviate the need for regularization hyperparameters while achieving performance that is competitive with traditional approaches. In Chapter 3, we turn our attention to prescriptive applications, specifically presenting a portfolio optimization model that employs a distributionally robust approach to managing both risk and return. Our results highlight the superior performance of this model when applied to real-world financial data, compared to conventional benchmarks.

This thesis is organized into two main chapters, each designed to stand alone, allowing for independent reading and understanding.

2
Predictive PolieDRO – Machine Learning

Predictive analytics comprises statistical/machine learning methods that leverage the nowadays data-rich environment to perform accurately the tasks of classification or regression. Most learning methods contain hyperparameters (e.g., regularization coefficients), and their calibration is usually performed in an ad-hoc manner. Such calibration is usually computationally intensive and can become prohibitive to the point of hampering the use of some algorithms by practitioners (SIVAPRASAD et al., 2020). The overall training of machine learning models can be energy intensive to the point of generating environmental concerns (ANTHONY; KANDING; SELVAN, 2020; HAO, 2019; LACOSTE et al., 2019), and the hyperparameter calibration only amplifies the problem.

This work aims to devise a high-performance alternative method that bypasses the need for such hyperparameter calibration. One of our starting points is the recent work of (BLANCHET; KANG; MURTHY, 2019). The authors prove the equivalence of many popular machine learning (ML) methods (e.g., SVM, square-root LASSO) to the Wasserstein-based DRO model

$$\min_{\beta\in\mathcal{B}} \left\{ \sup_{P:D(P,\hat{P}_N)\leq\lambda} \mathbb{E}_P[h(W;\beta)] \right\}, \tag{2-1}$$

where $h(W,\beta)$ is the loss function, the decision variable $\beta \in \mathcal{B} \subset \mathbb{R}^d$ is the vector of model coefficients and $W = (X,Y)$ is a random vector, with probability distribution P, comprising the dependent variable Y and its covariates X. In the inner problem we have an ambiguity set or distributional uncertainty set (DUS) in which function $D(\cdot)$ is some distance function between P and the nominal distribution $\hat{P}_N$ constructed based on the available data, and the nonnegative scalar λ is a radius.

In (BLANCHET; KANG; MURTHY, 2019) the authors show that the radius λ can be interpreted as a regularization coefficient for several ML methods. For instance, if $h(W,\beta)$ is the hinge loss, the formulation (2-1) is equivalent to the SVM classification method. If $h(W,\beta)$ is the log-exponential loss, the formulation (2-1) renders the regularized logistic regression. For regression methods, the square-root Lasso (BELLONI; CHERNOZHUKOV; WANG, 2011) can be represented by defining $h(W,\beta)$ as the root mean square error of a linear model.

In this context, we argue that the DRO is a suitable framework for

ML, whereby the aforementioned classical methods are particular cases. This abstract view renders opportunities to develop new ML models by simply changing the ambiguity set in (2-1) for a given loss function $h(W;\beta)$. The work presents a potential bridge between two fields of research, namely DRO and ML, which pave the way for the investigation of new parallels and connections.

That said, our objective is to propose a novel predictive analytics framework by reformulating problem (2-1) with an ambiguity set that does not depend on the user-defined regularization coefficient λ. While the formulation proposed in (BLANCHET; KANG; MURTHY, 2019) yields known ML methods, our goal is to extend the connection between the two worlds in order to construct new methods. To this end, we construct an intuitive and efficient way to forge a (hyperparameter-free) ambiguity set, which leads to a computationally tractable DRO for the case of convex loss functions. We obtain a tractable formulation by considering the representation of the ambiguity set proposed by (WIESEMANN; KUHN; SIM, 2014) combined with the polyhedral structure of data-driven uncertainty sets based on (FERNANDES et al., 2016).

In their work, the authors in (WIESEMANN; KUHN; SIM, 2014) propose a framework that generalizes several DRO approaches from the literature, such as constraints on the mean, variance, coefficient of variation, and higher-order moment information, among others. This framework ensures the tractability of the DRO problems it encompasses by defining a set of regularity conditions on the distributional uncertainty set. In particular, we assume the regularity condition described as consecutively nested convex hulls (see Figure 2.1), each one associated with a confidence interval of its coverage probability. A particular case of the ambiguity set proposed by (WIESEMANN; KUHN; SIM, 2014) can be written as

$$\mathcal{P} = \left\{ P \in \mathcal{M}_+(\mathbb{R}^d) \;\middle|\; P(W \in \mathcal{C}_i) \in [\underline{p_i}, \overline{p_i}], \forall i \in \mathcal{F} \right\}, \tag{2-2}$$

where $\mathcal{F} = \{0, 1, 2, \ldots, \mathcal{I}\}$ and the nested sets $\mathcal{C}_\mathcal{I} \subset \mathcal{C}_{\mathcal{I}-1} \subset \ldots \subset \mathcal{C}_1 \subset \mathcal{C}_0$ can be interpreted as contour lines of the joint probability P[1]. This formulation is powerful and flexible, but the literature lacks an efficient and intuitive way to construct these sets in a data-driven manner. In Figure 2.1 we anticipate the result of a polyhedral shape for the ambiguity set (in a two-dimensional space), where the actual observations are used as vertices for the nested sets.

Based on this framework, we propose an entirely data-driven method to specify the distributional uncertainty set, based on the observations of a random vector X, with associated dependent variable Y. Contrary to most of the

[1]Note that we enforce $\underline{p}_0 = \overline{p}_0 = 1$ to ensure that P is a probability measure.

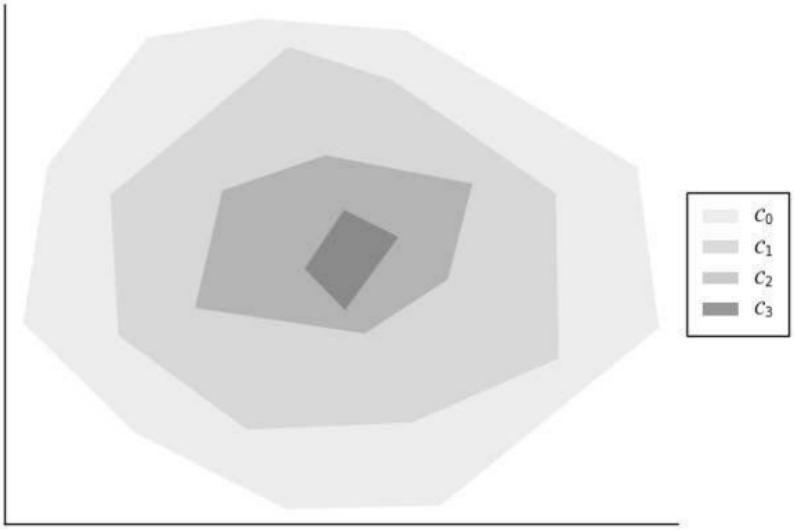


Figure 2.1: A polyhedral ambiguity set examples.

literature so far (see (ESFAHANI; KUHN, 2018)), we do not use the available data only to define the center of the distributional uncertainty set. We also define the *shape* of the nested convex sets of $\mathcal{P}$ and the associated probabilities, in a non-parametric fashion with ideas taken from (FERNANDES et al., 2016). In short, we first obtain the polyhedral convex hull of all data points to obtain the outer set, then we exclude extreme points, and repeat the process recursively to obtain the inner sets, as illustrated in Figure (2.1) and explained in detail in Section 2.2.2. With the construction of data-driven convex hulls along with coverage probability estimates, the PolieDRO problem is fully specified as a single-level convex optimization problem and can be efficiently solved by off-the-shelf convex solvers. In summary, our main contributions are:

- A novel predictive analytics framework based on a data-driven DRO formulation that does not rely on hyperparameter calibration;
- A computationally tractable DRO reformulation for any problem with a convex loss function;
- A new and iterative procedure to construct the data-driven convex hulls that define the ambiguity set along with coverage probability estimates;
- Extensive numerical tests to evidence the competitive performance of the framework for classification and regression problems.

2.1 Relevant literature

The DRO framework can be interpreted as a generalization of the two main paradigms of optimization under uncertainty. On the one hand, Robust

Optimization (RO) attempts to replace parameter uncertainty by considering an uncertainty set for its values. On the other hand, Stochastic Optimization (SO) deals with parameter uncertainty by considering a probability distribution of its values. Both methodologies are well-established and proven to have excellent capabilities in solving real-world problems arising from many areas, such as transportation, finance, and energy, among others (see (BEN-TAL; GHAOUI; NEMIROVSKI, 2009a), (BERTSIMAS; BROWN; CARAMANIS, 2011), (SHAPIRO; DENTCHEVA; RUSZCZYNSKI, 2021)).

Nonetheless, each approach has some drawbacks. RO, for instance, may lead to an under-specification of uncertainty as it does not consider possibly available distributional knowledge, which can result in overly conservative decisions. SO typically assumes full distributional knowledge; if the assumed distribution is far from the true one the model can yield sub-optimal solutions.

The increased amount of noisy and incomplete data, together with the need to consider both risk and ambiguity simultaneously, laid the ground for the growth of a third paradigm: DRO, see (GOH; SIM, 2010; WIESEMANN; KUHN; SIM, 2014; PARYS; ESFAHANI; KUHN, 2021) A typical formulation of a DRO problem is given by 2-1, where the decision β must be taken to minimize some function h, considering the random variable W which follows some distribution P. Such distribution itself is uncertain and belongs to an ambiguity set, i.e., the feasible set of the inner problem.

The DRO framework allows an extra degree of flexibility, as it can replicate RO and SO formulations by the proper ambiguity set specification. If we characterize $\mathcal{P}$ as a single possible distribution, we retrieve a classical SO formulation. Instead, if we allow all possible distributions, with no imposed structure based on the current observations, we reach the RO version. Thus, the way we incorporate distributional knowledge into the ambiguity set is crucial to balance how the decision-maker deals with the parameters' uncertainty and the distribution's ambiguity.

In (WIESEMANN; KUHN; SIM, 2014) the authors study a generic class of DRO problems that allow a tractable reformulation that offers striking modeling power. Such a class of problems requires a set of regularity conditions that ensure tractability while retaining expressiveness capabilities. As a starting point, the optimization problem must be tractable if stripped of all distributionally robust constraints. In addition, the ambiguity set can be represented in a standard form following two regularity conditions regarding its shape. Finally, the function h must be convex in the decision variable β and the random vector W.

In (ESFAHANI; KUHN, 2018), the authors apply the Wasserstein metric

to construct an ambiguity set for DRO problems that, under some assumptions, can be reformulated as convex programs and, in some cases, even linear programs. They also dwell on performance guarantees that validate such formulations, with a particular application in mean-risk portfolio optimization, as well as uncertainty quantification.

Such reformulation achieved empirical success for various applications. In (KUHN et al., 2019) the authors present a variety of problems under which this framework is applicable, such as classification and regression, under an ML setting. It is important to remark that some classes of problems are more challenging than others, and may require decomposition schemes as efficient solution methodologies, see (GAMBOA et al., 2021). Part of this success is associated with its regularization properties, first explored in particular settings, for example in (SHAFIEEZADEH-ABADEH; ESFAHANI; KUHN, 2015), (CHEN; PASCHALIDIS, 2018) and (SHAFIEEZADEH-ABADEH; KUHN; ESFAHANI, 2019), and later unified under the concept termed as variation of loss by (GAO; CHEN; KLEYWEGT, 2020).

In this work, we expand the literature at the intersection of DRO and ML by proposing a new framework. Based on the structured albeit generic class of DRO problems studied in (WIESEMANN; KUHN; SIM, 2014) and the underlying connections with the ML literature presented in (BLANCHET; KANG; MURTHY, 2019), we develop a new methodology to construct data-driven ambiguity sets. Such ambiguity sets comply with the constraints needed for its tractability and give birth to a new framework for predictive analytics that results in tractable variations of any ML method driven by a convex loss function.

2.2 Proposed PolieDRO framework

In this section, we propose a novel Data-Driven DRO formulation as a tractable convex optimization problem, assuming a specific structure in the ambiguity set, namely convexity and polyhedral shape. Starting from the general formulation proposed by (WIESEMANN; KUHN; SIM, 2014) and briefly explained in Section 2.2.1, we present an efficient method for extracting such structure from observable data, including how to calculate the coverage probabilities associated with each portion of the ambiguity set obtained in Section 2.2.2. Such structure aims to reflect the shape of the empirical probability density of the observed random variable, with enough flexibility to consider variations of it. Although entirely data-driven, such a procedure forms an ambiguity set with characteristics that satisfy the requirements

needed for the reformulation proposed in Section 2.2.3. Standard solvers can efficiently solve such reformulation for any convex loss function. The choice of the accompanying loss function—such as hinge loss, and least squares, among others—enables the application on both regression and classification tasks. The proposed DRO with a purely data-driven ambiguity set bypasses the need for hyperparameter calibration on regression and classification tasks.

2.2.1 Theoretical background

For completeness, in this section, we adapt the results of (WIESEMANN; KUHN; SIM, 2014) to our context. The following section uses these adapted results to develop the PolieDRO framework.

Our starting point is the following formulation:

$$\begin{aligned} \min_{\beta\in\mathcal{B}} \quad & \max_{P} \quad \mathbb{E}_P[h(W,\beta)] \\ & s.t. \quad P \in \Big\{ P \in \mathcal{M}_+(\mathbb{R}^m) \Big| P(W \in \mathcal{C}_i) \in [\underline{p_i}, \overline{p_i}], \forall i \in \mathcal{F} \Big\}, \end{aligned} \tag{2-3}$$

which is a particular case of the DRO proposed in (WIESEMANN; KUHN; SIM, 2014). Here, $h(W,\beta)$ is the loss function, $\beta \in \mathcal{B} \subset \mathbb{R}^d$ is the decision variable, $W \in \mathbb{R}^m$ is a random variable with probability distribution P and $\mathcal{F}$ is a set of indices $\{1, 2, \ldots, \mathcal{J}\}$. Albeit unknown, P is required to satisfy the nested ambiguity set constraints on the convex sets whereas $\mathcal{C}_\mathcal{I} \subset \mathcal{C}_{\mathcal{I}-1} \subset \ldots \subset \mathcal{C}_1 \subset \mathcal{C}_0$, with a bounded $\mathcal{C}_0$ with probability one, i.e., $\underline{p}_0 = \overline{p}_0 = 1$. We can then write the inner maximization problem (2-3) as

$$\begin{aligned} \max_{P\in\mathcal{P}} \quad & \int_{\mathcal{C}_0} h(w;\beta) dP \\ \text{s.t.} \quad & \int_{\mathcal{C}_0} \mathbb{I}_{\{w\in\mathcal{C}_i\}} dP \geq \underline{p_i}, \qquad \forall i \in \mathcal{F} \quad : \lambda \\ & \int_{\mathcal{C}_0} \mathbb{I}_{\{w\in\mathcal{C}_i\}} dP \leq \overline{p_i}, \qquad \forall i \in \mathcal{F} \quad : \kappa, \end{aligned} \tag{2-4}$$

where λ and κ are the dual variables associated with the respective constraints.

Naturally, this is not a solvable problem in its current form. Following (WIESEMANN; KUHN; SIM, 2014), we write the problem's Lagrangian formulation:

$$\begin{aligned} \mathcal{L}(P,\kappa,\lambda) = \quad & \int_{\mathcal{C}_0} h(w;\beta) dP \\ & - \sum_{i\in\mathcal{F}} \Big[\Big(\underline{p_i} - \int_{\mathcal{C}_0} \mathbb{I}_{\{w\in\mathcal{C}_i\}} dP \Big) \lambda_i \Big] \\ & - \sum_{i\in\mathcal{F}} \Big[\Big(\int_{\mathcal{C}_0} \mathbb{I}_{\{w\in\mathcal{C}_i\}} dP - \overline{p_i} \Big) \kappa_i \Big] \end{aligned} \tag{2-5}$$

Reorganizing the terms, we can then write:

$$\begin{aligned}\mathcal{L}(P,\kappa,\lambda) = \quad & \int_{\mathcal{C}_0}\Big[h(w;\beta) - \sum_{i\in\mathcal{F}}\mathbb{I}_{\{w\in\mathcal{C}_i\}}(\kappa_i-\lambda_i)\Big]dP \\ & - \sum_{i\in\mathcal{F}}(\lambda_i\underline{p_i} - \kappa_i\overline{p_i})\end{aligned} \tag{2-6}$$

The Lagrange dual function $g(\lambda,\kappa)$ can then be written as:

$$\begin{aligned} g(\lambda,\kappa) &= \sup_{P\in\mathcal{P}}\mathcal{L}(P,\kappa,\lambda) \\ &= \sup_{P\in\mathcal{P}}\left\{\int_{\mathcal{C}_0}\Big[h(W;\beta) - \sum_{i\in\mathcal{F}}\mathbb{I}_{\{w\in\mathcal{C}_i\}}(\kappa_i-\lambda_i)\Big]dP\right\} \\ &\qquad - \sum_{i\in\mathcal{F}}(\lambda_i\underline{p_i} - \kappa_i\overline{p_i})\end{aligned} \tag{2-7}$$

Notice that the term within parenthesis can be analyzed as:

$$\begin{aligned}&\sup_{P\in\mathcal{P}}\left\{\int_{\mathcal{C}_0}\Big[h(w;\beta) - \sum_{i\in\mathcal{F}}\mathbb{I}_{\{w\in\mathcal{C}_i\}}(\kappa_i-\lambda_i)\Big]dP\right\} \\ &= \begin{cases}0, & \text{if } h(w;\beta) - \sum_{i\in\mathcal{F}}\mathbb{I}_{\{w\in\mathcal{C}_i\}}(\kappa_i-\lambda_i) \le 0 \\ \infty, & \text{otherwise}\end{cases}\end{aligned} \tag{2-8}$$

Since we are only interested in the finite cost case, the dual problem $\min_{\lambda\ge0,\kappa\ge0} g(\lambda,\kappa)$ is given by:

$$\begin{aligned}\min_{\lambda,\kappa} \quad & \sum_{i\in\mathcal{F}}(\kappa_i\overline{p_i} - \lambda_i\underline{p_i}) \\ \text{s.t.} \quad & h(w;\beta) - \sum_{i\in\mathcal{F}}\mathbb{I}_{\{w\in\mathcal{C}_i\}}(\kappa_i-\lambda_i) \le 0, \quad \forall w\in\mathcal{C}_0 \\ & \lambda_i \ge 0, \qquad\qquad \forall i\in\mathcal{F} \\ & \kappa_i \ge 0, \qquad\qquad \forall i\in\mathcal{F}\end{aligned} \tag{2-9}$$

Although we were able to remove the integrals by writing the dual version of the problem, the problem is still not tractable, as the first constraint implies an infinite amount of points to evaluate.

Proposition 2: Let $\mathcal{R}(w)$ be a constraint valid $\forall w\in\mathcal{C}_0$, and $\bigcup_{i\in\mathcal{F}}\overline{\mathcal{C}}_i$ be a partition of $\mathcal{C}_0$. Then, the following are equivalent:

$$\mathcal{R}(w), \forall w\in\mathcal{C}_0 \iff \mathcal{R}(w), \forall w\in\overline{\mathcal{C}}_i, \forall i\in\mathcal{F} \tag{2-10}$$

Based on Proposition 2, we can rewrite the constraint:

$$h(w;\beta) - \sum_{i\in\mathcal{F}}\mathbb{I}_{\{w\in\mathcal{C}_i\}}(\kappa_i-\lambda_i) \le 0, \forall w\in\mathcal{C}_0 \tag{2-11}$$

as

$$h(w;\beta) - \sum_{i\in\mathcal{F}}\mathbb{I}_{\{w\in\mathcal{C}_i\}}(\kappa_i-\lambda_i) \le 0, \forall w\in\overline{\mathcal{C}}_i, \forall i\in\mathcal{F} \tag{2-12}$$

Proposition 3: Consider the sets $\mathcal{F} = \{0, 1, ..., I\}$, $\{\mathcal{V}_i\}_{i\in\mathcal{F}}$ and $\{\mathcal{C}_i\}_{i\in\mathcal{F}}$ obtained from a procedure as defined in Algorithm 1. In addition, let $w' \in \mathcal{C}_0$ and $\bigcup_{i\in\mathcal{F}} \overline{\mathcal{C}}_i$ a partition of $\mathcal{C}_0$. Therefore, $w' \in \overline{\mathcal{C}}_i$ for some $i \in \mathcal{F}$. In addition, we define the index sets of all supersets (antecedents of $\mathcal{C}_i$) by $\mathcal{A}(i) = \{i\} \cap \{i' \in \mathcal{F} : \mathcal{C}_i \subsetneq \mathcal{C}_{i'}\}$ and the index sets of all subsets (descendants of $\mathcal{C}_i$) as $\mathcal{D}(i) = \mathcal{F} \setminus \mathcal{A}(i)$.

Thus we can write:

1. $w' \in \{\mathcal{C}_j\}_{j\in\mathcal{A}(i)}$, $w' \notin \{\mathcal{C}_j\}_{j\in\mathcal{D}(j)}$ for some $j \in \mathcal{F}$
2. $\sum_{i\in\mathcal{F}} \mathbb{I}_{\{w'\in\mathcal{C}_i\}} = \sum_{i'\in\mathcal{A}(i)} 1$

Based on Proposition 3, we can rewrite the constraint:

$$h(w;\beta) - \sum_{i\in\mathcal{F}} \mathbb{I}_{\{w\in\mathcal{C}_i\}}(\kappa_i - \lambda_i) \leq 0, \forall w \in \overline{\mathcal{C}}_i, \forall i \in \mathcal{F} \tag{2-13}$$

as

$$h(w;\beta) - \sum_{i'\in\mathcal{A}(i)} (\kappa_i - \lambda_i) \leq 0, \forall w \in \overline{\mathcal{C}}_i, \forall i \in \mathcal{F} \tag{2-14}$$

Since the constraint above is valid $\forall w \in \overline{\mathcal{C}}_i, \forall i \in \mathcal{F}$, we can write:

$$\min_{w\in\overline{\mathcal{C}}_i}\left\{h(w;\beta)\right\} - \sum_{i'\in\mathcal{A}(i)} (\kappa_i - \lambda_i) \leq 0, \forall i \in \mathcal{F} \tag{2-15}$$

Finally, we rewrite this problem as

$$\begin{aligned} \min_{\lambda,\kappa} \quad & \sum_{i\in\mathcal{F}} (\kappa_i \overline{p_i} - \lambda_i \underline{p_i}) \\ \text{s.t.} \quad & h(w;\beta) - \sum_{l\in\mathcal{A}(i)} (\kappa_l - \lambda_l) \leq 0, \quad w \in \mathcal{C}_i, \forall i \in \mathcal{F} \\ & \lambda_i \geq 0, \qquad\qquad \forall i \in \mathcal{F} \\ & \kappa_i \geq 0, \qquad\qquad \forall i \in \mathcal{F}, \end{aligned} \tag{2-16}$$

In (WIESEMANN; KUHN; SIM, 2014), the authors show that a similar reformulation is a generalization for many approaches from the DRO literature, such as models with constraints on the mean, variance, coefficient of variation, and higher-order moment information, among others. It ensures DRO tractability by requiring a set of regularity conditions on the ambiguity set, mainly the consecutively nested convex hulls, each associated with a confidence interval of its coverage probability. We enhance such reformulation by constructing a purely data-driven polyhedral uncertainty set that does not depend on a user-defined hyperparameter.

2.2.2 Proposed Data-Driven Ambiguity Set

In this section, we propose a procedure to specify an ambiguity set with desired properties, built entirely from a data-driven perspective. While the majority of the literature uses the empirical distribution, that is, the observations from the random data generating process, to define the center of the distributional uncertainty set, our method leverages the data to define the shape of the nested convex sets of $\mathcal{P}$ and the associated probabilities.

Our main idea takes advantage of the well-known N-dimensional Quick Hull Algorithm (BARBER; DOBKIN; HUHDANPAA, 1996), which in turn is a generic dimensional version of the algorithms proposed independently by (GREENFIELD, 1990), (EDDY, 1977), (BYKAT, 1978) and (GREEN; SILVERMAN, 1979).

The Quick Hull Algorithm aims to retrieve the polyhedron convex hull $\mathcal{C} = Conv(A)$ of a given set of data points A, and the associated vertices $Vert(\mathcal{C}) \subseteq A$. Since the vertices are themselves points in the original set, we iteratively apply the algorithm in the resulting subsets, after removing the retrieved vertices, to construct a nested set of convex and polyhedral sets, as specified by the original formulation (2-3).

Before formalizing the proposed procedure, we illustrate the construction process with an example in a 2-dimensional space. We highlight that all steps conducted are easily extended to higher dimensions, as there are no particular properties of lower-dimensional spaces explored. Let $A_0 = \{w_1, w_2, ..., w_N\}$ be the set of all observations of the random variable W, whose distribution is unknown. As an illustrative example, consider $w_i = (w_{i,1}, w_{i,2}) \in \mathbb{R}^2, i = 1, ..., N$, whereby we can visualize a given sample in Figure 2.2.

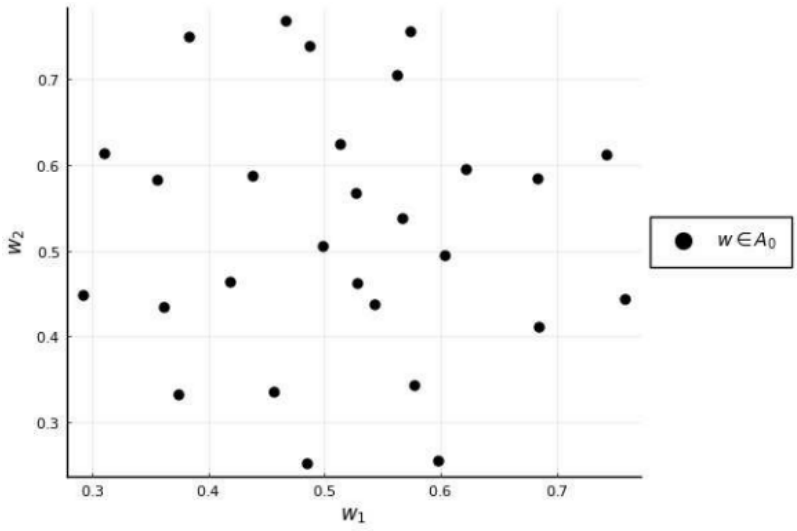


Figure 2.2: Observed sample from random variable W.

