Models and Efficient Algorithms for Convex Optimization under Uncertainty

Nam Ho-Nguyen

Tepper School of Business
Carnegie Mellon University
May 2019

Submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in Operations Research

Committee:
Fatma Kılınç-Karzan
Daniel Kuhn
Andrew Li
Javier Peña
Stephen Wright

Abstract*

Optimization is a key analytical technique used for quantitative decision-making in real-world problems. In practice, many situations call for decision-making in the face of incomplete knowledge and/or dynamic environments. Making high-quality decisions in these settings requires optimization techniques that are designed to account for uncertainty. Furthermore, as new technologies are developed, more complex higher-dimensional optimization models become prevalent. This dissertation examines various models for optimization under uncertainty, as well as efficient algorithms for solving such models that are scalable as the model size grows.

We study three models for optimization under uncertainty: robust optimization (RO), joint estimation-optimization (JEO), and joint prediction-optimization (JPO). Robust optimization accounts for inexact information by finding solutions, which remain feasible to all perturbations of inputs within a given uncertainty set. Joint estimation-optimization considers a dynamic setting where inputs are updated over time as new data is collected and converge to some ideal input that is not revealed to the modeller. Joint prediction-optimization considers the use of a prediction model to obtain optimization inputs from side information, an approach that is widely used amongst practitioners. The dissertation considers theoretical properties and algorithmic performance guarantees for these three models.

We first present a generic framework to derive primal-dual algorithms for both RO and JEO. Previously, algorithms for such models were derived in an ad-hoc manner, and analyzed on a case-by-case basis. Our framework considers both of these optimization under uncertainty models through a common lens of saddle point problems. By analyzing these, we highlight three quantities which directly bound the performance guarantees for our respective models, and show how regret minimization techniques from online convex optimization can be used to control these three quantities. Thus, our framework allows us to transfer regret bounds for these quantities into performance guarantees for the associated algorithms. Since regret minimization algorithms from online convex optimization are key to our framework, we also examine these, and in particular derive improved regret bounds for RO and JEO in the presence of favourable structure such as strong convexity and smoothness.

We show that a number of previous algorithms for both robust optimization and joint estimation-optimization can be derived from our unified framework. More importantly, our framework can be used to derive more efficient algorithms for both models in a principled manner. For robust optimization, our framework is used to derive algorithms that can drastically reduce the cost of iterative methods by replacing nominal oracles with cheaper first-order updates. For joint estimation-optimization, we derive algorithms for the non-smooth strongly convex setting, which has not been considered previously.

We demonstrate the use of our framework through two examples: robust quadratic programming with ellipsoidal uncertainty sets, and dynamic non-parametric choice model estimation. For robust quadratic programming, we analyze the trust-region subproblem (TRS). The TRS is the well-studied problem of minimizing a non-convex quadratic function over the unit ball, and it arises naturally in the context of robust quadratic constraints. We give a second-order cone based convexification of TRS which, in contrast to previous work, is still in the space of original variables. We then show how to apply this convexification to robust quadratic programming, and derive two efficient algorithms for it using our framework. We carry out a numerical study on robust portfolio

*This research was supported in part by NSF grant CMMI 1454548.

optimization problems, and the numerical results show improvement of our approach over previous approaches in the high-dimensional regime. We frame dynamic non-parametric choice model estimation as an instance of JEO. A particular challenge in this setting is the high-dimensionality of the resulting primal problem. Nevertheless, our generic primal-dual framework encompassing JEO applications is quite flexible and allows us to derive algorithms that can bypass this high dimensionality challenge. We test our approach for non-parametric choice estimation computationally, and highlight interesting trade-offs between data updating and convergence rates.

Finally, we give a joint analysis of prediction and optimization. A natural performance measure in this setting is the optimality gap. Unfortunately, it is difficult to directly tune prediction models using this performance measure due to its non-convexity. We thus characterize sufficient conditions under which the more common prediction performance measures arising in statistics/machine learning, such as squared error, can be related to the true optimality gap performance measure. We derive conditions on a performance measure that guarantee that the optimality gap will be minimized, and give an explicit relationship between the squared error and the optimality gap. Such conditions allow practitioners to choose prediction methods for obtaining optimization parameters in a more judicious manner.

Acknowledgements

First and foremost, I thank my advisor, Fatma Kılınç-Karzan. It is hard to put into words how valuable her guidance, patience and support has been throughout my PhD journey. I am very lucky to be advised by her, and I am truly grateful. I know that her advice and wisdom will stay with me throughout my career.

I thank the other members of my dissertation committee, Daniel Kuhn, Andrew Li, Javier Peña, and Stephen Wright, for their time and effort in reading my dissertation and providing valuable feedback on my work.

I thank all of the professors at Carnegie Mellon who I have had the privilege to learn from, and in particular the faculty within the Operations Research department at the Tepper School of Business. Before arriving in Pittsburgh, I knew next to nothing about operations research, and I have thoroughly enjoyed learning about OR under their guidance.

I thank Lawrence Rapp and Laila Lee for taking care of all the administrative needs of the PhD program at Tepper. Life would be a lot harder if it was not for their professionalism and generosity.

I thank the friends I've made throughout my time at CMU: Alex Kazachkov, Yang Jiao, Thiago Serra, Siddharth Singh, Anirudh Subramanyam, David Huck Gutman, Rijnard van Tonder, Amin Hosseininasab, Arash Haddadan, Franco Berbeglia, Neda Mirzaeian and Thomas Lavastida. Whether discussing research or just hanging out, it has always been a pleasure. Special thanks to my classmates and close friends Michael Anastos, Gerdus Benade, Ryo Kimura, Dabeen Lee, Stelios Despotakis and Christian Tjandraatmadja. I would not have been able to complete this PhD without their friendship.

I thank my family in Australia: my parents Luan Ho-Trieu and Dao Nguyen, my brother Khai The Ngo, and my sister-in-law Sarah Blackman. Their love and support throughout my journey has been a great source of comfort and inspiration. I would not be here without them. (Also, shout-out to my nephew Oscar Ngo, and my future niece/nephew! If you ever decide to do a PhD, I'll support you.)

Finally, I thank God, for all of the above.

Contents

Chapter 1

Introduction

Optimization is a key tool used for quantitative decision-making in real-world problems. The generic form of an optimization model is the following:

$$\min_x \left\{ f(x, u^0) : \ f^i(x, u^i) \leq 0, \ \forall i = 1, \ldots, m, \quad x \in X \right\}. \tag{1.1}$$

Here, x are *decision variable*s, and is required to belong to the domain X. The function $f(\cdot, u^0)$ is the *objective*, which quantitatively distinguishes between good and bad decisions. The functions $f(\cdot, u^i)$ are *constraint functions* which represent requirements on the decisions; in (1.1), the decisions are required to satisfy $f^i(x, u^i) \leq 0$. The $u^0, u^1, \ldots, u^m$ are *parameters*, which capture problem-specific information.

As an example, in portfolio optimization, where the goal is to invest an amount of wealth in certain assets to maximize return, a decision variable x denotes how much wealth to allocate each asset, the domain X consists of all allocations which fully invests the wealth, the objective is the return of the chosen portfolio x, and constraints may capture risk tolerances on the portfolios. In practice, however, computing the return and risk of a portfolio x requires knowing the parameters of the return distribution for the assets, but since we do not know the distribution exactly, the parameters are estimated from data.

The ever-increasing availability of data has introduced many opportunities to make better decisions using a 'data-driven' approach. At the same time, data remains inherently noisy, and this translates to noise on parameter estimates. Thus, optimization models which can incorporate parameter uncertainty are key analytical tools to effectively utilize data in decision-making processes. Furthermore, as new technologies are developed, more complex optimization models become prevalent. These two observations highlight the need for a better understanding of the challenges and capabilities of optimization under uncertainty, then leveraging this understanding to improve upon the state-of-the-art models and algorithms. This dissertation aims to address this need via two directions.

- The first direction is to develop scalable algorithms for optimization under uncertainty. Recently, first-order iterative algorithms have seen a resurgence due to the attractive property that they scale linearly with the number of decision variables. However, it is not clear how these can be applied to optimization problems which model parameter uncertainty, as these are usually much more complicated than deterministic problems. In this dissertation, we

develop a framework to solve optimization problems with uncertainty using first-order techniques. The result of this is a new class of algorithms that come with rigorous performance guarantees and scale well with the dimension of the problem.

- The second direction is to develop and analyse models that can mitigate parameter uncertainty as more data becomes available, thereby leading to improved decision-making. There are two main challenges here. First, how can decision-making frameworks capture dynamic information gathering? Specifically, can existing algorithms be applied to such problems with rigorous performance guarantees? (This is also related to the first direction.) Second, how can decision-making frameworks capture side information? Specifically, how can parameter prediction models from the statistics/machine learning literature be incorporated into optimization models, and what kind of guarantees do they admit?

1.1 State of the Art

There are several modelling techniques that have been developed to capture parameter uncertainty in optimization models. We give a brief summary of these.

Robust optimization (RO). This is one of the leading modeling paradigms for optimization problems under uncertainty. RO seeks a solution that is immunized against *all* possible realizations of uncertain model parameters from a given uncertainty set. Thus, to capture uncertainty in objective parameters, RO looks to optimize for the worst-case parameter u^i from a given uncertainty set U^i, i.e., we replace the constraint

$$f(x, u^i) \leq 0 \rightarrow \max_{u^i \in U^i} f(x, u^i) \leq 0.$$

Choosing an x which satisfies the new *robust* constraint ensures that the constraint will be satisfied for all possible $u^i \in U^i$, thus guarding against any potential deviations. A similar principle can be applied to the objective function.

RO is widely adopted in practice mainly because of its computational tractability. Since the literature is too vast to comprehensively cover, we refer the reader to the seminal paper by Ben-Tal and Nemirovski [16], the book by Ben-Tal et al. [23] and surveys Ben-Tal and Nemirovski [18, 19], Bertsimas et al. [29], Caramanis et al. [47] for a detailed account of RO theory and numerous applications. The primary method for handling a robust constraint is to reformulate it via duality theory into an equivalent without the maximum, called the *robust counterpart*. Under mild assumptions, this yields a convex and tractable robust counterpart problem which can then be solved using existing convex optimization software and tools. This has seen much success in decision-making applications, nevertheless it has a major drawback that the reformulated robust counterpart is often not as scalable as the deterministic nominal program. In particular, the robust counterpart can easily belong to a different class of optimization problems as opposed to the underlying original deterministic problem. For example, a linear program (LP) with ellipsoidal uncertainty is equivalent to a convex quadratic program (QP), and similarly, a conic-quadratic program with ellipsoidal uncertainty is equivalent to a semidefinite program (SDP). This then presents a critical challenge in applying RO methodology in high-dimensional applications. To address this drawback, iterative schemes have been developed by Mutapcic and Boyd [107], Ben-Tal et al. [25] that alternate between the generation/update of candidate decisions x and parameter realizations u^i by solving instances of an optimization model with constraint structures similar to (or the same as) the nominal model.

Joint estimation-optimization (JEO). This is a relatively newer paradigm introduced by Jiang and Shanbhag [91, 92] and Ahmadi and Shanbhag [5] under the name *misspecified optimization*, which we call JEO. This considers the setting where the parameters u are unknown, but are approximated through a converging sequence of parameters $u_t \to u$. In many practical situations, JEO is solved by choosing a sufficiently close u_t then solving the optimization problem with the chosen parameters u_t. With such a strategy, under mild continuity assumptions, the gap between the 'true' optimal solution (with true parameters u) and the one obtained via u_t will be proportional to the distance between u_t and u. Nevertheless, this creates the following 'inconsistency' problem: even optimizing to full accuracy on parameters u_t does not solve the true problem with parameters u. A naïve scheme to achieve consistency is to simply re-optimize our decision for each u_t. This naïve scheme comes with two disadvantages: each step t involves solving a complete minimization problem up to some accuracy, and furthermore the accuracy must improve at each new step. The main problem is that at each step t, the information from the previous steps cannot be utilized, hence they are essentially wasted. To address this, Jiang and Shanbhag [91, 92] and Ahmadi and Shanbhag [5] propose some algorithms to efficiently generate a sequence of points x_t which converge to the true optimum of the problem. In particular, their scheme can exploit previous information in a principled manner by ensuring that the effort in each step consists only of first-order updates.

Joint prediction and optimization (JPO). This is a line of research that, as the name suggests, looks at techniques and/or analyses that combine the steps of prediction and optimization. Specifically, the *prediction* step aims to incorporate side information in order to more accurately estimate model parameters via training a prediction model. The process of training a prediction model from data, and deriving performance guarantees, is well-studied in statistics and machine learning domains. Optimization and statistics/machine learning have long had close relationships. However, most research has focused on optimization techniques for statistics/machine learning problems. In contrast, JPO aims to explore how statistics/machine learning affects the decision-making process through the optimization models.

To our knowledge, the earliest known work to jointly consider a prediction problem and an optimization problem in a systematic way is Kao et al. [96], where the use of least squares regression in parameter prediction for quadratic programming is analysed. Hannah et al. [72], Hanasusanto and Kuhn [71], Bertsimas and Kallus [26], Ban and Rudin [9], Bertsimas and Van Parys [28], Ho and Hanasusanto [79] all analysed the use of density estimation techniques to incorporate side information into the optimization problem. In contrast, Donti et al. [55], Elmachtoub and Grigas [58] explore training a prediction model using loss functions defined to better capture the subsequent optimization performance. This is most closely related to the work presented in this dissertation. We also mention that Bertsimas and Kallus [26], Ban and Rudin [9] also explored predicting the decision directly from the side information, rather than the parameters, and give some convergence guarantees for this scheme.

Stochastic programming [33, 136]. While stochastic programming is a fascinating technique with many successful applications, our dissertation does not focus on it. However, since it is the most widely-used method to capture parameter uncertainty, we mention it briefly. Stochastic programming models the parameters as random variables, and hence the objective and constraints in (1.1) are replaced by expectations:

$$f(x, u^0) \to \mathbb{E}_{u^0 \sim \mathbb{P}^0}[f(x, u^0)], \quad f^i(x, u^i) \to \mathbb{E}_{u^i \sim \mathbb{P}^i}[f^i(x, u^i)].$$

The the goal of stochastic programming is to optimize the decisions x given distributional knowledge $\mathbb{P}^0, \mathbb{P}^1, \ldots, \mathbb{P}^m$, or more commonly, given only access to samples from these distributions. Of course, instead of expectation, we may instead utilize other risk measures [129] (such as conditional value-at-risk) which capture risk aversion attitudes of decision-makers. When only samples are available instead of full distributional knowledge, these samples are used to approximate the true distribution. However, a common downside is that the empirical approximation can often be poor, in particular in the low sample regime. A technique to mitigate against this is distributionally robust optimization [149], where objective and/or constraint functions are replaced in the following manner:

$$f(x,u) \to \max_{\mathbb{P} \in \mathcal{P}(\hat{\mathbb{P}})} \mathbb{E}_{u \sim \mathbb{P}}[f(x,u)].$$

Here, $\hat{\mathbb{P}}$ is the empirical distribution of the samples, and $\mathcal{P}(\hat{\mathbb{P}})$ is an ambiguity set of distributions containing $\hat{\mathbb{P}}$. In other words, the function is the worst-case expected value amongst all possible distributions which are reasonably close to the empirical distributions. Distributionally robust optimization is similar in spirit to RO described above, with the key difference being that the 'parameters' are now distributions, not vectors. However, the methodology is very similar to RO in the use of duality theory, with appropriate accommodations for the infinite-dimensional nature.

1.2 Notation

We denote the real numbers as $\mathbb{R}$, the non-negative reals as $\mathbb{R}_+$, and the positive integers as $\mathbb{N}$. Given $n \in \mathbb{N}$, we denote $[n] := \{1, \ldots, n\}$, and $\Delta_n := \left\{x \in \mathbb{R}^n : x \geq 0, \sum_{i \in [n]} x_i = 1\right\}$ to be the $(n-1)$-dimensional simplex. We denote by $\mathbf{1}_n \in \mathbb{R}^n$ to be the vector of all ones, and $e_i \in \mathbb{R}^n$ as the ith standard basis vector, for $i \in [n]$. Given a set Z in a real vector space, we denote its convex hull by $\text{Conv}(Z)$.

Given a function $f : \mathbb{R}^n \to \mathbb{R}$ and a domain $X \subset \mathbb{R}^n$, we denote the set of minimizers as $\arg\min_{x \in X} f(x) = \{x \in X : f(x) = \inf_{x' \in X} f(x')\}$. Given a convex function $f : X \to \mathbb{R}$, we denote by $\partial f(x)$ to be the set of subgradients of f at the point $x \in X$. We abuse notation slightly by denoting $\nabla f(x)$ to be a selection of one such subgradient from $\partial f(x)$. Note that if f is differentiable, this selection is unique.

1.3 Robust Optimization

Consider the deterministic optimization problem (1.1), where $X \subset \mathbb{R}^n$ is closed and convex, the functions $f^0(x)$ and $f^i(x, u^i)$ for $i \in [m]$ are convex functions of x, and $u = (u^1, \ldots, u^m)$ is a fixed parameter vector. Without loss of generality we assume the objective function $f^0(x)$ does not have uncertainty. This is a convex optimization problem, for which there are many known methods to solve. When parameter uncertainty is present, we can formulate the *robust convex optimization problem* associated with (1.1):

$$\text{Opt} := \min_x \left\{ f^0(x) : \ x \in X, \quad \max_{u^i \in U^i} f^i(x, u^i) \leq 0, \ \forall i \in [m] \right\}, \tag{1.2}$$

where $U^1, \ldots, U^m$ are the *uncertainty sets* given for the parameter u^i of constraint $i \in [m]$. Because we assume formulation (1.1) is convex, the overall optimization problem in (1.2) is also convex. Without loss of generality, we assume that the uncertainty set has a Cartesian product form $U^1 \times \ldots \times U^m$ (see e.g., [18]; we let $U = U^1 \times \ldots \times U^m$ and write $u = (u^1, \ldots, u^m) \in U$. We do not

further assume that the sets U^i are convex. However, for some algorithms we consider, convexity of U^i for $i \in [m]$ will be required.

A convex optimization problem can be solved by solving a polynomial number of associated feasibility problems in a standard way, via a binary search over its optimal value. In particular, let $[\underline{v}_0, \overline{v}_0]$ be an initial interval containing the optimal value of (1.2). At each iteration k of the binary search, we update the domain $X_k := X \cap \{x : \ f^0(x) \leq v_k\}$ for some $v_k \in [\underline{v}_k, \overline{v}_k]$ and arrive at the following robust feasibility problem:

$$\text{find} \quad x \in X_k \quad \text{s.t.} \quad \max_{u^i \in U^i} f^i(x, u^i) \leq 0 \quad \forall i \in [m]. \tag{1.3}$$

Then based on the feasibility/infeasibility status of (1.3), we update our range $[\underline{v}_{k+1}, \overline{v}_{k+1}]$ and go to iteration $k+1$. In this scheme, we are guaranteed to find a solution $x^* \in X$ whose objective value is within $\delta > 0$ of the optimum value of (1.2) in at most $\left\lfloor \log_2 \left(\frac{\overline{v}_0 - \underline{v}_0}{\delta} \right) \right\rfloor$ iterations. Therefore, one can equivalently study the complexity of solving robust feasibility problem (1.3) as opposed to (1.2). From now on, we focus on solving robust feasibility problem and assume that the constraint on the objective function $f^0(x)$ is already included in the domain X for simplicity in our notation.

Given functional constraints $f^i(x) \leq 0$, $i \in [m]$, most convex optimization methods will declare infeasibility or return an approximate solution $x \in X$ such that $f^i(x) \leq \epsilon$ for $i \in [m]$ for some tolerance level $\epsilon > 0$. Therefore, we consider the following *robust approximate feasibility problem*:

$$\begin{cases} \textit{Either}\text{: find} \quad x \in X \quad \text{s.t.} \quad \max_{u^i \in U^i} f^i(x, u^i) \leq \epsilon \quad \forall i \in [m]; \\ \textit{or}\text{: declare infeasibility, } \forall x \in X, \ \exists i \in [m] \quad \text{s.t.} \quad \max_{u^i \in U^i} f^i(x, u^i) > 0. \end{cases} \tag{1.4}$$

We refer to any feasible solution x to (1.4), i.e., $x \in X$ such that $\sup_{u^i \in U^i} f^i(x, u^i) \leq \epsilon$ holds for all $i \in [m]$ as a *robust ϵ-feasibility certificate*. Similarly, any realization of the uncertain parameters $\bar{u} \in U$ such that there exists no $x \in X$ satisfying $f^i(x, \bar{u}^i) \leq 0$ for all $i \in [m]$ is referred to as a *robust infeasibility certificate*.

1.4 Joint Estimation-Optimization

Joint estimation-optimization (JEO) considers the setting where we have data u in the objective $f(x, u)$, and that the 'correct' data value u is unknown to us, but that we have access to improving estimates $u_t \to u$. More precisely, JEO aims to solve

$$\min_x \{f(x, u) : \ x \in X\} \tag{1.5}$$

$$\text{given only } \{u_t\}_{t \geq 1}, \ \text{where } u_t \to u. \tag{1.6}$$

This means that we wish to generate a sequence $\{x_t\}_{t \geq 1} \subset X$ such that each x_t depends only on $\{x_s, u_s : s \in [t-1]\}$ and

$$f(x_t, u) \to \min_{x \in X} f(x, u).$$

In many practical situations, JEO is solved via a *sequential* one-step method: first estimate $u_t \approx u$, then solve (1.5) with u_t in place of u. With such a strategy, under mild continuity assumptions, the accuracy of (1.5) is controlled by the norm of $\|u_t - u\|$. Nevertheless, this creates the following 'inconsistency' problem: when minimizing $f(x, u_t)$, we create a sequence of points $x_t \in X$, $t \geq 1$, which converge to the minimum of $f(x, u_t)$; however, the sequence will not converge to the desired minimum (1.5), and in fact (under mild continuity conditions) will only be within $O(\|u_t -$

$u^*\|$) accuracy. That is, a sequential one-step approach cannot guarantee asymptotically accurate solutions x_t.

It is possible to achieve consistency if we obtain x_t by minimizing $f(x, u_t)$ up to accuracy $O(\|u_t - u^*\|)$, which will guarantee that $\lim_{t\to\infty} f(x_t, u_t)$ converges to the optimum value of (1.5). This naïve scheme comes with two disadvantages: each step t involves solving a whole minimization problem up to some accuracy, and furthermore the accuracy must improve at each new step. The main problem is that at each step t, the information from the previous steps cannot be utilized, hence they are essentially wasted. In JEO, we are interested in methods that update x_t in an efficient manner, e.g., through only gradient updates.

1.5 Joint Prediction and Optimization

We consider the convex program

$$\min_x \left\{ f(x) + c^\top x : x \in X \right\} \tag{1.7}$$

where $X \subset \mathbb{R}^m$ is a convex and compact domain, and the vector c is not known exactly, but instead is governed via covariates w. More precisely, suppose the covariates w belong to a set W, and the cost vectors belong to a set $C \subseteq \mathbb{R}^m$, both of which live in Euclidean spaces. In this setting, we have access to the observations $\{(w_i, c_i) : i \in [n]\}$ where $[n] := \{1, \ldots, n\}$, and we need to solve (1.7) for c_{n+1} yet we are only given information of w_{n+1}. We assume that the (w_i, c_i) are realizations of i.i.d. random variables according to some distribution $\mathbb{P}$ on $W \times C$. Furthermore, we also assume that the covariate w and a corresponding 'true' cost vector c is drawn according to $\mathbb{P}$. In this setting, first based on the observations $\{(w_i, c_i) : i \in [n]\}$, a prediction model resulting in the estimation of a function $g : W \to C$ is built to capture the dependency of c on w and then when given a covariate w, the problem (1.7) is solved with c being replaced with the prediction $g(w)$.

To assess the quality of using $g(w)$ in place of c in (1.7), we define the loss as the optimality gap of the solution obtained with $g(w)$ on the true objective vector c. More precisely, the quality of the prediction $d = g(w)$ with respect to (1.7) is given by the *true loss function*

$$L(d, c) := f(x^*(d)) + c^\top x^*(d) - \min_{x\in X} \left\{ f(x) + c^\top x \right\}. \tag{1.8}$$

Note that given any c, $L(d, c) \geq 0$ for all d and $L(c, c) = 0$. Since (w, c) is randomly drawn from $\mathbb{P}$, we assess the performance of a function $g : W \to C$ in terms of the expected true loss, i.e., the risk

$$R(g, \mathbb{P}) := \mathbb{E}[L(g(w), c)]. \tag{1.9}$$

The best possible risk we can achieve is

$$R(\mathbb{P}) := \inf_g \{R(g, \mathbb{P}) : g \text{ measurable}\}. \tag{1.10}$$

In this setting, we wish to study various ways to construct prediction functions $\hat{g}_n : W \to C$ from n data points to ensure that $R(\hat{g}_n, \mathbb{P}) \to \inf_g R(g, \mathbb{P})$ as $n \to \infty$. The challenge here is that L is non-convex in $g(w)$, hence traditional learning techniques based on empirical risk minimization becomes difficult. For joint prediction and optimization (JPO), we investigate constructing the $\hat{g}_n$ using traditional techniques from machine learning and/or statistical inference, which only focus on predicting c as accurately as possible from w, and not necessarily taking into account the optimization problem (1.7). However, it is not clear that focusing solely on prediction will give the desired property of $R(\hat{g}_n, \mathbb{P}) \to \inf_g R(g, \mathbb{P})$, thus we are interested in understanding which of these traditional prediction techniques are able to admit such guarantees.

1.6 Outline and Contributions

In Sections 1.3, 1.4 and 1.5, we described RO, JEO and JPO in a precise manner. In this dissertation, we make contributions to these three paradigms along the two directions of analysing models and developing algorithms described above. These contributions can be summarized as follows.

- Chapter 2 presents a primal-dual framework for deriving algorithms, together with performance guarantees, for RO and JEO. The key step is to analyse a unified saddle point representation of the RO and JEO models. The idea of analysing RO from a min-max saddle point view was first published in our paper [82]. In Chapter 2.4, we present a slightly different analysis of RO in our more general framework, but this does not fundamentally change our complexity results from [82], which we also present. In Chapter 2.5, we present the application of our framework for JEO. In JEO, the quality of our decision depends on the closeness of the parameter estimates u_t to the true parameter u. We show that as long as $u_t \to u$, then convergence holds, and present convergence rates under various structural assumptions. The material presented here is an amalgamation of ideas from the papers [80, 83]. The results and implications are unchanged from these papers, but the derivation is presented under the unified primal-dual framework.

- Chapter 3 presents results on online convex optimization (OCO), which is a key tool for deriving algorithms in our primal-dual framework from Chapter 2. OCO is used in many other contexts in machine learning, and is often studied under certain restrictions. When used in our primal-dual framework, some of these restructions can be relaxed (namely, using non-uniform weights and 1-lookahead). We derive new algorithms for OCO under the relaxed settings, and show how they lead to accelerated regret bounds under certain structural assumptions such as strong convexity and smoothness. The results presented here are from our papers [80, 84].

- Chapter 4 analyses the trust region subproblem (TRS), the problem of minimizing a non-convex quadratic over a unit ball domain, and its applications to robust quadratic programming. The main body of results on the TRS in Chapter 4 is from our paper [81]. We study the geometry of the TRS with additional conic constraints, and identify that the key factor to understand this is to study how the conic constraints interact with the non-convex quadratic term. We use this to present convexification results for the TRS under certain conditions on the conic constraints and the minimum eigenvalue of the quadratic. We also present explicit descriptions for the convex hull of the epigraph under slightly stronger conditions. Our convexification results generalize previous ones from the literature, and using them we are able to provide the best-known complexity guarantees for solving the TRS. We apply our convexification results to robust quadratic programming, done in the paper [82], and derive efficient algorithms for this via the primal-dual framework of Chapter 2 and the OCO algorithms of Chapter 3. We conduct a numerical study, comparing with previous RO methods, and demonstrate the superior scalability of our framework. This is in line with the complexity analysis for RO from Chapter 2.4.

- Chapter 5 presents our work from [83] on non-parametric choice model estimation, which is the problem of inferring a choice model from observational choice data which can explain past and predict future choice behaviour. Existing work on this has resulted in heuristic algorithms without rigorous performance guarantees. Furthermore, none of the existing work takes into

account the very natural dynamic data setting where parameters are updated as more data (in this case, choice observations) is collected. This problem admits, in a natural way, a JEO representation, thus our primal-dual framework from Chapter 2 can readily be applied to derive algorithms to solve it. However, due to the nature of the problem, the primal domain in general has exponentially many decision variables, thus some care must be taken in the design of algorithms for this problem. We discuss such considerations, derive appropriate algorithms, and give rigorous performance guarantees. Our computational study compares and contrasts different algorithms derived from our framework, and highlight tradeoffs between convergence rate, data updating rate, and estimated model sparsity.

- Finally, Chapter 6 presents new developments on establishing relationships between the performance of prediction models used to obtain parameters, and the subsequent optimization performance resulting from using these models. Considering convex optimization problems over bounded domains, we first establish conditions on the prediction error function which guarantee that solving the prediction problem will also result in minimizing the optimality gap of the subsequent optimization problem. Unfortunately, the exact relationship depends on the data distribution, which is not available for most learning problems. Thus, this first result does not provide insight into rates of convergence between these quantities, and should be thought of as an asymptotic guarantee. Therefore, we also establish stronger distribution-independent conditions which allow us to give explicit non-asymptotic guarantees on the prediction performance and the optimality gap. However, for most prediction error functions, computing this explicit relationship is difficult, so we focus on a specific prediction error function where this is possible, namely the squared error function. We give a distribution-independent guarantee on the relationship between prediction error obtained using the squared error function and the optimality gap.

Chapter 2

Primal-Dual Framework for Convex Optimization under Uncertainty

2.1 Introduction

In this chapter, we present a general primal-dual algorithmic framework for optimization under uncertainty, with particular focus on robust optimization (RO) and joint estimation-optimization (JEO). The algorithmic framework can be used to derive first-order algorithms under a variety of structural assumptions.

At the heart of our analysis is the saddle point (SP) problem:

$$\mathrm{SV}(u) := \min_{x \in X} \max_{y \in Y} \Psi(x, y; u). \tag{2.1}$$

Any convex optimization problem has an equivalent representation as a SP problem. Here, x is typically referred to as the primal variables, and y as the dual variables. Importantly, u here is input data for our convex optimization problem. We give examples of two such problems, which we will revisit in the context of RO and JEO later.

Example 2.1. Consider the convex feasibility problem:

$$\begin{cases} \textit{Either}\text{: find } \; x \in X \quad \text{s.t.} \quad f^i(x, u^i) \leq \epsilon \quad \forall i \in [m]; \\ \textit{or}\text{: declare infeasibility, } \forall x \in X, \; \exists i \in [m] \quad \text{s.t.} \quad f^i(x, u^i) > 0. \end{cases} \tag{2.2}$$

Here, $f^i(x, u^i)$ are constraint functions which are convex in x, and $u^i, i \in [m]$ are problem data. Defining

$$u := (u^1, \ldots, u^m), \quad \Psi(x, y; u) := \sum_{i \in [m]} y^i f^i(x, u^i), \quad Y := \Delta_m,$$

(2.2) is equivalent to

$$\min_{x \in X} \max_{y \in Y} \Psi(x, y; u) \leq \epsilon \quad \text{or} \quad \min_{x \in X} \max_{y \in Y} \Psi(x, y; u) > 0.$$

∎

In other words, we can evaluate the saddle point to solve (2.2).

Example 2.2. Consider a convex optimization problem with only objective function parameters:

$$\min_{x \in X} f(x, u). \tag{2.3}$$

A saddle point representation always exists through convex conjugacy. More precisely, assuming that $f(\cdot, u)$ is a closed convex function for each u, then

$$f^*(y, u) := \max_x \{\langle y, x \rangle - f(x, u)\}, \text{ hence } f(x, u) = \max_{y \in Y} \{\langle x, y \rangle - f^*(y, u)\} \tag{2.4}$$

$$f(x, u) = \max_{y \in Y} \Psi(x, y; u), \quad \Psi(x, y; u) = \langle x, y \rangle - f^*(y, u), \tag{2.5}$$

$$Y = \{y : \|y\| \leq G\}, \text{ where } G \text{ satisfies } \|\nabla_x f(x, u)\| \leq G \ \forall x, u. \tag{2.6}$$

Note that G is finite if, for example, f is uniformly Lipschitz continuous in x for all possible u. Then (2.3) is equivalent to the saddle point:

$$\min_{x \in X} f(x, u) = \min_{x \in X} \max_{y \in Y} \Psi(x, y; u).$$

■

In deterministic optimization, we are given u, and solve (2.1) using appropriate deterministic methods. In optimization under uncertainty, it is less straightforward what we mean by 'solving (2.1)'. In some situations, there may be a 'true' set of parameters u which we only have approximate knowledge of. As we will see in Section 2.5, this is relevant for JEO. In other situations, there may be no 'true' u, but instead we wish to say something about a given collection of parameters U, which in turn give rise to one such instance (2.1) for each $u \in U$. This is relevant for RO, as we will see in Section 2.4. In Section 2.3, however, we will consider the setting with some true u, and solving (2.1) for this u. While this is not accurate for our discussion on RO in Section 2.4, the machinery that we build in Section 2.3 will transfer to the setting of Section 2.4 easily.

2.1.1 Related Literature

Saddle point problems of the form (2.1) are well-studied in the literature; see [97] for a comprehensive account. However, we wish to highlight three particular references. Arrow et al. [8] introduced the well-known Arrow-Hurwicz-Uzawa subgradient method to solve saddle point problems, which has been extensively studied in the optimization and control literature. The method is a primal-dual method, which simultaneously updates primal varianbes x and dual variables y using subgradient descent/ascent. Nedić and Özdağlar [110] apply this idea to Lagrangian relaxations of convex programs, and derive constraint violation and suboptimality bounds for the subgradient method. Yu and Neely [154] examine the same Lagrangian relaxation under smoothness assumptions, and suggest a similar primal-dual algorithm with improved error bounds. Our paper [84] unifies these different algorithms by presenting a generic framework to derive primal-dual algorithms through bounding certain regret quantities. On a related note, Abernethy and Wang [3], Wang and Abernethy [148], Abernethy et al. [2] look at the saddle point problem formed via the Fenchel conjugate (similar to Example 2.2) and show that several known algorithms such as Frank-Wolfe and accelerated gradient descent can be recovered by bounding certain regret quantities associated with the saddle point gap.

Our framework in this chapter has applications to robust optimization (RO) and joint estimation-optimization (JEO), defined in Chapters 1.3 and 1.4 respectively, providing techniques to build algorithms that solve these. The primary method for solving a (convex) RO problem is to transform

it into an equivalent deterministic problem via duality theory, which was introduced in the seminal paper by Ben-Tal and Nemirovski [16]. Under mild assumptions, this yields a convex and tractable robust counterpart problem [23, 29, 24] which can then be solved using existing convex optimization software and tools. This traditional approach has seen much success in decision making domain, nevertheless it has a major drawback that the reformulated robust counterpart is often not as scalable as the deterministic nominal program. In particular, the robust counterpart can easily belong to a different class of optimization problems as opposed to the underlying original deterministic problem. For example, a linear program (LP) with ellipsoidal uncertainty is equivalent to a convex quadratic program (QP), and similarly, a conic-quadratic program with ellipsoidal uncertainty is equivalent to a semidefinite program (SDP) (see e.g., [23, 29]).

The iterative schemes that alternate between the generation/update of candidate solutions and the realizations of noises offer a convenient remedy to the scalability issues associated with the robust counterpart approach. Thus far, such approaches of Mutapcic and Boyd [107], Ben-Tal et al. [25] have relied on two oracles: (*i*) *solution oracles* to solve instances of extended (or nominal) problems with constraint structures similar to (or the same as) the deterministic problem, and (*ii*) *noise oracles* to generate/update particular realizations of the uncertain parameters. At each iteration of these schemes, both solution and noise oracles are called, and their outputs are used to update the inputs of each other oracle in the next iteration. Because solution oracles rely on a solver of the same class capable of solving the deterministic problem, these iterative approaches circumvent the issue of the robust counterpart approach potentially relying on a different solver. Nevertheless, these iterative approaches still suffer from a serious drawback: the solution oracles in Mutapcic and Boyd [107], Ben-Tal et al. [25] themselves can be expensive as they require solving extended or nominal optimization problems completely. While solving the nominal problem is not as computationally demanding as solving the robust counterpart, the overall procedure depending on repeated calls to such oracles can be prohibitive. In fact, each such call to a solution oracle may endure a significant computational cost, which is at least as much as the computational cost of solving an instance of the deterministic nominal problem.

Jiang and Shanbhag [91, 92] introduced and studied the JEO problem in a stochastic setting, and Ahmadi and Shanbhag [5] examined the deterministic case. We consider the deterministic JEO problem, for which Ahmadi and Shanbhag [5] provided some remarkable convergence results. Specifically, in the setting when f is strongly convex and smooth, [5, Proposition 3] states that a gradient descent-type algorithm is given with error bound of $O(\beta^T)$ after T iterations, for some $0 < \beta < 1$. In [5, Proposition 4] it is shown that, when f is only convex, the same algorithm (with different tuning parameters) achieves an error bound of $O(1/T)$. Furthermore, when f does not enjoy strong convexity or smoothness, [5, Proposition 6] provides an error bound of $O(1/\sqrt{T})$.

2.1.2 Contributions

While primal-dual algorithms for convex optimization have been studied extensively, in contrast to Nedić and Özdağlar [110], Yu and Neely [154], Ho-Nguyen and Kılınç-Karzan [84] the novelty of this chapter is to provide a principled framework for deriving such algorithms which also takes into account uncertainty in parameters u. More specifically, instead of u, we will consider an evolving sequence $\{u_t\}_{t\geq 1}$, which is generated or given to us in some fashion that is determined by the sense of the uncertainty in the problem. We also iteratively generate corresponding sequences $\{x_t, y_t\}_{t\geq 1}$ of primal-dual variables based on the sequence $\{u_t\}_{t\geq 1}$. These sequences will then be related to the SP problem at hand. We make this more precise in the coming sections.

As mentioned above, this framework can be used to derive algorithms for RO and JEO, which

also has a number of implications for these problems specifically. For RO, these are as follows.

- We obtain a general and flexible framework for iteratively solving robust feasibility problems, and demonstrate its flexibility by describing it as a meta-template. By customizing our framework appropriately, we modify the pessimization oracle-based approach of Mutapcic and Boyd [107] by replacing the extended nominal solver called at each iteration with more efficient subgradient-based. We demonstrate that this has both a much better bound on the number of oracle calls, and in Chapter 4.4.1 that it has superior numerical performance. We also provide a new interpretation of the nominal feasibility oracle-based approach of Ben-Tal et al. [25] as a special case within our framework, and extend the analysis of the approach of Ben-Tal et al. [25] under the same assumptions, e.g., access to a nominal optimization oracle, and show that it can solve the robust optimization problem directly without relying on a binary search.

- In contrast to the approaches of Mutapcic and Boyd [107] and Ben-Tal et al. [25], which rely on full nominal oracles to generate points, we can use our framework to derive algorithms which only rely on simple update rules in each iteration and thus has much lower per-iteration cost. This approach is designed for better performance as dimension of the problem increases, and in Chapter 4.4.1 we verify its scalability numerically.

- We demonstrate that the iteration complexity of our algorithms is at least as good as that of the efficient approach of [25], and better than the exponential complexity of [107]. Overall, our OFO-based approach leads to computational savings over the approach of [25] by a factor as large as $O(1/(\epsilon^2 \log(1/\epsilon)))$ arithmetic operations. Moreover, our iteration complexity is (almost) independent of both the number of robust constraints and the dimension of the deterministic problem. For further comparisons and discussion, see Section 2.4.1.3. In addition, our framework is amenable to exploiting favorable structural properties of the constraint functions such as strong concavity, smoothness, etc., through which better convergence rates can be achieved.

For JEO, in addition to recovering the standard results from Ahmadi and Shanbhag [5] in a unified manner, our framework allows us to derive, in a principled manner, an algorithm for the setting when f is non-smooth and strongly convex, which was not explored previously. In this setting, we provide an improved convergence rate of $O(1/T)$, which is the optimal rate even if we had the 'correct' data upfront.

2.2 Overview of Saddle Point Problems

To begin, let us state the following basic assumption that our SP problem (2.1) is part of the class of convex-concave minimax problems, which will be satisfied for all examples of interest.

Assumption 2.3. The domains X and Y are nonempty closed convex sets in Euclidean spaces $\mathbb{R}^{n_x}$ and $\mathbb{R}^{n_y}$ respectively, and at least one of X and Y is bounded. Furthermore, for any u, $\Psi(x, y; u)$ is convex in x and concave in y.

For the rest of the chapter, we assume that Assumption 2.3 is satisfied.

Saddle point (SP) problems play a vital role in our developments. Any convex-concave SP

problem (2.1) gives rise to two convex optimization problems that are dual to each other:

$$\mathrm{Opt}(P;u) = \min_{x\in X}[\overline{\Psi}(x;u) := \max_{y\in Y}\Psi(x,y;u)] \tag{2.7}$$

$$\mathrm{Opt}(D;u) = \max_{y\in Y}[\underline{\Psi}(y;u) := \min_{x\in X}\Psi(x,y;u)] \tag{2.8}$$

with $\mathrm{Opt}(P;u) = \mathrm{Opt}(D;u) = \mathrm{SV}(u)$ (this is guaranteed by the minimax theorem [137]). It is well-known that the solutions to (2.1) — the saddle points of $\Psi(\cdot,\cdot;u)$ on $X\times Y$ — are exactly the pairs $[x;y]$ formed by optimal solutions to the problems $\mathrm{Opt}(P;u)$ and $\mathrm{Opt}(D;u)$.

We quantify the accuracy of a candidate solution $[\bar{x},\bar{y}]$ to SP problem (2.1) with the *saddle point (SP) gap* defined as the sum of the optimality gaps for $\mathrm{Opt}(P;u)$ and $\mathrm{Opt}(D;u)$:

$$\begin{aligned}\epsilon_{\mathrm{sad}}^{\Psi}(\bar{x},\bar{y};u) &:= \max_{y\in Y}\Psi(\bar{x},y;u) - \min_{x\in X}\Psi(x,\bar{y};u) \\ &= \underbrace{\left[\max_{y\in Y}\Psi(\bar{x},y;u) - \mathrm{Opt}(P;u)\right]}_{\geq 0} + \underbrace{\left[\mathrm{Opt}(D;u) - \min_{x\in X}\Psi(x,\bar{y};u)\right]}_{\geq 0}.\end{aligned} \tag{2.9}$$

Note that when Ψ comes from some convex optimization problem (such as Examples 2.1 or 2.2), bounds on the SP gap give corresponding guarantees for the convex optimization problem of interest. We will examine this more precisely when we revisit RO and JEO.

Because convex-concave SP problems are simply convex optimization problems, they can in principle be solved by polynomial-time interior point methods (IPMs). However, the computational complexity of such methods depends heavily on the dimension of the problem. Thus, scalability of resulting algorithms becomes an issue in large-scale applications. As a result, for large-scale SP problems, one has to resort to first-order subgradient-type methods. On a positive note, there are many efficient first-order methods (FOMs) for convex-concave SP problems. These in particular include Nesterov's accelerated gradient descent algorithm [116] and Nemirovski's Mirror-Prox algorithm [112], both of which bound the saddle point gap at a rate of $\epsilon_{\mathrm{sad}}^{\Psi}(\bar{x}_T,\bar{y}_T;u) \leq O\left(\frac{1}{T}\right)$ where $\bar{x}_T,\bar{y}_T$ are solutions obtained after T iterations.

2.3 Primal-Dual Framework

Recall that our goal is to solve the SP problem (2.1) in the setting where the data vector u is unknown. As mentioned before, the general strategy is as follows: generate or use a given 'approximate' data sequence $\{u_t\}_{t\geq 1}$ and a corresponding primal-dual sequence $\{x_t,y_t\}_{t\geq 1}$. We will then try to quantify the SP gap $\epsilon_{\mathrm{sad}}^{\Psi}(\bar{x}_T,\bar{y}_T;u)$, where the points $\bar{x}_T,\bar{y}_T$ are built by aggregating the first T points of the primal-dual sequence in some manner, and u is the true, but unknown, data vector. Any iterative method for solving (2.1) would generate a primal-dual sequence in some form and build a final solution from some aggregation, so this part of the strategy is quite natural. On the other hand, generating an approximate data sequence $\{u_t\}_{t\geq 1}$ is perhaps a more novel aspect, but one can argue that it is the most natural strategy given no knowledge of u. However, there are two important questions to answer in order to implement this strategy:

(1) Where do the approximate data vectors u_t come from? If we are generating them, how should we do so?

(2) How do we choose the primal-dual points x_t, y_t?

Whether we generate $\{u_t\}_{t\geq 1}$ depends very much on how we model the uncertainty on u. We will re-examine this in the next two sections, when we apply our framework to the problems of interest, RO and JEO. On the other hand, one useful feature of our framework is that the method to generate the primal-dual sequence $\{x_t, y_t\}_{t\geq 1}$ is agnostic to how we model the uncertainty on u (although, it does depend on the sequence $\{u_t\}_{t\geq 1}$ itself, but not how it is obtained). To understand why this is possible, we present a fundamental theorem relating the sequences $\{u_t\}_{t\geq 1}, \{x_t, y_t\}_{t\geq 1}$ to the SP problem SV(u).

Theorem 2.4. *Let $\{\theta_t\}_{t\geq 1}$ be an arbitrary sequence of positive real numbers. Define*

$$\Theta_T := \sum_{t\in[T]} \theta_t, \quad \bar{x}_T^\theta := \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t x_t, \quad \bar{y}_T^\theta := \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t y_t.$$

Then, for any $T \geq 1$,

$$\epsilon_{\text{sad}}^\Psi\left(\bar{x}_T^\theta, \bar{y}_T^\theta; u\right) \leq \widehat{\epsilon}\left(\{x_t, y_t, u_t, \theta_t\}_{t\in[T]}\right) + \epsilon^\circ\left(\{x_t, u_t, \theta_t\}_{t\in[T]}\right) + \epsilon^\bullet\left(\{y_t, u_t, \theta_t\}_{t\in[T]}\right),$$

where

$$\begin{aligned}
\widehat{\epsilon}\left(\{x_t, y_t, u_t, \theta_t\}_{t\in[T]}\right) &:= \max_{y\in Y} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t \Psi(x_t, y; u_t) - \min_{x\in X} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t \Psi(x, y_t; u_t) \\
\epsilon^\circ\left(\{x_t, u_t, \theta_t\}_{t\in[T]}; u\right) &:= \max_{y\in Y} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t \left[\Psi(x_t, y; u) - \Psi(x_t, y; u_t)\right] \\
\epsilon^\bullet\left(\{y_t, u_t, \theta_t\}_{t\in[T]}; u\right) &:= \max_{x\in X} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t \left[\Psi(x, y_t; u_t) - \Psi(x, y_t; u)\right].
\end{aligned}$$

Before giving the proof of Theorem 2.4, we comment on the terms involved. The sequence $\{\theta_t\}_{t\geq 1}$ defines aggregation weights for the primal-dual sequence $\{x_t, y_t\}_{t\geq 1}$. Recall that our goal is to choose $\bar{x}_T^\theta, \bar{y}_T^\theta$ to make the SP gap term $\epsilon_{\text{sad}}^\Psi\left(\bar{x}_T^\theta, \bar{y}_T^\theta; u\right)$ small, which translates to guarantees on the related convex optimization problem with (unknown) parameters u. Theorem 2.4 gives us a strategy to do this through making the three terms $\widehat{\epsilon}, \epsilon^\circ, \epsilon^\bullet$ small. The $\widehat{\epsilon}$ term can be interpreted as an approximate SP gap term. Notice that since we have knowledge of X, Y and we generate the sequences $\{x_t, y_t\}_{t\geq 1}$, we should be able to control this term. On the other hand, in both the $\epsilon^\circ, \epsilon^\bullet$ terms, u is present, and since this is unknown to us, we cannot expect to have full control of these terms. Indeed, we will see that $\{x_t, y_t\}_{t\geq 1}$ will be chosen to make $\widehat{\epsilon}$ small, reagrdless of the setting. In contrast, making $\epsilon^\circ, \epsilon^\bullet$ small, is not guaranteed through the choice of $\{u_t\}_{t\geq 1}$ alone, but will also require us to take into account how the uncertainty in u is modelled.

Proof of Theorem 2.4. First, notice that $\{\theta_t/\Theta_T\}_{t\in[T]}$ form a set of convex combination weights. Thus, using the convex-concave structure of $\Psi(\cdot, \cdot; u)$ and the definition of $\epsilon_{\text{sad}}^\Psi\left(\bar{x}_T^\theta, \bar{y}_T^\theta; u\right)$ in (2.9), we get the standard bound

$$\epsilon_{\text{sad}}^\Psi\left(\bar{x}_T^\theta, \bar{y}_T^\theta; u\right) \leq \max_{y\in Y} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t \Psi(x_t, y; u) - \min_{x\in X} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t \Psi(x, y_t; u).$$

Let us examine the first term on the right hand side. Adding and subtracting $\Psi(x_t, y; u_t)$ for each term in the sum, we can bound this term by

$$\begin{aligned}
\max_{y\in Y} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \Psi(x_t, y; u) &= \max_{y\in Y} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \left[\Psi(x_t, y; u_t) + \Psi(x_t, y; u) - \Psi(x_t, y; u_t)\right] \\
&= \max_{y\in Y} \left\{ \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \Psi(x_t, y; u_t) + \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \left[\Psi(x_t, y; u) - \Psi(x_t, y; u_t)\right] \right\} \\
&\leq \max_{y\in Y} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \Psi(x_t, y; u_t) + \max_{y\in Y} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \left[\Psi(x_t, y; u) - \Psi(x_t, y; u_t)\right] \\
&= \max_{y\in Y} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \Psi(x_t, y; u_t) + \epsilon^\circ \left(\{x_t, u_t, \theta_t\}_{t\in[T]}; u\right).
\end{aligned}$$

Using a similar strategy for the second term $\min_{x\in X} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \Psi(x, y_t; u)$, we can get

$$\min_{x\in X} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \Psi(x, y_t; u) \geq \min_{x\in X} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \Psi(x, y_t; u_t) - \epsilon^\bullet \left(\{y_t, u_t, \theta_t\}_{t\in[T]}; u\right).$$

Subtracting this lower bound from the upper bound on the first term then gives us the result. □

To conclude this section, we briefly mention how to choose the primal-dual sequence $\{x_t, y_t\}_{t\geq 1}$ and weights $\{\theta_t\}_{t\geq 1}$ to bound the first term $\widehat{\epsilon}\left(\{x_t, y_t, u_t, \theta_t\}_{t\in[T]}\right)$. Note that we will not give explicit algorithms; we reserve this for Chapter 3, where we build the necessary tools to do so. Instead, in this chapter we wish to highlight just the general primal-dual framework, which allows us to plug the various techniques to be described in Chapter 3 in, resulting in various different algorithms for our problems.

Notice that $\widehat{\epsilon}\left(\{x_t, y_t, u_t, \theta_t\}_{t\in[T]}\right)$ can be written as the sum of two terms:

$$\begin{aligned}
\widehat{\epsilon}\left(\{x_t, y_t, u_t, \theta_t\}_{t\in[T]}\right) = \max_{y\in Y} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \Psi(x_t, y; u_t) - \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \Psi(x_t, y_t; u_t) \\
+ \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \Psi(x_t, y_t; u_t) - \min_{x\in X} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \Psi(x, y_t; u_t).
\end{aligned}$$

It turns out that these two terms are so-called *regret terms*. The area of *online convex optimization* (OCO), which we describe in detail in Chapter 3, provides for us various techniques that allow us to choose the sequences $\{x_t, y_t\}_{t\geq 1}$ to bound these terms, so-called *regret-minimizing* algorithms. As mentioned above, we defer a full account of these to Chapter 3, but we mention that a typical result is that the regret terms are bounded by $\leq O(1/\sqrt{T})$ after T iterations. Thus, after T steps, we can make $\widehat{\epsilon}\left(\{x_t, y_t, u_t, \theta_t\}_{t\in[T]}\right) \leq O(1/\sqrt{T})$, and making T large enough means that we can make these arbitrarily small. When combined with similar bounds on $\epsilon^\circ, \epsilon^\bullet$, this gives us a bound on the SP gap $\epsilon^\Psi_{\text{sad}}$, and hence gives guarantees on the convex optimization problem of interest. We now describe how such bounds may arise for RO and JEO.

2.4 Application to Robust Optimization

We first consider the RO problem (1.2) introduced in Chapter 1. As discussed in Chapter 1.3, it is sufficient to solve the (approximate) robust feasibility problem (1.4), since we can then conduct a binary search to find the best objective value. We restate this here for convenience:

$$\begin{cases} \textit{Either}\text{: find } \ x \in X \quad \text{s.t.} \quad \sup_{u^i \in U^i} f^i(x, u^i) \le \epsilon \quad \forall i \in [m]; \\ \textit{or}\text{: declare infeasibility, } \forall x \in X, \ \exists i \in [m] \quad \text{s.t.} \quad \sup_{u^i \in U^i} f^i(x, u^i) > 0. \end{cases} \tag{2.10}$$

We assume that the following holds.

Assumption 2.5. The constraint functions $f^i(x, u^i)$ for all $i \in [m]$ are finite-valued on the domain $X \times U^i$, convex in x and concave in u^i. Furthermore, X, the domain for x, is closed and convex, and U^i, the domains for u^i, are closed and bounded.

In fact, (2.10) is a version of (2.2) which takes into account the uncertainty in u: while we do not know $u = (u^1, \ldots, u^m)$ exactly, in many situations, it is possible to build a confidence set $U = U^1 \times \ldots \times U^m$ where u could belong. By solving (2.10), we ensure that whenever u falls within the confidence set U, the constraints will still be satisfied.

We first formulate (2.10) as a SP problem. Define

$$Y := \Delta_m, \quad \Psi(x, y; u) := \sum_{i \in [m]} y^i f^i(x, u^i), \quad \Phi(x, y) := \sup_{u \in U} \Psi(x, y; u) = \sum_{i \in [m]} y^i \sup_{u^i \in U^i} f^i(x, u^i). \tag{2.11}$$

That is, in RO, we do not have a target u, but we are finding the worst-case bound on $\Psi(x, y; u)$ over $u \in U$. By following Example 2.1, it is straightforward to see that (2.10) is equivalent to

$$\begin{cases} \textit{Either}\text{: } \min_{x \in X} \max_{y \in Y} \Phi(x, y) \le \epsilon, \\ \textit{or}\text{: declare infeasibility, } \min_{x \in X} \max_{y \in Y} \Phi(x, y) > 0. \end{cases} \tag{2.12}$$

Let us define the SP value

$$\text{SV} = \min_{x \in X} \max_{y \in Y} \Phi(x, y). \tag{2.13}$$

If we are able to accurately estimate this, then we can solve (2.12).

Proposition 2.6. *Let $\Psi : X \times Y \to \mathbb{R}$ be a given function associated with a SP (not necessarily admitting a convex-concave structure). Suppose we have $\bar{x} \in X$, $\bar{y} \in Y$, and $\tau \in (0, 1)$ such that $\epsilon^{\Psi}_{\text{sad}}(\bar{x}, \bar{y}) := \max_{y \in Y} \Psi(\bar{x}, y) - \min_{x \in X} \Psi(x, \bar{y}) \le \tau\epsilon$. Then if $\Psi(\bar{x}, \bar{y}) \le (1 - \tau)\epsilon$, we have $\sup_{y \in Y} \Psi(\bar{x}, y) \le \epsilon$. Moreover, if $\Psi(\bar{x}, \bar{y}) > (1 - \tau)\epsilon$ and $\tau \le 1/2$, we have $\min_{x \in X} \Psi(x, \bar{y}) > 0$.*

Proof. Suppose $\Psi(\bar{x}, \bar{y}) \le (1-\tau)\epsilon$. Because $\epsilon^{\Psi}_{\text{sad}}(\bar{x}, \bar{v}) \le \tau\epsilon$, we have $\max_{y \in Y} \Psi(\bar{x}, y) \le \min_{x \in X} \Psi(x, \bar{y}) + \tau\epsilon \le \Psi(\bar{x}, \bar{y}) + \tau\epsilon \le \epsilon$. On the other hand, when $\Psi(\bar{x}, \bar{y}) > (1 - \tau)\epsilon$, we have $(1 - \tau)\epsilon < \Psi(\bar{x}, \bar{y}) \le \max_{y \in Y} \Psi(\bar{x}, y) \le \min_{x \in X} \Psi(x, \bar{y}) + \tau\epsilon$, which implies $\min_{x \in X} \max_{y \in Y} \Phi(x, y) \ge \min_{x \in X} \Psi(x, \bar{y}) > (1 - 2\tau)\epsilon \ge 0$ when $\tau \le 1/2$. □

Estimating SV in (2.13), however, is not a straightforward task. Notice that since $\Phi(x, y)$ is defined with a supremum over u, convexity in x is preserved (through convexity f^i in Assumption

2.5), hence $\Phi(x, y)$ is a convex-concave function. However, applying the machinery of convex-concave SP problems to estimate SV requires us to handle the suprema over U^i somehow, which can be difficult. On the other hand, if we fix some $u \in U$, the closely related problem $\mathrm{SV}(u) := \min_{x\in X} \max_{y\in Y} \Psi(x, y; u)$ is also a convex-concave SP problem. The primal-dual framework in Section 2.3 gives us a way to handle the suprema over U^i in an iterative manner. The following result gives a relationship between the family of convex-concave SP problems $\mathrm{SV}(u)$ and our problem of interest (2.13).

Theorem 2.7. *Fix some $\bar{x} \in X$, $\bar{y} \in Y$. Then*

$$\max_{y\in Y} \Psi(\bar{x}, y) - \min_{x\in X} \Psi(x, \bar{y}) \leq \sup_{u\in U} \left\{ \max_{y\in Y} \Psi(\bar{x}, y; u) - \min_{x\in X} \Psi(x, \bar{y}; u) \right\}.$$

Consequently,

$$\forall u \in U,\ \epsilon^{\Psi}_{\mathrm{sad}}(\bar{x}, \bar{y}; u) \leq \epsilon \implies \max_{y\in Y} \Psi(\bar{x}, y) - \min_{x\in X} \Psi(x, \bar{y}) \leq \epsilon.$$

Proof. First note that $\max_{y\in Y} \Psi(\bar{x}, y) = \max_{y\in Y} \sup_{u\in U} \Psi(\bar{x}, y; u) = \sup_{u\in U} \max_{y\in Y} \Psi(\bar{x}, y; u)$. Now fix an arbitrary $u \in U$. Since $\min_{x\in X} \Psi(x, \bar{y}) \geq \min_{x\in X} \Psi(x, \bar{y}; u)$ we have

$$\max_{y\in Y} \Psi(\bar{x}, y; u) - \min_{x\in X} \Psi(x, \bar{y}) \leq \max_{y\in Y} \Psi(\bar{x}, y; u) - \min_{x\in X} \Psi(x, \bar{y}; u).$$

Taking the supremum over $u \in U$ both sides and using the equality

$$\max_{y\in Y} \Psi(\bar{x}, y) = \sup_{u\in U} \max_{y\in Y} \Psi(\bar{x}, y; u)$$

gives the result. □

While seemingly simple, Theorem 2.7 is quite powerful, as it allows us to quantify the Sp gap of (2.13) in terms of SP gaps of convex-concave SP problems $\mathrm{SV}(u)$ from (2.1). In Section 2.3 we presented a technique to bound $\epsilon^{\Psi}_{\mathrm{sad}}(\bar{x}, \bar{y}; u)$ for a *fixed* $u \in U$. Here, we wish to find a uniform bound for $\epsilon^{\Psi}_{\mathrm{sad}}(\bar{x}, \bar{y}; u)$ over *all* $u \in U$, but Theorem 2.4 may still prove useful in this setting. Let us analyse the three upper bound terms in Theorem 2.4 for the specific definition of $\Psi(x, y; u) = \sum_{i\in[m]} y^i f^i(x, u^i)$ and $Y = \Delta_m$, and for a fixed $u \in U$:

$$\widehat{\epsilon}\left(\{x_t, y_t, u_t, \theta_t\}_{t\in[T]}\right) = \max_{i\in[m]} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t f^i(x_t, u^i_t) - \min_{x\in X} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \sum_{i\in[m]} y^i_t f^i(x, u^i_t) \tag{2.14}$$

$$\epsilon^{\circ}\left(\{x_t, u_t, \theta_t\}_{t\in[T]}; u\right) = \max_{i\in[m]} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \left[f^i(x_t, u^i) - f^i(x_t, u^i_t)\right] \tag{2.15}$$

$$\epsilon^{\bullet}\left(\{y_t, u_t, \theta_t\}_{t\in[T]}; u\right) = \max_{x\in X} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \sum_{i\in[m]} y^i_t \left[f^i(x, u^i_t) - f^i(x, u^i)\right]. \tag{2.16}$$

Note that the first term is independent of the choice of $u \in U$, whereas the second and third terms contain u. To bound these *uniformly* over $u \in U$, we must find uniform bounds for the second and

third terms over $u \in U$. Thus, let us rewrite these terms with a supremum over $u \in U$:

$$\sup_{u\in U} \epsilon^\circ \left(\{x_t, u_t, \theta_t\}_{t\in[T]} ; u \right) = \max_{i\in[m]} \left\{ \sup_{u^i \in U^i} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t f^i(x_t, u^i) - \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t f^i(x_t, u_t^i) \right\} \tag{2.17}$$

$$\sup_{u\in U} \epsilon^\bullet \left(\{y_t, u_t, \theta_t\}_{t\in[T]} ; u \right) = \sup_{u\in U} \max_{x\in X} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \sum_{i\in[m]} y_t^i \left[f^i(x, u_t^i) - f^i(x, u^i) \right]. \tag{2.18}$$

(The first equality follows by exploiting the product form $U = U^1 \times \ldots \times U^m$ and the structure of ϵ°.) Notice that ϵ° is simply the maximum of m different *regret* terms, one for each constraint. As mentioned in Section 2.3, there exists efficient regret-minimizing algorithms to bound these, and in Chapter 3 we will present some examples of these. Such algorithms can be used to choose sequences $\{u_t^i\}_{t\geq 1}$ in order to bound each regret term in ϵ°. In fact, bounding these will give us *uniform* bounds of ϵ° over $u \in U$. However, the third $\epsilon^\bullet$ term has no such interpretation. Thus, currently, we only know how to bound two of the three upper bound terms, which is not sufficient to bound the SP gap of (2.13).

Fortunately, it turns out that, for robust feasibility, bounding the first two terms is sufficient. This phenomenon is captured in our main Theorem 2.8 below. Intuitively, this is due to the fact that we do not need to evaluate SV exactly to solve (2.12), we only need to determine whether SV $\leq \epsilon$ or SV > 0, and certifying this only requires bounds on the first two terms.

Theorem 2.8. *Suppose we have sequences $\{x_t, y_t, u_t, \theta_t\}_{t\in[T]}$ with $x_t \in X$, $u_t \in U$, $y_t \in Y$ and $\theta_t > 0$ for all $t \in [T]$. Let $\tau \in (0,1)$. If $\sup_{u\in U} \epsilon^\circ(\{x_t, u_t, \theta_t\}_{t\in[T]}; u) \leq \tau\epsilon$ and $\max_{i\in[m]} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t f^i(x_t, u_t^i) \leq (1-\tau)\epsilon$, then the solution $\bar{x}_T^\theta := \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t x_t$ is ϵ-feasible with respect to (2.10). If $\widehat{\epsilon}(\{x_t, y_t, u_t, \theta_t\}_{t\in[T]}) \leq (1-\tau)\epsilon$ and $\max_{i\in[m]} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t f^i(x_t, u_t^i) > (1-\tau)\epsilon$, then (2.10) is infeasible.*

Proof. First suppose there exists a $\tau \in (0,1)$ and sequences $\{x_t, u_t, \theta_t\}_{t\in[T]}$ such that $\epsilon^\circ(\{x_t, u_t, \theta_t\}_{t\in[T]}) \leq \tau\epsilon$ and $\max_{i\in[m]} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t f^i(x_t, u_t^i) \leq (1-\tau)\epsilon$ holds as well. Note that

$$\begin{aligned} \tau\epsilon \geq \max_{u\in U} \epsilon^\circ(\{x_t, u_t, \theta_t\}_{t\in[T]}; u) &= \max_{i\in[m]} \left\{ \sup_{u^i \in U^i} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t f^i(x_t, u^i) - \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t f^i(x_t, u_t^i) \right\} \\ &\geq \max_{i\in[m]} \sup_{u^i\in U^i} \sum_{t\in[T]} \theta_t f^i(x_t, u^i) - \max_{i\in[m]} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t f^i(x_t, u_t^i), \end{aligned} \tag{2.19}$$

where the last inequality follows since $\max_{i\in[m]}\{\alpha_i - \beta_i\} \geq \max_{i\in[m]} \alpha_i - \max_{i\in[m]} \beta_i$ for any sequence of numbers α_i, β_i, $i \in [m]$. Then $\bar{x}_T^\theta$ is an ϵ-feasible solution for (2.10) because

$$\begin{aligned} \max_{i\in[m]} \sup_{u^i\in U^i} f^i\left(\bar{x}_T^\theta, u^i\right) &= \max_{i\in[m]} \sup_{u^i\in U^i} f^i\left(\frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t x_t, u^i \right) \\ &\leq \max_{i\in[m]} \sup_{u^i\in U^i} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t f^i(x_t, u^i) \leq \tau\epsilon + \max_{i\in[m]} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t f^i(x_t, u_t^i) \leq \epsilon, \end{aligned}$$

where the first inequality follows from the convexity of the functions f^i and the fact that $\theta \in \Delta_T$, the second inequality from (2.19), and the last inequality holds since $\max_{i\in[m]} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t f^i(x_t, u_t^i) \leq (1-\tau)\epsilon$.