A first application of the Quick Hull algorithm gives us the polyhedron convex hull $\mathcal{C}_0 = Conv(A_0)$ and the associated set vertices $Vert(C_0) \subseteq A_0$, which is a subset of all data observations. The result is shown in Figure 2.3. After this first step, we define $A_1 = A_0 \setminus Vert(\mathcal{C}_0)$ as the set of interior points

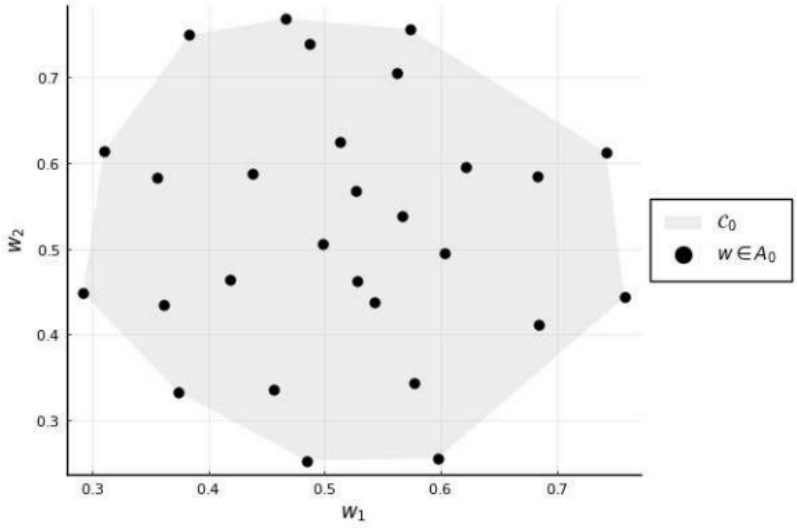


Figure 2.3: $\mathcal{C}_0$, the convex hull of A_0

of $\mathcal{C}_0$ and then apply the same procedure to A_1, obtaining its own convex hull $\mathcal{C}_1$ (see Figure 2.4). Note that we can also state that the set of vertices $Vert(\mathcal{C}_0) = A_0 \setminus A_1$ comprises the set of all observations disregarding its interior points.

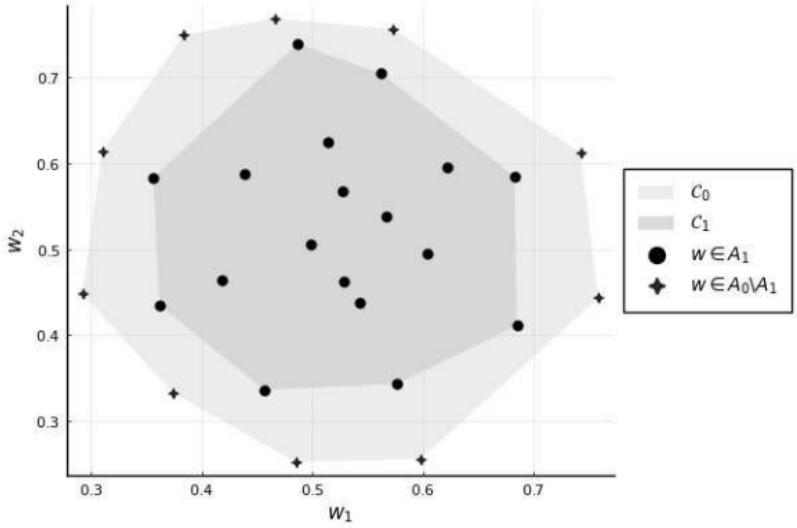


Figure 2.4: $\mathcal{C}_1$, the convex hull of A_1, nested inside $\mathcal{C}_0$

We apply this routine again until there are no remaining points left. We define $A_2 = A_1 \setminus Vert(\mathcal{C}_1)$ as the set of observations which are interior points of $\mathcal{C}_1$. Then we use the Quick Hull algorithm once again, obtaining the convex hull $\mathcal{C}_2$ and the associated vertices $Vert(\mathcal{C}_2)$ (see Figure 2.5). We repeat the

procedure once again, obtaining $\mathcal{C}_3$ and $Vert(\mathcal{C}_3)$ starting from A_3. The final configuration can be seen in Figure 2.6. Notice that this procedure implicitly results in nested polyhedral convex sets, as required to apply our results derived in the previous section.

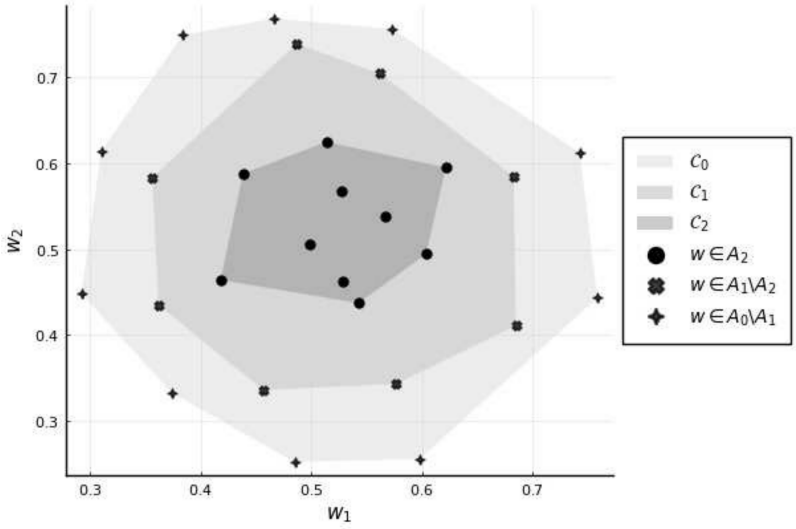


Figure 2.5: $\mathcal{C}_2$, the convex hull of A_2, nested inside $\mathcal{C}_0$ and $\mathcal{C}_1$

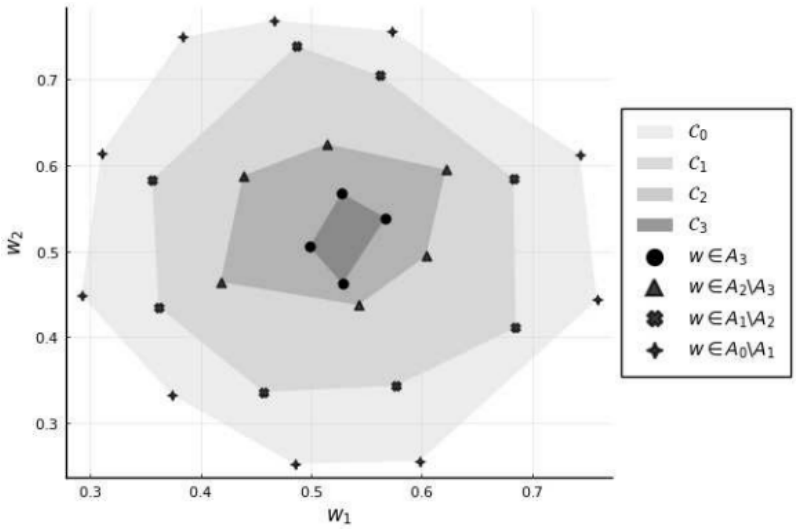


Figure 2.6: $\mathcal{C}_3$, the convex hull of A_3, nested inside $\mathcal{C}_0$, $\mathcal{C}_1$ and $\mathcal{C}_2$

More generally, we describe the construction of the nested convex hulls in Algorithm 1.

Algorithm 1: Nested Convex Hull Sets.

Input: Data sample $\{w_1, w_2, \ldots, w_N\} \in \mathbb{R}^d$, of the random variable W

Initialization:
$A_0 \leftarrow \{w_1, w_2, \ldots, w_N\}$;
Apply the Quick Hull algorithm to A_0;
$\mathcal{C}_0 \leftarrow$ Convex hull obtained from A_0;
$Vert(\mathcal{C}_0) \leftarrow$ Set of vertices of $\mathcal{C}_0$;
$i \leftarrow 0$;
while $|A_i \setminus Vert(\mathcal{C}_i)| > 0$ **do**
    $A_{i+1} \leftarrow A_i \setminus Vert(\mathcal{C}_i)$;
    Apply the Quick Hull algorithm to A_{i+1};
    $\mathcal{C}_{i+1} \leftarrow$ Convex hull obtained from A_{i+1};
    $Vert(\mathcal{C}_{i+1}) \leftarrow$ Set of vertices of $\mathcal{C}_{i+1}$;
    $i \leftarrow i + 1$;
$\mathcal{I} \leftarrow i - 1$;
Return: $\{A_i\}_{i=0}^{\mathcal{I}}$, $\{\mathcal{C}_i\}_{i=0}^{\mathcal{I}}$, $\{Vert(\mathcal{C}_i)\}_{i=0}^{\mathcal{I}}$, $\mathcal{F} \leftarrow \{0, 1, \ldots, \mathcal{I}\}$;

To fully describe the proposed data-driven ambiguity set, we must associate a probability coverage interval to each polyhedral convex set obtained in Algorithm 1. We start from the empirical probability distribution $\hat{\mathbb{P}}$, where $\hat{p}_i$ is the probability that the random variable $W \in \mathcal{C}_i$. In an ideal setting, we would have at our disposal enough observations that would allow us to construct the aforementioned $\{\mathcal{C}_i\}_{i=0}^{\mathcal{I}}$ with a fraction of the total samples and then use the remaining ones to calculate the empirical probabilities of being placed in one of such hulls. However, such a procedure is not data efficient as (i) it implies a choice between the partitions of the sample dedicated to which step and (ii) it needs huge amounts of data, which are usually not available. To circumvent this obstacle, we propose a single-step approximation in which the same data set is used to build the convex hulls and estimate the probabilities $\hat{p}_i$. In section 2.5.1 we provide a controlled experiment to assess the quality of this approximation.

The quantity empirically obtained by evaluating how many observations fall within each convex set $\mathcal{C}_i$, represented by its original points $\mathcal{A}_i$, is given by:

$$\hat{p}_i = \frac{1}{N}\sum_{j=1}^{N} \mathbb{I}_{A_i}(w_j)$$

where $\mathbb{I}_X(x) = 1$ if $x \in X$ and $\mathbb{I}_X(x) = 0$ otherwise. Naturally, the outermost convex hull $\mathcal{C}_0$ covers all observations A_0, hence $\hat{p}_0 = 1$. This associates the original observations with the support of all distributions considered in the ambiguity set.

After obtaining the empirical probabilities for each nested convex hull, we must expand it further to calculate coverage intervals instead of point-wise estimations. As we provide probability intervals, we allow the optimization model to navigate within a space of probability distributions whose shape is defined entirely by the observations of the random variable W to find the optimal solution. To do so, notice that each quantity $\hat{p}_i$ can be approximated by a weighted sum of N Bernoulli random variables with mean p_i. This implies that

$$N\hat{p}_i \sim Binomial(N, p_i)$$

We then make use of the Normal approximation of the Binomial distribution, which results from the Central Limit Theorem in the following relationship (CASELLA; BERGER, 2001):

$$\sqrt{N}\frac{\hat{p}_i - p_i}{\sqrt{p_i(1-p_i)}} \xrightarrow{D} \mathcal{N}\Big(0, 1\Big)$$

Based on this relation, we define an approximate $(1-\alpha)$-confidence interval of coverage probability for each set $\mathcal{C}_i$ as $[\underline{p_i}, \overline{p_i}]$ as a binomial proportion confidence interval, with $\underline{p}_i$ and $\overline{p}_i$ given by:

$$[\underline{p}_i,\, \overline{p}_i] = \left[\, \hat{p}_i - q_{\frac{\alpha}{2}}\sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{N}},\; \hat{p}_i + q_{\frac{\alpha}{2}}\sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{N}} \,\right], \forall i \in \mathcal{F}$$

where $\underline{p}_0 = \overline{p}_0 = 1$, which retains the desired probability measure properties. Using the distributional uncertainty (nested) subsets $\{\mathcal{C}_i\}_{i\in\mathcal{F}}$, and the associated confidence intervals of their coverage probability, $[\underline{p_i}, \overline{p_i}]$, we define a data-driven ambiguity set $\mathcal{P}$ that satisfies the general formulation of Equation 2-2.

2.2.3 Finite reformulation

Given the data-driven convex hulls and the coverage probabilities, we propose a finite convex reformulation of the semi-infinite programming problem (2-16). Our reformulation is computationally tractable and we can harness the power of off-the-shelf solvers for convex or even linear optimization problems, depending on the choice of the loss function $h(W;\beta)$.

Proposition 1: Suppose we have set of vertices $Vert(\mathcal{C}_i)$ associated with convex hulls $\mathcal{C}_i, \forall i \in \mathcal{F}$. In addition, assume $h(w;\beta)$ is a convex function and $\mathcal{B}$ is a convex set. Under these conditions, we can write (2-3) as the finite convex problem

$$\begin{aligned}
\min_{\beta,\lambda,\kappa} \quad & \sum_{i\in\mathcal{F}} (\kappa_i \overline{p_i} - \lambda_i \underline{p_i}) \\
\text{s.t.} \quad & h(w_j;\beta) - \sum_{l\in\mathcal{A}(i)} (\kappa_l - \lambda_l) \leq 0, \quad \forall j \in \mathcal{V}_i, \forall i \in \mathcal{F} \\
& \lambda_i \geq 0, \qquad\qquad \forall i \in \mathcal{F} \\
& \kappa_i \geq 0, \qquad\qquad \forall i \in \mathcal{F} \\
& \beta \in \mathcal{B},
\end{aligned} \tag{2-17}$$

where $\mathcal{V}_i \subseteq \{1,\ldots,N\}$ denotes the set of indices corresponding to the vertices of $\mathcal{C}_i$. We have that $Vert(\mathcal{C}_i) = \{w_j\}_{j\in\mathcal{V}_i}$, which were obtained by the Quick Hull algorithm.

Proof: We know from Section 2.2.1 that the inner problem in 2-3 can be written as (2-16). However, the constraint

$$h(w;\beta) - \sum_{l\in\mathcal{A}(i)} (\kappa_l - \lambda_l) \leq 0, \forall w \in \mathcal{C}_i, i \in \mathcal{F} \tag{2-18}$$

makes the problem semi-infinite and computationally expensive. Nonetheless, as the constraint must be valid for all $w \in \mathcal{C}_i$ for each $i \in \mathcal{F}$, we know that it must be valid for the maximum value of w as well. Given the maximization is over the variable w, the summation term can be written outside the optimization portion, resulting in:

$$\max_{w\in\mathcal{C}_i} \Big\{ h(w;\beta) \Big\} - \sum_{l\in\mathcal{A}(i)} (\kappa_l - \lambda_l) \leq 0, \forall i \in \mathcal{F} \tag{2-19}$$

Since by hypothesis we have convex polyhedral sets, and $h(W;\beta)$ is a convex function, the optimal objective value of these maximization problems, for each $i \in \mathcal{F}$, is achieved in one of their respective vertices, whose indices are given by $\mathcal{V}_i$. The sets $\mathcal{V}_i$, $i \in \mathcal{F}$, are also known by hypothesis, which allows us to rewrite the semi-infinite constraint (2-18) as the finite version

$$h(w_j;\beta) - \sum_{l\in\mathcal{A}(i)} (\kappa_l - \lambda_l) \leq 0, \forall j \in \mathcal{V}_i, i \in \mathcal{F}, \tag{2-20}$$

which concludes the proof. ■

We have proposed the PolieDRO framework that incorporates a data-driven regularization approach that does not require hyperparameters. Our framework is designed to construct ambiguity sets in an efficient and purely data-driven manner, making the resulting optimization problem numerically tractable.

2.3 Selected applications

Given any convex function $h(W;\beta)$, we can write its PolieDRO formulation (2-3), where the ambiguity set and its coverage probabilities are formed as proposed in Section 2.2.2 as the convex finite optimization problem described in equation (2-17). Such a class of problems can be efficiently solved using off-the-shelf solvers, which allows the usage of the PolieDRO framework in real-world applications.

For this section and the rest of this chapter, we explicitly represent the dataset $\{\mathbf{x_j}, y_j\}_{j=1}^N$ separating features $\mathbf{x_j}$ from labels y_j. Moreover, we consider that the probability distribution P in problem (2-3) only refers to the random vector of features X. More objectively, we define $W = X$, only considering the feature space, and the function $h(X;\beta) = E_{P_{Y|X}}[\ell(X,Y;\beta)]$ as the conditional expectation of the loss function ℓ over Y given X. In the finite reformulation (2-17), we approximate the conditional expectation by its empirical benchmark

$$h(w_j;\beta) \approx \frac{1}{|\mathcal{K}_j|} \sum_{k\in\mathcal{K}_j} \ell(x_k, y_k;\beta),$$

where $\mathcal{K}_j = \{k \in \{1,\ldots,N\} \,|\, x_k = x_j\}$. It's worth noting that in the majority of applications, the set $\mathcal{K}_j$ typically consists of only one element, where $k = j$. In simpler terms, each observation j possesses a unique set of features $\mathbf{x}_j$ that distinguish it. In such scenarios, we observe that the function $h(w_j;\beta)$ is approximated by the loss $\ell(x_j, y_j;\beta)$. For notation simplicity, we consider this case to present the PolieDRO reformulation in the following subsections, whereby we replace $h(w_j;\beta)$ by each specific loss function $\ell(x_j, y_j;\beta)$.

As a consequence, we use the training set of features $\{\mathbf{x}_j\}_{j=1}^N$ to determine the set of observations $\{A_i\}_{i=1}^{\mathcal{F}_\rangle}$, the convex hulls $\{\mathcal{C}_\rangle\}_{i=1}^{\mathcal{F}}$ and the set of vertices' indices $\{\mathcal{V}\}_{i=1}^{\mathcal{F}}$ following Algorithm (1), where $\mathcal{F}$ is also calculated in the algorithm execution.

To assess the PolieDRO framework applicability, we first investigate three common loss functions that are used in well-known ML methods. We briefly present such methods and their PolieDRO benchmarks that share the same loss function at their core. Due to their generality, we apply them to two main tasks in the field: classification and regression.

2.3.1 Hinge loss

The first loss function we investigate is the hinge loss, used in the task of classification. The hinge loss aims to linearly penalize whenever an observation is misclassified based on its features $\mathbf{x_i}$. In this formulation, we let $y_i \in \{-1, 1\}$ reflect the label of observation i, and β, the parameters to be tuned, be represented by the pair $(\beta_0, \boldsymbol{\beta}_1)$, $\beta_0 \in \mathbb{R}$, $\boldsymbol{\beta}_1 \in \mathbb{R}^d$:

$$\ell(\mathbf{x}_j, y_j; \beta_0, \boldsymbol{\beta}_1) = \max\{1 - y_j(\boldsymbol{\beta}_1^T \mathbf{x}_j - \beta_0), 0\} \tag{2-21}$$

Naturally, such values of the parameters $(\beta_0, \boldsymbol{\beta}_1)$ need to be specified to build a classifier.

A popular option is the so-called (soft-margin) support vector machine (SVM), which is a class of methods proposed by (CORTES; VAPNIK, 1995) that relax the requirement of linear separability of the simpler maximal margin classifier and allows for points to be incorrectly classified. The soft-margin support vector machine balances the minimization of total loss calculated following (2-21) with the size of the classification margin using a normed-value of $\boldsymbol{\beta}_1$. The final problem can be described as:

$$\min_{\boldsymbol{\beta}_1, \beta_0} \quad \tfrac{1}{2}\|\boldsymbol{\beta}_1\|_2^2 + C\sum_{j=1}^{N} \max\{1 - y_j(\boldsymbol{\beta}_1^T \mathbf{x}_j - \beta_0), 0\} \tag{2-22}$$

where C is a hyperparameter that needs to be tuned using a validation procedure, for example.

Problem (2-22) can be represented equivalently as:

$$\begin{aligned} \min_{\boldsymbol{\beta}_1, \beta_0} \quad & \tfrac{1}{2}\|\boldsymbol{\beta}_1\|_2^2 + C\sum_{j=1}^{N} \xi_i \\ s.t. \quad & y_j(\boldsymbol{\beta}_1^T \mathbf{x}_j - \beta_0) \geq 1 - \xi_j, \quad j = 1, ..., n \\ & \xi_j \geq 0, \qquad\qquad\qquad\quad j = 1, ..., n \end{aligned} \tag{2-23}$$

which is a convex quadratic optimization problem with efficient open-source implementations. More about the implementation and solution of this problem can be found in (HASTIE; TIBSHIRANI; FRIEDMAN, 2001).

In (BLANCHET; KANG; MURTHY, 2019), the authors show a reinter-

pretation of this model as a DRO problem, where the hyperparameter C is understood to be the radius of a Wasserstein distance measure between the empirical distribution and the set of considered distributions.

In contrast to both formulations, our implementation of the hinge loss for a classification task under the PolieDRO framework does not rely on a hyperparameterized balance between the total loss and the parameters' magnitude or a distribution ball radius under some measure.

Given a set of training data $\{\mathbf{x_j}, y_j\}_{j=1}^n$, we apply Algorithm 1 to obtain the convex hull sets and the vertices. We then plug the hinge loss function (2-21) into formulation (2-17) to obtain a novel classification method. For tractability, we make use of an additional variable η to implement the hinge loss function as a constraint, where each η_j is associated with each $w_j, j \in \mathcal{V}_i, i \in \mathcal{F}$:

$$\begin{aligned}
\min_{\beta_0,\beta_1,\lambda,\kappa,\eta} \quad & \sum_{i\in\mathcal{F}}(\kappa_i\overline{p_i} - \lambda_i\underline{p_i}) \\
\text{s.t.} \quad & \eta_j - \sum_{l\in\mathcal{A}(i)}(\kappa_l - \lambda_l) \leq 0, \quad \forall j \in \mathcal{V}_i, i \in \mathcal{F} \\
& \eta_j \geq 1 - y_j(\boldsymbol{\beta}_1^T\mathbf{x}_j - \beta_0), \quad \forall j \in \mathcal{V}_i, i \in \mathcal{F} \\
& \eta_j \geq 0, \qquad\qquad\qquad \forall j \in \mathcal{V}_i, i \in \mathcal{F} \\
& \lambda_i \geq 0, \qquad\qquad\qquad \forall i \in \mathcal{F} \\
& \kappa_i \geq 0, \qquad\qquad\qquad \forall i \in \mathcal{F}.
\end{aligned} \tag{2-24}$$

Remark: The final formulation of the PolieDRO version using the hinge loss function is a linear optimization problem that does not rely on any hyperparameter. In comparison with the SVM method, for example, we do not require a validation procedure to calibrate the parameter C as in (2-23), learning all the necessary structures from the observed data.

2.3.2 Logistic loss

Still, in the classification task realm, we consider the logistic loss function. Letting $y_j \in \{-1, 1\}$, the logistic loss of a given observation $\mathbf{x}_j$ with associated label y_j and parameters $(\beta_0, \boldsymbol{\beta}_1)$, $\beta_0 \in \mathbb{R}$, $\boldsymbol{\beta}_1 \in \mathbb{R}^p$ is given by:

$$\ell(\mathbf{x}_j, y_j; \beta_0, \boldsymbol{\beta}_1) = \log\left(1 + e^{-y_j(\beta_0 + \boldsymbol{\beta}_1^T\mathbf{x}_j)}\right). \tag{2-25}$$

Such loss function assumes the dependent variable Y, from which $\{y_j\}_{j=1}^N$ is observed, follows a Bernoulli distribution whose probability depends on the observed $\mathbf{X}$ and the parameters $(\beta_0, \boldsymbol{\beta}_1)$. To determine such parameters, we typically use the maximum-likelihood method, which results in the following unconstrained optimization problem:

$$\max_{\beta_0,\boldsymbol{\beta}_1} \sum_{j=1}^{N} -\log\left(1+e^{-y_j(\beta_0+\boldsymbol{\beta}_1^T\mathbf{x}_j)}\right). \tag{2-26}$$

Albeit popular, the logistic regression can quickly become unstable and sometimes it does not generalize well for untrained data. The regularized logistic regression classifier is often a more suitable method. Similar to SVM, this approach aims to control the overfitting to the training data by balancing the total loss with a metric on the size of the estimated parameters, typically using the l_2-norm. The formulation is given in equation 2-27.

$$\min_{\beta_0,\boldsymbol{\beta}_1} \sum_{i=1}^{N} \log\left(1+e^{-y_j(\beta_0+\boldsymbol{\beta}_1^T\mathbf{x}_j)}\right) + \lambda\|\boldsymbol{\beta}_1\|_2^2. \tag{2-27}$$

Under the PolieDRO framework, we bypass the need for hyperparameter—in this case, the λ value—calibration, which is dependent on a heuristic procedure. Instead, we make use of the logistic loss function and write the proposed formulation in (2-17) as:

$$\begin{aligned}
\min_{\beta_0,\beta_1,\lambda,\kappa} \quad & \sum_{i\in\mathcal{F}}(\kappa_i\overline{p_i} - \lambda_i\underline{p_i}) \\
\text{s.t.} \quad & \log\left(1+e^{-y_j(\beta_0+\boldsymbol{\beta}_1^T\mathbf{x}_j)}\right) - \sum_{l\in\mathcal{A}(i)}(\kappa_l-\lambda_l) \le 0, \quad \forall j\in\mathcal{V}_i, i\in\mathcal{F} \\
& \lambda_i \ge 0, \qquad\qquad \forall i\in\mathcal{F} \\
& \kappa_i \ge 0, \qquad\qquad \forall i\in\mathcal{F}.
\end{aligned} \tag{2-28}$$

Equation (2-28) is a convex optimization problem that can be efficiently solved. Notice that there is no need to use a heuristic procedure to select a hyperparameter to be used in this approach. The parameters $(\beta_0, \boldsymbol{\beta}_1)$ are directly obtained from a single run of the stated problem.

2.3.3 Mean Squared Error loss

Regression is another typical task in ML, where instead of predicting a class based on the available features, one predicts continuous values. Such quantities are called predictions, usually denoted as $\hat{y}$. For ease of notation, let $\hat{y}_j = f(\mathbf{x}_j; \beta_0, \boldsymbol{\beta}_1)$, that is, the j^{th}-prediction is function of its corresponding features and some need-to-be calibrated parameters.