On the other hand, suppose $\widehat{\epsilon}(\{x_t, u_t, y_t, \theta_t\}_{t\in[T]}) \le (1-\tau)\epsilon$ and $\max_{i\in[m]} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f^i(x, u_t^i) > (1-\tau)\epsilon$. Note that

$$\begin{aligned}\min_{x\in X} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t \sum_{i\in[m]} y_t^i f^i(x, u_t^i) &\le \min_{x\in X} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t \max_{i\in[m]} f^i(x, u_t^i) \\ &\le \min_{x\in X} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t \max_{i\in[m]} \sup_{u^i\in U^i} f^i(x, u^i) = \min_{x\in X} \max_{i\in[m]} \sup_{u^i\in U^i} f^i(x, u^i) \\ &= \min_{x\in X} \max_{y\in Y} \Phi(x, y), \end{aligned} \tag{2.20}$$

where the first inequality follows since $y_t \in \Delta_m$ for all $t \in [T]$, the second inequality holds because $f^i(x, u_t^i) \le \sup_{u^i\in U^i} f^i(x, u^i)$ for all $i \in [m]$ and $y_t^i \ge 0$ for $i \in [m]$, $t \in [T]$, and the last equation follows since $\{\theta_t/\Theta_T\}_{t\in[T]}$ are convex combination weights. Then using the bound

$$(1-\tau)\epsilon \ge \widehat{\epsilon}(\{x_t, u_t, y_t, \theta_t\}_{t\in[T]}) = \max_{i\in[m]} \sum_{t\in[T]} \theta_t f^i(x_t, u_t^i) - \min_{x\in X} \sum_{t\in[T]} \theta_t \sum_{i=1}^m y_t^i f^i(x, u_t^i), \tag{2.21}$$

we arrive at

$$\min_{x\in X} \max_{y\in Y} \Phi(x, y) \ge \min_{x\in X} \sum_{t\in[T]} \theta_t \sum_{i=1}^m y_t^i f^i(x, u_t^i) \ge \max_{i\in[m]} \sum_{t\in[T]} \theta_t f^i(x_t, u_t^i) - (1-\tau)\epsilon > 0,$$

where the first inequality follows from inequality (2.20), the second inequality from (2.21) and the last inequality holds because $\max_{i\in[m]} \sum_{t\in[T]} \theta_t f^i(x_t, u_t^i) > (1-\tau)\epsilon$. This implies (2.10) is infeasible, due to the equivalence of (2.12) and (2.10). □

Theorem 2.8 points to a general framework for solving the robust feasibility problem (2.10) given access to appropriate regret-minimizing algorithms for bounding $\widehat{\epsilon}, \epsilon^\circ$. We now describe this general framework precisely. First, we define some notation for the regret minimizing algorithms. Recall that we will use regret-minimizing algorithms to choose $\{x_t, y_t\}_{t\ge1}$ in order to bound $\widehat{\epsilon}$, while (possibly different) regret-minimizing algorithms will be used for choosing $\{u_t^i\}_{t\ge1}$ for each regret term $i \in [m]$ in ϵ°. Thus, we denote these algorithms by $\mathcal{A}_{xy}$ and $\mathcal{A}_i$, $i \in [m]$. A key feature of the regret-minimizing algorithms that we will describe in Chapter 3 is that choosing the t-th point only depends on the data up to step $t-1$ or step t, but not further. Thus, we denote the iterates generated by the algorithms at step t by

$$x_t, y_t = \mathcal{A}_{xy}\left(\{x_s, y_s, u_s, \theta_s\}_{s\in[t-1]}\right) \text{ or } \mathcal{A}_{xy}\left(\{x_s, y_s, u_s, \theta_s\}_{s\in[t-1]}, u_t, \theta_t\right)$$

and

$$u_t^i = \mathcal{A}_i\left(\{x_s, u_s^i, \theta_s\}_{s\in[t-1]}\right) \text{ or } \mathcal{A}_i\left(\{x_s, u_s^i, \theta_s\}_{s\in[t-1]}, x_t, \theta_t\right), \quad \forall i \in [m].$$

Traditionally, regret-minimizing algorithms from OCO do not use information from step t, but in our setting this may be allowed. The choice of whether we use information at step t or not presents further flexibility to our framework, and opportunities for obtaining faster convergence rates, but must be done with care. We elaborate on this in Remark 2.9. We denote the corresponding bounds

on the regret terms obtainable through the algorithms after T iterations as $\mathcal{R}_{xy}(T), \mathcal{R}_i(T)$. In other words, we have

$$\widehat{\epsilon}\left(\{x_t, y_t, u_t, \theta_t\}_{t\in T}\right) \le \mathcal{R}_{xy}(T)$$
$$\sup_{u^i\in U^i} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f^i(x_t, u^i) - \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f^i(x_t, u^i_t) \le \mathcal{R}_i(T)$$
$$\sup_{u\in U} \epsilon^\circ\left(\{x_t, u_t, \theta_t\}_{t\in[T]}; u\right) \le \max_{i\in[m]} \mathcal{R}_i(T).$$

Note that in general we have $\mathcal{R}_{xy}, \mathcal{R}_i \ge 0$. We are now ready to give the full description of our framework in Algorithm 1.

Algorithm 1 Robust feasibility framework.

input: regret-minimizing algorithms $\mathcal{A}_i$, $i \in [m]$, $\mathcal{A}_{xy}$, tolerance level $\epsilon > 0$, sufficiently large $T = T(\epsilon)$ such that $\max_{i\in[m]} \mathcal{R}_i(T) + \mathcal{R}_{xy}(T) \le \epsilon$, and positive weights $\theta_1, \dots, \theta_T > 0$.
output: either $\bar{x} \in X$ such that $\sup_{u^i\in U^i} f^i(\bar{x}, u^i) \le \epsilon$ for all $i \in [m]$, or an infeasibility certificate for (2.10).
initialize

$$u^i_1 = \mathcal{A}_i(\{\}) \ \forall i \in [m], \quad x_1, y_1 = \mathcal{A}_{xy}(\{\}, u_1, \theta_1)$$
$$\text{OR} \quad x_1, y_1 = \mathcal{A}_{xy}(\{\}), \quad u^i_1 = \mathcal{A}_i(\{\}, x_1, \theta_1) \ \forall i \in [m].$$

for $t = 2, \dots, T$ **do**
compute updates

$$u^i_t = \mathcal{A}_i(\{x_s, u^i_s, \theta_s\}_{s\in[t-1]}) \ \forall i \in [m], \quad x_t, y_t = \mathcal{A}_{xy}(\{x_s, y_s, u_s, \theta_s\}_{s\in[t-1]}, u_t, \theta_t)$$
$$\text{OR} \quad x_t, y_t = \mathcal{A}_{xy}(\{x_s, y_s, u_s, \theta_s\}_{s\in[t-1]}), \quad u^i_t = \mathcal{A}_i(\{x_s, u^i_s, \theta_s\}_{s\in[t-1]}, x_t, \theta_t) \ \forall i \in [m].$$

and obtain upper bounds

$$\sup_{u\in U} \epsilon^\circ(\{x_s, u_s, \theta_s\}^t_{s=1}; u) \le \max_{i\in[m]} \mathcal{R}_i(t) =: r^\circ$$
$$\widehat{\epsilon}(\{x_s, y_s, u_s, \theta_s\}^t_{s=1}) \le \mathcal{R}_{xy}(t) =: \hat{r}.$$

if $r^\circ + \hat{r} \le \epsilon$ **then**
{we can check for feasibility}
set $\vartheta_t := \max_{i\in[m]} \frac{1}{\Theta_t}\sum_{s\in[t]} \theta_s f^i(x_s, u^i_s)$ and $\tau_t := 1 - \hat{r}/\epsilon$.
if $\vartheta_t > (1-\tau_t)\epsilon = \hat{r}$ **then return** declare (2.10) infeasible
if $\vartheta_t \le (1-\tau_t)\epsilon = \hat{r}$ **then return** $\bar{x}^\theta_t = \frac{1}{\Theta_t}\sum_{s\in[t]} \theta_s x_s$ as a robust ϵ-feasible solution to (2.10).
else
{we only check for infeasibility}
set $\vartheta_t := \max_{i\in[m]} \frac{1}{\Theta_t}\sum_{s\in[t]} \theta_s f^i(x_s, u^i_s)$
if $\vartheta_t > \hat{r}$ **then return** declare (2.10) infeasible
end if
end for

We make one important remark on Algorithm 1.

Remark 2.9. Notice that in Algorithm 1 we list two possible ways to update x_t, y_t and u_t at each step t: one way computes u_t using information only up to step $t-1$, then computes x_t, y_t with knowledge of u_t, and the other way does the opposite. A third way is to compute both u_t and x_t, y_t only with information up to step $t-1$, but the other two ways can be more efficient if we choose the right regret-minimizing algorithms, and result in faster convergence. It is important to note, however, that it is impossible to choose *both* u_t and x_t, y_t with information at step t, thus appropriate care should be taken when choosing the regret-minimizing algorithms $\mathcal{A}_{xy}$ and $\mathcal{A}_i$ to avoid this situation.

2.4.1 Customizations of the Robust Feasibility Framework

We have given interpretations of the quantities $\widehat{\epsilon}, \epsilon^\circ$ as regret terms, discussed how we can use regret-minimizing algorithms to control these, and thus solve the robust feasibility problem (2.10). We will see in Chapter 3 that there exist a variety of regret-minimizing algorithms which only require first-order updates to compute the vectors x_t, y_t, u_t, thus a natural approach is to exclusively use these. We call this approach the *online first-order (OFO)* approach.

It turns out that previous iterative RO approaches for solving (2.10) can be interpreted within our framework. These approaches also generate sequences of solutions $\{x_t\}_{t\geq 1} \subset X$ and data $\{u_t\}_{t\geq 1} \subset U$. The main difference is the methods used to generate these sequences. Previously, Mutapcic and Boyd [107], Ben-Tal et al. [25] have used deterministic optimization oracles for these methods, which are more expensive than first-order updates. In Section 2.4.1.1, we examine the pessimization oracle-based approach of Mutapcic and Boyd [107]. We modify the pessimization oracle-based approach of Mutapcic and Boyd [107] to solving (2.10) within our framework, and demonstrate how we can obtain an improved convergence rate in this way. In Section 2.4.1.2, we examine the nominal feasibility oracle-based approach of Ben-Tal et al. [25] within the context of our general framework. Finally, in Section 2.4.1.3, we summarize and compare the convergence rates achievable via various customizations of our framework using these different approaches.

2.4.1.1 The Pessimization Oracle-Based Approach

Mutapcic and Boyd [107] generate solutions $x_t \in X$ at each iteration t by solving an extended nominal problem

$$\min_{x\in X} \left\{ f^0(x) : \ f^i(x, u^i) \leq 0, \ \forall u^i \in \hat{U}^i_{t-1}, \ i \in [m] \right\}, \tag{2.22}$$

where $\hat{U}^i_{t-1} \subset U^i$ are finite approximate uncertainty sets based on past noise realizations $\{u^i_s\}_{i=1}^m$ for $s \in [t-1]$. New noises u_t are then generated by calling the *pessimization oracles* on the current solution x_t. More precisely, given $x_t \in X$, the pessimization oracles solve $\sup_{u^i \in U^i} f^i(x_t, u^i)$ and return

$$u^i_t \in U^i \quad \text{s.t.} \quad f^i(x_t, u^i_t) \geq \sup_{u^i \in U^i} f^i(x_t, u^i) - \tau\epsilon. \tag{2.23}$$

In terms of our framework of Algorithm 1, the update policy of generating new noises u_t in this approach of [107] corresponds to selecting the algorithms $\mathcal{A}_{xy}$ to be a extended nominal solver for (2.22) (but not generating the y_t variables) and the algorithms $\mathcal{A}_i$ to be pessimization oracles that solve (2.23). Note that computing u^i_t requires knowledge of x_t (see Remark 2.9), and consequently the bound for the regret term in ϵ° is $\mathcal{R}_i(T) \leq \tau\epsilon$ for any T. We show this in the proof of Theorem 2.10. If for all $i \in [m]$ we have $f^i(x_t, u^i_t) \leq (1-\tau)\epsilon$, then we terminate and declare x_t is a robust ϵ-feasible and optimal solution; otherwise, we append $\hat{U}^i_t = \hat{U}^i_{t-1} \cup \{u^i_t\}$ and re-solve (2.22) with the new approximate sets $\hat{U}^i_t$. It is shown in Mutapcic and Boyd [107, Section 5.2] that the number of

iterations T needed before termination with a robust ϵ-feasible solution x_T is upper bounded by $(1+O(1/\epsilon))^n$ where n is the dimension of x.

Suppose now that we are interested in robust feasibility (2.10). Mutapcic and Boyd [107, Section 5.3] discusses a number of variations for generating x_t by modifying (2.22). In contrast, we propose the following modification: instead of solving (2.22), generate $\{x_t, y_t\}_{t\in[T]}$ via an algorithm $\mathcal{A}_{xy}$ to bound $\widehat{\epsilon}(\{x_t, y_t, u_t, \theta_t\}_{t\in[T]}) \leq \mathcal{R}_{xy}(T) \leq (1-\tau)\epsilon$. We call our modification *FO-based pessimization*, since there exist efficient first-order algorithms $\mathcal{A}_{xy}$ which can do this. This fits within our framework as a special case, and is a straightforward consequence of Theorem 2.8.

Theorem 2.10. *Let $\tau \in (0,1)$. Suppose $\{x_t, y_t\}_{t\in[T]}$ are generated iteratively to guarantee that $\widehat{\epsilon}(\{x_t, y_t, u_t, \theta_t\}_{t\in[T]}) \leq (1-\tau)\epsilon$. Suppose each u_t^i are generated by pessimization oracles (2.23) for $i \in [m]$, $t \in [T]$. If there exists $t \in [T]$ such that for all $i \in [m]$ we have $f^i(x_t, u_t^i) \leq (1-\tau)\epsilon$, then x_t is a robust ϵ-feasible solution to (2.10). If $\max_{i\in[m]} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f^i(x_t, u_t^i) \leq (1-\tau)\epsilon$, then $\bar{x}_T^\theta = \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t x_t$ is a robust ϵ-feasible solution to (2.10). If $\max_{i\in[m]} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f^i(x_t, u_t^i) > (1-\tau)\epsilon$, then we certify that (2.10) is robust infeasible.*

Theorem 2.10 can be used to certify robust feasibility/infeasibility. Hence, to find a robust ϵ-optimal solution to (1.2) with the FO-based pessimization approach, we must perform a binary search and solve at most $O(\log(1/\epsilon))$ instances of robust feasibility problems. Despite this, we shall see in Chapter 3 that to make $\mathcal{R}_{xy}(T) \leq \epsilon$, we need at most $T = O(1/\epsilon^2)$ iterations, hence using first-order updates results in much better complexity guarantees than using an extended nominal feasibility solver (2.22) as proposed by Mutapcic and Boyd [107], even when taking into account the additional $O(\log(1/\epsilon))$ factor.

Remark 2.11. In the pessimization oracle-based approach, the noises u_t need to be generated with knowledge of x_t, because it is not possible to guarantee $f^i(x_t, u_t^i) \geq \sup_{u^i\in U^i} f^i(x_t, u^i) - \tau\epsilon$ if the vectors u_t^i were chosen with only the knowledge of $x_1, \ldots, x_{t-1}$. ∎

2.4.1.2 The Nominal Feasibility Oracle-Based Approach

The nominal feasibility oracle-based approach of [25] suggest using OFO algorithms to choose a sequence $\{u_t\}_{t\in[T]}$ that guarantees $\sup_{u\in U} \epsilon^\circ(\{x_t, u_t, \theta_t\}_{t\in[T]}; u)$ is small, in a non-anticipatory fashion, for *any* sequence $\{x_t\}_{t\in[T]}$. For $\mathcal{A}_{xy}$, at step t, [25] utilizes a *nominal feasibility oracle*. That is, given parameters u_t, they call a powerful, and potentially expensive, nominal feasibility oracle that solves the following feasibility problem to ϵ-accuracy

$$\begin{cases} \textit{Either}\text{: find} \quad x \in X \quad \text{s.t.} \quad f^i(x, u_t^i) \leq (1-\tau)\epsilon \quad \forall i \in [m]; \\ \textit{or}\text{: declare infeasibility, } \forall x \in X,\ \exists i \in [m] \quad \text{s.t.} \quad f^i(x, u_t^i) > 0. \end{cases} \tag{2.24}$$

We denote $x_t \in X$ to be the point returned by this oracle at step t, if it exists, and note that we do not require the generation of a dual point $y_t \in Y$. For this approach, the outputs of a nominal feasibility oracle can be used to deduce a result similar to Theorem 2.8, except that we no longer need to evaluate $\widehat{\epsilon}(\{x_t, y_t, u_t, \theta_t\}_{t\in[T]})$, we just need to bound $\epsilon^\circ(\{x_t, u_t, \theta_t\}_{t\in[T]})$.

Theorem 2.12. *Suppose that the sequence $\{u_t\}_{t\in[T]}$ is generated in a non-anticipatory manner to guarantee $\sup_{u\in U} \epsilon^\circ(\{x_t, u_t, \theta_t\}_{t\in[T]}) \leq \tau\epsilon$ for any sequence $\{x_t\}_{t\in[T]}$. Also, suppose that at each step $t \in [T]$, x_t is generated by the nominal feasibility oracle which solves (2.24). If there exists $t \in [T]$ such that (2.24) declares infeasibility, then (2.10) is infeasible. Otherwise, $\bar{x}_T^\theta = \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t x_t$ solves (2.10).*

Proof of Theorem 2.12. If (2.24) declares infeasibility at time t, then it is obvious that (2.10) is infeasible since it is infeasible for u_t. We focus on the latter case. By the premise of the theorem, we have $\sup_{u\in U} \epsilon^\circ(\{x_t, u_t, \theta_t\}_{t\in[T]}; u) \le \tau\epsilon$. Let us evaluate $\max_{i\in[m]} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f^i(x_t, u_t^i)$. From the definition of the nominal feasibility oracle we have $f^i(x_t, u_t^i) \le (1-\tau)\epsilon$ for all $t \in [T]$ and $i \in [m]$, we conclude $\max_{i\in[m]} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f^i(x_t, u_t^i) \le (1-\tau)\epsilon$. The conclusion now follows from Theorem 2.8. □

Thus, the approach of [25], which works with nominal feasibility oracles, fits within our framework of Algorithm 1 right away. We next make three important remarks.

Remark 2.13. Similar to Remark 2.11, a critical property required in the approach of [25] of the vectors x_t is that $f^i(x_t, u_t^i) \le (1-\tau)\epsilon$. This is possible only if each x_t is chosen with the knowledge of u_t. ∎

Remark 2.14. Theorem 2.12 states that the nominal feasibility oracle-based approach can solve robust feasibility problems (2.10). This then recovers Ben-Tal et al. [25, Theorems 1,2]. In addition, we next make a nice and practical observation that was overlooked in [25]. We show that slightly adjusting this oracle will let us directly solve the robust *optimization* problem (1.2), i.e., optimize a convex objective function $f^0(x)$ instead of relying on a binary search over the optimal objective value. Recall that Opt is the optimal value of the RO problem (see (1.2)). Naively, to solve for Opt, we would embed f^0 into the constraint set, and then perform a binary search over the robust feasible set by repeatedly applying the oracle-based approach and Theorem 2.12 to check for robust feasibility. Suppose that now, instead of using a nominal feasibility oracle to solve (2.24), we work with a *nominal optimization oracle*. That is, given data $u_t \in U$, we have access to an oracle that solves

$$\mathrm{Opt}_t = \inf_x \left\{ f^0(x) : f^i(x, u_t^i) \le 0,\ i \in [m], \quad x \in X \right\}.$$

When solving for Opt_t, most convex optimization solvers will either declare that the constraints are infeasible, or return a point $x_t \in X$ such that $f^i(x_t, u_t^i) \le (1-\tau)\epsilon$ and $f^0(x_t) \le \mathrm{Opt}_t + \epsilon$. It is clear that $f^0(x_t) \le \mathrm{Opt}_t + \epsilon \le \mathrm{Opt} + \epsilon$. Given such a sequence of points $\{x_t\}_{t\in[T]}$, from Theorem 2.12 we deduced that $\bar{x}_T^\theta = \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t x_t$ is a robust ϵ-feasible solution. Moreover, convexity of f^0 implies

$$f^0(\bar{x}_T) \le \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f^0(x_t) \le \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t(\mathrm{Opt} + \epsilon) = \mathrm{Opt} + \epsilon.$$

Hence, not only do we claim that $\bar{x}_T^\theta$ is robust ϵ-feasible, but that it is also ϵ-optimal. Thus, when our oracle can return ϵ-optimal solutions, which most solvers can, we eliminate the need to perform a binary search. ∎

Below we elaborate on the differences between Theorem 2.12 and Theorem 2.8.

Remark 2.15. In contrast to Theorem 2.8, Theorem 2.12 does not need to control the term $\widehat{\epsilon}(\{x_t, y_t, u_t, \theta_t\}_{t\in[T]})$. The reason is that due to (2.24), each point x_t satisfies $f^i(x_t, u_t^i) \le (1-\tau)\epsilon$, hence $\max_{i\in[m]} \sum_{t\in[T]} \theta_t f^i(x_t, u_t^i) \le (1-\tau)\epsilon$ always holds. Therefore, the infeasibility part of Theorem 2.8 never becomes relevant. However, if we choose to solve (2.24) in a particular way, we can get a bound on $\widehat{\epsilon}(\{x_t, y_t, u_t, \theta_t\}_{t\in[T]})$.

Note that (2.24) is equivalent to checking $\mathrm{SV}(u_t) \le (1-\tau)\epsilon$ or $\mathrm{SV}(u_t) > 0$, where

$$\mathrm{SV}(u_t) := \min_{x\in X}\max_{y\in Y} \Psi(x, y; u_t) = \min_{x\in X}\left\{ \max_{i\in[m]} f^i(x, u_t^i) \right\}, \tag{2.25}$$

and Ψ, Y are defined as before in (2.11). Since each $f^i(x, u_t^i)$ is convex in x for fixed u_t^i, $\max_{i\in[m]} f^i(x, u_t^i)$ is convex in x also, hence standard convex optimization methods may be employed to find $x_t \in X$ such that

$$\mathrm{SV}(u_t) \leq \max_{i\in[m]} f^i(x_t, u_t^i) \leq \mathrm{SV}(u_t) + (1-\tau)\epsilon.$$

Then, by checking whether $\max_{i\in[m]} f^i(x_t, u_t^i) \leq (1-\tau)\epsilon$ or $\max_{i\in[m]} f^i(x_t, u_t^i) > (1-\tau)\epsilon$, we can determine whether $\mathrm{SV}(u_t) \leq (1-\tau)\epsilon$ or $\mathrm{SV}(u_t) > 0$ respectively. In particular, if we find that $\mathrm{SV}(u_t) \leq (1-\tau)\epsilon$, our point x_t is feasible for (2.24).

Also, when all the vectors x_t satisfy (2.25), and we choose $y_t^i = 1$ if $f^i(x_t, u_t^i) = \max_{i'\in[m]} f^{i'}(x_t, u_t^{i'})$ and $y_t^i = 0$ otherwise, we have the bound

$$\begin{aligned}
\widehat{\epsilon}(\{x_t, y_t, u_t, \theta_t\}_{t\in[T]}) &= \max_{i\in[m]} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t f^i(x_t, u_t^i) - \min_{x\in X} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \sum_{i\in[m]} y_t^i f^i(x, u_t^i) \\
&\leq \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \max_{i\in[m]} f^i(x_t, u_t^i) - \min_{x\in X} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \max_{i\in[m]} f^i(x, u_t^i) \\
&\leq \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \left[\max_{i\in[m]} f^i(x_t, u_t^i) - \min_{x\in X} \max_{i\in[m]} f^i(x, u_t^i)\right] \leq (1-\tau)\epsilon.
\end{aligned}$$

Consequently, we deduce that the nominal feasibility oracle, implemented as a convex optimization problem, also naturally bounds $\widehat{\epsilon}(\{x_t, y_t, u_t, \theta_t\}_{t\in[T]})$ although this bound is not utilized in Theorem 2.12. In terms of our framework of Algorithm 1, the update policy of generating new solutions x_t, y_t in this approach corresponds to selecting the algorithm $\mathcal{A}_{xy}$ to be a convex optimization solver that solves (2.25), which guarantees that $\widehat{\epsilon}(\{x_s, y_s, u_s, \theta_s\}_{s\in[t]}) \leq \mathcal{R}_{xy}(t) = (1-\tau)\epsilon$ for any t. However, at each iteration t, instead of evaluating $\mathrm{SV}(u_t)$ to $(1-\tau)\epsilon$ accuracy, there exist algorithms which can perform only simple first-order updates for x_t, y_t but still maintain similar guarantees on $\widehat{\epsilon}$, albeit at the end of the time horizon T, not at each iteration t. ∎

2.4.1.3 Convergence Rates and Discussion

We summarize the convergence rates achievable in our general RO framework for various cases. We first examine the number of iterations required for each approach discussed, then proceed to analyze the per-iteration cost of each approach. A summary of our discussion is given in Table 2.1. We use the notation $r_u(\epsilon)$ to denote the number of iterations T required for algorithms $\mathcal{A}_i$ to guarantee $\sup_{u\in U} \epsilon^\circ(\{x_t, u_t, \theta_t\}_{t\in[T]}; u) \leq \max_{i\in[m]} \mathcal{R}_i(T) \leq \epsilon/2$. Similarly, we let $r_{xy}(\epsilon)$ be the number of iterations T required for an algorithm $\mathcal{A}_{xy}$ to guarantee that $\widehat{\epsilon}(\{x_t, y_t, u_t, \theta_t\}_{t\in[T]}) \leq \mathcal{R}_{xy}(T) \leq \epsilon/2$. Then the resulting worst-case number of iterations needed in Algorithm 1 to obtain robust ϵ-feasibility/infeasibility certificates is $\max\{r_u(\epsilon), r_{xy}(\epsilon)\}$.

In Chapter 3, we shall see that under minimal assumptions, we have $r_u(\epsilon) = O(1/\epsilon^2)$ and $r_x(\epsilon) = O(1/\epsilon^2)$. Thus, our OFO-based approach requires $O(1/\epsilon^2)$ iterations to solve (2.10). Since our OFO-based approach returns only robust ϵ-feasible solutions, we need to perform a binary search and repeatedly invoke our method $O(\log(1/\epsilon))$ times to obtain ϵ-optimal solutions to (1.2), so the total number of iterations is $O(\log(1/\epsilon)/\epsilon^2)$.

Our FO-based pessimization approach, i.e., our modification of the pessimization oracle-based approach of Mutapcic and Boyd [107] outlined in Section 2.4.1.1, requires $r_x(\epsilon)$ iterations to solve (2.10) because by Theorem 2.10 we only need to guarantee $\widehat{\epsilon}(\{x_t, y_t u_t, \theta_t\}_{t\in[T]}) \leq \mathcal{R}_{xy}(T) \leq \epsilon/2$. Taking into account the binary search factor $O(\log(1/\epsilon))$ to find a robust ϵ-optimal solution, the

total number of iterations required is $O(\log(1/\epsilon)/\epsilon^2)$, which is much better than the exponential $(1+O(1/\epsilon))^n$ bound of Mutapcic and Boyd [107, Section 5.2] that uses a full nominal solution oracle (2.22). Similarly, the nominal feasibility/optimization oracle-based approach of [25] outlined in Section 2.4.1.2 requires $r_u(\epsilon) = O(1/\epsilon^2)$ iterations (or $r_u(\epsilon)\log(1/\epsilon) = O(\log(1/\epsilon)/\epsilon^2)$ iterations if only a feasibility oracle is used) to obtain robust ϵ-optimal solutions because by Theorem 2.12 we only need to bound $\epsilon^\circ(\{x_t, u_t, \theta_t\}_{t\in[T]}) \leq \max_{i\in[m]} \mathcal{R}_i(T) \leq \epsilon/2$.

Remark 2.16. The algorithms $\mathcal{A}_i$ and $\mathcal{A}_{xy}$ may be chosen to exploit certain structural properties. For example, when f^i are strongly convex, we will see in Chapter 3 that certain OCO algorithms achieve faster convergence rates. Moreover, unless explicitly required by the algorithms $\mathcal{A}_i$, we do not need to assume convexity of the sets U^i. As a result, the follow-the-leader or follow-the-perturbed-leader type algorithms from [95] can be utilized as $\mathcal{A}_i$ in our framework even when U^i are nonconvex but certain assumptions ensuring applicability of these algorithms are satisfied. Such assumptions are satisfied for example when $f^i(x, u^i)$ are linear in u^i and the nonconvex sets U^i admit a certain linear optimization oracle. This is for example the case in a certain lifted representation of the robust convex quadratic constraint discussed in Ben-Tal et al. [25, Section 4.2]. Similarly, when the functions $f^i(x, u^i)$ are exp-concave in u^i, applying the online Newton step algorithm of [77] for $\mathcal{A}_i$ results in a regret bound of at most $O(\log(T)/T)$ in T iterations. Such f^i that are exp-concave in u^i satisfying Assumption 2.5 arise in optimization under uncertainty problems where variance is used as a risk measure, e.g., mean-variance portfolio optimization problems, see for example Ben-Tal et al. [24, Example 25]. Essentially, the same flexibility for acceleration and/or working with nonconvex sets U^i is present in [25] as well.

In the presence of favorable problem structure, based on Table 2.1, if an accelerated algorithm to exploit problem structure is employed in the place of $\mathcal{A}_i$, the overall number of iterations of the nominal feasibility approach is immediately reduced accordingly. Analogous result holds for $\mathcal{A}_{xy}$ and the FO-based pessimization approach. However, in the case of our OFO-based approach, we need to have favorable structure in *both* x and u and utilize the corresponding accelerated algorithms $\mathcal{A}_{xy}, \mathcal{A}_i$ to attain the acceleration of the overall approach. ∎

We now discuss the per-iteration cost for each approach. In order to discuss the total *arithmetic complexity* of each approach, we let k be the maximum dimension of the uncertain parameters u^i for $i \in [m]$ and recall that n denotes the dimension of the decision variables x. In the case where our domains $X, \{U^i\}_{i=1}^m$ have favorable geometry, such as Euclidean ball or simplex, the vectors $x_t, \{u_t^i\}_{i=1}^m$ are updated via simple closed-form prox operations, which cost $O(n)$ and $O(km)$ per iteration respectively. The cost of computing the subgradients $\nabla_x f^i(x, u^i), \nabla_u f^i(x, u^i)$ is $O(km + mn)$ each iteration. This cost is incurred in each iteration of all of the approaches we discuss. From this, we deduce that the per-iteration cost of our OFO-based approach is at most $O(km + mn)$.

The per-iteration cost of the pessimization oracle based approaches involve calling m pessimization oracles (2.23) and the costs related to updating x_t. We denote by $\text{Pess}(\epsilon, k)$ the complexity of a pessimization oracle with tolerance ϵ and k variables. A summary of different possible implementations is given in Table 2.2. If $\sup_{u^i\in U^i} f^i(x, u^i)$ has a simple closed form solution, then the resulting arithmetic cost for $\text{Pess}(\epsilon, k)$ is $O(k)$ for each pessimization oracle. If we can use polynomial-time IPMs, this cost becomes $O(k^3 \log(1/\epsilon))$ [17, Section 6.6], and using FOMs has cost $O(k \log(1/\epsilon))$ in the best case when the functions f^i are smooth *and* strongly convex in u^i. In the case of our FO-based pessimization approach, the update involving x_t will be given by simple closed form formulas for prox operations when X has favorable geometry, resulting in a cost of $O(mn)$. The full pessimization approach of [107] incurs the cost of solving an extended nominal

Approach	Binary search	No. iterations	Per-iteration cost
OFO-based	$\log(1/\epsilon)$	$\max\{r_u(\epsilon), r_x(\epsilon)\}$	$O(km+mn)$
FO-based pessimization	$\log(1/\epsilon)$	$r_x(\epsilon)$	$m\,\mathrm{Pess}(\epsilon,k)+O(mn)$
Nominal oracle	see Table 2.3	$r_u(\epsilon)$	$O(km)+\mathrm{Nom}(\epsilon,m,n)$
Full pessimization	see Table 2.3	$O(1/\epsilon^n)$	$m\,\mathrm{Pess}(\epsilon,k)+\mathrm{Nom}(\epsilon,m+t,n)^*$
	*(number of constraints is $m+t$ as it grows by at least 1 each iteration t)		
Direct FOM via CoMirror	1	$O(1/\epsilon^2)$	$m\,\mathrm{Pess}(\epsilon,k)+O(mn)$

Table 2.1: Summary of different approaches to generate $\{x_t, u_t\}_{t=1}^T$.

feasibility problem for the update of x_t.

The per-iteration cost of the nominal feasibility/optimization oracle-based approach of [25], as well as that of [107], depends on the type of solver used to solve the nominal optimization/feasibility problem (2.24). We denote by $\mathrm{Nom}(\epsilon, m, n)$ the complexity of a nominal oracle with tolerance ϵ, m constraints and n variables. Note that nominal solvers can be either optimization or feasibility solvers. If it is the latter, an extra $\log(1/\epsilon)$ factor is incurred to perform binary search. A summary of different possible implementations for $\mathrm{Nom}(\epsilon, m, n)$ is given in Table 2.3. When applicable for $\mathrm{Nom}(\epsilon, m, n)$ implementation, polynomial-time IPMs are guaranteed to terminate in $O(\sqrt{m}\log(1/\epsilon))$ iterations with a solution to (2.24) and thus offer the best rates in terms of their dependence on ϵ. They also have the advantage that they can act as a nominal optimization oracle, and hence by Remark 2.14 there will be no need to perform an additional binary search to find an ϵ-optimal solution. On the other hand, they demand significantly more memory, and their per-iteration cost is quite high in terms of the dimension, usually around the order of $O(n^3+mn)$ [17, Chapter 6.6]. In order to keep both the memory requirements and the per-iteration cost associated with implementing the nominal feasibility oracle $\mathrm{Nom}(\epsilon, m, n)$ low, one may opt for a FOM called the CoMirror algorithm that can work with functional constraints, see [14] and Juditsky and Nemirovski [93, Section 5.3]. CoMirror algorithm is guaranteed to find a solution to the nominal ϵ-feasibility problem within $O(1/\epsilon^2)$ iterations, with a much cheaper per-iteration cost of $O(mn)$. Because CoMirror method can optimize as well, it does not need binary search. However, to the best of our knowledge, its possibility to exploit further structural properties of the functions f^i, such as smoothness in x, to improve the dependence on ϵ are not known. In order to exploit such properties in the implementation of $\mathrm{Nom}(\epsilon, m, n)$, it is possible to cast (2.24) as a convex-concave SP problem, and then apply efficient FOMs such as Nesterov's algorithm [116] or Nemirovski's Mirror Prox algorithm [112] to achieve a convergence rate of $O(\log(m)/\epsilon)$ and per-iteration cost of $O(mn)$. This convex-concave SP approach can only be used as a nominal feasibility oracle, so we must repeat the process $\log(1/\epsilon)$ times to obtain an ϵ-optimal solution.

Recall that Table 2.1 summarizes the rates for the various approaches, together with rates for the full pessimization approach of [107] and using the CoMirror with pessimization (discussed in Section 2.4.2). Note that the total *overall arithmetic complexity* of each approach is obtained by multiplying the quantities in each row in Table 2.1. The quantities $r_u(\epsilon), r_x(\epsilon)$ will generally be $O(1/\epsilon^2)$, with potential for application-specific acceleration when the functions f^i exhibit favorable structure. Table 2.1 indicates that our FO-based pessimization approach when it admits a closed

Implementation	$\text{Pess}(\epsilon, k)$
Closed form	$O(k)$
IPM	$O(k^3 \log(1/\epsilon))$
FOM*	$O\left(k \log(1/\epsilon)\right)$

*(when f^i are smooth, strongly convex in u^i)

Table 2.2: Arithmetic complexity for different implementations of pessimization oracles.

Implementation	$\text{Nom}(\epsilon, m, n)$	Type	Binary search
IPM	$O\left(km + \sqrt{m}(n^3 + mn)\log(1/\epsilon)\right)$	optimization	1
CoMirror	$O\left(mn/\epsilon^2\right)$	optimization	1
Convex-concave SP*	$O\left(\log(m)mn/\sqrt{\epsilon}\right)$	feasibility	$\log(1/\epsilon)$

*(when f^i are smooth, strongly convex in x)

Table 2.3: Arithmetic complexity for different implementations of nominal oracles

.

form solution for the implementation of $\text{Pess}(\epsilon, k)$ and the nominal feasibility oracle-based approach which uses a polynomial-time IPM solver to implement the nominal feasibility oracle $\text{Nom}(\epsilon, m, n)$ give the best dependence on ϵ among all of the methods. These are better than our OFO-based approach by factors of $\max\{1, r_u(\epsilon)/r_x(\epsilon)\}$ and $\max\{1, r_x(\epsilon)/r_u(\epsilon)\}$ respectively. However, in many applications, we can expect that $r_u(\epsilon) \approx r_x(\epsilon)$, so these factors will be constant. In this case, our OFO-based approach becomes competitive with having a closed form pessimization oracle in our FO-based pessimization approach or using a nominal IPM solver in [25]. That said, compared to IPMs, our OFO-based approach demands much less memory, and it is able to maintain a much lower dependence on the dimensions m, n and thus is much more scalable, whereas the cost per iteration of such IPMs has a rather high dependence on the dimension. In addition, the memory requirements of IPMs are far more than OFO algorithms, posing a critical disadvantage to their use in large-scale applications. Similar comparisons of our OFO-based approach against pessimization or nominal feasibility oracle-based approaches utilizing other methods point out its advantage, which is at least an order of magnitude better in terms of its dependence on ϵ. In fact, when $r_x(\epsilon) \approx r_u(\epsilon)$, our method can lead to savings over the approach of [25] with CoMirror algorithm used in its oracle by a factor as large as $O(1/(\epsilon^2 \log(1/\epsilon)))$.

2.4.2 Connections with Existing First-order Methods

Finally, we would like to discuss and contrast directly solving robust convex optimization problems (1.2) via general first-order methods. Many FOMs require domains that are simple so that the prox operations can be easily done. In that respect, domains defined by multiple functional constraints $g^i(x) \leq 0$ creates a challenge for directly applying many of these algorithms. We now discuss two existing classes of FOMs that are designed to handle such domains: primal-dual methods and the CoMirror approach. Applying these FOMs to the RO problem (1.2) can be viewed as another alternative solution methodology to solve RO problems without using the robust counterpart.

A general technique to address the functional constraints in the domain is to embed these constraints into the objective through Lagrange multipliers, and then solve the associated dual

problem via FOMs (see e.g., Nedić and Özdağlar [109]). Such methods are known as primal-dual methods. For the RO problem (1.2), this corresponds to solving

$$\max_{\lambda} \left\{ \mathcal{L}^*(\lambda) := \min_{x \in X} \left[f^0(x) + \sum_{i=1}^{m} \lambda^{(i)} g^i(x) \right] : \lambda \geq 0 \right\},$$

where we define $g^i(x) := \sup_{u^i \in U^i} f^i(x, u^i)$. Primal-dual methods (e.g., Nedić and Özdağlar [109]) commonly require us to solve the inner minimization problem over $x \in X$ at each iteration. For RO, this means we must solve an expensive SP problem at each iteration. Our OFO-based approach aims to improve on this by reducing the per-iteration cost of each step to simple first-order updates. Two exceptions within the primal-dual methods are the work of Nedić and Özdağlar [110] and Yu and Neely [154], which have cheap per-iteration cost based on only gradient computations and projection operations in the Euclidean setup. Nedić and Özdağlar [110] provide a convergence rate of $O(1/\sqrt{T})$ in the non-smooth case. While using such a primal-dual method has the advantage that no binary search is needed, we note that this requires two assumptions to guarantee convergence: we have access to exact first-order information for the robust constraint functions $g^i(x) := \sup_{u^i \in U^i} f^i(x, u^i)$, and the standard Slater constraint qualification condition (i.e., strict feasibility) is satisfied. The first assumption is often not satisfied, since we may only be able to compute $g^i(x)$ up to accuracy ϵ. While there exists some FOMs that work with inexact objective gradients over simple domains, see e.g., Devolder et al. [53], such methods have only been applied to specific max-type objectives, e.g., objectives obtained from smoothing. It is unclear how such methods can be extended for more general max-type functions which can arise in RO. Secondly, enforcing the Slater condition implicitly enforces feasibility of (1.2). In contrast, our framework directly uses the functions $f^i(x, u^i)$, so it does not need to take into account the inexact gradient information, and can certify infeasibility of (1.2). Yu and Neely [154] present a method that can guarantee $O(1/T)$ convergence when all functions are smooth. However, for RO problems, the constraint functions $g^i(x)$ are non-smooth due to the supremum operation, thus their results do not apply to RO.

The only FOM that we are aware of that can solve convex problems with functional constraints without assuming feasibility is the CoMirror algorithm [14] and its earlier variations in the Euclidean setup [113, 115, 121]. The CoMirror* algorithm finds an ϵ-optimal ϵ-feasible solution in $O(1/\epsilon^2)$ iterations to a convex program $\min_{x \in X} \left\{ f^0(x) : g^i(x) \leq 0,\ i \in [m] \right\}$ or certifies its infeasibility by using (sub)gradient information of the objective f^0 as well as the constraint functions g^i. In the RO problem (1.2) we defined $g^i(x) := \sup_{u^i \in U^i} f^i(x, u^i)$. As mentioned above, in many cases, we may only be able to compute $g^i(x)$ approximately, thus only have access to approximate/inexact gradient information. It is unknown to us whether or not techniques such as the ones from Devolder et al. [53] can be applied to the CoMirror algorithm in the presence of this type of gradient information. While the CoMirror algorithm's complexity is $O(1/\epsilon^2)$ (see also Nesterov [115, Chapter 3.2.4] for a similar result in the Euclidean case), our iterative framework can exploit favorable structure on the functions f^i that can improve on the iteration complexity $r_u(\epsilon), r_x(\epsilon)$. For the Euclidean case, Nesterov [115, Chapters 2.3.4-2.3.5] shows also that convergence can be obtained in $O(\log(1/\epsilon))$ iterations when the objective and all constraint functions are both smooth and strongly convex in x. However, such an improvement does not apply to the RO problem, since we cannot in general

*Recall that the CoMirror algorithm is also discussed in Section 2.4.1.3 as a method to implement the nominal feasibility solver; in that case we are given the noises $\bar{u}^i$ resulting in $g^i(x) := f^i(x, \bar{u}^i)$, and thus the subgradient of $g^i(x)$ is simply the subgradient of $f^i(x, \bar{u}^i)$.

guarantee that $g^i(x) = \sup_{u^i \in U^i} f^i(x, u^i)$ is smooth in x. It is unknown whether the iteration complexity of CoMirror algorithm can be improved when only the underlying function $f^i(x, u^i)$ is strongly convex or smooth, or when $g^i(x)$ is strongly convex but non-smooth.

2.5 Application to Joint Estimation-Optimization

We now consider the JEO problem (1.5)-(1.6), which we restate below for convenience:

$$\min_x \{f(x, u) : \ x \in X\} \tag{2.26}$$

$$\text{given only } \{u_t\}_{t \geq 1}, \text{ where } u_t \to u. \tag{2.27}$$

To apply our primal-dual framework, we write (2.26) as a SP problem, under the following assumption on $f(x, u)$:

Condition 2.17. For some domain Y and matrix B, we can represent

$$f(x, u) = \max_{y \in Y} \{\langle Bx, y\rangle - \phi(y, u)\}.$$

Note that such a representation always exists through convex conjugacy; see Example 2.2. Using the definition of $\Psi(x, y; u) = \langle Bx, y\rangle - \phi(y, u)$, (2.26) can be represented as the following SP problem:

$$\min_{x \in X} f(x, u) = \min_{x \in X} \max_{y \in Y} \{\langle Bx, y\rangle - \phi(y, u)\} = \min_{x \in X} \max_{y \in Y} \Psi(x, y; u). \tag{2.28}$$

Now, the optimality gap of a point $\frac{1}{\Theta_T} \sum_{t \in [T]} \theta_t x_t$ is bounded by the saddle point gap (given any $\bar{y}_T^\theta = \frac{1}{\Theta_T} \sum_{t \in [T]} \theta_t y_t$):

$$f(\bar{x}_T^\theta, u) - \min_{x \in X} f(x, u) \leq \max_{y \in Y} \Psi(\bar{x}_T^\theta, y; u) - \min_{x \in X} \Psi(x, \bar{y}_T^\theta; u), \tag{2.29}$$

which is bounded by the three terms in Theorem 2.4,

$$\widehat{\epsilon}\left(\{x_t, y_t, u_t, \theta_t\}_{t \in [T]}\right) = \max_{x \in X, y \in Y} \frac{1}{\Theta_T} \sum_{t \in [T]} [\langle Bx_t, y\rangle - \phi(y, u_t) - \langle Bx, y_t\rangle + \phi(y_t, u_t)] \tag{2.30}$$

$$\epsilon^\circ\left(\{x_t, u_t, \theta_t\}_{t \in [T]}; u\right) = \max_{y \in Y} \frac{1}{\Theta_T} \sum_{t \in [T]} \theta_t [\phi(y, u) - \phi(y, u_t)] \tag{2.31}$$

$$\epsilon^\bullet\left(\{y_t, u_t, \theta_t\}_{t \in [T]}; u\right) = \frac{1}{\Theta_T} \sum_{t \in [T]} \theta_t [\phi(y_t, u_t) - \phi(y_t, u)]. \tag{2.32}$$

As mentioned above, we will present algorithms to bound $\widehat{\epsilon}$ in Chapter 3, so we instead examine the latter two terms $\epsilon^\circ, \epsilon^\bullet$. We can think of these two terms as capturing *the price of estimation*, i.e., they are the error incurred from using inexact estimates $u_t \neq u$. Indeed, if u were known and we had $u_t = u$ for all t, then these terms will disappear.

We now give a sufficient condition that ensures $\lim_{T \to \infty} \{\epsilon^\circ + \epsilon^\bullet\} \leq 0$ whenever $u_t \to u$.

Condition 2.18. Fix a norm $\|\cdot\|$ on u vectors. For each $y \in Y$, $\phi(y, u)$ is Lipschitz continuous in u for norm $\|\cdot\|$ with constant L_y, i.e., $|\phi(y, u') - \phi(y, u)| \leq L_y \|u' - u\|$. In addition, $L := \max_{y \in Y} L_y < \infty$.

Theorem 2.19. *Suppose that Conditions 2.17 and 2.18 hold, and that $\{\theta_t\}_{t\geq 1}$ is chosen so that $\Theta_T = \sum_{t\in[T]} \theta_t \to \infty$. Then, whenever $u_t \to u$ for any fixed u,*

$$\lim_{T\to\infty} \left[\epsilon^\circ\left(\{x_t, u_t, \theta_t\}_{t\in[T]}; u\right) + \epsilon^\bullet\left(\{y_t, u_t, \theta_t\}_{t\in[T]}; u\right)\right] \leq 0.$$

Proof. First observe that for any $t \geq 1$, $y \in Y$ and u, by Condition 2.18,

$$\phi(y,u) - \phi(y,u_t) \leq L_y\|u - u_t\| \leq L\|u - u_t\|, \quad \phi(y_t,u_t) - \phi(y_t,u) \leq L_{y_t}\|u_t - u\| \leq L\|u_t - u\|.$$

This implies that

$$\epsilon^\circ\left(\{x_t, u_t, \theta_t\}_{t\in[T]}; u\right) + \epsilon^\bullet\left(\{y_t, u_t, \theta_t\}_{t\in[T]}; u\right) \leq \frac{2L}{\Theta_T}\sum_{t\in[T]} \theta_t\|u_t - u\|.$$

We now show the following:

$$a_t \to 0 \implies \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t a_t \to 0.$$

To get our result, we apply this to the sequence $a_t = 2L\|u_t - u\|$, which converges to 0 since $u_t \to u$. Fix some $\epsilon > 0$, and choose $S(\epsilon) \in \mathbb{N}$ sufficiently large such that for $t \geq S(\epsilon)$, $|a_t| \leq \epsilon/3$. Furthermore, choose T sufficiently large such that $\left|\frac{1}{\Theta_T}\sum_{t\in[S(\epsilon)]} \theta_t a_t\right| \leq \epsilon/2$ and $\left|\frac{1}{\Theta_T}\sum_{t\in[S(\epsilon)]} \theta_t\right| \leq 1/2$. We have

$$\left|\frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t a_t\right| \leq \left|\frac{1}{\Theta_T}\sum_{t\in[S(\epsilon)]} \theta_t a_t\right| + \left|\frac{1}{\Theta_T}\sum_{t=S(\epsilon)+1}^{T} \theta_t a_t\right|.$$

The first term is $\leq \epsilon/2$ by our choice of T, and the second term is also, since

$$\left|\frac{1}{\Theta_T}\sum_{t=S(\epsilon)+1}^{T} \theta_t a_t\right| \leq \frac{1}{\Theta_T}\sum_{t=S(\epsilon)+1}^{T} \theta_t|a_t| \leq \frac{\epsilon}{3\Theta_T}\sum_{t=S(\epsilon)+1}^{T} \theta_t = \frac{\epsilon}{3}\left(1 - \frac{1}{\Theta_T}\sum_{t=1}^{S(\epsilon)} \theta_t\right) \leq \frac{\epsilon}{2}.$$

□

Theorem 2.19 ensures that the error terms will be arbitrarily small if we make T large enough. Thus, the SP gap (2.29), and hence also the optimality gap of (2.26), can be made arbitrarily small. However, the rate at which the SP gap converges depends on the rate that $\|u_t - u\| \to 0$, and in fact is slower. This is quite a natural consequence, since in essence it supports the intuition that the performance is limited by the quality of given information $\{u_t\}_{t\geq 1}$; indeed, it is unreasonable to expect that faster rates are possible without assumptions on the dynamics of the sequence $\{u_t\}_{t\geq 1}$ beyond convergence.

Finally, we state the convergence rate of $\frac{2L}{\Theta_T}\sum_{t\in[T]} \theta_t\|u_t - u\| \to 0$ for different possible rates of $\|u_t - u\| \to 0$, as well as two common choices for θ_t, namely $\theta_t = 1$ and $\theta_t = t$. In Chapter 3, we will see how the choice of θ affects the regret bounds for $\widehat{\epsilon}$.

Proposition 2.20. *The convergence rates in Table 2.4 hold.*

rate that $\frac{2L}{\Theta_T}\sum_{t\in[T]}\theta_t\|u_t-u\|\to 0$	$\theta_t=1$, $\Theta_T=T$	$\theta_t=t$, $\Theta_T=\frac{T(T+1)}{2}$
$\|u_t-u\|=O(1/t^r)$, $r\in(0,1)$,	$\sim 1/T^r$	$\sim 1/T^r$
$\|u_t-u\|=O(1/t)$	$\sim \log(T)/T$	$\sim 1/T$
$\|u_t-u\|=O(1/t^r)$, $r\in(1,2)$,	$\sim 1/T$	$\sim 1/T^r$
$\|u_t-u\|=O(1/t^2)$	$\sim 1/T$	$\sim \log(T)/T^2$
$\|u_t-u\|=O(1/t^r)$, $r>2$	$\sim 1/T$	$\sim 1/T^2$
$\|u_t-u\|=O(\beta^t)$, $\beta\in(0,1)$	$\sim 1/T$	$\sim 1/T^2$

Table 2.4: Convergence rate of bound for $\epsilon^\circ+\epsilon^\bullet$.

Proof. First, we analyse $S(r,T):=\sum_{t\in[T]}\frac{1}{t^r}$ for $r\neq 1,2$. Observe that

$$\frac{1}{1-r}\left(\frac{1}{T^{r-1}}-1\right)=\int_{t=1}^{T}\frac{1}{t^r}dt\le S(r,T)\le\left(1+\int_{t=1}^{T}\frac{1}{t^r}dt\right)=\frac{1}{1-r}\left(\frac{1}{T^{r-1}}-r\right).$$

- We consider the case $\theta_t=1$, $\Theta_T=T$, $\|u_t-u\|=O(1/t^r)$, $r>0$. In this case we have

$$\frac{1}{1-r}\left(\frac{1}{T^r}-\frac{1}{T}\right)\le\frac{2L}{\Theta_T}\sum_{t\in[T]}\theta_t\|u_t-u\|\sim\frac{1}{T}S(r,T)\le\frac{1}{1-r}\left(\frac{1}{T^r}-\frac{r}{T}\right).$$

  When $r<1$, $1-r>0$ and $1/T=O(1/T^r)$, hence the lower and upper bounds are $\sim 1/T^r$. When $r>1$, $1-r<0$ and $1/T^r=O(1/T)$, hence the lower and upper bounds are $\sim 1/T$.

- Now consider the case $\theta_t=t$, $\Theta_T=2/(T(T+1))$, $\|u_t-u\|=O(1/t^r)$, $r>0$. In this case we have

$$\begin{aligned}\frac{2}{2-r}\left(\frac{1}{T^{r-1}(T+1)}-\frac{1}{T(T+1)}\right)&\le\frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t\|u_t-u\|\\&\sim\frac{2}{T(T+1)}S(r-1,T)\le\frac{2}{2-r}\left(\frac{1}{T^{r-1}(T+1)}-\frac{r-1}{T(T+1)}\right).\end{aligned}$$

  When $r<2$, $2-r>0$ and $1/(T(T+1))=O(1/(T^{r-1}(T+1)))=O(1/T^r)$, hence the lower and upper bounds are $\sim 1/T^r$. When $r>1$, $1-r<0$ and $1/(T^{r-1}(T+1))=O(1/(T(T+1)))=O(1/T^2)$, hence the lower and upper bounds are $\sim 1/T^2$.

- Now consider the case $\theta_t=1$, $\Theta_T=T$, $\|u_t-u\|=O(1/t)$. Then

$$\frac{2L}{\Theta_T}\sum_{t\in[T]}\theta_t\|u_t-u\|\sim\frac{1}{T}\sum_{t\in[T]}\frac{1}{t}\sim\frac{\log(T)}{T}.$$

- Now consider the case $\theta_t=t$, $\Theta_T=2/(T(T+1))$, $\|u_t-u\|=O(1/t^2)$. Then

$$\frac{2L}{\Theta_T}\sum_{t\in[T]}\theta_t\|u_t-u\|\sim\frac{1}{T(T+1)}\sum_{t\in[T]}\frac{1}{t}\sim\frac{\log(T)}{T^2}.$$

Finally, consider the case $\|u_t - u\| = O(\beta^t)$ for $\beta \in (0,1)$. When $\theta_t = 1$, $\Theta_T = T$, we have

$$\frac{2L}{\Theta_T} \sum_{t\in[T]} \theta_t \|u_t - u\| = \frac{2L}{\Theta_T} \sum_{t\in[T]} \beta^t \frac{2L\beta(1-\beta^T)}{T(1-\beta)} \sim 1/T.$$

When $\theta_t = t$, $\Theta_T = 2/(T(T+1))$, we have

$$\frac{4L}{T(T+1)} \sum_{t\in[T]} \theta_t \|u_t - u\| = \frac{4L}{T(T+1)} \sum_{t\in[T]} t\beta^t \frac{4L\beta(1-(T+1)\beta^T + T\beta^{T+1})}{T(T+1)(1-\beta)^2} \sim 1/T^2.$$

□

Chapter 3

Online Convex Optimization Algorithms

3.1 Introduction

In Chapter 2, we presented a unified primal-dual framework for solving robust optimization (RO) and joint estimation-optimization (JEO) problems. These generate a solution $\bar{x}$ in very similar ways: iteratively generate a *primal-dual* sequence $\{x_t, y_t\}_{t\in[T]}$ based on a *data* sequence $\{u_t\}_{t\in[T]}$, then perform averaging after a finite number of iterations T to build an approximate solution $\bar{x}$. We assume very little about the data sequence $\{u_t\}_{t\in[T]}$; in the case of JEO, we only know that it converges to the 'ideal' data u, and in RO we actually generate it ourselves, thus the data sequence is quite intricately dependent on the primal-dual sequence $\{x_t, y_t\}_{t\in[T]}$. To handle this complexity in both problems, we use tools from *online convex optimization* (OCO).

OCO is part of the broader online learning (or sequential prediction) framework, which was introduced as a method to optimize decisions in a dynamic environment where the objective is changing at every time period, and at each time period we are allowed to adapt to our changing environment based on accumulated information. The origin of the online learning model can be traced back to the work of Robbins [127] on compound statistical decision problems. This framework has found a diverse set of applications in many fields; for further details see [49, 73, 135].

In standard OCO, we are given a convex domain X and a finite time horizon T. In each time period $t = 1, \ldots, T$, an online player chooses a decision $x_t \in X$ based on *past* information from time steps $1, \ldots, t-1$ only. Then, a convex loss function $f_t : X \to \mathbb{R}$ is revealed, and the player suffers loss $f_t(x_t)$ and gets some feedback typically in the form of first-order information $\nabla f_t(x_t)$. We call this restriction on the player *non-anticipatory*, since the player cannot anticipate the next loss f_t ahead of deciding x_t.* In addition, it is usually assumed that the functions f_t are set in advance—possibly by an all-powerful adversary that has full knowledge of our learning algorithm—and we know of only the general class of these functions. As such, it is unreasonable to compare the loss of the player across the time horizon to the best possible loss, which would require full knowledge of f_t in advance of choosing x_t. Instead, the player's sequence of decisions x_t is evaluated against

*This is also referred to as a *0-lookahead* framework.

the best fixed decision in hindsight, and the (average) difference is defined to be the *regret*:

$$\frac{1}{T}\sum_{t=1}^{T} f_t(x_t) - \inf_{x\in X} \frac{1}{T}\sum_{t=1}^{T} f_t(x). \tag{3.1}$$

The goal in OCO is to design efficient *regret minimizing* algorithms that generate the points x_t so that the regret tends to zero as T increases. Therefore, in OCO we seek non-anticipatory algorithms to choose x_t that guarantee

$$\frac{1}{T}\sum_{t=1}^{T} f_t(x_t) - \inf_{x\in X} \frac{1}{T}\sum_{t=1}^{T} f_t(x) \le r(T), \quad \lim_{T\to\infty} r(T) = 0,$$

and the performance of our algorithms is measured by how quickly $r(T)$ tends to 0. While regret may seem like a weak evaluation metric, the fact that regret minimizing algorithms exist for *any* sequence of functions f_t is quite powerful. In particular, it allows us to handle the intricacies of simultaneously generating x_t and u_t.

3.1.1 Related Literature

In this chapter, we give an overview of OCO algorithms which are instrumental in our primal-dual framework RO and JEO introduced in Chapter 2. In particular, we explore the introduction of three simple flexibilities to the standard OCO framework: weighted regret, online saddle point (SP) problems and lookahead decisions (defined formally in Sections 3.4.1 and 3.6). These are permissible in the primal-dual framework, and allow us to exploit structural properties of certain problems.

To our knowledge, the concept of weighted regret in OCO has not been studied prior to our paper [80]. However, modification of aggregation weights as a means to speed up convergence has been explored in the stochastic optimization setting under strong convexity assumptions; see [75, 100, 108, 125]. Our work can be seen as an extension of these results to the adversarial setting, and in fact, one of our results, Theorem 3.12, is a simple generalization of a result from [100]. Nevertheless, by stating the result in the general adversarial setting of OCO, we are able to apply it to RO and JEO, which do not fit within the stochastic optimization framework. Subsequent to our paper [80], Abernethy et al. [2], Wang and Abernethy [148] use weighted regret to analyse offline convex optimization algorithms through the regret framework.

Mahdavi et al. [104] introduce a special case of online SP problems to handle difficult constraints in OCO problems. The difficult constraints $s^i(x) \le 0$ are embedded into each loss function $f_t(x)$ by aggregation with Lagrange dual multipliers y, to form a new loss function $\phi_t(x,y) = f_t(x) + \sum_{i=1}^{m} y^{(i)} s^i(x)$, which is convex in x and concave in y. Both primal and dual variables x, y are then updated each time step to obtain bounds on the regret and the violation $\sum_{t=1}^{T} s^i(x)$. The papers [98, 89] also use similar duality ideas for handling difficult constraints and objectives in online settings. Nevertheless, the convergence rates given in these papers are the usual $O(1/\sqrt{T})$ or slower. In this paper, we analyze online SP problems more generally, and explore faster rates in the 1-lookahead setting.

Online settings with 1-lookahead naturally arise in metrical task systems [37, 40, 7] and online display advertising [89]. In these settings, the variation of the decisions $x_1, \ldots, x_T$ across the time horizon is also penalized, and the performance of the sequence is measured as the competitive ratio of the realized loss with the best possible loss [37, 40, 7] or as a dynamic regret term [89]. Both

competitive ratio and dynamic regret objectives do not fit to our framework. Moreover, [7, Section 4] show that standard regret and competitive ratio cannot be simultaneously optimized.

From an algorithmic point of view, Mahdavi et al. [104], Chiang et al. [50] and Yang et al. [152] examine online variants of the Mirror Prox algorithm when it is limited to work with *only past information*. In particular, [50, 152] provide regret bounds with 'gradual variation' terms, which capture how quickly the sequence of functions f_t vary. Rakhlin and Sridharan [123, 124] analyze 1-lookahead decisions in OCO through the lens of *predictable sequences*. They explore how one can exploit information from a single sequence $M_1, \dots, M_T$ in an online framework, where each term M_t is revealed to the player *prior* to choosing the decision x_t. They provide the Optimistic Mirror Descent algorithm, which is essentially a generalization of Mirror Prox [112], to exploit the sequence $M_1, \dots, M_T$. Rakhlin and Sridharan [123, 124] focus on uncoupled dynamics and zero-sum games, whereas our work focuses on more general and flexible OCO problems, and designing and applying proper generalizations of FOMs such as Mirror Prox to more flexible OCO problems arising in the context of coupled optimization problems. That said, our work in Section 3.6 is related to exploiting a specific predictable sequence; we elaborate on this in Remark 3.31.

3.1.2 Contributions

In this chapter, we explore the introduction of simple flexibilities to the standard OCO framework, which allow us to exploit structural properties of RO and JEO and results in improved convergence rates. We explore three simple modifications to the standard OCO assumptions:

- We introduce the concept of *weighted regret*, where instead of taking uniform averages with weights $\theta_t = 1/T$ in (3.1), we are allowed to use *nonuniform* weighted averages. From a modeling perspective, this allows us to capture situations where decisions x_t at different time steps t have varying importance.

- We introduce the *online saddle point (SP) problem*, where at each step we receive a convex-concave function $\phi_t(x, y)$ and must choose x and y. This is an extension of the well-studied offline convex-concave SP problem, and can be thought of as a dynamic zero-sum two-player game where at each step the players are restricted to make only one move.

- We explore the implications of *1-lookahead* or *anticipatory* decisions, where the learner can receive limited information (e.g., gradient) on the function f_t before making the decision x_t. This is in contrast to most OCO settings where the learner must choose x_t before any information on f_t is revealed.

Under this new OCO framework with flexibilities, we present and discuss algorithms accompanied with new regret bounds that can be better than the standard OCO ones when favorable problem structure is present. Our algorithms are based on online adaptations of two commonly used offline first-order methods (FOMs) from convex optimization, namely Mirror Descent and Mirror Prox. We present our developments in the flexible proximal setup of Juditsky and Nemirovski [93, 94] which can be further customized to the geometry of the domains.

Our analyses demonstrate that these flexibilities introduced into the OCO framework have significant consequences whenever they are applicable. For example, in the strongly convex case, minimizing unweighted regret has a proven optimal bound of $O(\log(T)/T)$, whereas we show that a bound of $O(1/T)$ is possible when we consider weighted regret. Similarly, for the smooth case, considering 1-lookahead decisions results in a $O(1/T)$ bound, compared to $O(1/\sqrt{T})$ in the standard OCO setting (see Remarks 3.14 and 3.32). Consequently, these new regret bounds are pivotal in

exploiting structural properties of functions to achieve improved convergence rates for both RO and JEO.

3.2 Weighted Regret and Online Saddle Point Problems

Suppose we have weights $\theta_t > 0$, $t \geq 1$. As in Chapter 2, we denote $\Theta_T = \sum_{t\in[T]} \theta_t$. The *weighted regret* is defined by simply scaling each time step t of the usual regret (3.1) by weight θ_t, then re-normalizing by Θ_T:

$$\frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f_t(x_t) - \min_{x\in X} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f_t(x). \tag{3.2}$$

We seek algorithms for choosing selecting $\{x_t\}_{t\geq 1}$ that guarantee

$$\frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f_t(x_t) - \min_{x\in X} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f_t(x) \leq r(T), \quad \lim_{T\to\infty} r(T) = 0.$$

From a modeling perspective, weighted regret enables us to model situations where later decisions x_t carry higher importance by placing higher weights θ_t on subsequent periods t (or vice versa). For example, in a repeated game where performance of a player is aggregated from the loss at each stage, we may want to weigh the later stages more heavily than the earlier stages, since earlier stages might be used to explore the opponents' strategy, whereas in later stages we expect the player to have converged to a (near)-optimal strategy.

On the practical side, weighted regret lets us choose weights θ_t to speed up convergence. In particular, when the functions f_t are strongly convex, our bounds of $O(1/T)$ for weighted regret improve on the optimal regret bounds $O(\log(T)/T)$ for the uniform weight case. Furthermore, weighted regret bounds become important in the primal-dual framework of Chapter 2, where we combine two regret terms *which must have the same weights* θ to obtain bounds. Note that while it is possible to view weighted regret as a rescaling of the functions f_t with weights θ_t, such a view will inevitably change the parameters associated with functions f_t such as strong convexity. In contrast, working with the weighted regret concept circumvents this issue; see Section 3.5 for our study on exploiting strong convexity.

In Chapter 2.2, we introduced the (offline) saddle point (SP) problem (2.1). A natural extension of SP problems to an online setup is as follows: We are given domains X, Y and a time horizon T. At each time period $t \in [T]$, we simultaneously select $(x_t, y_t) \in X \times Y$ and receive $\Psi_t(x_t, y_t)$ based on a convex-concave function $\Psi_t(x, y)$ revealed at the time period. We can think of this as a dynamic two-player zero-sum game, where at each stage t, each player makes only one move (decision) $x_t \in X$ and $y_t \in Y$ as opposed to reaching to an approximate equilibrium. Then the goal of each player is to minimize their weighted regrets given the sequence of moves of the other player, i.e.,

$$\frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t \Psi_t(x_t, y_t) - \min_{x\in X} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t \Psi_t(x, y_t) \quad \text{and} \quad \max_{y\in Y} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t \Psi_t(x_t, y) - \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t \Psi_t(x_t, y_t).^{\dagger}$$

In this setup, we assume that at each period t, the decisions and actions (queries made to the function Ψ_t) of each player, i.e., x_t and y_t, are revealed to the other and vice versa *immediately*

[†]Note that the y-player receives a *concave reward* $\Psi_t(x_t, y_t)$ at each time step, so their regret is written with the supremum.

after they make their decision or action. This revealed information from period t can then be used by both players in their subsequent decisions and actions in the same period t or in future rounds $t+1$ and so on. We define the *weighted online SP gap* to be the sum of these weighted regrets (i.e., the average social loss):

$$\max_{y\in Y} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \Psi_t(x_t, y) - \min_{x\in X} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \Psi_t(x, y_t). \tag{3.3}$$

We call the problem of minimizing the weighted online SP gap the *online SP problem*. More precisely, the online SP problem seeks algorithms to generate $\{x_t, y_t\}_{t\geq 1}$ that bound the online SP gap by a term $r(T)$, where $\lim_{T\to\infty} r(T) = 0$. When the functions Ψ_t remain the same throughout the time horizon, i.e., $\Psi_t(x, y) = \Psi(x, y; u)$ for all $t \in [T]$, and $\bar{x}_T^\theta$, $\bar{y}_T^\theta$ are taken to be the weighted sums of $\{x_t\}_{t\in[T]}$, $\{y_t\}_{t\in[T]}$ respectively, the weighted online SP gap naturally bounds the standard SP gap for the underlying offline SP problem, i.e., $\epsilon_{\text{sad}}^\Psi(\bar{x}_T^\theta, \bar{y}_T^\theta)$ in (2.9).

An offline (online) SP problem can be solved by solving two related OCO problems, which can also be interpreted as two regret-minimizing players playing a static (dynamic) zero-sum game. Note that the reverse is not true in general: solving an offline (online) SP problem does not in general give us bounds on the individual regrets of each player.

The online SP gap interpretation of (3.3) is advantageous when we relax the non-anticipatory restriction. In an online setup where 1-lookahead decisions are allowed, by examining specialized algorithms for minimizing the weighted online SP gap (3.3) rather than employing two separate regret-minimization algorithms for the players, we can exploit both the fact that our choices $[x_t; y_t]$ may utilize the current function Ψ_t and any favorable structural properties of the functions Ψ_t such as smoothness. In Section 3.6, we introduce algorithms that minimize the weighted online SP gap (3.3) directly. Our analysis demonstrates that exploiting favorable structural properties of functions Ψ_t plays a crucial role for obtaining better convergence rates for (3.3). See also Remark 3.32.

3.3 Algorithmic Setup

Many OCO algorithms are closely related to offline iterative FOMs. In this section, we first introduce some notation and key concepts related to the proximal setup for FOMs along with general properties of two classical FOMs, namely the *Mirror Descent* and *Mirror Prox* algorithms, that are crucial in our analysis for OCO. We then analyze the general versions of these FOMs to develop upper bounds on the weighted regret and weighted online SP gap. We follow the presentation and notation of the excellent survey [93, 94].

Most FOMs capable of solving OCO and online SP problems are quite flexible in terms of adjusting to the geometry of the problem characterized by its domain $\mathcal{Z}$. In the case of SP problems, the domain is given by $\mathcal{Z} = X \times Y$. The following components are standard in forming the setup for such FOMs and their convergence analysis:

- *Norm*: $\|\cdot\|$ on the Euclidean space $\mathbb{E}$ where the domain $\mathcal{Z}$ lives, along with its dual norm $\|\zeta\|_* := \max_{\|z\|\leq 1} \langle \zeta, z\rangle$.
- *Distance-Generating Function* (d.g.f.): A function $\omega(z) : \mathcal{Z} \to \mathbb{R}$, which is convex and continuous on $\mathcal{Z}$, admits a selection of subgradients $\nabla\omega(z)$ that is continuous on the set $\mathcal{Z}^\circ := \{z \in \mathcal{Z} : \partial\omega(z) \neq \emptyset\}$ (here $\partial\omega(z)$ is a subdifferential of ω taken at z), and is strongly convex with modulus 1 with respect to $\|\cdot\|$:

$$\forall z', z'' \in \mathcal{Z}^\circ : \ \langle \nabla\omega(z') - \nabla\omega(z''),\ z' - z''\rangle \geq \|z' - z''\|^2.$$

- *Bregman distance*: $V_z(z') := \omega(z') - \omega(z) - \langle \nabla\omega(z), z' - z\rangle$ for all $z \in \mathcal{Z}^\circ$ and $u \in \mathcal{Z}$.

  Note that $V_z(z') \geq \frac{1}{2}\|z - z'\|^2 \geq 0$ for all $z \in \mathcal{Z}^\circ$ and $z' \in \mathcal{Z}$ follows from the strong convexity of ω.

- *Prox-mapping*: Given a *prox center* $z \in \mathcal{Z}^\circ$,

$$\text{Prox}_z(\xi) := \arg\min_{z' \in \mathcal{Z}} \left\{ \langle \xi, z' \rangle + V_z(z') \right\} : \mathbb{E} \to \mathcal{Z}^\circ.$$

  When the d.g.f. is taken as the squared ℓ_2-norm, the prox mapping becomes the usual projection operation of the vector $z - \xi$ onto $\mathcal{Z}$.

- ω-*center*: $z_\omega := \arg\min_{z \in \mathcal{Z}} \omega(z)$.

- *Set width*: $\Omega = \Omega_z := \max_{z \in \mathcal{Z}} V_{z_\omega}(z) \leq \max_{z \in \mathcal{Z}} \omega(z) - \min_{z \in \mathcal{Z}} \omega(z)$.

For common domains $\mathcal{Z}$ such as simplex, Euclidean ball, and spectahedron, standard proximal setups, i.e., selection of norm $\|\cdot\|$, d.g.f. $\omega(\cdot)$, the resulting Prox computations and set widths Ω are discussed in [93, Section 5.7].

When we have a decomposable domain $\mathcal{Z} = X \times Y$, we can build a proximal setup for $\mathcal{Z}$ from the individual proximal setups on X and Y. Given a norm $\|\cdot\|_x$ and a d.g.f. $\omega_x(\cdot)$ for the domain X, similarly $\|\cdot\|_y$, $\omega_y(\cdot)$ for the domain Y, and two scalars $\beta_x, \beta_y > 0$, we build the d.g.f. $\omega(z)$ and ω-center z_ω for $\mathcal{Z} = X \times Y$ as

$$\omega(z) = \beta_x \omega_x(x) + \beta_y \omega_y(y) \quad \text{and} \quad z_\omega = [x_{\omega_x}; y_{\omega_y}],$$

where $\omega_x(\cdot)$ and $\omega_y(\cdot)$ as well as x_{ω_x} and y_{ω_y} are customized based on the geometry of the domains X and Y. In this construction, the flexibility in determining the scalars $\beta_x, \beta_y > 0$ is useful in optimizing the overall convergence rate. Moreover, by letting $\xi = [\xi_x; \xi_y]$ and $z = [x; y]$, the prox mapping becomes decomposable as

$$\text{Prox}_z(\xi) = \left[\text{Prox}_x^{\omega_x}\left(\frac{\xi_x}{\beta_x}\right) ; \text{Prox}_y^{\omega_y}\left(\frac{\xi_y}{\beta_y}\right) \right],$$

where $\text{Prox}_x^{\omega_x}(\cdot)$ and $\text{Prox}_y^{\omega_y}(\cdot)$ are respectively prox mappings with respect to $\omega_x(\cdot)$ in domain X and $\omega_y(\cdot)$ in domain Y. We refer the reader to the references [93, Section 5.7.2] and [94, Section 6.3.3] for further details on how to optimally choose the parameters β_x, β_y for SP problems.

3.4 Regret Minimization under Minimal Assumptions

In the most basic setup, our functions f_t (resp. Ψ_t) are convex (resp. convex-concave) and non-smooth. In this case, we analyze a generalization of Mirror Descent, outlined in Algorithm 2 for bounding the weighted regret and weighted online SP gap.

Remark 3.1. In Algorithm 2, computation of z_t depends on only z_{t-1} and ξ_{t-1}. In the following we will examine Algorithm 2 by allowing ξ_{t-1} to depend on only the past information on functions $f_1, \ldots, f_{t-1}$ (or $\Psi_1, \ldots, \Psi_{t-1}$). Then the iterations in Algorithm 2 will be based on solely the past information allowing us to carry out a *non-anticipatory* analysis for Algorithm 2. ∎

Algorithm 2 Generalized Mirror Descent

input: ω-center z_ω, time horizon T, positive step sizes $\{\gamma_t\}_{t\in[T]}$, and a sequence $\{\xi_t\}_{t\in[T]}$.
output: sequence $\{z_t\}_{t\in[T]}$.
$z_1 := z_\omega$.
for $t = 1, \dots, T$ **do**
  $z_{t+1} = \mathrm{Prox}_{z_t}(\gamma_t \xi_t)$
end for

Proposition 3.2 describes a fundamental property exhibited by the Mirror Descent updates. Its proof can be found in [93, Proposition 5.1, Equation 5.13], and we include it here for completeness.

Proposition 3.2. *Suppose that the sequence of vectors $\{z_t\}_{t\in[T]}$ is generated by Algorithm 2 for a given sequence of vectors $\{\xi_t\}_{t\in[T]}$ and step sizes $\gamma_t > 0$ for $t \in [T]$. Then for any $z \in \mathcal{Z}$ and $t \in [T]$, we have*

$$\gamma_t \langle \xi_t, z_t - z \rangle \le V_{z_t}(z) - V_{z_{t+1}}(z) + \frac{1}{2}\gamma_t^2 \|\xi_t\|_*^2. \tag{3.4}$$

Proof. Recall that

$$z_{t+1} = \mathrm{Prox}_{z_t}(\gamma_t \xi_t) = \arg\min_{z\in\mathcal{Z}} \{\gamma_t \langle \xi_t, z\rangle + V_{z_t}(z)\} = \arg\min_{z\in\mathcal{Z}} \{\langle \gamma_t \xi_t - \nabla\omega(z_t), z\rangle + \omega(z)\}.$$

We first prove that, for all $z \in \mathcal{Z}$, $\langle \gamma_t \xi_t - \nabla\omega(z_t) + \nabla\omega(z_{t+1}), z - z_{t+1}\rangle \ge 0$. Fix some $z \in Z$ and consider the function $h_{z_{t+1},z}(s) = \langle \gamma_t \xi_t - \nabla\omega(z_t), z_{t+1} + s(z - z_{t+1})\rangle + \omega(z_{t+1} + s(z - z_{t+1}))$ defined for $s \in [0, 1]$. In general, $h_{z_{t+1},z}$ may not be differentiable since ω may not be, but we know that it is convex, hence subgradients exist, and by definition of z_{t+1} as the minimizer, it is non-decreasing, hence all subgradients of h are non-negative. In particular, all subgradients of $h_{z_{t+1},z}$ at $s = 0$ are non-negative, and it is a simple exercise to check that $\langle \gamma_t \xi_t - \nabla\omega(z_t) + \nabla\omega(z_{t+1}), z - z_{t+1}\rangle$ is one such subgradient. We now know that for all $z \in \mathcal{Z}$,

$$\langle \gamma_t \xi_t - \nabla\omega(z_t) + \nabla\omega(z_{t+1}), z - z_{t+1}\rangle \ge 0.$$

We thus have

$$\begin{aligned}
\gamma_t \langle \xi_t, z_t - z\rangle &\le \langle \nabla\omega(z_{t+1}) - \nabla\omega(z_t), z - z_{t+1}\rangle + \gamma_t \langle \xi_t, z_t - z_{t+1}\rangle \\
&= V_{z_t}(z) - V_{z_{t+1}}(z) - V_{z_t}(z_{t+1}) + \gamma_t \langle \xi_t, z_t - z_{t+1}\rangle \\
&\le V_{z_t}(z) - V_{z_{t+1}}(z) - \frac{1}{2}\|z_t - z_{t+1}\|^2 + \gamma_t \langle \xi_t, z_t - z_{t+1}\rangle \\
&\le V_{z_t}(z) - V_{z_{t+1}}(z) - \frac{1}{2}\|z_t - z_{t+1}\|^2 + \gamma_t \|\xi_t\|_* \|z_t - z_{t+1}\|,
\end{aligned} \tag{3.5}$$

where the second inequality follows by strong convexity of ω and the third inequality follows by the definition of the dual norm. The result now follows by recognizing that $\max_s \{\gamma_t \|\xi_t\|_* s - s^2/2\} = \gamma_t^2 \|\xi_t\|_*^2/2$. □

3.4.1 Weighted Regret

From Proposition 3.2, we may derive a bound on the weighted regret (3.2) in the most general case where our functions $f_t(x)$ need only satisfy convexity and Lipschitz continuity. More precisely, we will assume the following.

Assumption 3.3. A proximal setup of Section 3.3 exists for the domain $\mathcal{Z} = X$. Each function f_t is convex, and there exists $G \in (0, \infty)$ such that the subgradients of f_t are bounded, i.e., $\|\nabla f_t(x)\|_* \leq G$ for all $x \in X$ and $t \in [T]$.

Theorem 3.4. *Suppose Assumption 3.3 holds, and we are given weights $\{\theta_t\}_{t\geq 1}$. Then running Algorithm 2 with $z_t = x_t$, $\xi_t = \theta_t \nabla f_t(x_t)$, and step sizes $\gamma_t = \gamma := \sqrt{\frac{2\Omega}{G^2 \sum_{t\in[T]} \theta_t^2}}$ for all $t \in [T]$ results in*

$$\sum_{t\in[T]} \theta_t f_t(x_t) - \min_{x\in X} \sum_{t\in[T]} \theta_t f_t(x) \leq \sqrt{2\Omega G^2 \sum_{t\in[T]} \theta_t^2}.$$

Consequently, the weighted regret is bounded by

$$\frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t f_t(x_t) - \min_{x\in X} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t f_t(x) \leq \sqrt{2\Omega G^2 \sum_{t\in[T]} \left(\frac{\theta_t}{\Theta_T}\right)^2}.$$

Note that Theorem 3.4 is a simple generalization of the fundamental result of [157]. We include its proof for completeness.

Proof. By summing up (3.4) for $t \in [T]$ and writing $\gamma_t = \gamma$ as a constant we obtain

$$\sum_{t\in[T]} \gamma_t \langle \xi_t, x_t - x\rangle = \gamma \sum_{t\in[T]} \theta_t \langle \nabla f_t(x_t), x_t - x\rangle \leq V_{x_1}(x) - V_{x_{T+1}}(x) + \frac{\gamma^2}{2} \sum_{t\in[T]} \theta_t^2 \|\nabla f_t(x_t)\|_*^2.$$

Because $\|\theta_t \nabla f_t(x_t)\|_* \leq \theta_t G$, $V_{x_1}(x) \leq \Omega$ by our choice of x_1 in Algorithm 2, $-V_{x_{T+1}}(x) \leq 0$, and dividing through by γ, we reach to

$$\sum_{t\in[T]} \theta_t \langle \nabla f_t(x_t), x_t - x\rangle \leq \frac{\Omega}{\gamma} + \frac{\gamma}{2} G^2 \sum_{t\in[T]} \theta_t^2.$$

Optimizing the right hand side over $\gamma \geq 0$ gives us the desired upper bound. The left hand side of inequality in the theorem follows from $\theta_t \geq 0$ for all $t \in [T]$ and the convexity of functions f_t implying for all $x \in X$, $\langle \xi_t, x_t - x\rangle = \theta_t \langle \nabla f_t(x_t), x_t - x\rangle \geq \theta_t f_t(x_t) - \theta_t f_t(x)$. □

The bound on weighted regret in Theorem 3.4 is optimized when the weights θ_t are set to be *uniform*, i.e., $\theta_t = 1$; in this case, the regret bound becomes $O(1/\sqrt{T})$.

Remark 3.5. We would like to highlight the importance of customizing our proximal setup based on the geometry of the domain. In many cases, weighted regret or weighted online SP gap bounds have a dependence on the set width parameter Ω associated with the proximal setup; see e.g., Theorem 3.4. For example, when our domain $X = \Delta_n$, equipping X with a proximal setup based on negative entropy d.g.f. $\omega(x) = \sum_{j=1}^n x^j \log(x^j)$ results in $\Omega = \log(n)$, which is almost dimension independent. Using the Euclidean d.g.f. $\omega(x) = \frac{1}{2}\langle x, x\rangle$ on $X = \Delta_n$ leads to a suboptimal (and dimension-dependent) set width of $\Omega = \sqrt{n}$. Moreover, certain domains admit d.g.f.s that lead to quite efficient Prox computations given either in closed form or by simple computations, taking only $O(n)$ arithmetic operations. Negative entropy d.g.f. over simplex and Euclidean d.g.f. over the Euclidean unit ball are such examples. A possible issue for equipping the simplex with a Euclidean proximal setup is that the prox-mapping (usual projection) no longer has a closed form, but it still can be done efficiently in $O(n \log(n))$ arithmetic operations. See [93] for a complete discussion. ∎

3.4.2 Weighted Online SP Gap

Algorithm 2 can also be utilized in bounding the weighted online SP gap (3.3). In this case, in addition to a convex-concave structure assumption on functions $\Psi_t(x, y)$, we assume boundedness of specific monotone gradient operators associated with $\Psi_t(x, y)$.

Assumption 3.6. A proximal setup of Section 3.3 exists for the domain $\mathcal{Z} = X \times Y$. Each function $\Psi_t(x, y)$ is convex in x and concave in y, and there exists $G \in (0, \infty)$ such that $\|[\nabla_x \Psi_t(x, y); -\nabla_y \Psi_t(x, y)]\|_* \leq G$ for all $x \in X$, $y \in Y$ and $t \in [T]$.

Theorem 3.7. *Suppose Assumption 3.6 holds, and we are given convex combination weights $\theta \in \Delta_T$. Then running Algorithm 2 with $z_t = [x_t; y_t]$, $\xi_t = \theta_t[\nabla_x \Psi_t(x_t, y_t); -\nabla_y \Psi_t(x_t, y_t)]$, and step sizes $\gamma_t = \gamma := \sqrt{\frac{2\Omega}{G^2 \sum_{t\in[T]} \theta_t^2}}$ for all $t \in [T]$ gives us*

$$\max_{y\in Y} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \Psi_t(x_t, y) - \min_{x\in X} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \Psi_t(x, y_t) \leq \sqrt{2\Omega G^2 \sum_{t\in[T]} \left(\frac{\theta_t}{\Theta_T}\right)^2}.$$

Proof. The proof proceeds exactly as the proof of Theorem 3.4 to arrive at

$$\sum_{t\in[T]} \langle \xi_t, z_t - z \rangle \leq \sqrt{2\Omega G^2 \sum_{t\in[T]} \left(\frac{\theta_t}{\Theta_T}\right)^2}$$

for all $z = [x; y] \in X \times Y$. Then, from the convex-concave structure of the function Ψ_t, we have for all $z = [x; y] \in X \times Y$ and all $t \in [T]$,

$$\begin{aligned}
\langle \xi_t, z_t - z \rangle &= \theta_t \langle \nabla_x \Psi_t(x_t, y_t), x_t - x \rangle + \theta_t \langle \nabla_y \Psi_t(x_t, y_t), y - y_t \rangle \\
&\geq \theta_t(\Psi_t(x_t, y_t) - \Psi_t(x, y_t)) + \theta_t(\Psi_t(x_t, y) - \Psi_t(x_t, y_t)) \\
&= \theta_t \Psi_t(x_t, y) - \theta_t \Psi_t(x, y_t).
\end{aligned}$$

The result then follows by combining the inequality above with the inequality that provides the upper bound on the term $\sum_{t\in[T]} \langle \xi_t, z_t - z \rangle$. □

Remark 3.8. Uniform weights $\theta_t = 1$ minimize the upper bounds in Theorems 3.4 and 3.7, resulting in $O(1/\sqrt{T})$ bounds. Moreover, Theorems 3.4 and 3.7 can accommodate a variety of weights $\{\theta_t\}_{t\geq 1}$ via adapting their step sizes γ_t and still achieve bounds of form $O(1/\sqrt{T})$. For example, this is the case when the nonuniform weights $\theta_t = t$ from Theorem 3.12 are used in these results. Employing nonuniform weights becomes more consequential when we have to run several OCO or online SP algorithms in conjunction with each other using the *same* weights θ_t in all of them. This arises, for example, in our primal-dual framework from Chapter 2. ∎

3.5 Exploiting Strong Convexity

When our functions f_t admit further favorable structure in the form of strong convexity, it is possible to customize Algorithm 2 using specific *nonuniform* weights θ_t and achieve a bound of $O(1/T)$, which is significantly better than the standard $O(1/\sqrt{T})$ bound of Theorem 3.4 given by uniform weights. Our developments here are based on the following structural assumption.

Assumption 3.9.

- A proximal setup of Section 3.3 exists for the domain $\mathcal{Z} = X$.
- The loss functions $f_t(x)$ for $t \in [T]$ have the property that the functions $f_t(x) - \alpha\,\omega(x)$ is convex for some $\alpha > 0$ independent of t, or equivalently

$$f_t(x) \le f_t(x') + \langle \nabla f_t(x), x - x' \rangle - \alpha V_x(x'), \quad \forall x, x' \in X,\ t \in [T].$$

- The subgradients of the loss functions are bounded, i.e., there exists $G \in (0, \infty)$ such that $\|\nabla f_t(x)\|_* \le G$ for all $x \in X,\ t \in [T]$.

Remark 3.10. When our proximal setup for X is based on a Euclidean d.g.f. $\omega(x) = \frac{1}{2}\langle x, x\rangle$ and Euclidean norm $\|x\|_2$, then Assumption 3.9 simply states that the functions f_t are α-strongly convex. In this paper, we will abuse terminology slightly and say that f_t is α-strongly convex when $f_t(x) - \alpha\,\omega(x)$ is convex, where the dependence on the d.g.f. ω will be clear from the context. ∎

By selecting the step sizes γ_t and weights θ_t in a clever fashion, we are able to exploit the extra $-\alpha V_{x_t}(x)$ terms to improve the regret bound. This result is a generalization of the offline stochastic gradient descent algorithm equipped with a Euclidean d.g.f. based proximal setup presented in Lacoste-Julien et al. [100] to the online setting with domain X admitting a general proximal setup. We first prove a preliminary result, which we will use for a more detailed analysis in Chapter 5.

Proposition 3.11. *Suppose Assumption 3.9 holds. Suppose the weights θ_t and the step sizes γ_t satisfy the relations $\theta_{t+1}\left(\frac{1}{\gamma_{t+1}} - \alpha\right) \le \frac{\theta_t}{\gamma_t}$ for all $t \ge 1$, and $\theta_1\left(\frac{1}{\gamma_1} - \alpha\right) \le 0$. Algorithm 2 with $z_t = x_t$, $\xi_t = \nabla f_t(x_t)$ and step sizes γ_t results in*

$$\frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f_t(x_t) - \min_{x\in X} \frac{1}{\Theta^T}\sum_{t\in[T]} \theta_t f_t(x) \le \frac{1}{\Theta_T}\sum_{t\in[T]} \frac{\theta_t}{\gamma_t}\left(\gamma_t\langle \nabla f_t(x_t), x_t - x_{t+1}\rangle - V_{x_t}(x_{t+1})\right).$$

Proof. In Algorithm 2, we have

$$x_{t+1} = \mathrm{Prox}_{x_t}\left(\gamma_t \nabla f_t(x_t)\right) = \arg\min_{x\in X}\left\{\langle \gamma_t \nabla f_t(x_t), x\rangle + V_{x_t}(x)\right\}.$$

Since $\nabla_x V_{x_t}(x) = \nabla\omega(x) - \nabla\omega(x_t)$, following the proof of Proposition 3.2 up to the second line of (3.5), we get

$$\gamma_t\langle \nabla f_t(x_t), x_t - x\rangle \le V_{x_t}(x) - V_{x_{t+1}}(x) + \gamma_t\langle \nabla f_t(x_t), x_t - x_{t+1}\rangle - V_{x_t}(x_{t+1}).$$

Subtracting $\alpha\gamma_t V_{x_t}(x)$ from both sides and multiplying by θ_t/γ_t gives us

$$\begin{aligned}\theta_t\left(\langle \nabla f_t(x_t), x_t - x\rangle - \alpha V_{x_t}(x)\right) \le{}& \theta_t\left(\frac{1}{\gamma_t}\left(V_{x_t}(x) - V_{x_{t+1}}(x)\right) - \alpha V_{x_t}(x)\right)\\ &+ \frac{\theta_t}{\gamma_t}\Big(\gamma_t\langle \nabla f_t(x_t), x_t - x_{t+1}\rangle - V_{x_t}(x_{t+1})\Big).\end{aligned}$$

Summing this from $t = 1, \ldots, T$, we get

$$\begin{aligned}\sum_{t\in[T]} \theta_t\left(\langle \nabla f_t(x_t), x_t - x\rangle - \alpha V_{x_t}(x)\right) \le{}& \sum_{t=1}^{T-1}\left(\theta_{t+1}\left(\frac{1}{\gamma_{t+1}} - \alpha\right) - \frac{\theta_t}{\gamma_t}\right) V_{x_{t+1}}(x)\\ &+ \theta^1\left(\frac{1}{\gamma^1} - \alpha\right) V_{x_1}(x) - \frac{\theta^T}{\gamma^T} V_{x_{T+1}}(x)\\ &+ \sum_{t\in[T]} \frac{\theta_t}{\gamma_t}\Big(\gamma_t\langle \nabla f_t(x_t), x_t - x_{t+1}\rangle - V_{x_t}(x_{t+1})\Big).\end{aligned}$$

Now recall that $V_x(x') \geq 0$ holds for any x, x', and also from the theorem hypothesis we have $\left(\theta_{t+1}\left(\frac{1}{\gamma_{t+1}} - \alpha\right) - \frac{\theta_t}{\gamma_t}\right) \leq 0$ and $\theta^1\left(\frac{1}{\gamma^1} - \alpha\right) \leq 0$. Therefore, on the right hand side of the above inequality, the first sum and second and third terms are all nonpositive. The final result follows by recognizing that, since $f_t - \alpha\omega$ is convex, we have $f_t(x_t) - f_t(x) \leq \langle \nabla f_t(x_t), x_t - x\rangle - \alpha V_{x_t}(x)$. □

Theorem 3.12 (Strongly convex MD)**.** *Suppose Assumption 3.9 holds. Let $\theta_t = t$ for $t \in [T]$, $\Theta_T = T(T+1)/2$. Then running Algorithm 2 with $z_t = x_t$, $\xi_t = \nabla f_t(x_t)$, and step sizes $\gamma_t = \frac{2}{\alpha(t+1)}$ for all $t \in [T]$ results in*

$$\frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f_t(x_t) - \min_{x\in X} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f_t(x) \leq \frac{2G^2}{\alpha\,(T+1)}.$$

Proof. By Proposition 3.11, we have

$$\frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f_t(x_t) - \min_{x\in X} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f_t(x) \leq \frac{1}{\Theta_T}\sum_{t\in[T]} \frac{\theta_t}{\gamma_t}\left(\gamma_t\langle \nabla f_t(x_t), x_t - x_{t+1}\rangle - V_{x_t}(x_{t+1})\right).$$

By strong convexity of ω with respect to $\|\cdot\|$, we have $V_{x_t}(x_{t+1}) \geq \frac{1}{2}\|x_t - x_{t+1}\|^2$. Now, by Cauchy-Schwarz, we have

$$\gamma_t\langle \nabla f_t(x_t), x_t - x_{t+1}\rangle - V_{x_t}(x_{t+1}) \leq \gamma_t\|\nabla f_t(x_t)\|_*\|x_t - x_{t+1}\| - \frac{1}{2}\|x_t - x_{t+1}\|^2 \leq \frac{1}{2}\gamma_t^2\|\nabla f_t(x_t)\|_*^2 \leq \frac{1}{2}\gamma_t^2 G^2.$$

Substituting this into the inequality, we have

$$\begin{aligned}\frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f_t(x_t) - \min_{x\in X} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t f_t(x) &\leq \frac{1}{\Theta_T}\sum_{t\in[T]} \frac{\theta_t}{\gamma_t}\left(\gamma_t\langle \nabla f_t(x_t), x_t - x_{t+1}\rangle - V_{x_t}(x_{t+1})\right)\\ &\leq \frac{1}{\Theta_T}\sum_{t\in[T]} \frac{1}{2}\theta_t\gamma_t G^2 = \frac{1}{\Theta_T}\sum_{t\in[T]} \frac{t}{\alpha(t+1)}G^2 \leq \frac{2G^2}{\alpha(T+1)}.\end{aligned}$$

□

Let us revisit Remark 3.5 on customizing the proximal setup based on the geometry of the domain.

Remark 3.13. In contrast to Theorem 3.4, the bound of Theorem 3.12 has no dependence on set width Ω. Nevertheless, customization of the proximal setup, in particular selection of d.g.f. ω plays an important role in Theorem 3.12 through Assumption 3.9. In many cases, it is much more likely to encounter functions f_t that are α-strongly convex in the usual sense, i.e., $f_t(x) - \alpha\|x\|_2^2/2$ is convex, but it may not be possible to ensure the convexity of $f_t(x) - \alpha\,\omega(x)$ with respect to a different d.g.f. ω. In such cases, it is possible (and more desirable) to select a d.g.f. ω that will ensure that the strong convexity requirement of Assumption 3.9 is satisfied. Because the bound of Theorem 3.12 has no dependence on Ω, such a selection of ω will not adversely affect overall the weighted regret bound of Theorem 3.12. ∎

Remark 3.14. For strongly convex losses, Theorem 3.12 establishes an upper bound of $O(1/T)$ on weighted regret. In contrast to this, Hazan and Kale [75] established a lower bound of $O(\log(T)/T)$ for minimizing standard regret in OCO with strongly convex loss functions. The main distinguishing

feature of [75] and our result in Theorem 3.12 is that while [75] considers the case of using uniform weights $\theta_t = 1/T$ only, we are allowed to use nonuniform (in fact increasing) weights $\theta_t = t$. The faster rate of $O(1/T)$ in Theorem 3.12 is a result of this flexibility in our setup due to the weighted regret concept that lets us choose nonuniform weights. ∎

We present two other regret minimizing algorithms which exploit strong convexity and the weights to give the faster $O(1/T)$ regret bound. The first is an adaptation of Nesterov's dual averaging method [117], an alternative FOM for convex optimization. Xiao [150] analyzed dual averaging for the online and strongly convex setting, achieving an unweighted regret bound of $O(\log(T)/T)$. We show how to extend this with weights $\theta_t = t$ to achieve the $O(1/T)$ bound. Our result requires a slight modification of Assumption 3.9.

Assumption 3.15.

- A proximal setup of Section 3.3 exists for the domain $\mathcal{Z} = X$, and additionally $\max_{x\in X}\langle\nabla\omega(x_\omega), x - x\omega\rangle \le \Omega' < \infty$.
- The loss functions $f_t(x)$ for $t \in [T]$ have the property that the functions $f_t(x) - \alpha\,\omega(x)$ is convex for some $\alpha > 0$ independent of t, or equivalently

$$f_t(x) \le f_t(x') + \langle\nabla f_t(x), x - x'\rangle - \alpha V_x(x'), \quad \forall x, x' \in X,\ t \in [T].$$

- Denote $h_t(x) := f_t(x) - \alpha\omega(x)$, and note that these are convex functions. The subgradients of $h_t(x)$ are bounded, i.e., there exists $G_h < \infty$ such that $\|\nabla h_t(x)\|_* \le G$ for all $x \in X,\ t \in [T]$.

Theorem 3.16 (Strongly convex dual averaging)**.** *Suppose Assumption 3.15 holds. Denote $h_t(x) = f_t(x) - \alpha\omega(x)$. Let $\theta_t = t$ for all $t \in [T]$ and $\{x_t\}_{t\in[T]}$ be computed according to Algorithm 2, with $\gamma_t = \frac{1}{\alpha\Theta_t+1} = \frac{2}{\alpha t(t+1)+2}$ and*

$$\xi_t = \sum_{s\in[t]} \theta_s \nabla h_s(x_s) + (\alpha\Theta_t + 1)(\nabla\omega(x_t) - \nabla\omega(x_1)).$$

Then

$$\frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t f_t(x_t) - \min_{x\in X}\frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t f_t(x) \le \frac{2G_h^2}{T(T+1)}\sum_{t\in[T]}\frac{t}{\alpha(t-1)+2/t} + \frac{2(\alpha+1/T)\Omega}{T+1} + \frac{2\alpha\Omega'}{T+1}.$$

The proof of Theorem 3.16 is based on ideas from [117, 150]. We begin with a series of technical lemmas.

Lemma 3.17. *Suppose Assumption 3.15 holds. Define*

$$U_\eta(s) := \max_{x\in X}\{\langle s, x - x_1\rangle - \eta\omega(x)\}$$
$$W_\eta(s) := \max_{x\in X}\{\langle s, x - x_1\rangle - \eta\omega(x) - V_{x_1}(x)\}.$$

Then $U_\eta(s) \le W_\eta(s) + \Omega$.

Proof. Since $V_{x_1}(x) \le \Omega$ for all $x \in X$, adding $\Omega - V_{x_1}(x)$ to the objective of U_η does not decrease. □

Lemma 3.18. *Suppose Assumption 3.15 holds. For any $x \in X$ we have*

$$\begin{aligned}\sum_{t\in[T]} \theta_t f_t(x_t) - \sum_{t\in[T]} \theta_t f_t(x) &\le \sum_{t\in[T]} \theta_t \left(\langle \nabla h_t(x_t), x_t - x_1\rangle + \alpha\omega(x_t)\right) \\ &\quad + \sum_{t\in[T]} \theta_t \langle \nabla h_t(x_t), x_1 - x\rangle - \alpha\Theta_T\omega(x) \\ &\le \sum_{t\in[T]} \theta_t \left(\langle \nabla h_t(x_t), x_t - x_1\rangle + \alpha\omega(x_t)\right) + U_{\alpha\Theta_T}(-d_T).\end{aligned}$$

Proof. The first inequality follows by observing that for any $t \ge 1$, due to convexity of $h_t(x)$, we have $f_t(x_t) - f_t(x) \le \langle \nabla h_t(x_t), x_t - x\rangle + \alpha\omega(x_t) - \alpha\omega(x)$. The second inequality follows by maximizing over $x \in X$ and applying the definition of U. □

Lemma 3.19. *Suppose Assumption 3.15 holds. Let*

$$\pi_\eta(s) := \arg\max_{x\in X} \left\{\langle s, x - x_1\rangle - \eta\omega(x) - V_{x_1}(x)\right\}.$$

Then $\nabla W_\eta(s) = \pi_\eta(s) - x_1$ and is Lipschitz continuous:

$$\left\|\nabla W_\eta(s) - \nabla W_\eta(s')\right\| \le \frac{1}{\eta+1}\|s - s'\|_*.$$

Therefore, for all s, g,

$$W_\eta(s+g) \le W_\eta(s) + \langle g, \nabla W_\eta(s)\rangle + \frac{1}{2(\eta+1)}\|g\|_*^2.$$

Proof. The result follows from standard conjugacy arguments. In particular, Lipschitz continuity follows since W_η is the convex conjugate of $\eta\omega + V_{x_1}$ which is $(\eta+1)$-strongly convex with respect to $\|\cdot\|$. □

Lemma 3.20. *Denote $d_t := \sum_{s\in[t]} \theta_s \nabla h_s(x_s)$. For each $t \ge 1$,*

$$W_{\alpha\Theta_t}(-d_t) + \alpha\theta_t\omega(x_{t+1}) \le W_{\alpha\Theta_{t-1}}(-d_t).$$

Proof. We have for $t \ge 1$

$$\begin{aligned}W_{\alpha\Theta_{t-1}}(-d_t) &= \max_{x\in X} \left\{\langle d_t, x_1 - x\rangle - \alpha\Theta_{t-1}\omega(x) - V_{x_1}(x)\right\} \\ &\ge \langle d_t, x_1 - x_{t+1}\rangle - \alpha\Theta_{t-1}\omega(x_{t+1}) - V_{x_1}(x_{t+1}) \\ &= \max_{x\in X} \left\{\langle d_t, x_1 - x\rangle - \alpha\Theta_t\omega(x) - V_{x_1}(x)\right\} + \alpha\theta_t\omega(x_{t+1}) \\ &= W_{\alpha\Theta_t}(-d_t) + \alpha\theta_t\omega(x_{t+1})\end{aligned}$$

where the second equality follows from the update rule for x_{t+1}:

$$\begin{aligned}
x_{t+1} &= \mathrm{Prox}_{x_t}(\gamma_t \xi_t) \\
&= \arg\min_{x \in X} \left\{ \frac{1}{\alpha\Theta_t + 1} \langle \xi_t, x \rangle + V_{x_t}(x) \right\} \\
&= \arg\min_{x \in X} \left\{ \frac{1}{\alpha\Theta_t + 1} \langle d_t + (\alpha\Theta_t + 1)(\nabla\omega(x_t) - \nabla\omega(x_1)), x \rangle + \omega(x) - \omega(x_t) - \langle \nabla\omega(x_t), x - x_t \rangle \right\} \\
&= \arg\min_{x \in X} \left\{ \frac{1}{\alpha\Theta_t + 1} \langle d_t, x \rangle + \omega(x) - \langle \nabla\omega(x_1), x \rangle \right\} \\
&= \arg\min_{x \in X} \left\{ \langle d_t, x - x_1 \rangle + \alpha\Theta_t \omega(x) + \omega(x) - \omega(x_1) - \langle \nabla\omega(x_1), x - x_1 \rangle \right\} \\
&= \arg\max_{x \in X} \left\{ \langle d_t, x_1 - x \rangle - \alpha\Theta_t \omega(x) - V_{x_1}(x) \right\}.
\end{aligned}$$

□

We are now ready to prove Theorem 3.16.

Proof of Theorem 3.16. In light of Lemmas 3.17 and 3.18, we have

$$\begin{aligned}
\sum_{t \in [T]} \theta_t f_t(x_t) - \sum_{t \in [T]} \theta_t f_t(x) &\leq \sum_{t \in [T]} \theta_t \left(\langle \nabla h_t(x_t), x_t - x_1 \rangle + \alpha\omega(x_t) \right) + U_{\alpha\Theta_T}(-d_T) \\
&\leq \sum_{t \in [T]} \theta_t \left(\langle \nabla h_t(x_t), x_t - x_1 \rangle + \alpha\omega(x_t) \right) + W_{\alpha\Theta_T}(-d_T) + \Omega. \qquad (3.6)
\end{aligned}$$

Using Lemma 3.20 we get that for $t \geq 1$,

$$\begin{aligned}
& W_{\alpha\Theta_t}(-d_t) + \alpha\theta_t \omega(x_{t+1}) \\
& \leq W_{\alpha\Theta_{t-1}}(-d_t) \\
& = W_{\alpha\Theta_{t-1}}(-d_{t-1} - \theta_t \nabla h_t(x_t)) \\
& \leq W_{\alpha\Theta_{t-1}}(-d_{t-1}) - \theta_t \left\langle \nabla h_t(x_t), \nabla W_{\alpha\Theta_{t-1}}(-d_{t-1}) \right\rangle + \frac{\theta_t^2}{2(\alpha\Theta_{t-1} + 1)} \|\nabla h_t(x_t)\|_*^2 \\
& = W_{\alpha\Theta_{t-1}}(-d_{t-1}) - \theta_t \langle \nabla h_t(x_t), x_t - x_1 \rangle + \frac{\theta_t^2}{2(\alpha\Theta_{t-1} + 1)} \|\nabla h_t(x_t)\|_*^2.
\end{aligned}$$

Here, the first equality follows by definition, the second inequality follows from Lemma 3.19, i.e., smoothness of W, and the second equality follows since we can write $x_t = \pi_{\alpha\Theta_{t-1}}(-d_{t-1})$.

Rearranging the above inequality, we see that

$$\theta_t \langle \nabla h_t(x_t), x_t - x_1 \rangle + \theta_t \alpha\omega(x_{t+1}) \leq W_{\alpha\Theta_{t-1}}(-d_{t-1}) - W_{\alpha\Theta_t}(-d_t) + \frac{\theta_t^2}{2(\alpha\Theta_{t-1} + 1)} \|\nabla h_t(x_t)\|_*^2$$

Summing this for $t \in [T]$ and recognising the telescoping terms with $W_{\alpha\Theta_0}(0) = -V_{x_1}(x_1) = 0$, we

get

$$\begin{aligned}
&\sum_{t\in[T]} \theta_t \left(\langle \nabla h_t(x_t), x_t - x_1\rangle + \alpha\omega(x_t)\right) \\
&= \sum_{t\in[T]} \theta_t \left(\langle \nabla h_t(x_t), x_t - x_1\rangle + \alpha\omega(x_{t+1})\right) + \alpha \sum_{t\in[T]} \theta_t(\omega(x_t) - \omega(x_{t+1})) \\
&\le -W_{\alpha\Theta_T}\left(-d_T\right) + \sum_{t\in[T]} \frac{\theta_t^2}{2(\alpha\Theta_{t-1}+1)} \|\nabla h_t(x_t)\|_*^2 + \alpha \sum_{t\in[T]} \theta_t(\omega(x_t) - \omega(x_{t+1}))
\end{aligned}$$

Combining this with (3.6), we get

$$\begin{aligned}
&\sum_{t\in[T]} \theta_t f_t(x_t) - \sum_{t\in[T]} \theta_t f_t(x) \\
&\le \sum_{t\in[T]} \frac{\theta_t^2}{2(\alpha\Theta_{t-1}+1)} \|\nabla h_t(x_t)\|_*^2 + \alpha \sum_{t\in[T]} \theta_t(\omega(x_t) - \omega(x_{t+1})) + \Omega \\
&= \sum_{t\in[T]} \frac{\theta_t^2}{2(\alpha\Theta_{t-1}+1)} \|\nabla h_t(x_t)\|_*^2 + \Omega + \alpha \sum_{t=2}^{T} (\theta_t - \theta_{t-1})\omega(x_t) + \alpha\theta_1\omega(x_1) - \alpha\theta_T\omega(x_{T+1})
\end{aligned}$$

Now we will pick sequences $\theta_t = t$ for $t \ge 1$, thus $\Theta_{t-1} = (t-1)t/2$. Then

$$\begin{aligned}
&\sum_{t\in[T]} \frac{\theta_t^2}{2(\alpha\Theta_{t-1}+1)} \|\nabla h_t(x_t)\|_*^2 + \alpha \sum_{t=2}^{T} (\theta_t - \theta_{t-1})\omega(x_t) + \alpha\theta_1\omega(x_1) - \alpha\theta_T\omega(x_{T+1}) + \Omega \\
&= \sum_{t\in[T]} \frac{t^2}{2(\alpha(t-1)t/2+1)} \|\nabla h_t(x_t)\|_*^2 + \alpha \sum_{t=2}^{T} \omega(x_t) + \alpha\omega(x_1) - \alpha T\omega(x_{T+1}) + \Omega \\
&= \sum_{t\in[T]} \frac{t}{\alpha(t-1)+2/t} \|\nabla h_t(x_t)\|_*^2 + \alpha \sum_{t=2}^{T} (\omega(x_t) - \omega(x_1)) - \alpha T\left[\omega(x_{T+1}) - \omega(x_1)\right] + \Omega \\
&\le \sum_{t\in[T]} \frac{t}{\alpha(t-1)+2/t} \|\nabla h_t(x_t)\|_*^2 + (\alpha T + 1)\Omega + \alpha T\Omega'.
\end{aligned}$$

The last inequality follows since $0 \le \omega(x) - \omega(x_1) \le \Omega + \Omega'$ for all x. Using the bound $\|\nabla h_t(x_t)\|_* \le G_h$ and dividing by Θ_T gives us the result. □

The second algorithm we present is known in the OCO literature as *follow-the-leader*. The idea is to minimize the weighted sum of the functions $\{f_s\}_{s\in[t]}$ to get x_t at each time step t. The regret bound obtained in Theorem 3.21 was proved in Abernethy et al. [2, Section 2.4], so we do not give the proof here.

Theorem 3.21 (Strongly convex follow-the-leader [2, Section 2.4])**.** *Suppose that Assumption 3.9 is satisfied, except that the domain need not have a prox-setup. Choose $x_1 \in X$ arbitrarily, and*

$$x_{t+1} = \arg\min_{x\in X} \left\{ \sum_{s\in[t]} \theta_s f_s(x) \right\}.$$

Then

$$\frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t f_t(x_t) - \min_{x\in X}\frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t f_t(x) \le \frac{2G^2}{\alpha(T+1)}.$$

Remark 3.22. We now have three weighted regret minimizing algorithms that achieve $O(1/T)$ regret bound for strongly convex functions. We comment briefly on their differences. Note that the per-iteration cost of the MD and dual averaging algorithms from Theorems 3.12 and 3.16 are identical: both require implementing the prox-operator to compute x_t. On the other hand, computing x_t in follow-the-leader is potentially much harder if $f_t(x) - \alpha\omega(x)$ is non-linear; if it is linear, then follow-the-leader has the same per-iteration cost as the others. The constants in the bounds for MD and dual averaging are also slightly different. For MD, we have the gradient bound $\|\nabla f_t(x_t)\|_* \le G$, whereas for dual averaging, we have the bound $\|\nabla h_t(x_t)\|_* \le G_h$ and the bound $\langle \nabla\omega(x_\omega), x_t - x_\omega\rangle \le \Omega'$. In general, the gradient $\nabla\omega(x)$ may be unbounded over $x \in X$, even if X is compact (e.g., if $\omega(x)$ is negative entropy over simplex domain $X = \Delta$), so bounding $\|\nabla f_t(x_t)\|_* = \|\nabla h_t(x_t) + \nabla\omega(x_t)\|_*$ can be more challenging than just $\|\nabla h_t(x_t)\|_*$. Furthermore, if X is compact, then we know that $\Omega' = \max_{x\in X}\langle \nabla\omega(x_\omega), x - x_\omega\rangle$ is finite. ∎

3.6 Exploiting Smoothness via Lookahead

In this section we explore the online setting when our functions exhibit a smooth structure. We exploit this by allowing for *1-lookahead*—that is, we are allowed to a *limited* query access to our current function f_t (or Ψ_t) at time period t before we make our decision z_t. In fact, we will query the gradient only once in each period t.

As discussed in the Introduction, the 1-lookahead setting may prevent it being applicable in general online settings. In addition, if at iteration t we are given multiple query access to f_t (or Ψ_t), we can *guarantee* that the weighted regret (online SP gap) will be non-positive by directly minimizing f_t (solving for the SP of Ψ_t). However, solving a complete optimization problem at each iteration may be expensive, and hence even in the situations where we have multiple query access to f_t at iteration t, it may be preferable to use our more efficient methods to bound the weighted regret (online SP gap).

Our analysis is based on the generalized version of the Mirror Prox algorithm of Nemirovski [111] outlined in Algorithm 3.

Algorithm 3 Generalized Mirror Prox

input: ω-center z_ω, time horizon T, positive step sizes $\{\gamma_t\}_{t\in[T]}$, and sequences $\{\eta_t, \xi_t\}_{t\in[T]}$.
output: sequence $\{z_t\}_{t\in[T]}$.
$v_1 := z_\omega$
for $t = 1, \dots, T$ **do**
  $z_t = \mathrm{Prox}_{v_t}(\gamma_t\eta_t)$.
  $v_{t+1} = \mathrm{Prox}_{v_t}(\gamma_t\xi_t)$.
end for

Proposition 3.23 states a fundamental property of Mirror Prox updates which is instrumental in the derivation of our bounds. Its proof can be found in [94, Lemma 6.2 and Proposition 6.1], which we reproduce here for completeness.

Proposition 3.23. *Suppose that the sequences of vectors $\{v_t, z_t\}_{t\in[T]}$ are generated by Algorithm 3 for the given sequences $\{\eta_t, \xi_t\}_{t\in[T]}$ and step sizes $\gamma_t > 0$ for $t \in [T]$. Then for any $z \in \mathcal{Z}$ and $t \in [T]$, we have*

$$\gamma_t\langle \xi_t, z_t - z\rangle \le V_{v_t}(z) - V_{v_{t+1}}(z) + \frac{1}{2}\left(\gamma_t^2\|\xi_t - \eta_t\|_*^2 - \|z_t - v_t\|^2\right).$$

Proof. Recall that

$$\begin{aligned} z_t &= \mathrm{Prox}_{v_t}(\gamma_t\eta_t) = \arg\min_{z\in\mathcal{Z}}\{\langle \gamma_t\eta_t - \nabla\omega(v_t), z\rangle + \omega(z)\} \\ v_{t+1} &= \mathrm{Prox}_{v_t}(\gamma_t\xi_t) = \arg\min_{z\in\mathcal{Z}}\{\langle \gamma_t\xi_t - \nabla\omega(v_t), z\rangle + \omega(z)\}. \end{aligned}$$

Using the same optimality condition proved in Proposition 3.2, we have for all $z \in \mathcal{Z}$

$$\begin{aligned} \langle \gamma_t\eta_t - \nabla\omega(v_t) + \nabla\omega(z_t), z - z_t\rangle &\ge 0 \\ \langle \gamma_t\xi_t - \nabla\omega(v_t) + \nabla\omega(v_{t+1}), z - v_{t+1}\rangle &\ge 0. \end{aligned}$$

Rearranging the second inequality, we see that

$$\begin{aligned} \gamma_t\langle \xi_t, z_t - z\rangle &\le \gamma_t\langle \xi_t, z_t - v_{t+1}\rangle + \langle \nabla\omega(v_{t+1}) - \nabla\omega(v_t), z - v_{t+1}\rangle \\ &= \gamma_t\langle \xi_t, z_t - v_{t+1}\rangle + V_{v_t}(z) - V_{v_{t+1}}(z) - V_{v_t}(v_{t+1}). \end{aligned}$$

Substituting $z = v_{t+1}$ into the first inequality gives

$$\begin{aligned} \gamma_t\langle \xi_t, z_t - v_{t+1}\rangle &\le \gamma_t\langle \xi_t - \eta_t, z_t - v_{t+1}\rangle + \langle \nabla\omega(z_t) - \nabla\omega(v_t), v_{t+1} - z_t\rangle \\ &= \gamma_t\langle \xi_t - \eta_t, z_t - v_{t+1}\rangle + V_{v_t}(v_{t+1}) - V_{z_t}(v_{t+1}) - V_{v_t}(z_t). \end{aligned}$$

Combining the previous two inequalities, we have for all $z \in \mathcal{Z}$

$$\begin{aligned} \gamma_t\langle \xi_t, z_t - z\rangle &\le \gamma_t\langle \xi_t - \eta_t, z_t - v_{t+1}\rangle + V_{v_t}(z) - V_{v_{t+1}}(z) - V_{z_t}(v_{t+1}) - V_{v_t}(z_t) \\ &\le V_{v_t}(z) - V_{v_{t+1}}(z) + \gamma_t\|\xi_t - \eta_t\|_*\|z_t - v_{t+1}\| - \frac{1}{2}\|z_t - v_{t+1}\|^2 - \frac{1}{2}\|z_t - v_t\|^2, \end{aligned}$$

where the second inequality follows by Cauchy-Schwarz and strong convexity of ω. The result now follows by recognizing that for any $s \ge 0$, $\gamma_t\|\xi_t - \eta_t\|_* s - s^2/2 \le \gamma_t^2\|\xi - \eta_t\|_*^2/2$. □

We analyze Algorithm 3 under the following smoothness assumption and derive an improved rate of convergence for minimizing weighted regret.

Assumption 3.24. A proximal setup of Section 3.3 exists for the domain $\mathcal{Z} = X$. Each function $f_t(x)$ is convex in x, and there exists $L \in (0, \infty)$ such that $\|\nabla f_t(x) - \nabla f_t(v)\|_* \le L\|x - v\|$ holds for all $x, v \in X$ and all $t \in [T]$.

Theorem 3.25. *Suppose Assumption 3.24 holds. Then running Algorithm 3 with $z_t = x_t$, $\eta_t = \theta_t\nabla f_t(v_t)$, $\xi_t = \theta_t\nabla f_t(z_t)$, and step sizes $\gamma_t = \frac{1}{(L\max_{t\in T}\theta_t)}$ for all $t \in [T]$ leads to*

$$\frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t f_t(x_t) - \min_{x\in X}\frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t f_t(x) \le \Omega L\frac{\max_{t\in[T]}\theta_t}{\Theta_T}.$$

Proof. From Assumption 3.24, we have for all $t \in [T]$

$$\|\xi_t - \eta_t\|_* = \theta_t \|\nabla f_t(x_t) - \nabla f_t(v_t)\|_* \leq L\,\theta_t \|x_t - v_t\| \leq L \max_{t\in[T]} \theta_t \|x_t - v_t\|.$$

Thus, by setting $\gamma_t = \frac{1}{(L \max_{t\in[T]} \theta_t)}$, we deduce $\gamma_t^2 \|\xi_t - \eta_t\|_*^2 - \|x_t - v_t\|^2 \leq 0$ for all $t \in [T]$. Then from Proposition 3.23 we obtain for all $x \in X$ and $t \in [T]$

$$\langle \xi_t, x_t - x\rangle = \theta_t \langle \nabla f_t(x_t), x_t - x\rangle \leq \big(V_{v_t}(x) - V_{v_{t+1}}(x)\big)\, L \max_{t\in[T]} \theta_t.$$

Summing this inequality over $t \in [T]$ and using $V_{v_1}(x) \leq \Omega$, $V_{v_{T+1}}(x) \geq 0$, we get

$$\sum_{t\in[T]} \langle \xi_t, x_t - x\rangle = \sum_{t\in[T]} \theta_t \langle \nabla f_t(x_t), x_t - x\rangle \leq \Omega L \max_{t\in[T]} \theta_t.$$

The result then follows from convexity of f_t and using the subgradient inequality $\langle \nabla f_t(x_t), x_t - x\rangle \geq f_t(x_t) - f_t(x)$. □

A similar result holds for the online SP gap under the following analogous smoothness assumption.

Assumption 3.26. A proximal setup of Section 3.3 exists for the domain $\mathcal{Z} = X \times Y$, and we denote $z = [x; y]$. Each function $\Psi_t(x, y)$ is convex in x and concave in y. Denoting $F_t(z) = [\nabla_x \Psi_t(x, y); -\nabla_y \Psi_t(x, y)]$, there exists $L \in (0, \infty)$ such that for all $v, z \in \mathcal{Z}$ and all $t \in [T]$, we have

$$\|F_t(z) - F_t(v)\|_* \leq L\|z - v\|.$$

Remark 3.27. A sufficient condition for the Lipschitz continuity of monotone gradient operators F_t of Assumption 3.26 is Lipschitz continuity of their partial subgradients. For brevity, we omit the proof of this; see [94, 112] for further details. ∎

Theorem 3.28. *Suppose Assumption 3.26 holds. Then running Algorithm 3 with $z_t = [x_t; y_t]$, $\eta_t = \theta_t F_t(v_t)$, $\xi_t = \theta_t F_t(z_t)$, and step sizes $\gamma_t = \frac{1}{(L \max_{t\in T} \theta_t)}$ for all $t \in [T]$ leads to*

$$\max_{y\in Y} \sum_{t\in[T]} \theta_t \Psi_t(x_t, y) - \min_{x\in X} \sum_{t\in[T]} \theta_t \Psi_t(x, y_t) \leq \Omega L \max_{t\in[T]} \theta_t.$$

Proof. Following the outline of the proof of Theorem 3.25, we obtain

$$\sum_{t\in[T]} \langle \xi_t, z_t - z\rangle = \sum_{t\in[T]} \theta_t \langle F_t(z_t), z_t - z\rangle \leq \Omega L \max_{t\in[T]} \theta_t$$

for all $z = [x; y] \in X \times Y$. As in the proof of Theorem 3.7, using the convex-concave structure of the functions Ψ_t, we arrive at

$$\theta_t \langle F_t(z_t), z_t - z\rangle \geq \theta_t \Psi_t(x_t, y) - \theta_t \Psi_t(x, y_t),$$

which establishes the result. □

Remark 3.29. When the weights θ_t are set to be either uniform $\theta_t = 1$ or nonuniform $\theta_t = t$ from Theorem 3.12, we have $\frac{1}{\Theta_T} \max_{t\in[T]} \theta_t = O(1/T)$, and thus we achieve better weighted regret/online SP gap bounds of $O(1/T)$ in Theorems 3.25 and Theorem 3.28 than the $O(1/\sqrt{T})$ bounds of Theorems 3.4 and Theorem 3.7. ∎

There is a fundamental distinction between Algorithms 2 and 3 in terms of their anticipatory/non-anticipatory behavior. This distinction between anticipatory/non-anticipatory behavior is important in the context of using these algorithms for coupled optimization problems. We discuss this next.

Remark 3.30. When Algorithm 3 is utilized in Theorems 3.25 and 3.28, at step t, in order to compute the decision $z_t = \text{Prox}_{v_t}(\gamma_t \eta_t)$, where $v_t \in \mathcal{Z}$ is a point computed in the previous step, we utilize the knowledge of the current function f_t or Ψ_t because $\eta_t = \theta_t \nabla f_t(v_t)$ or $\eta_t = \theta_t F_t(v_t)$. Therefore, Algorithm 3 is categorized as *1-lookahead* or *anticipatory*. This is in contrast to the non-anticipatory nature of Algorithm 2 analyzed in Theorems 3.4, 3.12, and 3.7, where computing $z_t = \text{Prox}_{z_{t-1}}(\gamma_{t-1}\xi_{t-1})$ only required knowledge of the previous step $t-1$ because ξ_{t-1} was determined based on only $\nabla f_{t-1}(z_{t-1})$ or $F_{t-1}(z_{t-1})$. ∎

Remark 3.31. Rakhlin and Sridharan [123, 124] also explore OCO with anticipatory decisions through the lens of *predictable sequences* $\{M_t\}_{t\in[T]}$. More precisely, they also examine how regret bounds are affected when the player is allowed to utilize side information M_t before choosing x_t at time t. They propose the Optimistic Mirror Descent (OpMD) algorithm, which is a special case of Algorithm 3 for $\eta_t = M_t$, $\xi_t = \nabla f_t(z_t)$ and $\theta_t = 1/T$, and are able to recover the offline Mirror Prox algorithm from [112] for smooth offline convex optimization and smooth offline SP problems. In fact, our results in Theorem 3.25 and Theorem 3.28 can be derived from [124, Lemma 1] by specifying the predictable sequences $M_t = \theta_t \nabla f_t(v_t)$ and $M_t = \theta_t F_t(v_t)$ respectively. Here, we allow the player to have access *only to gradient information* of f_t or Ψ_t at time t. Because the focus of [123, 124] was different, the observation that the OpMD algorithm can obtain faster $O(1/T)$ convergence rates in the 1-lookahead setting was not made before. ∎

Remark 3.32. It is known that the OCO regret bounds with general smooth loss functions have a lower bound complexity of at least $O(1/\sqrt{T})$ (this holds even for the case of linear loss functions [1, Theorem 5]). This is in contrast to the faster rate of $O(1/T)$ established in Theorem 3.25. The lookahead nature of our analysis of Algorithm 3 discussed in Remark 3.30 plays a crucial role for achieving the speedup established in Theorem 3.25. ∎

3.7 Application to the Primal-Dual Framework of Chapter 2

Although we believe that the developments in this chapter may be of independent interest in OCO, our primary motivation was to build the tools necessary to bound the relevant terms in the primal-dual framework of Chapter 2. Specifically, for RO, recall that the first aim is to bound $\widehat{\epsilon}\left(\{x_t, y_t, u_t, \theta_t\}_{t\in[T]}\right)$ from (2.14) by choosing x_t, y_t appropriately. Notice that this term is simply an online SP gap term. We can choose two regret minimizing algorithms (amongst the ones described in Theorems 3.4, 3.12, 3.16, 3.21, 3.25) to compute the primal sequence $\{x_t\}_{t\geq 1}$ and the dual sequence $\{x_t\}_{t\geq 1}$. Note that we cannot choose Theorem 3.25 to compute both x_t and y_t, since computing x_t requires knowing $\Psi(x, y_t; u_t)$ and computing y_t requires knowing $\Psi(x_t, y; u_t)$, so one must be computed without knowledge of the other. Alternatively, we can utilize Theorem 3.7 or 3.28 to compute both x_t, y_t simultaneously. Furthermore, recall that the second aim is to bound $\sup_{u\in U} \epsilon^\circ\left(\{x_t, u_t, \theta_t\}_{t\in[T]}; u\right)$ from (2.17) by choosing the $u_t \in U$ appropriately. These are weighted

regret terms, so we can utilize any of the algorithms in Theorems 3.4, 3.12, 3.16, 3.21, 3.25. Note again, however, we must be careful when we choose the 1-lookahead algorithm in Theorem 3.25; see Remark 2.9.

For JEO, the aim is to bound $\widehat{\epsilon}\big(\{x_t, y_t, u_t, \theta_t\}_{t\in[T]}\big)$ from (2.30). Again, we can choose two regret minimizing algorithms (amongst the ones described in Theorems 3.4, 3.12, 3.16, 3.21, 3.25) to compute the primal sequence $\{x_t\}_{t\geq 1}$ and the dual sequence $\{x_t\}_{t\geq 1}$. Again, we cannot choose Theorem 3.25 for both. Alternatively, we can utilize Theorem 3.7 or 3.28 to compute both x_t, y_t simultaneously.

Chapter 4

Second-Order Cone Reformulation for the Trust Region Subproblem with Applications to Robust Quadratic Programming

4.1 Introduction

In this chapter, we study the classical *trust-region subproblem* (TRS) and its polynomial-time solvable variants given by

$$\mathrm{Opt}_h := \min_{y\in\mathbb{R}^n} \left\{ h(y) := y^\top Q y + 2\, g^\top y : \quad \begin{array}{rcl} \|y\| & \leq & 1 \\ Ay - b & \in & \mathcal{K} \end{array} \right\}, \tag{4.1}$$

where $\|y\|$ denotes the Euclidean norm of y, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and $\mathcal{K} \subseteq \mathbb{R}^m$ is a closed convex cone. Throughout the chapter, we assume that the minimum eigenvalue of Q is negative, that is, $\lambda_Q := \lambda_{\min}(Q) < 0$ and the domain of the problem is nonempty. Problem (4.1) is equivalent to the *classical TRS* when there are no additional conic constraints, i.e., $A = I_n$, $b = 0$, and $\mathcal{K} = \mathbb{R}^n$. That is, the classical TRS is given by

$$\min_{y\in\mathbb{R}^n} \left\{ h(y) := y^\top Q y + 2 g^\top y : \ \|y\| \leq 1 \right\}. \tag{4.2}$$

The classical TRS is an essential ingredient of trust-region methods that are commonly used to solve continuous nonconvex optimization problems (see [51, 119, 122] and references therein). In each iteration of a trust-region method, a quadratic approximation of the objective function is built and then optimized over a ball, called trust region, (or intersection of a ball with linear or conic constraints originating from the original problem) to find the new search point. The TRS and its variants are also encountered in the context of robust optimization under matrix norm or polyhedral uncertainty (see [22, 29] and references therein), nonlinear optimization problems with discrete variables [41, 43], least-squares problems [155], constrained eigenvalue problems [66], and more. One particular application we wish to highlight is robust convex quadratic programming (QP). Robust convex QPs with ellipsoidal uncertainty are known to have close connections with the TRS (see [15]). The function $f(x, u)$ underlying a robust convex quadratic constraint $\max_{u\in U} f(x, u) \leq 0$ is convex

and quadratic in both the decision variable x and the uncertainty u, making the maximization over U exactly a TRS instance.

4.1.1 Related Literature

As stated above, the optimization problem in (4.1) is nonlinear and nonconvex when $\lambda_Q < 0$. Nevertheless, it is well-known that the semidefinite programming (SDP) relaxation for the classical TRS is exact. Furthermore, the classical TRS and a number of its variants can be solved in polynomial time via SDP-based techniques [126, 64] or using specialized nonlinear algorithms, e.g., [69, 106].

Several variants of the classical TRS that enforce additional constraints on the trust region have been proposed. Among these the most commonly studied is the case when $\mathcal{K}$ is taken to be a nonnegative orthant, i.e., the unit ball is intersected with additional linear constraints modeled via the polyhedral set $\{y \in \mathbb{R}^n : \ Ay - b \in \mathcal{K}\}$. The TRS with additional linear inequalities arises in nonlinear programming and robust optimization (see Burer [42], Jeyakumar and Li [90] and references therein) and is studied in [32, 42, 43, 45, 90, 145, 153] under a variety of assumptions. Specifically, Burer and Anstreicher [43], Sturm and Zhang [145] give a tight semidefinite formulation when there is a single linear constraint $a^\top y \leq b$ based on an additional constraint derived from second-order cone (SOC) based reformulation linearization technique (SOC-RLT). This approach was extended to two linear constraints in [43, 153] and the tightness of the SDP relaxation is shown when the linear constraints are parallel. More recently, Burer and Yang [45] give a tight SDP relaxation with additional SOC-RLT constraints for an arbitrary number of linear constraints, under the condition that these additional linear inequalities do not intersect on the interior of the unit ball. We refer the readers to Burer [42] for a recent survey and related references for the results on tight SDP relaxations associated with these variants. Following a different approach, Bienstock and Michalka [32] show that TRS with linear inequality constraints is polynomial-time solvable under the milder condition that the number of faces of the linear constraints intersecting with the unit ball is polynomially bounded.

TRS with additional conic constraints originate when the trust-region algorithm is applied to conic constrained optimization problems with nonconvex objective. The most notable example in this context is the well-known Celis–Dennis–Tapia (CDT) problem [48] where a nonconvex quadratic is minimized over the intersection of two-ellipsoids. See also Ben-Tal and den Hertog [15] for several applications of the TRS with additional conic quadratic constraints arising in the context of robust quadratic programming. Recently, Jeyakumar and Li [90] proved convexity of the joint numerical range, exactness of the SDP relaxation, and strong Lagrangian duality for the TRS with additional linear and SOC constraints. A key tool in their analysis is to recast the TRS as a convex quadratic minimization problem under a dimensionality condition.

Hollow constraints defined by a single ellipsoid [21, 31, 122, 144, 153], several ellipsoids [32, 151], or arbitrary quadratics constraints [30] have also attracted some attention in the literature. These approaches are once again either lifted SOC-based or SDP-based convexification schemes or customized algorithms. We discuss these further in Section 4.3.3.

While the SDP reformulations of the classical TRS and its variants can be solved using interior-point methods in polynomial time [6, 118], this approach is not practical because the worst-case complexity of these methods for solving SDPs is a relatively large polynomial and there exist faster methods. That said, the classical TRS is closely connected to eigenvalue problems. In the specific case of classical TRS where the objective is convex, i.e., when Q is positive semidefinite, this problem becomes simply the minimization of a smooth convex function over the Euclidean ball, and thus

it can be solved efficiently via iterative first-order methods (FOMs) such as Nesterov's accelerated gradient descent algorithm [114]. Moreover, in the nonconvex case with $\lambda_Q < 0$, when the problem is purely quadratic, i.e., when $g = 0$ as well, the classical TRS reduces to finding the minimum eigenvalue of Q. This can be approximated efficiently via the Lanczos method [68, Chapter 10.1] in practice. When $g \neq 0$, even though the classical TRS is no longer equivalent to an eigenvalue problem and these methods cannot be applied directly, this observation has led to the development of efficient, matrix-free algorithms that are based solely on matrix-vector products. The dual-based algorithms of [106, 126, 138], the generalized Lanczos trust-region method of [69], and the recent developments of [4, 59, 60, 70, 76, 130] are examples of such iterative algorithms. More recently, for TRS with a single additional linear constraint, the papers [132, 133, 134] explore strong Lagrangian duality, and derive numerically efficient algorithms from this. In most cases, these algorithms for classical TRS and its variants are presented together with their convergence proofs. Nevertheless, to the best of our knowledge, the theoretical runtime evaluation of these algorithms lacks formal guarantees with the exception of recent work [76] (done in a probabilistic fashion). In addition, in most of these iterative methods, numerical difficulties are reported in the so-called "hard case" [106], when the linear component vector g is nearly orthogonal to the eigenspace of the smallest eigenvalue of Q. In many cases, the lack of provable worst-case convergence bounds for the classical TRS is attributed to the hard case. As a result, most research on specific algorithms for the classical TRS thus far focuses on addressing this issue.

Recently, Hazan and Koren [76] suggested a linear-time algorithm for approximately solving the classical TRS within a given tolerance ϵ on the objective value. Their approach relies on an efficient, linear-time solver for a specific SDP relaxation of a feasibility version of the classical TRS and reduces the classical TRS into a series of eigenvalue computations. Specifically, they exploit the special structure of the dual problem, a one-dimensional problem for which bisection techniques can be applied, to avoid using interior-point solvers. Each dual step of their algorithm requires a single approximate maximal eigenvalue computation which takes $O\left(N\frac{\sqrt{\Gamma}}{\sqrt{\epsilon}}\log\left(\frac{n}{\delta}\log\left(\Gamma/\epsilon\right)\right)\right)$ time to achieve an ϵ-accurate estimate with probability at least $1-\delta/\log\left(\Gamma/\epsilon\right)$, where N is the number of nonzero entries in Q, $\Gamma := \max\left\{2(\|Q\|+\|g\|),1\right\}$, and $\|Q\|$ stands for the spectral norm of the matrix Q, i.e., the maximum absolute eigenvalue. Their overall algorithm converges in $O\left(\log\left(\frac{\Gamma}{\epsilon}\right)\right)$ iterations. Then a primal solution is recovered by solving a small linear program formed by the dual iterates. Finally, they provide an efficient and accurate rounding procedure for converting the SDP solution into a feasible solution to the classical TRS. Consequently, their approach does not require the use of interior-point SDP solvers and bypasses the difficulties noted for the hard case of the classical TRS. The overall complexity (elementary arithmetic operations) of their approach is

$$O\left(N\frac{\sqrt{\Gamma}\log\left(\Gamma/\epsilon\right)}{\sqrt{\epsilon}}\log\left(\frac{n}{\delta}\log\left(\frac{\Gamma}{\epsilon}\right)\right)\right).$$

Thus, the runtime of their approach is linear in the number of nonzero entries of the input and it can exploit data sparsity.

These algorithmic developments have been complemented with research on convex hull characterization of sets associated with the TRS. In this respect, [42] presents a nice summary of such results given for the lifted SDP representations. The epigraph of TRS is closely related to convex hulls of sets defined as the intersection of convex and nonconvex quadratics. Such sets cover two-term disjunctions applied to an SOC or its cross-sections arising in the context of mixed integer conic programming or reverse convex constraints based on ellipsoids, and thus have been

studied under a variety of assumptions (see Burer and Kılınç-Karzan [44] and references therein). In particular, nonconvex sets obtained from the intersection of a second-order-cone representable (SOCr) cone and a nonconvex cone defined by a single homogeneous quadratic, and possibly an affine hyperplane were studied in [44]. For such sets, under several easy-to-verify conditions, [44] suggests a simple, computable convex relaxation where the nonconvex cone is replaced by an additional SOCr cone, and identifies several stronger conditions guaranteeing the tightness of these relaxations, in terms of giving the associated closed conic hulls and closed convex hulls of these sets. These conditions have been further verified in many specific cases, and it was shown in [44] that the classical TRS can then be solved via the optimization of two SOC-based programs. Similar convex hull descriptions of a single SOC or its cross-section intersected with a general nonconvex quadratic are also studied recently in [105] under different assumptions.

4.1.2 Contributions

In this chapter, as opposed to the previous specialized algorithms or approaches that work in a lifted space, e.g., SDP-based relaxations, we follow an SOC-based approach in the original space of variables to solve the classical TRS and its variants with conic constraints (4.1) or hollows. That is, under easy-to-verify conditions, we derive tight SOC-based convex reformulations and convex hull characterizations of sets associated with the TRS with additional conic constraints (4.1). Our contributions can be summarized as follows.

- In Section 4.2, we study an SOC-based convex relaxation of (4.1) in the original space of variables obtained by simply replacing the nonconvex objective function $h(y)$ in (4.1) with the convex objective $f(y) := y^\top (Q - \lambda_Q I_n)\, y + 2\, g^\top y + \lambda_Q$. We prove tightness of this relaxation under an easily checkable structural condition on the additional conic constraints $Ay - b \in \mathcal{K}$ (see Theorem 4.5). For classical TRS our convex relaxation is immediately tight without any condition. In the case of nontrivial conic constraints $Ay - b \in \mathcal{K}$ in (4.1), the conditions ensuring tightness of our convex relaxation can be somewhat stringent. We discuss these issues and relation of our condition to the existing ones from the literature in Section 4.2.2.