To assess the quality of this function f, we need a loss function to compare the predicted values $\hat{y}$ with the actual observations y. The most popular one is the mean squared loss (MSE). Given a set of observations $\{\mathbf{x}_j\}_{j=1}^N$ and actual observations of the dependent variable $\{y\}_{j=1}^N$, we aim to estimate the model f that minimizes the MSE, given by:

$$\sum_{j=1}^{N}\Big(y_j-\hat{y}_j\Big)^2=\sum_{j=1}^{N}\Big(y_j-f(\mathbf{x}_j;\beta_0,\boldsymbol{\beta}_1)\Big)^2 \tag{2-29}$$

Although there are several classes of functions to choose from, the most common one is the linear regression model. In this case, under some assumptions (see (HASTIE; TIBSHIRANI; FRIEDMAN, 2001)), we have $\hat{y}=\beta_0+\boldsymbol{\beta}_1^T\mathbf{x}$. In this simpler setting, also known as the linear regression method, the pair of parameters $(\beta_0,\boldsymbol{\beta}_1)$ is the one that minimizes the resulting MSE (2-30).

$$(\beta_0,\boldsymbol{\beta}_1)=\underset{\beta_0,\boldsymbol{\beta}_1}{\arg\min}\sum_{j=1}^{N}\Big(y_j-\beta_0+\boldsymbol{\beta}_1^T\mathbf{x}_j\Big)^2 \tag{2-30}$$

From an ML perspective, it is well known that there are some drawbacks when implementing the linear regression method directly. For example, its estimation instability under high-dimensional settings and occasional overfitting paved the way for several suggestions for improvement. One highly effective option is the LASSO (least absolute shrinkage and selection operator) regression (TIBSHIRANI, 1996) that performs an l_1 regularization to enhance the resulting model's generalization power. To do so, it introduces a hyperparameter λ that needs to be heuristically calibrated to optimize the balance between the resulting loss calculated and the parameter's freedom of choice. The LASSO method parameters are determined following the given convex optimization problem:

$$(\beta_0,\boldsymbol{\beta}_1)=\underset{\beta_0,\boldsymbol{\beta}_1}{\arg\min}\sum_{j=1}^{N}\Big(y_j-\beta_0-\boldsymbol{\beta}_1^T\mathbf{x}_j\Big)^2+\lambda\|\boldsymbol{\beta}_1\|_1 \tag{2-31}$$

It has been shown in (BLANCHET; KANG; MURTHY, 2019) that a slight variation on the LASSO formulation (square-root LASSO, (BELLONI; CHERNOZHUKOV; WANG, 2011)) can be written as a DRO problem under a Wasserstein-based metric. In this context, the hyperparameter λ has a similar interpretation to the hyperparameter in the SVM and Regularized Logistic Regression formulations explored before.

By using the PolieDRO framework we define $\ell(\mathbf{x}_j,y_j;\beta)=\Big(y_j-(\beta_0+\boldsymbol{\beta}_1^T\mathbf{x}_j)\Big)^2$ and perform a regression by using the optimal solution $(\beta_0^*,\boldsymbol{\beta}_1^*)$ of the following hyperparameter-free, convex quadratic optimization problem:

$$\begin{aligned}
\min_{\beta_0,\beta_1,\lambda,\kappa}\quad & \sum_{i\in\mathcal{F}}(\kappa_i\overline{p_i}-\lambda_i\underline{p_i}) & & \\
\text{s.t.}\quad & \Big(y_j-(\beta_0+\boldsymbol{\beta}_1^T\mathbf{x}_j)\Big)^2-\sum_{l\in\mathcal{A}(i)}(\kappa_l-\lambda_l)\le 0, & \forall j\in\mathcal{V}_i, i\in\mathcal{F} & \\
& \lambda_i\ge 0, & \forall i\in\mathcal{F} & \\
& \kappa_i\ge 0, & \forall i\in\mathcal{F}. &
\end{aligned} \tag{2-32}$$

2.4 Computational Experiments

In this section, we perform an extensive computational experiment to compare the PolieDRO variation of the three loss functions in both classification and regression tasks, detailed in Section 2.3. Such experiments were performed both on synthetic data sets and real-world data sets, with great variability between them.

2.4.1 Experimental setup

On the classification front, we tested two implementations of the PolieDRO framework using common loss functions (hinge loss and logistic loss). Such performance is measured in terms of accuracy. On the regression front, we applied the common Mean Squared Loss function, measuring the performance using the mean squared error. In both cases, we considered a confidence level of 90% (that is, $\alpha = 10\%$). In the section 2.5.2 we discuss more details about such selection and perform a sensitivity analysis for all conducted experiments.

To report in a more representative way the performance of the models, we conducted a 5-fold cross-validation procedure for the standard models. Although applied in the same data set for each problem, the training-testing procedure is slightly different between the PolieDRO and the benchmarks. While those models require some sort of calibration procedure, which ultimately leads to a different sequence of steps, the PolieDRO approach avoids such time-consuming and error-prone steps. In what follows we give a precise description of the steps we undertook for each case:

Experiment iteration

1. **Data Split:** Randomly split the dataset into 2 parts: cross-validation (80%) and testing set (20%).

2. **benchmark models:**

   (a) Split the cross-validation set into 5 folds

   (b) Train the model for a (discrete) variety of possible hyperparameters using 4 out of the 5 folds as the training set.

   (c) Calculate each trained model's suitable performance metric applying it to the remaining fifth fold (validation fold).

(d) Repeat items (2.2) and (2.3) for each possible combination of the training set (combining 4 folds) and validation fold. This results in a total of 5 executions of steps (2.2) and (2.3).

(e) For each hyperparameter-trained model, average its performance metric from each repetition of the previous steps and select the best-performing model.

(f) Retrain the best-performing model on the complete cross-validation set, using the highlighted hyperparameter.

(g) Calculate the final performance on the (unused) testing set.

3. **PolieDRO framework:**

(a) Apply Algorithm (1) to the complete cross-validation set.

(b) Solve the appropriate optimization model using the output of the previous step for a predefined desired significance level.

(c) Calculate the final performance on the (unused) testing set.

It is worth highlighting that the training, validation and testing sets available for the PolieDRO and benchmark models are the same at each iteration.

Notice that at each iteration we need to perform a significantly more cumbersome sequence of steps for the traditional models, which are ultimately dependent on the predefined set of possible values for the hyperparameters. Additionally, such selection is highly sample-dependent, as a different random split of the folds may ultimately lead to a different optimal hyperparameter.

It is crucial to point out that there is an inherent trade-off when searching for the appropriate hyperparameter. Due to the discrete nature of the procedure, the search for the optimal hyperparameter must balance exploration (the number of candidate values) and feasibility (the time required to calculate such sub-steps). That is, as the need for a more precise selection increases the time and energy spent in the training step also increases. In addition, it is well known that the configuration for selecting the best hyperparameter for ML models has a direct impact on the model's performance (see (YANG; SHAMI, 2020)). In the same work, the authors present several automatic optimization techniques, highlighting different strengths and drawbacks.

On the opposite end, the PolieDRO-based models have a much more streamlined process to be constructed, first by constructing the convex hulls and then solving a tractable optimization problem. Naturally, such hulls vary according to the sampled data.

2.4.2 Computational Experiments with Real World Data Sets

To comprehensively compare the performance of the PolieDRO models against their benchmarks with real-world data sets, we performed the experiments on a selection of 100 problems from the UCI Machine Learning Repository. On the classification front, we compared the SVM model and the logistic regression with the PolieDRO applications of the hinge loss and the logistic loss, respectively, using 70 different data sets. We conducted the *one-versus-rest* problem of predicting the occurrence of the first class in each data set. Such performance is measured in terms of accuracy. On the regression front, we compared the PolieDRO framework applied to the MSE loss function with the LASSO model on a total of 30 different data sets.

We repeat the entire experiment iteration five times for each comparison and calculate the mean value of the relevant metrics, following (BERTSIMAS et al., 2019). The results are organized in tables 2.1, 2.2 and 2.3. For each data set, the best result (or multiple in the case of ties) for each comparison (PolieDRO version against its nominal benchmark) is indicated in bold, according to the selected performance measure. In addition, for the classification task, the best method overall for the data set is underlined.

Finally, Table 2.4 summarizes the pairwise results for each loss function. Such results are also seen in Figure 2.7, where we count as a win whenever the PolieDRO framework application achieves a better result than the nominal benchmark. Naturally, when comparing accuracy, in classification tasks, the higher the better. On the regression tasks, the smaller the MSE the better the result. In addition to pairwise comparisons, we also report in Table 2.5 the number of wins, ties, and losses between the best PolieDRO classification method and the best nominal benchmark - that is, the underlined results in tables 2.1 and 2.2

To capture some intuition behind the results, figures 2.8, 2.9, and 2.10 plot the comparison results for each loss function against two attributes of the data set - the number of samples available n and the dimension of the feature space d. As can be seen, there is no clear pattern indicating whether there is an advantageous set of characteristics at first for the PolieDRO framework usage. However, we highlight that the competitive overall performance indicates that the PolieDRO framework application achieves superior results as compared to the classical methods in the literature while bypassing the need for hyperparameter calibration. In other words, there is no apparent trade-off in place for applying a distributionally robust and tractable optimization problem sharing the same loss function as the common methods explored.

			Hinge Loss		Logistic Regression	
Data set Name	n	p	PolieDRO	Nominal	PolieDRO	Nominal
acute-inflammations-1	120	6	**1.0000**	**1.0000**	**1.0000**	**1.0000**
acute-inflammations-2	120	6	**1.0000**	**1.0000**	**1.0000**	0.9833
balance-scale	625	4	**0.9488**	**0.9488**	0.8928	**0.9632**
balloons-a	20	4	**1.0000**	0.8500	**1.0000**	**1.0000**
balloons-b	20	4	**1.0000**	0.9000	**1.0000**	0.8500
balloons-c	20	4	**1.0000**	0.8500	**1.0000**	0.8500
balloons-d	16	4	**0.6667**	**0.6667**	**0.7333**	0.5333
banknote-authentication	1372	4	0.9883	**0.9890**	**0.9898**	**0.9898**
blood-transfusion-service-center	748	4	**0.7573**	0.7530	0.7920	**0.7922**
breast-cancer	277	31	**0.7614**	0.7578	**0.7600**	0.7392
b-cancer-wisconsin-diagnostic	569	30	0.9385	**0.9561**	0.9491	**0.9526**
b-cancer-wisconsin-original	683	9	0.9676	**0.9693**	0.9588	**0.9605**
b-cancer-wisconsin-prognostic	194	32	0.7692	**0.8205**	**0.7589**	0.7333
car-evaluation	1728	15	**0.9479**	0.9472	0.9456	**0.9576**
climate-model-simulation-crashes	540	18	**0.9625**	0.9500	**0.9500**	**9500**
congressional-voting-records	232	16	**0.9958**	0.9785	**0.9751**	**0.9751**
connectionist-bench	990	10	**0.9707**	0.9696	**0.9545**	**0.9545**
connectionist-bench-sonar	208	60	0.7761	**0.8195**	0.7238	**0.8195**
contraceptive-method-choice	1473	11	**0.6861**	0.6795	**0.6904**	**0.6904**
credit-approval	690	9	**0.8656**	**0.8656**	**0.8398**	**0.8398**
dermatology	358	34	**1.0000**	**1.0000**	**0.9944**	0.9915
echocardiogram	62	7	0.7333	**0.7538**	**0.8000**	0.7846
ecoli	336	7	**0.9611**	0.9582	**0.9611**	0.9552
fertility	100	12	**0.8500**	**0.8500**	**0.8500**	**0.8500**
flags	194	58	0.7538	**0.8264**	**0.8410**	0.7743
glass-identification	214	9	**0.7249**	0.7163	**0.7385**	**0.7385**
haberman-survival	306	3	**0.7475**	**0.7475**	**0.7180**	0.7114
hayes-roth	132	4	**0.6846**	**0.6846**	**0.7555**	0.7329
heart-disease-cleveland	297	18	**0.8610**	0.8556	**0.8400**	**0.8400**
heart-disease-hungarian	294	6	**0.7263**	0.6947	**0.8384**	**0.8384**
heart-disease-switzerland	123	6	**0.6500**	**0.6500**	0.5272	**0.6727**
heart-disease-va	200	7	**0.7411**	0.7111	**0.7307**	**0.7307**
hepatitis	155	4	**0.8000**	**0.8000**	0.7250	**0.8250**
image-segmentation	210	19	0.9619	**0.9857**	**0.9952**	**0.9952**
indian-liver-patient	583	9	**0.7258**	**0.7258**	**0.7517**	0.7396

Table 2.1: Mean out of sample accuracy.

			Hinge Loss		Logistic Regression	
Data set Name	n	p	PolieDRO	Nominal	PolieDRO	Nominal
ionosphere	351	34	0.8285	**0.8514**	**0.8742**	**0.8742**
iris	150	4	**1.0000**	**1.0000**	**1.0000**	**1.0000**
lenses	24	5	**0.8000**	0.6800	0.7200	**0.7600**
letter-recognition	20000	16	0.9711	**0.9728**	0.9885	**0.9902**
libras-movement	360	90	0.8833	**0.9155**	0.9388	**0.9722**
mammography-mass	830	10	**0.8384**	0.8152	**0.8277**	**0.8277**
monks-problems-1	124	11	**0.8400**	0.7520	**0.6880**	**0.6880**
monks-problems-2	169	11	**0.6470**	**0.6470**	**0.6352**	0.6117
monks-problems-3	122	11	**0.9284**	0.8880	**0.9326**	0.8880
mushroom	5644	76	**1.0000**	**1.0000**	**1.0000**	**1.0000**
nursery	12690	19	**1.0000**	**1.0000**	**1.0000**	**1.0000**
ozone-level-detection-eight	1847	72	0.9268	**0.9324**	**0.9322**	0.9152
ozone-level-detection-one	1848	72	**0.9696**	0.9529	**0.9700**	0.9665
parkinsons	195	21	**0.8615**	0.8358	**0.8615**	0.8358
plannig-relax	182	12	**0.7155**	**0.7155**	0.7005	**0.7225**
qsar-biodegradation	1055	41	**0.8786**	0.8663	**0.8473**	**0.8473**
seeds	210	7	**0.9333**	0.9165	**0.9619**	0.9380
seismic-bumps	2584	20	**0.9342**	**0.9342**	0.9279	**0.9346**
soybean-large	266	63	0.7745	**0.7872**	**0.7764**	0.7625
soybean-small	47	37	**1.0000**	**1.0000**	**1.0000**	**1.0000**
spambase	4601	57	**0.9265**	**0.9265**	**0.9230**	**0.9230**
statlog-project-landsat-sat	4435	36	0.9846	**0.9862**	**0.9833**	**0.9833**
teaching-assistant-evaluation	151	52	**0.7466**	0.6800	**0.7000**	0.6933
thoracic-surgery	470	16	0.8446	**0.8510**	0.8744	**0.8808**
thyroid-disease-allbp	1947	25	0.9696	**0.9697**	0.9562	**0.9600**
thyroid-disease-allhyper	1947	25	**0.9830**	0.9794	**0.9789**	**0.9789**
thyroid-disease-allrep	1947	25	0.9697	**0.9778**	**0.9742**	0.9723
thyroid-disease-sick	1947	25	**0.9537**	0.9523	0.9475	**0.9625**
tic-tac-toe-endgame	958	18	**0.9843**	0.9842	**0.9801**	0.9732
wall-following-robot-nav-2	5456	2	**0.6300**	0.6120	**0.6584**	0.6557
wall-following-robot-nav-24	5456	24	**0.7547**	0.7543	0.7065	**0.7536**
wall-following-robot-nav-4	5456	4	**0.6381**	0.6229	**0.6489**	0.6139
wine	120	6	**0.9714**	0.9666	**0.9028**	0.8955
yeast	1484	8	**0.6740**	**0.6740**	**0.6996**	0.6828
zoo	101	16	**1.0000**	**1.0000**	**1.0000**	**1.0000**

Table 2.2: Mean out of sample accuracy.

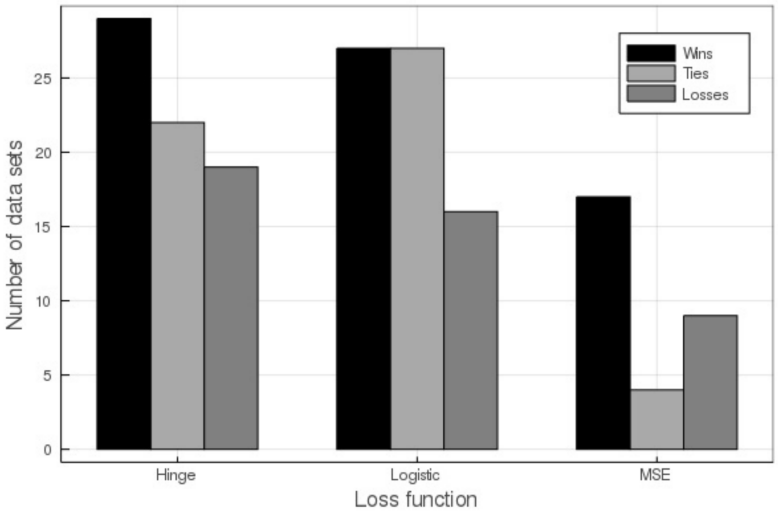


Figure 2.7: Pairwise comparison of the different methods for each loss function.

To further investigate the experiment's results, figures 2.12, 2.13, and 2.14 plot the wins-ties-losses of the PolieDRO variation of each loss function against the dimension of the feature space d and the winning method's performance measured out-of-sample (and reported in tables 2.1, 2.2, and 2.3). Both axes are in log-scale for better visualization. In addition, since the MSE is not a comparable metric across different datasets, we replace it with the root mean square error (RMSE) normalized by the average value of the predicted variable.

			Regression	
Data set Name	n	p	PolieDRO	LASSO
abalone	4177	9	**2.1179**	2.1216
airfoil-self-noise	1503	5	**22.3516**	23.3844
airline-costs	31	9	**0.04281**	0.04923
auto-mpg	392	8	3.5776	**3.5618**
automobile	159	31	**0.2461**	0.3684
beer-aroma	23	7	**7.4046**	8.5012
communities-and-crime	1993	100	**383.4650**	383.8383
computer-hardware	209	36	**32.9909**	33.6611
concrete-slump-test-compressive	103	7	**2.89302**	3.1134
concrete-slump-test-slump	103	7	7.5673	**7.5291**
construction-maintenance	33	4	3.5981	**3.2313**
cpu-act	8192	21	**10.4719**	**10.4719**
forest-fires	517	27	**42.9497**	43.1722
home-mortgage	18	6	**18.1800**	18.6598
housing	506	13	**4.7854**	4.7872
immigrant-salaries	35	3	**1.7360**	**1.7360**
japan-emigration	45	5	168.6030	**150.9830**
kin8nm	8192	8	**0.0413**	**0.0413**
lpga-2208	157	6	**0.4291**	0.4323
lpga-2009	146	11	**0.4676**	0.5037
parkinsons-telemonitoring-motor	5875	16	**7.7778**	7.7964
parkinsons-telemonitoring-total	5875	16	10.3067	**10.3043**
pyrim	74	27	**0.1434**	**0.1434**
texas-jan-temp	16	3	**1.2152**	1.27907
triazines	186	60	0.1368	**0.1330**
tv-sales	31	8	**3019.7855**	3028.6666
wiki4he	435	53	6.6882	**6.4113**
wine-quality-red	1599	11	**0.6169**	0.6520
wine-quality-white	4898	11	0.7710	**0.7650**
yatch-hydrodynamics	308	6	9.2546	**9.1323**

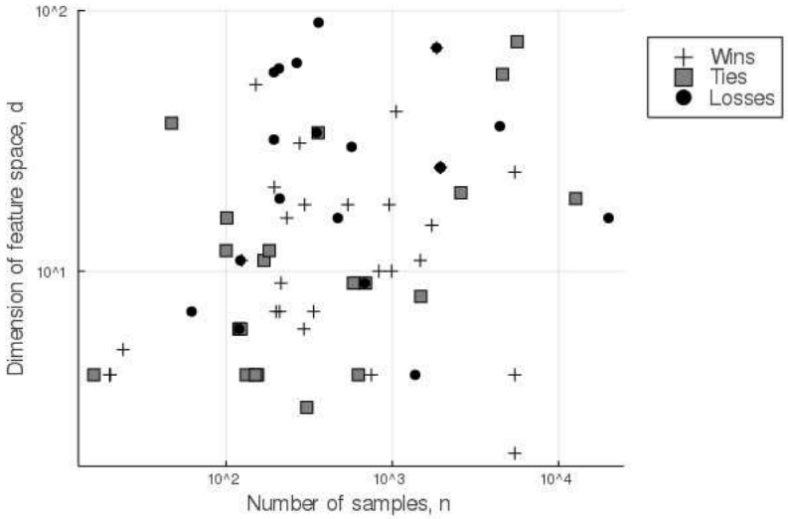
Table 2.3: Average out of sample MSE.

Loss function	Metric	Wins	Ties	Losses	Total data sets
Hinge Loss	Accuracy	29	22	19	70
Logistic Loss	Accuracy	27	27	16	70
MSE Loss	MSE	17	4	9	30

Table 2.4: Pairwise comparison for each loss function.

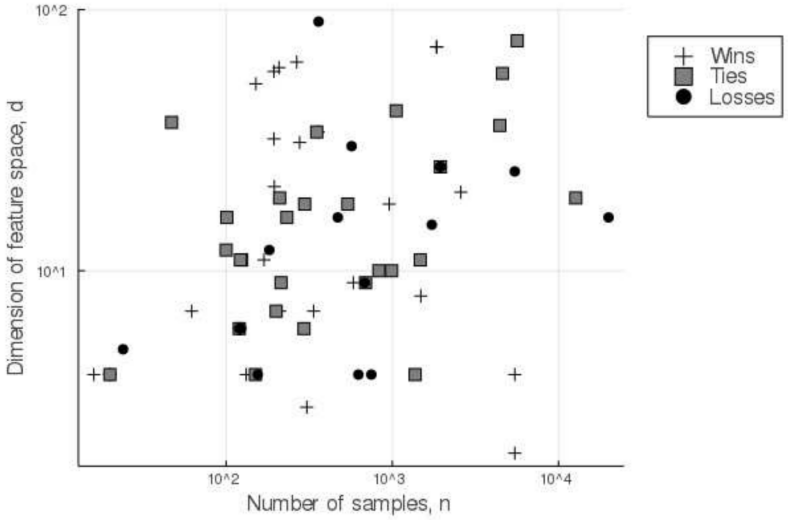
	Metric	Wins	Ties	Losses	Total data sets
Classification	Accuracy	29	21	20	70

Table 2.5: General classification task comparison.



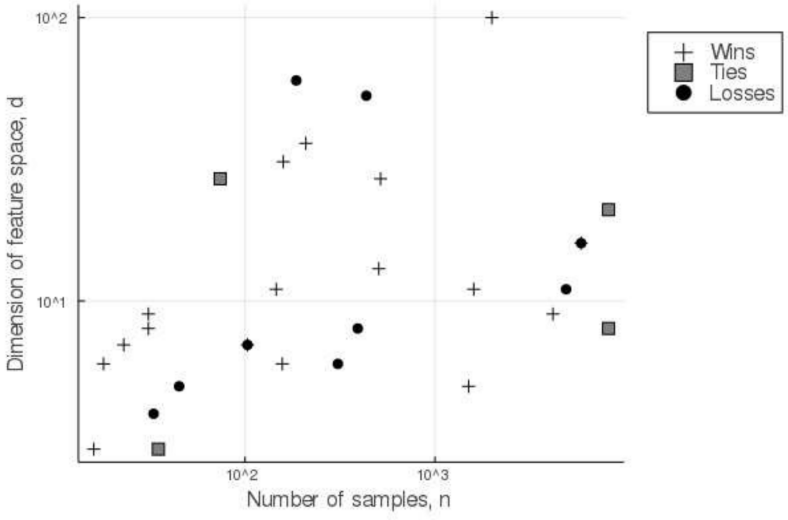
0.75

Figure 2.8: Hinge loss performance against nominal SVM.



0.75

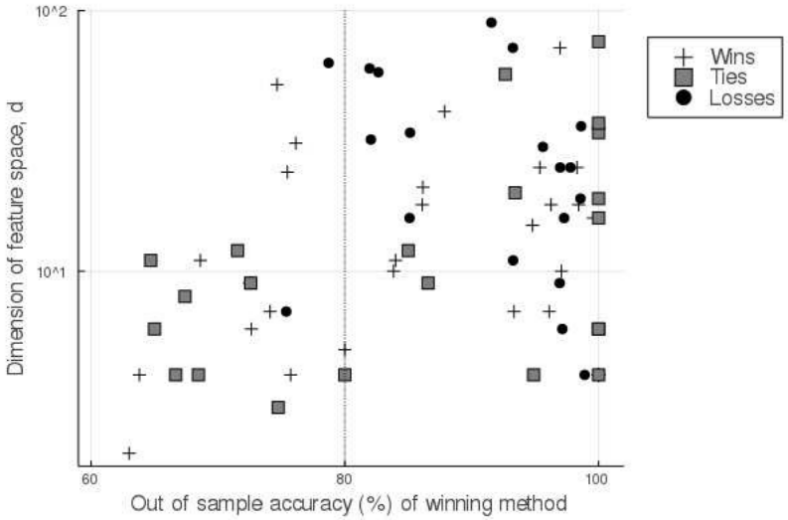
Figure 2.9: Logistic loss performance against nominal Regularized Logistic regression.



0.75

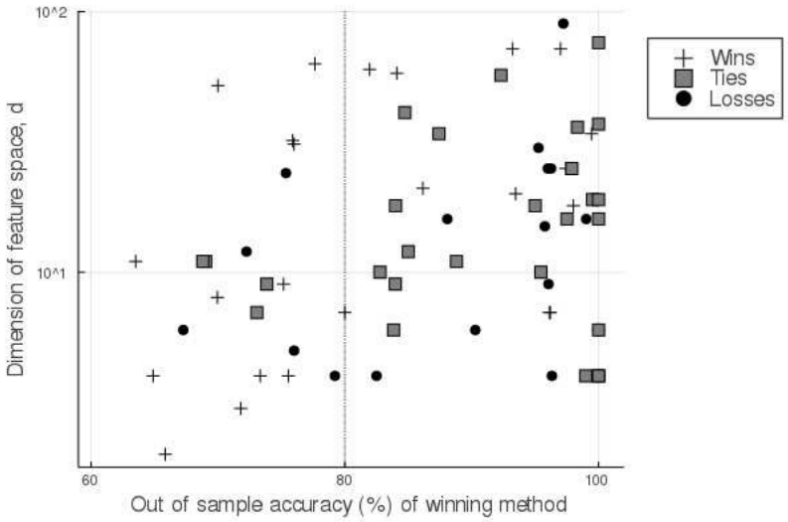
Figure 2.10: Mean squared loss performance against nominal LASSO.

Figure 2.11: Plot of PolieDRO version performance against benchmark for each data set the dimension of feature space and number of samples.



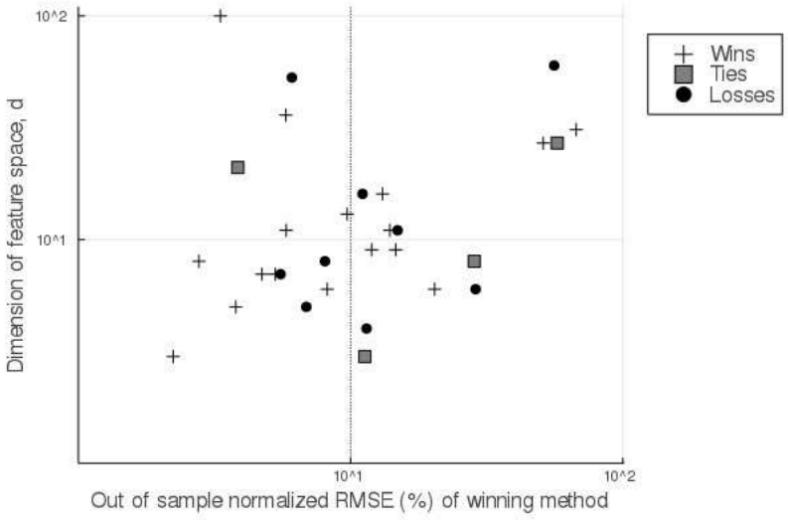
0.75

Figure 2.12: Hinge loss performance against nominal SVM.



0.75

Figure 2.13: Logistic loss performance against nominal Regularized Logistic regression.



0.75

Figure 2.14: Mean squared loss performance against nominal LASSO.

Figure 2.15: Plot of PolieDRO version performance against benchmark for each data set the dimension of feature space and out-of-sample accuracy of the winning method.

An interesting observation can be made regarding the classification loss functions based on figures 2.12 and 2.13. In tables 2.6 and 2.7 we split the performance for each comparison between so-called "hard" (optimal achieved out of sample accuracy lower or equal to 80%) and "easy" problems (optimal achieved out of sample accuracy greater than 80%). Similarly, we split the regression task between "easy" and "hard" problems in the 10% out of sample normalized RMSE line (see table 2.8 and figure 2.14).

	Wins	Ties	Losses	Total
Hard	11	9	2	22
Easy	18	13	17	48

Table 2.6: Performance of the PolieDRO hinge loss split between "hard" and "easy" problems.

	Wins	Ties	Losses	Total
Hard	13	4	5	22
Easy	14	23	11	48

Table 2.7: Performance of the PolieDRO logistic loss split between "hard" and "easy" problems.

	Wins	Ties	Losses	Total
Hard	10	1	4	15
Easy	7	5	3	15

Table 2.8: Performance of the PolieDRO Mean Squared Error loss split between "hard" and "easy" problems.

Noticeably, the PolieDRO framework variation in each loss function application has an even superior performance in "hard" problems against its nominal benchmarks. Even though the nominal benchmarks have a regularization procedure that aims to deal with overfitting and add robustness to the estimation, the PolieDRO framework seems to do it in a more effective way, and without hyperparameters. By construction, it optimizes over a distribution space shaped after the empirical observations without limiting itself to it. Evidence indicates that especially in harder problems this approach delivers better results.

2.4.3 Computational Experiments with Synthetic Data Sets

To further investigate the performance of the PolieDRO models against their benchmarks under conditions not present in the real-world data sets used in 2.4.2, we conducted an additional experiment using synthetically generated data sets. Both on the classification and regression fronts we measured the performance by varying the number of available samples and the number of features, thus varying the ratio between those parameters for each model to assess the impact in each case.

To accommodate the particular characteristics of the classification and the regression problems, we designed each variation distinctly.

Synthetic classification data

Inspired by (BERTSIMAS et al., 2019), the classification setting is generated in three parts. For a given ratio n/d (sample size versus number of features):

1. n_m points are generated as multivariate random normal, $N(1.5\mathbf{e}, \mathbf{I})$, where $\mathbf{e} \in \mathbb{R}^d$ is the vector of ones of dimension d and $\mathbf{I} \in \mathbb{R}^{d\times d}$ is the identity matrix. These points are given the label $+1$.

2. n_m points are generated as multivariate random normal, $N(-1.5\mathbf{e}, \mathbf{I})$ and given the label -1.

3. n_o outlier points are generated as multivariate random normal, $N(0, 3\mathbf{I})$. These points are randomly given the label $+1$ or -1.

where $n = 2 \times n_m + n_o$. We designed the data generation process in a way that the n_o represents 20% of the total available data.

Synthetic regression data

Similarly, the regression setting is designed to allow the investigation of different values of the ratio n/d. We start with a basic linear relationship with an added random noise: $y = \boldsymbol{\beta}^T\mathbf{x} + \boldsymbol{\varepsilon}$. We consider $\beta \in \mathbb{R}^d$ as a vector of ones, $X \sim N(1.5\mathbf{e}, \mathbf{I})$ and $\varepsilon \sim N(0, 0.1\mathbf{I})$.

In both classification and regression settings, we considered the possible number of dimensions (number of features) $d = \{2, 10, 100, 1000\}$ and the sample size of $n = \{5, 10, 50, 100, 500, 1000\}$ (available for the cross-validation

step). In each combination (d, n), we repeated the whole process 1,000 times. Results are organized in Table 2.10.

The calculated means and standard deviation (SD) in Table 2.10 refer to the difference in performance for each instance between the PolieDRO models and their benchmarks. For classification problems, we used the accuracy of class prediction as the proper measure, while for the regression we used the RMSE. To provide a unified basis for comparison, the difference between performance metrics is calculated as follows:

– For the classification problems, we calculate the difference between the results of the PolieDRO model and their benchmarks. The greater the difference, the better the relative performance for the PolieDRO version.
– For the regression problem, since the smaller the absolute metric the better, we inverted the order and calculated the difference between the benchmark models and the PolieDRO versions. In this way, we also have the greater the difference, the better relative performance achieved by the PolieDRO framework.

Besides reporting the mean and standard deviation of the difference between the performance measures in each case, we also report the total wins (W), ties (T) and losses (L) incurred by each PolieDRO version – that is, for the hinge loss, the logistic loss and the MSE loss applications of the proposed framework against their nominal benchmarks in Table 2.9.

Table 2.10 shows that the PolieDRO models consistently outperform their benchmarks, with more wins than losses, suggesting that the PolieDRO framework more frequently induces a slightly different and superior estimation for the models compared to nominal methods. This trend is particularly pronounced in high-dimensional conditions (when the n/d ratio is smaller). To make it easier to understand, we have bolded the highest number among the wins, ties, and losses for each row and model comparison.

It is important to note, however, that there are still a significant number of ties, especially in the classification experiments. This indicates that in some cases, both methodologies result in the same estimated values for the loss function parameters. Nevertheless, when the estimated values differ, the PolieDRO framework generally produces a better out-of-sample performance metric than the nominal methods.

	Metric	Wins	Ties	Losses	Total
Hinge Loss	Accuracy	11	9	4	24
Logistic Loss	Accuracy	10	11	3	24
MSE Loss	RMSE	17	0	7	24

Table 2.9: Pairwise performance of the PolieDRO models against their nominal benchmarks.

			Classification – Hinge Loss					Classification – Logistic Loss					Regression – MSE Loss				
d	n	n/d	Mean	SD.	W	T	L	Mean	SD.	W	T	L	Mean	SD.	W	T	L
2	5	2.5	0.0164	0.0364	**621**	298	81	0.0152	0.0098	**607**	302	91	0.1525	0.2322	**592**	12	396
2	10	5	0.0324	0.0854	**696**	258	46	0.0212	0.0539	**650**	238	112	0.1233	0.1785	**527**	43	430
2	50	25	0.0185	0.0065	326	**482**	192	0.0177	0.0082	347	**475**	178	0.1852	0.1652	**612**	51	337
2	100	50	0.0025	0.0031	**422**	416	162	0.0037	0.0101	**427**	413	154	0.1109	0.1361	**609**	37	354
2	500	250	-0.0036	0.0223	295	307	**398**	-0.0036	0.0223	281	323	**395**	0.0995	0.0562	**482**	50	468
2	1000	500	0.0094	0.0581	276	**482**	222	0.0099	0.0311	271	**475**	228	-0.0152	0.0328	450	51	**499**
10	5	0.5	0.0012	0.0027	**511**	387	102	0.0018	0.0035	**526**	382	92	0.2122	0.2366	**483**	106	411
10	10	1	0.0038	0.0064	**395**	375	230	0.0046	0.0094	**385**	350	265	0.2852	0.2366	**412**	189	399
10	50	5	0.0084	0.0069	**491**	400	109	0.0091	0.0124	405	**501**	94	0.1349	0.1741	**592**	164	244
10	100	10	-0.0095	0.0014	295	**480**	225	-0.0098	0.0017	300	**460**	240	-0.0985	0.2006	392	196	**412**
10	500	50	-0.0014	0.0051	231	**515**	254	-0.0024	0.0066	231	**480**	289	-0.1851	0.2011	397	112	**491**
10	1000	100	0.0067	0.0021	310	312	**378**	0.0067	0.0021	300	307	**393**	0.1067	0.1124	**450**	211	339
100	5	0.05	0.0055	0.0025	**396**	384	220	0.0065	0.0022	**390**	374	236	1.2511	1.3351	**527**	137	336
100	10	0.1	0.0002	0.0033	203	**517**	180	0.0008	0.0038	198	**515**	187	1.4412	1.5112	**572**	95	333
100	50	0.5	0.0082	0.0035	**415**	298	287	0.0078	0.0045	**418**	296	286	0.9714	0.9416	**528**	121	351
100	100	1	0.0009	0.0065	263	**399**	338	0.0012	0.0088	250	**412**	338	0.5651	0.7528	439	40	**521**
100	500	5	-0.0017	0.0062	271	270	**359**	-0.0021	0.0081	261	275	**364**	0.3229	0.3281	484	13	**503**
100	1000	10	0.0021	0.0092	231	**488**	281	0.0021	0.0092	226	**499**	275	0.1145	0.1285	**381**	272	347
1000	5	0.005	0.0033	0.0072	**427**	399	174	0.0073	0.0088	**437**	394	169	4.281	5.2221	**501**	51	448
1000	10	0.01	0.0089	0.0029	**458**	401	141	0.0092	0.0052	**461**	396	143	2.5941	4.6652	**499**	87	414
1000	50	0.05	0.0010	0.0071	319	**477**	204	0.0019	0.0052	456	**441**	103	-2.6521	2.5561	457	62	**481**
1000	100	0.1	-0.0016	0.0065	298	342	**360**	-0.00290	0.0072	**387**	352	261	-2.9852	1.6322	447	66	**487**
1000	500	0.5	0.0021	0.0054	**350**	338	312	0.0028	0.0088	299	**452**	249	0.9985	1.5374	**482**	41	477
1000	1000	1	-0.0036	0.0052	336	**401**	263	-0.0040	0.0092	357	**467**	176	0.8794	1.1052	**493**	31	432

Table 2.10: Experiments with synthetically generated data sets.

2.5 Additional computational experiments

In this section we discuss additional computational experiments conducted in order to provide numerical evidence regarding some choices made in this work.

2.5.1 The estimation of empirical probabilities

During the process of estimating the data-driven ambiguity set, we must obtain two results: (i) the polyhedral convex sets and (ii) their associated probability coverage intervals. In section 2.2.2, we describe the data-driven procedure and present the approximation used to estimate the intervals. In this section, we present some results to endorse the approximation methodology usage.

Consider the outcome of the Algorithm 1 application and the coverage intervals associated to each convex hull $\mathcal{C}_i$. Ideally, the probability interval $[\underline{p}_i, \overline{p}_i]$ should include the true probability of a random point obtained from the original data distribution to fall within the convex hull $\mathcal{C}_i$, considering the specified significance level α. In this context, we refer to *accuracy* as the percentage of probability intervals $[\underline{p}_i, \overline{p}_i]$ that includes the true probability p_i. In addition, another interesting property would be how well the approximation that the outside hull $\mathcal{C}_0$ replicates the true distribution support - which we refer to as *coverage*.

To assess such properties, we ran the following experiment considering a random variable X that follows a Multivariate Normal distribution as the data-generating process:

1. Let $X \sim N(\mu, \Sigma)$, where μ is the unit vector of dimension $d = 3$ and Σ is the $d \times d$ identity matrix;

2. Let N be the sample size used to construct the ambiguity set;

3. For a given N, we generate a sample and apply Algorithm 1, obtaining the convex hulls $\{\mathcal{C}_i\}_{i=0}^{\mathcal{I}}$ and probability intervals $\{[\underline{p}_i, \overline{p}_i]\}_{i=0}^{\mathcal{I}}$;

4. For each convex hull $\mathcal{C}_i$ we approximate the true probability coverage value p_i considering the data generating process distribution by generating an extremely large sample size $S = 1.000.000$ and verify whether $p_i \in [\underline{p}_i, \overline{p}_i]$. Such interval is calculated in step 3 using the sample with size N;

5. We calculate the experiment's accuracy (percentage of estimated intervals that contain the true probability) and coverage (percentage of the distribution support within

We vary the sample size N from 10 to 10.000 and repeat the experiment 5.000 times for each value. We consider the significance level $\alpha = 10\%$. The average accuracy is presented in Figure 2.16 and the average coverage is presented in Figure 2.17.

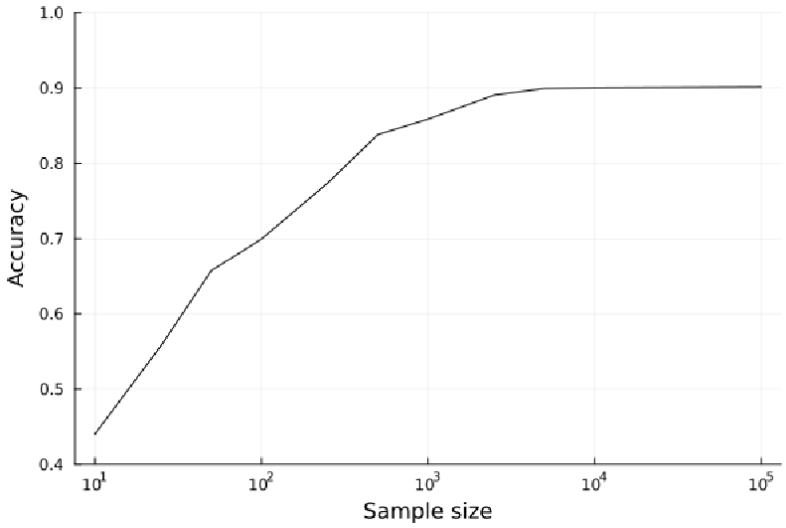


Figure 2.16: Average accuracy for a given sample size

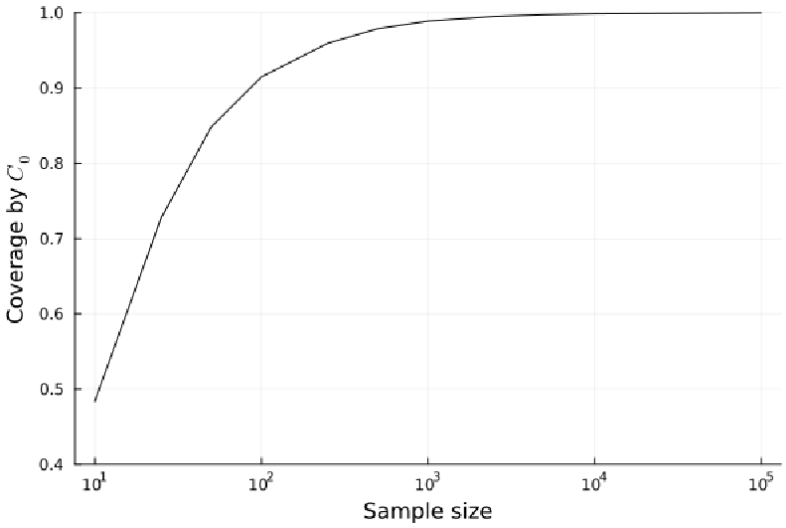


Figure 2.17: Average coverage for a given sample size

One can observe that as the sample size grows, both accuracy and coverage converge to the expected values considering the significance level of the experiment. Naturally, the quality of such approximation methodology grows as the sample size grows. However, we argue that for a relatively small

sample size, one can obtain a decent approximation. In addition, our empirical results validate the method as the experiments in sections 2.4.2 and 2.4.3 show.

2.5.2 The choice of significance level

To apply the PolieDRO framework in classification or regression models, one should define the value of the significance level α. Such value can be interpreted as the flexibility of the probability coverage of each convex hull that defines the hyperspace of distributions considered, that is, the ambiguity set. The idea is that such convex hulls imply some structure that arises from the data. The added flexibility controls the degree to which the resulting ambiguity set considers possible distributions.

Those values should be considered statistical significance parameters, using typical values such as $\alpha = 10\%$, $\alpha = 5\%$, or $\alpha = 1\%$. We repeat the experiment considering the different values and display them in tables 2.11, 2.12, 2.13, 2.14 and 2.15 for the real world data sets and in tables 2.16, 2.17 and 2.18 for the synthetic data.

In tables 2.11, 2.12, 2.13, 2.14 and 2.15, we have highlighted in bold the cases where the PolieDRO version outperformed its nominal benchmark for each value of α. We have summarized the results in table 2.19.

For the synthetic datasets, we followed the same criteria as in Section 2.4.3. We identified the highest number of wins (W), ties (T), or losses (L) for each experiment in tables 2.16, 2.17 and 2.18, and provided a summary of the results in table 2.20.

Our results indicate that the choice of the statistical parameter α has little impact on the study results. In most cases, it does not alter the performance of the PolieDRO models, and in the few cases where it does, the change is not substantial.

			Log. Regression	SVM	PolieDRO Hinge Loss		PolieDRO Logistic Loss			
Data set Name	n	p	Nominal	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$	Nominal	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
acute-inflammations-1	120	6	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
acute-inflammations-2	120	6	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9833	**1.0000**	**1.0000**	**1.0000**
balance-scale	625	4	**0.9488**	**0.9488**	0.9398	**0.9488**	**0.9632**	0.8928	0.9125	0.9333
balloons-a	20	4	0.8500	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
balloons-b	20	4	0.9000	**1.0000**	**1.0000**	**1.0000**	0.8500	**1.0000**	**1.0000**	**1.0000**
balloons-c	20	4	0.8500	**1.0000**	**1.0000**	**0.9500**	0.8500	**1.0000**	**0.9500**	**0.9000**
balloons-d	16	4	**0.6667**	**0.6667**	**0.6667**	**0.6667**	0.5333	**0.7333**	**0.7333**	**0.7333**
banknote-authentication	1372	4	0.9890	0.9883	**0.9995**	**0.9995**	**0.9898**	**0.9898**	**0.9898**	**0.9898**
blood-transfusion-service-center	748	4	0.7530	**0.7573**	**0.7573**	**0.7573**	**0.7922**	0.7920	0.7850	0.7890
breast-cancer	277	31	0.7578	**0.7614**	**0.7614**	**0.7614**	0.7392	**0.7600**	**0.7600**	**0.7600**
b-cancer-wisconsin-diagnostic	569	30	**0.9561**	0.9385	0.9415	0.9535	**0.9526**	0.9491	0.9315	0.9515
b-cancer-wisconsin-original	683	9	0.9693	0.9676	**0.9710**	**0.9705**	**0.9605**	0.9588	0.9588	0.9588
b-cancer-wisconsin-prognostic	194	32	**0.8205**	0.7692	0.7715	0.7680	0.7333	**0.7589**	**0.7589**	**0.7589**
car-evaluation	1728	15	0.9472	**0.9479**	**0.9479**	**0.9479**	**0.9576**	0.9456	09490	0.9546
climate-model-simulation-crashes	540	18	0.9500	**0.9625**	**0.9625**	**0.9625**	**0.9500**	**0.9500**	**0.9500**	**0.9500**
congressional-voting-records	232	16	0.9785	**0.9958**	**0.9958**	**0.9958**	**0.9751**	**0.9751**	**0.9751**	**0.9751**
connectionist-bench	990	10	0.9696	**0.9707**	0.9555	0.9595	**0.9545**	**0.9545**	0.9440	0.9440
connectionist-bench-sonar	208	60	**0.8195**	0.7761	0.7883	0.7991	**0.8195**	0.7238	0.7655	0.7445
contraceptive-method-choice	1473	11	0.6795	**0.6861**	**0.6861**	**0.6861**	**0.6904**	**0.6904**	**0.6904**	**0.6904**
credit-approval	690	9	**0.8656**	**0.8656**	**0.8656**	**0.8656**	**0.8398**	**0.8398**	**0.8398**	**0.8398**
dermatology	358	34	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9915	**0.9944**	**0.9944**	**0.9944**
echocardiogram	62	7	**0.7538**	0.7333	0.7333	0.7333	0.7846	**0.8000**	**0.8000**	**0.8000**
ecoli	336	7	0.9582	**0.9611**	**0.9598**	**0.9823**	0.9552	**0.9611**	**0.9611**	**0.9611**
fertility	100	12	**0.8500**	**0.8500**	**0.8500**	**0.8500**	**0.8500**	**0.8500**	**0.8500**	**0.8500**
flags	194	58	**0.8264**	0.7538	0.7538	0.7538	**0.8410**	0.7743	0.7743	0.7743

Table 2.11: Mean out of sample accuracy for different α

			Log. Regression	SVM	PolieDRO Hinge Loss			PolieDRO Logistic Loss		
Data set Name	n	p	Nominal	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$	Nominal	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
glass-identification	214	9	0.7163	**0.7249**	**0.7249**	0.7135	**0.7385**	**0.7385**	**0.7385**	**0.7385**
haberman-survival	306	3	**0.7475**	**0.7475**	**0.7475**	**0.7475**	0.7114	**0.7180**	**0.7180**	**0.7180**
hayes-roth	132	4	**0.6846**	**0.6846**	**0.6846**	**0.6846**	0.7329	**0.7555**	**0.7555**	**0.7555**
heart-disease-cleveland	297	18	0.8556	**0.8610**	**0.8610**	**0.8610**	**0.8400**	**0.8400**	**0.8400**	**0.8400**
heart-disease-hungarian	294	6	0.6947	**0.7263**	**0.7341**	**0.7155**	**0.8384**	**0.8384**	**0.8384**	**0.8384**
heart-disease-switzerland	123	6	**0.6500**	**0.6500**	**0.6500**	**0.6500**	**0.6727**	0.5272	0.5272	0.5272
heart-disease-va	200	7	0.7111	**0.7411**	**0.7522**	**0.7245**	**0.7307**	**0.7307**	**0.7307**	**0.7307**
hepatitis	155	4	**0.8000**	**0.8000**	**0.8000**	**0.8000**	**0.8250**	0.7250	0.7250	0.7250
image-segmentation	210	19	**0.9857**	0.9619	0.9515	0.9602	**0.9952**	**0.9952**	**0.9952**	**0.9952**
indian-liver-patient	583	9	**0.7258**	**0.7258**	**0.7258**	**0.7258**	0.7396	**0.7517**	**0.7620**	**0.7450**
ionosphere	351	34	**0.8514**	0.8285	0.8285	0.8355	**0.8742**	**0.8742**	**0.8742**	**0.8742**
iris	150	4	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
lenses	24	5	0.6800	**0.8000**	**0.8000**	**0.7500**	**0.7600**	0.7200	0.7200	0.7200
letter-recognition	20000	16	**0.9728**	0.9711	0.9711	0.9688	**0.9902**	0.9885	0.9885	0.9885
libras-movement	360	90	**0.9155**	0.8833	0.8833	0.9033	**0.9722**	0.9388	0.9495	0.9333
mammography-mass	830	10	0.8152	**0.8384**	**0.8384**	**0.8384**	**0.8277**	**0.8277**	**0.8277**	**0.8277**
monks-problems-1	124	11	0.7520	**0.8400**	**0.8200**	**0.8100**	**0.6880**	**0.6880**	**0.6880**	**0.6880**
monks-problems-2	169	11	**0.6470**	**0.6470**	**0.6470**	**0.6470**	0.6117	**0.6352**	**0.6470**	**0.6250**
monks-problems-3	122	11	0.8880	**0.9284**	**0.9284**	**0.9284**	0.8880	**0.9326**	**0.9104**	**0.9251**
mushroom	5644	76	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
nursery	12690	19	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
ozone-level-detection-eight	1847	72	**0.9324**	0.9268	0.9268	0.9288	0.9152	**0.9322**	**0.9322**	**0.9322**
ozone-level-detection-one	1848	72	0.9529	**0.9696**	**0.9696**	**0.9696**	0.9665	**0.9700**	**0.9700**	**0.9700**
parkinsons	195	21	0.8358	**0.8615**	**0.8555**	**0.8455**	0.8358	**0.8615**	**0.8615**	**0.8510**
plannig-relax	182	12	**0.7155**	**0.7155**	**0.7155**	**0.7155**	0.7225	0.7005	0.7005	**0.7255**

Table 2.12: Mean out of sample accuracy for different α

			Log. Regression	SVM	PolieDRO Hinge Loss			PolieDRO Logistic Loss		
Data set Name	n	p	Nominal	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$	Nominal	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
qsar-biodegradation	1055	41	0.8663	**0.8866**	**0.8866**	**0.8786**	**0.8473**	**0.8473**	**0.8473**	**0.8473**
seeds	210	7	0.9165	**0.9333**	**0.9333**	**0.9333**	0.9380	**0.9619**	**0.9619**	**0.9619**
seismic-bumps	2584	20	**0.9342**	**0.9342**	**0.9342**	**0.9342**	**0.9346**	0.9279	0.9151	0.9232
soybean-large	266	63	**0.7872**	0.7745	0.7745	0.7745	0.7625	**0.7764**	**0.7764**	**0.7764**
soybean-small	47	37	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
spambase	4601	57	**0.9265**	**0.9265**	**0.9265**	**0.9265**	**0.9230**	**0.9230**	**0.9230**	**0.9230**
statlog-project-landsat-sat	4435	36	**0.9862**	0.9846	0.9812	0.9820	**0.9833**	**0.9833**	**0.9833**	**0.9833**
teaching-assistant-evaluation	151	52	0.6800	**0.7466**	**0.7466**	**0.7466**	0.6933	**0.7000**	**0.7000**	**0.7000**
thoracic-surgery	470	16	**0.8510**	0.8446	0.8505	0.8498	**0.8808**	0.8744	0.8744	0.8744
thyroid-disease-allbp	1947	25	**0.9697**	0.9696	0.9690	0.9690	**0.9600**	0.9562	0.9562	0.9562
thyroid-disease-allhyper	1947	25	0.9794	**0.9830**	**0.9830**	**0.9830**	**0.9789**	**0.9789**	**0.9789**	**0.9789**
thyroid-disease-allrep	1947	25	**0.9778**	0.9697	0.9697	0.9697	0.9723	**0.9742**	**0.9742**	**0.9742**
thyroid-disease-sick	1947	25	0.9523	**0.9537**	0.9352	0.9445	**0.9625**	0.9475	0.9525	0.9495
tic-tac-toe-endgame	958	18	0.9842	**0.9843**	0.9842	0.9842	0.9732	**0.9801**	**0.9801**	**0.9801**
wall-following-robot-nav-2	5456	2	0.6300	0.6120	0.6230	**0.6424**	**0.6584**	0.6557	0.6557	0.6557
wall-following-robot-nav-24	5456	24	0.7543	**0.7547**	**0.7547**	**0.7547**	**0.7536**	0.7065	0.7095	0.7158
wall-following-robot-nav-4	5456	4	0.6229	**0.6381**	**0.6381**	**0.6381**	0.6139	**0.6489**	**0.6489**	**0.6489**
wine	120	6	0.9666	**0.9714**	**0.9714**	**0.9714**	0.8955	**0.9028**	**0.9008**	**0.8998**
yeast	1484	8	**0.6740**	**0.6740**	**0.6740**	**0.6740**	0.6828	**0.6996**	**0.6996**	**0.6996**
zoo	101	16	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**