- Due to the fact that our convex relaxation/reformulation works in the original space of variables and thus preserves the domain, it is immediately amenable to work with existing iterative FOMs; we discuss the associated complexity results in Section 4.2.3. In particular, our convex relaxation/reformulation can be built via a single minimum eigenvalue computation. In the case of classical TRS, it can then be solved by minimizing a smooth convex quadratic over the unit ball via Nesterov's accelerated gradient descent algorithm [114]. Thus, with probability $1 - \delta$, our approach solves the classical TRS to accuracy ϵ in running time

$$O\left(N\left(\frac{\sqrt{\|Q\|}}{\sqrt{\epsilon}}\log\left(\frac{n}{\delta}\right) + \frac{\sqrt{\|Q\|}}{\sqrt{\epsilon}}\right)\right).$$

- In Section 4.3, we study exact and explicit SOC-based convex hull results for the epigraph of the TRS given by

$$X := \left\{ \begin{bmatrix} y \\ t \end{bmatrix} \in \mathbb{R}^{n+1} : \begin{array}{rcl} \|y\| & \leq & 1 \\ Ay - b & \in & \mathcal{K} \\ h(y) & \leq & t \end{array} \right\}.$$

  In Theorem 4.33, under a slightly stronger condition, we provide an explicit characterization of the convex hull of X in the space of original variables.

We also examine the inclusion of additional hollow constraints $y \in \mathcal{R} = \mathbb{R}^n \setminus \mathcal{P}$ to the TRS in Section 4.3.3. In particular, these developments immediately lead to convex reformulations for several variants of TRS, including interval-bounded TRS (see [21, 31, 122, 144, 153]), and thus have algorithmic implications.

- In Section 4.4, we show how the convex reformulation of the TRS can be used in the context of robust convex quadratic programs (QPs) with ellipsoidal uncertainty. In particular, the convex reformulation allows us to recast robust quadratic constraint as a convex-concave function. This then allows us to apply the primal-dual framework of Chapter 2 and the associated OCO algorithms from Chapter 3 to improve upon theoretical guarantees of an existing approach by Ben-Tal et al. [25]. We also conducted a numerical study, outlined in Section 4.4.1, that shows the scalability of our primal-dual approach compared to other iterative RO approaches.

From a convex reformulation perspective, the papers [64, 90, 44, 15, 102] are closely related to our approach. To handle the hard case in classical TRS, Fortin and Wolkowicz [64] discuss a shift of the matrix Q, which results in the same SOC-based convex reformulation as ours. Nevertheless, Fortin and Wolkowicz [64] solves the resulting problem using a modification of the SDP-based Rendl–Wolkowicz algorithm [126]. Their approach requires a case-by-case analysis to handle the hard case and lacks formal convergence guarantees. In contrast to such an approach, we propose using Nesterov's algorithm [114], which is not only oblivious to the hard case and thus does not require a case-by-case analysis, but also provides formal convergence guarantees. Jeyakumar and Li [90] study TRS with additional linear and conic-quadratic constraints. They obtain a convex reformulation via a similar shift in the Q matrix under a certain dimensionality condition on the additional constraints. We show that the conditions of Jeyakumar and Li [90] imply our structural condition and we provide an example where our condition is satisfied but the ones of [90] are not. Burer and Kılınç-Karzan [44] also give a scheme to solve the classical TRS via SOC programming. The scheme suggested in [44] is in a lifted space with one additional variable and requires solving two related SOC optimization problems. In contrast, our convex reformulation is in the space of original variables and requires solving only a single minimization problem. Ben-Tal and den Hertog [15] study a different SOC-based convex reformulation in a lifted space of the TRS and its variants under a simultaneously diagonalizable assumption. However, this relaxation requires a full eigenvalue decomposition of the matrix Q as opposed to our relaxation which only needs a maximum eigenvalue computation. Based on the same convex reformulation as Ben-Tal and den Hertog [15], Locatelli [102] studies the TRS with additional linear constraints under a structural condition on the constraints derived from a KKT system. We show that in the case of additional linear constraints, our geometric condition is equivalent to the structural condition of Locatelli [102] (see Lemma 4.13). To the best of our knowledge, the KKT-based derivations of conditions of Locatelli [102] are not extended to the conic case, yet our condition handles additional conic constraints generalizing the one of Locatelli [102] and highlights the features of underlying geometry.

On the algorithmic side, our transformation of the TRS (4.1) is mainly based on the minimum eigenvalue of Q, which can be computed to accuracy $\epsilon > 0$ with probability $1 - \delta$ in $O\left(N\sqrt{\|Q\|}\log(n/\delta)/\sqrt{\epsilon}\right)$ arithmetic operations using the Lanczos method (see [99, Section 4] and [76, Section 5]), where N is the number of nonzero entries in Q. Due to the fact that $f(y)$ is a convex quadratic function, our convex relaxation/reformulation for (4.1) can simply be cast as a conic optimization problem. Specifically, when there are no additional constraints, this exact

convex reformulation becomes minimizing a smooth convex function over the Euclidean ball, and thus it is readily amenable to efficient FOMs. For this class of convex problems, given a desired accuracy of ϵ, a classical FOM, Nesterov's accelerated gradient descent algorithm [114], involves only elementary operations such as addition, multiplication, and matrix-vector product computations and achieves the optimal iteration complexity of $O\left(\sqrt{\|Q\|}/\sqrt{\epsilon}\right)$. Note when the problem is convex (when Q is positive semidefinite), the same complexity guarantees can be obtained by applying Nesterov's accelerated gradient descent [114] to the problem. Thus, our approach can be seen as an analog of the latter algorithm to the general nonconvex case. This is the first-time that such an observation is made that the classical TRS problem can be solved by a single minimum eigenvalue computation and Nesterov's accelerated gradient descent [114]. Moreover, our analysis highlights the connection between the TRS and eigenvalue problems, and in fact demonstrates that, up to constant factors, the complexity of solving the classical TRS is no worse than solving a minimum eigenvalue problem. This was empirically observed in [126, Section 5] and our analysis provides a theoretical justification for it. In a similar spirit, recently Adachi et al. [4] suggested an approach for the standard TRS by first solving a single generalized eigenvalue problem and then doing some conjugate gradient steps. Note, however, that generalized eigenvalue problems are computationally more demanding than the minimum eigenvalue computations, and that Adachi et al. [4] does not provide an explicit convergence rate analysis.

Convexification-based approaches such as ours and [15, 21, 76, 90, 102] work directly with convex formulations and provide a uniform treatment of the problem and thus bypass the so-called "hard case." Moreover, the resulting convex formulations are then amenable to iterative FOMs from convex optimization literature which only require matrix-vector product type operations. To the best of our knowledge, iterative algorithms for SDP-based relaxations of the TRS have not been studied in the literature with the exception of Hazan and Koren [76]. As compared to the approach of Hazan and Koren [76], we believe our approach is straightforward, easy to implement, and achieves a slightly better convergence guarantee in the worst case. In particular, our approach directly solves the TRS, as opposed to only solving a feasibility version of the TRS; thus we save an extra logarithmic factor. While Hazan and Koren [76] relies on repeatedly calling a minimum eigenvalue, our approach, as well as that of Jeyakumar and Li [90], works with an SOC-based reformulation of the problem in the original space and requires only a single minimum eigenvalue computation. The convex reformulations given by Ben-Tal and Teboulle [21] or the one studied in Ben-Tal and den Hertog [15] and Locatelli [102] requires a full eigenvalue decomposition which is more expensive, i.e., $O(n^3)$ time. Moreover, these reformulations from [15, 21, 102] involve additional variables and constraints, and thus FOMs applied to these entail more complicated and expensive projection operations.

Our convex hull results on the epigraph of the TRS are inspired by the recent work of Burer and Kılınç-Karzan [44] on convex hulls of general quadratic cones. While the SOC-based convex hull results in [44] are applicable to many problems, including the epigraph set associated with the classical TRS, we present a much more direct analysis specialized for TRS. There are two main benefits of our approach. First, the approach outlined in [44] for solving classical TRS requires the assumption that the optimal value is nonpositive. While this is not an issue for the classical TRS since its optimal value is always negative under the assumption of $\lambda_Q < 0$, with the existence of additional constraints, this may no longer be true for (4.1). In contrast, our direct analysis does not rely on any nonpositivity assumptions of the objective value, and hence we are able to extend our results to include additional conic constraints. Second, our direct analysis of the TRS allows

us to bypass verifying several conditions from [44] and to work directly with a single structural condition on additional conic constraints which is always satisfied in the case of the classical TRS.

Several papers [11, 12, 90] exploit convexity results on the joint numerical range of quadratic mappings to explore strong duality properties of the TRS and its variants. These convexity results are based on Yakubovich's $\mathcal{S}$-lemma [65] and Dines [54], see also the survey by Pólik and Terlaky [120] for a more detailed discussion. While these results as well as ours both analyze sets associated with the TRS, the actual sets in question are quite different. In the context of the TRS, the joint numerical range is a set of the form

$$\left\{[h(y);\ \|y\|^2;\ Ay-b]:\ y\in\mathbb{R}^n\right\}\subseteq\mathbb{R}^{m+2}.$$

Under certain conditions, this set is shown to be convex. In contrast, we study the epigraphical set X, which is nonconvex if $h(y)$ is nonconvex, and we give its convex hull description in the original space of variables.

Notation . We use MATLAB notation to denote vectors and matrices. Given a matrix, $A\in\mathbb{R}^{m\times n}$, we let $\text{Null}(A)$ and $\text{Range}(A)$ denote its nullspace and range. Furthermore, we denote the minimum eigenvalue of a symmetric matrix Q as $\lambda_Q:=\lambda_{\min}(Q)$ and we let I_n be the $n\times n$ identity matrix. For a given symmetric matrix Q, the notation $Q\succeq 0$ ($Q\succ 0$) corresponds to the requirement that Q is positive semidefinite (positive definite). Given a vector $\xi\in\mathbb{R}^n$, $\text{Diag}(\xi)$ corresponds to an $n\times n$ diagonal matrix with its diagonal equal to ξ. For a set $S\subseteq\mathbb{R}^n$, we define $\text{int}(S),\text{relint}(S),\text{bd}(S),\text{Ext}(S),\text{Rec}(S)$, $\text{conv}(S)$, $\overline{\text{conv}}(S)$, $\text{cone}(S)$, and $\overline{\text{cone}}(S)$ to be the interior, relative interior, boundary, set of extreme points, recession cone, convex hull, closed convex hull, conic hull, and closed conic hull of S, respectively. For a cone $\mathcal{K}\subseteq\mathbb{R}^n$, we denote its dual cone by $\mathcal{K}^*$.

4.2 Tight Low-Complexity Convex Reformulation of the TRS

In this section, we first present an exact SOC-based convex reformulation for the classical TRS and extend this reformulation to the TRS with additional conic constraints (4.1) under an appropriate condition. We then compare and relate our condition to handle conic constraints to other conditions studied in the literature. Finally, we explore algorithmic aspects of solving our SOC-based reformulation.

4.2.1 Convex Reformulation

We start with the following simple observation, which we present without proof.

Observation 4.1. *Let $\mathcal{C}\subset\mathbb{R}^n$ be some bounded domain, and let $h:\mathcal{C}\to\mathbb{R}$ be a (possibly nonconvex) function such that h has no local minimum on* $\text{int}(\mathcal{C})$. *Then any optimal solution y^* of the program*

$$\min_y\{h(y):\ y\in\mathcal{C}\}$$

must be on $\text{bd}(\mathcal{C})$.

We next observe that when our domain $\mathcal{C}$ is defined by (possibly nonconvex) constraints $c_j(y)\le 0$, we can obtain relaxations of the nonconvex program in Observation 4.1 by simply aggregating these constraints with appropriate weights.

Lemma 4.2. *Let $C\subseteq\mathbb{R}^n$ be a given set, and let $c_j(y):C\to\mathbb{R}$ for $j=1,\dots,m$ be given functions. Suppose $h(y)$ is a given function and $f_j(y)$ are functions on the domain $\mathcal{C}:=\{y:\ c_j(y)\le 0,\ \forall j=$*

$1, \ldots, m\} \cap C$ *such that* $f_j(y) = h(y) - \alpha_j c_j(y)$ *for some* $\alpha_j \leq 0$. *Let* $F(y) := \max_{j=1,\ldots,m} f_j(y)$. *Then*

$$\mathrm{Opt}_h := \min_y \{h(y) : \ y \in \mathcal{C}\} \geq \min_y \{F(y) : \ y \in \mathcal{C}\} =: \mathrm{Opt}_f \,.$$

Moreover, $\mathrm{Opt}_h = \mathrm{Opt}_f$ *if and only if there exists an optimal solution* y^* *to the problem on the right-hand side satisfying* $\alpha_j c_j(y^*) = 0$ *for some* $j \in \{1, \ldots, m\}$.

Proof. First, we note that for any $y \in \mathcal{C}$, we have $\alpha_j c_j(y) \geq 0$ since $\alpha_j \leq 0$, and thus for all $j \in \{1, \ldots, m\}$, $f_j(y) = h(y) - \alpha_j \, c_j(y) \leq h(y)$. This establishes $\mathrm{Opt}_h \geq \mathrm{Opt}_f$.

Let y^* be an optimal solution to $\min_y \{F(y) : \ y \in \mathcal{C}\}$ for which $\alpha_j c_j(y^*) = 0$ for some j. Then we have $F(y^*) = f_j(y^*) = h(y^*)$, which implies that y^* is also optimal to Opt_h. Now consider the case where every optimal solution $y^* \in \arg\min_y \{F(y) : \ y \in \mathcal{C}\}$ satisfies $\alpha_j c_j(y^*) > 0$ for all j. Note that for any $y \in \mathcal{C}$ satisfying $\alpha_j c_j(y) > 0$ for all j, we have $F(y) < h(y)$. Thus, for such optimal solutions y^*, we have $F(y^*) < h(y^*)$, and for any other nonoptimal solution $y \in \mathcal{C}$, we have $F(y^*) < F(y) \leq h(y)$, which implies $\mathrm{Opt}_f < \mathrm{Opt}_h$. □

Let us now turn our attention back to the TRS (4.1). Henceforth, we define $h(y) := y^\top Q y + 2g^\top y$ to be our nonconvex quadratic objective function, where Q is some symmetric matrix with $\lambda_Q < 0$. It is easy to see that on any bounded domain $\mathcal{C}$, $h(y)$ has no local minimum on $\mathrm{int}(\mathcal{C})$. Hence, Observation 4.1 points out the important role of the boundary of the domain $\{y : \ \|y\| \leq 1, \ Ay - b \in \mathcal{K}\}$ to the TRS (4.1).

A possible convex relaxation for (4.1) suggested by Lemma 4.2 is that we embed the conic constraints $Ay - b \in \mathcal{K}$ into the ground set C and aggregate the constraint $\|y\| \leq 1$ with weight $\alpha = \lambda_Q$ to obtain the objective function

$$f(y) := h(y) + \lambda_Q(1 - \|y\|^2) = y^\top (Q - \lambda_Q I_n) y + 2g^\top y + \lambda_Q. \tag{4.3}$$

Note $Q - \lambda_Q I_n \succeq 0$, and thus the function $f(y)$ is convex, and clearly is also an underestimator of $h(y)$, hence minimizing $f(y)$ over our domain is still a convex relaxation. Lemma 4.2 then gives us a precise characterization for when the convex relaxation using $f(y)$ is tight.

Corollary 4.3. *Suppose* $\lambda_Q < 0$. *Consider the convex relaxation for problem* (4.1) *given by*

$$\mathrm{Opt}_f = \min_y \left\{ f(y) : \begin{array}{rcl} \|y\| & \leq & 1 \\ Ay - b & \in & \mathcal{K} \end{array} \right\}, \tag{4.4}$$

where $f(y)$ *is defined in* (4.3). *This convex relaxation is tight if and only if there exists an optimal solution* y^* *to* (4.4) *such that* $\|y^*\| = 1$.

Because $Q - \lambda_Q I_n$ is not full rank, when g is not orthogonal to $\mathrm{Null}(Q - \lambda_Q I_n)$, it is easy to see that the function $f(y)$ has no local minima on the interior of our domain. Then by Observation 4.1, the optimal solutions to (4.4) lie on $\mathrm{bd}(\{y : \|y\| \leq 1, \ Ay - b \in \mathcal{K}\})$. When g is orthogonal to $\mathrm{Null}(Q - \lambda_Q I_n)$, then we can add $d \in \mathrm{Null}(Q - \lambda_Q I_n)$ to any point y without changing the objective $f(y + d)$, hence there will always exist an optimal solution of (4.4) on $\mathrm{bd}(\{y : \|y\| \leq 1, \ Ay - b \in \mathcal{K}\})$. However, $f(y) = h(y)$ if and only if $\|y\| = 1$, but $f(y)$ may not be equal to $h(y)$ on all of $\mathrm{bd}(\{y : \|y\| \leq 1, \ Ay - b \in \mathcal{K}\})$. More precisely, we will have $f(y) < h(y)$ for $y \in \mathrm{bd}(\{y : \|y\| \leq 1, \ Ay - b \in \mathcal{K}\}) \cap \{y : \|y\| < 1\}$, so if all minima of $f(y)$ lie on this set, the convex relaxation (4.4) will not be tight. Therefore, we next state a sufficient condition that ensures that there is always an optimal solution of (4.4) on the boundary of the unit ball.

Condition 4.4. There exists a vector $d \neq 0$ such that $Qd = \lambda_Q d$, $Ad \in \mathcal{K}$ and $g^\top d \leq 0$.

Theorem 4.5. *Suppose that $\lambda_Q < 0$ and that Condition 4.4 holds for the TRS given in (4.1). Then the convex relaxation given by (4.4) is tight.*

Proof. Let y^* be an optimum solution for (4.4). If $\|y^*\| = 1$, then from Corollary 4.3, the result follows immediately. Hence, we assume $\|y^*\| < 1$.

Let $d \neq 0$ be the vector from Condition 4.4, thus $Qd = \lambda_Q d$, $Ad \in \mathcal{K}$, and $g^\top d \leq 0$. Then for any $\epsilon > 0$, $A(y^* + \epsilon d) - b = (Ay^* - b) + \epsilon Ad \in \mathcal{K}$ because $\mathcal{K}$ is a convex cone and $Ad \in \mathcal{K}$ by assumption. Because $\|y^*\| < 1$, we may increase ϵ until $\|y^* + \epsilon d\| = 1$ and the vector $y^* + \epsilon d$ is still feasible. Note $(Q - \lambda_Q I_n)d = 0$, so for any $\epsilon > 0$,

$$f(y^* + \epsilon d) = f(y^*) + 2(g^\top d)\epsilon \leq f(y^*).$$

If $g^\top d < 0$, this violates optimality of y^* since $\epsilon > 0$, thus $g^\top d = 0$. Then the vector $y^* + \epsilon d$ is an alternative optimum solution to (4.4) satisfying $\|y^* + \epsilon d\| = 1$. Hence, the tightness of the relaxation (4.4) follows from Corollary 4.3. □

Remark 4.6. From the definition of λ_Q, Condition 4.4 is immediately satisfied for the classical TRS (4.2) without additional conic constraints, i.e., when $A = I_n$, $b = 0$, and $\mathcal{K} = \mathbb{R}^n$. ∎

Consequently, in the case of classical TRS, Remark 4.6 implies the following specialization of Theorem 4.5.

Theorem 4.7. *When $\lambda_Q < 0$, a tight convex relaxation of classical TRS (4.2) is given by*

$$\mathrm{Opt}_f = \min_y \left\{ f(y) := y^\top (Q - \lambda_Q I_n) y + 2g^\top y + \lambda_Q : \ \|y\| \leq 1 \right\}. \tag{4.5}$$

Remark 4.8. In order to handle a particular "hard case" of classical TRS, Fortin and Wolkowicz [64] introduce and analyze the convex reformulation (4.5) (see [64, Lemma 2.3] and [64, Section 7]). We believe (4.5) can be of more use than stated in [64]. In particular, by reanalyzing (4.2), we are able to both

(i) improve on the previously best-known theoretical convergence rate guarantees for solving the classical TRS (see Remark 4.22 in Section 4.2.3), and

(ii) establish the tightness of the convex reformulation (4.4) for TRS with conic constraints under appropriate conditions (see Theorem 4.5) and also for TRS with hollow constraints covering interval-bounded TRS (see [21, 31, 122, 144, 153]), under a condition well-studied in the literature (see Corollary 4.40 and Theorem 4.39).

∎

4.2.2 Discussion of Condition 4.4 and Related Conditions from the Literature

For TRS with conic constraints (4.1), Condition 4.4 is related to and generalizes many other conditions examined in the literature.

A result similar to Theorem 4.5 was implicitly proven by Jeyakumar and Li [90] under a dimensionality condition for the case of linear and conic quadratic constraints. We state the linear version of their condition below; the conic quadratic one is very similar.

Condition 4.9. Consider the case of nonnegative orthant, i.e., $\mathcal{K} = \mathbb{R}^m_+$. Suppose that the system of linear inequalities, i.e., the constraint $Ay - b \in \mathcal{K}$ satisfies the requirement that $\dim(\text{Null}(Q - \lambda_Q I_n)) \geq n - \dim(\text{Null}(A)) + 1$.

Lemma 4.10. *Condition 4.4 generalizes the dimensionality condition of Jeyakumar and Li [90], i.e., Condition 4.9, stated for linear and conic quadratic constraints.*

Proof. Suppose Condition 4.9 holds. Then

$$\dim(\text{Null}(A)) + \dim(\text{Null}(Q - \lambda_Q I_n)) \geq n + 1;$$

thus, there must exist $d \neq 0$ which is in the intersection $\text{Null}(A) \cap \text{Null}(Q - \lambda_Q I_n)$. That is, $Qd = \lambda_Q d$ and $Ad = 0 \in \mathbb{R}^m_+ = \mathcal{K}$. If $g^\top d \leq 0$, then Condition 4.4 holds with the vector d. If $g^\top d > 0$, then Condition 4.4 holds with the vector $d' = -d$. □

Jeyakumar and Li [90] demonstrates that Condition 4.9 is satisfied in a number of cases related to the robust least squares and robust SOC programming problems. As a consequence of Lemma 4.10, our Condition 4.4 is satisfied in these cases as well. That said, Condition 4.4 is more general than Condition 4.9 as demonstrated by the following example.

Example 4.11. For the problem data given by

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad g = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix}, \quad b = \frac{1}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mathcal{K} = \mathbb{R}^2_+,$$

Condition 4.4 is satisfied with $d = [0; -1]$, but Condition 4.9 is not. ∎

Ben-Tal and den Hertog [15] and Locatelli [102] study a different SOC-based convex relaxation of (4.1) given in a lifted space when Q is a diagonal matrix and the additional constraints are linear, i.e., $\mathcal{K} = \mathbb{R}^m_+$. Let $Q = \text{Diag}(\{q_1, \ldots, q_n\})$; then this reformulation is given by

$$\min_{y,z} \left\{ \sum_{i=1}^n q_i z_i + 2g^\top y : \begin{array}{l} y_i^2 \leq z_i, \quad i = 1, \ldots, n \\ \sum_{i=1}^n z_i \leq 1 \\ Ay \geq b \end{array} \right\}. \tag{4.6}$$

It was established in [15] that for the classical TRS this convex reformulation is tight. Tightness of this relaxation for the TRS with additional linear constraints is studied in [102] under the following condition.

Condition 4.12. Denote $Q = \text{Diag}(\{q_1, \ldots, q_n\})$ and $J = \{j : q_j = \lambda_Q\}$. Also, define A_J to be the matrix composed of columns of A which correspond to the indices in J, and define g_J analogously. For all $\epsilon > 0$, there exists h_ϵ with $\|h_\epsilon\| \leq \epsilon$ such that $\{\mu \geq 0 : \ A_J^\top \mu + g_J + h_\epsilon = 0\} = \emptyset$.

Lemma 4.13. *When Q is diagonal and $\mathcal{K} = \mathbb{R}^m_+$, Conditions 4.4 and 4.12 are equivalent.*

Proof. It is shown in [102, Proposition 3.3] that Condition 4.12 is equivalent to the program $\max_{\hat{y} \in \mathbb{R}^{|J|}} \{-g_J^\top \hat{y} : A_J \hat{y} \leq 0\}$ being unbounded above or having multiple optima. In the former case, there must exist an extreme ray $\hat{d} \neq 0$ for which $g_J^\top \hat{d} < 0$ and $A_J \hat{d} \leq 0$. Setting d to be the vector consisting of $\hat{d}$ in the J entries and 0 otherwise gives us $Qd = \lambda_Q d$, $Ad \leq 0$, and $g^\top d < 0$, which satisfies Condition 4.4. In the latter case, we know that the zero vector is always an optimal

solution with objective value 0, so having multiple optima means there exists $\hat{d} \neq 0$ such that $A_J \hat{d} \leq 0$ and $g_J^\top \hat{d} = 0$. Then a similar argument follows to show that Condition 4.4 holds.

Conversely, if Condition 4.4 holds, because Q is diagonal, the vector d given must have zeros everywhere except for entries in J. If $g^\top d < 0$, then the program above is unbounded, but if $g^\top d = 0$, then the program above has multiple optima since we can add d to any optimal solution. Thus, Condition 4.12 holds. □

Remark 4.14. Condition 4.4 is equivalent to the conic program

$$\min_d \left\{ g^\top d : \ (Q - \lambda_Q I_n) d = 0, \ Ad \in \mathcal{K} \right\}$$

being unbounded below or having multiple optimal solutions. This follows from an extension of the proof of Lemma 4.13 to the conic case. ∎

Remark 4.15. Despite Conditions 4.4 and 4.12 being equivalent when $\mathcal{K} = \mathbb{R}^m_+$, there are two major distinctions between our convex reformulation (4.4) and the one from [15, 102]. First, in order to diagonalize the matrix Q in TRS and hence form the convex reformulation of [15, 102], one needs to perform a full eigenvalue decomposition, which takes approximately $O(n^3)$ time and is more expensive than computing only the minimum eigenvalue (approximately $O(n^2)$ time) that is needed by our convex reformulation. Second, our convex reformulation (4.4) works in the original space of variables and thus preserves the nice structure of the domain, yet (4.6) introduces new variables $z_1, \ldots, z_n$. Preserving the nice structure of the original convex domain becomes important when FOMs are applied to a convex reformulation of TRS. We discuss this issue in the case of classical TRS in Section 4.2.3. ∎

Remark 4.16. In contrast to the results given in [90] and [102], Theorem 4.5 holds for general conic constraints when Condition 4.4 holds. Note that such general conic constraints can represent a variety of convex restrictions, and in particular, they may include positive semidefiniteness requirements. ∎

We next present an example to illustrate that when Condition 4.4 is violated, we may not be able to give the exact convex reformulation. Moreover, a slight modification of this example demonstrates further that Condition 4.4 is not necessary for giving the exact convex reformulation.

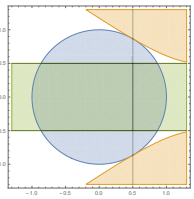
Example 4.17. Suppose we are given the problem data:

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix}, \quad g = \begin{bmatrix} -3 \\ 0 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}, \quad b = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mathcal{K} = \mathbb{R}^2_+.$$

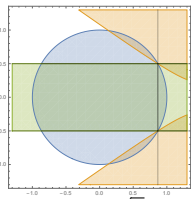
Then Condition 4.4 is violated. To see this, note that any d satisfying $Qd = \lambda_Q d$ is of the form $d = [0; d_2]$. However, $Ad = [d_2; -d_2]$, so if $d_2 \neq 0$, $Ad \notin \mathcal{K} = \mathbb{R}^2_+$. For this problem data, $h(y) = y_1^2 - 2y_2^2 - 3y_1$ and $f(y) = 3y_1^2 - 3y_1 - 2$. It is easy to compute the minimizers of $f(y)$ over the unit ball to be the line $y_1 = 1/2$, with value $-11/4$. The constraints $Ay - b \in \mathcal{K}$ are equivalent to $-1/2 \leq y_2 \leq 1/2$.

Figure 4.1 shows that the minimizers of $h(y)$ over just the unit ball $\|y\| \leq 1$ lie on the boundary at $y = [1/2; \pm\sqrt{3}/2]$. Due to the linear constraints $-1/2 \leq y_2 \leq 1/2$, these points are cut off from the feasible region. As a result, any minimizer of $f(y)$ (i.e., the line $y_1 = 1/2$) inside the feasible region has norm strictly less than 1. Then by Corollary 4.3, the relaxation (4.4) is not tight.

Figure 4.1: Contour plots of $h(y)$ over the feasible set.



(a) $h(y) \leq -11/4 = \text{Opt}_f$.



(b) $h(y) \leq (1 - 6\sqrt{3})/2 = \text{Opt}_h$

Finally, note that if we were to change our linear constraints to $-0.9 \leq y_2 \leq 0.9$, then our relaxation would be tight, while Condition 4.4 would still not be satisfied. However, for both cases in this example, the SDP relaxation of [145, 153, 43] strengthened with additional SOC-RLT inequalities is tight. ∎

A variant of Condition 4.4 is instrumental in giving exact convex hull characterization of the sets associated with the TRS (4.1). We discuss these further in Section 4.3.

4.2.3 Complexity of Solving Our Convex Reformulations

In this section, we explore the complexity of solving our convex relaxation/reformulation of TRS via FOMs. Our convex relaxation/reformulation of TRS (4.4) and its variants have the same domain as their original nonconvex counterparts (4.1) and thus are solvable via interior point methods and standard software as long as the cone $\mathcal{K}$ has an explicit barrier function. However, because the standard polynomial-time interior point methods have expensive iterations in terms of their dependence on the problem dimension, here we mainly focus on FOMs with cheap iterations. We next discuss the complexity of solving our convex reformulation of the classical TRS given by (4.2) via Nesterov's accelerated gradient descent algorithm [114], an optimal FOM for this class of problems. Once again, the main distinction between solving (4.4) as opposed to (4.5) via FOMs lies in how the projection onto the respective domain is handled. That is, whenever efficient projection onto the original domain is present, our discussion below will remain applicable to the conic case (4.4) as well.

The reformulation (4.5) of classical TRS (4.2) (or the convex relaxation (4.4) of TRS (4.1)) is an SOC program (convex program) and can easily be built whenever λ_Q is available to us. Moreover, computing λ_Q, the minimum eigenvalue of Q, itself is a TRS with no linear term because

$$\lambda_Q = \min_y \left\{ y^\top Q y : \ \|y\| \leq 1 \right\}.$$

There exist many efficient algorithms for computing the minimum eigenvalue of a symmetric matrix Q. One such algorithm that is effective for large sparse matrices is the Lanczos method [68, Chapter 10]. Implemented with a random start, this method enjoys the following probabilistic convergence guarantee (see [99, Section 4] and [76, Section 5]): with probability at least $1-\delta$, the Lanczos method correctly estimates λ_Q to within ϵ-accuracy in $O\left(\sqrt{\|Q\|}\log(n/\delta)/\sqrt{\epsilon}\right)$ iterations. Furthermore, each iteration requires only matrix-vector products, and hence takes $O(N)$ time, where N is the number of nonzero entries in Q. Consequently, with probability at least $1-\delta$, the randomized Lanczos method estimates λ_Q to within ϵ-accuracy in time $O\left(N\sqrt{\|Q\|}\log(n/\delta)/\sqrt{\epsilon}\right)$.

Given λ_Q, problem (4.5) is simply minimizing a smooth convex quadratic function $f(y)$ with smoothness parameter $2(\lambda_{\max}(Q) - \lambda_Q) \leq 4\|Q\|$ over the unit ball. Note that due to the nature of this transformation, $f(y)$ is not strongly convex, and hence may have multiple optima. This problem (4.5) can be efficiently solved by using Nesterov's accelerated gradient descent algorithm [114], which obtains an ϵ-accurate solution in $O\left(\sqrt{\|Q\|}/\sqrt{\epsilon}\right)$ iterations. This is the optimal rate for FOMs for solving this class of problems. The major computational burden in each iteration in these FOMs is the evaluation of the gradient of $f(y)$, which involves simply a matrix-vector product, and hence each iteration costs $O(N)$ time. The only other main operation in each iteration of Nesterov's algorithm applied to this problem is the projection onto the Euclidean ball, and this can be done in $O(n)$ time. Consequently, Nesterov's algorithm [114] applied to the optimization problem in our convex reformulation (4.5) of the classical TRS runs in time $O\left(N\sqrt{\|Q\|}/\sqrt{\epsilon}\right)$.

Thus, taking into account the complexity of computing λ_Q to build our convex reformulation (4.5) and using Nesterov's algorithm [114], we establish the following upper bound on the worst case number of elementary operations needed.

Theorem 4.18. *With probability $1 - \delta$, a solution $\bar{y}$ to the classical TRS (4.2) satisfying $h(\bar{y}) - h(y) \leq \epsilon$ for all y in the unit ball can be found in time*

$$O\left(N\left(\frac{\sqrt{\|Q\|}}{\sqrt{\epsilon}}\log\left(\frac{n}{\delta}\right) + \frac{\sqrt{\|Q\|}}{\sqrt{\epsilon}}\right)\right) = O\left(N\frac{\sqrt{\|Q\|}}{\sqrt{\epsilon}}\log\left(\frac{n}{\delta}\right)\right) \tag{4.7}$$

using randomized Lanczos method to compute λ_Q and Nesterov's algorithm [114].

Remark 4.19. This discussion shows that the classical TRS decomposes into two special TRS problems: one without a linear term, i.e., $g = 0$, making it a pure minimum eigenvalue problem, and the other with a convex quadratic objective function. This once again highlights the connection between the TRS and eigenvalue problems, and in fact demonstrates that, up to constant factors, the complexity of solving the classical TRS is no worse than solving a minimum eigenvalue problem because the complexity in Theorem 4.18 is essentially determined by the complexity of computing minimum eigenvalue of a matrix. Rendl and Wolkowicz [126, Section 5] have empirically observed this connection between complexity of solving classical TRS and computing the minimum eigenvalue; our analysis complements their study with a theoretical justification. ∎

Remark 4.20. Using a deterministic algorithm to compute λ_Q eliminates the probabilistic component in Theorem 4.18 at the expense of a slightly worse dependence on ϵ and n in the iteration complexity.

Unlike other methods [64, 106, 126], our proposed method need not differentiate between the easy case and the hard case. ∎

Remark 4.21. In practice, we will not be able to form the objective $f(y)$ exactly, since λ_Q will be computed only approximately. Let us consider an underestimate $\gamma \approx \lambda_Q$ and working with the convex objective $f_\gamma(y) = y^\top(Q - \gamma I_n)y + 2g^\top y$. We show in Appendix A.1 how such an underestimate γ of λ_Q can be obtained using the Lanzcos method. In Appendix A.1, we show that by using $f_\gamma(y)$ instead of $f(y)$, the error we incur is linearly dependent on the error of estimating λ_Q with γ, which for our purposes is $O(\epsilon)$. ∎

Remark 4.22. Let us compare our bound (4.7) to the running time from [76]. The approach of [76, Theorem 1] requires

$$O\left(N\frac{\sqrt{\Gamma}\log\left(\Gamma/\epsilon\right)}{\sqrt{\epsilon}}\log\left(\frac{n}{\delta}\log\left(\frac{\Gamma}{\epsilon}\right)\right)\right)$$

elementary operations to obtain an ϵ-accurate solution for (4.2) with probability $1-\delta$, where $\Gamma=\max\left\{2(\|Q\|+\|g\|),1\right\}$. By using the convex reformulation (4.5) as opposed to the method of [76], we remove (at least) a factor of $\log\left(\Gamma/\epsilon\right)$ and the dependence on $\|g\|$.

Our method is simpler to implement than the method of [76] as well because it decomposes the TRS into two well-studied problems as discussed in Remark 4.19. In contrast, since [76] relies on solving the dual SDP, at the end of its iterations, it requires additional operations to obtain the primal solution from the dual one, and then a rounding procedure to find the solution in the original space. ∎

4.3 Convexification of the Epigraph of TRS

In this section, we study the convex hull of the epigraph of TRS. In general, a tight convex relaxation for a nonconvex optimization problem does not necessarily imply that the epigraph of the convex relaxation is giving the exact convex hull of the epigraph of the nonconvex optimization problem. However, in the particular case of TRS with additional conic constraints, i.e., problem (4.1), under a slightly more stringent variant of Condition 4.4, we will establish that not only our convex relaxation given by (4.4) is tight but also its epigraph exactly characterizes the convex hull of the epigraph of underlying TRS (4.1) (see Corollary 4.34).

By defining a new variable x_{n+2} (where the variable x_{n+1} is reserved for later homogenization), and moving the nonconvex function from the objective to the constraints, we can equivalently recast (4.1) as minimizing x_{n+2} over its epigraph

$$\text{Opt}_h=\min_{y,x_{n+2}}\left\{x_{n+2}:\begin{array}{rcl}\|y\| & \leq & 1\\ Ay-b & \in & \mathcal{K}\\ h(y)=y^\top Qy+2g^\top y & \leq & x_{n+2}\end{array}\right\}. \tag{4.8}$$

Since the objective x_{n+2} is linear, optimizing over the epigraph is equivalent to optimizing over its convex hull. We define the associated epigraph as

$$X:=\left\{x=[y;1;x_{n+2}]\in\mathbb{R}^{n+2}:\begin{array}{rcl}\|y\| & \leq & 1\\ Ay-b & \in & \mathcal{K}\\ y^\top Qy+2g^\top y & \leq & x_{n+2}\end{array}\right\}. \tag{4.9}$$

Our convex hull characterizations are also SOC based. That is, as in Section 4.2.1, we focus mainly on the quadratic parts of the TRS (4.1), namely the nonconvex quadratic $y^\top Qy+2g^\top y$ and the unit ball constraint $\|y\|\leq 1$ and provide the convexification of this set X via a single new SOC constraint.

Our approach is a refinement of the one from Burer and Kılınç-Karzan [44]. We first summarize the approach of [44] in Section 4.3.1 and then give our direct characterization in Section 4.3.2. As opposed to general SOCs and their cross-sections examined in Section 4.3.1, we present a direct study of $\overline{\text{conv}}(X)$ in Section 4.3.2 that utilizes the fact that our domain in the context of TRS is a subset of an ellipsoid. Consequently, our analysis in Section 4.3.2 eliminates the need to verify several conditions from [44] completely and allows possibilities to handle additional conic constraints under appropriate assumptions. Finally, in Section 4.3.3, we extend our analysis to cover additional hollow constraints in the domain.

4.3.1 Summary and Discussion of Results from Burer and Kılınç-Karzan [44]

We start with a number of relevant definitions and conditions and then present the main result of [44].

A cone $\mathcal{F}^+ \subseteq \mathbb{R}^k$ is said to be *second-order-cone representable* (or *SOCr*) if there exists a matrix $0 \neq R \in \mathbb{R}^{k\times(k-1)}$ and a vector $r \in \mathbb{R}^k$ such that the nonzero columns of R are linearly independent, $r \notin \text{Range}(R)$, and

$$\mathcal{F}^+ = \left\{x : \ \|R^\top x\| \leq r^\top x\right\}. \tag{4.10}$$

Given an SOCr cone $\mathcal{F}^+$, the cone $\mathcal{F}^- := -\mathcal{F}^+$ is also SOCr. Based on $\mathcal{F}^+$ from (4.10), we define $W := RR^\top - rr^\top$ and consider the union $\mathcal{F}^+ \cup (\mathcal{F}^-) = \mathcal{F}^+ \cup (-\mathcal{F}^+) =: \mathcal{F}$. Note that $\mathcal{F}$ corresponds to a nonconvex cone defined by the homogeneous quadratic inequality $x^\top W x \leq 0$:

$$\mathcal{F} := \mathcal{F}^+ \cup (\mathcal{F}^-) = \left\{x : \ \|R^\top x\|^2 \leq (r^\top x)^2\right\} = \left\{x : \ x^\top W x \leq 0\right\}.$$

We define $\text{apex}(\mathcal{F}^+) = \text{apex}(\mathcal{F}^-) = \text{apex}(\mathcal{F}) = \{x : R^\top x = 0, \ r^\top x = 0\}$. Any matrix W of the form $W = RR^\top - rr^\top$ as described above has exactly one negative eigenvalue, and given $\mathcal{F}$, we can recover $\mathcal{F}^+$ by performing an eigenvalue decomposition of W, see [44, Propositions 1 and 3].

Given matrices $W_0, W_1 \in \mathbb{R}^{k\times k}$ and a vector $h \in \mathbb{R}^k$, we let $W_t = (1-t)W_0 + tW_1$ for $t \in [0,1]$, and define the sets

$$\begin{aligned} &\mathcal{F}_0 := \{x : x^\top W_0 x \leq 0\}, \quad \mathcal{F}_1 := \{x : x^\top W_1 x \leq 0\}, \quad \mathcal{F}_t = \{x : x^\top W_t x \leq 0\}, \\ &H^0 := \{x : h^\top x = 0\}, \quad H^1 := \{x : h^\top x = 1\}. \end{aligned}$$

Burer and Kılınç-Karzan [44] provide a general scheme to build an SOC-based convex relaxation of $\mathcal{F}_0^+ \cap \mathcal{F}_1$ and establish that under appropriate conditions their relaxations are exactly describing $\overline{\text{cone}}(\mathcal{F}_0^+ \cap \mathcal{F}_1)$ and $\overline{\text{conv}}(\mathcal{F}_0^+ \cap \mathcal{F}_1 \cap H^1)$. Their analysis relies on the following conditions.

Condition 4.23. W_0 has at least one positive eigenvalue and exactly one negative eigenvalue.

Condition 4.24. There exists $\bar{x}$ such that $\bar{x}^\top W_0 \bar{x} < 0$ and $\bar{x}^\top W_1 \bar{x} < 0$.

Condition 4.25. Either (i) W_0 is nonsingular, (ii) W_0 is singular and W_1 is positive definite on $\text{Null}(W_0)$, or (iii) W_0 is singular and W_1 is negative definite on $\text{Null}(W_0)$.

Conditions 4.23–4.25 ensure the existence of a maximal $s \in [0,1]$ such that W_t has a single negative eigenvalue for all $t \in [0,s]$, W_t is invertible for all $t \in (0,s)$, and W_s is singular—that is, $\text{Null}(W_s)$ is nontrivial whenever $s < 1$. Then, for all W_t with $t \in [0,s]$, the set $\mathcal{F}_t^+$ is well defined by computing an eigenvalue decomposition of W_t. We also need the following conditions on the value of s.

Condition 4.26. When $s < 1$, $\text{apex}(\mathcal{F}_s^+) \cap \text{int}(\mathcal{F}_1) \neq \emptyset$.

Condition 4.27. When $s < 1$, $\text{apex}(\mathcal{F}_s^+) \cap \text{int}(\mathcal{F}_1) \cap H^0 \neq \emptyset$ or $\mathcal{F}_0^+ \cap \mathcal{F}_s^+ \cap H^0 \subseteq \mathcal{F}_1$.

Conditions 4.23–4.27 are all that is needed to state the main result of [44]. Here, we state [44, Theorem 1] for completeness.

Theorem 4.28 ([44, Theorem 1])**.** *Suppose Conditions 4.23–4.25 are satisfied. Let s be the maximal $s \in [0,1]$ such that $W_t := (1-t)W_0 + tW_1$ has a single negative eigenvalue for all $t \in [0,s]$. Then, $\overline{\text{cone}}(\mathcal{F}_0^+ \cap \mathcal{F}_1) \subseteq \mathcal{F}_0^+ \cap \mathcal{F}_s^+$, and equality holds under Condition 4.26. Moreover, Conditions 4.23–4.27 imply $\mathcal{F}_0^+ \cap \mathcal{F}_s^+ \cap H^1 = \overline{\text{conv}}(\mathcal{F}_0^+ \cap \mathcal{F}_1 \cap H^1)$.*

These convexification results were also applied to the classical TRS (4.2) in [44]. In particular, it is shown in [44, Section 7.2] that the classical TRS (4.2) can be reformulated in the form of

$$\mathrm{Opt}_h = \min_{\tilde{y},x_{n+2}} \left\{ -x_{n+2}^2 : \begin{array}{rcl} \|\tilde{y}\| & \leq & 1 \\ \tilde{y}^\top \tilde{Q}\tilde{y} + 2\tilde{g}^\top \tilde{y} & \leq & -x_{n+2}^2 \end{array} \right\}, \tag{4.11}$$

where $\tilde{g} = [g; 0]$ and $\tilde{Q} := \begin{bmatrix} Q & 0 \\ 0 & \lambda_Q \end{bmatrix}$ is defined to ensure $\lambda_{\min}(\tilde{Q}) = \lambda_Q$ and the multiplicity of λ_Q in $\tilde{Q}$ is at least two. Note that here $\tilde{y} = [y; \tilde{y}_{n+1}] \in \mathbb{R}^{n+1}$. Then [44] suggests to solve (4.11) in two stages after the nonconvex domain in (4.11) is replaced by its convex hull. Specifically, [44] defines a new variable $\tilde{x} = [\tilde{y}; x_{n+1}; x_{n+2}]$ and the matrices

$$\tilde{W}_0 = \begin{bmatrix} I_{n+1} & 0 & 0 \\ 0^\top & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \tilde{W}_1 = \begin{bmatrix} \tilde{Q} & \tilde{g} & 0 \\ \tilde{g}^\top & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.12}$$

which then leads to

$$\begin{aligned} Y &:= \left\{ [\tilde{y}; 1; x_{n+2}] \in \mathbb{R}^{n+3} : \begin{array}{rcl} \|\tilde{y}\| & \leq & 1 \\ \tilde{y}^\top \tilde{Q}\tilde{y} + 2\tilde{g}^\top \tilde{y} & \leq & -x_{n+2}^2 \end{array} \right\} \\ &= \left\{ \tilde{x} = [\tilde{y}; x_{n+1}; x_{n+2}] \in \mathbb{R}^{n+3} : \begin{array}{rcl} \tilde{x}^\top \tilde{W}_0 \tilde{x} & \leq & 0 \\ \tilde{x}^\top \tilde{W}_1 \tilde{x} & \leq & 0 \\ x_{n+1} & = & 1 \end{array} \right\} \\ &= \mathcal{F}_0^+ \cap \mathcal{F}_1 \cap \left\{ \tilde{x} \in \mathbb{R}^{n+3} : \ x_{n+1} = 1 \right\}, \end{aligned} \tag{4.13}$$

where $\mathcal{F}_0 = \{\tilde{x} : \tilde{x}^\top \tilde{W}_0 \tilde{x} \leq 0\}$ and $\mathcal{F}_1 = \{\tilde{x} : \tilde{x}^\top \tilde{W}_1 \tilde{x} \leq 0\}$. Then the conditions of Theorem 4.28 are satisfied, and we deduce that there exists some $s \in (0, 1)$ ensuring

$$\overline{\mathrm{conv}}(\mathcal{F}_0^+ \cap \mathcal{F}_1 \cap \{\tilde{x} : x_{n+1} = 1\}) = \mathcal{F}_0^+ \cap \mathcal{F}_s^+ \cap \{\tilde{x} : x_{n+1} = 1\}. \tag{4.14}$$

While the precise value of s is not given in [44], one can show that in fact $s = \frac{1}{1-\lambda_Q}$. We present the verification of conditions of Theorem 4.28 for matrices in (4.12) and the derivation for this s value in Appendix A.2.

Remark 4.29. The reformulation (4.11) of classical TRS (4.2) implicitly requires that $\mathrm{Opt}_h \leq 0$ because of the constraint $\tilde{y}^\top \tilde{Q}\tilde{y} + 2\tilde{g}^\top \tilde{y} \leq -x_{n+2}^2 \leq 0$. For the classical TRS (4.2) with no additional constraints, this is not an additional limitation because $\tilde{y} = 0$ will always be a feasible solution with objective value 0 and thus the optimum solution will have a nonpositive objective value. However, this becomes a limitation when we want to extend such arguments for the TRS (4.1) with additional conic constraints $Ay - b \in \mathcal{K}$ because Opt_h may no longer be nonpositive. ∎

4.3.2 Direct Convexification of the Epigraph of TRS

Due to Remark 4.29, we instead choose to study the epigraph of TRS (4.1) as in (4.9), which allows for positive objective values in (4.8) and avoids the additional lifting of the problem $Q \to \tilde{Q}$. To this end, we define the matrices

$$W_0 = \begin{bmatrix} I_n & 0 & 0 \\ 0^\top & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad W_1 = \begin{bmatrix} Q & g & 0 \\ g^\top & 0 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & 0 \end{bmatrix}, \tag{4.15}$$

and the corresponding sets

$$\begin{aligned}
\mathcal{F}_0^+ &= \left\{x = [y; x_{n+1}; x_{n+2}] \in \mathbb{R}^{n+2} : \ \|y\|^2 \le x_{n+1}^2, \ x_{n+1} \ge 0\right\} \\
&= \left\{x \in \mathbb{R}^{n+2} : \ x^\top W_0 x \le 0, \ x_{n+1} \ge 0\right\}, \\
\mathcal{F}_1 &= \left\{x \in \mathbb{R}^{n+2} : \ y^\top Q y + 2g^\top y\, x_{n+1} \le x_{n+1} x_{n+2}\right\} = \left\{x : \ x^\top W_1 x \le 0\right\}, \\
\widehat{\mathcal{K}} &= \left\{x \in \mathbb{R}^{n+2} : Ay - bx_{n+1} \in \mathcal{K}\right\}, \\
H^1 &= \left\{x \in \mathbb{R}^{n+2} : \ x_{n+1} = 1\right\}.
\end{aligned} \tag{4.16}$$

Note that $\lambda_Q < 0$, and thus $\mathcal{F}_1$ is not convex. With these definitions, the epigraph X from (4.9) can be written as

$$X = \mathcal{F}_0^+ \cap \mathcal{F}_1 \cap \widehat{\mathcal{K}} \cap H^1.$$

It is mentioned in [44] that the matrices (4.15) do not satisfy the necessary conditions to apply Theorem 4.28 directly. In particular, Condition 4.25 is violated for the choice of matrices (4.15). As a result, [44, Section 7.2] reformulates the classical TRS with matrices (4.12) instead. In contrast, we next show that in the special case of the classical TRS, via a direct analysis, finding the convex hull through linear aggregation of constraints will still carry through for the matrices in (4.15). This then indicates that while Condition 4.25 is sufficient, it is not necessary to obtain the convex hull result. In fact, we show that the value of $s = \frac{1}{1-\lambda_Q}$ that works for the matrices (4.12) will also work for our matrices (4.15). More precisely, for $s = \frac{1}{1-\lambda_Q}$, we define

$$\mathcal{F}_s = \left\{x : x^\top W_s x \le 0\right\} = \left\{x : y^\top (Q - \lambda_Q I_n) y + 2g^\top y x_{n+1} + \lambda_Q x_{n+1}^2 \le x_{n+1} x_{n+2}\right\}, \tag{4.17}$$

and prove that $\overline{\text{conv}}(X) = \text{conv}(X) = \text{conv}(\mathcal{F}_0^+ \cap \mathcal{F}_1 \cap \widehat{\mathcal{K}} \cap H^1) = \mathcal{F}_0^+ \cap \mathcal{F}_s \cap \widehat{\mathcal{K}} \cap H^1$ directly under the following condition that handles additional conic constraints.

Condition 4.30. There exists a vector $d \neq 0$ such that $Qd = \lambda_Q d$, $Ad \in \mathcal{K}$, and $-Ad \in \mathcal{K}$.

Note that when $\mathcal{K}$ is pointed and A is full rank, Condition 4.30 assumes $Ad = 0$.

Remark 4.31. Condition 4.30 implies Condition 4.4. To see this, suppose $d \neq 0$ satisfies Condition 4.30. Then if $g^\top d \le 0$, d satisfies Condition 4.4 also. Otherwise, $-d$ will satisfy Condition 4.4. We demonstrate that Condition 4.4 does not imply Condition 4.30 in Example 4.36.

Furthermore, Condition 4.30 holds whenever Condition 4.9 of [90] is satisfied because Condition 4.9 implies that there exists d such that $Qd = \lambda_Q d$ and $Ad = 0$ and since $\mathcal{K}$ is a closed convex cone, $\pm Ad = 0 \in \mathcal{K}$ as well. ∎

One of the ingredients of our convex hull result is given in the next lemma.

Lemma 4.32. *Let $\mathcal{F}_s$ be defined as in (4.17). Then the cone $\mathcal{F}_s \cap \{x : x_{n+1} > 0\}$ is convex, and the set $\mathcal{F}_s \cap H^1$ where H^1 is as defined in (4.16) is SOC representable.*

Proof. Let $x = [y; x_{n+1}; x_{n+2}] \in \mathbb{R}^{n+2}$. Note that by definition we have

$$\begin{aligned}
&\mathcal{F}_s \cap \{x : x_{n+1} > 0\} \\
&= \left\{x : \ y^\top (Q - \lambda_Q I_n) y + 2g^\top y x_{n+1} + \lambda_Q x_{n+1}^2 \leq x_{n+1} x_{n+2}, \ x_{n+1} > 0\right\} \\
&= \left\{x : \ y^\top (Q - \lambda_Q I_n) y \leq x_{n+1}(x_{n+2} - 2g^\top y - \lambda_Q x_{n+1}), \ x_{n+1} > 0\right\} \\
&= \left\{x : \begin{array}{l} y^\top (Q - \lambda_Q I_n) y \leq x_{n+1}(x_{n+2} - 2g^\top y - \lambda_Q x_{n+1}), \\ x_{n+1} > 0, \quad x_{n+2} - 2g^\top y - \lambda_Q x_{n+1} \geq 0 \end{array}\right\},
\end{aligned}$$

where the last equation follows because $Q - \lambda_Q I_n \succeq 0$, we have $y^\top (Q - \lambda_Q I_n) y \geq 0$ for all y and then $x_{n+1} > 0$ implies $x_{n+2} - 2g^\top y - \lambda_Q x_{n+1} \geq 0$. As a result, $x_{n+1} + x_{n+2} - 2g^\top y - \lambda_Q x_{n+1} \geq 0$ holds for all $x \in \mathcal{F}_s \cap \{x : x_{n+1} > 0\}$. In addition, from these derivations, we immediately deduce that the set $\mathcal{F}_s \cap \{x : x_{n+1} = 1\}$ is an SOC representable set. □

Theorem 4.33. *Let $\mathcal{F}_0^+, \mathcal{F}_1, H^1, \widehat{\mathcal{K}}, \mathcal{F}_s$ be defined as in (4.16) and (4.17). Assume that $\lambda_Q < 0$ and Condition 4.30 holds. Then*

$$\overline{\text{conv}}(\mathcal{F}_0^+ \cap \mathcal{F}_1 \cap \widehat{\mathcal{K}} \cap H^1) = \mathcal{F}_0^+ \cap \mathcal{F}_s \cap \widehat{\mathcal{K}} \cap H^1.$$

Proof. We will first establish that $\text{conv}(\mathcal{F}_0^+ \cap \mathcal{F}_1 \cap \widehat{\mathcal{K}} \cap H^1) = \mathcal{F}_0^+ \cap \mathcal{F}_s \cap \widehat{\mathcal{K}} \cap H^1$. Since the sets $\mathcal{F}^+, \mathcal{F}_s, \widehat{\mathcal{K}}, H^1$ are closed, this will immediately imply our closed convex hull result.

It is clear from the definition of $\mathcal{F}_s$ and Lemma 4.32 that $\text{conv}(\mathcal{F}_0^+ \cap \mathcal{F}_1 \cap \widehat{\mathcal{K}} \cap H^1) \subseteq \mathcal{F}_0^+ \cap \mathcal{F}_s \cap \widehat{\mathcal{K}} \cap H^1$. We will prove the reverse direction.

Let $x = [y; x_{n+1}; x_{n+2}]$ be a vector in $\mathcal{F}_0^+ \cap \widehat{\mathcal{K}} \cap H^1 \cap \mathcal{F}_s$. Then x satisfies

$$\begin{aligned}
x^\top W_0 x &\leq 0, \\
Ay - b x_{n+1} &\in \mathcal{K}, \\
x_{n+1} &= 1, \\
x^\top W_s x &\leq 0.
\end{aligned}$$

We will show that $x \in \text{conv}(\mathcal{F}_0^+ \cap \mathcal{F}_1 \cap \widehat{\mathcal{K}} \cap H^1)$. If $x \in \mathcal{F}_1$, then we are done. Suppose $x \notin \mathcal{F}_1$, that is, $x^\top W_1 x > 0$. Then, from the definition of s, $0 < x^\top W_1 x$, and $x^\top W_s x \leq 0$, we have

$$0 < s(x^\top W_1 x) - x^\top W_s x = -(1-s) x^\top W_0 x = \frac{\lambda_Q}{1 - \lambda_Q}(\|y\|^2 - x_{n+1}^2).$$

Because $\lambda_Q < 0$, this implies $\|y\|^2 < x_{n+1}^2$. Let d be the vector given by Condition 4.30 such that $Qd = \lambda_Q d$, $Ad \in \mathcal{K}$, $-Ad \in \mathcal{K}$, and $\|d\|^2 = 1$. We now consider the points $x^\eta := [y + \eta d; \, x_{n+1}; \, x_{n+2} + 2g^\top d\eta]$ for $\eta \in \mathbb{R}$. We first argue that $x^\eta \in \mathcal{F}_s$ holds for all $\eta \in \mathbb{R}$. To see this,

note that

$$\begin{aligned}
&(y+\eta d)^\top (Q-\lambda_Q I_n)(y+\eta d) + 2g^\top (y+\eta d)x_{n+1} + \lambda_Q x_{n+1}^2 \\
&= (y+\eta d)^\top Q(y+\eta d) + 2g^\top (y+\eta d)x_{n+1} + \lambda_Q(x_{n+1}^2 - \|y+\eta d\|^2) \\
&= y^\top Qy + 2\underbrace{y^\top Qd}_{=\lambda_Q y^\top d}\eta + \underbrace{d^\top Qd}_{=\lambda_Q}\eta^2 + 2g^\top y\ x_{n+1} + 2g^\top d\ x_{n+1}\eta \\
&\quad + \lambda_Q(x_{n+1}^2 - \|y\|^2 - 2y^\top d\eta - \eta^2) \\
&= y^\top Qy + 2g^\top y\ x_{n+1} + \lambda_Q(x_{n+1}^2 - \|y\|^2) + 2g^\top d\ x_{n+1}\eta \\
&= y^\top (Q-\lambda_Q I_n)y + 2g^\top y\ x_{n+1} + \lambda_Q x_{n+1}^2 + 2g^\top d\ x_{n+1}\eta \\
&= x_{n+1}x_{n+2} + (1-\lambda_Q)(x^\top W_s x) + 2g^\top d\ x_{n+1}\eta \\
&\le x_{n+1}x_{n+2} + 2g^\top d\ x_{n+1}\eta \qquad (4.18) \\
&= x_{n+1}(x_{n+2} + 2g^\top d\eta),
\end{aligned}$$

where the third equation follows from $Qd = \lambda_Q d$ and $\|d\|^2 = 1$, and the inequality holds because $x^\top W_s x \le 0$ and $\lambda_Q < 0$. Then from the inequality (4.18) and the definition of $\mathcal{F}_s$ in (4.17), we conclude $x^\eta \in \mathcal{F}_s$ for all $\eta \in \mathbb{R}$. Moreover, because $\|y\|^2 < x_{n+1}^2$ and $d \neq 0$, there must exist $\delta, \epsilon > 0$ such that $\|y - \delta d\|^2 = \|y + \epsilon d\|^2 = x_{n+1}^2$. We define

$$\begin{aligned}
x^\delta &:= [y - \delta d;\ x_{n+1};\ x_{n+2} - 2g^\top d\delta] \\
x^\epsilon &:= [y + \epsilon d;\ x_{n+1};\ x_{n+2} + 2g^\top d\epsilon].
\end{aligned}$$

Then by our choice of δ, ϵ, we have $x^\delta, x^\epsilon \in \mathrm{bd}(\mathcal{F}_0^+)$. From $s \in (0,1)$, $x^\eta \in \mathcal{F}_s$ for all $\eta \in \mathbb{R}$, and the relation

$$(x^\eta)^\top W_s x^\eta = (1-s)[(x^\eta)^\top W_0 x^\eta] + s[(x^\eta)^\top W_1 x^\eta],$$

we conclude that $x^\eta \in \mathcal{F}_1$ for all η such that $x^\eta \in \mathrm{bd}(\mathcal{F}_0^+)$. In particular, $x^\delta, x^\epsilon \in \mathcal{F}_1$. Furthermore, by Condition 4.30, $\pm Ad \in \mathcal{K}$, and since $\mathcal{K}$ is a cone, $-Ad\delta, Ad\epsilon \in \mathcal{K}$; thus $x^\delta, x^\epsilon \in \widehat{\mathcal{K}}$. Finally, $x_{n+1} = 1$ in both x^δ, x^ϵ, and so we have $x^\delta, x^\epsilon \in \mathcal{F}_0^+ \cap \mathcal{F}_1 \cap \widehat{\mathcal{K}} \cap H^1$. Now it is easy to see that

$$x = \frac{\epsilon}{\delta+\epsilon}x^\delta + \frac{\delta}{\delta+\epsilon}x^\epsilon \in \mathrm{conv}(\mathcal{F}_0^+ \cap \mathcal{F}_1 \cap \widehat{\mathcal{K}} \cap H^1).$$

As a consequence, we have the relation

$$\mathcal{F}_0^+ \cap \mathcal{F}_1 \cap \widehat{\mathcal{K}} \cap H^1 \subseteq \mathcal{F}_0^+ \cap \mathcal{F}_s \cap \widehat{\mathcal{K}} \cap H^1 \subseteq \mathrm{conv}(\mathcal{F}_0^+ \cap \mathcal{F}_1 \cap \widehat{\mathcal{K}} \cap H^1).$$

By Lemma 4.32, the set $\mathcal{F}_s \cap H^1$ is SOC representable and hence convex; this implies that $\mathcal{F}_0^+ \cap \mathcal{F}_s \cap \widehat{\mathcal{K}} \cap H^1$ is convex also. Then taking the convex hull of all terms in the above inequality gives us the result. □

Note that the set $\mathcal{F}_0^+ \cap \mathcal{F}_s \cap \widehat{\mathcal{K}} \cap H^1$ is closed. We give our explicit convex hull result for TRS below.

Corollary 4.34. *Let X be the set defined in (4.9). When $\lambda_Q < 0$, under Condition 4.30 we have*

$$\mathrm{conv}(X) = \left\{ x = [y; 1; x_{n+2}] : \begin{array}{rcl} \|y\| & \le & 1 \\ y^\top(Q-\lambda_Q I_n)y + 2g^\top y + \lambda_Q & \le & x_{n+2} \\ Ay - b & \in & \mathcal{K} \end{array} \right\}.$$

As a result,

$$\begin{aligned}\mathrm{Opt}_h &= \min_y \left\{ h(y) = y^\top Q y + 2g^\top y : \begin{array}{rcl} \|y\| & \leq & 1 \\ Ay - b & \in & \mathcal{K} \end{array} \right\} \\ &= \min_y \left\{ f(y) = y^\top (Q - \lambda_Q I_n) y + 2g^\top y + \lambda_Q : \begin{array}{rcl} \|y\| & \leq & 1 \\ Ay - b & \in & \mathcal{K} \end{array} \right\}.\end{aligned}$$

Remark 4.35. In the particular case of TRS with additional conic constraints, i.e., problem (4.1), under Condition 4.30, Corollary 4.34 shows that not only our convex relaxation given by (4.4) is tight but also we can characterize the convex hull of its epigraph exactly. Because Condition 4.30 holds for the classical TRS, this then recovers the results from [44, Section 6.2]. ∎

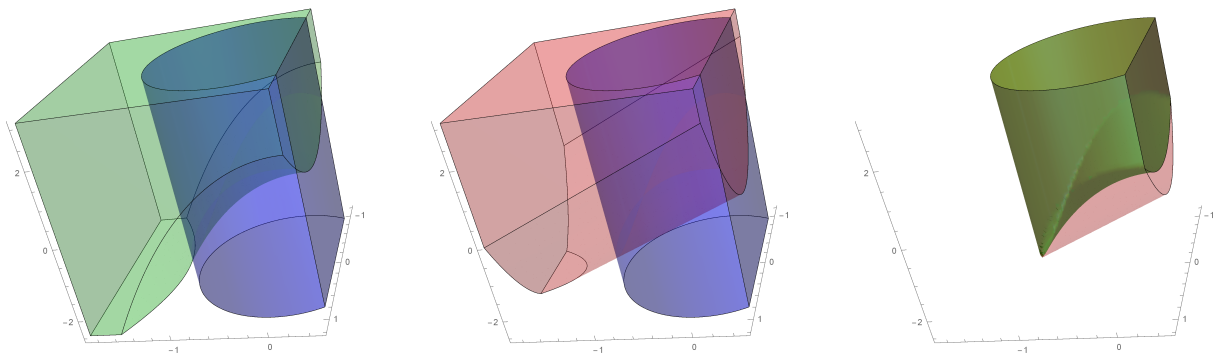
As a consequence of Remark 4.31 and Corollary 4.34, in all of the cases where Jeyakumar and Li [90] show the tightness of their convex reformulation, i.e., for robust least squares and robust SOC programming, we can further give the exact convex hull characterizations of the associated epigraphs.

We next present an example to illustrate that when Condition 4.30 is violated, we may not be able to obtain the convex hull description. We also give a variant of this example to demonstrate that there are cases where our convex relaxation is tight while Condition 4.30 is still violated.

Example 4.36. Consider the following problem with the data given by

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad g = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & -1 \end{bmatrix}, \quad b = \frac{1}{2}, \quad \mathcal{K} = \mathbb{R}_+.$$

In this example, Condition 4.30 is violated. To see this, any vector d such that $Qd = \lambda_Q d$ is of the form $d = [0; d_2]$. But then $Ad = -d_2$. Hence, if $d_2 > 0$ then $Ad \notin \mathcal{K}$, and similarly, if $d_2 < 0$ then $-Ad \notin \mathcal{K}$.

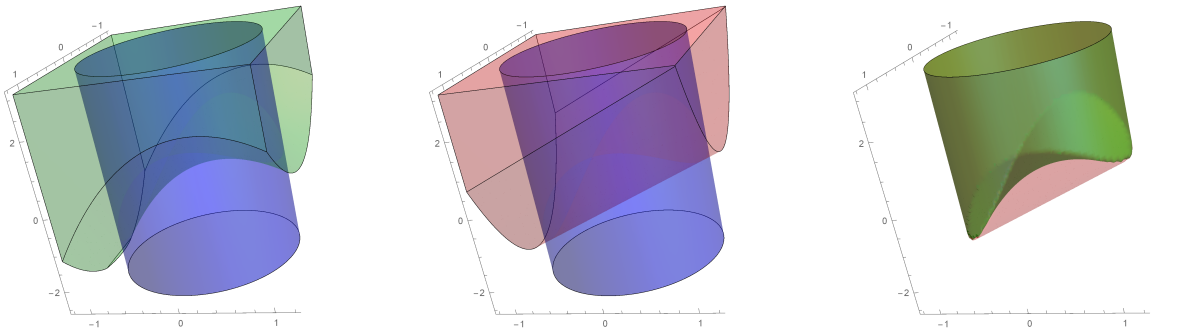


(a) $X = \mathcal{F}_0^+ \cap \mathcal{F}_1 \cap \widehat{\mathcal{K}} \cap H^1$. (b) $\mathcal{F}_0^+ \cap \mathcal{F}_s \cap \widehat{\mathcal{K}} \cap H^1$. (c) X vs $\mathcal{F}_0^+ \cap \mathcal{F}_s \cap \widehat{\mathcal{K}} \cap H^1$.

Figure 4.2: Plots of the epigraph of Example 4.36.

Figure 4.2c shows that the convex relaxation for the epigraph $X = \mathcal{F}_0^+ \cap \mathcal{F}_1 \cap \widehat{\mathcal{K}} \cap H^1$ given by $\mathcal{F}_0^+ \cap \mathcal{F}_s \cap \widehat{\mathcal{K}} \cap H^1$ does not give the convex hull of X. Note also that Condition 4.4 is satisfied for this example by taking $d = [0; 1]$, and so by Theorem 4.5, the SOC optimization problem (4.4) is a tight relaxation for (4.1). Despite this, we cannot give the exact convex hull characterization because Condition 4.30 is violated.

If we were to set $b = 1$ instead of $b = \frac{1}{2}$ in this example, then the linear inequality would become redundant. In this case, our convex relaxation would give the convex hull, as illustrated in Figure 4.3 below. Nevertheless, even in this case Condition 4.30 would still be violated. This demonstrates that Condition 4.30 is not necessary to obtain the convex hull. ∎



(a) $X = \mathcal{F}_0^+ \cap \mathcal{F}_1 \cap \widehat{\mathcal{K}} \cap H^1$. (b) $\mathcal{F}_0^+ \cap \mathcal{F}_s \cap \widehat{\mathcal{K}} \cap H^1$. (c) X vs $\mathcal{F}_0^+ \cap \mathcal{F}_s \cap \widehat{\mathcal{K}} \cap H^1$.

Figure 4.3: Plots of the epigraph of Example 4.36 without the linear inequality.

We close this section with a simple result which highlights a particularly important structure of the extreme points of X.

Lemma 4.37. *Let X be defined as in (4.9). Assume that $\lambda_Q < 0$ and Condition 4.30 also holds. Then any point $[y; 1; x_{n+2}] \in X$ is an extreme point of $\operatorname{conv}(X)$ only if $\|y\| = 1$.*

Proof. Consider $[y; 1; x_{n+2}] \in \operatorname{conv}(X)$ with $\|y\| < 1$. Let $d \neq 0$ be the vector given by Condition 4.30. Because d satisfies $Qd = \lambda_Q d$ and $\pm Ad \in \mathcal{K}$, for any $\epsilon \in \mathbb{R}$,

$$\begin{aligned} f(y + \epsilon d) &= (y + \epsilon d)^\top (Q - \lambda_Q I_n)(y + \epsilon d) + 2g^\top (y + \epsilon d) + \lambda_Q \\ &= [y^\top (Q - \lambda_Q I_n) y + 2g^\top y + \lambda_Q] + 2g^\top d\epsilon \\ &\leq x_{n+2} + 2g^\top d\epsilon. \end{aligned}$$

Now choose $\epsilon^+ > 0$ such that $\|y + \epsilon^+ d\| = 1$, and define $x^+ = [y + \epsilon^+ d; 1; x_{n+2} + 2g^\top d\epsilon^+]$. Then we have $x^+ \in X$, since $\|y + \epsilon^+ d\| = 1$ guarantees $f(y + \epsilon^+ d) = h(y + \epsilon^+ d) \leq x_{n+2} + 2g^\top d\epsilon^+$. Note that we will have $\epsilon^+ > 0$ and finite since $\|y\| < 1$ and $d \neq 0$. Similarly, choosing $\epsilon^- > 0$ such that $\|y - \epsilon^- d\| = 1$, $x^- = [y - \epsilon^- d; 1; x_{n+2} - 2g^\top d\epsilon^-] \in X$ also. Then the point x will be a convex combination of $x^+, x^- \in X$ with weights $\epsilon^-/(\epsilon^- + \epsilon^+)$ and $\epsilon^+/(\epsilon^- + \epsilon^+)$, respectively. □

4.3.3 Additional Hollow Constraints

In this section we explore additional constraints $y \in \mathcal{R}$ included in the domain of TRS (4.1), where $\mathcal{R} = \mathbb{R}^n \setminus \mathcal{P}$ and $\mathcal{P}$ is a given possibly nonconvex set. More precisely, we characterize the convex hull of the set $X \cap \widehat{\mathcal{R}} = \mathcal{F}_0^+ \cap \mathcal{F}_1 \cap \widehat{\mathcal{K}} \cap H^1 \cap \widehat{\mathcal{R}}$ where $\mathcal{F}_0^+$, $\mathcal{F}_1$, $\widehat{\mathcal{K}}$, and H^1 are as defined in (4.16), and $\widehat{\mathcal{R}} := \{[y; x_{n+1}; x_{n+2}] : y \in \mathcal{R}\}$.

We impose the following condition on $\mathcal{R} = \mathbb{R}^n \setminus \mathcal{P}$.

Condition 4.38. The set $\mathcal{P} \subseteq \mathbb{R}^n$ satisfies $\mathcal{P} \subseteq \{y : \|y\| < 1,\ Ay - b \in \mathcal{K}\}$.

Consider the case where $Ay - b \in \mathcal{K}$ is non-existent. If $\mathcal{P} = \bigcup_{i=1}^{m} E_i$ is a union of ellipsoids $E_i = \left\{y : y^\top W_i y + 2b_i^\top y + c_i \leq 0\right\}$ where each $W_i \succ 0$, then Condition 4.38 can be checked by solving

$$v_i = \min_y \left\{1 - \|y\|^2 : \ y^\top W_i y + 2b_i^\top y + c_i \leq 0\right\}.$$

That is, E_i satisfies Condition 4.38 if and only if $v_i > 0$. The computation of v_i as stated above requires solving a nonconvex quadratic program, which is nothing but a classical TRS after an appropriate affine transformation of the variables is applied. Hence, our developments from Section 4.2 give a tight SOC reformulation for it. In addition, the inhomogeneous $\mathcal{S}$-lemma [20, Proposition 3.5.2] ensures that the associated semidefinite relaxation is tight. Thus, Condition 4.38 can be verified efficiently when $\mathcal{P}$ is a union of ellipsoids.

Hollow constraints have been studied in TRS literature under conditions similar to Condition 4.38. Most notably, the *interval-bounded TRS* [21, 31, 122, 144, 153] corresponds to the case when $\mathcal{R}$ is a single lower-bounded quadratic constraint $y^\top D y \geq l$, where $D \succeq 0$ and l is a positive number. The interval-bounded TRS is used to generate new steps in the context of the trust-region algorithm where minimum step lengths are enforced. In the case of interval-bounded TRS, Condition 4.38 is automatically satisfied. It is shown in a number of these papers [122, 153] that the natural SDP relaxation of interval-bounded TRS is tight. More recently, Yang et al. [151] showed the tightness of the SDP relaxation when the hollow set $\mathcal{P}$ is the disjoint union of ellipsoids which do not intersect the boundary of the unit ball $\{y : \|y\| \leq 1\}$. As opposed to these results on tight SDP relaxations, Bienstock [30] has established that the general quadratically constrained quadratic programming problem

$$\min_y \left\{y^\top Q_0 y + 2g_0^\top y : \ y^\top Q_i y + 2g_i^\top y + c_i \leq 0, \ i = 1, \ldots, m\right\}$$

is polynomially solvable for a fixed number of constraints m using a weak feasibility oracle, under the assumption that at least one quadratic constraint $y^\top Q_i y + 2g_i^\top y + c_i \leq 0$ is strictly convex. In a similar vein, Bienstock and Michalka [32] also study TRS with additional ellipsoidal hollow constraints. Instead of giving the convex hull, [32] explores conditions that allow for polynomial solvability using a combinatorial enumeration technique and thus is able to cover cases where the set $\mathcal{P}$ may not be contained in the unit ball. On a related subject, [31] studies the characterization and separation of valid linear inequalities that convexify the epigraph of a convex, differentiable function whose domain is restricted to the complement of a convex set defined by linear or convex quadratic inequalities.

We note that these papers [21, 30, 31, 32, 122, 144, 151, 153] consider the more general case of minimizing an arbitrary quadratic objective, which can be convex, over a domain given by possibly nonconvex quadratic constraints. On the other hand, our result applies to the special case of minimizing a nonconvex quadratic, i.e., $\lambda_Q < 0$, over the unit ball, a convex quadratic constraint. As a result, we are able to relax the assumptions that the set $\mathcal{P}$ is generated by quadratics and the ellipsoidal hollows are disjoint. Specifically, we show that under Condition 4.38, our main convex hull result, i.e.,Theorem 4.33, obtained without the constraint $y \in \mathcal{R}$ is tight.

Theorem 4.39. *Let X be defined in (4.9), and let $\mathcal{R} = \mathbb{R}^n \setminus \mathcal{P}$ be a set satisfying Condition 4.38.*

Assume that $\lambda_Q < 0$ and Condition 4.30 also holds. Then

$$\operatorname{conv}\left(\left\{[y;1;x_{n+2}]: \begin{array}{rcl} \|y\| & \leq & 1 \\ y & \in & \mathcal{R} \\ Ay-b & \in & \mathcal{K} \\ y^\top Q y + 2g^\top y & \leq & x_{n+2} \end{array}\right\}\right) = \operatorname{conv}(X).$$

Proof. Denoting $\widehat{\mathcal{R}} := \{[y; x_{n+1}; x_{n+2}] : y \in \mathcal{R}\}$, our aim is to prove $\operatorname{conv}(X \cap \widehat{\mathcal{R}}) = \operatorname{conv}(X)$. We trivially have $\operatorname{conv}(X \cap \widehat{\mathcal{R}}) \subseteq \operatorname{conv}(X)$. To prove $\operatorname{conv}(X \cap \widehat{\mathcal{R}}) \supseteq \operatorname{conv}(X)$, note that from Lemma 4.37 any point $x = [y; 1; x_{n+2}] \in \operatorname{Ext}(\operatorname{conv}(X))$ satisfies $\|y\| = 1$. Also, by Condition 4.38, the constraint $x \in \widehat{R}$ does not remove any of the points with $\|y\| = 1$, in particular, all of the extreme points of X are also in $\widehat{R}$. Thus, $\operatorname{Ext}(X \cap \widehat{R}) = \operatorname{Ext}(X)$. Moreover, because $\|y\| \leq 1$, the only recessive direction of $\operatorname{conv}(X)$ is $[0; 0; 1]$, i.e., $\operatorname{Rec}(\operatorname{conv}(X)) = \operatorname{cone}([0; 0; 1])$. Note $[0; 0; 1]$ is also a recessive direction in $\widehat{R}$. Then the result follows from

$$\begin{aligned} \operatorname{conv}(X \cap \widehat{R}) &= \operatorname{conv}(\operatorname{Ext}(X \cap \widehat{R})) + \operatorname{Rec}(X \cap \widehat{R}) \\ &= \operatorname{conv}(\operatorname{Ext}(X)) + \operatorname{Rec}(X) = \operatorname{conv}(X). \end{aligned}$$

□

Theorem 4.39 has the following immediate implication.

Corollary 4.40. *When $\lambda_Q < 0$ and $l \leq 1$, an exact convex reformulation of the interval-bounded TRS*

$$\min_y \left\{ y^\top Q y + 2g^\top y : \; l \leq \|y\| \leq 1 \right\}$$

is given by (4.5).

Corollary 4.40 gives a convex reformulation for the interval-bounded TRS with $\lambda_Q < 0$, as do results from [21, 122, 144, 153]. These results were often derived as a consequence of a simultaneously diagonalizable assumption of the underlying matrices associated with TRS, or through SDP relaxations. In contrast, Condition 4.38 and Theorem 4.39 highlight the important geometric aspect, and provide a convex reformulation without additional variables. In addition, Corollary 4.40 together with Theorem 4.18 establish the convergence rate of FOMs to solve interval-bounded TRS as opposed to specialized algorithms suggested in [122].

4.4 Application to Robust Quadratic Programming

In this section we walk through the setup and resulting convergence rates of our primal-dual framework for robust optimization from Chapter 2.4 together with the convexification of the TRS for a robust quadratic program (QP) with ellipsoidal uncertainty. To be precise, our deterministic feasibility problem is

$$\text{find} \quad x \in X \quad \text{s.t.} \quad \|A_i x\|_2^2 \leq b_i^\top x + c_i, \quad \forall i \in [m],$$

where $X \subseteq \mathbb{R}^n$ is the unit Euclidean ball, $A_i \in \mathbb{R}^{n \times n}$, $b_i \in \mathbb{R}^n$, and $c_i \in \mathbb{R}$ for all $i \in [m]$. We consider the robust quadratic feasibility problem given by

$$\text{find} \quad x \in X \quad \text{s.t.} \quad \max_{u \in \widehat{U}} \left\| \left(A_i + \sum_{k=1}^{K} P_k^i u^k \right) x \right\|_2^2 - b_i^\top x - c_i \leq 0, \quad \forall i \in [m], \tag{4.19}$$

where $P_1^i, \ldots, P_K^i$ are uncertainty matrices for each constraint $i \in [m]$, for simplicity we assume uncertainty sets $U^i = \widehat{U} = \{u \in \mathbb{R}^K : \|u\|_2 \leq 1\}$ for all $i \in [m]$, and u^k denotes the k-th entry of u.

It is well known that the robust counterpart of this feasibility problem is a semidefinite program [23, 29]. Because current state-of-the-art QP solvers can handle two to three orders of magnitude larger QPs than semidefinite programs (SDPs), Ben-Tal et al. [25, Section 4.2] suggest an approach that avoids solving SDPs associated with robust QPs. Their approach relies on running a probabilistic OCO algorithm in which a trust region subproblem (TRS) is solved in each iteration. Our primal-dual framework in Chapter 2.4 together with the convexification results in this chapter further enhance this approach. In particular, we show that we can achieve the same rate of convergence in our framework while working with a deterministic OCO algorithm through using our convexification result for the TRS. In fact, the most expensive operation involved with each iteration of our approach is a maximum eigenvalue computation. Because maximum eigenvalue computation is cheaper than solving a TRS, we not only present a deterministic approach but also reduce the cost of each iteration.

To simplify our exposition, let us introduce some notation. For each $i \in [m]$ and fixed $x \in X$, we define the matrix $\mathcal{P}_x^i \in \mathbb{R}^{n \times K}$ whose columns are given by the vectors $P_k^i x$ for $k \in [K]$ together with

$$Q_x^i := (\mathcal{P}_x^i)^\top \mathcal{P}_x^i \in \mathbb{S}_+^K, \quad r_x^i := (\mathcal{P}_x^i)^\top A_i x \in \mathbb{R}^K, \quad \text{and} \quad s_x^i := \|A_i x\|_2^2 - b_i^\top x - c_i \in \mathbb{R};$$

then it is easy to check that for all $i \in [m]$ and $u \in \mathbb{R}^K$ we have

$$\left\| \left(A_i + \sum_{k=1}^K P_k^i \, u^k \right) x \right\|_2^2 - b_i^\top x - c_i = u^\top Q_x^i u + 2(r_x^i)^\top u + s_x^i.$$

This shows that for each $i \in [m]$, the maximum over $u \in \widehat{U}$ in (4.19) is an instance of a TRS: we must maximize a non-concave (in fact convex) quadratic over the unit ball. For each $i \in [m]$, we define $f^i : X \times \widehat{U} \to \mathbb{R}$ as

$$\begin{aligned} f^i(x,u) &:= \left\| \left(A_i + \sum_{k=1}^K P_k^i \, u^k \right) x \right\|_2^2 - b_i^\top x - c_i + \lambda_{\max}(Q_x^i) \left(1 - \|u\|_2^2\right) \\ &= u^\top Q_x^i u + 2(r_x^i)^\top u + s_x^i + \lambda_{\max}(Q_x^i) \left(1 - \|u\|_2^2\right). \end{aligned} \tag{4.20}$$

The convexification now follows from Theorem 4.7. We also check that $f^i(x,u)$ is a convex-concave function in x and u, which allows us to apply the algorithms from Chapter 3 to choose x_t, u_t within our primal-dual framework of Chapter 2.4.

Lemma 4.41. *For each $i \in [m]$, the function $f^i(x,u)$ defined in (4.20) is convex in x for any fixed $u \in \widehat{U}$ and concave in u for any given x. Moreover, for all $i \in [m]$ and for any $x \in X$,*

$$\max_{u \in \widehat{U}} \left\| \left(A_i + \sum_{k=1}^K P_k^i \, u^k \right) x \right\|_2^2 - b_i^\top x - c_i = \max_{u \in \widehat{U}} f^i(x,u).$$

Proof of Lemma 4.41. Fix $i \in [m]$. By rearranging terms in (4.20), we obtain $f^i(x,u) = u^\top (Q_x^i - \lambda_{\max}(Q_x^i) I_K) u + 2(r_x^i)^\top u + s_x^i + \lambda_{\max}(Q_x^i)$. Since $Q_x^i - \lambda_{\max}(Q_x^i) I_K \in \mathbb{S}_+^K$ for any given x, $f^i(x,u)$ is concave in u for any given x.

Now consider a fixed $u \in \widehat{U}$. Note that

$$\lambda_{\max}(Q_x^i) = \max_{\|v\|_2 \le 1} v^\top (Q_x^i) v = \max_{\|v\|_2 \le 1} \sum_{1 \le j,k \le K} v^{(j)} v^k x^\top (P_j^i)^\top P_k^i \, x = \max_{\|v\|_2 \le 1} x^\top \left(\sum_{k=1}^K P_k^i \, v^k \right)^\top \left(\sum_{k=1}^K P_k^i \, v^k \right) x.$$

Because $\left(\sum_{k=1}^K P_k^i \, v^k \right)^\top \left(\sum_{k=1}^K P_k^i \, v^k \right) \in \mathbb{S}_+^n$, then $\lambda_{\max}(Q_x^i)$ is a maximum of convex quadratic functions of x and hence is convex in x. Thus, for fixed $u \in \widehat{U}$, $f^i(x,u)$ is convex in x.

Reformulation of the nonconvex QP over an ellipsoid into a convex QP over the ellipsoid via the relation between $u^\top Q_x^i u + 2(r_x^i)^\top u + s_x^i$ and $f^i(x,u)$ in (4.20) follows from Theorem 4.7. □

Lemma 4.41 implies that $\max_{u \in \widehat{U}} f^i(x,u) \le 0$ is an alternate representation of our robust quadratic constraint. We next state the convergence rate in our framework for the associated feasibility problem. For this, we define the quantities

$$\begin{aligned} \sigma^2 &:= \max_{i \in [m]} \sum_{k=1}^K \|P_k^i\|_{\text{Fro}}^2, \quad & \chi &:= \max_{i \in [m]} \max_{k \in [K]} \|P_k^i\|_{\text{Spec}}, \quad \text{and} \\ \rho &:= \max_{i \in [m]} \|A_i\|_{\text{Spec}}, & \beta &:= \max_{i \in [m]} \|b_i\|_2, \end{aligned} \tag{4.21}$$

where $\|\cdot\|_{\text{Fro}}$ is the Frobenius norm of a matrix, and $\|\cdot\|_2$ is the spectral norm. Note that $\chi \le \sigma$. Furthermore, Ben-Tal et al. [25, Lemma 7] proves that $\|Q_x^i\|_{\text{Fro}} \le \sigma^2$ and $\|r_x^i\|_2 \le \sigma\rho$ holds for all x such that $\|x\|_2 \le 1$.

Corollary 4.42. *Let our domain be given by $X = \{x \in \mathbb{R}^n : \|x\|_2 \le 1\}$. The customization of our OFO-based approach to the problem (4.19) ensures that within $O\left(((\rho + \sqrt{K}\sigma)^2 + \beta)^2\right) \epsilon^{-2}$ iterations, we obtain a robust feasibility/infeasibility certificate. Moreover, each iteration in our framework relies on a first-order update where the most expensive operation in the case of (4.19) is computing $\lambda_{\max}(Q_x^i)$, which can be done efficiently.*

Proof of Corollary 4.42. Let $U = \widehat{U} \times \ldots \times \widehat{U}$ (m times). Recall that the primal-dual framework from Chapter 2.4 states that we must bound the two terms

$$\begin{aligned} \widehat{\epsilon}\left(\{x_t, y_t, \tilde{u}_t, \theta_t\}_{t \in [T]} \right) &= \max_{i \in [m]} \sum_{t \in [T]} \theta_t f^i(x_t, u_t^i) - \min_{x \in X} \sum_{t \in [T]} \theta_t \sum_{i \in [m]} y_t^i f^i(x, u_t^i) \\ &= \max_{y \in Y} \frac{1}{\Theta_T} \sum_{t \in [T]} \theta_t \sum_{i \in [m]} y^i f^i(x_t, u_t^i) - \min_{x \in X} \frac{1}{\Theta_T} \sum_{t \in [T]} \theta_t \sum_{i=1}^m y_t^i f^i(x, u_t^i) \\ &= \max_{y \in Y} \frac{1}{\Theta_T} \sum_{t \in [T]} \theta_t \sum_{i \in [m]} y^i f^i(x_t, u_t^i) - \frac{1}{\Theta_T} \sum_{t \in [T]} \theta_t \sum_{i \in [m]} y_t^i f^i(x_t, u_t^i) \\ &\quad + \frac{1}{\Theta_T} \sum_{t \in [T]} \theta_t \sum_{i \in [m]} y_t^i f^i(x_t, u_t^i) - \min_{x \in X} \frac{1}{\Theta_T} \sum_{t \in [T]} \theta_t \sum_{i \in [m]} y_t^i f^i(x, u_t^i) \end{aligned}$$

$$\max_{\tilde{u} \in U} \epsilon^\circ \left(\{x_t, \tilde{u}_t, \theta_t\}_{t \in [T]} ; \tilde{u} \right) = \max_{i \in [m]} \left\{ \max_{u \in \widehat{U}} \frac{1}{\Theta_T} \sum_{t \in [T]} \theta_t f^i(x_t, u) - \frac{1}{\Theta_T} \sum_{t \in [T]} \theta_t f^i(x_t, u_t^i) \right\}$$

and then we can apply Theorem 2.8 to get out feasibility guarantee. Here, $y_t \in Y := \Delta_m$ and $\tilde{u}_t \in U$ for $t \in [T]$.

To get convergence rates, we apply the mirror descent algorithm (Theorem 3.4) to choose sequences $\{x_t\}_{t\in[T]}$, $\{u_t^i\}_{t\in[T]}$ for $i \in [m]$, and then we choose $y_t^i = 1$ if $f^i(x_t, u_t^i) = \max_{i'\in[m]} f^{i'}(x_t, u_t^{i'})$ and $y_t^i = 0$ otherwise for $i \in [m]$ (if there is more than one such i, we choose one arbitrarily). This guarantees that

$$\begin{aligned}
\max_{y\in Y} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \sum_{i\in[m]} y^i f^i(x_t, u_t^i) - \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \sum_{i\in[m]} y_t^i f^i(x_t, u_t^i) &\le 0 \\
\frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \sum_{i\in[m]} y_t^i f^i(x_t, u_t^i) - \min_{x\in X} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \sum_{i\in[m]} y_t^i f^i(x, u_t^i) &\le \frac{O(1)}{\sqrt{T}} \\
\max_{u\in\widehat{U}} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t f^i(x_t, u) - \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t f^i(x_t, u_t^i) &\le \frac{O(1)}{\sqrt{T}}, \quad \forall i \in [m].
\end{aligned}$$

This gives us bounds on $\widehat{\epsilon}\left(\{x_t, y_t, \tilde{u}_t, \theta_t\}_{t\in[T]}\right), \max_{\tilde{u}\in U} \epsilon^\circ\left(\{x_t, \tilde{u}_t, \theta_t\}_{t\in[T]}; \tilde{u}\right)$ above.

To get explicit rates, we now compute the constants in Theorem 3.4 in the context of our robust QP setting. We first need to customize our proximal setup. Given that the sets X and $\widehat{U}$ are Euclidean balls, we choose the proximal setup for generating the iterates $\{x_t, u_t^i\}_{t=1}^T$ to be the standard Euclidean d.g.f. $\omega(\cdot) = \|\cdot\|/2$ with $\|\cdot\|_2$-norm, and thus $\Omega_X = \Omega_{\widehat{U}} = \frac{1}{2}$.

We must bound the magnitude of the gradients measured by the $\|\cdot\|_2$-norm. Note that for any $i \in [m]$, the gradients of f^i are given by

$$\begin{aligned}
\nabla_u f^i(x, u) &= 2\left(Q_x^i - \lambda_{\max}(Q_x^i) I_K\right) u + 2r_x^i \\
\nabla_x f^i(x, u) &= 2\left(A_i + \sum_{k=1}^K P_k^i\, u^k\right)^\top \left(A_i + \sum_{k=1}^K P_k^i\, u^k\right) x + 2\left(1 - \|u\|_2^2\right)\left(\sum_{k=1}^K P_k^i\, v^k\right)^\top \left(\sum_{k=1}^K P_k^i\, v^k\right) x - b_i,
\end{aligned}$$

where $v \in \widehat{U}$ is an eigenvector of Q_x^i corresponding to $\lambda_{\max}(Q_x^i)$.

Let us fix an $i \in [m]$. We first bound $\|\nabla_u f^i(x, u)\|_2$ for any $u \in \widehat{U}$ as follows:

$$\begin{aligned}
\|\nabla_u f^i(x, u)\|_2 &= 2\left\|\left(Q_x^i - \lambda_{\max}(Q_x^i) I_K\right) u + r_x^i\right\|_2 \\
&\le 2\left(\left\|\left(Q_x^i - \lambda_{\max}(Q_x^i) I_K\right) u\right\|_2 + \left\|r_x^i\right\|_2\right) \quad \le 2\lambda_{\max}(Q_x^i)\|u\|_2 + 2\sigma\rho \le 2(\sigma^2 + \sigma\rho),
\end{aligned}$$

where the second inequality follows from $\left\|Q_x^i - \lambda_{\max}(Q_x^i) I_K\right\|_{\text{Spec}} \le \lambda_{\max}(Q_x^i)$ and $\|r_x^i\|_2 \le \sigma\rho$ which is implied by Ben-Tal et al. [25, Lemma 7], and the last inequality follows from the facts that $u \in \widehat{U}$, the definitions given in (4.21), and $\lambda_{\max}(Q_x^i) = \|\mathcal{P}_x^i\|_{\text{Spec}}^2 \le \|\mathcal{P}_x^i\|_{\text{Fro}}^2 \le \sum_{k=1}^K \|P_k^i\|_{\text{Fro}}^2 \le \sigma^2$ for any $x \in X$. Therefore, we deduce from Theorem 3.4 that the rate of convergence for bounding the weighted regret associated with constraint $i \in [m]$ using the online mirror descent algorithm is

$$\max_{u\in\widehat{U}} \frac{1}{T} \sum_{t=1}^T f^i(x_t, u) - \frac{1}{T} \sum_{t=1}^T f^i(x_t, u_t^i) \le \frac{2(\sigma^2 + \sigma\rho)}{\sqrt{T}}.$$

We next bound the $\|\cdot\|_2$-norm of $\nabla_x \left(\sum_{i\in[m]} y_t^i f^i(x,u_t^i)\right) = \sum_{i\in[m]} y_t^i \nabla_x f^i(x,u_t^i)$. Notice that

$$\begin{aligned}\left\|\nabla_x \left(\sum_{i\in[m]} y_t^i f^i(x,u_t^i)\right)\right\|_2 &\le \sum_{i\in[m]} y_t^i \|\nabla_x f^i(x,u_t^i)\|_2 \\ &\le \max_{i\in[m]} \|\nabla_x f^i(x,u_t^i)\|_2.\end{aligned}$$

Thus, we must bound $\|\nabla_x f^i(x,u)\|_2$ for all $x\in X$, $u\in\widehat{U}$. To this end, note that for any $u\in\widehat{U}$

$$\left\|\sum_{k=1}^K P_k^i\, u^k\right\|_{\text{Spec}} \le \sum_{k=1}^K \|P_k^i\|_{\text{Spec}}\,|u^k| \le \sqrt{K}\max_{k\in[K]}\|P_k^i\|_{\text{Spec}} \le \sqrt{K}\chi,$$

where the second inequality holds because $\|u\|_1 \le \sqrt{K}\|u\|_2 \le \sqrt{K}$ holds for all $u\in\widehat{U}$. Then for any $x\in X$, $u\in\widehat{U}$, and eigenvector $v\in\widehat{U}$, we have

$$\begin{aligned}\|\nabla_x f^i(x,u)\|_2 &\le 2\left\|A_i + \sum_{k=1}^K P_k^i\, u^k\right\|_{\text{Spec}}^2 \|x\|_2 + 2\left(1-\|u\|_2^2\right)\left\|\sum_{k=1}^K P_k^i\, v^k\right\|_{\text{Spec}}^2 \|x\|_2 + \|b_i\|_2 \\ &\le 2(\rho+\sqrt{K}\chi)^2 + 2K\chi^2 + \beta \\ &\le 4(\rho+\sqrt{K}\sigma)^2 + \beta.\end{aligned}$$

Hence, $\left\|\nabla_x \left(\sum_{i\in[m]} y_t^i f^i(x,u_t^i)\right)\right\|_2 \le 4(\rho+\sqrt{K}\sigma)^2+\beta$. Then Theorem 3.4 implies

$$\frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t \sum_{i\in[m]} y_t^i f^i(x_t,u_t^i) - \min_{x\in X}\frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t\sum_{i\in[m]} y_t^i f^i(x,u_t^i) \le \frac{\left(4(\rho+\sqrt{K}\sigma)^2+\beta\right)}{\sqrt{T}}.$$

Note that each iteration of our approach requires a first-order update that is composed of computing the gradients $\nabla_x f^i(x,u)$ and $\nabla_u f^i(x,u)$ and prox computations. Because our domains involve only direct products of Euclidean balls, they admit efficient prox computations which take $O(Km+mn)$ time. In order to evaluate the gradients $\nabla_x f^i(x,u)$ and $\nabla_u f^i(x,u)$, in addition to the elementary matrix vector operations, we need to compute $\lambda_{\max}(Q_x^i)$ which is the most expensive operation in our first-order update. Fortunately, computing the maximum eigenvalue of a matrix is a well-studied problem and can be computed very efficiently. □

In the case of robust QP feasibility problem (4.19), Ben-Tal et al. [25, Corollary 3] states that with probability $1-\delta$, their framework returns robust feasibility/infeasibility certificates in at most $O\left(K^2\sigma^2(\rho^2+\sigma^2)\log(m/\delta)\epsilon^{-2}\right)$ calls (iterations) to their oracle. In each call to their oracle, a nominal feasibility problem is solved to the accuracy $\epsilon/2$. In comparison we deduce from Corollary 4.42 that our framework requires comparable number of iterations as the approach of Ben-Tal et al. [25]. Even so, there are a number of reasons that considerably favor our approach. First, our approach is deterministic as opposed to the high $1-\delta$ probability guarantee of Ben-Tal et al. [25] which requires using an adaptation of the follow-the-perturbed-leader type OCO. Second, each iteration of their approach requires solving a nominal feasibility problem for solution oracle

as well as solving TRSs for the computation of noises u_t. In contrast to this, in each iteration we carry out mainly elementary operations such as matrix vector multiplications and our most computationally expensive operation is the maximum eigenvalue computations $\lambda_{\max}(Q_x^i)$. While there are established algorithms to solve the TRS, it is inherently more complicated than finding the maximum eigenvalue of a positive semidefinite matrix. Moreover, the approach of Ben-Tal et al. [25] suffers from the additional computational cost of their solution oracle which solves the nominal feasibility problem. Hence, our approach, while requiring a comparable number of iterations, reduces the cost per iteration remarkably.

4.4.1 Numerical Study

In this section, we conduct a numerical study comparing the approaches discussed so far. We consider the following quadratic program inspired by mean-variance portfolio optimization problems with a factor model for the return vector (see, e.g., Goldfarb and Iyengar [67]):

$$\min_x \left\{ \|Vx\|_2^2 + x^\top Dx - \lambda \mu^\top x : \ x \in \Delta_n \right\}, \tag{4.22}$$

where $\mu \in \mathbb{R}^n$ is the expected return vector, the term $x^\top (V^\top V + D)x$ captures the risk associated with the portfolio via a factor model, and $\lambda \geq 0$ represents the trade-off between the expected return of the portfolio and the risk associated with the portfolio.

In the robust formulation of (4.22), we consider the case where the true parameters $\mu \in \mathbb{R}^n$ and $V \in \mathbb{R}^{m \times n}$ belong to uncertainty sets $\mathcal{M}$ and $\mathcal{V}$ of form

$$\mathcal{M} := \{\mu : \ \mu_0 - \gamma \leq \mu \leq \mu_0 + \gamma\}, \quad \mathcal{V} := \left\{ V = V_0 + \sum_{k=1}^K P_k u_k : \ \|u\|_2 \leq 1 \right\},$$

where the nominal data $\mu_0 \in \mathbb{R}^n$, $\gamma \in \mathbb{R}^n$, and $V_0 \in \mathbb{R}^{m \times n}, \{P_k \in \mathbb{R}^{m \times n}\}_{k=1}^K$ are given to us. Then the robust problem is given by

$$\min_x \left\{ \max_{V \in \mathcal{V}} \|Vx\|_2^2 + x^\top Dx - \lambda \min_{\mu \in \mathcal{M}} \mu^\top x : \ x \in \Delta_n \right\}. \tag{4.23}$$

Our test instances are synthetically generated, largely following the random instance generation model from Goldfarb and Iyengar [67]. We begin by specifying three parameters: n, the number of variables; m, the number of factors (which controls the rank of V); and $\alpha \in (0, 1)$, a parameter controlling the size of the uncertainty sets. For each instance, we randomly generate matrices $V \in \mathbb{R}^{m \times n}$ and $F \in \mathbb{R}^{m \times m}$, where we ensure F is positive semidefinite, and define $D = 0.1 \operatorname{Diag}(V^\top F V)$. We then generate $p > m$ factor samples $f_{(l)} \in \mathbb{R}^m$, $l \in [p]$, where each $f_{(l)} \sim N(0, F)$, and we also generate $\mu \in \mathbb{R}^n$ where each entry $\mu_i \sim U(1, 5)$. We then set $\mu_{(l)} = \mu + V^\top f_{(l)} + \epsilon_l$, where $\epsilon_{(l)} \sim N(0, D)$ are independent of the factor sample $f_{(l)}$. The matrices μ and V are estimated via linear regression on $\mu_{(l)}$ and $f_{(l)}$, to obtain $\bar{\mu}, \bar{V}$. The nominal data for (4.22) are set to be $\mu_0 = \bar{\mu}$, $V_0 = F^{1/2}\bar{V}$. To define the uncertainty sets, we first compute the scaled sum of squared errors for each $i \in [n]$, $s_i^2 = \frac{1}{p-m-1} \sum_{l=1}^p (\mu_{(l),i} - \mu_{0,i} - V_{0,i}^\top f_{(l)})^2$. Let $c_J(\alpha)$ be the α-critical value of an F-distribution with J degrees of freedom, and let ν be the top-left entry of A^{-1}, where $A \in \mathbb{R}^{(p+1)\times(p+1)}$ is the Gram matrix of the vectors $\mathbf{1}_m, \{f_{(l)}\}_{l=1}^p$. Then we set $\gamma_i = \sqrt{\nu c_1(\alpha) s_i^2}$ for $i \in [n]$, which defines the uncertainty set for μ. The uncertainty set for V is chosen by randomly

generating matrices P_k, and then scaling them appropriately so that the norm of each column i of $V - V_0$ is at most $\sqrt{mc_m(\alpha)s_i^2}$ for every $V \in \mathcal{V}$.

We set $p = 90$ and $\alpha = 0.95$, while varying $m \in \{3, 5, 7, 10, 15, 20, 25\}$ and $n \in \{100, 200, \ldots, 1000\}$. We fix the underlying dimension of the uncertainty set $\mathcal{V}$ to be $K = \min\{2m, 15\}$. We generate five instances for each combination of m and n.

The four approaches we test are our fully online first-order (OFO) based approach outlined in Section 4.4, our FO-based pessimization approach from Section 2.4.1.1 (see Theorem 2.10), the nominal oracle-based approach of Ben-Tal et al. [25] from Section 2.4.1.2 (see Theorem 2.12), and the full pessimization approach of [107], which requires both a pessimization and an extended nominal feasiblity oracle. Since (4.23) is an instance of a robust quadratic program, the form for nominal and pessimization oracles can be derived from Section 4.4. One-dimensional line search using Brent's algorithm [39] was used to choose step sizes for each iteration of FO-based methods. An error tolerance of $\epsilon = 0.002$ is used in all instances.

Experiments are performed on a Linux machine with 2.8GHz processor and 64GB memory using Python v3.5.2. Whenever the nominal (extended nominal) oracles and pessimization oracles do not have closed form solutions, they are implemented in Gurobi v7.0.2. We use standard Gurobi tolerances and parameter choices. We employ the implementation of Brent's algorithm in Python's `scipy.optimize` package.

Figure 4.4 plots the average solve times in seconds against different n for each of the approaches, averaging across all m. As we expect, for low dimensions $n \leq 300$, the oracle-based approaches solve the instances very quickly compared to our first-order based approaches. However, when $n \geq 400$, we see that the solution times of our first-order based approaches beat the nominal oracle approach. When $n \geq 700$, our FO-based pessimization approach beats the full pessimization approach, while our OFO-based approach beats the full pessimization approach when $n \geq 900$.

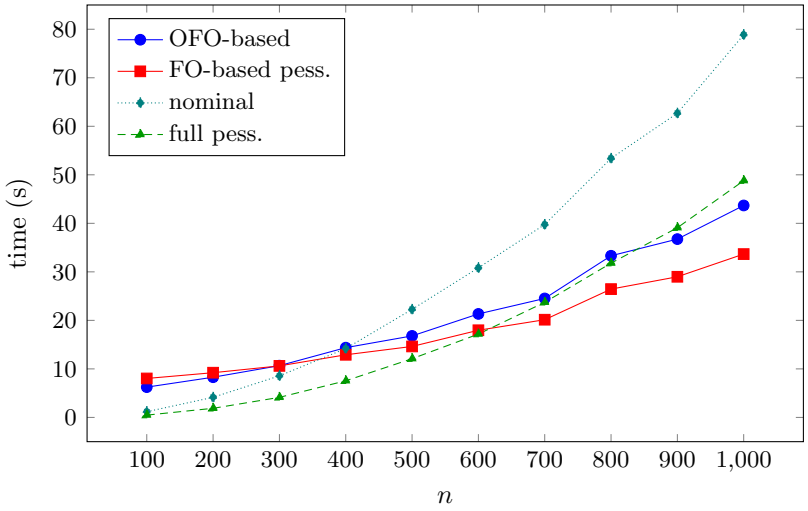


Figure 4.4: Average solve times (seconds) for different n.

The parameter m influences the rank of the nominal matrix $V_0^\top V_0$ and controls the difficulty of the problems. Examining the results for different m further highlights the benefits of utilizing the first-order based approaches. Figure 4.5 plots average solve times for different m while fixing

$n = 700, 800, 900, 1000$. For the oracle-based methods, the solution times increase with m, while the solution times for first-order based methods remains relatively constant with m. For $m \geq 15$, we observe that our first-order based approaches significantly outperforms the oracle-based methods which require a nominal solver. Notice that, while we expect our OFO-based approach

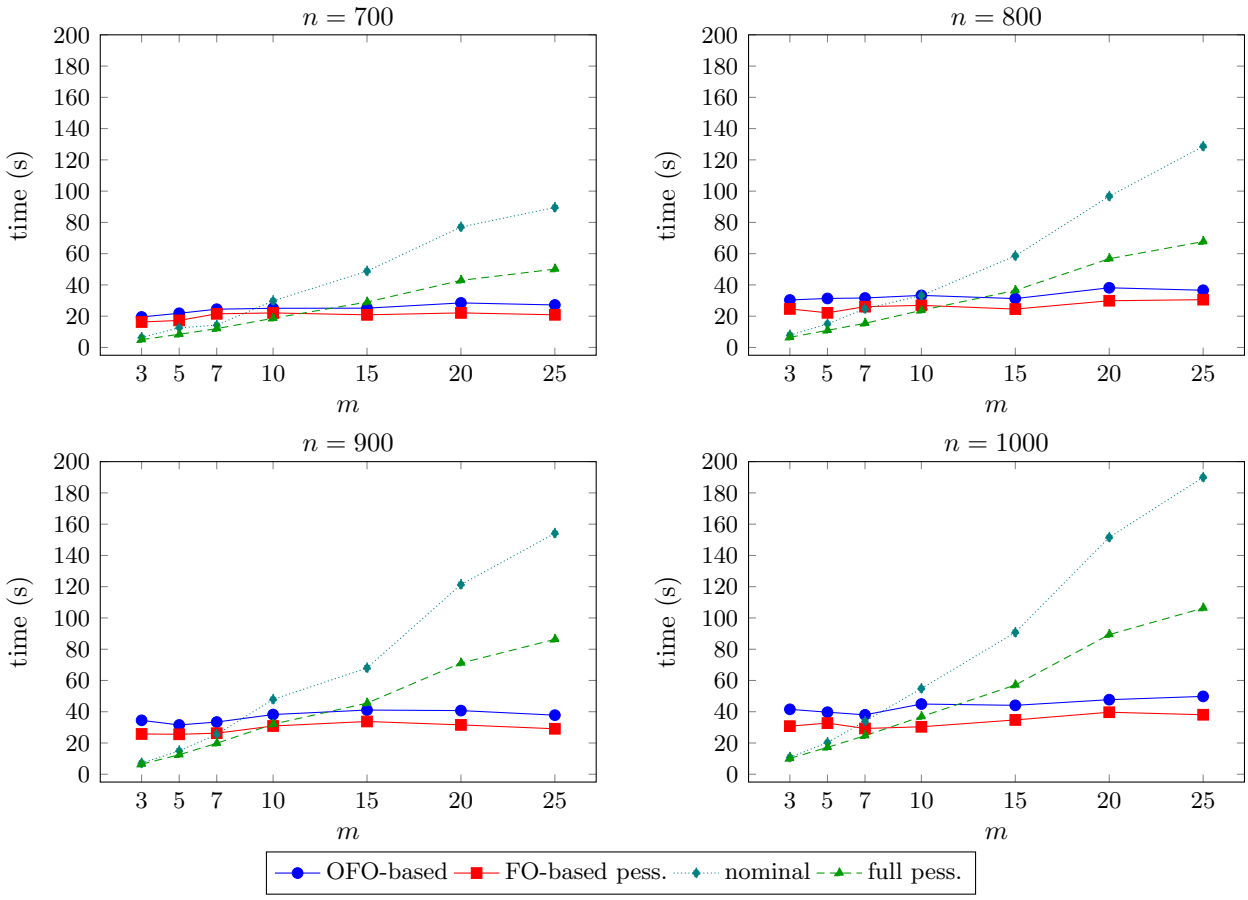


Figure 4.5: Average solve times (seconds) for different n and m.

to outperform the FO-based pessimization approach due to the burden of solving an eigenvalue problem in each iteration for computing the pessimization oracle, our results indicate the opposite. This is because for small values of K, calling a pessimization oracle is faster than the line search performed in the FO-based noise update. However, we believe that as K increases, one-dimensional line search will become more efficient.

Finally, we examine the number of iterations and cost per iteration of different approaches averaged across all instances in Table 4.1. We observe that, contrary to their theoretical iteration guarantees, the oracle-based approaches of [107, 25] need very few iterations to find a solution. However, as expected, the average time per iteration is significantly higher for these methods due to their reliance on full nominal optimization solvers. This further highlights the benefit of utilizing first-order methods for robust optimization when the deterministic version of the problem is already very expensive, and hence nominal oracles become expensive.

	# iterations	seconds per iteration
OFO-based	977.064	0.022
FO-based pessimization	1009.541	0.018
nominal	3.480	9.412
full pessimization	1.974	9.464

Table 4.1: Average number of iterations and average time per iteration for each approach.

Chapter 5

Dynamic Data-Driven Estimation of Non-Parametric Choice Models via the Primal-Dual Framework

5.1 Introduction

A choice model is an effective tool to summarize and understand the preferences of a consumer population over a set of items. Such models give choice probabilities, that is, the probability that a consumer will choose a particular item from a given subset. Choice models are prevalent in several application areas such as revenue management, web page ranking, betting theory, social choice, marketing, and economics (see Dwork et al. [57], Talluri and van Ryzin [146], Jagabathula and Shah [87], Farias et al. [61], Desir et al. [52] and references therein). A good choice model aims to capture complex substitution behaviors of consumers in order to accurately describe demand from limited observations.

Choice model estimation has received quite a bit of interest. Traditional choice models often specify a *parametric* structure for the probability distribution (examples include the multinomial logit (MNL), nested logit, and mixed MNL models), see Talluri and van Ryzin [146] and references therein. In most cases, imposing a parametric structure makes estimation of the necessary parameters a simpler task. However, this is often at the expense of overly facile assumptions on consumer behavior (such as independence of irrelevant alternatives in MNL models) preventing us from accurately capturing the substitution behavior. Therefore, the *non-parametric* approach of directly estimating the probability distribution for rankings has drawn growing interest in academia and in practice [131, 62, 63], and through case studies, it is shown to lead to substantial improvement in prediction accuracy (see Farias et al. [62], Jagabathula and Rusmevichientong [86]). In this paper, we focus on the non-parametric approach to choice modeling in a dynamic observation setting, and develop convex optimization based approaches that are equipped with convergence guarantees and has direct implications on the sparsity of the final model.

5.1.1 Related Literature

Earliest studies on non-parametric choice models appear in the economics and psychology literatures, e.g., Block and Marschak [34]. Mahajan and van Ryzin [103] showed that non-parametric models capture a number of parametric models as special cases. The recent literature focuses on the *static* estimation of non-parametric choice models where the aim is to find a model that either

matches the observed empirical probabilities with the model-based choice probabilities exactly, or minimizes a distance measure between the two. Farias et al. [62] aim to estimate a choice model that recovers the empirical choice probabilities precisely, and suggest a solution procedure via a dual-based constraint sampling method. In contrast to this, more recent literature focuses on distance minimization approaches, which differ essentially in terms of their choice of the distance measure used. In this respect, van Ryzin and Vulcano [147] and Bertsimas and Mišic [27] seek to minimize the Kullback-Leibler (KL) divergence and ℓ_1-norm respectively, and both suggest column generation to solve their associated minimization problems. While Farias et al. [62] and van Ryzin and Vulcano [147] provide useful recovery results under some assumptions on the observational data, none of the aforementioned methods discuss conditions for convergence, guarantees on the convergence rates, or implications on the sparsity of the resulting estimated choice models.

As opposed to the specific distance measures used in the prior literature, the recent work of Jagabathula and Rusmevichientong [86] focuses on general distance measures. While they suggest using the Frank-Wolfe (F-W) algorithm to estimate a non-parametric choice model, the bulk of their work focuses on a particular combinatorial subproblem that arises in all of the previous methods for non-parametric choice estimation. Their main contribution is the characterization of sufficient conditions (in terms of the subset structure of the items) under which this subproblem becomes polynomial-time solvable. Nevertheless, much like the rest of the literature, they establish neither conditions ensuring convergence of the overall F-W algorithm for the usual distance measures used in choice modeling nor the rate of convergence or potential implications on the model sparsity. In the static setting, these four approaches, Farias et al. [62], van Ryzin and Vulcano [147], Bertsimas and Mišic [27], and Jagabathula and Rusmevichientong [86], are closely related to our work, thus we further discuss and compare them in Appendix B.1.

We now discuss a few important aspects in the estimation of non-parametric choice models that have not been scrutinized explicitly in the prior literature, namely, sparsity, finite sample errors and dynamic data.

In a full non-parametric model, there are a factorially many (in the number of items) probabilities to estimate, and thus estimating (or even specifying) a full non-parametric model is intractable even for moderate-sized problems. Thus, to tractably estimate such a model, we must aim to estimate sparse models where most of these probabilities are zero. This necessarily places an importance on characterizing the model sparsity versus model accuracy trade off in the estimation of non-parametric choice models. Moreover, sparse choice models are more efficient (and thus preferable) when computing the associated choice probabilities. Even though some of the existing approaches stem from the desire to achieve model simplicity, none of them is equipped with explicit convergence guarantees and consequently none can provide sparsity guarantees on the estimated model. Thus, efficient solution methods for non-parametric model estimation (specifically the ones that promote sparsity as a way to achieve model simplicity) are of great interest.

Another consideration is finite sample error of the empirical choice probabilities. Specifically, since empirical choice probabilities are collected from a finite number of observations, it may be impossible to fit a choice model that is able to recover these probabilities exactly. This issue has already been encountered in Farias et al. [62], in which it was mitigated with an uncertainty set approach. Because of their focus on distance measure minimization, the approaches of [147, 27, 86], do not explicitly face such an "exact fit" issue. Nevertheless, when a distance measure minimization approach is taken, such as these latter cases, the finite sample error issue results in the phenomena that it may not be possible to have the objective converge to zero without utilizing additional data.

Finally, the techniques for non-parametric choice model estimation thus far are not designed to work with the dynamic data, i.e., exploit the possibility of continuously updating the empirical choice probabilities as more observations are collected. This setup is not only very realistic with today's data collection capabilities, but also critical in terms of properly addressing the finite sample error issue. As the empirical choice probabilities are updated with more observations, (under mild regularity assumptions) the finite sample error goes to zero, making it possible to fit a choice model for which the model-based choice probabilities are closer (in distance) to the empirical ones. A naïve way to work with dynamic data is to simply re-solve the estimation problem each time we update the choice probabilities. However, in the case of non-parametric choice model estimation, the estimation problem is expensive to solve, thus such a naïve approach significantly compounds the existing computational challenges.

5.1.2 Contributions

We present two iterative convex optimization-based approaches for the non-parametric estimation of choice probabilities modeled as a general distance minimization problem. These are a Frank-Wolfe type approach, and an application of the primal-dual framework developed in Chapter 2. Our developments simultaneously address the aspects of sparsity and dynamic data that have not been studied so far in the existing literature. Specifically, we provide error bounds and convergence guarantees (on the number of iterations needed to achieve a certain estimation accuracy) for our methods, in both static and dynamic data settings, which in turn provide guarantees on the sparsity of our estimated choice model. Consequently, our results highlight a natural trade-off between desired estimation accuracy and model sparsity.

- We first examine a primal approach based on the Frank-Wolfe (F-W) algorithm, as suggested in Jagabathula and Rusmevichientong [86] for the static estimation problem. So far, the F-W algorithm (without formal convergence guarantees) has been the classical choice when seeking sparsity in choice model estimation. We show that for two classes of distance measures (based on norms and KL divergence) suggested previously in Bertsimas and Mišic [27], van Ryzin and Vulcano [147], and studied in Jagabathula and Rusmevichientong [86], the standard convergence analysis of the F-W algorithm (see Jaggi [88]) cannot be applied due to the unboundedness of an important quantity (the curvature constant). Nonetheless, using the smoothing technique of Beck and Teboulle [13], we derive explicit convergence guarantees for the F-W algorithm applied to a class of smoothed versions of the distance measures based on norms, complementing the existing literature. The F-W method has a natural extension to the dynamic data setting. We establish an error bound on this natural dynamic variant of the F-W algorithm, and show that the error bound goes to zero when the empirical choice probabilities converge at a certain rate.

- We also utilize our primal-dual framework from Chapter 2 in both the static and dynamic settings. From this, we derive another variant of the F-W algorithm in the dynamic setting, which enjoys the same convergence guarantees but does not rely on the data convergence rate assumption needed in the analysis of the natural dynamic F-W extension. Our primal-dual framework also allows us to derive dual-based alternative algorithms to F-W via different choices of regret minimization algorithms (Chapter 3), and we compare how these affect the convergence rates.

- We carry out a computational study to examine the performance of our methods, as well as

the choice of the distance measure D, for the estimation of non-parametric choice models, in both the static and the dynamic settings. We implemented three different algorithms: the natural F-W method (with smoothing), a dual mirror descent method, and a modified dual method (using the dual representation of the smoothed distance measure).

Our numerical results show that all of our three methods, and all distance measures D based on norms, are able to learn a non-parametric model with low out-of-sample error equally well. However, in terms of algorithm efficiency (number of iterations until termination), our dual approach without smoothing clearly outperforms the others. We also observe that the sparsity of the estimated model is closely correlated with algorithm efficiency, which supports the theoretical results for our methods. In the dynamic setting, the rate at which we collect new observations has a demonstrable impact on the algorithm efficiency, but not on out-of-sample error.

Our numerical results also indicate that in all of the performance metrics, the choice of distance measure D leads to only very small variations, except that in the dynamic setting, the dual method with smoothing exhibits improvement in sparsity and algorithm efficiency when the ℓ_∞-norm based distance measure is used. In our additional testing with the variations of the ground truth choice model in the static setting, we observe that these conclusions still remain valid.

Outline. In Section 5.2, we describe the non-parametric choice model, and the choice observations used to estimate it. In Section 5.3 we formally describe the choice model estimation problem. The F-W approach is described in Section 5.3.1, and the application of our primal-dual framework is in Section 5.3.2. In Section 5.4 we present the numerical results for our computational study. Appendix B.1 summarizes existing approaches, and Appendix B.2 contains supplementary numerical results.

Notation. For a positive integer $n \in \mathbb{N}$, we let $[n] = \{1, \ldots, n\}$, define $\Delta_n := \{x \in \mathbb{R}^n_+ : \sum_{i\in[n]} x_i = 1\}$ to be the standard simplex, and S_n to be the collection of rankings of the set $[n]$. We denote the vector in $\mathbb{R}^n$ whose entries are all equal to 1 by $\mathbf{1}_n$, and the identity matrix in $\mathbb{R}^{n\times n}$ by I_n. We refer to a collection of objects b_j, $j \in J$ by the notation $\{b_j\}_{j\in J}$. Throughout the paper, the subscript, e.g., y_t, z_t, f_t, is used to attribute items to the t-th time period or iteration. The subscript is used to denote coordinates of a vector or matrix, e.g., β_{ij}. Given vectors x and y, $\langle x, y\rangle$ corresponds to the usual inner product of x and y. Given a norm $\|\cdot\|$ on a Euclidean space $\mathbb{E}$ and a real number $a > 0$, we denote its dual norm by $\|x\|_* = \min_y\{\langle x, y\rangle : \|x\| \le 1\}$. For $q \in [1, \infty]$, $\|x\|_q$ denotes the ℓ_q norm of x. We let $\partial f(x)$ be the subdifferential of f taken at x. We abuse notation slightly by denoting $\nabla f(x)$ for both the gradient of function f at x if f is differentiable and a subgradient of f at x, even if f is not differentiable. If ϕ is of the form $\phi(x, y)$, then $\nabla_x\phi(x, y)$ denotes the subgradient of ϕ at x while keeping the other variables fixed at y. We denote the indicator function as $\mathbb{I}$, i.e., $\mathbb{I}(\mathcal{S}) = 1$ if statement $\mathcal{S}$ holds, and $\mathbb{I}(\mathcal{S}) = 0$ otherwise.

5.2 Model and Data

Our goal is to understand the preferences of a certain population over a set of n items, $[n] = \{1, \ldots, n\}$, by estimating a non-parametric choice model from choice observations. We first describe the model, and how to compute choice probabilities. A non-parametric choice model is described by a probability distribution $\lambda \in \Delta_{n!}$ over all rankings S_n of the items $[n]$. Given a ranking $\sigma \in S_n$, we think of $\lambda(\sigma)$ as the probability that a member of the population will rank the items according

to σ. Also, when a particular member with ranking σ is presented a subset of items $\mathcal{A} \subseteq [n]$, they will choose the highest σ-ranked item $i(\sigma, \mathcal{A}) = \arg\min_{i \in \mathcal{A}} \sigma(i)$. Thus, the choice probability of a random member of the population choosing an item $i \in \mathcal{A}$ when presented with a subset $\mathcal{A}$ is

$$\mathbb{P}_\lambda[i \mid \mathcal{A}] = \sum_{\sigma \in S_n(i,\mathcal{A})} \lambda(\sigma), \quad S_n(i, \mathcal{A}) = \{\sigma \in S_n : i \text{ is the highest } \sigma\text{-ranked item in } \mathcal{A}\}.$$

Our choice observation set can be described as a collection of K pairs $\left\{i^k, \mathcal{A}^k\right\}_{k=1}^K$, where $i^k \in \mathcal{A}^k$ is the item chosen when the subset of items $\mathcal{A}_k \subseteq [n]$ was presented. There are a finite number of possible subsets amongst the observations, we denote these by $\mathcal{A}_j$, $j \in [m] = \{1, \ldots, m\}$. We also denote $N := \sum_{j=1}^m |\mathcal{A}_j|$. In practice, the collection of possible subsets $\{\mathcal{A}_j\}_{j \in [m]}$ can be controlled. Indeed, structural properties of these can have an impact on a combinatorial subproblem which appears in all non-parametric choice model estimation methods (see Section 5.3.3). However, in this paper, we will take this collection as given; see Jagabathula and Rusmevichientong [86] for a study on how the structure of $\{\mathcal{A}_j\}_{j \in [m]}$ impacts the combinatorial subproblem. Based on this observation set, we define

$$q_{ij} := \frac{1}{K} \sum_{k=1}^K \mathbb{I}(i^k = i, \mathcal{A}^k = \mathcal{A}_j), \quad q_j := \frac{1}{K} \sum_{k=1}^K \mathbb{I}(\mathcal{A}^k = \mathcal{A}_j) \quad \text{and} \quad p_{ij} := \frac{q_{ij}}{q_j}. \tag{5.1}$$

In words, q_{ij} is the proportion of observations where assortment $\mathcal{A}_j$ was displayed and item i was chosen, q_j is the proportion of observations where assortment $\mathcal{A}_j$ was displayed, and p_{ij} is the proportion of consumers who chose item i *given that* assortment $\mathcal{A}_j$ was displayed. Indeed, p_{ij} are the empirical choice probabilities which we will use to tune the probability distribution λ. We denote the collection of these empirical choice probabilities as $p = \{p_{ij}\}_{i \in \mathcal{A}_j, j \in [m]} \in \mathbb{R}^N$.

Our goal is to tune λ so that the choice probabilities $\mathbb{P}_\lambda[i \mid \mathcal{A}_j]$ are close to the empirical probabilities p_{ij}. We now define some notation to succinctly describe $\mathbb{P}_\lambda[i \mid \mathcal{A}_j]$. For a given pair $i \in \mathcal{A}_j$ and ranking $\sigma \in S_n$, we define $a_{ij}(\sigma) = 1$ if i is the highest σ-ranked item in $\mathcal{A}_j$, and $a_{ij}(\sigma) = 0$ otherwise. We define a_{ij} to be the $n!$-dimensional binary vector $\{a_{ij}(\sigma)\}_{\sigma \in S_n}$. Then

$$\mathbb{P}_\lambda[i \mid \mathcal{A}_j] = \sum_{\sigma \in S_n} a_{ij}(\sigma)\lambda(\sigma) = \langle a_{ij}, \lambda \rangle.$$

We define A to be the binary matrix of dimension $N \times n!$ with rows $a_{ij}^\top$. Each column corresponds to a ranking, and we denote these by $a(\sigma) \in \{0,1\}^N$. Now the collection of choice probabilities $\{\mathbb{P}_\lambda[i \mid \mathcal{A}_j]\}_{i \in \mathcal{A}_j, j \in [m]}$ can be written succinctly as $A\lambda$. We denote the polytope of all possible choice probabilities on observed pairs $i \in \mathcal{A}_j$ consistent with some distribution λ as

$$X := \{A\lambda : \ \lambda \in \Delta_{n!}\} = \operatorname{conv}\left(\{a(\sigma) : \ \sigma \in S_n\}\right) \subseteq \mathbb{R}^N.$$

Our goal can now be stated informally as finding a point $x \in X$ which is close to p, and our choice model will be the weights λ such that $x = \sum_{\sigma \in S_n} \lambda(\sigma) a(\sigma)$.

Finally, we describe the dynamic data setting, where we obtain additional observations over time. We denote a point in time by $t \in \mathbb{N}$, and the number of observations collected by time t as $K(t) \in \mathbb{N}$, which is non-decreasing in t. The set of observations at time t is $\{i^k, \mathcal{A}^k\}_{k=1}^{K(t)}$. For simplicity, we assume that the collection of observed subsets $\{\mathcal{A}_j\}_{j \in [m]}$ remains the same over time. We can compute empirical probabilities $p_{ij,t}$ at time t by using the observation set $\{i^k, \mathcal{A}^k\}_{k=1}^{K(t)}$

and formulas (5.1), and denote $p_t = \{p_{ij,t}\}_{i\in\mathcal{A}_j, j\in[m]} \in \mathbb{R}^N$. We will assume that the appropriate statistical regularity conditions hold so that $p_t \to p$ for some $p \in \mathbb{R}^N$. (If the observations are generated via some 'true model' λ^*, then $p \in X$, however our methods will not require this.) In the dynamic data setting, our goal will still be to tune λ so that $A\lambda$ is close to p, but only with access to the sequence $\{p_t\}_{t\geq 1}$.

5.3 Dynamic Estimation of a Non-Parametric Choice Model

We estimate a non-parametric choice model λ by solving

$$\begin{aligned} &\min_{\lambda} \{D(A\lambda, p) : \ \lambda \in \Delta_{n!}\} = \min_{x} \{D(x, p) : \ x \in X\} \\ &\text{given} \quad \{p_t\}_{t\geq 1}, \quad p_t \to p. \end{aligned} \tag{5.2}$$

where $D(x, p)$ is some distance measure which is convex in x for any fixed p. This is exactly an instance of a joint estimation and optimization (JEO) problem described in Chapter 1.4, (1.5)–(1.6). We assume that $D(x, x) = 0$. This is the case for all approaches from the existing literature (we give an overview of these in Appendix B.1). The main challenge is that, in general, the set X only admits a high-dimensional representation as $X = \{x = A\lambda : \ \lambda \in \Delta_{n!}\}$. Thus, some care must be taken in applying our primal-dual framework of Chapter 2.

The high-dimensional representation of X makes projection-based optimization methods or interior point methods challenging. On the other hand, *linear* optimization over X (which will be discussed in more detail in Section 5.3.3), while non-trivial, is a manageable problem. This naturally points to the Frank-Wolfe (F-W) algorithm, as suggested by Jagabathula and Rusmevichientong [86]. However, some care must be taken when using F-W to solve (5.2) and obtaining convergence rates, because not only the underlying assumptions must be checked, but also that we now have dynamic data $p_t \to p$. We show that a naïve adaptation of F-W in Section 5.3.1 exhibits a dependence on the rate of data convergence. In contrast, applying the primal-dual framework of Chapter 2 to solve (5.2), outlines in Section 5.3.2, circumvents some of the deficiencies arising from the naïve F-W analysis. This approach also constructs a solution $\bar{x} \in X$ by solving a sequence of linear optimizations over X. We compare how both approaches perform computationally in Section 5.4.

Before continuing, we discuss how both approaches give us a sparse choice model λ, and how convergence guarantees on the methods in turn give sparsity guarantees on λ. Since $X = \text{conv}(\{a(\sigma) : \sigma \in S_n\})$ is a polytope, a linear optimization oracle returns a single vertex $a(\sigma)$. Thus, any update to the solution $\bar{x}$ with a single linear optimization can only increase the support of λ by at most one. Given that our problem is of the form of a general distance measure minimization as opposed to a feasibility variant, the initial point in our algorithms can be taken to be a vertex of X. Therefore, if we have a guarantee on the number of linear optimizations $T(\epsilon)$ required to build a solution such that $D(\bar{x}, p) - \min_{x\in X} D(x, p) \leq \epsilon$, then the support of λ also contains at most $T(\epsilon)$ rankings. This is particularly useful when $T(\epsilon)$ is much lower than $n!$, which is often the case.

5.3.1 A Naïve Application of the Frank-Wolfe Algorithm

We first discuss solving (5.2) via the Frank-Wolfe (F-W) algorithm under a finite curvature constant assumption. In the static data setting, Jaggi [88] derives the standard F-W error bound of $O(1/T)$ after T iterations. However, this bound is explicitly based on the finiteness of an important quantity referred to as the *curvature constant* (5.3). We show that when D is an ℓ_q-norm or the weighted

KL-divergence, the curvature constant is infinite; so the error bounds from the usual F-W analysis Jaggi [88] cannot apply. To circumvent this, we employ smoothing techniques to guarantee a rate of $O(1/\sqrt{T})$ convergence for important distance measures D commonly used in non-parametric choice estimation. Moreover, in the dynamic data setting, we establish that a natural variant of the F-W algorithm also enjoys convergence guarantees under an additional minimum data convergence rate assumption.

In addition to our general convexity assumption on $D(\cdot, p)$, our primal approach is based on the following regularity condition, which can be viewed as a generalized triangle inequality.

Assumption 5.1. There exists some non-negative continuous function g_D such that $g_D(p, p) = 0$ and $D(x, p) - D(x, p') \leq g_D(p', p)$ for all x, p, p'.

Following Jaggi [88], we define the curvature constant critical in analyzing F-W methods and assume it is finite.

Assumption 5.2. The *curvature constant* of D defined below is finite and uniformly bounded in p_t:

$$C_{D,t} := \sup_{\substack{x,s \in X \\ \alpha \in [0,1]}} \frac{1}{\alpha^2} \left(D((1-\alpha)x + \alpha s, p_t) - D(x, p_t) - \alpha \langle s - x, \nabla_x D(x, p_t) \rangle \right) \tag{5.3}$$

and $C_{D,t} \leq C_D < \infty$ for all $t \geq 1$.

The variant of F-W algorithm for the dynamic setup is stated in Algorithm 4. The key difference of Algorithm 4 as opposed to the standard F-W algorithm is that each step t, Algorithm 4 works with a gradient of a dynamically changing $D(\cdot, p_t)$ instead of a fixed $D(\cdot, p)$. This setup is similar to the online F-W algorithms; see e.g., Hazan and Kale [74]. Note that Hazan and Kale [74] and other online algorithms are usually concerned with obtaining regret bounds. As opposed to this, our analysis of Algorithm 4 in the dynamic setting is not based on such regret bounds.

Algorithm 4 Frank-Wolfe (F-W) algorithm for solving (5.2).

Input: time horizon T, initial point $x^1 \in X$, step sizes $\{\gamma_t\}_{t \in [T]}$, $\gamma_t \in [0, 1]$.
for $t = 1, \ldots, T-1$ **do**
  $z_t \in \arg\min_{z \in X} \langle \nabla_x D(x_t, p_t), z \rangle$.
  $x_{t+1} = (1 - \gamma_t) x_t + \gamma_t z_t$.
end for
Output: solution $x_T \in X$.

Note in particular that in Algorithm 4, z_t is the result of a linear optimization over X, and the final x_T is a convex combination of previous $x^1, z^1, \ldots, z_T$, so when we select $x^1 \in X$ to be a vertex of X, the corresponding $\lambda_T \in \Delta_{n!}$ has support at most $T + 1$.

We next derive the convergence guarantee of Algorithm 4 in the dynamic setting under our assumptions.

Theorem 5.3. *Suppose that Assumptions 5.1 and 5.2 hold, and that Algorithm 4 runs for $T \geq 4$ iterations to find a point $x_T \in X$ with step sizes $\gamma_t = 2/(t+1)$. Then for any $x \in X$,*

$$\begin{aligned}
&D(x_T, p) - D(x, p) \\
&\leq \frac{4C_D}{T} + \frac{1}{(T-1)T} \sum_{t \in [T-3]} t(t+1) g_D(p_t, p) + \frac{1}{(T-1)T} \sum_{t \in [T-3]} \left((t-1)t + \frac{4t}{t+1} \right) g_D(p, p_t) \\
&\quad + \frac{T-2}{T} \left(\frac{T-3}{T-1} + \frac{4}{(T-1)^2} \right) g_D(p, p_{T-2}) + \left(\frac{T-2}{T} + \frac{4}{T^2} \right) g_D(p, p_{T-1}).
\end{aligned}$$

Proof of Theorem 5.3. From Assumption 5.2 applied to $D(\cdot, p_t)$ and $x_{t+1} = (1-\gamma_t)x_t + \gamma_t z_t$, we get

$$D(x_{t+1}, p_t) \leq D(x_t, p_t) + \gamma_t \langle \nabla_x D(x_t, p_t), z_t - x_t \rangle + (\gamma_t)^2 C_D.$$

From the definition of z_t and convexity of $D(\cdot, p_t)$, we have $\langle \nabla_x D(x_t, p_t), z_t - x_t \rangle \leq \langle \nabla_x D(x_t, p_t), x - x_t \rangle \leq D(x, p_t) - D(x_t, p_t)$ for any $x \in X$, so

$$D(x_{t+1}, p_t) \leq (1-\gamma_t) D(x_t, p_t) + \gamma_t D(x, p_t) + (\gamma_t)^2 C_D.$$

Adding and subtracting the appropriate terms, we arrive at

$$\begin{aligned}
D(x_{t+1}, p) - D(x, p) \leq\ & (1-\gamma_t)(D(x_t, p) - D(x, p)) + (\gamma_t)^2 C_D \\
& + D(x_{t+1}, p) - D(x_{t+1}, p_t) + (1-\gamma_t)(D(x_t, p_t) - D(x_t, p)) \\
& + \gamma_t (D(x, p_t) - D(x, p)).
\end{aligned}$$

Defining $\delta_t := D(x_t, p) - D(x, p)$ and $\alpha_t := (\gamma_t)^2 C_D + D(x_{t+1}, p) - D(x_{t+1}, p_t) + (1-\gamma_t)(D(x_t, p_t) - D(x_t, p)) + \gamma_t (D(x, p_t) - D(x, p))$, we now have the recursion $\delta_{t+1} \leq (1-\gamma_t)\delta_t + \alpha_t$.