Table 2.13: Mean out of sample accuracy for different α

			LASSO	PolieDRO		
Data set Name	n	p	Nominal	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
abalone	4177	9	2.1216	**2.1179**	**2.1179**	**2.1179**
airfoil-self-noise	1503	5	23.3844	**22.3516**	**22.3318**	**22.3233**
airline-costs	31	9	0.04923	**0.04281**	**0.04398**	**0.04470**
auto-mpg	392	8	**3.5618**	3.5776	3.5776	3.5776
automobile	159	31	0.3684	**0.2461**	**0.2461**	**0.2461**
beer-aroma	23	7	8.5012	**7.4046**	**7.5721**	**7.4293**
communities-and-crime	1993	100	383.8383	**383.4650**	**383.5590**	384.5262
computer-hardware	209	36	33.6611	**32.9909**	**32.9909**	**32.9909**
concrete-slump-test-compressive	103	7	3.1134	**2.89302**	**2.9821**	**2.9142**
concrete-slump-test-slump	103	7	**7.5291**	7.5673	7.5532	7.5532
construction-maintenance	33	4	**3.2313**	3.5981	3.4985	3.5361
cpu-act	8192	21	**10.4719**	**10.4719**	**10.4719**	**10.4719**
forest-fires	517	27	43.1722	**42.9497**	**42.9299**	**42.9223**
home-mortgage	18	6	18.6598	**18.1800**	**18.1800**	**18.1800**
housing	506	13	4.7872	**4.7854**	4.7901	4.7892

Table 2.14: Average out of sample MSE for different α

			LASSO	PolieDRO		
Data set Name	n	p	Nominal	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
immigrant-salaries	35	3	**1.7360**	**1.7360**	**1.7360**	**1.7360**
japan-emigration	45	5	**150.9830**	168.6030	170.5555	169.0058
kin8nm	8192	8	**0.0413**	**0.0413**	**0.0413**	**0.0413**
lpga-2208	157	6	0.4323	**0.4291**	0.4343	**0.4310**
lpga-2009	146	11	0.5037	**0.4676**	**0.4676**	**0.4676**
parkinsons-telemonitoring-motor	5875	16	7.7964	**7.7778**	**7.7892**	**7.7745**
parkinsons-telemonitoring-total	5875	16	10.3043	10.3067	**10.2997**	**10.3013**
pyrim	74	27	**0.1434**	**0.1434**	**0.1434**	**0.1434**
texas-jan-temp	16	3	1.27907	**1.2152**	**1.2513**	**1.2494**
triazines	186	60	**0.1330**	0.1368	0.1355	0.1359
tv-sales	31	8	3028.6666	**3019.7855**	**3016.5532**	**3018.3151**
wiki4he	435	53	**6.4113**	6.6882	6.6882	6.6882
wine-quality-red	1599	11	0.6520	**0.6169**	**0.6344**	**0.6215**
wine-quality-white	4898	11	**0.7650**	0.7710	0.7690	0.7854
yatch-hydrodynamics	308	6	9.1323	9.2546	9.2113	**9.1274**

Table 2.15: Average out of sample MSE for different α

			Classification – Hinge Loss					Classification – Logistic Loss					Regression – MSE Loss				
d	n	n/d	Mean	SD.	W	T	L	Mean	SD.	W	T	L	Mean	SD.	W	T	L
2	5	2.5	0.0164	0.0364	**621**	298	81	0.0152	0.0098	**607**	302	91	0.1525	0.2322	**592**	12	396
2	10	5	0.0324	0.0854	**696**	258	46	0.0212	0.0539	**650**	238	112	0.1233	0.1785	**527**	43	430
2	50	25	0.0185	0.0065	326	**482**	192	0.0177	0.0082	347	**475**	178	0.1852	0.1652	**612**	51	337
2	100	50	0.0025	0.0031	**422**	416	162	0.0037	0.0101	**427**	413	154	0.1109	0.1361	**609**	37	354
2	500	250	-0.0036	0.0223	295	307	**398**	-0.0036	0.0223	281	323	**395**	0.0995	0.0562	**482**	50	468
2	1000	500	0.0094	0.0581	276	**482**	222	0.0099	0.0311	271	**475**	228	-0.0152	0.0328	450	51	**499**
10	5	0.5	0.0012	0.0027	**511**	387	102	0.0018	0.0035	**526**	382	92	0.2122	0.2366	**483**	106	411
10	10	1	0.0038	0.0064	**395**	375	230	0.0046	0.0094	**385**	350	265	0.2852	0.2366	**412**	189	399
10	50	5	0.0084	0.0069	**491**	400	109	0.0091	0.0124	405	**501**	94	0.1349	0.1741	**592**	164	244
10	100	10	-0.0095	0.0014	295	**480**	225	-0.0098	0.0017	300	**460**	240	-0.0985	0.2006	392	196	**412**
10	500	50	-0.0014	0.0051	231	**515**	254	-0.0024	0.0066	231	**480**	289	-0.1851	0.2011	397	112	**491**
10	1000	100	0.0067	0.0021	310	312	**378**	0.0067	0.0021	300	307	**393**	0.1067	0.1124	**450**	211	339
100	5	0.05	0.0055	0.0025	**396**	384	220	0.0065	0.0022	**390**	374	236	1.2511	1.3351	**527**	137	336
100	10	0.1	0.0002	0.0033	203	**517**	180	0.0008	0.0038	198	**515**	187	1.4412	1.5112	**572**	95	333
100	50	0.5	0.0082	0.0035	**415**	298	287	0.0078	0.0045	**418**	296	286	0.9714	0.9416	**528**	121	351
100	100	1	0.0009	0.0065	263	**399**	338	0.0012	0.0088	250	**412**	338	0.5651	0.7528	439	40	**521**
100	500	5	-0.0017	0.0062	271	270	**359**	-0.0021	0.0081	261	275	**364**	0.3229	0.3281	484	13	**503**
100	1000	10	0.0021	0.0092	231	**488**	281	0.0021	0.0092	226	**499**	275	0.1145	0.1285	**381**	272	347
1000	5	0.005	0.0033	0.0072	**427**	399	174	0.0073	0.0088	**437**	394	169	4.281	5.2221	**501**	51	448
1000	10	0.01	0.0089	0.0029	**458**	401	141	0.0092	0.0052	**461**	396	143	2.5941	4.6652	**499**	87	414
1000	50	0.05	0.0010	0.0071	319	**477**	204	0.0019	0.0052	456	**441**	103	-2.6521	2.5561	457	62	**481**
1000	100	0.1	-0.0016	0.0065	298	342	**360**	-0.00290	0.0072	**387**	352	261	-2.9852	1.6322	447	66	**487**
1000	500	0.5	0.0021	0.0054	**350**	338	312	0.0028	0.0088	299	**452**	249	0.9985	1.5374	**482**	41	477
1000	1000	1	-0.0036	0.0052	336	**401**	263	-0.0040	0.0092	357	**467**	176	0.8794	1.1052	**493**	31	432

Table 2.16: Experiments with synthetically generated data sets, $\alpha = 0.10$

			Classification – Hinge Loss					Classification – Logistic Loss					Regression – MSE Loss				
d	n	n/d	Mean	SD.	W	T	L	Mean	SD.	W	T	L	Mean	SD.	W	T	L
2	5	2.5	0.0174	0.0304	**621**	298	81	0.0152	0.0098	**607**	302	91	0.1525	0.2322	**592**	12	396
2	10	5	0.0454	0.0812	**696**	258	46	0.0212	0.0539	**650**	238	112	0.1233	0.1785	**527**	43	430
2	50	25	0.0212	0.0069	320	**485**	195	0.0157	0.0087	347	**475**	178	0.1892	0.1622	**612**	51	337
2	100	50	0.0052	0.0039	412	**426**	162	0.0042	0.0109	**427**	413	160	0.1109	0.1321	**612**	35	353
2	500	250	-0.0046	0.0227	290	312	**398**	-0.0036	0.0224	281	323	**395**	0.0995	0.0522	**480**	50	470
2	1000	500	0.0104	0.0541	276	**482**	222	0.0099	0.0311	271	**475**	228	-0.0152	0.0328	450	51	**499**
10	5	0.5	0.0013	0.0097	**518**	392	90	0.0028	0.0038	**526**	382	92	0.2122	0.2366	**483**	106	411
10	10	1	0.0035	0.0024	**390**	365	245	0.0046	0.0099	**388**	345	267	0.2852	0.2366	**412**	189	399
10	50	5	0.0095	0.0055	**495**	396	109	0.0092	0.0127	405	**501**	94	0.1349	0.1741	**592**	164	244
10	100	10	-0.0094	0.0019	295	**480**	225	-0.0098	0.0017	300	**460**	240	-0.0985	0.2006	392	196	**412**
10	500	50	-0.0007	0.0052	234	**513**	251	-0.0029	0.0062	229	**484**	287	-0.1851	0.2011	397	112	**491**
10	1000	100	0.0077	0.0028	310	310	**390**	0.0067	0.0024	300	307	**393**	0.1067	0.1124	**450**	211	339
100	5	0.05	0.0095	0.0029	**396**	384	220	0.0065	0.0022	**390**	374	236	1.2511	1.3351	**527**	137	336
100	10	0.1	0.0090	0.0034	203	**517**	180	0.0008	0.0038	198	**515**	187	1.4412	1.5112	**572**	95	333
100	50	0.5	0.0089	0.0056	**415**	298	287	0.0028	0.0035	**414**	300	286	0.9714	0.9416	**528**	121	351
100	100	1	0.0018	0.0062	263	**399**	338	0.0012	0.0088	250	**412**	338	0.5651	0.7528	439	40	**521**
100	500	5	-0.0035	0.0056	271	270	**359**	-0.0021	0.0081	261	275	**364**	0.3229	0.3281	484	13	**503**
100	1000	10	0.0023	0.0096	231	**488**	281	0.0021	0.0092	226	**499**	275	0.1145	0.1285	**381**	272	347
1000	5	0.005	0.0033	0.0044	**427**	399	174	0.0073	0.0088	**437**	394	169	4.281	5.2221	**501**	51	448
1000	10	0.01	0.0090	0.0031	**458**	401	141	0.0092	0.0052	**461**	396	143	2.5941	4.6652	**499**	87	414
1000	50	0.05	0.0019	0.0081	319	**477**	204	0.0039	0.0052	456	**441**	103	-2.2365	2.5861	459	60	**481**
1000	100	0.1	-0.0018	0.0055	298	342	**360**	-0.00290	0.0072	**387**	352	261	-2.9852	1.6322	447	66	**487**
1000	500	0.5	0.0023	0.0049	**350**	338	312	0.0028	0.0088	299	**452**	249	0.9985	1.5374	**482**	41	477
1000	1000	1	-0.0039	0.0057	336	**401**	263	-0.0040	0.0092	357	**467**	176	0.8794	1.1052	**493**	31	432

Table 2.17: Experiments with synthetically generated data sets, $\alpha = 0.05$

			Classification – Hinge Loss					Classification – Logistic Loss					Regression – MSE Loss				
d	n	n/d	Mean	SD.	W	T	L	Mean	SD.	W	T	L	Mean	SD.	W	T	L
2	5	2.5	0.0168	0.0774	**611**	305	84	0.0158	0.0096	**610**	297	93	0.1525	0.2922	**592**	12	396
2	10	5	0.0344	0.0559	**689**	262	49	0.0212	0.0539	**650**	238	112	0.1237	0.1765	**527**	43	430
2	50	25	0.0112	0.0069	320	**485**	195	0.0157	0.0087	347	**475**	178	0.1892	0.1622	**612**	51	337
2	100	50	0.0052	0.0039	412	**426**	162	0.0042	0.0109	**427**	413	160	0.1109	0.1321	**612**	35	353
2	500	250	-0.0046	0.0227	290	312	**398**	-0.0036	0.0224	281	323	**395**	0.0995	0.0522	**480**	50	470
2	1000	500	0.0093	0.0531	280	**480**	240	0.0099	0.0319	271	**475**	228	-0.0152	0.0358	451	50	**499**
10	5	0.5	0.0013	0.0097	**518**	392	90	0.0028	0.0038	**526**	382	92	0.2122	0.2366	**483**	106	411
10	10	1	0.0035	0.0024	**390**	365	245	0.0046	0.0099	**388**	345	267	0.2852	0.2366	**412**	189	399
10	50	5	0.0095	0.0055	**495**	396	109	0.0092	0.0127	405	**501**	94	0.1349	0.1741	**592**	164	244
10	100	10	-0.0125	0.0042	292	**484**	223	-0.0038	0.0012	297	**468**	235	-0.0985	0.2006	392	196	**412**
10	500	50	-0.0007	0.0052	234	**513**	251	-0.0029	0.0062	229	**484**	287	-0.1851	0.2011	397	112	**491**
10	1000	100	0.0077	0.0028	310	310	**390**	0.0067	0.0024	300	307	**393**	0.1067	0.1124	**450**	211	339
100	5	0.05	0.0074	0.0021	**396**	384	220	0.0065	0.0022	**398**	370	232	1.2511	1.3351	**527**	137	336
100	10	0.1	0.0013	0.0044	203	**517**	180	0.028	0.0039	198	**525**	177	1.4412	1.5112	**572**	95	333
100	50	0.5	0.0089	0.0056	**415**	298	287	0.0028	0.0035	**414**	300	286	0.9714	0.9416	**528**	121	351
100	100	1	0.0039	0.0068	263	**399**	338	0.0212	0.0038	250	**412**	338	0.5651	0.7528	439	40	**521**
100	500	5	-0.0029	0.0062	271	270	**359**	-0.0022	0.0011	274	262	**364**	0.3229	0.3281	484	13	**503**
100	1000	10	0.0028	0.0099	231	**488**	281	0.0029	0.0102	225	**500**	275	0.1145	0.1285	**381**	272	347
1000	5	0.005	0.0039	0.0078	**427**	399	174	0.0063	0.0089	**438**	393	169	4.521	5.6321	**504**	44	452
1000	10	0.01	0.0091	0.0070	**458**	401	141	0.0049	0.0051	**461**	396	143	2.5541	4.6692	**484**	92	424
1000	50	0.05	0.0019	0.0081	319	**477**	204	0.0039	0.0052	456	**441**	103	-2.2365	2.5861	459	60	**481**
1000	100	0.1	-0.0021	0.0031	298	342	**360**	-0.00360	0.0072	367	**372**	261	-3.0032	1.6572	454	56	**490**
1000	500	0.5	0.0029	0.0024	**350**	338	312	0.0069	0.0082	299	**452**	249	0.9756	1.0246	**477**	51	472
1000	1000	1	-0.0039	0.0072	336	**401**	263	-0.0043	0.0098	357	**468**	177	0.9026	1.10311	**494**	30	432

Table 2.18: Experiments with synthetically generated data sets, $\alpha = 0.01$

α	Loss Function	Metric	Wins	Ties	Losses	Total
0.10	Hinge Loss	Accuracy	29	22	19	70
	Logistic Loss	Accuracy	27	27	16	70
	MSE Loss	RMSE	17	4	9	30
0.05	Hinge Loss	Accuracy	29	22	19	70
	Logistic Loss	Accuracy	25	26	19	70
	MSE Loss	RMSE	16	4	10	30
0.01	Hinge Loss	Accuracy	29	23	18	70
	Logistic Loss	Accuracy	25	26	19	70
	MSE Loss	RMSE	17	4	9	30

Table 2.19: Pairwise performance of the PolieDRO models against their nominal benchmarks using real-world data sets, for varying α

α	Loss Function	Metric	Wins	Ties	Losses	Total
0.10	Hinge Loss	Accuracy	11	9	4	24
	Logistic Loss	Accuracy	10	11	3	24
	MSE Loss	RMSE	17	0	7	24
0.05	Hinge Loss	Accuracy	10	10	4	24
	Logistic Loss	Accuracy	10	11	3	24
	MSE Loss	RMSE	17	0	7	24
0.01	Hinge Loss	Accuracy	10	10	4	24
	Logistic Loss	Accuracy	9	12	3	24
	MSE Loss	RMSE	17	0	7	24

Table 2.20: Pairwise performance of the PolieDRO models against their nominal benchmarks using synthetic data sets, for varying α

2.6 Conclusions

In this chapter, we develop the predictive PolieDRO, a novel analytics framework based on a data-driven DRO formulation that does not rely on the calibration of regularization hyperparameters. Our framework results in a computationally tractable DRO formulation with practical applications, given a convex loss function. At the core of our proposed methodology, there is a new and iterative procedure to construct data-driven convex hulls that define the ambiguity set along with probability estimates. We go beyond simpler data-driven procedures that only define the first moments of the distribution and we use the available data to define the whole shape of the distribution. Such a procedure is based on well-known algorithms with efficient off-the-shelf implementations.

By exploring a bridge between the realms of Machine Learning and Distributionally Robust Optimization, our new proposed framework results in intuitive methods for prediction tasks, namely classification, and regression. Besides its theoretical soundness, we evaluate the performance of such ideas in an extensive numerical experiment with 100 real-world data sets. The findings of such a broad experiment suggest that the resulting formulations of the PolieDRO framework are competitive with common methods and their variations that are known in the literature and practice. Our experiments yielded an equal or better result in 72.8% of cases for the hinge loss, 77.14% for the logistic loss, and 60.0% for the MSE loss.

In particular, our findings indicate a superior performance in so-called harder problems, validating the DRO approach to the tasks of classification and regression: 90.9% of cases for the hinge loss, 77.27% for the logistic loss, and 73.3% for the MSE loss.

In addition, we explored the performance of the PolieDRO framework under a synthetically generated experiment aimed at measuring the impact of the ratio between the number of features and the number of samples available. Based on a computationally intensive study, we achieved consistent results in favor of the PolieDRO variations of the loss functions, indicating once again its superior empirical evidence. On top of that, such competitive performance is accompanied by the absence of a critical (and often criticized) step in an ML experiment: the hyperparameterization step.

We believe that this work highlights the potential for DRO researchers to contribute to the ML literature. By proposing new designs for the ambiguity sets—ideally in a hyperparameter-free way—new frameworks can be developed, resulting in novel and computationally tractable approaches to tasks handled by the ML community. In addition, our work reinforces the importance of having data-driven ambiguity sets that can model the shape of plausible distributions more accurately, as opposed to simply requiring them to be in a ball close to the nominal distribution. Moreover, the size of the "ball" is a hyperparameter that usually has to be set by cross-validation to generate empirically accurate models.

Future work includes embedding more advanced methods within our framework such as CART, ensembles and deep learning architectures, as well as Generative Adversarial Networks.

3
Prescriptive PolieDRO – Portfolio Optimization

One of the paramount challenges faced by financial firms is the strategic allocation of their budgets across a diverse range of investment assets to maximize company wealth, as outlined in (KOLM; TÜTÜNCÜ; FABOZZIC, 2014). Traditionally, this critical task was carried out through discretionary trading, where decisions on portfolio composition were largely influenced by the insights and judgments of experienced managers. However, recent years have witnessed a significant pivot towards quantitative trading, propelled by substantial advancements in mathematical modeling and computational tools, alongside an increase in processing power and capabilities (SPIERS; WALLEZ, 2010). Roughly speaking, the latter investment technique is based on signals produced by a computer program or investment model, with scarcely any intervention from the portfolio manager. Following this upward trending, this work focus on devising a quantitative-based methodology to identify a wealth-maximizing portfolio allocation.

The concept of quantitative portfolio allocation has been extensively explored since Markowitz's (1952) seminal work (MARKOWITZ, 1952), which introduced the trade-off between high expected returns and controlled risk exposure. Markowitz advocated for an "efficient allocation" that harmonizes the portfolio's expected return and its variance, setting a foundation for the Mean-Variance analysis. Subsequent research challenged the use of variance as the sole risk measure, leading to the exploration of semi-variance and other risk metrics that better capture an investor's risk aversion ((MAO, 1970; CHOOBINEH; BRANTING, 1986; MARKOWITZ et al., 1993) and (HUANG, 2008)). This body of work expanded the analytical framework to include not just expected value and semi-variance but also the impacts of higher moments like skewness and kurtosis, as well as performance-based regularization metrics on portfolio allocation ((CVITANIć; POLIMENIS; ZAPATERO, 2008; HARVEY et al., 2010) and (LI; QIN; KARC, 2010)).

In the quest for more effective risk management strategies, the 1990s saw the emergence of distribution quantile-based risk measures, offering an alternative to moment-based metrics. The Value-at-Risk (VaR), popularized by the Basel II accord in 1999, became a pivotal tool in the industry, providing a quantifiable measure of the potential financial losses at a specific confidence level (we refer to (JORION, 2006) for a thorough discussion and applications of this measure in the financial industry). Roughly speaking, VaR was designed

to quantify the lower-tail quantile of the reward probability distribution associated with a pre-specified confidence level, thus gauging the exposure of the financial company and indicating the amount of capital that is needed to fully cover possible losses. Despite its popularity in risk management, VaR-based portfolio allocation models face several shortcomings, both from a technical and computational viewpoints. On the one hand, the Value-at-Risk lacks of convexity, and its ability to quantify extremely hazardous events are limited. On the other hand, its incorporation into optimization problems drastically increase the computational complexity. Hence, to cope with these issues, specially the technical ones, an axiomatic approach were pursuit in order to define desirable properties for a quantile-based risk measure, aiming at better characterizing the agent's behavior towards risk (ARTZNER et al., 1999; GIORGI, 2005; FOLLMER; SCHIED, 2011). As a byproduct of this axiomatization, a novel risk measure, named the Conditional Value-at-Risk (CVaR), came up as a promising tool to devise desirable risk-averse portfolio allocations. In a few words, the CVaR quantifies the average of the worst-valued rewards of a portfolio up to its VaR level and cover all the Value-at-Risk aforementioned shortcomings (PFLUG, 2000; STREET, 2010; ROCKAFELLAR; URYASEV, 2002).

A critical aspect of implementing portfolios based on moment- or quantile-based risk measures is their reliance on accurately specifying the joint probability distribution of asset returns. An imprecise probabilistic model can lead to ineffective decision-making, exposing investors to unexpected risks, a concern highlighted by (SHAPIRO; NEMIROVSKI, 2005). In the particular context of this work, this issue is of great significance since a decent characterization of the assets return uncertainty is still a challenging task[1]. In fact, there is an strong empirical evidence against fitting a single candidate distribution to the available information (e.g., historical data) and using such unique probabilistic characterization to define risk-constrained portfolio allocations due to a poor out-of-sample performance (see, for instance, (MICHAUD, 1989) and the so-called error maximization effect).

Despite considerable efforts by both academics and practitioners, accurately forecasting asset return dynamics remains unresolved. In response, robust optimization has gained popularity for its ability to manage uncertainty through constraints that enable worst-case scenario analysis, offering a way to identify resilient portfolio allocations ((FABOZZI; HUANG; ZHOU, 2010; LIM; SHANTHIKUMAR; VAHN, 2012; KIM; KIM; FABOZZI, 2014;

[1]We refer to the review work done by (RESCHENHOFER et al., 2020) for an extensive analysis.

BAN; KAROUI; LIM, 2018; JIN; LUO; ZENG, 2021)). From a modeling viewpoint, its main advantage relies on the capability of the manager to characterize the assets returns uncertainty by means of a set of constraints and perform a worst-case analysis, thus identifying a feasible portfolio allocation that is robust against the most unfavorable scenario within this set (BENTAL; GHAOUI; NEMIROVSKI, 2009b). Nevertheless, a recurrent criticism over robust-optimization-based models is its high level of conservatism since the robust approach generally assimilate little available information regarding the uncertainty probabilistic nature.

In order to extract the benefits from both robust- and distributionally-based methodologies, a decision modeling structure, known as distributionally robust optimization (DRO), has recently emerged as a powerful tool for addressing general decision-making problems in uncertain environments (BERTSIMAS; SIM; ZHANG, 2019; PARYS; ESFAHANI; KUHN, 2021). Generally speaking, DRO-based models make use of a probabilistic description of the uncertain factors, but recognizes that a precise characterization is of difficult assessment, thus employing a worst-case analysis over a set of "credible" probability distributions – commonly referred to as a "Distributional Uncertainty Set" (DUS). The major challenge for practical implementation of such decision-modeling methodologies is precisely the appropriate design of the DUS. It should encompass reliable information regarding the true probability distribution, while keeping computational tractability. In the past years, many arrangement ideas have been put forward in technical literature. For instance, (SCARF, 1958; BERTSIMAS; POPESCU, 2005) and (LOTFI; ZENIOS, 2018) proposed to describe the DUS as the set of all probability distributions with a pre-specified mean and covariance matrix. Additionally, (DELAGE; YE, 2010) extended this formulation in order to consider a confidence interval for these distribution moments and discussed a procedure to estimate such interval based on historical data. Following a similar path, (WIESEMANN; KUHN; SIM, 2014) conceived a general format for DUSs which comprises distributions with conic-representable confidence sets and mean within an affine manifold. More recently, (BERTSIMAS; GUPTA; KALLUS, 2018) devised a data-driven scheme to design computationally tractable uncertainty sets from several statistical hypothesis tests. Furthermore, (PFLUG; WOZABAL, 2007) make use of the Wasserstein metric to design the DUS as a ball centered at the empirical distribution within a portfolio-allocation model and (ESFAHANI; KUHN, 2018) studied the general properties of DRO problems built using the Wasserstein ball and its computational tractability. Finally, (BERTSIMAS; SIM; ZHANG, 2019) developed a framework for solving adaptive distribution-

ally robust linear optimization problems with a Second-Order Conic (SOC) distributional uncertainty sets.

Despite the advancements in decision-making models, early research on Distributionally Robust Optimization (DRO) often resorts to somewhat arbitrary choices in defining the Distributional Uncertainty Set (DUS). These choices include the criteria for distinguishing between distributions in distance-based sets or specifying statistical properties like moments or independence for moment-based sets. Such decisions can be particularly challenging to assess accurately, especially in the context of portfolio allocation. This study aims to introduce an innovative approach for constructing data-driven DUSs tailored for distributionally robust portfolio allocation challenges. By leveraging historical data, we develop a series of nested convex hulls, aligning with the uncertainty set framework as conceptualized by (WIESEMANN; KUHN; SIM, 2014). A key part of our methodology involves establishing a confidence interval for the probability measures of these convex hulls, ensuring that the dynamics of asset returns are captured endogenously and non-parametrically, directly from the data, thereby reducing the risk of model misspecification.