By induction, for $t \geq 3$,

$$\delta_t \leq \delta_1 \prod_{k \in [t-1]} (1-\gamma_k) + \sum_{k \in [t-2]} \alpha_k \prod_{l=k+1}^{t-1} (1-\gamma_l) + \alpha_{t-1}.$$

Moreover, the first term disappears because $\gamma_k = 2/(k+1)$, $1 - \gamma_1 = 0$. Furthermore, $\prod_{l=k+1}^{t-1}(1-\gamma_l) = \frac{k(k+1)}{(t-1)t}$ for $k = 1, \ldots, t-3$ and $(t-2)/t$ for $k = t-2$, so

$$\delta_t \leq \frac{1}{(t-1)t} \sum_{k \in [t-3]} k(k+1)\alpha_k + \frac{t-2}{t}\alpha_{t-2} + \alpha_{t-1}.$$

Substituting the definitions of δ_t, α_t and simplifying, we get

$$\begin{aligned}
&D(x_T, p) - D(x, p) \\
&\leq \frac{4C_D}{T} + \frac{1}{(T-1)T} \sum_{t \in [T-3]} t(t+1) \left(D(x_{t+1}, p) - D(x_{t+1}, p_t) \right) \\
&\quad + \frac{1}{(T-1)T} \left(\sum_{t \in [T-3]} (t-1)t \left(D(x_t, p_t) - D(x_t, p) \right) + \sum_{t \in [T-3]} \frac{4t}{t+1} \left(D(x, p_t) - D(x, p) \right) \right) \\
&\quad + \frac{T-2}{T} \left(\frac{T-3}{T-1} (D(x_{T-2}, p_{T-2}) - D(x_{T-2}, p)) + \frac{4}{(T-1)^2} (D(x, p_{T-2}) - D(x, p)) \right) \\
&\quad + \frac{T-2}{T} (D(x_{T-1}, p_{T-1}) - D(x_{T-1}, p)) + \frac{4}{T^2} (D(x, p_{T-1}) - D(x, p)).
\end{aligned}$$

Applying Assumption 5.1 completes the proof. □

Notice in Theorem 5.3 that the error bound has the $4C_D/T$ rate along with four error terms which are accumulated from using approximate data $p_t \approx p$. Indeed, if $p_t = p$ for all t, i.e., the static case, these additional error terms disappear, and we are left with only the $4C_D/T$ term. In the dynamic case, in order to get convergence from Theorem 5.3, these four error terms must go to 0 as $T \to \infty$. To ensure this, we must require that $g_D(p_t, p) \to 0$ sufficiently fast to guarantee that $\sum_{t\in[T-3]} t(t+1)g_D(p_t, p) = o(T^2)$. However, this is not always true: consider $g_D(p_t, p) \sim 1/t$. Approaches based on our primal-dual framework, outlined in Section 5.3.2, avoid such a requirement, and instead need only $D(p_t, p), D(p, p_t) \to 0$.

In order to apply Theorem 5.3, we need to verify the underlying assumptions for the chosen distance measure D. Two distance measures D used in the existing literature are the weighted KL-divergence (which stems from the maximum likelihood, and is studied in van Ryzin and Vulcano [147]) and ℓ_1-norm examined in Bertsimas and Mišic [27]. See B.1 for precise details on how the approaches in van Ryzin and Vulcano [147], Bertsimas and Mišic [27] correspond to particular choices of D in our framework. Unfortunately, for both of these choices of D, in Proposition 5.4 we demonstrate that the curvature constant (5.3), which is critical in the standard convergence analysis of F-W algorithm, is infinite. Thus, we cannot rely on the existing standard convergence analysis of the classical F-W algorithm in the static case or the application of Theorem 5.3 in the dynamic case when D is selected to be either an ℓ_q-norm (as in Bertsimas and Mišic [27]) or the weighted KL-divergence (as in van Ryzin and Vulcano [147]).

Proposition 5.4. *Suppose $n > 2$. For any $q \in [1, \infty]$, the function $D(x, p) = \|x - p\|_q$ has infinite curvature constant (5.3) for any $p \in X$. Furthermore, when $D(x, p) = \sum_{j\in[m]} w_j \operatorname{KL}(p_j, x_j)$ for any positive weights w_j, and $p \in X$ such that $p_{ij} > 0$ for all $i \in \mathcal{A}_j$, the curvature constant is infinite.*

Proof of Proposition 5.4. We will first show that the curvature constant C_D defined in (5.3) of $D(x, p) = \|x - p\|$ is infinite for any $p \in X$. Let us choose $x = p$, reserving the choice of $\alpha \in [0, 1]$ and $s \in X$ for later. Then $D(x, p) = 0$, $D((1-\alpha)x + \alpha s, p) = \alpha\|s - p\|$, and the subgradients of $D(x, p)$ are $\{y : \|y\|_* \le 1\}$. Thus, for any selection of subgradient mapping $y(\hat{x}) \in \nabla_x D(\hat{x}, p)$ we have

$$\begin{aligned}\frac{1}{\alpha^2}\Big[D((1-\alpha)x + \alpha s, p) - D(x, p) - \alpha\langle s - x, y(x)\rangle\Big] &= \frac{1}{\alpha^2}\Big[\alpha\|s - p\| - \alpha\langle s - p, y(x)\rangle\Big] \\ &= \frac{1}{\alpha}\Big[\|s - p\| - \langle s - p, y(x)\rangle\Big].\end{aligned}$$

Note that whenever there is a choice $s \in X$ with $\|s - p\| - \langle s - p, y(p)\rangle > 0$, we can send $\alpha \to 0$ and conclude that the curvature constant C_D is infinite.

To choose the appropriate s, we denote the set of subgradients of $\|\cdot\|_q$ at $s - p$ as $G_{\|\cdot\|}(s - p)$. Observe that for a norm $\|\cdot\|$, if $y \in G_{\|\cdot\|}(s - p)$ then $\|y\|_* \le 1$ and $\langle s - p, y\rangle = \|s - p\|$. Thus, we need to choose $s \in X$ such that $y(x) \notin G_{\|\cdot\|}(s - p)$. To do this, we exploit the following property of ℓ_q norms. It is simple to check that for $q \in [1, \infty]$ and $y \in G_{\|\cdot\|_q}(s - p)$, we have the property that $y_{ij} > 0 \implies s_{ij} - p_{ij} > 0$. For our selection $y(x)$, first suppose that there exists $i \in \mathcal{A}_j$ such that $y(x)_{ij} > 0$. Then a ranking σ that ranks i last will have $a(\sigma)_{ij} = 0$, so $a(\sigma)_{ij} - p_{ij} \le 0$ because $p_{ij} \ge 0$. We cannot have $p = a(\sigma)$ for all $(n-1)!$ rankings σ that ranks i last (note that $n > 2$);

hence, there exists one σ such that $a(\sigma) \neq p$, and we choose $s = a(\sigma)$. This implies that $y(x)_{ij} > 0$ while $s_{ij} - p_{ij} \leq 0$, hence $y(x) \notin G_{\|\cdot\|_q}(s-p)$. Now suppose that $y(x)_{ij} \leq 0$ for all item-subset pairs (i,j). If $y(x) = 0$, then the result follows trivially by choosing any $s \neq p$. Suppose now there exists some $y(x) < 0$. It is again simple to check that for $q \in [1,\infty]$ and $y \in G_{\|\cdot\|_q}(s-p)$, we have the property that $y_{ij} < 0 \implies s_{ij} - p_{ij} < 0$. Then a ranking σ that ranks i first will have $a(\sigma)_{ij} = 1$, so $a(\sigma)_{ij} - p_{ij} \geq 0$ because $p_{ij} \leq 1$. We cannot have $p = a(\sigma)$ for all $(n-1)!$ rankings σ that ranks i first, so there exists one such that $a(\sigma) \neq p$, and we choose $s = a(\sigma)$. This implies that $y(x)_{ij} < 0$ while $s_{ij} - p_{ij} \geq 0$, hence $y(x) \notin G_{\|\cdot\|_q}(s-p)$. Thus, in all cases for $y(x)$, we can choose the appropriate $s \in X$.

Now consider the weighted KL-divergence $D(x,p) = -\sum_{j\in[m]} w_j \sum_{i\in\mathcal{A}_j} p_{ij} \log(x_{ij}/p_{ij})$. We can assume that $p_{ij} > 0$ by simply ignoring terms in the sum for which $p_{ij} = 0$. Choose $x = p$, which ensures that $D(\cdot,p)$ is differentiable at x with $\nabla_x D(x,p)_{ij} = -w_j/x_{ij}$. Then we have

$$\begin{aligned}
&\frac{1}{\alpha^2}\Big[D((1-\alpha)x + \alpha s, p) - D(x,p) - \alpha\langle s - x, \nabla_x D(x,p)\rangle\Big] \\
&= -\frac{1}{\alpha^2}\sum_{j\in[m]} w_j \sum_{i\in\mathcal{A}_j} p_{ij} \log\left(1 - \alpha + \alpha\frac{s_{ij}}{p_{ij}}\right) + \frac{1}{\alpha}\sum_{j\in[m]} w_j \sum_{i\in\mathcal{A}_j}\left(\frac{s_{ij}}{p_{ij}} - 1\right).
\end{aligned}$$

Note that the second term is bounded by $\frac{1}{\alpha}\left(\sum_{j\in[m]} w_j\right)(\max_{i,j} 1/p_{ij} - 1)$. Choose $s_{ij} = a(\sigma)$ for any $\sigma \in S_n$. Then there exists some i,j such that $s_{ij} = 0$. Sending $\alpha \to 1$ results in $\log\left(1 - \alpha + \alpha\frac{s_{ij}}{p_{ij}}\right) \to \infty$, and the second term is bounded, so the curvature constant C_D is infinite. □

Proposition 5.4 implies that Theorem 5.3 cannot be directly applied to the classical distance measures D used in non-parametric choice modeling. In order to develop a F-W based framework, we instead employ a smooth approximation to these distance measures D.

Definition 5.5. A differentiable (on the relative interior of its domain) convex function $h : X \to \mathbb{R} \cup \{\infty\}$ is *L-smooth* with respect to a norm $\|\cdot\|$ if, for all $s, x \in \operatorname{relint}(X)$, we have

$$h(s) \leq h(x) + \langle \nabla h(x), s - x\rangle + \frac{L}{2}\|s - x\|^2.$$

A function $h_\alpha : X \to \mathbb{R}$ is a *(L/α)-smooth approximation* to h if it is (L/α)-smooth and there exists constants $\beta_1, \beta_2 \geq 0$ such that for all $x \in X$,

$$h(x) - \beta_1\alpha \leq h_\alpha(x) \leq h(x) + \beta_2\alpha.$$

Proposition 5.6. *For $\alpha > 0$, let $D_\alpha(x,p)$ be a (L/α)-smooth approximation with respect to the norm $\|\cdot\|$ of $D(x,p)$ such that $D(x,p) - \beta_1\alpha \leq D_\alpha(x,p) \leq D(x,p) + \beta_2\alpha$ for all $x \in X$. Denote $R_{X,\|\cdot\|} := \max_{x,x'\in X}\|x - x'\|$. Then, under Assumption 5.1, Algorithm 4 applied to D_α after $T \geq 4$*

iterations with step sizes $\gamma_t = 2/(t+1)$ *results in*

$$\begin{aligned}
&D(x_T, p) - D(x, p) \\
&\leq \frac{4LR^2_{X,\|\cdot\|}}{\alpha\, T} + (\beta_1 + \beta_2)\alpha \\
&\quad + \frac{1}{(T-1)T} \sum_{t\in[T-3]} t(t+1) g_{D_\alpha}(p_t, p) + \frac{1}{(T-1)T} \sum_{t\in[T-3]} \left((t-1)t + \frac{4t}{t+1}\right) g_{D_\alpha}(p, p_t) \\
&\quad + \frac{T-2}{T}\left(\frac{T-3}{T-1} + \frac{4}{(T-1)^2}\right) g_{D_\alpha}(p, p_{T-2}) + \left(\frac{T-2}{T} + \frac{4}{T^2}\right) g_{D_\alpha}(p, p_{T-1}).
\end{aligned}$$

Proof. Since $D(x,p) \leq D_\alpha(x,p) + \beta_1\alpha$ and $-D(x',p) \leq -D_\alpha(x',p) + \beta_2\alpha$, we have $D(x,p) - D(x',p) \leq D_\alpha(x,p) - D_\alpha(x',p) + (\beta_1+\beta_2)\alpha$. Since D_α is L/α-smooth with respect to $\|\cdot\|$, from Jaggi [88, Appendix D] we deduce that the curvature constant of D_α is $C_{D_\alpha} \leq LR^2_{X,\|\cdot\|}/\alpha$, thus Algorithm 4 applied to D_α has convergence guarantee given by Theorem 5.3. Combining this with the approximation guarantee, we get our result. □

Remark 5.7. The norm $\|\cdot\|$, the constants $L, \beta_1, \beta_2, R_{X,\|\cdot\|}$, and the function g_{D_α} will vary depending on the smoothing technique chosen. However, in most situations, we can control the smoothness parameter α, and for several important examples outlined below, g_{D_α} is independent of α. We thus choose α to minimize $\frac{4LR^2_{X,\|\cdot\|}}{\alpha\, T} + (\beta_1+\beta_2)\alpha$, which results in $\alpha^* = \sqrt{\frac{4LR^2_{X,\|\cdot\|}}{(\beta_1+\beta_2)T}}$, and

$$\frac{4LR^2_{X,\|\cdot\|}}{\alpha^*\, T} + (\beta_1+\beta_2)\alpha^* = 4R_{X,\|\cdot\|}\sqrt{\frac{L(\beta_1+\beta_2)}{T}}.$$

∎

In Table 5.1 we present some choices for smoothing $D(x,p) = \|x-p\|_q$ for $q = 1, 2, \infty$ from Beck and Teboulle [13, Examples 4.2, 4.1, 4.5], where H_α denotes the Huber function

$$H_\alpha(r) := \begin{cases} \frac{1}{2\alpha}|r|^2, & |r| \leq \alpha \\ |r| - \frac{\alpha}{2}, & |r| > \alpha. \end{cases}$$

D_q	$D_{q,\alpha}$	smooth w.r.t.	L	$R_{X,\|\cdot\|}$	$\beta_1+\beta_2$	α	gap
ℓ_1	$\sum_{j\in[m]}\sum_{i\in\mathcal{A}_j} H_\alpha(x_{ij}-p_{ij})$	$\|\cdot\|_2$	1	$\sqrt{2m}$	$\frac{N}{2}$	$4\sqrt{\frac{m}{NT}}$	$4\sqrt{\frac{Nm}{T}}$
ℓ_2	$H_\alpha(\|x-p\|_2)$	$\|\cdot\|_2$	1	$\sqrt{2m}$	$\frac{1}{2}$	$4\sqrt{\frac{m}{T}}$	$4\sqrt{\frac{m}{T}}$
ℓ_∞	$\alpha\log\left(\sum_{j\in[m]}\sum_{i\in\mathcal{A}_j} 2\cosh\left(\frac{x_{ij}-p_{ij}}{\alpha}\right)\right)$	$\|\cdot\|_\infty$	1	2	$\log(2N)$	$\frac{4}{\sqrt{\log(2N)T}}$	$8\sqrt{\frac{\log(2N)}{T}}$

Table 5.1: Summary of the F-W approach to solve (5.2).

We can derive these smooth approximations via the so-called 'Nesterov smoothing' technique, i.e.,

$$D_{1,\alpha}(x,p) = \max_{y:\|y\|_\infty \le 1} \{\langle x-p, y\rangle - \alpha\omega_1(y)\}, \quad \omega_1(y) = \frac{1}{2}\|y\|_2^2$$
$$D_{2,\alpha}(x,p) = \max_{y:\|y\|_2 \le 1} \{\langle x-p, y\rangle - \alpha\omega_2(y)\}, \quad \omega_2(y) = \frac{1}{2}\|y\|_2^2 \tag{5.4}$$
$$D_{\infty,\alpha}(x,p) = \max_{y\in\Delta_{2N}} \{\langle B(x-p), y\rangle - \alpha\omega_\infty(y)\}, \quad \omega_\infty(y) = \sum_{k=1}^{2N} y_k \log(y_k), \quad B = \begin{bmatrix} I_N \\ -I_N \end{bmatrix}.$$

Here, when $\alpha = 0$, the original ℓ_q-norms are recovered. We make use of this representation in our primal-dual approach described in Section 5.3.2.

The smoothness norm and constants L and $\beta_1 + \beta_2$ can be found in the respective examples from Beck and Teboulle [13]. The constant $R_{X,\|\cdot\|}$ is derived by using the fact that, for any $x \in X$, we have the property that $x \ge 0$ and $\sum_{i\in\mathcal{A}_j} x_{ij} = 1$. The parameter α and optimality gap are computed using Remark 5.7. Finally, we demonstrate next that Assumption 5.1 holds for all such choices smoothing functions $D_{q,\alpha}$ presented in Table 5.1.

Remark 5.8. For $q \in \{1, 2, \infty\}$, we have $D_{q,\alpha}(x,p) - D_{q,\alpha}(x,' p) \le \|p-p'\|_q$ for all x, p, p' and $\alpha \ge 0$.

We first examine $q = 1$. Note that H_α is convex $\alpha \ge 0$, so $H_\alpha(r) - H_\alpha(r') \le \nabla_r H_\alpha(r)\cdot(r-r') \le |\nabla_r H_\alpha(r)|\,|r-r'| \le |r-r'|$ since $|\nabla_r H_\alpha(r)| \le 1$. Therefore, for any fixed x, $D_{1,\alpha}(x,p) - D_{1,\alpha}(x,p_t) \le \sum_{j\in[m]}\sum_{i\in\mathcal{A}_j} |p_{ij} - p_{ij,t}| = \|p - p_t\|_1$, hence $g_{D_{1,\alpha}}(p,p') = \|p-p'\|_1$.

For $q = 2$, we again use $H_\alpha(r) - H_\alpha(r') \le |r-r'|$, so that $D_{2,\alpha}(x,p) - D_{2,\alpha}(x,p_t) \le |\|x-p\|_2 - \|x-p_t\|_2| \le \|p-p_t\|_2$, hence $g_{D_{2,\alpha}}(p,p') = \|p-p'\|_2$.

For $q = \infty$, let $h_\alpha(z) := \alpha\log\left(\sum_k 2\cosh(z_k/\alpha)\right)$, which is convex in z. Hence, $h_\alpha(z) - h_\alpha(z') \le \langle \nabla_z h_\alpha(z), z - z'\rangle \le \|\nabla_z h_\alpha(z)\|_1 \|z - z'\|_\infty \le \|z - z'\|_\infty$ since $\nabla_z h_\alpha(z)_k = \sinh(z_k/\alpha)/\left(\sum_{k'}\cosh(z_{k'}/\alpha)\right)$ thus $\|\nabla_z h_\alpha(z)\|_1 \le 1$. Finally, since $D_{\infty,\alpha}(x,p) = h_\alpha(x-p)$, we have $D_{\infty,\alpha}(x,p) - D_{\infty,\alpha}(x,p') \le \|p-p'\|_\infty$ for any $x \in X$, so $g_{D_{\infty,\alpha}}(p,p') = \|p-p'\|_\infty$. ∎

Remark 5.9. The three rates $4\sqrt{Nm/T}, 4\sqrt{m/T}, 8\sqrt{\log(2N)/T}$ in Table 5.1 all have the same dependence on T, which implies that when D is an ℓ_q-norm we can get an ϵ-optimal solution to (5.2) after $O(1/\epsilon^2)$ iterations using the F-W algorithm. However, the dependence on the problem parameters m, N vary significantly. We see that the best dependence of $\sqrt{\log(2N)}$ is given by the ℓ_∞-norm, followed by $\sqrt{m}$ given by the ℓ_2-norm, and finally $\sqrt{Nm}$ for the ℓ_1-norm. This intuitively makes sense since $\|z\|_\infty \le \|z\|_2 \le \|z\|_1$ for any fixed z. ∎

5.3.2 Applying the Primal-Dual Framework

A major handicap of the F-W algorithm in Theorem 5.3 is that, in the dynamic case, the overall optimality gap converges to 0 only if $p_t \to p$ sufficiently fast. In this section, we remove such a condition completely by utilizing our primal-dual framework of Chapter 2.5.

Suppose that D admits a representation of the form

$$D(x,p) = \max_{y\in Y} \{\langle Bx, y\rangle - \phi(y,p)\} \tag{5.5}$$

for some convex set Y and matrix B of appropriate dimension, where $\phi(y,p)$ is convex in y for any p. Note that when $D(\cdot,p)$ is convex and continuous on its domain, such a representation always exists via the convex conjugate:

$$D(x,p) = \sup_y \left\{\langle x,y\rangle - D^*(y,p) :\ y \in \mathbb{R}^N\right\}, \quad \text{where } D^*(y,p) = \sup_z \left\{\langle y,z\rangle - D(z,p) :\ z \in \mathbb{R}^N\right\}.$$

Denote

$$\Psi(x, y; p) := \langle Bx, y\rangle - \phi(y, p).$$

As in (2.29), given p, we know that to bound the optimality gap of a point $\bar{x} \in X$, we need only find a dual certificate $\bar{y} \in Y$ and evaluate the saddle point (SP) gap:

$$D(\bar{x}, p) - \min_{x\in X} D(x, p) \le \max_{y\in Y} \Psi(\bar{x}, y; p) - \min_{x\in X} \Psi(x, \bar{y}; p). \tag{5.6}$$

Now, since we do not have access to p directly, but instead only gain knowledge of p through a sequence $p_t \to p$, each time we receive a new p_t, we will generate a new primal-dual pair x_t, y_t, for $t \ge 1$. Aggregating these with non-negative weights θ_t for each time $t \in [T]$, and setting $\Theta_T = \sum_{t\in[T]} \theta_t$ according to Theorem 2.4 and (5.6), the optimality gap of $\bar{x}_T^\theta = \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t x_t$ can be certified with the corresponding dual point $\bar{y}_T^\theta = \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t y_t$:

$$D(\bar{x}_T^\theta, p) - \min_{x\in X} D(x, p) \le \widehat{\epsilon}\left(\{x_t, y_t, p_t, \theta_t\}_{t\in[T]}\right) + \epsilon^\circ\left(\{p_t, \theta_t\}_{t\in[T]}; p\right) + \widehat{\epsilon}\left(\{y_t, p_t, \theta_t\}_{t\in[T]}; p\right),$$

where

$$\widehat{\epsilon}\left(\{x_t, y_t, p_t, \theta_t\}_{t\in[T]}\right) = \max_{x\in X, y\in Y} \frac{1}{\Theta_T}\sum_{t\in[T]} \left[\langle Bx_t, y\rangle - \phi(y, p_t) - \langle Bx, y_t\rangle + \phi(y_t, p_t)\right]$$

$$\epsilon^\circ\left(\{p_t, \theta_t\}_{t\in[T]}; p\right) = \max_{y\in Y} \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t \left[\phi(y, p) - \phi(y, p_t)\right]$$

$$\epsilon^\bullet\left(\{y_t, p_t, \theta_t\}_{t\in[T]}; p\right) = \frac{1}{\Theta_T}\sum_{t\in[T]} \theta_t \left[\phi(y_t, p_t) - \phi(y_t, p)\right].$$

As discussed in Chapter 2.3, we will use tools from online convex optimization to choose the primal-dual sequence $\{x_t, y_t\}_{t\in[T]}$ to guarantee that $\widehat{\epsilon} \to 0$. We will discuss this below, but first, we discuss the error term $\epsilon^\circ + \epsilon^\bullet$. In order to guarantee that $\epsilon^\circ + \epsilon^\bullet \to 0$, we assume the following regularity conditions on D. Note that these are simply restating Condition 2.18 from Chapter 2.5, except that the norm $\|p - p'\|$ is replaced by a continuous function $g(p - p')$.

Assumption 5.10. The convex set Y is compact, and there exists some continuous function $g : \mathbb{R}^N \to \mathbb{R}$, $\phi(y, p) - \phi(y, p') \le g(p - p')$ for all $y \in Y$ and for all p, p'.

Theorem 2.19 then certifies that when Assumption 5.10 holds, $\epsilon^\circ + \epsilon^\bullet \to 0$, while Proposition 2.20 gives rates which depend on the rate at which $g(p_t - p) \to 0$.

In the following, we will consider an important class of distance functions D: when $D(x, p) = \|x - p\|$ for some norm $\|\cdot\|$, or an appropriate smoothed version of the norm. Generalizing (5.4), these are of the form

$$D(x, p) = \max_{y\in Y} \left\{\langle B(x - p), y\rangle - \alpha\omega(y)\right\},$$

for appropriately chosen B, $\alpha \ge 0$ and ω. Note that the norm is recovered when $Y = \{y : \|y\|_* \le 1\}$ is the dual norm ball, $B = I_N$ and $\alpha = 0$. Clearly, these are all of the same form as (5.5) with $\phi(y, p) = \langle Bp, y\rangle + \alpha\omega(y)$. Henceforth, we assume that ϕ has this form.

Assumption 5.11. For some $\alpha \ge 0$ and 1-strongly convex function $\omega : Y \to \mathbb{R}$,

$$D(x, p) = \max_{y\in Y} \left\{\langle Bx, y\rangle - \phi(y, p)\right\}, \quad \phi(y, p) = \langle p, y\rangle + \alpha\omega(y).$$

We now show that Assumption 5.10 is satisfied, and hence $\epsilon^\circ + \epsilon^\bullet \to 0$.

Lemma 5.12. *Let Y be a compact set. Given $\alpha \geq 0$, consider*

$$D(x,p) = \max_{y \in Y} \left\{ \langle Bx, y \rangle - \phi(y,p) \right\}, \quad \phi(y,p) = \langle Bp, y \rangle + \alpha\omega(y).$$

Then for any $y \in Y$ and p, p',

$$\leq \underline{g}(p-p') \leq \phi(y,p) - \phi(y,p') \leq g(p-p'), \quad g(p) := \max_{y' \in Y} \langle Bp, y' \rangle, \quad \underline{g}(p) := \min_{y' \in Y} \langle Bp, y' \rangle.$$

Furthermore, $g, \underline{g}$ are continuous functions. Therefore Assumption 5.10 is satisfied.

Proof. Observe that $\phi(y,p) - \phi(y,p') = \langle B(p-p'), y \rangle \leq \max_{y' \in Y} \langle B(p-p'), y' \rangle = g(p-p')$ for any $y \in Y$ and p, p'. The continuity of g follows since $g(p)$ is convex in p and is finite valued for all p since Y is compact. Similarly, $\underline{g}$ is concave and finite-valued. □

Corollary 5.13. *When Y is compact and $\phi(y,p) = \langle Bp, y \rangle + \alpha\omega(y)$ for some matrix B and $\alpha \geq 0$,*

$$\lim_{T \to \infty} \left[\epsilon^\circ \left(\{p_t, \theta_t\}_{t \in [T]}; p \right) + \epsilon^\bullet \left(\{y_t, p_t, \theta_t\}_{t \in [T]}; p \right) \right] = 0$$

whenever $p_t \to p$.

Proof. This is a straightforward adaptation of Theorem 2.19, noting that $g(p_t - p) \to 0$ whenever $p_t \to p$ due to continuity, except that the equality of the limit instead of inequality in Theorem 2.19 stems from having both upper and lower bounds on $\phi(y,p) - \phi(y,p')$. □

We now turn our attention to bounding $\widehat{\epsilon}\left(\{x_t, y_t, p_t, \theta_t\}_{t \in [T]} \right)$. As mentioned in Chapter 3.7, this is an online SP gap term (3.3) where $\Psi_t(\cdot,\cdot) = \Psi(\cdot,\cdot;p_t)$. Recall, also, that there are a variety of ways outlined in Chapter 3 to compute the primal-dual sequence $\{x_t, y_t\}_{t \geq 1}$ which bound $\widehat{\epsilon}$. One option is to employ algorithms that directly bound it (i.e., Theorems 3.7 and 3.28); another is to recognize that $\widehat{\epsilon}$ decomposes into two regret terms

$$\begin{aligned} \widehat{\epsilon}\left(\{x_t, y_t, p_t, \theta_t\}_{t \in [T]} \right) = &\max_{y \in Y} \frac{1}{\Theta_T} \sum_{t \in [T]} \theta_t \left[\langle Bx_t, y \rangle - \phi(y, p_t) \right] - \frac{1}{\Theta_T} \sum_{t \in [T]} \theta_t \left[\langle Bx_t, y_t \rangle - \phi(y_t, p_t) \right] \\ &+ \frac{1}{\Theta_T} \sum_{t \in [T]} \theta_t \left[\langle Bx_t, y_t \rangle - \phi(y_t, p_t) \right] - \min_{x \in X} \frac{1}{\Theta_T} \sum_{t \in [T]} \theta_t \left[\langle Bx, y_t \rangle - \phi(y_t, p_t) \right], \end{aligned}$$

then employ two regret minimizing algorithms (Theorems 3.4, 3.12, 3.16, 3.21, 3.25), one for computing $\{x_t\}_{t \in [T]}$, and the other for computing $\{x_t\}_{t \in [T]}$.

However, in our particular application of choice model estimation, the domain X in general only admits a high dimensional representation as $X = \{A\lambda : \lambda \in \Delta_{n!}\}$. Therefore, projection-type algorithms are intractable to implement in practice, which excludes using the above-mentioned direct methods of Theorems 3.7 and 3.28. On the other hand, *linear* optimization over X is tractable in practice (as we will discuss in Section 5.3.3), thus the only avenue left to us is to choose

$$x_t \in \arg\min_{x \in X} \langle Bx, y_t \rangle = \arg\min_{x \in X} \left\{ \langle Bx - p_t, y_t \rangle - \alpha\omega(y_t) \right\} = \arg\min_{x \in X} \left\{ \langle Bx, y_t \rangle - \phi(y_t, p_t) \right\}. \quad (5.7)$$

	$D_{1,\alpha}$	$D_{2,\alpha}$	$D_{\infty,\alpha}$
Y	ℓ_∞-ball	ℓ_2-ball	Δ_{2N}
B	I_N	I_N	$\begin{bmatrix} I_N \\ -I_N \end{bmatrix}$
$\omega(y)$	$\frac{1}{2}\|y\|_2^2$	$\frac{1}{2}\|y\|_2^2$	$\sum_{k\in[2N]} y_k \log(y_k)$
form	$\sum_{j\in[m],i\in\mathcal{A}_j} H_\alpha(x_{ij}-p_{ij})$	$H_\alpha(\|x-p\|_2)$	$\alpha\log\left(\sum_{j\in[m],i\in\mathcal{A}_j} 2\cosh\left(\frac{x_{ij}-p_{ij}}{\alpha}\right)\right)$
$\|\cdot\|$	$\|\cdot\|_2$	$\|\cdot\|_2$	$\|\cdot\|_1$
Ω	$N/2$	$1/2$	$\log(2N)$
Ω'	0	0	0
G	$\sqrt{2m+\alpha N}$	$\sqrt{2m+\alpha}$	(See Theorem 5.18)

Table 5.2: Summary of distance functions defined via $D(x,p) = \max_{y\in Y}\{\langle B(x-p), y\rangle - \alpha\omega(y)\}$, their prox-setups for the dual domain Y, and relevant constants.

This immediately (and trivially) guarantees that for any dual sequence $\{y_t\}_{t\in[T]}$, the primal regret term is bounded:

$$\frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t\left[\langle Bx_t, y_t\rangle - \phi(y_t,p_t)\right] - \min_{x\in X}\frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t\left[\langle Bx, y_t\rangle - \phi(y_t,p_t)\right] \le 0.$$

Then, to bound $\widehat{\epsilon}$, it remains to choose the dual sequence $\{y_t\}_{t\in[T]}$ to bound the regret term

$$\begin{aligned}&\max_{y\in Y}\frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t\left[\langle Bx_t, y\rangle - \phi(y,p_t)\right] - \frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t\left[\langle Bx_t, y_t\rangle - \phi(y_t,p_t)\right]\\ &= \max_{y\in Y}\frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t\left[\langle B(x_t-p_t), y\rangle - \alpha\omega(y)\right] - \frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t\left[\langle B(x_t-p_t), y_t\rangle - \alpha\omega(y_t)\right].\end{aligned}$$

We can choose a regret minimizing algorithm, e.g. one of Theorems 3.4, 3.12, 3.16, 3.21, and use it to compute y_t. However, we cannot use Theorem 3.25, since implementing this requires knowledge of x_t to compute y_t, but by (5.7), we are computing x_t after y_t.

Our choice of regret minimizing algorithm determines the bound on $\widehat{\epsilon}$, which often involves a bound on the gradient $\|\nabla_y\Psi(x_t,y_t;p_t)\|_*$ and a bound on the set width term $\Omega := \max_{y\in Y}\{\omega(y) - \omega(y_1) - \langle\nabla\omega(y_1), y-y_1\rangle\}$, where ω is the chosen distance generating function (in accordance with Chapter 3.3) which is strongly convex with respect to norm $\|\cdot\|$, and $y_1 = \arg\min_{y\in Y}\omega(y)$.

Explanation of Table 5.2. In Table 5.2 we give a summary of constants Ω, Ω' and gradient bounds which appear in Theorems 3.4, 3.12, 3.16, 3.21, for the ℓ_1-, ℓ_2- and ℓ_∞-norms using the representations in (5.4) (note that these satisfy Assumption 5.11). We give brief explanations for how these constants are derived.

First, note that the columns of Table 5.2 represent smoothed versions of $D(x,p) = \|x-p\|_q$, with $q = 1, 2, \infty$ respectively; these are identical to Table 5.1. Note that the function $\Psi(x,y;p) = \langle B(x-p), y\rangle - \alpha\omega(y)$ is α-strongly concave with respect to the smoothing function ω given in the

third row, which in turn is 1-strongly convex with respect to the norm in the fifth row. We will also take ω to be the d.g.f. for the prox-setup of Y (Chapter 3.3), with the corresponding norm in the fifth row. The constant Ω in the sixth row is derived from $\Omega_Y = \max_{y\in Y}\omega(y) - \min_{y\in Y}\omega(y)$, and the seventh row $\Omega' = \max_{y\in Y}\langle\nabla\omega(y_1), y - y_1\rangle$ where $y_1 = \arg\min_{y\in Y}\omega(y)$ (used for dual averaging Theorem 3.16). For the first two columns, the gradient bound G in the eight row is derived from analysing

$$\|\nabla_y\Psi(x,y;p)\|_* = \|B(x-p) - \alpha\nabla\omega(y)\|_* = \|x - p - \alpha y\|_2 \le \|x-p\|_2 + \alpha\|y\|_2.$$

Note that for any $x \in X$, $\sum_{i\in\mathcal{A}_j} x_{ij}^2 \le 1$, and the same holds for p, since we know these are normalized, and for the first column, $\|y\|_2 \le N$, while for the second column we have $\|y\|_2 \le 1$. We will analyse the third column later, which requires more care.

Optimality bounds via regret minimization. We now give the optimality bounds from several choices of regret minimizing algorithms for bounding

$$\max_{y\in Y}\frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t\Psi(x_t,y;p_t) - \frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t\Psi(x_t,y_t;p_t).$$

Before doing so, however, we give a F-W interpretation of the case when follow-the-leader (Theorem 3.21) is used to compute the sequence $\{y_t\}_{t\in[T]}$. Note that since $\Psi(x_t,y;p_t)$ is concave in y, we will take $f_t(y) = -\Psi(x_t,y;p_t)$ and apply Theorem 3.21 to this, which yields the update rule (denoting $\bar{p}_t^\theta = \frac{1}{\Theta_t}\sum_{s\in[t]}\theta_s p_s$)

$$y_{t+1} = \arg\max_{y\in Y}\left\{\sum_{s\in[t]}\theta_s\Psi(x_s,y;p_s)\right\} = \arg\max_{y\in Y}\left\{\langle B(\bar{x}_t^\theta - \bar{p}_t^\theta), y\rangle - \alpha\omega(y)\right\} = \nabla_x D(\bar{x}_t^\theta, \bar{p}_t^\theta). \quad (5.8)$$

The last equality follows from the convex envelope theorem. Recalling that

$$x_{t+1} = \arg\min_{x\in X}\langle y_{t+1}, x\rangle = \arg\min_{x\in X}\langle\nabla_x D(\bar{x}_t^\theta, \bar{p}_t^\theta), x\rangle,$$

this is in fact a F-W type update.

Remark 5.14. In fact, if $p_t = p$, $\theta_t = 2/(t+1)$ for all $t \ge 1$, we have $p_t = \bar{p}_t^\theta$, and hence the update of Theorem 5.3 and the update in (5.8) are identical. In other words, in the static case, this choice exactly recovers the classical F-W algorithm. This was realized for the static case in Abernethy et al. [2]. However, for the dynamic case, there is a subtle but significant difference between a naïve choice of F-W update in Theorem 5.3, and an update rule derived from our principled primal-dual framework. We see that the naive rule of using p_t at each iteration incurs penalty terms in Theorem 5.3 that disappear when we use $\bar{p}_t^\theta$ instead, which is derived from our primal-dual framework (see Lemma 5.12). ∎

For the first two columns, i.e., for D derived from ℓ_1-norm and ℓ_2-norm, deriving optimality bounds is involves a relatively straightforward application of the regret bounds from Chapter 3 and Table 5.2. One note is that when $\alpha > 0$, we should use one of Theorems 3.12, 3.16, 3.21, since we have strong concavity of $\Psi(x_t,y;p_t)$, and need to exploit it to get the faster $O(1/T)$ convergence rate; however, when $\alpha = 0$, we should use Theorem 3.4. We focus on using Theorem 3.12, although using the others will give us similar guarantees.

Theorem 5.15. *Assume that $p_t \to p$. When $\alpha > 0$, computing $\{y_t\}_{t\in[T]}$ according to Theorem 3.12 and $\{x_t\}_{t\in[T]}$ according to (5.7) achieves an optimality bound of*

$$D_{1,\alpha}(\bar{x}_t^\theta, p) - \min_{x\in X} D_{1,\alpha}(x,p) \le \frac{2(2m+\alpha N)}{\alpha(T+1)} + o(1)$$
$$D_{2,\alpha}(\bar{x}_t^\theta, p) - \min_{x\in X} D_{2,\alpha}(x,p) \le \frac{2(2m+\alpha)}{\alpha(T+1)} + o(1).$$

When $\alpha = 0$, computing $\{y_t\}_{t\in[T]}$ according to Theorem 3.4 and $\{x_t\}_{t\in[T]}$ according to (5.7) achieves an optimality bound of

$$\|\bar{x}_t^\theta - p\|_1 - \min_{x\in X} \|x-p\|_1 \le \sqrt{\frac{2mN}{T}} + o(1)$$
$$\|\bar{x}_t^\theta - p\|_2 - \min_{x\in X} \|x-p\|_2 \le \sqrt{\frac{2m}{T}} + o(1).$$

For the third row with negative entropy ω, we can apply the bound of Theorem 3.16 immediately, since the gradient bound in that only involves $\|x_t - p_t\|_\infty$, which is ≤ 1. However, we need a more careful analysis of the gradient term G to apply Theorems 3.12 and 3.21. This is due to the fact that in the case when $\omega(y) = \sum_{k\in[2N]} y_k \log(y_k)$, we have

$$\nabla_y \Psi(x_t, y; p_t) = B(x_t - p_t) - \alpha(\log(y) + \mathbf{1}_{2N}),$$

where $\log(y)$ denotes the vector of entry-wise logarithms of y, but $\log(y)$ is not bounded over $y \in \Delta_{2N}$. We now give some results to do this. We start with a fundamental lemma on the updates.

Lemma 5.16. *Let $\omega(y) = \sum_{k\in[2N]} y_k \log(y_k)$. Then for any $\xi \in \mathbb{R}^{2N}$,*

$$y = \operatorname*{arg\,min}_{y\in\Delta_{2N}} \{\langle \xi, y\rangle + \omega(y)\} \implies \nabla\omega(y) = -\xi + c(\xi)\mathbf{1}_{2N},$$

where $c(\xi)$ is a finite constant depending only on ξ.

Proof. It is easy to check that each $y_k = \exp(-\xi_k)\left(\sum_{k'\in[2N]} \exp(-\xi_{k'})\right)^{-1}$ and $\nabla_k\omega(y) = \log(y_k) + 1$. The result then follows with $c(\xi) = 1 - \log\left(\sum_{k'\in[2N]} \exp(-\xi_{k'})\right)$. □

Denote $f_t(y) = -\Psi(x_t, y; p_t)$, $h_t(y) = f_t(y) - \alpha\omega(y) = \langle B(p_t - x_t), y\rangle$, and $c_t = c(\gamma_t\nabla f_t(y_t) - \nabla\omega(y_t))$. We consider the mirror descent algorithm from Theorem 3.12:

$$y_{t+1} = \mathrm{Prox}_{y_t}(\gamma_t\nabla f_t(y_t)) = \operatorname*{arg\,min}_{y\in Y} \{\langle \gamma_t\nabla f_t(y_t) - \nabla\omega(y_t), y\rangle + \omega(y)\}, \quad \gamma_t = \frac{2}{\alpha(t+1)}. \tag{5.9}$$

Lemma 5.17. *Let $\omega(y) = \sum_{k\in[2N]} y_k \log(y_k)$, $Y = \Delta_{2N}$, and suppose that $\{y_t\}_{t\in[T]}$ is updated according to (5.9). For all $t \ge 1$, we have*

$$\alpha\nabla\omega(y_{t+1}) = \hat{y}_t + C_t\mathbf{1}_{2N}, \quad \hat{y}_t \in \mathrm{Conv}(\{-\nabla h_s(y_s) : s \in [t]\})$$

and C_t is some finite constant.

Proof. By Lemma 5.16, (5.9) implies that

$$\nabla\omega(y_{t+1}) = \nabla\omega(y_t) - \gamma_t\nabla f_t(y_t) + c_t\mathbf{1}_{2N} = (1-\alpha\gamma_t)\nabla\omega(y_t) + \gamma_t\left(-\nabla h_t(y_t)\right) + c_t\mathbf{1}_{2N},$$

or equivalently,

$$\alpha\nabla\omega(y_{t+1}) = (1-\alpha\gamma_t)\left(\alpha\nabla\omega(y_t)\right) + \alpha\gamma_t\left(-\nabla h_t(y_t)\right) + \alpha c_t\mathbf{1}_{2N},$$

Note that the choice $\gamma_t = 2/(\alpha(t+1))$ ensures that $\alpha\gamma_t \in (0,1]$, and $\alpha\gamma_1 = 1$. This implies that $\alpha\nabla\omega(y_2) = -\nabla h_1(y_1) + c_1\mathbf{1}_{2N}$. Thus, the result follows from induction with base case $t = 1$. □

Theorem 5.18. *Let $\omega(y) = \sum_{k\in[2N]} y_k\log(y_k)$, $Y = \Delta_{2N}$, and suppose that $\{y_t\}_{t\in[T]}$ is updated according to (5.9). Then*

$$\begin{aligned}\max_{y\in Y}\frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t\Psi(x_t,y;p_t) - \frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t\Psi(x_t,y;p_t) &= \frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t f_t(y_t) - \min_{y\in Y}\frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t f_t(y)\\ &\le \frac{8}{\alpha\left(T+1\right)}.\end{aligned}$$

Consequently, for $\alpha > 0$, we have

$$D_{\infty,\alpha}(\bar{x}_T^\theta, p) - \min_{x\in X} D_{\infty,\alpha}(x,p) \le \frac{8}{\alpha(T+1)}.$$

Proof. Proposition 3.11 gives us that

$$\frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t f_t(y_t) - \min_{y\in Y}\frac{1}{\Theta_T}\sum_{t\in[T]}\theta_t f_t(y) \le \frac{1}{\Theta_T}\sum_{t\in[T]}\frac{\theta_t}{\gamma_t}\left(\gamma_t\langle\nabla f_t(y_t), y_t - y_{t+1}\rangle - V_{y_t}(y_{t+1})\right).$$

We now bound the term $\langle\nabla f_t(y_t), y_t - y_{t+1}\rangle$. By Lemma 5.17, we have that

$$\begin{aligned}\langle\nabla f_t(y_t), y_t - y_{t+1}\rangle &= \langle\nabla h_t(y_t) + \alpha\nabla\omega(y_t), y_t - y_{t+1}\rangle\\ &= \langle\nabla h_t(y_t) + \hat{y}_t + C_t\mathbf{1}_{2N}, y_t - y_{t+1}\rangle, \quad \hat{y}_t \in \mathrm{Conv}\left(\{-\nabla h_s(y_s) : s\in[t]\}\right).\end{aligned}$$

Now, consider the matrix $U = I_{2N} - \frac{1}{2N}\mathbf{1}_{2N}\mathbf{1}_{2N}^\top$, which is the projection matrix onto the subspace $S = \{y\in\mathbb{R}^{2N} : \langle\mathbf{1}_{2N}, y\rangle = 0\}$. Note that $U\mathbf{1}_{2N} = 0$. Since $y_t, y_{t+1}\in Y = \Delta_{2N}$, $y_t - y_{t+1}\in S$, therefore we have

$$\begin{aligned}\langle\nabla f_t(y_t), y_t - y_{t+1}\rangle &= \langle\nabla f_t(y_t), U(y_t - y_{t+1})\rangle\\ &= \langle U\nabla f_t(y_t), y_t - y_{t+1}\rangle\\ &= \langle U\nabla h_t(y_t) + \alpha U\nabla\omega(y_t), y_t - y_{t+1}\rangle\\ &= \langle U\nabla h_t(y_t) + U\hat{y}_t, y_t - y_{t+1}\rangle\\ &= \langle\nabla h_t(y_t) + \hat{y}_t, y_t - y_{t+1}\rangle\\ &\le \|\nabla h_t(y_t) + \hat{y}_t\|_\infty\|y_t - y_{t+1}\|_1\\ &\le 2\|y_t - y_{t+1}\|_1\end{aligned}$$

The last equality follows because $x_s - p_s \in S$ for all $s \in [t]$, hence $\nabla h_t(y_t) + \hat{y}_t \in S$, and the inequalities follows by Cauchy-Schwarz and the fact that each entry of x_t, p_t is contained in $[0,1]$. The bound

$$\gamma_t \langle \nabla f_t(y_t), y_t - y_{t+1} \rangle - V_{y_t}(y_{t+1}) \leq 2\gamma_t \|y_t - y_{t+1}\|_1 - \frac{1}{2}\|y_t - y_{t+1}\|_1^2 \leq 2\gamma_t^2$$

then gives us the result. □

If we wish to use $\alpha = 0$, then we can utilize Theorem 3.4 to compute the $\{y_t\}_{t\in[T]}$, which results in the following bound.

Theorem 5.19. *Let $\omega(y) = \sum_{k\in[2N]} y_k \log(y_k)$, $Y = \Delta_{2N}$, and suppose that $\{y_t\}_{t\in[T]}$ is updated according to Theorem 3.4. Then*

$$\max_{y\in Y} \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \Psi(x_t, y; p_t) - \frac{1}{\Theta_T} \sum_{t\in[T]} \theta_t \Psi(x_t, y; p_t) = \sqrt{\frac{8\log(2N)}{T}}.$$

Consequently,, we have

$$\|\bar{x}_T^\theta - p\|_\infty - \min_{x\in X} \|x - p\|_\infty \leq \sqrt{\frac{8\log(2N)}{T}}.$$

Finally, we remark that in all cases, bounds on $D_{q,\alpha}(\bar{x}_T^\theta, p)$ (e.g., Theorems 5.18, 5.19) can be translated to bounds on $\|\bar{x}_T^\theta - p\|_q$; see Table 5.1. Furthermore, the smoothing parameter α can be chosen to optimize the associated regret bound similar to Remark 5.7.

5.3.3 Combinatorial Subproblem

In both approaches of Sections 5.3.1 and 5.3.2, we must solve a linear optimization problem over X. Since X is a polytope with vertices $a(\sigma)$, we have, for a cost vector c,

$$z^* = a(\sigma^*), \quad \text{where} \quad \sigma^* = \arg\min_{\sigma\in S_n} \langle a(\sigma), c \rangle,$$

Thus, in each iteration of the primal and the dual approaches, we must solve the following combinatorial optimization problem over rankings:

$$\min_{\sigma} \left\{ \sum_{j\in[m]} \sum_{i\in\mathcal{A}_j} y_{ij,t} a_{ij}(\sigma) : \ \sigma \in S_n \right\}. \tag{5.10}$$

On the one hand, this problem is NP-hard, since it is a generalization of the linear ordering problem and the maximum weighted independent set problem, see e.g., [147, Proposition 3]. Indeed, this presents a major drawback of the iterative approaches from Sections 5.3.1 and 5.3.2. However, we note that the exact same combinatorial problem must be solved in all other approaches of learning a non-parametric choice model (see Appendix B.1 and in particular the equations (B.3), (B.6), (B.9)). On the other hand, while we cannot avoid the NP-hardness in learning a non-parametric choice model from data, we note that (5.10) can be formulated as a (relatively) compact integer program with $O(n^2)$ variables and $O(n^3)$ constraints, and also it can be handled efficiently by off-the-shelf integer programming solvers (see Figure 5.2). Furthermore, Jagabathula and Rusmevichientong

[86] prove polynomial-time solvability of (5.10) under a number of assumptions on the subsets $\mathcal{A}_1, \ldots, \mathcal{A}_m$. Thus, by employing our suggested first order approaches in Sections 5.3.1 and 5.3.2, we avoid the problem of having to deal with the high-dimensional representation of X (which can involve $n!$ variables and the resulting full optimization model may not fit in the computer memory), and instead solve a relatively compact integer program at each iteration.

5.4 Computational Study

In our computational study, we compare the performance of our primal and dual approaches for learning non-parametric choice models in both the static and the dynamic settings with a variety of ground truth choice models with mixed MNL structure. We assess performance by examining model fit, sparsity of our learned model, and algorithm efficiency. We test the impact of the choice of different distance measures $D(\cdot, \cdot)$, which allows us to compare and contrast some of the distance measures studied in the literature. In the dynamic setting, we also examine the effect of the rate at which observations are added during the solution process on our performance metrics.

All experiments are conducted on a server with 2.8 GHz processor and 64GB memory, using Python 3.6. Gurobi 8.0 (with default Gurobi settings except we limit the number of threads to 2) is used to solve the integer programming subproblems.

Test Instances. We employ a setup similar to Bertsimas and Mišic [27, Section 5.3]. Our ground truth choice model over $n = 10$ items (plus one no-choice option) is a mixed MNL model with $K = 5$ segments. Given mixing probabilities $w \in \Delta_K$ and K sets of utilities $\{u_{i,k}\}_{i \in \{0\} \cup [n]}$, $k \in [K]$, the mixed MNL model chooses an item $i \in \mathcal{A} \subseteq [n]$ with probability

$$\mathbb{P}[i \mid \mathcal{A}] = \sum_{k \in [K]} w_k \frac{u_{i,k}}{u_{0,k} + \sum_{i' \in \mathcal{A}} u_{i',k}}.$$

For each $k \in [K]$, we generate $n+1$ parameters $q_{i,k} \sim U(0,1)$, $i \in \{0\} \cup [n]$ (recall that 0 denotes the no-choice option present in each subset). The utilities $u_{i,k}$ are then set as follows: four randomly chosen $i \in \{0\} \cup [n]$ are set to $u_{i,k} = Lq_{i,k}$, where $L = 5$, while the rest are set to $u_{i,k} = q_{i,k}/10$. The mixing probabilities $\{w_k\}_{k \in [K]}$ are chosen randomly from the $(K-1)$-dimensional simplex. We test on 100 randomly generated instances of this ground truth model. In Appendix B.2, we also provide results for varying the parameters to $K \in \{1, 10\}$, $L \in \{10, 100\}$ and $m \in \{10, 50\}$, and thus test the effect of different ground truth models on the conclusions drawn. We observe that in these different ground truth models, the conclusions are in general in line with the ones from $K = L = 5$, $m = 20$ setting; therefore, in the main text we focus on this latter case and defer to Appendix B.2 for further details and discussion on the results using other parameter values.

Algorithm implementation. We consider three different choices of distance measures $D(\cdot, \cdot)$ arising from ℓ_1-, ℓ_2-, and ℓ_∞-norms. For each one of these distance measures $D(\cdot, \cdot)$, we implemented the following solution methods from Section 5.3

- the naïve primal approach via the Frank-Wolfe algorithm, where we (necessarily) smoothed D according to Table 5.1 (see Algorithm 4).
- the primal-dual approach without smoothing using online mirror descent (MD), i.e., Theorem 5.19.
- the primal-dual approach with smoothing using online MD, i.e., Theorem 5.18.

For online MD, we used constant step size policies based on constants Ω_Y, G and the maximum iteration count T. For each of these methods and norms, we set a maximum iteration limit of $T = 10,000$.

In the static setup, our non-parametric estimation procedure is as follows. We first generate $m = 20$ subsets of $[n]$ of maximum size $\lfloor n/2 \rfloor$ uniformly at random. We append the no-choice option 0 to all of these. Using the ground truth model, we compute the p_{train} vector, where $p_{\text{train},ij} = \mathbb{P}[i \mid \mathcal{A}_j]$, and $\mathcal{A}_j$ is a subset from our training set. Then we set $p_t = p_{\text{train}}$ at each iteration of these three solution methods.

In the dynamic setup, we also use the three methods outlined above, but only examine the non-parametric estimation model where the distance measure $D(\cdot, \cdot)$ is based on ℓ_2-norm. We generated a sequence of $p_t \to p_{\text{train}}$, and at each iteration we supply p_t to the algorithm. We initially generate 2000 random choice observations (i, j), where $\mathcal{A}_j$ is one of the training subsets chosen randomly, and $i \in \mathcal{A}_j$ is chosen with the probability $p_{\text{train},ij}$. We then compute p^1 using these observations according to (5.1). For $t > 1$, we generate κ more observations, then update p^{t-1} with these new observations. We tested various choices of κ between 0 (no new observations) and 1000.

For both data regimes, we terminate training according to the mean absolute error (MAE), defined as $\text{MAE}(p, p') = \frac{1}{\text{length}(p)} \sum_{i,j} |p_{ij} - p'_{ij}|$, where $\text{length}(p)$ is the length of the vector p. In the static setup, we terminate training when $\text{MAE}(p_{\text{train}}, \hat{p}_t) \leq 0.001$, where $\hat{p}_t$ is the vector of estimated choice probabilities for the $m = 20$ subsets after t iterations. In the dynamic setup, we terminate training when $\text{MAE}(p_t, \hat{p}_t) \leq 0.001$.

Performance metrics. We compare the effectiveness of our methods using three criteria: model fit, sparsity, and algorithm efficiency.

To evaluate model fit, we examine the mean absolute error of choice probabilities on subsets generated independently from the training set. Specifically, we generate 100 subsets of $[n]$ of maximum size $\lfloor n/2 \rfloor$ uniformly at random (independently to the training subsets), and append the no-choice option 0 to each of them. We compute the vector of choice probabilities p_{test} using our ground truth model. Letting $\hat{p}$ be the choice probabilities on the test subsets computed from the choice model estimated at the training phase, we then calculate $\text{MAE}(p_{\text{test}}, \hat{p})$.

To evaluate sparsity of our estimated model, we examine the number of different rankings σ with positive probability $\lambda(\sigma) > 0$ in our estimated model. Sparsity is very much desired for non-parametric models, since choice probabilities for sparser models can be computed more efficiently.

To evaluate algorithm efficiency, we examine the number of iterations until the termination criterion is reached. While we could have used solution time as another metric for this purpose, we observed in the static setup that solution time is highly correlated with the number of iterations; see Figure 5.2. Because runtimes are affected by how fast the combinatorial subproblem (5.10) is solved, but the focus of our work is not on this aspect, in our discussions we focused on the number of iterations as a more accurate representation of algorithm efficiency for our purposes.

5.4.1 Static estimation results

Figure 5.1 plots the test MAE, the average number of rankings and the average number of iterations for each of the three solution methods when using different norms for constructing D. We observe that the three methods have roughly the same test MAE for each of the norms, with perhaps the smoothed dual method performing slightly better when D is based on certain norms. On the other hand, the non-smooth dual method clearly learns a sparser model, and clearly terminates

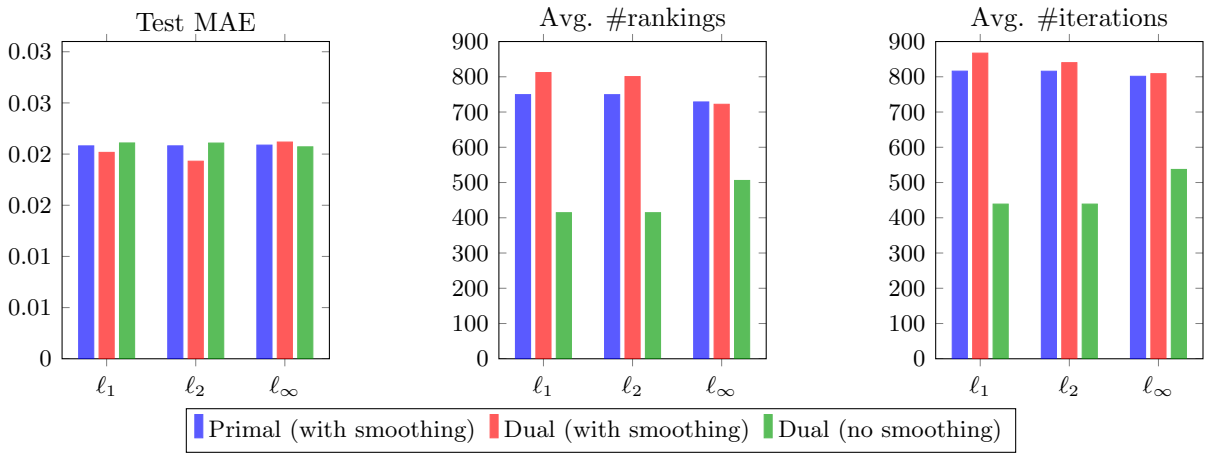


Figure 5.1: Performance metrics using different norms in the static setup.

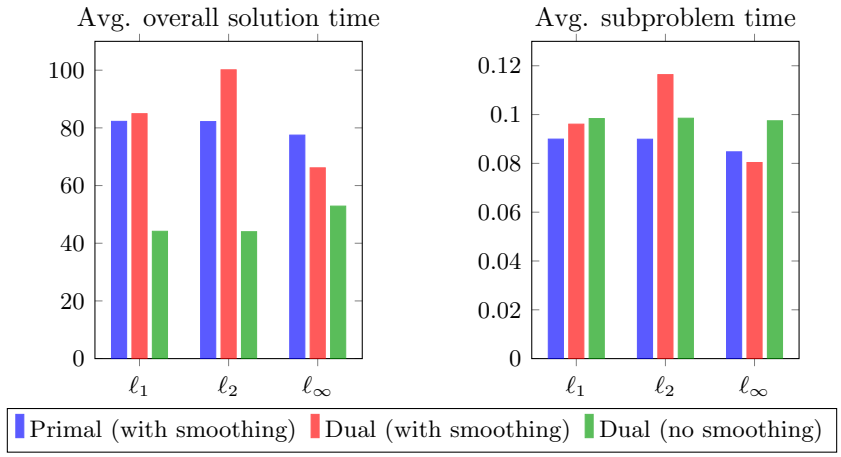


Figure 5.2: Overall solution times and subproblem times per iteration (both in seconds) in the static setup.

in less number of iterations than the other two methods, which are similar in these two metrics. Therefore, we conclude that, regardless of the type of norm used in the estimation procedure, the non-smooth dual method is superior in the static setup, since it manages to learn a sparser model more efficiently, while maintaining the same model fit. In addition, Figure 5.1 also indicates that in terms of the number of rankings and the number of iterations metrics, the performance of our primal with smoothing and our dual with smoothing approaches do not vary much based on the norm used. Moreover, in these performance metrics, while our dual approach without smoothing has a clear advantage, its performance also has a slight dependence on the norm used, with slight deterioration when ℓ_∞ norm is used.

Figure 5.2 shows the average solution times, the average number of iterations and the average subproblem times for each method and norm. From Figure 5.2 we conclude that the non-smooth primal approach is still outperforming the other two with respect to overall solution time. Note that there are slight differences in performance between the number of iterations and the overall

solution times. We attribute this to the slight variation in the average subproblem times between the methods and norms used. We do not believe that the variation in average subproblem time is the result of any inherent property of the methods used or norms. Moreover, the average subproblem solution times are quite small, and thus the variations in subproblem solution times are relatively small.

In Appendix B.2, we further examine whether the ground truth choice model has an effect on algorithm performance. In general, the trends observed here persist when we vary the model parameters.

5.4.2 Dynamic estimation results

For the dynamic data regime, we examine the performance of the three solution methods for $\kappa = 50, 100, \ldots, 950, 1000$ in Figure 5.3, where the distance measure D is based on the ℓ_2 norm. In this figure, $\kappa = \infty$ corresponds to using the true static p_{train} choice probabilities computed from the ground truth model. From Figure 5.3 we observe that the test MAE is similar for all three methods, where the profiles are almost identical. In terms of the average number of iterations and rankings (numbers above 2000 are not shown), we observe that, regardless of the value of κ, the non-smooth dual approach clearly outperforms the other two, followed by the smoothed primal method, and then the smoothed dual method. Figure 5.3 provides another insight that the performance of dual method with smoothing is far more sensitive to κ and hence the number of observations seen. That is, when the number of observations is limited, the estimated model from the smoothed dual method tends to be far more dense in terms of the number of rankings used. Interestingly, for $\kappa = \infty$, we notice that in terms of the average number of iterations and the average number of rankings, the performance of the dual approach with smoothing shows a drastic improvement and it almost matches with the performance of primal approach with smoothing. Appendix B.2.3 presents the analogous figures for the dynamic data regime, where the distance measure D is based on the ℓ_1 and ℓ_∞ norms.

Finally, we would like to also highlight the importance of using additional observations at each iteration in the dynamic setup. To examine this, we set $\kappa = 0$, i.e., we do not add any observations at each iteration, and simply set $p_t = p_1$ for all t. In this case we found that all methods failed to find an estimated model such that $\text{MAE}(p_t, \hat{p}_t) \leq 0.001$ within the maximum iteration limit. This is due to the finite sample error. The estimated choice probabilities $\hat{p}_t$ and the true probabilities p_{train} both belong to the set X. However, in general, $p^1 \notin X$ because p^1 is generated using samples from p_{train}. For a low number of samples (in our case, 2000 samples), the distance between p^1 and X turns out to be too large for our termination criterion. In fact, using exact methods, we found that the average minimum MAE between p^1 and X is ≥ 0.007, which is above our termination criterion. Therefore, in the dynamic case, additional observations at each iteration are crucial for convergence.

5.4.3 Additional remarks

For each of the three methods, in both the static and the dynamic setups, the average number of rankings and iterations are highly correlated. In fact, the Spearman correlation between these two metrics in our result is ≈ 0.922, thus we conclude that the average number of iterations is a good proxy for model sparsity. This can be seen in the theory: all of our algorithms start with one ranking, and at each iteration adds at most one ranking to the estimated model, which provides an explicit bound on the model sparsity.

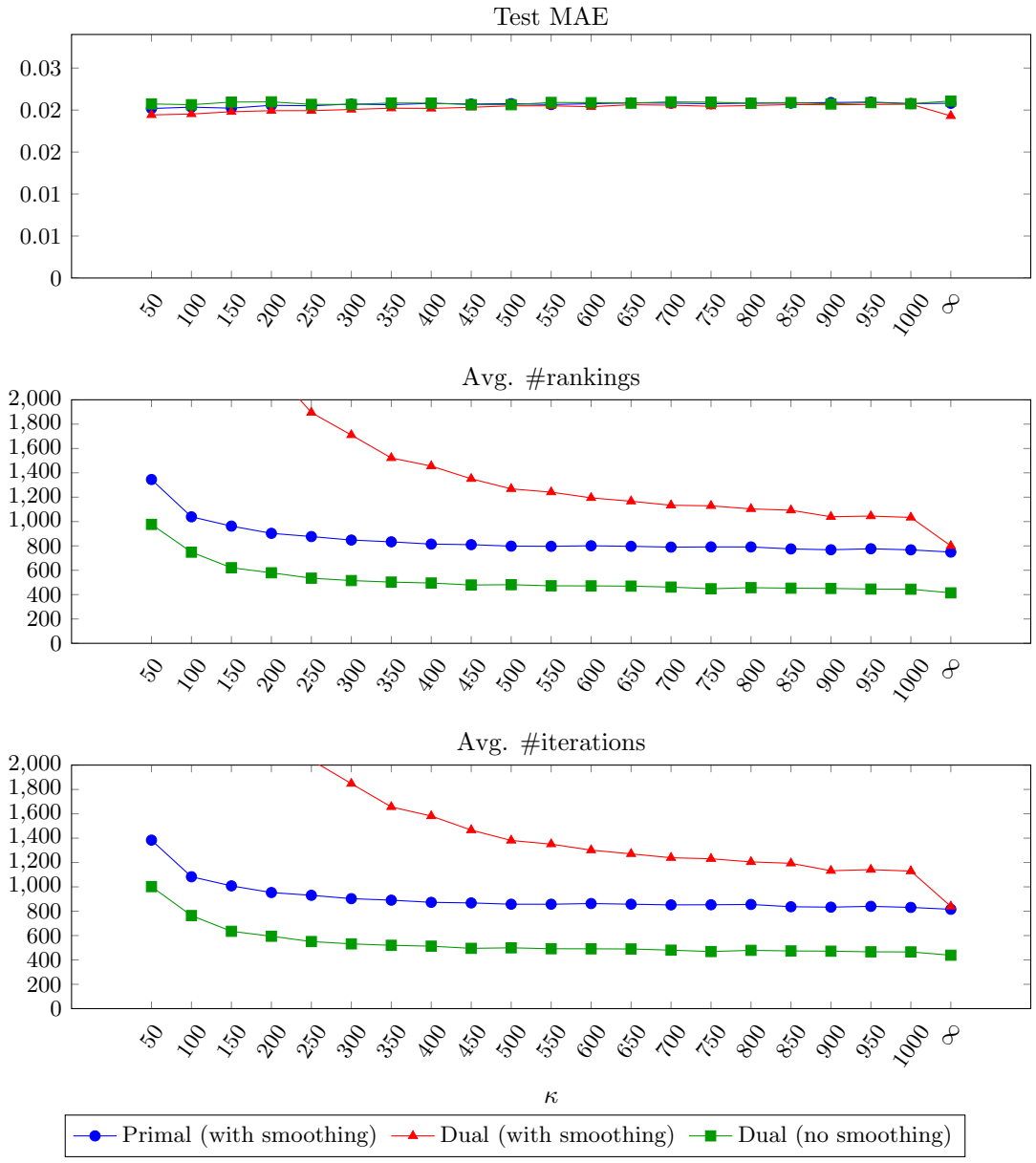


Figure 5.3: Performance metrics using ℓ_2-norm for dynamic data.

For dynamic data regime, the MAE does not vary significantly as κ increases, even compared to using the true choice probabilities ($\kappa = \infty$). This can be explained by examining the average number of iterations. For low κ, the average number of iterations until termination is much higher than for higher κ, thus the total number of observations seen is still enough to learn a well-fitted model. On the other hand, there is a significant difference in terms of model sparsity when we compare low κ and high κ. This is perhaps due to the fact that on average, the number of iterations needed for convergence when $\kappa = 50$ is roughly twice that of $\kappa = 1000$, but this means that throughout the whole solution procedure, the number of observations seen for $\kappa = 1000$ is roughly 10 times the amount seen for $\kappa = 50$. However, the gains to sparsity rapidly diminish as κ increases.

Finally, we remark that comparing the first-order methods FW and online MD, one can argue that FW is much simpler to implement, since there is essentially no parameter tuning in Algorithm 4. The only tuning involved for the primal approach was computing the smoothing parameters for D via Table 5.1, but this is separate from Algorithm 4. On the other hand, online MD requires tuning the selection of step size, time horizon T, and computing the constants Ω_Y, G (which in turn affects the smoothing parameters). However, for the particular choice model estimation problem, these quantities are quite straightforward to compute, and our analysis and numerical results are based on such 'textbook' constant step size policies derived from these, which worked quite well. In terms of performance, we see that the extra sophistication in non-smooth MD can significantly outperform FW.

Chapter 6

Joint Risk Analysis of Prediction and Optimization

6.1 Introduction

The fields of optimization, statistics and machine learning have long had a close relationship. However, most research done has been on how to employ optimization techniques to solve statistical or machine learning problems. In this chapter, we explore some steps in the other direction: how can statistics and machine learning help us handle optimization problems under uncertainty? In practice many optimization problems face uncertainty in their data in either the objective or the problem constraints, especially when the model parameters are estimated from data. Statistical inference and machine learning is often used to estimate model parameters. Such processes often come with performance guarantees in terms of the accuracy of the parameter estimation. However, considering that the eventual goal in this process is not to have the best parameter estimation, but to make the best decision in the subsequent optimization problem under uncertainty. Hence, the optimality gap presents a more appropriate performance measure in this end-to-end process. In this chapter, we explore how the parameter estimation guarantees can be related to optimality gap guarantees for the subsequent optimization problem.

Note that in Chapter 1.4 we introduced joint estimation and optimization, which considered optimizing a function $f(x, u)$ given access to a sequence of converging parameter estimates $\{u_t\}_{t\geq 1}$, $u_t \to u$. In Chapter 2.5 we presented an algorithmic framework for this problem. However, there we did not have control of the estimating sequence $\{u_t\}_{t\geq 1}$. In contrast, this chapter considers the construction of the parameter estimates themselves, and explores what the 'right' method, statistically speaking, to construct these estimates is.

We consider the following optimization problem:

$$\min_x \left\{ f(x) + c^\top x : x \in X \right\} \tag{6.1}$$

where $X \subset \mathbb{R}^m$ is a convex compact domain, and $f : X \to \mathbb{R}$ is a convex function (and hence continuous on the relative interior of X). We consider the setting when the linear perturbation vector c is not known exactly, but instead is governed via covariates w. More precisely, suppose the covariates w belong to a set W, and the perturbation vectors belong to a set $C \subseteq \mathbb{R}^m$, both of which live in Euclidean spaces. We have access to historical data $H_n := \{(w_i, c_i) : i \in [n]\}$, and we need to solve (6.1) for c yet we are only given information of w. We assume that the (w_i, c_i) are

realizations of i.i.d. random variables according to some distribution $\mathbb{P}$ on $W \times C$. Furthermore, we also assume that $(w, c) \sim \mathbb{P}$ also, otherwise the data would be useless to us.

Traditionally, first based on the data H_n, a prediction model resulting in the estimation of a function $g : W \to C$ is built to capture the dependency of c on w. Then, when given a covariate w, the problem (6.1) is solved with c replaced by the prediction $g(w)$. Many real-world problems can be captured in this setting, and in fact, this two-step process is commonly used amongst practitioners in decision-making domains. Here, we give two examples, although many more exist.

Example 6.1. A classical application of the network shortest path problem is maintenance scheduling. Suppose we have a collection of service items (e.g., machines, vehicles) which we maintain over a certain time horizon. These items need refurbishment or replacement after a certain number of time periods. Then the optimal maintenance schedule can be solved as a shortest path problem over an appropriately defined network, with arc 'distances' given by the maintenance costs. Typically, these costs are assumed to be deterministic. However, it is not hard to imagine that since these are future costs, they are obtained via forecasts. In this setting, X are the constraints from the shortest paths problem, each $x \in X$ represents a path through the network, $f(x) = 0$ for all $x \in X$, and c is the vector of arc distances from the maintenance costs. Side information (covariates) w can consist of (amongst others) seasonality, demand, supply and other economic factors. ∎

Example 6.2. Consider a portfolio optimization problem, where the task is to allocate wealth to different m different assets to maximize investment return. In the typical mean-variance formulation, the goal is to simultaneously minimize the variance of the portfolio return, while maximizing the expected return. Thus, X is the set of all possible asset allocations, each $x \in X$ represents an asset allocation (i.e., x_i represents how much wealth to invest into asset i), $f(x) = \gamma x^\top \Sigma x$ is the variance of the portfolio return with Σ being the covariance matrix of the returns between the assets, and $c = -\mu$ is the vector of returns for each assets. In many settings, Σ is assumed stable, and μ is predicted through market factors (e.g., liquidity, value, momentum, volume) which we can take to be the side information w. ∎

6.1.1 Related Literature

Both prediction and optimization have been studied extensively on their own, yet their joint analysis is notably lacking. That is, the prediction function $g : W \to C$ is selected to minimize some measure of error on the given data H_n, but there are relatively few studies on the effect of utilizing the prediction $g(w)$ in the optimization problem (6.1).

A compelling method to incorporate the covariates w is to use density estimation, and solve a variant of sample average approximation, without appealing to an explicit prediction model. This was studied by Hannah et al. [72], Hanasusanto and Kuhn [71], Bertsimas and Kallus [26], Ban and Rudin [9], Bertsimas and Van Parys [28], Ho and Hanasusanto [79] who all gave various guarantees on this approach. While we believe that this is a compelling method, in practice this is not the approach that is taken. Practitioners prefer to use the 'traditional' pipeline of first building a prediction model $g : W \to C$, then using the predictions $g(w)$, which may be due to its simplicity. Technically, there are two advantages of a two-step approach. First, solving the optimization problem is simpler, due to not relying on a sample average approximation, which can be expensive when n is large. Second, the two-step approach allows the potential to simultaneously develop an explanatory model between the covariates w and the vectors c, which is useful in a variety of settings.

The relationship between prediction models used to obtain model parameters and the subse-

quent optimization performance has, to our knowledge, only been examined by Kao et al. [96], Elmachtoub and Grigas [58] and Donti et al. [55]. Kao et al. [96] examined this in the specialized setting when $X = \mathbb{R}^m$, f is a strongly convex quadratic, and the prediction model g is linear. They presented theoretical guarantees under a particular data distribution. Donti et al. [55] propose a scheme to differentiate the optimality gap, which gives rise to a stochastic gradient descent scheme for direct risk minimization. However, they provide no theoretical guarantees for the convergence of the risk quantities in their algorithmic scheme. Closest to the work in this chapter is the paper of Elmachtoub and Grigas [58], who examine the true optimality gap loss, and propose a convex surrogate loss from an upper bound on this. They show Fisher consistency of their surrogate loss under certain distributional assumptions, but do not give explicit relationships between the performance of the prediction part and the optimization part, which is the topic of this chapter.

A key challenge in considering prediction and optimization jointly is that the optimality gap, the natural measure of performance to use in this context, is no longer convex in the model parameters (see Observation 6.4). Therefore, this necessitates the use of convex surrogate loss functions. Surrogate loss functions have been used extensively in other machine learning contexts such as classification and robust regression. The relationship between surrogate losses and the true 0-1 loss is well-understood in the classification context. For example, [140, 141, 101, 156, 142, 10] have developed a general theory for the minimization of the true 0-1 risk via a surrogate risk which satisfies certain criteria. In the context of robust regression and density estimation problems, Steinwart [143] builds a theory for the relationship between true and surrogate risk.

6.1.2 Contributions

In this chapter, we examine how the performance of the prediction part relates to the performance of the optimization part. In particular, we describe conditions for the existence of explicit relationships between the learning performance, i.e., the surrogate risk, and the optimization performance, i.e., the true risk. In Section 6.2, we precisely define the problem we address, and outline the challenges.

- In Section 6.3, we rigorously derive the technical conditions on the surrogate loss that allows us to asymptotically minimize the true risk by minimizing the surrogate risk. These conditions are based solely on the choice of prediction performance measure, rather than the class of prediction models g or the optimization domain X or function f. These conditions allow us to compare and contrast the use of different prediction model training methods in the context of optimization, and provide us with guidelines into selecting these for such applications. We examine several methods, and show how to check the conditions for these.

- Often in statistical learning, we assume minimal knowledge of the distribution $\mathbb{P}$. Therefore, the distribution dependent nature of our results from Section 6.3 is not so desirable. In Section 6.4, building on the results from Steinwart [143], we establish conditions for distribution independent relationships between true risk and the surrogate risk. In general checking these conditions is difficult for many existing methods. Thus, we focus on a tractable special case, using the squared loss function $\ell(d, c) = \|d - c\|_2^2$ to measure prediction performance (i.e., the least squares method to train a prediction model). Using the conditions established, we prove an explicit relationship between the surrogate squared risk and the true optimality gap risk.

Notation. We make use of the following notation throughout. Given a set Z in a topological space, we denote its closure by $\mathrm{cl}(Z)$. We denote the power set, the collection of all subsets of X, as 2^X. We let $X^*(d) = \arg\min_{x \in X} \left\{ f(x) + d^\top x \right\} \in 2^X$ be the argmin mapping, and $x^*(d)$ be a

selection from $X^*(d)$ obtained from some deterministic algorithm. More precisely, $x^* : \mathbb{R}^m \to X$ is a function such that for any $d \in \mathbb{R}^m$, $x^*(d) \in X^*(d)$. Our results are agnostic to the specific choice of algorithm.

6.2 Risk Minimization for Prediction and Optimization

Suppose we have a function $g : W \to C$ and a point (w, c). To assess the quality of using $g(w)$ in place of c in (6.1), we define the *loss* as the optimality gap of the solution obtained with $g(w)$ on the true objective vector c, that is, the quality of the prediction $d = g(w)$ with respect to (6.1) is given by the *true loss function*

$$L(d, c) := f(x^*(d)) + c^\top x^*(d) - \min_{x \in X} \left\{ f(x) + c^\top x \right\}. \tag{6.2}$$

Note that (a slight variant of) this loss function is examined by Elmachtoub and Grigas [58]. Given any c, $L(d, c) \geq 0$ for all d and $L(c, c) = 0$. Also note that L depends on the function x^*, i.e., the algorithm that we use to solve $\min_{x \in X} \left\{ f(x) + d^\top x \right\}$ for different $d \in \mathbb{R}^m$. We will take x^* to be fixed throughout the chapter. Note, however, that the specific choice of x^* only affects our results up to measurability concerns; we show in Lemma 6.12 that any x^* is Lebesgue measurable, so we can safely fix x^* without changing the results as long as our distribution $\mathbb{P}$ is Lebesgue measurable. In practice, any distribution we encounter will be Lebesgue measurable; we explicitly impose this in Assumption 6.13. Henceforth, when measurability of functions is discussed, we will understand this to be in the sense of Lebesgue.

Since (w, c) is randomly drawn from $\mathbb{P}$, we assess the performance of a function $g : W \to C$ in terms of the expected true loss, i.e., the *true risk*

$$R(g, \mathbb{P}) := \mathbb{E}[L(g(w), c)]. \tag{6.3}$$

The best possible risk we can achieve is

$$R(\mathbb{P}) := \inf_g \{ R(g, \mathbb{P}) : g \text{ measurable} \}. \tag{6.4}$$

We first state a basic fact about (6.4).

Lemma 6.3. *The function $g(w) = \mathbb{E}[c \mid w]$ minimizes (6.4). Furthermore,*

$$R(\mathbb{P}) = \mathbb{E}\left[\min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d', c) \mid w] \right].$$

Proof. Note that measurability of $w \mapsto \mathbb{E}[c \mid w]$ is obvious by definition of the conditional expectation. Fix some measurable $g : W \to \mathbb{R}^m$. Observe that for $w \in W$,

$$\begin{aligned}
\mathbb{E}[L(g(w), c) \mid w] &= \mathbb{E}\left[f(x^*(g(w))) + c^\top x^*(g(w)) - \min_{x \in X} \left\{ f(x) + c^\top x \right\} \mid w \right] \\
&= \mathbb{E}\left[c \mid w \right]^\top x^*(g(w)) - \mathbb{E}\left[\min_{x \in X} \left\{ f(x) + c^\top x \right\} \mid w \right] \\
&\geq f(x^*(\mathbb{E}[c \mid w])) + \mathbb{E}\left[c \mid w \right]^\top x^*\left(\mathbb{E}[c \mid w] \right) - \mathbb{E}\left[\min_{x \in X} \left\{ f(x) + c^\top x \right\} \mid w \right] \\
&= \mathbb{E}[L(\mathbb{E}[c \mid w], c) \mid w],
\end{aligned}$$

where the inequality follows from the definition of $x^*(\cdot)$. The first result then follows from integrating both sides of this relation over $w \in W$.

The second result follows because

$$\begin{aligned}\min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d', c) \mid w] &= \min_{d' \in \mathbb{R}^m} f(x^*(d')) + \mathbb{E}[c \mid w]^\top x^*(d') - \mathbb{E}\left[\min_{x \in X} \left\{ f(x) + c^\top x \right\}\right] \\ &= f(x^*(\mathbb{E}[c \mid w])) + \mathbb{E}[c \mid w]^\top x^*(\mathbb{E}[c \mid w]) - \mathbb{E}\left[\min_{x \in X} \left\{ f(x) + c^\top x \right\}\right],\end{aligned}$$

and integrating both sides over $w \in W$. □

Lemma 6.3 shows that the conditional expectation is a minimizer of (6.4). There are many regression techniques which aim to recover the conditional expectation $g(w) = \mathbb{E}[c \mid w]$, thus any one of these are applicable in practice. However, it is not yet well-understood how these techniques perform for the true risk (6.3). Thus, this chapter aims to understand how these existing techniques relate to the risk as defined by (6.3). In other words, if we employ a traditional regression technique, what can we say about the risk minimization problem (6.4)? More precisely, given n data points, suppose g_n is constructed from these using a traditional regression technique. We would like to understand which techniques can guarantee that $R(g_n, \mathbb{P}) \to R(\mathbb{P})$ as the number of data points $n \to \infty$.

A naïve attempt to solve (6.4) is to directly use empirical risk minimization (ERM) with the loss L to attain risk bounds with the usual methodology. However, $L(d, c)$ is not convex in d.

Observation 6.4. *For any fixed c, the loss function $L(d, c)$ is not convex in d. For simplicity take $f(x) = 0$, so that we have a linear objective $c^\top x$ and that minimizers are always at points on the boundary of X. Consider two extreme points of X, x_0, x_1 with $c^\top x_0 > c^\top x_1$. Choose d_0, d_1 such that minimizing $d_k^\top x$ over $x \in X$ results in the unique minimum x_k for $k = 0, 1$. Now note that $L(d_0, c) - L(d_1, c) = c^\top x_0 - c^\top x_1 > 0$. Let us now consider $d_\gamma = (1 - \gamma)d_0 + \gamma d_1$ for very small $\gamma \in (0, 1)$. When γ is sufficiently small, then d_γ will also have x_0 as a unique minimizer, so $L(d_\gamma, c) = L(d_0, c)$. Then because $L(d_0, c) > L(d_1, c)$, we have $L(d_\gamma, c) = L(d_0, c) > (1 - \gamma)L(d_0, c) + \gamma L(d_1, c)$. Hence, $L(d, c)$ is not convex in d for any c.*

Given Observation 6.4, it is not possible in general to obtain a tractable approach with certified performance guarantees from minimizing the empirical risk based on the true loss function L. In this case, the natural remedy is to use a *surrogate loss function* ℓ in place of L. The use of surrogate loss functions to ensure algorithmic tractability is very common in machine learning. For example, convex surrogates such as hinge loss are used instead of the non-convex true 0-1 loss in classification problems.

A good surrogate loss function ℓ should mimic the natural properties of the true loss function L, i.e., $\ell(c, c) = 0$, $\ell(d, c) \geq 0$ for any d, c. However, the most important feature of a surrogate loss function is how its risk bound relate to the true risk (6.3). Consequently, in this chapter we will explore this relationship and identify important properties of surrogate loss functions that enable us to derive guarantees on the true risk. We would like to identify essential properties of surrogate loss functions $\ell(g(w), c)$ such that they can accurately, in some sense, assess the quality of using $g(w)$ in place of c in (6.1), while remaining computationally tractable to optimize (e.g., being convex in $g(w)$). To this end, we define the surrogate risk as:

$$R_\ell(g, \mathbb{P}) := \mathbb{E}\left[\ell(g(w), c)\right]. \tag{6.5}$$

For any given surrogate loss function ℓ, analogous to (6.4), we also define:

$$R_\ell(\mathbb{P}) := \inf_g \{R_\ell(g, \mathbb{P}) : g \text{ measurable}\}.$$

In practice, the distribution $\mathbb{P}$ is not given explicitly, but instead we only have access to historical data $H_n = \{(w_i, c_i) : i \in [n]\}$. To build a predictor $g : W \to C$, we optimize the empirical surrogate risk

$$\hat{R}_\ell(g, H_n) := \frac{1}{n} \sum_{i=1}^{n} \ell(g(w_i), c_i).$$

Statistical learning theory has rich literature on relating $\hat{R}_\ell$ to R_ℓ; see, e.g., Bousquet et al. [38]. In particular, for well-chosen classes of predictors $\mathcal{G}$, the following consistency result holds:

$$\hat{g}_n = \arg\min_{g \in \mathcal{G}} \hat{R}_\ell(g, H_n), \quad R_\ell(\hat{g}_n, \mathbb{P}) \to R_\ell(\mathbb{P}) \text{ in probability},$$

where convergence in probability is used due to the randomness in H_n, which translates to randomness of $\hat{g}_n$. This states that, for large n, we can get high-probability bounds on the excess surrogate risk $R_\ell(\hat{g}_n, \mathbb{P}) - R_\ell(\mathbb{P})$ of a predictor $\hat{g}_n$.

However, since we use $\hat{g}_n$ for optimization, we are actually interested in the excess true risk $R(\hat{g}_n, \mathbb{P}) - R(\mathbb{P})$ which depends on (6.1) explicitly. Thus, in this chapter, we give relationships between the excess surrogate risk and the true risk. This will then allow us to translate guarantees on the excess surrogate risk, of which many can be derived from statistical learning theory, to the excess true risk.

Our study on the relationship between the surrogate and true risks depends on understanding the relationship between the losses ℓ and L. Note that ℓ need not contain any information about the optimization problem (6.1). Further note that our results do not depend on the class of predictors $\mathcal{G}$. While studying $\mathcal{G}$ is important to relate the empirical surrogate risk $\hat{R}_\ell$ to the population surrogate risk R_ℓ, as well as for computational considerations of optimizing the surrogate risk, it turns out that it is secondary to understanding the relationship between R_ℓ and R, and our results are independent of the choice of $\mathcal{G}$. This is important for the application of our theory: by keeping the class $\mathcal{G}$ unspecified, our results are applicable to all settings, regardless of $\mathcal{G}$.

6.3 Risk Minimization via Admissible Surrogate Loss Functions

Given n data points, existing statistical learning results give us methods to obtain a predictor $g_n : W \to C$ with certifiable upper bounds on the excess surrogate risk $R_\ell(g_n, \mathbb{P}) - R_\ell(\mathbb{P})$, and in particular, as we obtain more data (i.e., as n grows), the upper bounds shrink to 0. Ideally, however, as $n \to \infty$, we also want the excess true risk to shrink, i.e., $R(g_n, \mathbb{P}) - R(\mathbb{P}) \to 0$. In order to understand the kind of results that we are after, let us explore the negation of this. In this case, we have $R_\ell(g_n, \mathbb{P}) - R_\ell(\mathbb{P}) \to 0$ but, for some $\epsilon > 0$, $R(g_n, \mathbb{P}) - R(\mathbb{P}) > \epsilon$ for infinitely many n. In other words, there exists $\epsilon > 0$ such that for all $\delta > 0$, there exists g_n such that $R_\ell(g_n, \mathbb{P}) - R_\ell(\mathbb{P}) \leq \delta$ but $R(g_n, \mathbb{P}) - R(\mathbb{P}) > \epsilon$. To prevent this bad outcome, we want to guarantee the following relationship between the risks:

$$\begin{aligned} &\text{for all } \epsilon > 0, \text{ there exists } \delta > 0 \text{ such that:} \\ &\quad \text{if } g : W \to C \text{ satisfies } R_\ell(g, \mathbb{P}) - R_\ell(\mathbb{P}) \leq \delta, \text{ then } R(g, \mathbb{P}) - R(\mathbb{P}) \leq \epsilon. \end{aligned} \tag{6.6}$$

We will show that (6.6) can be guaranteed by checking a simpler condition on the losses ℓ and L called *calibration*. This was introduced by Bartlett et al. [10] for binary classification and extended by Steinwart [143] for other machine learning applications. We extend this concept to the context of prediction and optimization.

Definition 6.5. A surrogate loss function ℓ for L is *calibrated* with respect to a distribution $\mathbb{P}$, or $\mathbb{P}$-*calibrated*, if, for all $w \in W$ and $\epsilon > 0$, there exists $\delta > 0$ (which may depend on w) such that

$$\text{if } d \in \mathbb{R}^m \text{ satisfies } \mathbb{E}[\ell(d,c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d,c) \mid w] \leq \delta, \text{ then } \mathbb{E}[L(d,c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d,c) \mid w] \leq \epsilon.$$

Observe that Definition 6.5 is very similar to (6.6), except that predictors g (i.e., functions mapping onto vectors) are replaced with vectors $d \in \mathbb{R}^m$. The task now is to translate this property on the behaviour of ℓ on vectors to functions $g : W \to C$, for which Steinwart [143, Theorem 2.8] gives a proof technique to translate $\mathbb{P}$-calibration to risk bounds. We apply this technique to give our main result, Theorem 6.7 below. We use the following technical assumption:

Assumption 6.6. Let the probability distribution $\mathbb{P}$ and the surrogate loss function ℓ be given. For any fixed $c \in C$, the surrogate loss function $\ell(d,c)$ is convex in $d \in \mathbb{R}^m$. For any $w \in W$ and $d \in \mathbb{R}^m$, the set $\arg\min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d',c) \mid w]$ is non-empty and bounded, and $\mathbb{E}[\ell(d,c) \mid w] < \infty$. Furthermore, c is an integrable random vector (that is, each component is integrable) so that $\mathbb{E}[\|c\|_1] < \infty$.

Theorem 6.7. *Suppose that ℓ is $\mathbb{P}$-calibrated, and that Assumption 6.6 holds. Then for all $\epsilon > 0$, there exists a $\delta > 0$ such that*

$$R_\ell(g, \mathbb{P}) \leq R_\ell(\mathbb{P}) + \delta \implies R(g, \mathbb{P}) \leq R(\mathbb{P}) + \epsilon.$$

We give the proof, together with the verification of the necessary technicalities, in Section 6.3.2.

6.3.1 A Note on the Regularity of X^* and x^*

An important property that we exploit is that the argmin mapping $X^*(d)$ is, in a sense, well-behaved as we change d. More precisely, the sense of regularity that we use is upper semicontinuity, which stems from a result in perturbation analysis [36].

Definition 6.8. A multivalued function $F : \mathbb{R}^m \to 2^X$ is *upper semi-continuous* at a point $d \in \mathbb{R}^m$ if, for any open set U containing $F(d)$, there exists an open set U_d containing d such that for all $d' \in U_d$, $F(d') \subseteq U$. Equivalently, F is upper semi-continuous if, for any closed set V, the following set is closed:

$$\{d \in \mathbb{R}^m : F(d) \cap V \neq \emptyset\}.$$

Lemma 6.9. *Suppose X is compact. Then the multivalued mapping $X^* : \mathbb{R}^m \to 2^X$ is upper semi-continuous.*

Proof. This follows immediately from verifying the conditions of Bonnans and Shapiro [36, Proposition 4.4], which are straightforward to check due to the fact that the domain X does not change with the vector d. □

We can use Lemma 6.9 to show the existence of a measurable selection $x^*(d) \in X^*(d)$ via an application of the Kuratowski–Ryll-Nardzewski theorem on the existence of measurable selectors for multivalued mappings. We use the version stated in Bogachev [35, Theorem 6.9.3].

Lemma 6.10. *Suppose X is compact. Then there exists a measurable mapping $x^* : \mathbb{R}^m \to X$ such that $x^*(d) \in X^*(d)$ for all $d \in \mathbb{R}^m$.*

Proof. Consider the multivalued function $X^* : \mathbb{R}^m \to 2^X$ defined by $X^*(d) = \arg\min_{x \in X} d^\top x$. Note that since $d^\top x$ is continuous, $X^*(d) = \{x \in X : d^\top x = \min_{x' \in X} d^\top x'\}$ is closed (it is the inverse of a singleton). Now consider an open set U, and the sets

$$\hat{X}^*(U) := \{d \in \mathbb{R}^m : X^*(d) \cap U \neq \emptyset\}.$$

It is known that U can be represented as the countable union of closed sets: $U = \bigcup_{k \in \mathbb{N}} V_k$ where V_k are closed. Thus, we can write

$$\hat{X}^*(U) = \{d \in \mathbb{R}^m : \exists k \in \mathbb{N} \text{ s.t. } X^*(d) \cap V_k \neq \emptyset\} = \bigcup_{k \in \mathbb{N}} \{d \in \mathbb{R}^m : X^*(d) \cap V_k \neq \emptyset\}.$$

Now, since $X^*(d)$ is upper semicontinuous, $\{d \in \mathbb{R}^m : X^*(d) \cap U_k \neq \emptyset\}$ is closed, hence $\hat{X}^*(U)$ is a countable union of closed sets, hence measurable. This shows that $X^*(\cdot)$ satisfies the conditions of Bogachev [35, Theorem 6.9.3], therefore there exists a measurable selection $x^*(d) \in X^*(d)$ for all $d \in \mathbb{R}^m$. □

Furthermore, we can show that *any* selection x^* is at least *Lebesgue* measurable, using the following result of Drusvyatskiy and Lewis [56].

Lemma 6.11 (Drusvyatskiy and Lewis [56, Corollary 3.5])**.** *The set*

$$D := \{d \in \mathbb{R}^m : X^*(d) \text{ is not a singleton}\}$$

has Lebesgue measure zero.

Lemma 6.12. *Any selection $x^* : \mathbb{R}^m \to X$ such that $x^*(d) \in X^*(d)$ for all $d \in \mathbb{R}^m$ is Lebesgue measurable.*

Proof. Lemma 6.10 tells us that there exists one such measurable selection $\bar{x}^*$. Consider another selection x^*. Then by Lemma 6.11, $\bar{x}^*$ and x^* differ on at most a set D with Lebesgue measure 0, which is Lebesgue measurable. Furthermore, all subsets of D are also Lebesgue measurable, so x^* must be Lebesgue measurable. □

For the rest of the chapter, in order for our expectations to be well-defined, we make the following assumption.

Assumption 6.13. Any probability distribution $\mathbb{P}$ is defined on the σ-algebra of Lebesgue measurable sets.

This is not restrictive, since any probability distribution we encounter in practice can be written as a mixture of a distribution which is absolutely continuous with respect to Lebesgue measure (i.e., it has a density function), and a discrete distribution supported on a countable set. Such a probability distribution is Lebesgue measurable.

6.3.2 Proof of Theorem 6.7

Define

$$\delta_\ell(\epsilon, w; \mathbb{P}) := \inf_{d \in \mathbb{R}^m} \left\{ \mathbb{E}[\ell(d,c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d',c) \mid w] : \mathbb{E}[L(d,c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d',c) \mid w] > \epsilon \right\}. \tag{6.7}$$

Note that if ℓ is $\mathbb{P}$-calibrated, then $\delta_\ell(\epsilon, w; \mathbb{P}) > 0$ for all $\epsilon > 0, w \in W$. In order to prove Theorem 6.7, we first verify measurability for δ_ℓ, since later we will be integrating it.

Lemma 6.14. *Suppose ℓ is measurable and satisfies Assumption 6.6, and that X is compact. For any $\epsilon > 0$, the function $\delta_\ell(\epsilon, \cdot; \mathbb{P}) : W \to \mathbb{R}$ is measurable.*

Proof. Consider the set

$$W_r := \{w \in W : \delta_\ell(\epsilon, w; \mathbb{P}) \le r\}.$$

Showing measurability of $\delta_\ell(\epsilon, \cdot; \mathbb{P})$ boils down to showing that W_r is measurable. Rewrite

$$\begin{aligned} W_r &= \left\{ w \in W : \forall k \in \mathbb{N},\ \exists d \in \mathbb{R}^m \text{ s.t. } \begin{array}{c} \mathbb{E}[\ell(d,c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d',c) \mid w] \le r + 1/k \\ \mathbb{E}[L(d,c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d',c) \mid w] > \epsilon \end{array} \right\} \\ &= \bigcap_{k \in \mathbb{N}} \left\{ w \in W : \exists d \in \mathbb{R}^m \text{ s.t. } \begin{array}{c} \mathbb{E}[\ell(d,c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d',c) \mid w] \le r + 1/k \\ \mathbb{E}[L(d,c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d',c) \mid w] > \epsilon \end{array} \right\} \end{aligned}$$

To this end, first consider the subset

$$\begin{aligned} W_L(\epsilon) &= \left\{ (w,d) \in W \times \mathbb{R}^m : \mathbb{E}[L(d,c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d',c) \mid w] > \epsilon \right\} \\ &= \left\{ (w,d) \in W \times \mathbb{R}^m : f(x^*(d)) + \mathbb{E}[c \mid w]^\top x^*(d) - \min_{x \in X} \left\{ f(x) + \mathbb{E}[c \mid w]^\top x \right\} > \epsilon \right\}. \end{aligned}$$

This is measurable since $\mathbb{E}[c \mid w]$ is measurable in w by definition of conditional expectation, f is continuous hence measurable, and we have assumed $x^*(d)$ is measurable in d, which is possible by Lemma 6.10.

Now consider the subset

$$W_\ell(\alpha) = \left\{ (w,d) \in W \times \mathbb{R}^m : \mathbb{E}[\ell(d,c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d',c) \mid w] \le \alpha \right\}.$$

First observe that the function h defined by $h(w,d) = \mathbb{E}[\ell(d,c) \mid w]$ is continuous in d and measurable in w. Continuity in d follows because $\ell(d,c)$ is convex in d, and $h(w,d)$ is finite for any w by Assumption 6.6, and all convex functions are continuous in the relative interiors of their domains (see e.g., Rockafellar [128, Theorem 10.1]). Measurability follows from measurability of ℓ and the definition of conditional expectation.

We now show that h is jointly measurable in (w,d) by showing that it is a pointwise limit of measurable functions. For $k \in \mathbb{N}$, consider the box $B_k := [-k,k]^m \subset \mathbb{R}^m$ and a finite set of grid points $G_k \subset B_k$ such that any point $d \in B_k$ is at most distance $1/k$ away from a grid point in Euclidean norm. If $d \in B_k$, define $h_k(w,d) = h(w,g)$ where $g \in B_k$ is the closest grid point to d (with ties broken arbitrarily), and if $d \notin B_k$ define $h_k(w,d) = 0$. Note that fixing g,

$w \mapsto h_g(w) := h(w, g)$ is measurable in w. Now, h_k is the sum of finitely many functions of the form $\mathbf{1}_D(d)h_g(w)$ for some measurable set D and grid point g. It is easy to check that this is measurable, therefore h_k is measurable. Furthermore, by continuity of h in d, $h_k(w,d) \to h(w,d)$ pointwise. Therefore, h is measurable. Finally, the function $(w,d) \mapsto \min_{d' \in \mathbb{R}^m} h(w,d')$ is measurable because by continuity of h in d, we can write

$$\left\{(w,d) : \min_{d' \in \mathbb{R}^m} h(w,d') \le \alpha\right\} = \bigcup_{d' \in D_*, k \in \mathbb{N}} \left\{(w,d) : h(w,d') \le \alpha + 1/k\right\}$$

where D_* is a countable dense subset of $\mathbb{R}^m$ (e.g., $\mathbb{Q}^m$). This shows that $W_\ell(\alpha)$ is measurable because the function $h(w,d) - \min_{d' \in \mathbb{R}^m} h(w,d') = \mathbb{E}[\ell(d,c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d',c) \mid w]$ is measurable.

Now notice that the set

$$\left\{(w,d) \in W \times \mathbb{R}^m : \begin{array}{c} \mathbb{E}[\ell(d,c) \mid w] - \min\limits_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d',c) \mid w] \le r + 1/k \\ \mathbb{E}[L(d,c) \mid w] - \min\limits_{d' \in \mathbb{R}^m} \mathbb{E}[L(d',c) \mid w] > \epsilon \end{array}\right\} = W_\ell(r + 1/k) \cap W_L(\epsilon)$$

is measurable. Therefore, its projection onto W is measurable, which is

$$\left\{w \in W : \exists d \in \mathbb{R}^m \text{ s.t. } \begin{array}{c} \mathbb{E}[\ell(d,c) \mid w] - \min\limits_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d',c) \mid w] \le r + 1/k \\ \mathbb{E}[L(d,c) \mid w] - \min\limits_{d' \in \mathbb{R}^m} \mathbb{E}[L(d',c) \mid w] > \epsilon \end{array}\right\}.$$

This shows that W_r is measurable, concluding our proof. □

Proof of Theorem 6.7. Fix a predictor $g : W \to C$. For w such that $\mathbb{E}[L(g(w),c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d',c) \mid w] > \epsilon$, by $\mathbb{P}$-calibration we have $\mathbb{E}[\ell(g(w),c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d',c) \mid w] > \delta_\ell(\epsilon, w; \mathbb{P})$. Therefore, defining $E_g(\epsilon) := \{w \in W : \mathbb{E}[L(g(w),c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d',c) \mid w] > \epsilon\}$, we have

$$\begin{aligned} R_\ell(g, \mathbb{P}) - R_\ell(\mathbb{P}) &= \mathbb{E}\left[\mathbb{E}[\ell(g(w),c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d',c) \mid w]\right] \\ &\ge \int_{w \in E_g(\epsilon)} \delta_\ell(\epsilon, w; \mathbb{P}) d\mathbb{P}(w). \end{aligned}$$

Let Ω be the ℓ_∞-diameter of the set X, which is finite since X is compact. Observe that for any $d \in \mathbb{R}^m$,

$$\begin{aligned} \mathbb{E}[L(d,c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d,c) \mid w] &= f(x^*(d)) + \mathbb{E}[c \mid w]^\top x^*(d) - \min_{x \in X} \left\{f(x) + \mathbb{E}[c \mid w]^\top x\right\} \\ &\le \max_{x, x' \in X} \left\{f(x) - f(x') + \mathbb{E}[c \mid w]^\top (x' - x)\right\} \\ &\le \max_{x, x' \in X} \left\{f(x) - f(x') + \|\mathbb{E}[c \mid w]\|_1 \|x' - x\|_\infty\right\} \\ &\le \Omega \|\mathbb{E}[c \mid w]\|_1 + \max_{x, x' \in X} \left\{f(x) - f(x')\right\}. \end{aligned}$$

Let us now define two (non-negative) measures on W as

$$\mu(E) = \int_{w \in E} \left(\Omega \|\mathbb{E}[c \mid w]\|_1 + \max_{x, x' \in X} \left\{f(x) - f(x')\right\}\right) d\mathbb{P}(w), \quad \mu_\ell(E) = \int_{w \in E} \delta_\ell(\epsilon, w; \mathbb{P}) d\mathbb{P}(w).$$

We now show that μ is absolutely continuous with respect to μ_ℓ. Fix some $E \subseteq W$ such that $\mu_\ell(E) = 0$. Consider $E_t := \{w \in E : \delta_\ell(\epsilon, w; \mathbb{P}) \geq \left(\Omega \|\mathbb{E}[c \mid w]\|_1 + \max_{x,x' \in X} \{f(x) - f(x')\}\right)/t\}$ for $t \in \mathbb{N}$. Since $\delta_\ell(\epsilon, w; \mathbb{P})$, we have $E_t \uparrow E$ as $t \to \infty$, i.e., every point $w \in E$ is eventually in E_t for sufficiently large t. So by the monotone convergence theorem, $\mu(E_t) \uparrow \mu(E)$ as $t \to \infty$. But now observe that

$$\begin{aligned} 0 = \mu_\ell(E) \geq \mu_\ell(E_t) &= \int_{w \in E_t} \delta_\ell(\epsilon, w; \mathbb{P}) d\mathbb{P}(w) \\ &\geq \frac{1}{t} \int_{w \in E_t} \left(\Omega \|\mathbb{E}[c \mid w]\|_1 + \max_{x,x' \in X} \{f(x) - f(x')\} \right) d\mathbb{P}(w) = \frac{1}{t} \mu(E_t), \end{aligned}$$

where the first inequality follows from $E_t \subseteq E$ for all $t \in \mathbb{N}$, and the second inequality follows from the definition of E_t. Therefore, each $\mu(E_t) = 0$ for any $t \in \mathbb{N}$, hence $\mu(E) = 0$. Note that, as a function of w, $\Omega \|\mathbb{E}[c \mid w]\|_1 + \max_{x,x' \in X} \{f(x) - f(x')\}$ is integrable since the conditional expectation $\mathbb{E}[c \mid w]$ is integrable (which follows since c is integrable), therefore $\mu(W) < \infty$, hence μ is a finite measure. Stein and Shakarchi [139, Chapter 6, Proposition 4.2] implies that for all $\epsilon > 0$, there exists $\delta > 0$ such that

$$\mu_\ell(E) \leq \delta \implies \mu(E) \leq \epsilon.$$

Now, for a function $g : W \to C$, and $w \in W$ such that $\mathbb{E}[L(g(w), c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d', c) \mid w] > \epsilon$, by $\mathbb{P}$-calibration we have $\mathbb{E}[\ell(g(w), c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d', c) \mid w] > \delta_\ell(\epsilon, w; \mathbb{P})$. Define

$$E_g(\epsilon) := \left\{ w \in W : \mathbb{E}[L(g(w), c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d', c) \mid w] > \epsilon \right\}.$$

We have

$$R_\ell(g, \mathbb{P}) - R_\ell(\mathbb{P}) = \mathbb{E}\left[\mathbb{E}[\ell(g(w), c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d', c) \mid w] \right] \geq \int_{w \in E_g(\epsilon)} \delta_\ell(\epsilon, w; \mathbb{P}) d\mathbb{P}(w) = \mu_\ell(E_g(\epsilon)).$$

Thus, when $R_\ell(g, \mathbb{P}) - R_\ell(\mathbb{P}) \leq \delta$, we have $\mu_\ell(E_g(\epsilon)) \leq \delta$, hence $\mu(E_g(\epsilon)) \leq \epsilon$, so

$$\begin{aligned} R(g, \mathbb{P}) - R(\mathbb{P}) &= \mathbb{E}\left[\mathbb{E}[L(g(w), c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d, c) \mid w] \right] \\ &= \int_{w \in E_g(\epsilon)} \left(\mathbb{E}[L(g(w), c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d, c) \mid w] \right) d\mathbb{P}(w) \\ &\quad + \int_{w \in W \setminus E_g(\epsilon)} \left(\mathbb{E}[L(g(w), c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d, c) \mid w] \right) d\mathbb{P}(w) \\ &\leq \int_{w \in E_g(\epsilon)} \left(\Omega \|\mathbb{E}[c \mid w]\|_1 + \max_{x,x' \in X} \{f(x) - f(x')\} \right) d\mathbb{P}(w) + \int_{w \in W \setminus E_g(\epsilon)} \epsilon d\mathbb{P}(w) \\ &\leq \mu(E_g(\epsilon)) + \epsilon \\ &\leq 2\epsilon. \end{aligned}$$

□

6.3.3 Admissible Loss Functions

In general, checking that a given surrogate loss ℓ is $\mathbb{P}$-calibrated may not be straightforward. A much simpler condition to check is admissibility, also known as Fisher consistency, defined in Definition 6.15 below, which relates to the *minimizers* of the loss functions, instead of approximate minimizers as in Definition 6.5. In this section, we show that admissibility is equivalent to calibration, thus allowing us to check the simpler condition to verify Theorem 6.7. We also discuss some different loss functions and their admissibility properties.

Definition 6.15. A surrogate loss function ℓ is *admissible* with respect to a distribution $\mathbb{P}$, or $\mathbb{P}$-*admissible*, if for all w,

$$\arg\min_d \mathbb{E}\left[\ell(d,c) \mid w\right] \subseteq \arg\min_d \mathbb{E}\left[L(d,c) \mid w\right].$$

The following theorem shows that admissibility is equivalent to calibration. Of course, the fact that calibration implies admissibility is straightforward; the challenge is to show the other direction.

Theorem 6.16. *Given a distribution $\mathbb{P}$, let $\ell(d,c)$ be loss function that satisfies Assumption 6.6. Then ℓ is $\mathbb{P}$-calibrated if and only if ℓ is $\mathbb{P}$-admissible.*

The key tool that we exploit is upper semi-continuity of the argmin mapping $X^*(\cdot)$. Informally, this states that if we are given $X^*(d)$ for some vector d, and we are interested in vectors d' for which $X^*(d')$ does not move 'too far away' from $X^*(d)$, then we can guarantee that when d' is sufficiently close to d, this will indeed be the case. In particular, in the context of proving Theorem 6.16, we use this to show that when $\mathbb{E}[L(d,c) \mid w]$ is large, then vectors close by to d will also have large true expected loss.

Proof. Denote

$$D_\ell(\alpha; w) := \left\{ d \in \mathbb{R}^m : \mathbb{E}[\ell(d,c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d,c) \mid w] \leq \alpha \right\}$$
$$D(\alpha; w) := \left\{ d \in \mathbb{R}^m : \mathbb{E}[L(d,c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d,c) \mid w] \leq \alpha \right\}.$$

Note that

$$\arg\min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d',c) \mid w] = \bigcap_{\alpha > 0} D_\ell(\alpha; w), \quad \arg\min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d',c) \mid w] = \bigcap_{\alpha > 0} D(\alpha; w).$$

Suppose first that ℓ is $\mathbb{P}$-calibrated. Then for any $\epsilon > 0$, there exists $\delta > 0$ (which can depend on w) such that $D_\ell(\delta; w) \subseteq D(\epsilon; w)$. In particular, since $D_\ell(\alpha; w) \subseteq D_\ell(\alpha'; w)$ for $\alpha \leq \alpha'$, we have

$$\arg\min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d',c) \mid w] = \bigcap_{0 < \alpha \leq \delta} D_\ell(\alpha; w) \subseteq D(\epsilon; w).$$

Taking the intersection of the right hand side over $\epsilon > 0$, we have

$$\arg\min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d',c) \mid w] \subseteq \arg\min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d',c) \mid w],$$

hence ℓ is $\mathbb{P}$-admissible.

Suppose now that ℓ is not $\mathbb{P}$-calibrated. We show that it is also not $\mathbb{P}$-admissible. Fix an arbitrary $w \in W$. Note that the function $h : \mathbb{R}^m \to \mathbb{R}$ defined by $h(d) = \mathbb{E}[\ell(d, c) \mid w]$ is convex by convexity of $\ell(d, c)$, and hence under Assumption 6.6, it is continuous (see e.g., Rockafellar [128, Theorem 10.1]).

Since ℓ is not $\mathbb{P}$-calibrated, there exists $w \in W$ and $\epsilon > 0$ such that for all $\delta > 0$, there exists $d(\delta) \in \mathbb{R}^m$ such that $h(d(\delta)) - \min_{d' \in \mathbb{R}^m} h(d') \leq \delta$ but $\mathbb{E}[L(d(\delta), c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d, c) \mid w] > \epsilon$.

Now, let $d_k = d(1/k)$ for $k \in \mathbb{N}$. Note that $\{d_k\}_{k \in \mathbb{N}} \subset D_\ell(1; w)$ which is compact since by Assumption 6.6 $\arg\min_{d' \in \mathbb{R}^m} h(d')$ is compact, so all level sets are compact (see, e.g., Rockafellar [128, Corollary 8.7.1]). Therefore, there exists a convergent subsequence $d'_k \to d$. Since h is continuous, we must have $d \in \arg\min_{d' \in \mathbb{R}^m} h(d')$.

We now want to show that $d \notin \arg\min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d', c) \mid w]$. We know from Lemma 6.9 that the argmin mapping $X^*(\cdot)$ is upper semi-continuous at d. Suppose for contradiction that $d \in \arg\min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d', c) \mid w]$. Then we must have $X^*(d) \subseteq X^*(\mathbb{E}[c \mid w])$. Thus, for $\epsilon > 0$ the set

$$X^\circ(\epsilon') = \left\{ x' : f(x') + \mathbb{E}[c \mid w]^\top x' < \min_{x \in X} \left\{ f(x) + \mathbb{E}[c \mid w]^\top x \right\} + \epsilon' \right\}$$

is an open* set (as $x \mapsto f(x) + \mathbb{E}[c \mid w]^\top x$ is continuous) containing $X^*(d)$. Then, by Definition 6.8 of upper semi-continuity, there exists a neighbourhood $D^\circ(\epsilon')$ of d such that for any $d^\circ \in D^\circ(\epsilon')$, $X^*(d^\circ) \subset X^\circ(\epsilon')$, which means that $\mathbb{E}[L(d^\circ, c) \mid w] < \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d', c) \mid w] + \epsilon'$ since $x^*(d^\circ) \in X^*(d^\circ) \subseteq X^\circ(\epsilon')$.

But now consider $\epsilon' < \epsilon$. Since $d'_k \to d$, $D^\circ(\epsilon')$ is open, and $d \in D^\circ(\epsilon')$, we eventually have $d'_k \in D^\circ(\epsilon')$ for sufficiently large k. But this contradicts the fact that by construction of the sequence $\{d_k\}_{k \in \mathbb{N}}$ we have $\epsilon' < \epsilon < \mathbb{E}[L(d'_k, c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d', c) \mid w] = \mathbb{E}[c \mid w]^\top x^*(d'_k) - \min_{x \in X} \mathbb{E}[c \mid w]^\top x$. □

Armed with Theorem 6.16, we have the following corollaries, which are straightforward consequences of previous results.

Corollary 6.17. *Suppose that ℓ is $\mathbb{P}$-admissible, and that Assumption 6.6 holds. Then for all $\epsilon > 0$, there exists a $\delta > 0$ such that*

$$R_\ell(g, \mathbb{P}) \leq R_\ell(\mathbb{P}) + \delta \implies R(g, \mathbb{P}) \leq R(\mathbb{P}) + \epsilon.$$

Corollary 6.18. *Suppose that ℓ is $\mathbb{P}$-admissible, and that Assumption 6.6 holds. If we have a sequence of functions g_n such that $R_\ell(g_n, \mathbb{P}) \to R_\ell(\mathbb{P})$. Then $R(g_n, \mathbb{P}) \to R(\mathbb{P})$.*

Proof. Fix some $\epsilon > 0$. Take $\delta > 0$ corresponding to ϵ in Corollary 6.17. Since $R_\ell(g_n, \mathbb{P}) \to R_\ell(\mathbb{P})$, we have $R_\ell(g_n, \mathbb{P}) \leq R_\ell(\mathbb{P}) + \delta$ eventually. By Theorem 6.17, we will also have $R(g_n, \mathbb{P}) \to R(\mathbb{P}) + \epsilon$ eventually. □

We now examine several different loss functions and their admissibility properties. Before doing so, let us summarize the properties on ℓ and $\mathbb{P}$ in order to get risk guarantees of the form (6.6). These are:

1. the surrogate loss $\ell(\cdot, c)$ is convex for any fixed $c \in C$.

*Note that this is not open in $\mathbb{R}^m$ by the usual topology, since $f(x)$ may be infinite for $x \notin X$. However, it is open when we work with $X \subset \mathbb{R}^m$ as the *entire* topological space with the induced topology from $\mathbb{R}^m$.

2. for any $w \in W, d \in \mathbb{R}^m$, the expected loss $\mathbb{E}[\ell(d,c) \mid w]$ is finite.

3. for any $w \in W$, the set of minimizers $\arg\min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d',c) \mid w]$ is non-empty and bounded.

4. the surrogate loss ℓ is $\mathbb{P}$-admissible according to Definition 6.15.

We first examine the loss function that is admissible for any class of distributions.

Example 6.19. Consider the squared loss $\ell(d,c) = \|d-c\|_2^2$. Then ℓ is $\mathbb{P}$-admissible for any distribution $\mathbb{P}$ over $W \times C$. Note that

$$\mathbb{E}[\ell(d,c) \mid w] = \mathbb{E}\left[\|d-c\|_2^2\right] = \|d - \mathbb{E}[c \mid w]\|_2^2 + \mathbb{E}[\|c\|_2^2 \mid w] - \|\mathbb{E}[c \mid w]\|_2^2.$$

Thus, the unique minimizer of $\mathbb{E}[\ell(d,c) \mid w]$ is $d^* = \mathbb{E}[c \mid w]$. Since we know this is also a minimizer of $\mathbb{E}[L(d,c) \mid w]$, this gives us $\mathbb{P}$-admissibility of the squared loss.

Also note that property 1 and 3 are clearly satisfied. Property 2 will be satisfied if the conditional distribution $\mathbb{P}[\cdot \mid w]$ is square integrable for every $w \in W$. ∎

A common loss function used in regression to safeguard against outliers is the absolute deviation loss.

Example 6.20. Consider the absolute deviation loss $\ell(d,c) = \|d-c\|_1$. Then we claim that ℓ is $\mathbb{P}$-admissible as long as, for every w, $\mathbb{P}[\cdot \mid w]$ is centrally symmetric about some vector d_w. A distribution $\mathbb{P}$ is centrally symmetric about d if, for a random variable $c \sim \mathbb{P}$, $c-d$ has the same distribution as $d-c$. Note that $\arg\min_{d' \in \mathbb{R}^m} \mathbb{E}[\|d'-c\|_1 \mid w]$ recovers the vector of coordinate-wise medians, which for a centrally symmetric distribution will be the point of symmetry d_w, i.e., d_w minimizes $\mathbb{E}[\|d-c\|_1 \mid w]$. Furthermore, we have $\mathbb{E}[c \mid w] = d_w$ also. Therefore d_w minimizes $\mathbb{E}[L(d,c) \mid w]$. ∎

We now discuss the loss function proposed in Elmachtoub and Grigas [58], which aims to incorporate knowledge of the domain X into the loss, in the hopes of achieving low true risk R, which is based on the optimization problem.

Example 6.21. In the setting when $f(x) = 0$ for all $x \in X$, Elmachtoub and Grigas [58, Definition 3] defined the following loss function:

$$\ell(d,c) = (2d-c)^\top x^*(c) - \min_{x \in X} (2d-c)^\top x. \tag{6.8}$$

Elmachtoub and Grigas [58, Theorem 1] shows that ℓ is admissible as long as $\mathbb{P}[c \mid w]$ is centrally symmetric and continuous. We remark also that Elmachtoub and Grigas [58] achieve good numerical results, particularly when the hypothesis class is misspecified versus the true distribution. ∎

We now highlight some positive and negative aspects of the loss function of Elmachtoub and Grigas [58]. An important observation made in Elmachtoub and Grigas [58, Proposition 1] is that, by carefully choosing the set C and domain X, the true loss L from (6.2) becomes the 0-1 loss from binary classification. The loss in (6.8) also has a familiar interpretation in this setting.

Example 6.22. Let $m = 1$, $C = \{-1, 1\}$, $X = [-1/2, 1/2]$ and $f(x) = 0$ for all $x \in X$. Then the 0-1 loss for classification is exactly equivalent to the true loss function L. More precisely, note that $x^*(d) = -\operatorname{sign}(d)/2$, and $\min_{x \in X} c^\top x = -1/2$ for any $c \in C$, so

$$L(d,c) = \frac{c \operatorname{sign}(d) - 1}{2} = \begin{cases} 0, & c = \operatorname{sign}(d) \\ 1, & c \neq \operatorname{sign}(d). \end{cases}$$

Elmachtoub and Grigas [58, Proposition 3] shows that the loss from (6.8) reduces to the hinge loss in this case: since $x^*(c) = -c/2$ for $c \in C$ and $\min_{x \in X} d^\top x = -|d|/2$,

$$\ell(d, c) = \frac{|2d - c| - (2d - c)c}{2} = \frac{|1 - 2dc| + 1 - 2dc}{2} = \max\{0, 1 - 2dc\}$$

Lin [101, Theorem 3.1] states that the hinge loss, and thus ℓ, is admissible for any distribution over $C = \{-1, 1\}$ except the uniform one. ∎

On the other hand, the loss function of Elmachtoub and Grigas [58] is not admissible for some very natural settings. We demonstrate this with two examples.

Example 6.23. Consider the setting where $m = 1$, $X = [-1/2, 1/2]$ and $f(x) = 0$ for all $x \in X$, but C is an arbitrary subset of $\mathbb{R}$. Then $x^*(c) = -\operatorname{sign}(c)/2$, $\min_{x \in X} d^\top x = -|d|/2$, hence the loss function from (6.8) becomes

$$\ell(d, c) = \frac{|2d - c| - (2d - c)\operatorname{sign}(c)}{2} = \frac{|2d - c| - 2d\operatorname{sign}(c) + |c|}{2}.$$

Let $\mathbb{P}$ be a distribution over $W \times C$. For any $w \in W$, note that the minimizers of $\mathbb{E}[L(d, c) \mid w]$ are $D^*_w = \{d \in \mathbb{R} : \operatorname{sign}(d) = \operatorname{sign}(\mathbb{E}[c \mid w])\}$. Thus, checking $\mathbb{P}$-admissibility requires showing that $\arg\min_{d' \in \mathbb{R}} \mathbb{E}[\ell(d', c) \mid w] \subseteq D^*_w$ for every $w \in W$, i.e., we need to show that the minimizers have the same sign as the mean $\mathbb{E}[c \mid w]$.

Let us now explore what $\arg\min_{d' \in \mathbb{R}} \mathbb{E}[\ell(d, c) \mid w]$ is for our setting. For convenience, we fix $w \in W$, and omit the w in the notation, so that $D^* = D^*_w$, $\mathbb{E}[\cdot] = \mathbb{E}[\cdot \mid w]$ and $\mathbb{P}[\cdot] = \mathbb{P}[\cdot \mid w]$. Then

$$2\mathbb{E}[\ell(d, c)] = \mathbb{E}[|2d - c|] - 2d\mathbb{E}[\operatorname{sign}(c)] + \mathbb{E}[|c|] = \mathbb{E}[|2d - c|] + 2d\left(\mathbb{P}[c < 0] - \mathbb{P}[c > 0]\right) + \mathbb{E}[|c|].$$

This is a convex function in d, so we look at the subdifferential to determine its minimizers. Note that

$$\partial_d \mathbb{E}[|2d - c|] = \left\{2\left(\mathbb{P}[c < 2d] - \mathbb{P}[c > 2d] + s\mathbb{P}[c = 2d]\right) : s \in [-1, 1]\right\},$$

so

$$\partial_d \mathbb{E}[\ell(d, c)] = \left\{\mathbb{P}[c < 2d] - \mathbb{P}[c > 2d] + \mathbb{P}[c < 0] - \mathbb{P}[c > 0] + s\mathbb{P}[c = 2d] : s \in [-1, 1]\right\}.$$

For simplicity, let us assume that $\mathbb{P}[c = 2d] = 0$ for any d (many such distributions exist). Then $\mathbb{E}[\ell(d, c)]$ is differentiable with

$$\nabla_d \mathbb{E}[\ell(d, c)] = \mathbb{P}[c < 2d] - \mathbb{P}[c > 2d] + \mathbb{P}[c < 0] - \mathbb{P}[c > 0].$$

Denote d^* to be a minimizer of $\mathbb{E}[\ell(d, c)]$. If $\mathbb{P}[c < 0] = \mathbb{P}[c > 0]$, then setting $d = 0$ gives $\nabla_d \mathbb{E}[\ell(d, c)] = 0$, so $d^* = c = 0$. If $\mathbb{P}[c < 0] - \mathbb{P}[c > 0] < 0$, then $\nabla_d \mathbb{E}[\ell(d, c)]|_{d=0} < 0$, so increasing d from 0 will decrease $\mathbb{E}[\ell(d, c)]$. Thus, $d^* > 0$. However, note that $\mathbb{P}[c < 0] - \mathbb{P}[c > 0] < 0$ implies that the median of c is also > 0. If $\mathbb{P}[c < 0] - \mathbb{P}[c > 0] > 0$, then $\nabla_d \mathbb{E}[\ell(d, c)]|_{d=0} > 0$, so decreasing d from 0 will decrease $\mathbb{E}[\ell(d, c)]$. Thus, $d^* < 0$. However, note that $\mathbb{P}[c < 0] - \mathbb{P}[c > 0] > 0$ implies that the median of c is also < 0. In all cases, the minimizer d^* is of the same sign as the median of c. Now, if $\mathbb{P}$ is a symmetric distribution, then the mean $\mathbb{E}[c]$ is equal to the median, and thus d^* has the same sign as $\mathbb{E}[c]$, so also minimizes $\mathbb{E}[L(d, c)]$. However, if the median has a different sign to the mean, then ℓ is not $\mathbb{P}$-admissible. Such distributions can be constructed by shifting a log-normal distribution, for example. ∎

Example 6.24. In Example 6.22, we showed that for appropriately chosen X, f and C, L specializes to the 0-1 loss for binary classification, and ℓ specializes to the hinge loss. Thus, ℓ defined in (6.8) can be seen as a generalization of the hinge loss for optimization problems. We show that this setting can also capture the multiclass classification loss, i.e., choosing X and C appropriately we can make L represent the 0-1 loss for multiclass classification. However, the generalization of hinge loss given by (6.8) to this setting is not admissible.

Suppose we have pairs (w, c), where w are features, and $c \in C'$ is a label from one of $m \in \mathbb{N}$ different classes, i.e., $C' = [m]$. We want a predictor $g' : W \to C'$ which classifies w according to $g'(w)$. If we classify w incorrectly (i.e., $g'(w)$ is in a different class to c) we suffer loss 1; otherwise, our loss is 0. We can capture this in our optimization framework as follows.

Consider $C = \{c_j := \mathbf{1}_m - e_j : j \in [m]\} \subset \mathbb{R}^m$, $X = \operatorname{Conv}\{e_j : j \in [m]\} \subset \mathbb{R}^m$ and $f(x) = 0$ for all $x \in X$. Then $\min_{x \in X} d^\top x = \min_{j' \in [m]} d_{j'}$, $\min_{x \in X} c_j^\top x = 0$ and $x^*(d) = e_j$ for $j \in \arg\min_{j' \in [m]} d_{j'}$, so for any $j \in [m]$ and vector d with unique minimum entry

$$L(d, c_j) = \begin{cases} 0, & \arg\min_{j' \in [m]} d_{j'} = j \\ 1, & \arg\min_{j' \in [m]} d_{j'} \neq j. \end{cases}$$

In other words, if we have a function $g : W \to \mathbb{R}^m$, we can use it to build a classifier $g' : W \to C'$ by classifying w according to the minimum entry of $g(w) \in \mathbb{R}^m$. Then L is exactly the 0-1 loss for this classifier. Suppose that we have a distribution $\mathbb{P}[c = c_j] = p_j > 0$, $\sum_{j \in [m]} p_j = 1$. Then, letting $j^*(d) = \arg\min_{j' \in [m]} d_{j'}$,

$$\mathbb{E}[L(d, c)] = 1 - p_{j^*(d)},$$

so the vectors d which minimize $\mathbb{E}[L(d, c)]$ must satisfy $j^*(d) \in \arg\max_{j' \in [m]} p_{j'}$.

The loss (6.8) becomes

$$\ell(d, c_j) = (2d - c_j)^\top e_j - \min_{j' \in [m]} \left\{2d_{j'} - c_{j'}\right\} = 2d_j - \min_{j' \in [m]} \left\{2d_{j'} - \mathbf{1}(j' \neq j)\right\}.$$

Then with the same distribution $\mathbb{P}$ as above, $\min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d', c)]$ can be expressed as the following linear program (making the change of variables $2d' = d$):

$$\begin{aligned} \min_{d, \gamma} \quad & \sum_{j \in [m]} p_j (d_j - \gamma_j) \\ \text{s.t.} \quad & \gamma_j \leq d_j, \ j \in [m] \\ & \gamma_j \leq d_k - 1, \ j, k \in [m], k \neq j \\ & d, \gamma \in \mathbb{R}^m. \end{aligned}$$

We analyse this linear program. Fix a vector $d \in \mathbb{R}^m$. Let $j^* \in \arg\min_{j' \in [m]} d_{j'}$. Then since $p_k > 0$ for all $k \neq j^*$, the optimal choice of γ_k makes it as large as possible, so we set $\gamma_k = d_{j^*} - 1$ for $k \neq j$. In other words, for all but one index $j^* \in \arg\min_{j' \in [m]} d_{j'}$, we set $\gamma_j = \min_{j' \in [m]} d_{j'} - 1$. For j^*, we set $\gamma_{j^*} = \min\left\{d_{j^*}, \min_{j' \neq j^*} d_{j'} - 1\right\}$.

If there exists $j \neq j^*$ such that $d_{j^*} \leq d_j - 1$, then decreasing $d_j \downarrow d_{j^*} + 1$ does not violate any constraints since $\gamma_j = d_{j^*} - 1 < d_j$ and $\gamma_{j^*} = d_{j^*} \leq d_j - 1$, and decreases the objective. Therefore, without loss of generality, we assume that $d_j - 1 \leq d_{j^*}$ for all $j \neq j^*$. This implies that $\gamma_{j^*} = \min_{j' \neq j^*} d_{j'} - 1$.

Furthermore, if we have $j, k \in [m] \setminus \{j^*\}$, $j \neq k$ such that $d_j < d_k$, note that we can decrease $d_k \downarrow d_j$ without violating any constraints, since $\gamma_{j'} = d_{j^*} - 1 \leq d_j - 1 < d_k - 1 < d_k$ for all $j' \neq j^*$ and $\gamma_{j^*} \leq d_j - 1 < d_k - 1$. This implies that, without loss of generality, we can assume that for $j \neq j^*$, we have $d_j = \delta$ for some $\delta \in [d_{j^*}, d_{j^*} + 1]$. In particular, this implies that $\gamma_{j^*} = \delta - 1$, thus the objective becomes

$$\sum_{j \in [m]} p_j (d_j - \gamma_j) = (\delta - d_{j^*} + 1) \sum_{j \neq j^*} p_j + p_{j^*}(d_{j^*} - \delta + 1) = (1 - 2p_{j^*})(\delta - d_{j^*}) + 1.$$

This shows that if $p_{j^*} > 1/2$, then we should make δ as large as possible, i.e., $\delta = d_{j^*} + 1$. On the other hand, when $p_{j^*} < 1/2$, we set $\delta = d_{j^*}$, i.e., the optimal vector d^* is constant.

This implies that, if there exists $j^* \in [m]$ such that $p_{j^*} > 1/2$, and necessarily $j^* = \arg\max_{j' \in [m]} p_{j'}$, then the minimizers of $\mathbb{E}[\ell(d, c)]$ take the form $d_\alpha = (\alpha \mathbf{1}_m - e_{j^*})/2$ for $\alpha \in \mathbb{R}$. Clearly, $\arg\min_{j' \in [m]} d_{\alpha, j'} = j^*$, so for such distributions $\mathbb{P}$, ℓ is $\mathbb{P}$-admissible.

On the other hand, for distributions $\mathbb{P}$ with $\max_{j' \in [m]} p_{j'} < 1/2$, ℓ is not $\mathbb{P}$-admissible, since the set of minimizers of $\mathbb{E}[\ell(d, c)]$ are the vectors $d_\alpha = \alpha \mathbf{1}_m$, $\alpha \in \mathbb{R}$, which cannot in general pick out the maximum probability class $j \in [m]$, i.e., the highest p_j. ∎

6.4 Non-Asymptotic Risk Guarantees via Uniform Calibration

Notice that Corollary 6.18 is an asymptotic result, that is, we only assert that minimizing the surrogate risk will minimize the true risk in the limit. This does not present much insight about the rate of convergence of these quantities, which is governed by the relationship between ϵ and δ in Corollary 6.17. Moreover, the δ in Corollary 6.17 depends on the distribution $\mathbb{P}$. In general, this is undesirable, since often in statistical learning, we assume minimal knowledge of $\mathbb{P}$. Furthermore, when given n data points $\{(w_i, c_i) : i \in [n]\}$ we can build a predictor g_n with quantified guarantees on the excess surrogate risk $R_\ell(g_n, \mathbb{P}) - R_\ell(\mathbb{P})$ via standard learning theoretic results. We would ideally like to translate these into quantified guarantees on the excess true risk $R(g_n, \mathbb{P}) - R(\mathbb{P})$.

Steinwart [143] builds a theory for non-asymptotic relationships between true and surrogate risk for various types of learning problems, such as classification, regression, and density estimation, giving necessary and sufficient conditions for the existence of distribution-independent guarantees. In this section, building on the results from Steinwart [143], we provide conditions for the existence of similar guarantees in the prediction and optimization context. Then, using these conditions, we identify a non-asymptotic distribution-independent guarantee between the risk of the surrogate squared loss function and the true optimzality gap risk.

6.4.1 Outline of the Key Idea

In this section, our aim is to identify an increasing function $\eta : [0, \infty) \to [0, \infty)$ with $\eta(0) = 0$ such that for any distribution $\mathbb{P}$, we have

$$\eta\left(R(g, \mathbb{P}) - R(\mathbb{P})\right) \leq R_\ell(g, \mathbb{P}) - R_\ell(\mathbb{P}).$$

Thus, any bound on the excess surrogate risk $R_\ell(g, \mathbb{P}) - R_\ell(\mathbb{P})$ translates to a bound on the excess true risk $R(g, \mathbb{P}) - R(\mathbb{P})$. Let us explore how we would derive such bounds. First, suppose that η is a convex function. Then, using Jensen's inequality,

$$\begin{aligned}
\eta\left(R(g, \mathbb{P}) - R(\mathbb{P})\right) &= \eta\left(\mathbb{E}\left[\mathbb{E}\left[L(g(w), c) \mid w\right] - \min_{d' \in \mathbb{R}^m} \mathbb{E}\left[L(d', c) \mid w\right]\right]\right) \\
&\leq \mathbb{E}\left[\eta\left(\mathbb{E}\left[L(g(w), c) \mid w\right] - \min_{d' \in \mathbb{R}^m} \mathbb{E}\left[L(d', c) \mid w\right]\right)\right].
\end{aligned}$$

Now, suppose that η and ℓ are chosen to ensure that, for any $w \in W$ and $d \in \mathbb{R}^m$,

$$\eta\left(\mathbb{E}\left[L(d,c) \mid w\right] - \min_{d' \in \mathbb{R}^m} \mathbb{E}\left[L(d',c) \mid w\right]\right) \leq \mathbb{E}\left[\ell(d,c) \mid w\right] - \min_{d' \in \mathbb{R}^m} \mathbb{E}\left[\ell(d',c) \mid w\right], \tag{6.9}$$

this implies that

$$\begin{aligned}
\eta\left(R(g,\mathbb{P}) - R(\mathbb{P})\right) &\leq \mathbb{E}\left[\eta\left(\mathbb{E}\left[L(g(w),c) \mid w\right] - \min_{d' \in \mathbb{R}^m} \mathbb{E}\left[L(d',c) \mid w\right]\right)\right] \\
&\leq \mathbb{E}\left[\mathbb{E}\left[\ell(g(w),c) \mid w\right] - \min_{d' \in \mathbb{R}^m} \mathbb{E}\left[\ell(d',c) \mid w\right]\right] \\
&= R_\ell(g,\mathbb{P}) - R_\ell(\mathbb{P}).
\end{aligned}$$

So, the task is to choose η and ℓ such that (6.9) holds. However, we have already seen an example where something similar holds, namely δ_ℓ defined in (6.7) when ℓ is $\mathbb{P}$-calibrated. Indeed, fixing $w \in W$, consider $d \in \mathbb{R}^m$ such that $\mathbb{E}\left[L(d,c) \mid w\right] - \min_{d' \in \mathbb{R}^m} \mathbb{E}\left[L(d',c) \mid w\right] = \epsilon$. Then

$$\begin{aligned}
\mathbb{E}\left[\ell(d,c) \mid w\right] - \min_{d' \in \mathbb{R}^m} \mathbb{E}\left[\ell(d',c) \mid w\right] &\geq \delta_\ell(\epsilon, w; \mathbb{P}) \\
&= \delta_\ell\left(\mathbb{E}\left[L(d,c) \mid w\right] - \min_{d' \in \mathbb{R}^m} \mathbb{E}\left[L(d',c) \mid w\right], w; \mathbb{P}\right).
\end{aligned}$$

To ensure that δ_ℓ is convex, we can instead use $\eta = \delta_\ell^{**}$, where, given a function $h : \mathbb{R} \to \mathbb{R} \cup \{\infty\}$,

$$h^{**}(\epsilon) = \sup_{h'} \left\{h'(\epsilon) : h' \text{ convex function on } \mathbb{R}, h' \leq h \text{ pointwise}\right\}.$$

Clearly, h^{**} is convex since it is a supremum of convex functions, and it can be obtained via convex conjugacy. However, we will not need to appeal to this representation for our results.