Our methodology's standout feature is its simplicity and lack of reliance on parametric assumptions, allowing for a direct learning process from the data to understand the probabilistic behavior of asset returns. This approach not only provides a nuanced characterization of the uncertainty but also avoids the pitfalls of predefined structural assumptions. To address the potential non-convexity in the resulting optimization problem, we offer an efficient solution strategy that converts the complex problem into a manageable, single-level formulation. This adjustment makes the problem compatible with existing mathematical programming techniques or straightforward application in standard software solutions. The contributions of this research are twofold:

1. We propose a cutting-edge method for designing data-driven DUSs for use in distributionally robust portfolio allocation. This method includes optimizing the expected value of the portfolio while incorporating Conditional Value-at-Risk (CVaR) for risk assessment. The approach utilizes nested convex hulls derived from historical data, integrated within the general uncertainty set structure as suggested by (WIESEMANN; KUHN; SIM, 2014), alongside a novel procedure for calculating the confidence interval for each convex hull's probability measure.

2. We introduce a single-level equivalent reformulation that simplifies the computational complexity to depend merely on the size of the historical data used. Through exploiting the unique properties of our distributional

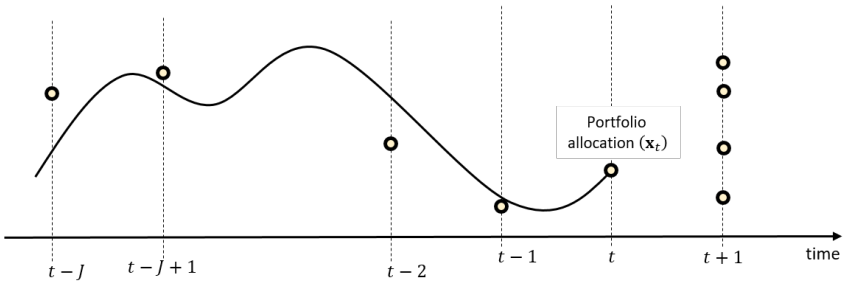


Figure 3.1: Manager decision structure at a given time frame t

uncertainty set construction, we streamline the portfolio allocation problem into a form that is both practical and efficient for high-frequency trading applications, thanks to its reduced computational demands.

Notation. In the remainder of this chapter, n-dimensional vectors will be represented by lower-case bold symbols, e.g., $\boldsymbol{c} = [c_1, \ldots, c_n]^\top$. Additionally, following the standard practice for decision under uncertainty modeling, all random variables will be defined from a measurable space $(\Omega, \mathcal{F})$ to a compact set and represented with a tilde symbol (e.g., $\tilde{r}$). We also denote by $\mathcal{M}_+(\mathbb{R}^n)$ and $\mathcal{P}_0(\mathbb{R}^n)$ the spaces of non-negative measures and probability functions on $\mathbb{R}^n$, respectively, and by $H(\mathcal{Z})$ the convex hull of a set of points $\mathcal{Z} = \{z_1, \ldots, z_n\}$.

For the sake of illustration, Figure 3.1 depicts the framework and manager decision structure at a given time frame t.

3.1 Portfolio Allocation under Uncertainty

The main objective of this work is to propose a novel methodology to design data-driven distributional uncertainty sets for distributionally robust portfolio allocation problems. For expository purposes, the decision structure here assumed follows a standard one-period model, in which, at a given time frame t, the portfolio manager should define its strategy aiming at maximizing the company's wealth at $t+1$, without complete knowledge regarding the asset price dynamics between both time frames. More precisely, we consider available for investment a set of $\mathcal{N} = \{1, \ldots, n\}$ assets, whose returns over the capital invested from t to $t+1$ are uncertain and represented by a random vector $\tilde{\boldsymbol{r}}_t \triangleq [\tilde{r}_{1,t}, \cdots, \tilde{r}_{n,t}]^\top$ defined on a compact set $\Xi_t \subset \mathbb{R}^n$. A collection of J sequentially observed past data $\mathcal{R}_{t,J} \triangleq [\hat{\boldsymbol{r}}_{t-J}, \cdots, \hat{\boldsymbol{r}}_{t-1}]$ up to time frame t is also assumed available. In this context, the manager must define a financial

allocation of its current budget — $\mathbf{x}_t \triangleq [x_{1,t}, \ldots, x_{n,t}]^\top$, that maximizes the company's wealth at the beginning of $t+1$.

3.1.1 Ambiguity-Constrained Portfolio Allocation

In practical applications, the set of feasible allocations is generally assembled as a byproduct of two main guidelines. From a managerial viewpoint, on the one hand, a collection of constraints are commonly imposed to the quantitative model aiming at shaping the set of feasible allocations to comprise the manager's so-called "operational region". Such constraints are *a priori* defined by the decision maker and may consist of budget, leverage, operational costs, short-sale or portfolio norm restrictions (DEMIGUEL et al., 2009; BROWN; SMITH, 2011; FERNANDES et al., 2016). On the other hand, from a risk management point-of-view, the concept of acceptability functionals and acceptance sets are typically adapted into the portfolio allocation model (PFLUG; WOZABAL, 2007; FOLLMER; SCHIED, 2011). Roughly speaking, the acceptance set comprises only those allocations whose risk level, measured by an acceptability functional, is bearable.

Formally, for a given time frame t, let $\mathcal{X}_t \subseteq \mathbb{R}^n$ to denote the *a priori* defined "operational region" and $\phi : \mathcal{X}_t \times \Xi_t \to \mathbb{R}$ the company's wealth at the beginning of $t+1$ for a given portfolio allocation $\mathbf{x}_t \in \mathcal{X}_t$ under an uncertain return over capital invested $\tilde{\boldsymbol{r}}_t \in \Xi_t$, i.e.,

$$\phi\Big(\mathbf{x}_t, \tilde{\boldsymbol{r}}_t\Big) = (\mathbf{1} + \tilde{\boldsymbol{r}}_t)^\top \mathbf{x}_t. \tag{3-1}$$

In order to characterize the acceptance set, in this work, we resort to the widely used α-percentile acceptability functional, the Conditional Value-at-Risk (ROCKAFELLAR; URYASEV, 2002; STREET, 2010). A key challenge, however, for the practical implementation of such measure is its intrinsic dependence upon a well-defined probability function that drives the future dynamics of the uncertain factors. For the particular context of this work, this representation is of difficult assessment from all available information (e.g., $\mathcal{R}_{t,J}$), thus most portfolio allocation decisions are made under ambiguity (PFLUG; WOZABAL, 2007; RESCHENHOFER et al., 2020). Hence, in order to cope with this modeling issue, and accommodate the uncertainty on a precise probabilistic specification, we take into account a distributional (uncertainty) set $\mathcal{P}$, with probability functions defined over Ξ_t, and extend the concept of a CVaR-based acceptance set to ensure acceptability for all probability

functionals within this pre-defined distributional set as follows:

$$\mathcal{A}_t = \left\{ \mathbf{x}_t \in \mathcal{X}_t \;\middle|\; \mathrm{CVaR}_{\alpha}^{(\mathbb{P})}\Big(\phi\Big(\mathbf{x}_t, \tilde{\boldsymbol{r}}_t\Big)\Big) \geq \Gamma_t, \quad \forall\, \mathbb{P} \in \mathcal{P}\Big(\mathcal{R}_{t,J}\Big) \right\}. \tag{3-2}$$

In (3-2), the functional $\mathrm{CVaR}_{\alpha}^{(\mathbb{P})}$ is evaluated with respect to the probability function $\mathbb{P}$ and Γ_t represents a minimum level of wealth desired by the manager at the beginning of the next time frame. For didactic purposes, we explicitly identify the distributional uncertainty set $\mathcal{P}$ with the collection of historical data $\mathcal{R}_{t,J}$, wherein J indicates the "size" of window chosen by the manager to design the DUS. We highlight that the acceptance set $\mathcal{A}_t$ captures both the manager's risk aversion, by means of the CVaR functional, and its aversion to ambiguity through the imposition of acceptance over all probability functions within the distributional uncertainty set $\mathcal{P}$.

In this context, following a Markowitz decision structure, at a given time frame t, the manager's problem resumes to identify, within all acceptable allocations $\mathbf{x}_t \in \mathcal{A}_t$, the one that maximize its expected wealth at the beginning of $t+1$,

$$\mathbf{x}_t^* \in \underset{\mathbf{x}_t \in \mathcal{A}_t}{\arg\max} \left\{ \inf_{\mathbb{Q} \in \mathcal{P}(\mathcal{R}_{t,J})} \mathbb{E}^{(\mathbb{Q})}\Big[\phi\Big(\mathbf{x}_t, \tilde{\boldsymbol{r}}_t\Big)\Big] \right\}. \tag{3-3}$$

Note that, similarly to (3-2), the expected wealth also requires a probabilistic representation for the future dynamics of the uncertain factors, thus also of difficult assessment from available information. Therefore, aiming at ensuring robustness on the portfolio allocation, a worst-case functional over the distributional set $\mathcal{P}(\mathcal{R}_{t,J})$ is applied in (3-3). In the next section, the methodology proposed in this work to design the data-driven distributional sets for the portfolio allocation model is presented in detail.

3.2 Data-Driven Distributional Uncertainty Set

A crucial aspect for a practical implementation of the distributionally robust portfolio allocation model (3-3) is an adequate design of the distributional uncertainty set $\mathcal{P}$. Generally speaking, the main ideas discussed on technical literature are based on distributional moments or distance between probability functions. We argue, however, that these design methods largely rely on an appropriate specification of a set of parameters, for which a poor choice may lead to low quality or over-conservative solutions (SHAPIRO, 2017). Furthermore, for the particular context of this work, an appropriate estimation of

the standard moment- and distance-based DUS parameters based on available information is a difficult task. In this context, the main goal of this section is, thus, to describe an alternative methodology to efficiently design and construct data-driven, non-parametric, distributional uncertainty sets for the portfolio allocation problem (3-3).

The topological structure of the distributional set considered in this work is based on the canonical form developed by (WIESEMANN; KUHN; SIM, 2014). More precisely, for a given time frame t and a training dataset $\mathcal{R}_{t,J}$ of size J, let $\mathcal{C}_i(\mathcal{R}_{t,J}) \subseteq \Xi_t,\ \forall\ i \in \mathcal{I} \triangleq \{1,\ldots,I\}$ be a collection of the so-called "confidence subsets" of the support set Ξ_t. The concept behind the distributional uncertainty set is, thus, to take into account all probability functions $\mathbb{P} \in \mathcal{P}_0(\mathbb{R}^n)$ whose measure of each set $\mathcal{C}_i(\cdot)$ matches their "true" value $p_{i,t}$. It should be highlighted, however, that a precise specification of each $\{p_{i,t}\}_{i\in\mathcal{I}}$ is of difficult assessment. Therefore, in order to cope with this issue, we soften this restriction and comprise a coverage probability interval for each $\{p_{i,t}\}_{i\in\mathcal{I}}$. Formally, the distributional uncertainty set considered in this work is given by

$$\mathcal{P}\Big(\mathcal{R}_{t,J}\Big) = \left\{ \mathbb{P} \in \mathcal{P}_0(\mathbb{R}^n) \;\middle|\; \mathbb{P}\Big(\tilde{\boldsymbol{r}}_t \in \mathcal{C}_i\Big(\mathcal{R}_{t,J}\Big)\Big) \in \Big[\underline{p}_{i,t}, \overline{p}_{i,t}\Big],\ \ \forall\ i \in \mathcal{I} \right\}, \quad \text{(3-4)}$$

where $\overline{\boldsymbol{p}}_t = [\overline{p}_{1,t},\ldots,\overline{p}_{I,t}]^\top$ and $\underline{\boldsymbol{p}}_t = [\underline{p}_{1,t},\ldots,\underline{p}_{I,t}]^\top$ defines the boundaries of the coverage probability interval for $p_{i,t}$, with $\underline{\boldsymbol{p}}_t \leq \overline{\boldsymbol{p}}_t$.

The methodology proposed in this work to assemble the distributional set (3-4) involves the construction of the confidence subsets $\Big\{\mathcal{C}_i(\cdot)\Big\}_{i\in\mathcal{I}}$ as a sequence of nested convex hulls and the coverage probability from confidence intervals of proportion measures, both wholly extracted from the given historical dataset $\mathcal{R}_{t,J}$. More precisely, the proposed procedure starts by identifying the extreme points $\mathcal{E}_{1,t,J}$ that define the convex hull of $\mathcal{R}_{t,J}$ — i.e., $\mathcal{E}_{1,t,J} \triangleq \Pi(\mathcal{R}_{t,J})$; and assign $\mathcal{C}_1(\mathcal{R}_{t,J}) = H(\mathcal{R}_{t,J})$. From a modeling perspective, we assume that $H(\mathcal{R}_{t,J})$ covers the support set Ξ_t, thus we attribute $\overline{p}_{1,t} = \underline{p}_{1,t} = 1$. Then, since from convex theory $\mathcal{E}_{1,t,J} \subseteq \mathcal{R}_{t,J}$, we continue the procedure by removing $\mathcal{E}_{1,t,J}$ from the training data and assigning the next confidence subset as $\mathcal{C}_2(\mathcal{R}_{t,J}) = H\Big(\mathcal{R}_{t,J} \setminus \mathcal{E}_{1,t,J}\Big)$, with corresponding set of extreme points denoted by $\mathcal{E}_{2,t,J}$. At this point, in order to define the probability interval $\Big[\underline{p}_{2,t}, \overline{p}_{2,t}\Big]$ of the confidence subset $\mathcal{C}_2(\mathcal{R}_{t,J})$, we make use of the so-called Wilson Interval

(WILSON, 1927) of a proportion measure, i.e.,

$$\hat{p}_{2,t} \triangleq \frac{1}{J}\left[\sum_{\hat{\boldsymbol{r}}\in\mathcal{R}_{t,J}}\left(\mathbb{I}_{\left\{\hat{\boldsymbol{r}}\in\mathcal{C}_2(\mathcal{R}_{t,J})\right\}}\right)\right] \xrightarrow{\quad D \quad} N\left(p_{2,t}, \frac{p_{2,t}(1-p_{2,t})}{J}\right), \tag{3-5}$$

by the Central Limit Theorem (CASELLA; BERGER, 2001). Therefore, a probability coverage for $p_{2,t}$ can be constructed from the standard Wald bilateral β-confidence interval (AGRESTI; COULL, 1998; BROWN; CAI; DASGUPTA, 2001), i.e.,

$$\underline{p}_{2,t} = \hat{p}_{2,t} - \Phi^{-1}\left(\frac{1-\beta}{2}\right)\sqrt{\frac{\hat{p}_{2,t}(1-\hat{p}_{2,t})}{J}}, \tag{3-6}$$

$$\overline{p}_{2,t} = \hat{p}_{2,t} + \Phi^{-1}\left(\frac{1-\beta}{2}\right)\sqrt{\frac{\hat{p}_{2,t}(1-\hat{p}_{2,t})}{J}}; \tag{3-7}$$

where $\Phi(\cdot)$ denotes the standard Normal cumulative distribution function.

The procedure carries on until the number of historical points remaining to construct coverage probability interval falls behind a given threshold ε, which implicates that the significance of the confidence interval (3-6)–(3-7) is sufficiently low (AGRESTI; COULL, 1998; BROWN; CAI; DASGUPTA, 2001). Algorithm 2 depicts the procedure proposed in this work to construct the distributional uncertainty set (3-4).

Algorithm 2: Convex Hull Sequence-based Distributional Uncertainty Set

Input: Training dataset $\mathcal{R}_{t,J}$, confidence level $\beta \in (0,1)$ and threshold $\varepsilon \geq 0$;

Initialization: Set $\mathcal{K}_1 \leftarrow \mathcal{R}_{t,J}$, $\mathcal{E}_{1,t,J} \leftarrow \emptyset$ and $I \leftarrow 1$;

while `TRUE` **do**

  for $\hat{\boldsymbol{r}} \in \mathcal{K}_I$ **do**

    if $\hat{\boldsymbol{r}} \in \Pi\Big(\mathcal{K}_I\Big)$ **then**

      $\mathcal{E}_{I,t,J} \leftarrow \mathcal{E}_{I,t,J} \cup \{\hat{\boldsymbol{r}}\}$

  Set $\mathcal{C}_I(\mathcal{R}_{t,J}) = H\Big(\mathcal{E}_{I,t,J}\Big)$ and compute $\underline{p}_{I,t}$ and $\overline{p}_{I,t}$ using (3-6) and (3-7), respectively.

  if $\Big|\mathcal{K}_I \setminus \mathcal{E}_{I,t,J}\Big| < \varepsilon$ **then**

    Return: $\Big\{\mathcal{C}_i(\cdot), \mathcal{E}_{i,t,J}\Big\}_{i\in\mathcal{I}}$, $\overline{\boldsymbol{p}}_t$ and $\underline{\boldsymbol{p}}_t$.

  else

    Set $I \leftarrow I+1$, $\mathcal{K}_I \leftarrow \mathcal{K}_{I-1} \setminus \mathcal{E}_{I-1,t,J}$ and $\mathcal{E}_{I,t,J} \leftarrow \emptyset$;

Note that, from a computational point-of-view, Algorithm 2 has the desirable characteristics of running in a finite number of steps, in polynomial time. Furthermore, at a given iteration I, the steps to identify the level-set-wise extreme points $\mathcal{E}_{I,t,J}$ can be computed in parallel, thus potentially improving the scalability and computational performance of the DUS construction algorithm. Figure 3.2 illustrates the concept of the distributional uncertainty set design proposed in this work. Essentially, each gray-shaded polytope in the left panel of Figure 3.2 represents the sequence of confidence sets in (3-4) constructed from the available training dataset (black dots) and the right panel of Figure 3.2 depicts an interpretation of the distributional set considered in this work. In fact, note that the topological description of the DUS in (3-4) is based on the specification of a collection of level sets whose probability measure fit within an *a priori* defined interval. We argue that this representation of a distribution function embeds significantly more structural information regarding the "true" unknown data-generating distribution into the distributional set than other widely used metrics for DUS construction (e.g., moment- or distance-based metrics). However, we also recognize that the proper specification of

these elements (level sets and probability intervals) is a key challenge for the practical implementation of (3-4) in real applications. Therefore, the main purpose of the design procedure proposed in this work is to entirely extract from the training dataset information both the level sets and their probability interval, without relying in a particular parametrization.

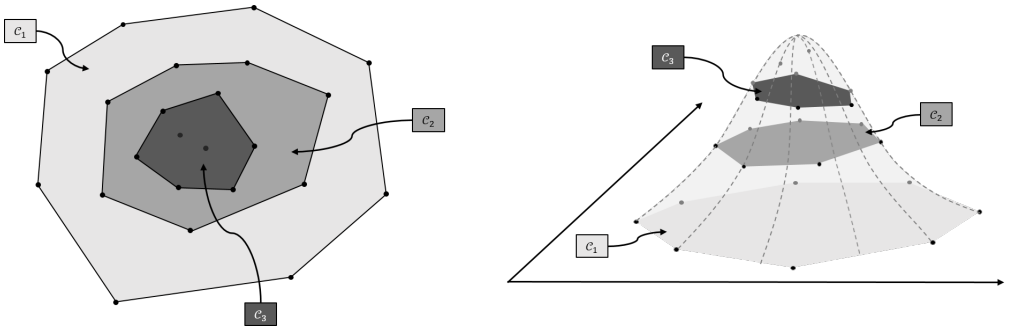


Figure 3.2: Left panel represents the sequence of confidence sets (each gray-shaded polytope) in (3-4) constructed from the available training dataset (black dots) and the right panel depicts an interpretation of the distributional set considered in this work.

3.3 Single-Level Equivalent Formulation

The structure of the portfolio allocation problem (3-3) considered in this work falls within the class of a two-level system of optimization problems, not suitable thus for the application of standard mathematical programming algorithms or the direct implementation on commercial solvers. Nevertheless, by leveraging on convex analysis and duality theory, in this section, a procedure to reformulate the proposed non-convex optimization problem (3-3) into a single-level, linear programming problem is discussed. Fundamentally, the main challenge to solve (3-3) relies on the capability to efficiently handle the distributional uncertainty set $\mathcal{P}$, both in the acceptance set (3-2) and in (3-3). For expository purposes, to begin with, Lemma 1 shows that the acceptance set $\mathcal{A}_t$ has a tractable reformulation.

Lemma 1. *For a given time frame t, let $\mathcal{R}_{t,J}$ be a finite set of historical data with size J and $\mathcal{P}\left(\mathcal{R}_{t,J}\right)$ the distributional uncertainty set defined in* (3-4) *with confidence subsets $\left\{\mathcal{C}_i\left(\mathcal{R}_{t,J}\right)\right\}_{i\in\mathcal{I}}$ with respective probability interval $\left\{\left(\underline{p}_i,\overline{p}_i\right)\right\}_{i\in\mathcal{I}}$, and level-set-wise extreme points $\left\{\mathcal{E}_{i,t,J}\right\}_{i\in\mathcal{I}}$ constructed using Algorithm 2. Then, a portfolio allocation $\hat{\mathbf{x}}_t \in \mathcal{A}_t$ if and only if there is $\eta \in \mathbb{R}$*

and $\Big(\boldsymbol{\lambda}, \boldsymbol{\theta}\Big) \in \mathbb{R}_{+}^{|\mathcal{I}|} \times \mathbb{R}_{+}^{|\mathcal{I}|}$ *that satisfy the constraint system*

$$\begin{cases} \eta - \Gamma_t \geq (1-\alpha)^{-1}\left(\sum\limits_{i\in\mathcal{I}}\Big(\overline{p}_i\theta_i - \underline{p}_i\lambda_i\Big)\right); \\ \sum\limits_{j=1}^{i}\Big(\theta_i - \lambda_i\Big) \geq \max\Big\{\eta - \phi(\hat{\mathbf{x}}_t, \hat{\boldsymbol{r}}_t), 0\Big\}, \qquad\qquad \forall\ \hat{\boldsymbol{r}}_t \in \mathcal{E}_{i,t,J},\ i \in \mathcal{I}. \end{cases}$$

Proof. : Recalling from (3-2), by making use of the CVaR-representability result presented in (ROCKAFELLAR; URYASEV, 2002) and (STREET, 2010), the acceptance set proposed in this work can be equivalently written with the following hierarchical structure of optimization problems:

$$\mathcal{A}_t = \left\{\mathbf{x}_t \in \mathcal{X}_t \;\middle|\; \inf_{\mathbb{P}\in\mathcal{P}(\mathcal{R}_{t,J})}\left\{\max_{\eta\in\mathbb{R}}\Big\{f_{\mathbf{x}_t}(\eta, \mathbb{P})\Big\}\right\} \geq \Gamma_t\right\}, \tag{3-8}$$

with $f_{\mathbf{x}_t} : \mathbb{R} \times \mathcal{P}(\mathcal{R}_{t,J})$ defined as:

$$f_{\mathbf{x}_t}(\eta, \mathbb{P}) \triangleq\ \eta - \Big(1-\alpha\Big)^{-1}\mathbb{E}^{(\mathbb{P})}\Big[\max\Big\{\eta - \phi\Big(\mathbf{x}_t, \tilde{\boldsymbol{r}}_t\Big), 0\Big\}\Big], \qquad \forall\ \mathbf{x}_t \in \mathcal{X}_t$$

and $\phi\Big(\mathbf{x}_t, \tilde{\boldsymbol{r}}_t\Big) = (\mathbf{1} + \tilde{\boldsymbol{r}}_t)^{\top}\mathbf{x}_t$, as in (3-1). Firstly, note that $\tilde{\boldsymbol{r}}_t$ has finite support since $\Xi_t \subseteq \mathcal{C}_1(\mathcal{R}_{t,J})$, by construction. Thus, there exists a compact set $\mathcal{Z} \subset \mathbb{R}$ such that:

$$\arg\max_{\eta\in\mathbb{R}}\Big\{f_{\mathbf{x}_t}(\eta, \mathbb{P})\Big\} \subset \mathcal{Z}, \qquad \forall\ \mathbb{P} \in \mathcal{P}(\mathcal{R}_{t,J}). \tag{3-9}$$

Furthermore,

1. $f_{\mathbf{x}_t}(\eta, \cdot)$ is concave[2] in $\eta \in \mathbb{R}$, hence continuous in $\mathcal{Z}$; and
2. $f_{\mathbf{x}_t}(\cdot, \mathbb{P})$ is linear, thus convex, in $\mathbb{P} \in \mathcal{P}(\mathcal{R}_{t,J})$ because $\mathbb{E}$ is a linear operator[3] in $\mathbb{P}$.

Therefore, based on *Fan's MiniMax Theorem* (BORWEIN; ZHUANG,

[2]See, for instance, Theorem 10 in (ROCKAFELLAR; URYASEV, 2002).

[3]We refer to Chapter 3 of (KUBRUSLY, 2006) for a thorough discussion on integral representations and properties.

1985), the CVaR-based the acceptance set is equivalent to:

$$\begin{aligned}
\mathcal{A}_t &= \left\{ \mathbf{x}_t \in \mathcal{X}_t \;\middle|\; \max_{\eta \in \mathcal{Z}} \left\{ \inf_{\mathbb{P} \in \mathcal{P}(\mathcal{R}_{t,J})} \left\{ f_{\mathbf{x}_t}(\eta, \mathbb{P}) \right\} \right\} \geq \Gamma_t \right\} \\
&= \left\{ \mathbf{x}_t \in \mathcal{X}_t \;\middle|\; \max_{\eta \in \mathcal{Z}} \left\{ \eta - \Big(1-\alpha\Big)^{-1} \left(\sup_{\mathbb{P} \in \mathcal{P}(\mathcal{R}_{t,J})} \left\{ \mathbb{E}^{(\mathbb{P})} \Big[\max\Big\{ \eta - \phi\Big(\mathbf{x}_t, \tilde{\boldsymbol{r}}_t\Big), 0 \Big\} \Big] \right\} \right) \right\} \geq \Gamma_t \right\} \\
&= \left\{ \mathbf{x}_t \in \mathcal{X}_t \;\middle|\; \begin{array}{l} \exists\, \eta \in \mathcal{Z}; \\ \\ \eta - \Big(1-\alpha\Big)^{-1} \left(\sup_{\mathbb{P} \in \mathcal{P}(\mathcal{R}_{t,J})} \left\{ \mathbb{E}^{(\mathbb{P})} \Big[\max\Big\{ \eta - \phi\Big(\mathbf{x}_t, \tilde{\boldsymbol{r}}_t\Big), 0 \Big\} \Big] \right\} \right) \geq \Gamma_t \end{array} \right\}.
\end{aligned} \tag{3-10}$$

Note that the remaining optimization problem in (3-10) coincides with the optimal value of the following optimization problem over non-negative measures:

$$\max_{\mu \in \mathcal{M}_+(\mathbb{R}^n)} \int_{\hat{\boldsymbol{r}} \in \mathcal{C}_1(\mathcal{R}_{t,J})} \Big(\max\Big\{ \eta - \phi\Big(\mathbf{x}_t, \hat{\boldsymbol{r}}\Big), 0 \Big\} \Big) \mathrm{d}\mu(\hat{\boldsymbol{r}}) \tag{3-11}$$

subject to:

$$\int_{\hat{\boldsymbol{r}} \in \mathcal{C}_i(\mathcal{R}_{t,J})} \mathrm{d}\mu(\hat{\boldsymbol{r}}) \geq \underline{p}_{i,t}, \qquad : \lambda_i \quad \forall\, i \in \mathcal{I}; \tag{3-12}$$

$$\int_{\hat{\boldsymbol{r}} \in \mathcal{C}_i(\mathcal{R}_{t,J})} \mathrm{d}\mu(\hat{\boldsymbol{r}}) \leq \overline{p}_{i,t}, \qquad : \theta_i \quad \forall\, i \in \mathcal{I}, \tag{3-13}$$

where $\mathcal{M}_+(\mathbb{R}^n)$ is the set of non-negative measures on $\mathbb{R}^n$, and $\boldsymbol{\lambda} = \{\lambda_1, \ldots, \lambda_I\}$ and $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_I\}$ are the Lagrangian multiplier of constraints (3-12) and (3-13), respectively. It worth highlighting that, since $\underline{p}_{1,t} = \overline{p}_{1,t} = 1$ by construction, every feasible measure in (3-12)–(3-13) is directly identified with a probability measure $\mathbb{P} \in \mathcal{P}_0(\mathbb{R}^n)$. The dual of the problem (3-11)–(3-13) is given by:

$$\min_{\boldsymbol{\lambda}, \boldsymbol{\theta}} \sum_{i \in \mathcal{I}} \Big(\overline{p}_i \theta_i - \underline{p}_i \lambda_i \Big) \tag{3-14}$$

subject to:

$$\sum_{i \in \mathcal{I}} \mathbb{I}_{\left\{ \hat{\boldsymbol{r}} \in \mathcal{C}_i(\mathcal{R}_{t,J}) \right\}} \Big(\theta_i - \lambda_i \Big) \geq \max\Big\{ \eta - \phi\Big(\mathbf{x}_t, \hat{\boldsymbol{r}}\Big), 0 \Big\}, \quad \forall\, \hat{\boldsymbol{r}} \in \mathcal{C}_1(\mathcal{R}_{t,J}); \tag{3-15}$$

$$\theta_i, \lambda_i \geq 0, \qquad \forall\, i \in \mathcal{I}. \tag{3-16}$$

Since the confidence subsets $\Big\{ \mathcal{C}_i(\mathcal{R}_{t,J}) \Big\}_{i \in \mathcal{I}}$ are nested in an increasing order within $\mathcal{I}$, for each $i \in \{1, \ldots, I-1\}$, let $\overline{\mathcal{C}}_i(\mathcal{R}_{t,J}) \triangleq \mathcal{C}_i(\mathcal{R}_{t,J}) \setminus \mathcal{C}_{i+1}(\mathcal{R}_{t,J})$ be a partition of the support set of $\tilde{\boldsymbol{r}}_t$. In this context, equation (3-15) can be

equivalently re-written as:

$$\sum_{j=1}^{i}\Big(\theta_j-\lambda_j\Big)\geq\max\Big\{\eta-\phi\Big(\mathbf{x}_t,\hat{\boldsymbol{r}}\Big),0\Big\},\qquad\forall\;\hat{\boldsymbol{r}}\in\overline{\mathcal{C}}_i(\mathcal{R}_{t,J}),\;i\in\mathcal{I}.\tag{3-17}$$

Then, for a given $i\in\mathcal{I}$, we can reformulate equation (3-17) as:

$$\sum_{j=1}^{i}\Big(\theta_j-\lambda_j\Big)\geq\max_{\hat{\boldsymbol{r}}\in\overline{\mathcal{C}}_i(\mathcal{R}_{t,J})}\Big\{\max\Big\{\eta-\phi\Big(\mathbf{x}_t,\hat{\boldsymbol{r}}\Big),0\Big\}\Big\}.\tag{3-18}$$

Since $\max\Big\{\eta-\phi\Big(\mathbf{x}_t,\hat{\boldsymbol{r}}\Big),0\Big\}$ is convex in $\hat{\boldsymbol{r}}\in\overline{\mathcal{C}}_i(\mathcal{R}_{t,J})$, the optimal set of the optimization problem in the right-hand-side of (3-18) is contained within the set of extreme points of $\mathcal{C}_i(\mathcal{R}_{t,J})$. Thus,

$$\sum_{j=1}^{i}\Big(\theta_j-\lambda_j\Big)\geq\max\Big\{\eta-\phi\Big(\mathbf{x}_t,\hat{\boldsymbol{r}}\Big),0\Big\},\qquad\forall\;\hat{\boldsymbol{r}}\in\mathcal{E}_{i,t,J},\;i\in\mathcal{I}.\tag{3-19}$$

Following this result, the acceptability constraint is satisfied if $\exists\;\eta\in\mathcal{Z}$ such that

$$\eta-\Gamma_t\geq\Big(1-\alpha\Big)^{-1}\min_{\boldsymbol{\lambda},\boldsymbol{\theta}}\left\{\sum_{i\in\mathcal{I}}\Big(\overline{p}_i\theta_i-\underline{p}_i\lambda_i\Big)\;\middle|\;\begin{array}{l}\displaystyle\sum_{j=1}^{i}\Big(\theta_j-\lambda_j\Big)\geq\max\Big\{\eta-\phi\Big(\mathbf{x}_t,\hat{\boldsymbol{r}}\Big),0\Big\},\\ \qquad\qquad\qquad\forall\;\hat{\boldsymbol{r}}\in\mathcal{E}_{i,t,J},\;i\in\mathcal{I}.\\ \theta_i,\lambda_i\geq0,\\ \qquad\qquad\qquad\forall\;i\in\mathcal{I}.\end{array}\right\}$$

Thus, the CVaR-based acceptance set can be equivalently re-written as follows:

$$\mathcal{A}_t=\left\{\mathbf{x}_t\in\mathcal{X}_t\;\middle|\;\begin{array}{l}\exists\;\eta\in\mathbb{R},\;(\boldsymbol{\lambda},\boldsymbol{\theta})\in\mathbb{R}^I_+\times\mathbb{R}^I_+;\\ \quad\eta-\Gamma_t\geq\Big(1-\alpha\Big)^{-1}\left(\displaystyle\sum_{i\in\mathcal{I}}\Big(\overline{p}_i\theta_i-\underline{p}_i\lambda_i\Big)\right)\\ \quad\displaystyle\sum_{j=1}^{i}\Big(\theta_j-\lambda_j\Big)\geq\max\Big\{\eta-\phi\Big(\mathbf{x}_t,\hat{\boldsymbol{r}}\Big),0\Big\},\quad\forall\;\hat{\boldsymbol{r}}\in\mathcal{E}_{i,t,J},\;i\in\mathcal{I}.\end{array}\right\}.\tag{3-20}$$

□

Generally speaking, Lemma 1 implies that verifying the acceptability of a given portfolio allocation $\hat{\mathbf{x}}_t\in\mathcal{X}_t$ can be performed in polynomial time. Furthermore, from a prescriptive viewpoint, the CVaR-based acceptance set

(3-2) with $\mathcal{P}\Big(\mathcal{R}_{t,J}\Big)$ constructed based on Algorithm 2 keeps the tractability of the main decision-making problem, whenever the latter is tractable. In the particular context of this work, next, we show that the proposed portfolio allocation problem (3-3) has also a tractable, single-level linear, reformulation.

Theorem 1. *For a given time frame t, let $\mathcal{R}_{t,J}$ be a finite set of historical data with size J and $\mathcal{P}\Big(\mathcal{R}_{t,J}\Big)$ the distributional uncertainty set, as defined in* (3-4)*, with confidence subsets $\Big\{\mathcal{C}_i\Big(\mathcal{R}_{t,J}\Big)\Big\}_{i\in\mathcal{I}}$ with respective probability interval $\Big\{\Big(\underline{p}_i, \overline{p}_i\Big)\Big\}_{i\in\mathcal{I}}$, and level-set-wise extreme points $\Big\{\mathcal{E}_{i,t,J}\Big\}_{i\in\mathcal{I}}$, constructed using Algorithm 2. The data-driven distributionally robust portfolio allocation model* (3-3) *has the following equivalent single-level linear reformulation*

$$\max_{\substack{\mathbf{x}_t,\boldsymbol{\lambda}^{(1)},\boldsymbol{\theta}^{(1)},\\ \eta^{(1)},\boldsymbol{\delta}^{(1)},\boldsymbol{\lambda}^{(2)},\boldsymbol{\theta}^{(2)}}} \sum_{i\in\mathcal{I}}\Big(\underline{p}_{i,t}\lambda_i^{(2)} - \overline{p}_{i,t}\theta_i^{(2)}\Big) \tag{3-21}$$

subject to:

$$\sum_{j=1}^{i}\Big(\lambda_j^{(2)} - \theta_j^{(2)}\Big) \le \phi\Big(\mathbf{x}_t, \hat{\boldsymbol{r}}\Big) \qquad \forall\ \hat{\boldsymbol{r}} \in \mathcal{E}_{i,t,J},\ i \in \mathcal{I}; \tag{3-22}$$

$$\lambda_i^{(2)}, \theta_i^{(2)} \ge 0, \qquad \forall\ i \in \mathcal{I}; \tag{3-23}$$

$$\eta^{(1)} - \Gamma_t \ge \Big(1-\alpha\Big)^{-1}\left(\sum_{i\in\mathcal{I}}\Big(\overline{p}_i\theta_i^{(1)} - \underline{p}_i\lambda_i^{(1)}\Big)\right); \tag{3-24}$$

$$\sum_{j=1}^{i}\Big(\theta_j^{(1)} - \lambda_j^{(1)}\Big) \ge \delta_{\hat{\boldsymbol{r}}}^{(1)}, \qquad \forall\ \hat{\boldsymbol{r}} \in \mathcal{E}_{i,t,J},\ i \in \mathcal{I}; \tag{3-25}$$

$$\delta_{\hat{\boldsymbol{r}}}^{(1)} \ge \eta^{(1)} - \phi\Big(\mathbf{x}_t, \hat{\boldsymbol{r}}\Big), \qquad \forall\ \hat{\boldsymbol{r}} \in \mathcal{E}_{i,t,J},\ i \in \mathcal{I}; \tag{3-26}$$

$$\delta_{\hat{\boldsymbol{r}}}^{(1)} \ge 0, \qquad \forall\ \hat{\boldsymbol{r}} \in \mathcal{E}_{i,t,J},\ i \in \mathcal{I}; \tag{3-27}$$

$$\lambda_i^{(1)}, \theta_i^{(1)} \ge 0, \qquad \forall\ i \in \mathcal{I}; \tag{3-28}$$

$$\mathbf{x}_t \in \mathcal{X}_t. \tag{3-29}$$

Proof. : Following the results of Lemma 1, it remains only to derive the reformulation of the inner optimization problem $\inf_{\mathbb{Q}\in\mathcal{P}(\mathcal{R}_{t,J})}\ \mathbb{E}^{(\mathbb{Q})}\Big[\phi\Big(\mathbf{x}_t, \tilde{\boldsymbol{r}}_t\Big)\Big]$, which has optimal value coincident with the following optimization problem over non-negative measures:

$$\min_{\mu\in\mathcal{M}_+(\mathbb{R}^n)}\ \int_{\hat{\boldsymbol{r}}\in\mathcal{C}_1(\mathcal{R}_{t,J})}\phi\Big(\mathbf{x}_t, \hat{\boldsymbol{r}}\Big)\mathrm{d}\mu(\hat{\boldsymbol{r}}) \tag{3-30}$$

subject to:

$$\int_{\hat{\boldsymbol{r}}\in\mathcal{C}_i(\mathcal{R}_{t,J})}\mathrm{d}\mu(\hat{\boldsymbol{r}}) \le \overline{p}_{i,t}, \qquad : \lambda_i^{(2)} \qquad \forall\ i \in \mathcal{I}; \tag{3-31}$$

$$\int_{\hat{\boldsymbol{r}}\in\mathcal{C}_i(\mathcal{R}_{t,J})}\mathrm{d}\mu(\hat{\boldsymbol{r}}) \ge \underline{p}_{i,t}, \qquad : \theta_i^{(2)} \qquad \forall\ i \in \mathcal{I}, \tag{3-32}$$

with $\mathcal{M}_+(\mathbb{R}^n)$ denoting the set of non-negative measures on $\mathbb{R}^n$, and $\boldsymbol{\lambda}^{(2)} = \{\lambda_1^{(2)}, \ldots, \lambda_{I_2}^{(2)}\}$ and $\boldsymbol{\theta}^{(2)} = \{\theta_{1_2}^{(2)}, \ldots, \theta_I^{(2)}\}$ the Lagrangian multiplier of constraints (3-31) and (3-32), respectively. We highlight, nevertheless, that by construction, $\underline{p}_{1,t} = \overline{p}_{1,t} = 1$, thus every measure that is feasible in (3-31)–(3-32) has a direct identification with a probability measure $\mathbb{Q} \in \mathcal{P}_0(\mathbb{R}^n)$. From now on, the roadmap of the reformulation procedure follows similarly to Lemma 1. Firstly, we take the dual of problem (3-30)–(3-32), presented next in (3-33)–(3-35).

$$\max_{\boldsymbol{\lambda}^{(2)}, \boldsymbol{\theta}^{(2)}} \sum_{i \in \mathcal{I}} \left(\underline{p}_{i,t} \lambda_i^{(2)} - \overline{p}_{i,t} \theta_i^{(2)} \right) \tag{3-33}$$

subject to:

$$\sum_{i \in \mathcal{I}} \mathbb{I}_{\left\{\hat{\boldsymbol{r}} \in \mathcal{C}_i(\mathcal{R}_{t,J})\right\}} \left(\lambda_i^{(2)} - \theta_i^{(2)} \right) \leq \phi\Big(\mathbf{x}_t, \hat{\boldsymbol{r}}\Big), \qquad \forall\, \hat{\boldsymbol{r}} \in \mathcal{C}_1(\mathcal{R}_{t,J}); \tag{3-34}$$

$$\lambda_i^{(2)}, \theta_i^{(2)} \geq 0, \qquad \forall\, i \in \mathcal{I}. \tag{3-35}$$

Problem (3-33)–(3-35) is a semi-infinite optimization problem, thus non-tractable for standard mathematical programming algorithms. Next, by making use of the design structure of the confidence sets $\left\{\mathcal{C}_i(\mathcal{R}_{t,J})\right\}_{i \in \mathcal{I}}$, we create the following partition of the support set of $\tilde{\boldsymbol{r}}_t$ by setting $\overline{\mathcal{C}}_i(\mathcal{R}_{t,J}) \triangleq \mathcal{C}_i(\mathcal{R}_{t,J}) \setminus \mathcal{C}_{i+1}(\mathcal{R}_{t,J})$. As a consequence, equation (3-34) can be equivalently re-written as follows:

$$\begin{aligned}
\sum_{j=1}^{i} \left(\lambda_j^{(2)} - \theta_j^{(2)} \right) &\leq \phi\Big(\mathbf{x}_t, \hat{\boldsymbol{r}}\Big), && \forall\, \hat{\boldsymbol{r}} \in \overline{\mathcal{C}}_i(\mathcal{R}_{t,J}),\ i \in \mathcal{I}; \\
&\leq \min_{\hat{\boldsymbol{r}} \in \overline{\mathcal{C}}_i(\mathcal{R}_{t,J})} \left\{ \phi\Big(\mathbf{x}_t, \hat{\boldsymbol{r}}\Big) \right\}, && \forall\, i \in \mathcal{I}; \\
&\leq \phi\Big(\mathbf{x}_t, \hat{\boldsymbol{r}}\Big) && \forall\, \hat{\boldsymbol{r}} \in \mathcal{E}_{i,t,J},\ i \in \mathcal{I}.
\end{aligned}$$

Thus, the inner optimization problem $\inf_{\mathbb{Q} \in \mathcal{P}(\mathcal{R}_{t,J})} \mathbb{E}^{(\mathbb{Q})}\Big[\phi\Big(\mathbf{x}_t, \tilde{\boldsymbol{r}}_t\Big)\Big]$ in (3-3) is equivalent to the following linear programming problem:

$$\max_{\boldsymbol{\lambda}^{(2)}, \boldsymbol{\theta}^{(2)}} \sum_{i \in \mathcal{I}_2} \left(\underline{p}_{i,t} \lambda_i^{(2)} - \overline{p}_{i,t} \theta_i^{(2)} \right) \tag{3-36}$$

subject to:

$$\sum_{j=1}^{i} \left(\lambda_j^{(2)} - \theta_j^{(2)} \right) \leq \phi\Big(\mathbf{x}_t, \hat{\boldsymbol{r}}\Big) \qquad \forall\, \hat{\boldsymbol{r}} \in \mathcal{E}_{i,t,J},\ i \in \mathcal{I}; \tag{3-37}$$

$$\lambda_i^{(2)}, \theta_i^{(2)} \geq 0, \qquad \forall\, i \in \mathcal{I}. \tag{3-38}$$

Finally, by making use of the main result in Lemma 1, the data-

driven distributionally robust portfolio allocation problem model (3-3) has the following equivalent single-level linear reformulation:

$$\max_{\substack{\mathbf{x}_t,\boldsymbol{\lambda}^{(1)},\boldsymbol{\theta}^{(1)},\\ \eta^{(1)},\boldsymbol{\delta}^{(1)}\boldsymbol{\lambda}^{(2)},\boldsymbol{\theta}^{(2)}}} \sum_{i\in\mathcal{I}}\Big(\underline{p}_{i,t}\lambda_i^{(2)} - \overline{p}_{i,t}\theta_i^{(2)}\Big) \tag{3-39}$$

subject to:

$$\sum_{j=1}^{i}\Big(\lambda_j^{(2)} - \theta_j^{(2)}\Big) \leq \phi\Big(\mathbf{x}_t, \hat{\boldsymbol{r}}\Big) \qquad \forall\ \hat{\boldsymbol{r}} \in \mathcal{E}_{i,t,J},\ i \in \mathcal{I}; \tag{3-40}$$

$$\lambda_i^{(2)}, \theta_i^{(2)} \geq 0, \qquad \forall\ i \in \mathcal{I}; \tag{3-41}$$

$$\eta^{(1)} - \Gamma_t \geq \Big(1-\alpha\Big)^{-1}\left(\sum_{i\in\mathcal{I}}\Big(\overline{p}_i\theta_i^{(1)} - \underline{p}_i\lambda_i^{(1)}\Big)\right); \tag{3-42}$$

$$\sum_{j=1}^{i}\Big(\theta_j^{(1)} - \lambda_j^{(1)}\Big) \geq \delta_{\hat{\boldsymbol{r}}}^{(1)}, \qquad \forall\ \hat{\boldsymbol{r}} \in \mathcal{E}_{i,t,J},\ i \in \mathcal{I}; \tag{3-43}$$

$$\delta_{\hat{\boldsymbol{r}}}^{(1)} \geq \eta^{(1)} - \phi\Big(\mathbf{x}_t, \hat{\boldsymbol{r}}\Big), \qquad \forall\ \hat{\boldsymbol{r}} \in \mathcal{E}_{i,t,J},\ i \in \mathcal{I}; \tag{3-44}$$

$$\delta_{\hat{\boldsymbol{r}}}^{(1)} \geq 0, \qquad \forall\ \hat{\boldsymbol{r}} \in \mathcal{E}_{i,t,J},\ i \in \mathcal{I}; \tag{3-45}$$

$$\lambda_i^{(1)}, \theta_i^{(1)} \geq 0, \qquad \forall\ i \in \mathcal{I}; \tag{3-46}$$

$$\mathbf{x}_t \in \mathcal{X}_t, \tag{3-47}$$

where $\boldsymbol{\delta}^{(1)} = \Big\{\delta_{\hat{\boldsymbol{r}}}^{(1)}\Big\}_{\hat{\boldsymbol{r}}\in\mathcal{E}_{i,t,J}}$ is an auxiliary variable to account for the truncation function $\max\Big\{\eta - \phi\Big(\mathbf{x}_t, \hat{\boldsymbol{r}}\Big), 0\Big\}$ in Lemma 1.

□

By virtue of Theorem 1, the data-driven distributionally robust portfolio allocation model (3-3) can be efficiently solved by standard mathematical programming algorithms or direct implemented on commercial solvers. In the next section, a set of numerical experiments is performed aiming at illustrating the applicability and effectiveness of the proposed portfolio allocation methodology.

3.4 Numerical Experiments

In this section, we consider a numerical experiment for the single-period portfolio allocation problem to illustrate the applicability and effectiveness of the proposed methodology.

3.4.1 Experimental Set Up

To evaluate the performance of our methodology in a practical setting, we conduct experiments using real financial data. Specifically, the experiments utilize data sets from the Kenneth R. French database [4] . This includes stocks from the New York Stock Exchange (NYSE), American Stock Exchange (AMEX), and NASDAQ, which are represented through capitalization-weighted indexes across different industry sectors. For our analysis, we employ daily data from five industrial portfolios, namely Consumer (Cnsmr), Manufacturing (Manuf), High Technology (HiTec), Health (Hlth), and Others (Other). We use daily data since mid 1932 until the end of 2023, totaling almost 24.000 trading days. We conduct the experiment on a rolling-horizon fashion, where at each day the model determines the allocation of its current budget on each of the available investment options.

Without loss of generality and for expository reasons, in this work, we assume a standard short-selling constraint, where c represent the cost incurred as a percentage of the amount negotiated:

$$\mathcal{X}_t = \left\{ \mathbf{x}_t \in \mathbb{R}_+^n \;\middle|\; \exists \left(\mathbf{u}_t^+, \mathbf{u}_t^-\right) \in \mathbb{R}_+^n \times \mathbb{R}_+^n : \right.$$
$$\sum_{i=1}^{n} \left(u_{i,t}^+ - u_{i,t}^-\right) = 0;$$
$$\left. x_{i,t} = x_{i,t-1}(1 + \hat{r}_{i,t-1}) + (1-c)u_{i,t}^+ - (1+c)u_{i,t}^-, \quad \forall\, i \in \{1, \ldots, n\}; \right\}.$$

In addition, notice that the data-driven distributionally robust portfolio allocation model (3-3) requires the choice of a value for $\mathcal{J}$, which determines the quantity of past trading days used to construct the DUS.

3.4.2 Portfolio Allocation Analysis

Our initial analysis compares the strategy derived from the DRO model to a naive strategy that allocates an equal portion of the portfolio to each asset at the start of each trading session. To ensure a fair comparison of both strategies in terms of risk, we used the out-of-sample CVaR γ of the naive strategy as the risk constraint value for the DRO model. In case such value

[4] http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html

results in an unfeasible problem for a given step, we opted to minimize the risk measure, following the work of (STREET, 2008).

In this experiment, we consider an initial wealth of \$1 and three different values for $\mathcal{J}$: 30, 180, and 360 days. For each setting we present the cumulative wealth and the portfolio allocation for the whole trading period in Figures 3.3, 3.4, and 3.5

The results are illustrated in Figures 3.3, 3.4 and 3.5.

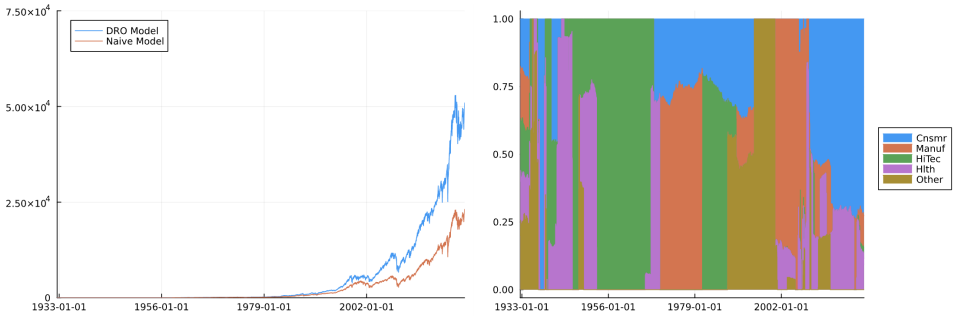


Figure 3.3: Left panel shows the cumulative wealth comparison between naive model and the DRO model with $\mathcal{J} = 30$. Right panel displays the portfolio composition.

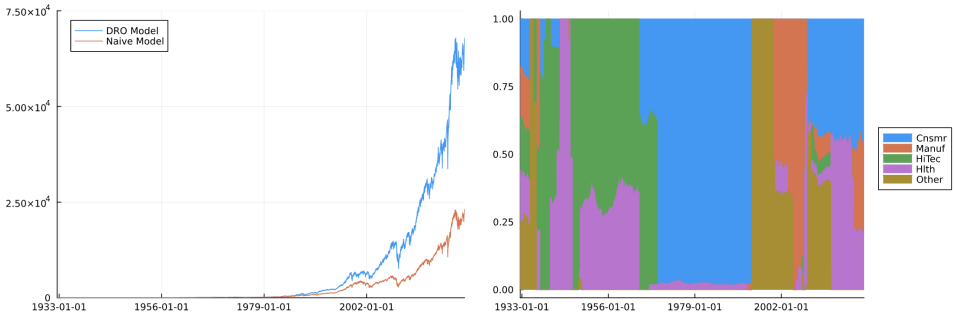


Figure 3.4: Left panel shows the cumulative wealth comparison between naive model and the DRO model with $\mathcal{J} = 180$. Right panel displays the portfolio composition.

Noticeably, the model utilizing the last $\mathcal{J} = 180$ days in each trading session demonstrates superior cumulative performance. Nonetheless, all variations of the DRO model surpass the benchmark performance. For better visualization, Figure 3.6 compares the three variations of the DRO model.

3.4.3 Efficient Frontier

A good measure of the suitability of a portfolio allocation model is the efficient frontier that results from adjusting the risk constraint. In Markowitz's (1952) seminal work (MARKOWITZ, 1952), the concept of the trade-off

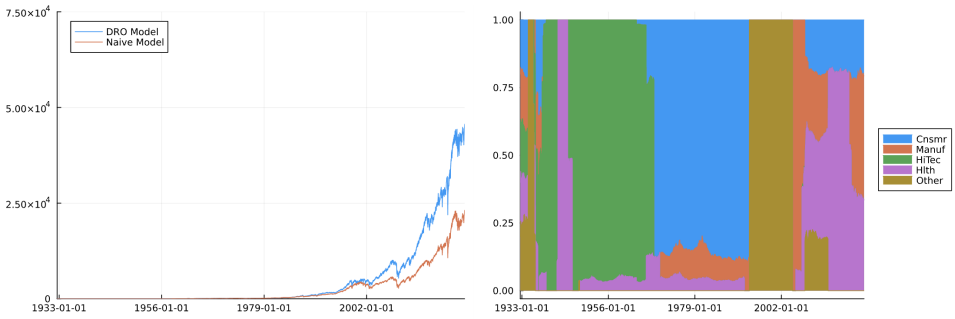


Figure 3.5: Left panel shows the cumulative wealth comparison between naive model and the DRO model with $\mathcal{J} = 360$. Right panel displays the portfolio composition.

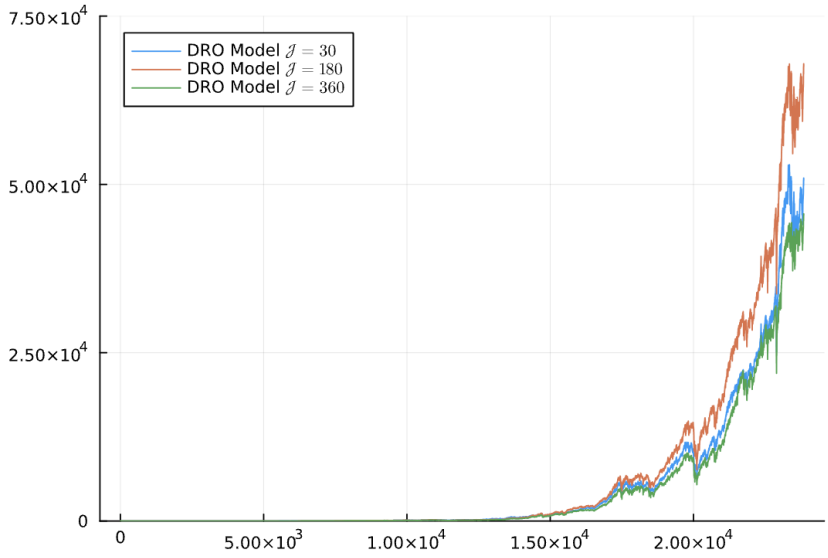


Figure 3.6: Cumulative wealth comparison the DRO model with $\mathcal{J} = 30$, $\mathcal{J} = 180$ and $\mathcal{J} = 360$

between higher risk tolerance and higher expected return was established. This concept has since been crucial in evaluating the balance between these values. In other words, an adequate model should reflect a higher expected return when the allocator tolerates a higher risk, at least within a certain range.

To assess the adequacy of our proposed model, we present the efficient frontiers for each variation of our model in Figure 3.7, varying the CVaR constraint from a loss of 10% to 0%. It is important to note that the nature of our model is inherently robust, as the optimal allocation considers the worst end of the spectrum of possible distribution of returns within the data-driven ambiguity set. This may lead to negative expected returns, as it does in our experiment settings. Nonetheless, this does not compromise the consistency of the frontier, which reveals coherent and quasi-monotonic risk-return profiles; i.e., the higher the risk tolerance, the higher the expected return.

3.4.4 Trailing Analysis

Although the results shown in Figures 3.3, 3.4, and 3.5 demonstrate superior cumulative performance, it can be argued that this performance is heavily dependent on the starting point of the trading session. To provide a more comprehensive analysis, we conduct a trailing analysis to assess cumulative results across different window sizes, varying both the starting and ending points.

Figures 3.8 and 3.9 present the results of the trailing analysis for the model using $\mathcal{J} = 180$. It is evident that the larger the trailing window, the better the model performs compared to the benchmark. The dates on the x-axis indicate the finishing points for given trading sessions, with each figure corresponding to a different rolling window size.

Finally, for the considered values of $\mathcal{J}$, we show in Figure 3.10 the proportion of trailing windows where the model outperformed the benchmark for a given trailing rolling window size. The results suggest that each option eventually outperforms the benchmark from a certain point. Additionally, the model with smaller values for the composition of the DUS achieves more consistent results.

3.5 Conclusions

In this chapter, we applied the PolieDRO framework to the prescriptive task of portfolio optimization. By simultaneously addressing both risk and return aspects, we developed a highly flexible and robust model. Our approach leverages the inherent strengths of Data-Driven Distributionally Robust Optimization to create a portfolio allocation strategy that adapts to varying market conditions and uncertainties.

The PolieDRO framework consistently outperformed a traditional benchmark, particularly over longer investment horizons. Our findings underscore the efficacy of the PolieDRO framework in providing a more reliable and advantageous portfolio optimization strategy compared to naive methods.

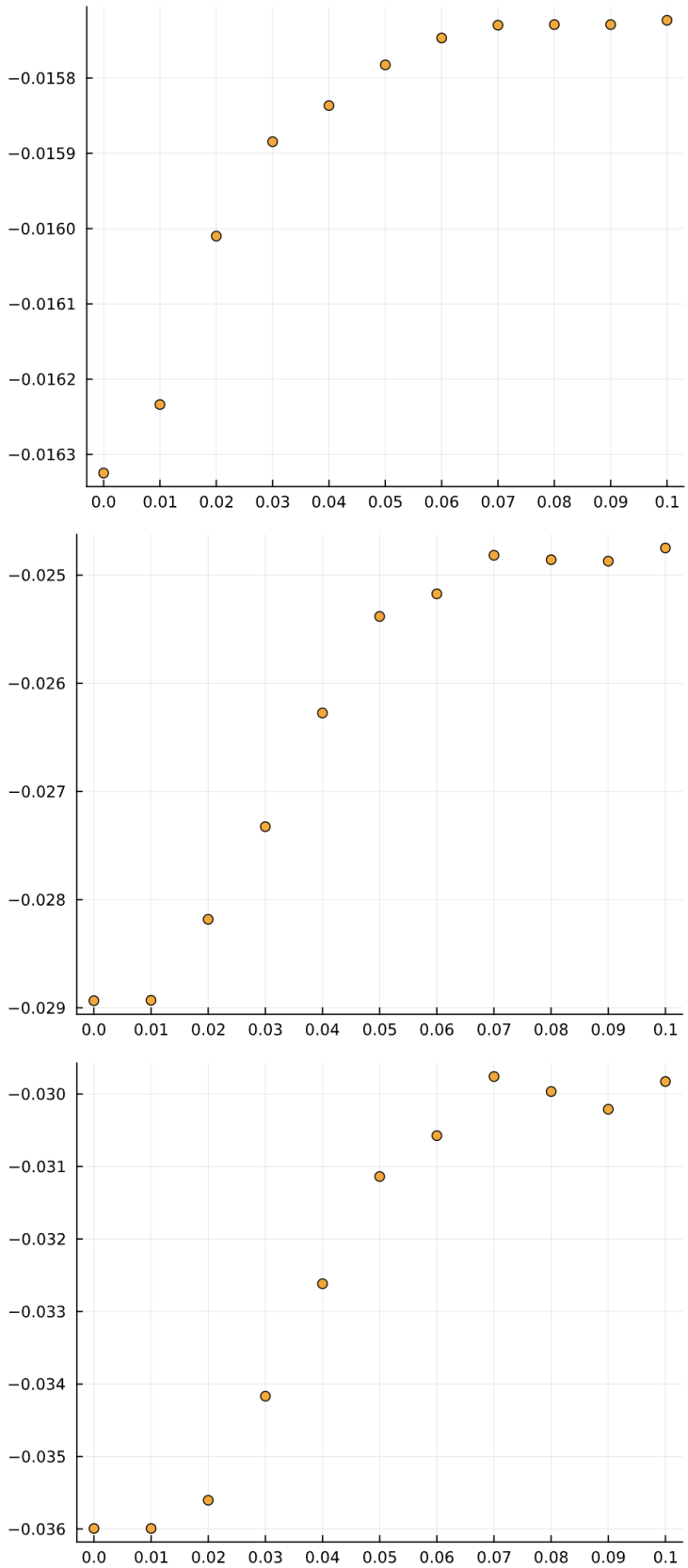


Figure 3.7: Top: Efficient frontier considering $\mathcal{J} = 30$. Middle: Efficient frontier considering $\mathcal{J} = 180$. Bottom: Efficient frontier considering $\mathcal{J} = 360$

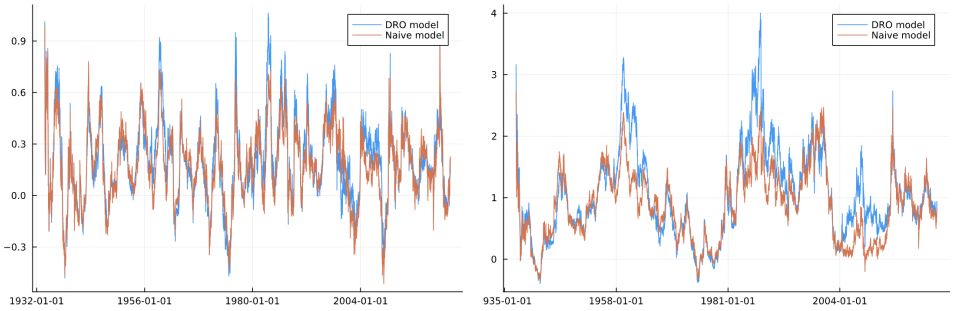


Figure 3.8: Left panel displays the trailing result for $\mathcal{J} = 180$, with rolling windows of 360 days. Right panel shows the result with rolling windows of 1440 days

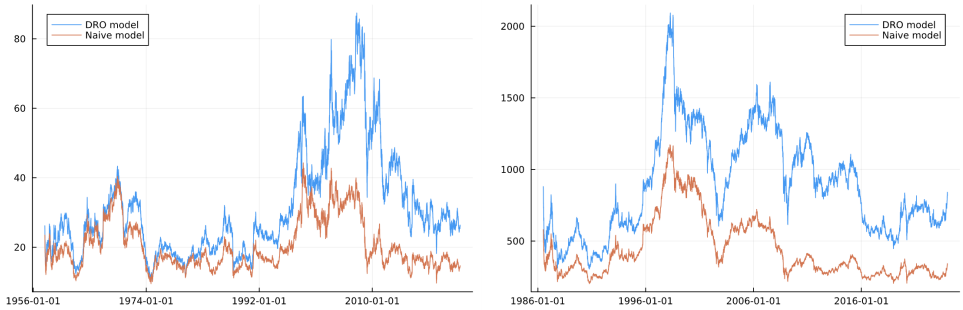


Figure 3.9: Trailing analysis for $\mathcal{J} = 180$, with rolling windows of 7200 days. Right panel shows the result with rolling windows of 14.400 days

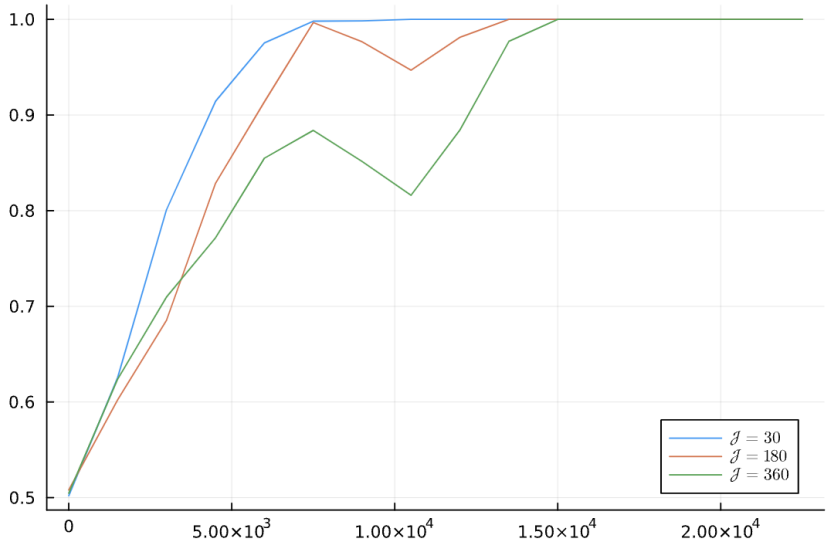


Figure 3.10: Proportion of trailing windows where the model outperformed the benchmark for a given trailing window size.

4
Closing remarks

In this thesis, we have introduced a novel analytical framework that leverages Data-Driven Distributionally Robust Optimization (DRO) for both predictive and prescriptive tasks. Our approach utilized a proposed data-driven ambiguity set with properties that enabled a finite reformulation of previously intractable distributionally robust optimization problems.

Chapter 2 explored predictive applications in the field of Machine Learning. By applying the PolieDRO framework, we proposed new methods for commonly used loss functions in both classification and regression tasks. These new models successfully eliminated the need for regularization hyperparameters while maintaining competitive performance compared to traditional approaches.

Chapter 3 focused on prescriptive applications by presenting a portfolio optimization model that employed a distributionally robust approach to both risk and return aspects. The results demonstrated the superior performance of our model in real-world financial data.

Overall, our findings highlight the potential of the PolieDRO framework to improve both predictive and prescriptive analytics, offering robust and efficient solutions across various applications.

5
Bibliography

AGRESTI, A.; COULL, B. A. Approximate is Better than "Exact" for Interval Estimation of Binomial Proportions. **The American Statistician**, v. 52, n. 2, p. 119–126, Mar. 1998.

ANTHONY, L. F. W.; KANDING, B.; SELVAN, R. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. **arXiv preprint arXiv:2007.03051**, 2020.

ARTZNER, P. et al. Coherent Measures of Risk. **Mathematical Finance**, v. 9, n. 3, p. 203–228, July 1999.

BAN, G.-Y.; KAROUI, N. E.; LIM, A. E. B. Machine Learning and Portfolio Optimization. **Management Science**, v. 64, n. 3, p. 1136–1154, March 2018.

BARBER, C. B.; DOBKIN, D. P.; HUHDANPAA, H. The quickhull algorithm for convex hulls. **ACM Transactions on Mathematical Software (TOMS)**, Acm New York, NY, USA, v. 22, n. 4, p. 469–483, 1996.

BELLONI, A.; CHERNOZHUKOV, V.; WANG, L. Square-root lasso: pivotal recovery of sparse signals via conic programming. **Biometrika**, [Oxford University Press, Biometrika Trust], v. 98, n. 4, p. 791–806, 2011. ISSN 00063444, 14643510. Disponível em: <http://www.jstor.org/stable/23076172>.

BEN-TAL, A.; GHAOUI, L. E.; NEMIROVSKI, A. **Robust optimization**. [S.l.]: Princeton university press, 2009.

BEN-TAL, A.; GHAOUI, L. E.; NEMIROVSKI, A. **Robust Optimization**. 1st. ed. [S.l.]: Princeton University Press, 2009.

BERTSIMAS, D.; BROWN, D. B.; CARAMANIS, C. Theory and applications of robust optimization. **SIAM review**, SIAM, v. 53, n. 3, p. 464–501, 2011.

BERTSIMAS, D. et al. Robust classification. **INFORMS Journal on Optimization**, v. 1, n. 1, p. 2–34, 2019.

BERTSIMAS, D.; GUPTA, V.; KALLUS, N. Data-Driven Robust Optimization. **Mathematical Programming**, v. 167, n. 2, p. 235–292, February 2018.

BERTSIMAS, D.; POPESCU, I. Optimal Inequalities in Probability Theory: A Convex Optimization Approach. **SIAM Journal on Optimization**, v. 15, n. 3, p. 780–804, 2005.

BERTSIMAS, D.; SIM, M.; ZHANG, M. Adaptive Distributionally Robust Optimization. **Management Science**, v. 65, n. 2, p. 604–618, February 2019.

BLANCHET, J.; KANG, Y.; MURTHY, K. Robust wasserstein profile inference and applications to machine learning. **Journal of Applied Probability**, Cambridge University Press, v. 56, n. 3, p. 830–857, 2019.

BORWEIN, J. M.; ZHUANG, D. On Fan's Minimax Theorem. **Mathematical Programming volume**, v. 34, n. 2, p. 232–234, March 1985.

BROWN, D. B.; SMITH, J. E. Dynamic Portfolio Optimization with Transaction Costs: Heuristics and Dual Bounds. **Management Science**, v. 57, n. 10, p. 1752–1770, Oct. 2011.

BROWN, L. D.; CAI, T. T.; DASGUPTA, A. Interval Estimation for a Binomial Proportion. **Statistical Science**, v. 16, n. 2, p. 101–117, May 2001.

BYKAT, A. Convex hull of a finite set of points in two dimensions. **Information Processing Letters**, Elsevier, v. 7, n. 6, p. 296–298, 1978.

CASELLA, G.; BERGER, R. L. **Statistical Inference**. 2nd. ed. [S.l.]: Cengage Learning, 2001.

CHEN, R.; PASCHALIDIS, I. C. A robust learning approach for regression models based on distributionally robust optimization. **Journal of Machine Learning Research**, v. 19, n. 13, 2018.

CHOOBINEH, F.; BRANTING, D. A Simple Approximation for Semivariance. **European Journal of Operational Research**, v. 27, n. 3, p. 364–370, Dec. 1986.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v. 20, n. 3, p. 273–297, 1995. Disponível em: <https://doi.org/10.1007/BF00994018>.

CVITANIć, J.; POLIMENIS, V.; ZAPATERO, F. Optimal Portfolio Allocation with Higher Moments. **Annals of Finance**, v. 4, n. 1, p. 1–28, Jan. 2008.

DELAGE, E.; YE, Y. Distributionally Robust Optimization under Moment Uncertainty with Application to Data-Driven Problems. **Operations Research**, v. 58, n. 3, p. 595–612, May–Jun. 2010.

DEMIGUEL, V. et al. A Generalized Approach to Portfolio Optimization: Improving Performance by Constraining Portfolio Norms. **Management Science**, v. 55, n. 5, p. 798–812, May 2009.

EDDY, W. F. A new convex hull algorithm for planar sets. **ACM Transactions on Mathematical Software (TOMS)**, ACM New York, NY, USA, v. 3, n. 4, p. 398–403, 1977.

ESFAHANI, P. M.; KUHN, D. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. **Mathematical Programming**, Springer, v. 171, n. 1, p. 115–166, 2018.

FABOZZI, F. J.; HUANG, D.; ZHOU, G. Robust Portfolios: Contributions from Operations Research and Finance. **Annals of Operations Research**, v. 176, n. 1, p. 191–220, Apr. 2010.

FERNANDES, B. et al. An adaptive robust portfolio optimization model with loss constraints based on data-driven polyhedral uncertainty sets. **European Journal of Operational Research**, v. 255, n. 3, p. 961 – 970, 2016. ISSN

0377-2217. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221716303757>.

FOLLMER, H.; SCHIED, A. **Stochastic Finance: An Introduction in Discrete Time**. 3rd. ed. [S.l.]: De Gruyter, 2011.

GAMBOA, C. A. et al. Decomposition methods for wasserstein-based data-driven distributionally robust problems. **Operations Research Letters**, Elsevier, v. 49, n. 5, p. 696–702, 2021.

GAO, R.; CHEN, X.; KLEYWEGT, A. J. **Wasserstein Distributionally Robust Optimization and Variation Regularization**. 2020.

GIORGI, E. D. Reward-Risk Portfolio Selection and Stochastic Dominance. **Journal of Banking & Finance**, v. 29, n. 4, p. 895–926, Apr. 2005.

GOH, J.; SIM, M. Distributionally robust optimization and its tractable approximations. **Operations research**, INFORMS, v. 58, n. 4-part-1, p. 902–917, 2010.

GREEN, P.; SILVERMAN, B. W. Constructing the convex hull of a set of points in the plane. **The Computer Journal**, Oxford University Press, v. 22, n. 3, p. 262–266, 1979.

GREENFIELD, J. S. A proof for a quickhull algorithm. In: . [S.l.: s.n.], 1990.

HAO, K. Training a single ai model can emit as much carbon as five cars in their lifetimes. **MIT technology Review**, v. 75, p. 103, 2019.

HARVEY, C. R. et al. Portfolio Selection with Higher Moments. **Quantitative Finance**, v. 10, n. 5, p. 469–485, Apr. 2010.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning**. New York, NY, USA: Springer New York Inc., 2001. (Springer Series in Statistics).

HUANG, X. Mean-Semivariance Models for Fuzzy Portfolio Selection. **Journal of Computational and Applied Mathematics**, v. 217, n. 1, p. 1–8, July 2008.

JIN, X.; LUO, D.; ZENG, X. Tail Risk and Robust Portfolio Decisions. **Management Science**, v. 67, n. 5, p. 3254–3275, May 2021.

JORION, P. **Value at Risk: The New Benchmark for Managing Financial Risk**. 3rd. ed. [S.l.]: McGraw-Hill Education, 2006.

KIM, J. H.; KIM, W. C.; FABOZZI, F. J. Recent Developments in Robust Portfolios with a Worst-Case Approach. **Journal of Optimization Theory and Applications**, v. 161, n. 1, p. 103–121, Apr. 2014.

KOLM, P. N.; TÜTÜNCÜ, R.; FABOZZIC, F. J. 60 Years of Portfolio Optimization: Practical Challenges and Current Trends. **European Journal of Operational Research**, v. 234, n. 2, p. 356–371, Apr. 2014.

KUBRUSLY, C. S. **Measure Theory: A First Course**. 1st. ed. [S.l.]: Academic Press, 2006.

KUHN, D. et al. Wasserstein distributionally robust optimization: Theory and applications in machine learning. In: **Operations Research & Management Science in the Age of Analytics**. [S.l.]: INFORMS, 2019. p. 130–166.

LACOSTE, A. et al. Quantifying the carbon emissions of machine learning. **arXiv preprint arXiv:1910.09700**, 2019.

LI, X.; QIN, Z.; KARC, S. Mean-Variance-Skewness Model for Portfolio Selection with Fuzzy Returns. **European Journal of Operational Research**, v. 202, n. 1, p. 239–247, Apr. 2010.

LIM, A. E. B.; SHANTHIKUMAR, J. G.; VAHN, G.-Y. Robust Portfolio Choice with Learning in the Framework of Regret: Single-Period Case. **Management Science**, v. 58, n. 9, p. 1732–1746, Sept. 2012.

LOTFI, S.; ZENIOS, S. A. Robust VaR and CVaR Optimization under Joint Ambiguity in Distributions, Means, and Covariances. **European Journal of Operational Research**, v. 269, n. 2, p. 556–576, Sept. 2018.

MAO, J. C. T. Models of Capital Budgeting, E-V vs E-S. **The Journal of Financial and Quantitative Analysis**, v. 4, n. 5, p. 657–675, Jan. 1970.

MARKOWITZ, H. Portfolio Selection. **The Journal of Finance**, v. 7, n. 1, p. 77–91, Mar. 1952.

MARKOWITZ, H. et al. Computation of Mean-Semivariance Efficient Sets by the Critical Line Algorithm. **Annals of Operations Research**, v. 45, n. 1, p. 307–317, Dec. 1993.

MICHAUD, R. O. The Markowitz Optimization Enigma: Is "Optimized" Optimal. **Financial Analysts Journal**, v. 45, n. 1, p. 31–42, Jan.–Feb. 1989.

PARYS, B. P. V.; ESFAHANI, P. M.; KUHN, D. From data to decisions: Distributionally robust optimization is optimal. **Management Science**, INFORMS, v. 67, n. 6, p. 3387–3402, 2021.

PFLUG, G. **Some Remarks on the Value-at-Risk and the Conditional Value-at-Risk**. 1st. ed. [S.l.]: Springer, 2000. 272–281 p.

PFLUG, G.; WOZABAL, D. Ambiguity in Portfolio Selection. **Quantitative Finance**, v. 7, n. 4, p. 435–442, Aug. 2007.

RESCHENHOFER, E. et al. Evaluation of Current Research on Stock Return Predictability. **Journal of Forecasting**, v. 39, n. 2, p. 334–351, March 2020.

ROCKAFELLAR, R. T.; URYASEV, S. Conditional Value-at-Risk for General Loss Distributions. **Journal of Banking & Finance**, v. 26, n. 7, p. 1443–1471, Jul. 2002.

SCARF, H. **A Min-Max Solution of an Inventory Problem**. 1st. ed. [S.l.]: Stanford University Press, Stanford, CA, 1958. 201–209 p.

SHAFIEEZADEH-ABADEH, S.; ESFAHANI, P. M.; KUHN, D. Distributionally robust logistic regression. In: **Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1**. Cambridge, MA, USA: MIT Press, 2015. (NIPS'15), p. 1576–1584.

SHAFIEEZADEH-ABADEH, S.; KUHN, D.; ESFAHANI, P. M. Regularization via mass transportation. **Journal of Machine Learning Research**, v. 20, n. 103, p. 1–68, 2019.

SHAPIRO, A. Distributionally Robust Stochastic Programming. **SIAM Journal on Optimization**, v. 27, n. 4, p. 2258–2275, 2017.

SHAPIRO, A.; DENTCHEVA, D.; RUSZCZYNSKI, A. **Lectures on stochastic programming: modeling and theory**. [S.l.]: SIAM, 2021.

SHAPIRO, A.; NEMIROVSKI, A. On Complexity of Stochastic Programming Problems. **Continuous Optimization: Current Trends and Modern Applications**, v. 99, n. 1, p. 111–146, 2005.

SIVAPRASAD, P. T. et al. Optimizer benchmarking needs to account for hyperparameter tuning. In: PMLR. **International Conference on Machine Learning**. [S.l.], 2020. p. 9036–9045.

SPIERS, B.; WALLEZ, D. High-Performance Computing on Wall Street. **Computer**, v. 43, n. 12, p. 53–59, Dec. 2010.

STREET, A. **Equivalente Certo e Medidas de Risco em Decisões de Comercialização de Energia Elétrica**. Tese (Doutorado) — PUC-Rio, 2008.

STREET, A. On the Conditional Value-at-Risk Probability-Dependent Utility Function. **Theory and Decision**, v. 68, n. 1–2, p. 49–68, Feb. 2010.

TIBSHIRANI, R. Regression shrinkage and selection via the lasso. **Journal of the Royal Statistical Society. Series B (Methodological)**, [Royal Statistical Society, Wiley], v. 58, n. 1, p. 267–288, 1996. ISSN 00359246. Disponível em: <http://www.jstor.org/stable/2346178>.

WIESEMANN, W.; KUHN, D.; SIM, M. Distributionally robust convex optimization. **Oper. Res.**, INFORMS, Linthicum, MD, USA, v. 62, n. 6, p. 1358–1376, dez. 2014. ISSN 0030-364X.

WILSON, E. B. Probable Inference, the Law of Succession, and Statistical Inference. **Journal of the American Statistical Association**, v. 22, n. 158, p. 209–212, May 1927.

YANG, L.; SHAMI, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. **Neurocomputing**, v. 415, p. 295–316, 2020. ISSN 0925-2312. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925231220311693>.