Note that δ_ℓ is only defined for $\epsilon > 0$, so we define $\delta_\ell(0, w; \mathbb{P}) = 0$ and $= +\infty$ when $\epsilon < 0$. Using $\eta = \delta_\ell^{**}$ guarantees convexity, and also that $\eta(\epsilon, w; \mathbb{P}) \leq \delta_\ell(\epsilon, w; \mathbb{P})$, hence the desired inequality (6.9) holds. Now, by the definition (6.7), we have δ_ℓ is non-decreasing in ϵ and positive for $\mathbb{P}$-calibrated ℓ. However, ℓ could be such that $\delta_\ell(\epsilon, w; \mathbb{P})$ does not increase once ϵ is sufficiently large, or only increases at a sublinear rate; in this case $\eta = \delta_\ell^{**}$ is going to be 0 for $\epsilon \geq 0$, so the inequality (6.9) will be useless. To prevent this, we make the assumption that $\mathbb{E}\left[L(d,c) \mid w\right] - \min_{d' \in \mathbb{R}^m} \mathbb{E}\left[L(d',c) \mid w\right] \leq B$ for all $w \in W$, $d \in \mathbb{R}^m$. We can then re-define $\delta_\ell(\epsilon, w; \mathbb{P}) = \infty$ for $\epsilon > B$, and take $\eta = \delta_\ell^{**}$. This ensures that $\eta(\epsilon) > 0$ for $\epsilon \in (0, B]$. To ensure that such a B exists, we define the following quantities:

$$B_X := \max_{x, x' \in X} \|x - x'\|_2, \quad B_f := \max_{x, x' \in X} \left\{f(x) - f(x')\right\}, \quad B_C := \max_{c \in C} \|c\|_2.$$

Note that since X is compact and f is continuous on X, $B_X, B_f < \infty$.

Assumption 6.25. The quantity $B_C < \infty$. (This means that $\mathbb{E}[c \mid w] \in \mathrm{Conv}(C)$ is uniformly bounded over $w \in W$.)

Remark 6.26. Under Assumption 6.25 and the fact that X is compact, we have

$$\begin{aligned}
&\mathbb{E}\left[L(d,c) \mid w\right] - \min_{d' \in \mathbb{R}^m} \mathbb{E}\left[L(d',c) \mid w\right] \\
&= f(x^*(d)) - f(x^*(\mathbb{E}[c \mid w])) + \mathbb{E}[c \mid w]^\top \left(x^*(d) - x^*(\mathbb{E}[c \mid w])\right) \\
&\leq f(x^*(d)) - f(x^*(\mathbb{E}[c \mid w])) + \|\mathbb{E}[c \mid w]\|_2 \, \|x^*(d) - x^*(\mathbb{E}[c \mid w])\|_2 \\
&\leq B_f + B_C B_X < \infty.
\end{aligned}$$

∎

Another subtlety that we need to consider is that there needs to be one fixed η for which (6.9) holds for all $w \in W$. Because of this, the definition $\eta = \delta_\ell^{**}$ is not well-defined as δ_ℓ in (6.7) depends on $w \in W$. To get around this, we need to strengthen the definition of calibration to be uniform across $w \in W$. In summary, the additions we need to make to the assumptions from Section 6.3 are Assumption 6.25, which ensures a uniform bound on the expected true loss, and a stronger definition of calibration, which we give next. Notice, however, that since the proof technique is different to that of Theorem 6.7, we only need measurability of ℓ, and not convexity in d. In practice, however, convexity of ℓ in d gives us implementable algorithms with performance guarantees.

6.4.2 Risk Bounds via Uniform Calibration

We consider the following strengthening of Definition 6.5.

Definition 6.27. We say that a loss function ℓ is *uniformly calibrated* with respect to a class of distributions $\mathcal{P}$ on $W \times C$, or *$\mathcal{P}$-uniformly calibrated*, if, for all $\epsilon > 0$, there exists $\delta > 0$ such that for all $\mathbb{P} \in \mathcal{P}$, $w \in W$ and $d \in \mathbb{R}^m$, we have

$$\mathbb{E}[\ell(d,c) \mid w] - \inf_{d'} \mathbb{E}[\ell(d',c) \mid w] \leq \delta \implies \mathbb{E}[L(d,c)] - \inf_{d'} \mathbb{E}[L(d',c)] \leq \epsilon. \tag{6.10}$$

Note that Definition 6.27 considers a class of distributions $\mathcal{P}$ so that we can get distribution-independent guarantees. In practice, we do not know $\mathbb{P}$, but we may know that $\mathbb{P}$ belongs to some class $\mathcal{P}$, so we may aim to get guarantees on the class $\mathcal{P}$.

If ℓ is $\mathcal{P}$-uniformly calibrated, then we define

$$\delta_\ell(\epsilon;\mathcal{P}) := \inf_{\substack{d\in\mathbb{R}^m \\ w\in W \\ \mathbb{P}\in\mathcal{P}}} \left\{ \mathbb{E}[\ell(d,c) \mid w] - \min_{d'\in\mathbb{R}^m} \mathbb{E}[\ell(d',c) \mid w] : \mathbb{E}[L(d,c) \mid w] - \min_{d'\in\mathbb{R}^m} \mathbb{E}[L(d',c)] > \epsilon \right\}. \tag{6.11}$$

Remark 6.28. If ℓ is $\mathcal{P}$-calibrated, then $\delta(\epsilon;\mathcal{P}) > 0$ for all $\epsilon > 0$, and is non-decreasing in ϵ. Furthermore, we have for any $d \in \mathbb{R}^m$, $w \in W$ and $\mathbb{P} \in \mathcal{P}$,

$$\delta_\ell\left(\mathbb{E}[L(d,c) \mid w] - \min_{d'\in\mathbb{R}^m} \mathbb{E}[L(d',c)]; \mathcal{P} \right) \leq \mathbb{E}[\ell(d,c) \mid w] - \min_{d'\in\mathbb{R}^m} \mathbb{E}[\ell(d',c)].$$

In addition, if Assumption 6.25 holds, then $\delta_\ell(\epsilon;\mathcal{P}) = \infty$ for $\epsilon > B_f + B_C B_X$ since the infimum is infeasible. Also, $\delta_\ell(\epsilon;\mathcal{P}) = 0$ for $\epsilon < 0$. Furthermore, measurability of $\delta_\ell(\cdot;\mathcal{P})$ follows by a similar proof to Lemma 6.14. ∎

Remark 6.28 shows that positivity of δ_ℓ is necessary for $\mathcal{P}$-uniform calibration. It turns out that it is also sufficient.

Lemma 6.29. *Suppose that $\delta_\ell(\epsilon;\mathcal{P}) > 0$ for all $\epsilon > 0$. Then ℓ is $\mathcal{P}$-uniformly calibrated.*

Proof. When $\delta_\ell(\epsilon;\mathcal{P}) > 0$, take $0 < \delta \leq \delta_\ell(\epsilon;\mathcal{P})$, and noting that $\delta_\ell(\cdot;\mathcal{P})$ is non-decreasing, we get for any $d \in \mathbb{R}^m$, $w \in W$ and $\mathbb{P} \in \mathcal{P}$,

$$\mathbb{E}[\ell(d,c) \mid w] - \min_{d'\in\mathbb{R}^m} \mathbb{E}[\ell(d',c) \mid w] \leq \delta < \delta_\ell(\epsilon;\mathcal{P}).$$

If $d \in \mathbb{R}^m$, $w \in W$ and $\mathbb{P} \in \mathcal{P}$ were such that $\mathbb{E}[L(d,c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d',c) \mid w] > \epsilon$, we reach a contradiction since we would then by definition of $\delta_\ell(\cdot;\mathcal{P})$ in (6.11) have $\mathbb{E}[\ell(d,c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d',c) \mid w] \geq \delta_\ell(\epsilon;\mathcal{P})$. Thus, for any $w \in W$ and $\mathbb{P} \in \mathcal{P}$,

$$\mathbb{E}[\ell(d,c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d',c) \mid w] \leq \delta \implies \mathbb{E}[L(d,c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d',c) \mid w] \leq \epsilon.$$

□

We now state the risk guarantee obtainable from uniform calibration. This is a generalization of Steinwart [143, Theorem 2.13], and the proof follows the outline described in Section 6.4.1.

Theorem 6.30. *Suppose that ℓ is $\mathcal{P}$-uniformly calibrated, and that Assumption 6.25 holds. Define*

$$\delta_\ell^{**}(\epsilon) := \sup_{h'} \left\{ h'(\epsilon) : h' \text{ convex function on } \mathbb{R}, h' \leq \delta_\ell \text{ pointwise} \right\}.$$

Then $\delta_\ell^{**}(\epsilon;\mathcal{P})$ is positive for $\epsilon \in (0, B_f + B_C B_X]$, and for any $\mathbb{P} \in \mathcal{P}$, $g : W \to C$,

$$\delta_\ell^{**}\left(R(g,\mathbb{P}) - R(\mathbb{P});\mathcal{P}\right) \leq R_\ell(g,\mathbb{P}) - R_\ell(\mathbb{P}).$$

Proof. We know that $\delta_\ell^{**}(\cdot;\mathcal{P})$ is convex by definition. Then, for any $\mathbb{P} \in \mathcal{P}$, using Jensen's inequality,

$$\begin{aligned}
\delta_\ell^{**}\left(R(g,\mathbb{P}) - R(\mathbb{P});\mathcal{P}\right) &= \delta_\ell^{**}\left(\mathbb{E}\left[\mathbb{E}\left[L(g(w),c) \mid w\right] - \min_{d' \in \mathbb{R}^m} \mathbb{E}\left[L(d',c) \mid w\right]\right];\mathcal{P}\right) \\
&\leq \mathbb{E}\left[\delta_\ell^{**}\left(\mathbb{E}\left[L(g(w),c) \mid w\right] - \min_{d' \in \mathbb{R}^m} \mathbb{E}\left[L(d',c) \mid w\right];\mathcal{P}\right)\right] \\
&\leq \mathbb{E}\left[\mathbb{E}\left[\ell(d,c) \mid w\right] - \min_{d' \in \mathbb{R}^m} \mathbb{E}\left[\ell(d',c) \mid w\right]\right] \\
&= R_\ell(g,\mathbb{P}) - R_\ell(\mathbb{P}).
\end{aligned}$$

We now prove that δ_ℓ^{**} is positive on $(0, B_f + B_X B_C]$. For convenience, define $B := B_f + B_C B_X$ and the epigraph

$$E := \left\{(\epsilon,\beta) \in (0, B_f + B_X B_C] \times \mathbb{R} : \delta_\ell(\epsilon) \leq \beta\right\}.$$

Since δ_ℓ is minorized by the zero function, Hiriart-Urruty and Lemaréchal [78, Proposition B.2.5.1] states that we can write

$$\delta_\ell^{**}(\epsilon) = \inf_{(\epsilon,\beta)} \left\{\beta : (\epsilon,\beta) \in \mathrm{Conv}(E)\right\}.$$

Suppose that there exists $\epsilon \in (0, B_f + B_X B_C]$ such that $\delta_\ell^{**}(\epsilon) = 0$. Then there exists a sequence $\beta_k \to 0$ such that $(\epsilon,\beta_k) \in \mathrm{Conv}(E)$ for all $k \in \mathbb{N}$. Now choose some $\epsilon^* < \epsilon$, and consider $k \in \mathbb{N}$ sufficiently large that $\beta_k < \delta_\ell(\epsilon^*)$. Such a k must exist since we know that $\delta_\ell(\epsilon^*) > 0$. Since $\epsilon > \epsilon^*$ and δ_ℓ is non-decreasing, we cannot have $(\epsilon,\beta_k) \in E$.

Since $(\epsilon,\beta_k) \in \mathrm{Conv}(E)$, there must exist two distinct points $(\epsilon_1,\beta_1), (\epsilon_2,\beta_2) \in E$ such that for some $\alpha \in (0,1)$,

$$\begin{aligned}
\epsilon &= \alpha\epsilon_1 + (1-\alpha)\epsilon_2 \\
\beta_k &= \alpha\beta_1 + (1-\alpha)\beta_2.
\end{aligned}$$

Note that while Carthéodory's theorem states that we need at most three points, the epigraphical structure of E implies that two points are sufficient. Assume without loss of generality that $\epsilon_1 < \epsilon_2$. If $\beta_1 = \beta_2 = \beta_k$, then we must have $\epsilon_1 < \epsilon_k < \epsilon_2$. This contradicts the fact that δ_ℓ is non-decreasing, since we have $\delta_\ell(\epsilon^*) > \beta_k = \beta_2 \geq \delta_\ell(\epsilon_2)$. Therefore, instead we must have $\beta_1 \neq \beta_2$. Since $\epsilon_2 > \epsilon^*$, we have $\beta_2 \geq \delta_\ell(\epsilon_2) \geq \delta_\ell(\epsilon^*) > \beta_k$, so we need $\beta_1 < \beta_k$. This further implies that $\epsilon_1 < \epsilon^*$ since $\delta_\ell(\epsilon_1) \leq \beta_1 < \beta_k < \delta_\ell(\epsilon^*)$. We can thus infer that

$$\alpha = \frac{\epsilon_2 - \epsilon}{\epsilon_2 - \epsilon_1} \leq \frac{\epsilon_2 - \epsilon}{\epsilon_2 - \epsilon^*} \leq \frac{B - \epsilon}{B - \epsilon^*} < 1,$$

where the second inequality follows since $\gamma \mapsto \frac{\gamma - \epsilon}{\gamma - \epsilon^*}$ is increasing in γ, and the last inequality follows sunce $B \geq \epsilon > \epsilon^*$. Furthermore, we have

$$\alpha = \frac{\beta_2 - \beta_k}{\beta_2 - \beta_1} \geq \frac{\beta_2 - \beta_k}{\beta_2} = 1 - \frac{\beta_k}{\beta_2} \geq 1 - \frac{\beta_k}{\delta_\ell(\epsilon^*)} > 0.$$

where the last inequality follows since $\beta_k < \delta_\ell(\epsilon^*)$. These bounds on α are independent of the points $(\epsilon_1, \beta_1), (\epsilon_2, \beta_2)$. If we choose β_k sufficiently close to 0, the lower bound becomes larger than the upper bound, which is a contradiction. □

6.4.3 Uniform Calibration of the Squared Loss

In general, ensuring uniform calibration of a loss function is much harder than showing admissibility. Thus, we now focus on a particular loss function, the squared loss, and show uniform calibration of this with respect to a rather large class of distributions $\mathcal{P}$, namely the class of all square integrable distributions. For this we will exploit the fact that the squared loss has a bias-variance decomposition. Henceforth, we will specify the following:

$$\begin{aligned}
\ell(d, c) &:= \|d - c\|_2^2 \\
\mathcal{P} &:= \{\mathbb{P} : \forall w \in W,\ \mathbb{P}[\cdot \mid w] \text{ is square integrable, and } \mathbb{E}[c \mid w] \in \mathrm{Conv}(C)\} \\
\delta(\cdot) &:= \delta_\ell(\cdot; \mathcal{P}).
\end{aligned}$$

We now give a positive lower bound for δ. First, we show that due to the structure of the squared loss ℓ, we can write δ depending only on the mean $\mathbb{E}[c \mid w]$.

Lemma 6.31. *We have*

$$\delta(\epsilon) = \inf_{\substack{d \in \mathbb{R}^m \\ c \in \mathrm{Conv}(C)}} \left\{ \|d - c\|_2^2 : f(x^*(d)) - f(x^*(c)) + c^\top (x^*(d) - x^*(c)) > \epsilon \right\}.$$

Proof. First, note that for any $w \in W$ and $\mathbb{P} \in \mathcal{P}$,

$$\begin{aligned}
\mathbb{E}[L(d, c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[L(d', c) \mid w] &= f(x^*(d)) + \mathbb{E}[c \mid w]^\top x^*(d) - \min_{x \in X} \left\{ f(x) + \mathbb{E}[c \mid w]^\top x \right\} \\
&= f(x^*(d)) - f(x^*(\mathbb{E}[c \mid w])) + \mathbb{E}[c \mid w]^\top \left(x^*(d) - x^*\left(\mathbb{E}[c \mid w]\right) \right).
\end{aligned}$$

Also observe that we have the usual bias-variance decomposition for squared error:

$$\begin{aligned}
\mathbb{E}[\ell(d, c) \mid w] &= \mathbb{E}[\|d - c\|_2^2 \mid w] \\
&= \|d - \mathbb{E}[c \mid w]\|_2^2 + 2\mathbb{E}\left[(d - \mathbb{E}[c \mid w])^\top (\mathbb{E}[c \mid w] - c) \right] + \mathbb{E}\left[\|\mathbb{E}[c \mid w] - c\|_2^2 \right] \\
&= \|d - \mathbb{E}[c \mid w]\|_2^2 + \mathbb{E}\left[\|\mathbb{E}[c \mid w] - c\|_2^2 \right].
\end{aligned}$$

Hence, we can minimize this by choosing $d = \mathbb{E}[c \mid w]$, and

$$\min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d', c) \mid w] = \mathbb{E}\left[\|\mathbb{E}[c \mid w] - c\|_2^2\right].$$

Therefore,

$$\mathbb{E}[\ell(d, c) \mid w] - \min_{d' \in \mathbb{R}^m} \mathbb{E}[\ell(d', c) \mid w] = \|d - \mathbb{E}[c \mid w]\|_2^2.$$

Thus, both the objective and the constraints in the expression for δ from (6.11) only depend on the mean $\mathbb{E}[c \mid w]$. However, if we consider $\mathcal{P}$ as the collection of all square integrable distributions, then any point $c' \in \mathrm{Conv}(C)$ is realizable as $\mathbb{E}[c \mid w] = c'$ by choosing $w \in W$ and $\mathbb{P} \in \mathcal{P}$ appropriately. Therefore, we can replace these in the definition of δ from (6.11) to get our result. □

Using Lemma 6.31, we can get an interpretable lower bound on δ, which also shows $\mathcal{P}$-uniform calibration.

Theorem 6.32. *The squared loss ℓ is $\mathcal{P}$-uniformly calibrated, with*

$$\delta(\epsilon) \geq \frac{\epsilon^2}{B_X^2} \quad \textit{for all } \epsilon > 0.$$

Proof. Define the set of vectors

$$D(d) := \left\{d' \in \mathbb{R}^m : x^*(d') = x^*(d)\right\}.$$

Note that from Lemma 6.31 we can rewrite δ as follows:

$$\begin{aligned}\delta(\epsilon) &= \inf_{\substack{d \in \mathbb{R}^m \\ c \in \mathrm{Conv}(C)}} \left\{\|d - c\|_2^2 : f(x^*(d)) - f(x^*(c)) + c^\top (x^*(d) - x^*(c)) > \epsilon\right\} \\ &= \inf_{\substack{d \in \mathbb{R}^m \\ c \in \mathrm{Conv}(C)}} \left\{\inf_{d' \in D(d)} \|d' - c\|_2^2 : f(x^*(d)) - f(x^*(c)) + c^\top (x^*(d) - x^*(c)) > \epsilon\right\}.\end{aligned}$$

In other words, to compute $\delta(\epsilon)$, we first fix $d \in \mathbb{R}^m$ and $c \in \mathrm{Conv}(C)$ such that $f(x^*(d)) - f(x^*(c)) + c^\top (x^*(d) - x^*(c)) > \epsilon$. Then we look at all vectors d' that give the same solution $x^*(d') = x^*(d)$, and get the minimum distance from these d' to c. Finally, we optimize this minimum distance over different choices of d and c.

Considering fixed $d \in \mathbb{R}^m$ and $c \in \mathrm{Conv}(C)$ satisfying $f(x^*(d)) - f(x^*(c)) + c^\top (x^*(d) - x^*(c)) > \epsilon$, note that the set $D(d)$ is contained in the halfspace

$$\mathcal{H}(d, c) = \left\{d' \in \mathbb{R}^m : f(x^*(d)) - f(x^*(c)) + (d')^\top (x^*(d) - x^*(c)) \leq 0\right\},$$

which, by definition, is the set of all vectors $d' \in \mathbb{R}^m$ for which $x^*(d)$ has lower objective than $x^*(c)$. Furthermore, we know that $c \notin H(d, c)$. Therefore, we know that the minimum ℓ_2-distance between c and a vector $d' \in D(d)$ is bounded by

$$\inf_{d' \in D(d)} \|d' - c\|_2 \geq \inf_{d' \in \mathcal{H}(d,c)} \|d' - c\|_2 = \frac{\left|f(x^*(d)) - f(x^*(c)) + c^\top (x^*(d) - x^*(c))\right|}{\|x^*(d) - x^*(c)\|_2} > \frac{\epsilon}{\|x^*(d) - x^*(c)\|_2}.$$

The first inequality follows since $D(d) \subseteq \mathcal{H}(d, c)$, the equality follows from the formula for the ℓ_2-distance between a point an a halfspace, and the second inequality follows by assumption on c and d.

Substituting this into the expression for δ gives

$$\delta(\epsilon) \geq \inf_{\substack{d \in \mathbb{R}^m \\ c \in \mathrm{Conv}(C)}} \left\{ \frac{\epsilon^2}{\|x^*(d) - x^*(c)\|_2^2} : f(x^*(d)) - f(x^*(c)) + c^\top (x^*(d) - x^*(c)) > \epsilon \right\} \geq \frac{\epsilon^2}{B_X^2}.$$

Then $\mathcal{P}$-uniform calibration follows from Lemma 6.29. □

Corollary 6.33. *When ℓ is the squared loss, we have*

$$\frac{1}{B_X^2} \left(R(g, \mathbb{P}) - R(\mathbb{P})\right)^2 \leq R_\ell(g, \mathbb{P}) - R_\ell(\mathbb{P}).$$

Proof. The result follows by observing that $\epsilon^2/B_X^2 \leq \delta^{**}(\epsilon)$ since $\epsilon \mapsto \epsilon^2/B_X^2$ is already convex, and then applying Theorem 6.30. □

Chapter 7

Conclusion

This dissertation presented two research directions on optimization under uncertainty: studying properties of models which can incorporate uncertain information on the optimization parameters, and finding efficient algorithms for such models which scale well as the dimension increases.

In Chapter 2, we presented a generic primal-dual algorithmic framework for two models in optimization under uncertainty: robust optimization (RO) and joint estimation-optimization (JEO). We analyse a generic parametric saddle point problem, which can capture both RO and JEO. By analysing this problem, we uncover three terms which upper bound the performance gap of a candidate primal-dual pair. Deriving algorithms from this then reduces to bounding the three terms. For RO, we analyse these terms and outline a strategy for solving the RO problem in Algorithm 1. We then show that previous iterative RO algorithms can be captured by our framework. We also derive more efficient iterative RO algorithms in a principled manner through our framework, and perform a complexity analysis of these. For JEO, we present a mild sufficient condition for convergence based on the given data sequence. Moreover, our analysis exposes a natural dependence of the optimalty gap on the data sequence convergence rate.

In Chapter 3 we presented results regret minimization results for online convex optimization (OCO), which are used to bound the three critical terms in the primal-dual framework of Chapter 2. In this setting, we were able to relax some of the requirements in OCO and study regret under more flexible conditions. These include weighted regret, online saddle point problems and lookahead decisions. It turns out that these are critical for exploiting favourable structure, such as strong convexity and smoothness, in RO and JEO. We presented algorithms which exhibit improved regret guarantees under such assumptions.

In Chapter 4, we studied the trust-region subproblem (TRS), and its implications for robust quadratic programming. We gave a tight second-order cone based convexification of the TRS that is still in the space of original variables. We also gave conditions for this convexification to remain tight when additional conic constraints are added to the domain. Our convexification allows us to improve the best known complexity bound for solving the TRS. We also gave results for obtaining the convex hull of the epigraph for the TRS with additional conic constraints. We illustrated how our convexification for the TRS applies to robust quadratic programming, and applied our primal-dual framework from Chapter 2 to derive efficient algorithms. Numerical results indicate that the algorithms we derived are much more scalable than previous ones from the literature.

In Chapter 5, we studied non-parametric choice model estimation in the dynamic data setting when we receive additional choice observations over time. This has a natural JEO interpretation,

thus we applied the framework of Chapter 2 to derive efficient algorithms. However, this problem has the particular challenge of having exponentially many primal variables, thus care had to be taken in designing the algorithms. We analysed a natural dynamic Frank-Wolfe algorithm for this problem which converged under a data convergence rate assumption, which is undesirable. To remove this, we used our primal-dual framework to derive algorithms in a principled manner. In particular, we derived a variant of the Frank-Wolfe algorithm which does not require the data convergence rate assumption.

Finally, in Chapter 6, we a risk analysis for joint prediction and optimization of objective functions with uncertain linear term. In particular, for the canonical but non-convex loss function L defined by the optimality gap, we gave sufficient conditions on the surrogate prediction loss ℓ for asymptotic risk minimization, with the main one being admissibility. Furthermore, we examined several surrogate losses used in practice, and looked at their admissibility properties. We further gave non-asymptotic bounds on the excess risk of the true loss in terms of the surrogate loss under a stronger uniform calibration condition. Specifically for the squared loss (which, in prediction terms, corresponds to minimizing the mean squared error), we showed that uniform calibration is satisfied, and derived the precise relationship between this and the true risk.

Future Directions

Chapter 2 showed that our framework covers RO and JEO. We believe, however, that this framework can also cover stochastic optimization. It would be interesting to see how current algorithms for stochastic optimization are related to this framework, if at all. For JEO, we have considered uncertainty in the objective only. It would be interesting to see how uncertainty in the constraints can be captured within the framework. We believe there is hope for this, since our paper [84] gives saddle point representations for *deterministic* convex optimization with functional constraints.

In Chapter 5.3.3, we saw that computing the primal updates for the non-parametric choice model estimation problem is NP-hard in general. A very interesting question is whether or not the algorithms will hold if these primal updates are instead computed approximately. This is related to a broader question: what theoretical guarantees, if any, are available for first-order algorithms (under uncertainty) when subgradient information is computed inexactly? For example, if we use an approximation algorithm with some guaranteed ratio to compute the primal updates, what guarantees can we have on the overall scheme?

In Chapter 6, the uncertain objective parameters c appeared only in the linear term of the objective $f(x) + c^\top x$. While this covers a large variety of applications, we believe a very interesting open direction is to extend these results to non-linear dependences on the parameters $f(x, c)$. Another interesting question is which other surrogate loss functions admit non-asymptotic guarantees besides squared loss. Furthermore, it would be interesting to study the use of different risk measures other than expectation in the definition of the risk $R(g, \mathbb{P})$, for example, conditional value-at-risk.

Bibliography

[1] J. Abernethy, P. L. Bartlett, A. Rakhlin, and A. Tewari. Optimal strategies and minimax lower bounds for online convex games. In *Proceedings of the 19th Annual Conference on Computational Learning Theory*, 2008.

[2] J. Abernethy, K. A. Lai, K. Y. Levy, and J.-K. Wang. Faster rates for convex-concave games. In S. Bubeck, V. Perchet, and P. Rigollet, editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 1595–1625. PMLR, 06–09 Jul 2018. URL http://proceedings.mlr.press/v75/abernethy18a.html.

[3] J. D. Abernethy and J.-K. Wang. On frank-wolfe and equilibrium computation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6584–6593. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7236-on-frank-wolfe-and-equilibrium-computation.pdf.

[4] S. Adachi, S. Iwata, Y. Nakatsukasa, and A. Takeda. Solving the trust region subproblem by a generalized eigenvalue problem. *SIAM Journal on Optimization*, 27(1):269–291, 2017.

[5] H. Ahmadi and U. V. Shanbhag. Data-driven first-order methods for misspecified convex optimization problems: Global convergence and rate estimates. In *53rd IEEE Conference on Decision and Control*, pages 4228–4233, Dec 2014.

[6] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5(1):13–51, 1995.

[7] L. L. Andrew, S. Barman, K. Ligett, M. Lin, A. Meyerson, A. Roytman, and A. Wierman. A tale of two metrics: Simultaneous bounds on competitiveness and regret. *Journal of Machine Learning Research: Workshop and Conference Proceedings*, 30:741–763, 2013. URL http://www.jmlr.org/proceedings/papers/v30/Andrew13.html.

[8] K. J. Arrow, L. Hurwicz, and H. Uzawa. *Studies in linear and non-linear programming*. Stanford University Press, 1958.

[9] G.-Y. Ban and C. Rudin. The big data newsvendor: Practical insights from machine learning. *Operations Research*, 67(1):90–108, 2019. doi: 10.1287/opre.2018.1757.

[10] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006. ISSN 01621459. URL http://www.jstor.org/stable/30047445.

[11] A. Beck. Convexity properties associated with nonconvex quadratic matrix functions and applications to quadratic programming. *Journal of Optimization Theory and Applications*, 142(1):1–29, 2009.

[12] A. Beck and Y. C. Eldar. Strong duality in nonconvex quadratic optimization with two quadratic constraints. *SIAM Journal on Optimization*, 17(3):844–860, 2006.

[13] A. Beck and M. Teboulle. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization*, 22(2):557–580, 2012. doi: 10.1137/100818327.

[14] A. Beck, A. Ben-Tal, N. Guttmann-Beck, and L. Tetruashvili. The CoMirror algorithm for solving nonsmooth constrained convex problems. *Operations Research Letters*, 38(6):493 – 498, 2010.

[15] A. Ben-Tal and D. den Hertog. Hidden conic quadratic representation of some nonconvex quadratic optimization problems. *Mathematical Programming*, 143(1):1–29, 2014.

[16] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998. doi: 10.1287/moor.23.4.769. URL http://dx.doi.org/10.1287/moor.23.4.769.

[17] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization*. Society for Industrial and Applied Mathematics, 2001.

[18] A. Ben-Tal and A. Nemirovski. Robust optimization – methodology and applications. *Mathematical Programming*, 92(3):453–480, 2002.

[19] A. Ben-Tal and A. Nemirovski. Selected topics in robust convex optimization. *Mathematical Programming*, 112(1):125–158, 2008.

[20] A. Ben-Tal and A. Nemirovski. Lectures on modern convex optimization. Technical report, August 2015. http://www2.isye.gatech.edu/~nemirovs/Lect_ModConvOpt.pdf.

[21] A. Ben-Tal and M. Teboulle. Hidden convexity in some nonconvex quadratically constrained quadratic programming. *Mathematical Programming*, 72(1):5163, 1996.

[22] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press. Princeton Series in Applied Mathematics, Philadelphia, PA, USA, 2009.

[23] A. Ben-Tal, L. Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, 2009. ISBN 9781400831050. URL https://books.google.com/books?id=DttjR7IpjUEC.

[24] A. Ben-Tal, D. den Hertog, and J.-P. Vial. Deriving robust counterparts of nonlinear uncertain inequalities. *Mathematical Programming*, 149(1-2):265–299, 2015. ISSN 0025-5610. doi: 10.1007/s10107-014-0750-8. URL http://dx.doi.org/10.1007/s10107-014-0750-8.

[25] A. Ben-Tal, E. Hazan, T. Koren, and S. Mannor. Oracle-based robust optimization via online learning. *Operations Research*, 63(3):628–638, 2015. doi: 10.1287/opre.2015.1374. URL http://dx.doi.org/10.1287/opre.2015.1374.

[26] D. Bertsimas and N. Kallus. From Predictive to Prescriptive Analytics. *arXiv e-prints*, art. arXiv:1402.5481, Feb 2014.

[27] D. Bertsimas and V. Mišic. Data-driven assortment optimization. Technical report, September 2015. https://www.algomus.com/wp-content/uploads/2016/11/Data-Driven-Assortment-Optimization.pdf.

[28] D. Bertsimas and B. Van Parys. Bootstrap Robust Prescriptive Analytics. *arXiv e-prints*, art. arXiv:1711.09974, Nov 2017.

[29] D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011. doi: 10.1137/080734510. URL http://dx.doi.org/10.1137/080734510.

[30] D. Bienstock. A note on polynomial solvability of the CDT problem. *SIAM Journal on Optimization*, 26(1):488–498, 2016.

[31] D. Bienstock and A. Michalka. Cutting-planes for optimization of convex functions over nonconvex sets. *SIAM Journal on Optimization*, 24(2):643–677, 2014.

[32] D. Bienstock and A. Michalka. Polynomial solvability of variants of the trust-region subproblem. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 380–390, 2014.

[33] J. R. Birge and F. Louveaux. *Introduction to stochastic programming*, volume 4 of *Springer Series in Operations Research and Financial Engineering*. Springer-Verlag New York, NY, USA, 2 edition, 2011.

[34] H. Block and J. Marschak. Random orderings and stochastic theories of responses. *Contributions to Probability and Statistics*, 2:97–132, 1960.

[35] V. Bogachev. *Measure Theory*. Springer-Verlag Berlin Heidelberg, 2007.

[36] J. F. Bonnans and A. Shapiro. *Perturbation analysis of optimization problems*. Springer Series in Operations Research. Springer, New York, NY, 2000. ISBN 978-1-4612-1394-9.

[37] A. Borodin, N. Linial, and M. E. Saks. An optimal on-line algorithm for metrical task system. *Journal of the ACM*, 39(4):745–763, Oct. 1992. doi: 10.1145/146585.146588.

[38] O. Bousquet, S. Boucheron, and G. Lugosi. *Introduction to Statistical Learning Theory*, pages 169–207. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-28650-9. doi: 10.1007/978-3-540-28650-9_8. URL https://doi.org/10.1007/978-3-540-28650-9_8.

[39] R. Brent. *Algorithms for Minimization Without Derivatives*. Dover Books on Mathematics. Dover Publications, 1973. ISBN 9780486419985.

[40] N. Buchbinder, S. Chen, J. Naor, and O. Shamir. Unified algorithms for online learning and competitive analysis. *Journal of Machine Learning Research: Workshop and Conference Proceedings*, 23:5.1–5.18, 2012. URL http://www.jmlr.org/proceedings/papers/v23/buchbinder12.html.

[41] C. Buchheim, M. De Santis, L. Palagi, and M. Piacentini. An exact algorithm for nonconvex quadratic integer minimization using ellipsoidal relaxations. *SIAM Journal on Optimization*, 23(3):1867–1889, 2013.

[42] S. Burer. A gentle, geometric introduction to copositive optimization. *Mathematical Programming*, 151(1):89–116, 2015.

[43] S. Burer and K. M. Anstreicher. Second-order-cone constraints for extended trust-region subproblems. *SIAM Journal on Optimization*, 23(1):432–451, 2013.

[44] S. Burer and F. Kılınç-Karzan. How to convexify the intersection of a second order cone and a nonconvex quadratic. *Mathematical Programming*, 162(1):393–429, 2017.

[45] S. Burer and B. Yang. The trust region subproblem with non-intersecting linear constraints. *Mathematical Programming*, 2015, volume=.

[46] G. Calafiore and M. Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102(1):25–46, 2005.

[47] C. Caramanis, S. Mannor, and H. Xu. Robust optimization in machine learning. In S. Sra, S. Nowozin, and S. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2012.

[48] M. Celis, J. Dennis, and R. Tapia. A trust region strategy for nonlinear equality constrained optimization. In *Numerical Optimization 1984: Proceedings of the SIAM Conference on Numerical Optimization*, pages 71–82. SIAM Philadelphia, PA, 1985.

[49] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006. ISBN 9781139454827. URL https://books.google.com/books?id=zDnRBlazhfYC.

[50] C.-K. Chiang, T. Yang, C.-J. Lee, M. Mahdavi, C.-J. Lu, R. Jin, and S. Zhu. Online optimization with gradual variations. In *Conference on Learning Theory*, pages 6–1, 2012.

[51] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-Region Methods*. MPS/SIAM Series on Optimization. SIAM, Philadelphia, PA, 2000.

[52] A. Desir, V. Goyal, S. Jagabathula, and D. Segev. Assortment optimization under the mallows model. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4700–4708. 2016.

[53] O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1):37–75, Aug 2014. ISSN 1436-4646.

[54] L. L. Dines. On the mapping of quadratic forms. *Bulletin of the American Mathematical Society*, 47(6):494–498, 06 1941.

[55] P. Donti, B. Amos, and J. Z. Kolter. Task-based end-to-end model learning in stochastic optimization. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5484–5494. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7132-task-based-end-to-end-model-learning-in-stochastic-optimization.pdf.

[56] D. Drusvyatskiy and A. S. Lewis. Generic nondegeneracy in convex optimization. *Proceedings of the American Mathematical Society*, 139(7):2519–2527, 2011. ISSN 00029939, 10886826. URL https://doi.org/10.1090/S0002-9939-2010-10692-5.

[57] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 613–622, New York, NY, USA, 2001. ACM.

[58] A. N. Elmachtoub and P. Grigas. Smart "Predict, then Optimize". Technical report, Oct. 2017. URL https://arxiv.org/abs/1710.08005.

[59] J. B. Erway and P. E. Gill. A subspace minimization method for the trust-region step. *SIAM Journal on Optimization*, 20(3):1439–1461, 2010.

[60] J. B. Erway, P. E. Gill, and J. D. Griffin. Iterative methods for finding a trust-region step. *SIAM Journal on Optimization*, 20(2):1110–1131, 2009.

[61] V. Farias, S. Jagabathula, and D. Shah. A data-driven approach to modeling choice. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 504–512. 2009.

[62] V. Farias, S. Jagabathula, and D. Shah. A nonparametric approach to modeling choice with limited data. *Management Science*, 59(2):305–322, 2013.

[63] V. F. Farias, S. Jagabathula, and D. Shah. Building optimized and hyperlocal product assortments: A nonparametric choice approach. Technical report, January 2017. https://ssrn.com/abstract=2905381.

[64] C. Fortin and H. Wolkowicz. The trust region subproblem and semidefinite programming. *Optimization Methods and Software*, 19(1):41–67, 2004.

[65] A. L. Fradkov and V. A. Yakubovich. The S-procedure and duality relations in nonconvex problems of quadratic programming. *Vestn. LGU, Ser. Mat., Mekh., Astron*, 6(1):101–109, 1979.

[66] W. Gander, G. H. Golub, and U. von Matt. A constrained eigenvalue problem. *Linear Algebra and its Applications*, 114115:815 – 839, 1989. Special Issue Dedicated to Alan J. Hoffman.

[67] D. Goldfarb and G. Iyengar. Robust portfolio selection problems. *Mathematics of Operations Research*, 28(1):1–38, 2003.

[68] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in Mathematical Sciences. The Johns Hopkins University Press, Baltimore, MD, 1996.

[69] N. I. M. Gould, S. Lucidi, M. Roma, and P. L. Toint. Solving the trust-region subproblem using the Lanczos method. *SIAM Journal on Optimization*, 9(2):504–525 (electronic), 1999.

[70] N. I. M. Gould, D. P. Robinson, and H. S. Thorne. On solving trust-region and other regularized subproblems in optimization. *Mathematical Programming Computation*, 2(1): 21–57, 2010.

[71] G. A. Hanasusanto and D. Kuhn. Robust data-driven dynamic programming. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 827–835. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5123-robust-data-driven-dynamic-programming.pdf.

[72] L. Hannah, W. Powell, and D. M. Blei. Nonparametric density estimation for stochastic optimization with an observable state variable. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 820–828. Curran Associates, Inc., 2010. URL http://papers.nips.cc/paper/4098-nonparametric-density-estimation-for-stochastic-optimization-with-an-observable-stat
pdf.

[73] E. Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016. ISSN 2167-3888. doi: 10.1561/2400000013. URL http://dx.doi.org/10.1561/2400000013.

[74] E. Hazan and S. Kale. Projection-free online learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1843–1850, USA, 2012. Omnipress. ISBN 978-1-4503-1285-1. URL http://dl.acm.org/citation.cfm?id=3042573.3042808.

[75] E. Hazan and S. Kale. Beyond the regret minimization barrier: Optimal algorithms for stochastic strongly-convex optimization. *Journal of Machine Learning Research*, 15:2489–2512, 2014. URL http://jmlr.org/papers/v15/hazan14a.html.

[76] E. Hazan and T. Koren. A linear-time algorithm for trust region problems. *Mathematical Programming*, 158(1):363–381, 2016.

[77] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007. ISSN 0885-6125. doi: 10.1007/s10994-007-5016-8. URL http://dx.doi.org/10.1007/s10994-007-5016-8.

[78] J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis*. Springer-Verlag Berlin Heidelberg, 2001. ISBN 978-3-642-56468-0.

[79] C. P. Ho and G. A. Hanasusanto. On data-driven prescriptive analytics with side information: A regularized nadaraya-watson approach. Technical report, March 2019.

[80] N. Ho-Nguyen and F. Kılınç-Karzan. Exploiting problem structure in optimization under uncertainty via online convex optimization. *Mathematical Programming*, Mar 2018. doi: 10.1007/s10107-018-1262-8.

[81] N. Ho-Nguyen and F. Kılınç-Karzan. A second-order cone based approach for solving the trust-region subproblem and its variants. *SIAM Journal on Optimization*, 27(3):1485–1512, 2017. doi: 10.1137/16M1065197.

[82] N. Ho-Nguyen and F. Kılınç-Karzan. Online first-order framework for robust convex optimization. *Operations Research*, 66(6):1670–1692, 2018. doi: 10.1287/opre.2018.1764.

[83] N. Ho-Nguyen and F. Kılınç-Karzan. Dynamic data-driven estimation of non-parametric choice models. Technical report, February 2018. https://arxiv.org/abs/1702.05702.

[84] N. Ho-Nguyen and F. Kılınç-Karzan. Primal-dual algorithms for convex optimization via regret minimization. *IEEE Control Systems Letters*, 2(2):284–289, 2018. doi: 10.1109/LCSYS.2018.2831721.

[85] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, New York, 2013.

[86] S. Jagabathula and P. Rusmevichientong. The limit of rationality in choice modeling: Formulation, computation, and implications. *(forthcoming) Management Science*, 2018. doi: 10.1287/mnsc.2018.3030.

[87] S. Jagabathula and D. Shah. Inferring rankings under constrained sensing. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 753–760. 2008.

[88] M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28(1), pages 427–435, 2013.

[89] R. Jenatton, J. Huang, D. Csiba, and C. Archambeau. Online optimization and regret guarantees for non-additive long-term constraints. Technical report, February 2016. http://arxiv.org/abs/1602.05394.

[90] V. Jeyakumar and G. Y. Li. Trust-region problems with linear inequality constraints: Exact SDP relaxation, global optimality and robust optimization. *Mathematical Programming*, 2014, volume=.

[91] H. Jiang and U. V. Shanbhag. On the solution of stochastic optimization problems in imperfect information regimes. In *2013 Winter Simulations Conference*, pages 821–832, Dec 2013.

[92] H. Jiang and U. V. Shanbhag. On the solution of stochastic optimization and variational problems in imperfect information regimes. *SIAM Journal on Optimization*, 26(4):2394–2429, 2016.

[93] A. Juditsky and A. Nemirovski. First-order methods for nonsmooth convex large-scale optimization, I: General purpose methods. In S. Sra, S. Nowozin, and S. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2012.

[94] A. Juditsky and A. Nemirovski. First-order methods for nonsmooth convex large-scale optimization, II: Utilizing problem's structure. In S. Sra, S. Nowozin, and S. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2012.

[95] A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291 – 307, October 2005.

[96] Y. Kao, B. V. Roy, and X. Yan. Directed regression. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 889–897. Curran Associates, Inc., 2009. URL http://papers.nips.cc/paper/3686-directed-regression.pdf.

[97] N. Komodakis and J. C. Pesquet. Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems. *IEEE Signal Processing Magazine*, 32(6):31–54, Nov 2015. ISSN 1053-5888. doi: 10.1109/MSP.2014.2377273.

[98] A. Koppel, F. Y. Jakubiec, and A. Ribeiro. A saddle point algorithm for networked online convex optimization. *IEEE Transactions on Signal Processing*, 63(19):5149–5164, Oct 2015.

[99] J. Kuczynski and H. Wozniakowski. Estimating the largest eigenvalue by the power and lanczos algorithms with a random start. *SIAM Journal on Matrix Analysis and Applications*, 13(4):1094–1122, 1992.

[100] S. Lacoste-Julien, M. W. Schmidt, and F. R. Bach. A simpler approach to obtaining an $O(1/t)$ convergence rate for the projected stochastic subgradient method. Technical report, December 2012. http://arxiv.org/abs/1212.2002.

[101] Y. Lin. A note on margin-based loss functions in classification. *Statistics & Probability Letters*, 68(1):73 – 82, 2004. ISSN 0167-7152. doi: 10.1016/j.spl.2004.03.002. URL https://doi.org/10.1016/j.spl.2004.03.002.

[102] M. Locatelli. Exactness conditions for an sdp relaxation of the extended trust region problem. *Optimization Letters*, 10(6):1141–1151, 2016.

[103] S. Mahajan and G. van Ryzin. Stocking retail assortments under dynamic consumer substitution. *Operations Research*, 49(3):334–351, 2001.

[104] M. Mahdavi, R. Jin, and T. Yang. Trading regret for efficiency: online convex optimization with long term constraints. *Journal of Machine Learning Research*, 13(Sep):2503–2528, 2012.

[105] S. Modaresi and J. Vielma. Convex hull of two quadratic or a conic quadratic and a quadratic inequality. Technical report, November 2014. http://www.optimization-online.org/DB_HTML/2014/11/4641.html.

[106] J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.

[107] A. Mutapcic and S. Boyd. Cutting-set methods for robust convex optimization with pessimizing oracles. *Optimization Methods and Software*, 24(3):381–406, June 2009. ISSN 1055-6788. doi: 10.1080/10556780802712889. URL http://dx.doi.org/10.1080/10556780802712889.

[108] A. Nedić and S. Lee. On stochastic subgradient mirror-descent algorithm with weighted averaging. *SIAM Journal on Optimization*, 24(1):84–107, 2014.

[109] A. Nedić and A. Özdağlar. Approximate primal solutions and rate analysis for dual subgradient methods. *SIAM Journal on Optimization*, 19(4):1757–1780, 2009. doi: 10.1137/070708111.

[110] A. Nedić and A. Özdağlar. Subgradient methods for saddle-point problems. *Journal of Optimization Theory and Applications*, 142(1):205–228, 2009. ISSN 1573-2878. doi: 10.1007/s10957-009-9522-7.

[111] A. Nemirovski. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15:229–251, 2004.

[112] A. Nemirovski. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[113] A. Nemirovski and D. B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience series in discrete mathematics. Wiley, Chichester, New York, 1983. ISBN 0-471-10345-4. URL http://opac.inria.fr/record=b1091338. A Wiley-Interscience publication.

[114] Y. Nesterov. A method for solving a convex programming problem with rate of convergence $O(1/k^2)$. *Soviet Math. Dokl.*, 27(2):372–376, 1983.

[115] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004. ISBN 9781441988539.

[116] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[117] Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259, 2009.

[118] Y. Nesterov and A. Nemirovski. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.

[119] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006.

[120] I. Pólik and T. Terlaky. A survey of the s-lemma. *SIAM Review*, 49(3):371–418, 2007.

[121] B. T. Polyak. A general method of solving extremum problems. *Soviet Mathematics Doklady*, 8(3):593–597, 1967.

[122] T. K. Pong and H. Wolkowicz. The generalized trust region subproblem. *Computational Optimization and Applications*, 58(2):273–322, 2014.

[123] A. Rakhlin and K. Sridharan. Online learning with predictable sequences. In *Conference on Learning Theory*, pages 993–1019, 2013.

[124] A. Rakhlin and K. Sridharan. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, pages 3066–3074, 2013.

[125] A. Rakhlin, O. Shamir, and K. Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ICML '12, pages 449–456, New York, NY, USA, July 2012. Omnipress. ISBN 978-1-4503-1285-1.

[126] F. Rendl and H. Wolkowicz. A semidefinite framework for trust region subproblems with applications to large scale minimization. *Mathematical Programming*, 77(2):273–299, 1997.

[127] H. Robbins. Asymptotically subminimax solutions of compound statistical decision problems. In *In Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, page 131149, 1950.

[128] R. T. Rockafellar. *Convex analysis*. Princeton Mathematical Series. Princeton University Press, Princeton, N. J., 1970.

[129] R. T. Rockafellar. *Coherent Approaches to Risk in Optimization Under Uncertainty*, chapter Chapter 3, pages 38–61. 2007. doi: 10.1287/educ.1073.0032. URL https://pubsonline.informs.org/doi/abs/10.1287/educ.1073.0032.

[130] M. Rojas, S. A. Santos, and D. C. Sorensen. A new matrix-free algorithm for the large-scale trust-region subproblem. *SIAM Journal on Optimization*, 11(3):611–646, 2001.

[131] P. Rusmevichientong, B. V. Roy, and P. Glynn. A nonparametric approach to multiproduct pricing. *Operations Research*, 54(1):82–98, 2006.

[132] M. Salahi and S. Fallahi. Trust region subproblem with an additional linear inequality constraint. *Optimization Letters*, 10(4):821–832, 2016.

[133] M. Salahi and A. Taati. A fast eigenvalue approach for solving the trust region subproblem with an additional linear inequality. *Computational and Applied Mathematics*, 37(1):329–347, Mar 2018. doi: 10.1007/s40314-016-0347-3.

[134] M. Salahi, A. Taati, and H. Wolkowicz. 2017, title=.

[135] S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[136] A. Shapiro, D. Dentcheva, and A. Ruszczynski. *Lectures on Stochastic Programming: Modeling and Theory, Second Edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2014. ISBN 1611973422, 9781611973426.

[137] M. Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958. URL http://projecteuclid.org/euclid.pjm/1103040253.

[138] D. C. Sorensen. Minimization of a large-scale quadratic function subject to a spherical constraint. *SIAM Journal on Optimization*, 7(1):141–161, 1997.

[139] E. M. Stein and R. Shakarchi. *Real analysis: measure theory, integration, and Hilbert spaces*. Princeton University Press, 2009.

[140] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *J. Mach. Learn. Res.*, 2:67–93, Mar. 2002. ISSN 1532-4435. doi: 10.1162/153244302760185252. URL https://doi.org/10.1162/153244302760185252.

[141] I. Steinwart. Support vector machines are universally consistent. *Journal of Complexity*, 18 (3):768 – 791, 2002. ISSN 0885-064X. doi: 10.1006/jcom.2002.0642. URL https://doi.org/10.1006/jcom.2002.0642.

[142] I. Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *IEEE Transactions on Information Theory*, 51(1):128–142, Jan 2005. ISSN 0018-9448. doi: 10.1109/TIT.2004.839514. URL https://doi.org/10.1109/TIT.2004.839514.

[143] I. Steinwart. How to compare different loss functions and their risks. *Constructive Approximation*, 26(2):225–287, Aug 2007. ISSN 1432-0940. doi: 10.1007/s00365-006-0662-3. URL https://doi.org/10.1007/s00365-006-0662-3.

[144] R. J. Stern and H. Wolkowicz. Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations. *SIAM Journal on Optimization*, 5(2):286–313, 1995.

[145] J. F. Sturm and S. Zhang. On cones of nonnegative quadratic functions. *Mathematics of Operations Research*, 28(2):246–267, 2003.

[146] K. Talluri and G. van Ryzin. *The theory and practice of revenue management*, volume 68 of *International Series in Operations Research & Management Science*. Springer US, 2004.

[147] G. van Ryzin and G. Vulcano. A market discovery algorithm to estimate a general class of nonparametric choice models. *Management Science*, 61(2):281–300, 2015.

[148] J.-K. Wang and J. D. Abernethy. Acceleration through optimistic no-regret dynamics. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 3824–3834. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7639-acceleration-through-optimistic-no-regret-dynamics.pdf.

[149] W. Wiesemann, D. Kuhn, and M. Sim. Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376, 2014. doi: 10.1287/opre.2014.1314.

[150] L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, Dec. 2010.

[151] B. Yang, K. Anstreicher, and S. Burer. Quadratic programs with hollows. Technical report, January 2016. http://www.optimization-online.org/DB_FILE/2016/01/5277.pdf.

[152] T. Yang, M. Mahdavi, R. Jin, and S. Zhu. Regret bounded by gradual variation for online convex optimization. *Machine Learning*, 95(2):183–223, 2014. ISSN 0885-6125. doi: 10.1007/s10994-013-5418-8. URL http://dx.doi.org/10.1007/s10994-013-5418-8.

[153] Y. Ye and S. Zhang. New results on quadratic minimization. *SIAM Journal on Optimization*, 14(1):245–267, 2003.

[154] H. Yu and M. J. Neely. A primal-dual type algorithm with the o(1/t) convergence rate for large scale constrained convex programs. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 1900–1905, Dec 2016. doi: 10.1109/CDC.2016.7798542.

[155] H. Zhang, A. R. Conn, and K. Scheinberg. A derivative-free algorithm for least-squares minimization. *SIAM Journal on Optimization*, 20(6):3555–3576, 2010.

[156] T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Ann. Statist.*, 32(1):56–85, 02 2004. doi: 10.1214/aos/1079120130. URL https://doi.org/10.1214/aos/1079120130.

[157] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages 928–936, 2003. URL http://www.aaai.org/Library/ICML/2003/icml03-120.php.

Appendix A

Appendix to Chapter 4

A.1 Working with Approximate Eigenvalues

Consider the classical TRS (4.2) and its convex reformulation (4.5). In practice, we will actually form the objective $y^\top(Q-\gamma I_n)y + 2g^\top y + \gamma$ where $\gamma \approx \lambda_Q$ is an approximation. Due to this imprecision, we must ensure that the objective remains convex. To do this, suppose that we solve the minimum eigenvalue problem of Q to within $\mathcal{E}$-accuracy, and obtain an overestimating approximate solution $\lambda_Q < \lambda < \lambda_Q + \mathcal{E}$. Subtracting $\mathcal{E}$ from this inequality, we obtain $\lambda_Q - \mathcal{E} < \lambda - \mathcal{E} < \lambda_Q$. As stated in Section 4.2.3, an $\mathcal{E}$-accurate overestimating approximation to λ_Q can be found using the Lanczos method (see [99, Section 4] and [76, Section 5]). To ensure the convexity of the objective, we set $\gamma := \lambda - \mathcal{E} < \lambda_Q$ which is an underestimate of λ_Q, ensuring that $Q - \gamma I_n \succ 0$. Let $\eta := \lambda_Q - \gamma$, which satisfies $0 < \eta < \mathcal{E}$, and

$$f_\eta(y) := y^\top(Q-\gamma I_n)y + 2g^\top y = y^\top(Q-(\lambda_Q-\eta)I_n)y + 2g^\top y = f(y) + \eta\|y\|^2.$$

Based on this scheme, we next explore the effects of solving

$$\min_y \{f_\eta(y) : \ \|y\| \le 1\} \tag{A.1}$$

instead of (4.5). Let y^* be an optimal solution to the true convex reformulation (4.5). Let y^η be an optimal solution to (A.1), and let $\bar{y}^\eta$ be an approximate optimal solution. Then, we can bound the objective value $f(\bar{y}^\eta)$ as

$$f(\bar{y}^\eta) - f(y^*) = f_\eta(\bar{y}^\eta) - f_\eta(y^*) + \eta(\|y^*\|^2 - \|\bar{y}^\eta\|^2) \le f_\eta(\bar{y}^\eta) - f_\eta(y^\eta) + \eta,$$

where the last inequality follows from $\|y^*\| \le 1$ and $\|\bar{y}^\eta\| \le 1$. Thus, the convergence rate of $\bar{y}^\eta$ to the optimum of (4.5) is controlled by the size of η and the convergence rate for solving (A.1).

We can also control the distance between y^η and y^*. Because $f_\eta(y)$ is a 2η-strongly convex function, we have

$$\begin{aligned}\eta\|y^* - y^\eta\|^2 &\le f_\eta(y^*) - f_\eta(y^\eta) + \nabla f_\eta(y^\eta)^\top(y^\eta - y^*) \\ &= f(y^*) - f(y^\eta) + \nabla f_\eta(y^\eta)^\top(y^\eta - y^*) + \eta(\|y^*\|^2 - \|y^\eta\|^2) \\ &\le \eta(\|y^*\|^2 - \|y^\eta\|^2),\end{aligned}$$

where the last inequality follows from the optimality of y^η for the problem (A.1), i.e., $\nabla f_\eta(y^\eta)^\top(y^\eta - y^*) \le 0$, and the optimality of y^* for the problem (4.5). Then $\|y^\eta\| \le \|y^*\|$. Also, from $\|y^*\| \le 1$,

we deduce that if $\|y^\eta\| = 1$, then $y^* = y^\eta$. When $\|y^\eta\| < 1$, the only constraint in our domain is inactive, and thus we conclude that y^η is also optimum for the unconstrained minimization problem. Then the optimality conditions lead to $\nabla f_\eta(y^\eta) = 0$. This implies that $y^\eta = -(Q + (\eta - \lambda_Q)I_n)^{-1}g$. Moreover, y^* satisfies the optimality condition $\nabla f(y^*)^\top (y^* - y) \leq 0$ for all y such that $\|y\| \leq 1$. Since our domain is the unit ball, this is true if and only if $\nabla f(y^*) = -\alpha y^*$, for some $\alpha \geq 0$. Therefore, $y^* = -(Q + (\alpha - \lambda_Q)I_n)^\dagger g$, where $A^\dagger$ denotes the pseudo-inverse of a matrix A. If we denote the ordered eigenvalues of Q by q_i and their corresponding orthonormal eigenvectors by u_i, we obtain

$$\|y^\eta\|^2 = \sum_{i=1}^{n} \frac{(u_i^\top g)^2}{(q_i - q_n + \eta)^2} \quad \text{and} \quad \|y^*\|^2 = \sum_{i=1}^{n} \frac{(u_i^\top g)^2}{(q_i - q_n + \alpha)^2}.$$

Note that it is possible to have $\alpha = 0$ and $q_i - q_n = 0$. However, this happens only when $u_i^\top g = 0$, so we follow the convention $\frac{0}{0} = 0$. After some simple algebra, we have the equality

$$\begin{aligned}
\|y^*\|^2 - \|y^\eta\|^2 &= \sum_{i=1}^{n} \frac{(u_i^\top g)^2}{(q_i - q_n + \alpha)^2} - \sum_{i=1}^{n} \frac{(u_i^\top g)^2}{(q_i - q_n + \eta)^2} \\
&= (\eta - \alpha) \sum_{i=1}^{n} (u_i^\top g)^2 \frac{2q_i - 2q_n + \eta + \alpha}{(q_i - q_n + \alpha)^2 (q_i - q_n + \eta)^2}.
\end{aligned}$$

Since $\|y^*\| \geq \|y^\eta\|$, $\alpha \geq 0$, and $\eta > 0$, we must have $\eta \geq \alpha$. Also, $\eta \leq \alpha$ is possible only if $y^\eta = y^*$. Hence, we have

$$\begin{aligned}
\|y^*\|^2 - \|y^\eta\|^2 &= (\eta - \alpha)_+ \sum_{i=1}^{n} (u_i^\top g)^2 \frac{2q_i - 2q_n + (\eta - \alpha)_+ + 2\alpha}{(q_i - q_n + \alpha)^2 (q_i - q_n + (\eta - \alpha)_+ + \alpha)^2} \\
&\leq (\eta - \alpha)_+ \sum_{i=1}^{n} (u_i^\top g)^2 \frac{2q_i - 2q_n + (\eta - \alpha)_+ + 2\alpha}{(q_i - q_n + \alpha)^4} \\
&= 2(\eta - \alpha)_+ \sum_{i=1}^{n} \frac{(u_i^\top g)^2}{(q_i - q_n + \alpha)^3} + (\eta - \alpha)_+^2 \sum_{i=1}^{n} \frac{(u_i^\top g)^2}{(q_i - q_n + \alpha)^4}.
\end{aligned}$$

This shows that $\|y^*\|^2 - \|y^\eta\|^2 \leq \phi\eta + o(\eta)$, where $\phi = 2(y^*)^\top (Q + (\alpha - \lambda_Q)I_n)^\dagger y^*$. Therefore,

$$\|y^\eta - y^*\|^2 \leq \|y^*\|^2 - \|y^\eta\|^2 \leq \phi\eta + o(\eta).$$

Thus y^η has error $O(\sqrt{\eta})$, which is expected since the error in the objective function is $O(\eta)$, and the objective function is quadratic.

A.2 Computation of s value

Recall the notation $\tilde{y} = [y; \tilde{y}_{n+1}]$ and $\tilde{x} = [\tilde{y}; x_{n+1}; x_{n+2}]$. For the set Y in (4.13), Condition 4.23 is satisfied by construction, and Condition 4.24 is satisfied by taking $\tilde{x}' = [y'; \tilde{y}'_{n+1}; x'_{n+1}; x'_{n+2}]$ with $y' = 0$, $\tilde{y}'_{n+1} = \frac{1}{2}$, $x'_{n+1} = 1$ and $x'_{n+2} = 0$. This ensures that for any $t \in [0, 1]$, we have

$$\tilde{W}_t = (1-t)\tilde{W}_0 + t\tilde{W}_1 = \begin{bmatrix} (1-t)I_{n+1} + t\tilde{Q} & t\tilde{g} & 0 \\ t\tilde{g}^\top & t-1 & 0 \\ 0^\top & 0 & t \end{bmatrix}, \tag{A.2}$$

and $(\tilde{x}')^\top \tilde{W}_t \tilde{x}' = (\tilde{x}')^\top((1-t)\tilde{W}_0 + t\tilde{W}_1)\tilde{x}' < 0$. Thus, by the variational characterization of eigenvalues, $\tilde{W}_t$ has at least one negative eigenvalue. Also, Condition 4.25(ii) is now satisfied.

We next show that the precise value of s is simply determined by λ_Q.

Lemma A.1. *Suppose $\lambda_Q < 0$. Consider $\tilde{W}_0, \tilde{W}_1$ as defined in (4.12). Then, the maximal $t \in [0,1]$ that ensures that the matrix $\tilde{W}_t$ in (A.2) has a single negative eigenvalue for all $t \in [0,s]$, is invertible for all $t \in (0,s)$, and $\tilde{W}_s$ is singular is given by*

$$s = \frac{1}{1-\lambda_Q} \in (0,1).$$

Proof. Define $\hat{s} := \frac{1}{1-\lambda_Q} \in (0,1)$. From (A.2), note that $\tilde{W}_t$ has a block structure and s is such that it equals to the smallest positive t ensuring

$$V_t := (1-t)\begin{bmatrix} I_{n+1} & 0 \\ 0 & -1 \end{bmatrix} + t\begin{bmatrix} \tilde{Q} & \tilde{g} \\ \tilde{g}^\top & 0 \end{bmatrix}$$

is singular.

Let $\lambda_{n+2,t}, \lambda_{n+1,t}$ be the two smallest eigenvalues of V_t, and let $\rho_{n+1,t}, \rho_{n,t}$ be the two smallest eigenvalues of $(1-t)I_{n+1} + t\tilde{Q}$. Notice that $(1-t)I_{n+1} + t\tilde{Q}$ has the same eigenvectors as $\tilde{Q}$, and the eigenvalues are simply scaled and shifted from those of Q, thus the minimum eigenvalue of $(1-t)I_{n+1} + t\tilde{Q}$ is $1 - t + t\lambda_Q$ for $t \in (0,1)$. Also, by construction, the multiplicity of λ_Q in $\tilde{Q}$ is at least two, so the multiplicity of the minimum eigenvalue of $(1-t)I_{n+1} + t\tilde{Q}$ is also at least two; therefore, $\rho_{n+1,t} = \rho_{n,t} = 1 - t + t\lambda_Q$.

For any $t \in (0,1)$, the last diagonal entry of V_t is negative implying V_t is not positive semidefinite, hence $\lambda_{n+2,t} < 0$. However, for $t \in (0,\hat{s})$, $\rho_{n+1,t} > 0$, and from Cauchy's interlacing theorem for eigenvalues [85, Theorem 4.3.17], we obtain

$$\lambda_{n+2,t} < 0 < \rho_{n+1,t} = \lambda_{n+1,t} = \rho_{n,t}, \quad t \in (0,\hat{s}).$$

Thus, for any $t \in (0,\hat{s})$, the matrix V_t, and hence $\tilde{W}_t$, is invertible, and $\tilde{W}_t$ has exactly one negative eigenvalue. When $t = \hat{s}$, $\rho_{n+1,\hat{s}} = \rho_{n,\hat{s}} = 1 - \hat{s} + \hat{s}\lambda_Q = 0$. By recalling that $\tilde{Q} := \begin{bmatrix} Q & 0 \\ 0 & \lambda_Q \end{bmatrix}$ and $\tilde{g} = [g;0]$, we immediately observe that $V_{\hat{s}}$, and thus $\tilde{W}_{\hat{s}}$, is singular since $V_{\hat{s}}$ has eigenvector $[y;\tilde{y}_{n+1};x_{n+1}] = [0;1;0]$ with eigenvalue 0. Also,

$$\lambda_{n+2,\hat{s}} < 0 = \rho_{n+1,\hat{s}} = \lambda_{n+1,\hat{s}} = \rho_{n,\hat{s}}$$

so $\tilde{W}_{\hat{s}}$ has exactly one negative eigenvalue. Moreover, for any $t > \hat{s}$, the minimum eigenvalue of $(1-t)I_n + tQ$ is $1 - t + t\lambda_Q < 0$. Hence, for any $t > \hat{s}$, $\lambda_{n+2,t} \le \rho_{n+1,t} = \lambda_{n+1,t} = \rho_{n,t} < 0$ follows from [85, Theorem 4.3.17]. As a result V_t, and thus $\tilde{W}_t$, has at least two negative eigenvalues. Therefore, $s = \hat{s} = \frac{1}{1-\lambda_Q}$ is the correct value. □

Choosing $\tilde{x}'' = [y'';\tilde{y}''_{n+1};x''_{n+1};x''_{n+2}]$ with $y'' = 0$, $\tilde{y}''_{n+1} = 1$, $x''_{n+1} = 0$, and $x''_{n+2} = 0$ ensures that $\tilde{x}'' \in \mathrm{Null}(\tilde{W}_s)$, $(\tilde{x}'')^\top \tilde{W}_1 \tilde{x}'' < 0$, and $x''_{n+1} = 0$. This simultaneously verifies Conditions 4.26 and 4.27.

Appendix B

Appendix to Chapter 5

B.1 Existing Approaches to Non-Parametric Choice Estimation

In this section, we examine the existing approaches to learn the non-parametric choice model, i.e., infer an appropriate probability vector λ using the data collected via the process outlined in Section 5.2, and demonstrate how they are particular instantiations of our general model. For a fixed subset $\mathcal{A}_j$, $j \in [m]$, we denote the collection of associated choice probabilities as $A_j\lambda = \{\mathbb{P}_\lambda[i \mid \mathcal{A}_j]\}_{i\in\mathcal{A}_j} \in \Delta_{|\mathcal{A}_j|}$.

B.1.1 Revenue Prediction Approach

Let r_i be the revenue of item $i \in [n]$. Then the expected revenue of an assortment $\mathcal{A} \subset [n]$ under distribution λ is $\sum_{i\in\mathcal{A}} r_i\mathbb{P}_\lambda[i \mid \mathcal{A}]$. Farias et al. [62] seek to find the worst-case expected revenue from a distribution λ consistent with the given data in the sense that the theoretical probabilities $\mathbb{P}_\lambda[i \mid \mathcal{A}_j] = \langle a_{ij}, \lambda\rangle$ are precisely consistent with their empirical estimates p_{ij}. Since the probabilities $\mathbb{P}_\lambda[i \mid \mathcal{A}]$ are linear in λ, this can be formulated as a linear program (LP)

$$\min_{\lambda} \left\{ \sum_{i\in\mathcal{A}} r_i\mathbb{P}_\lambda[i \mid \mathcal{A}] : \; A\lambda = p, \; \lambda \in \Delta_{n!} \right\}.$$

We first make a few observations related to this model of Farias et al. [62]. In fact, when $\mathcal{A} = \mathcal{A}_j$ for some $j \in [m]$, we have $\mathbb{P}_\lambda[i \mid \mathcal{A}] = \langle a_{ij}, \lambda\rangle = p_{ij}$ due to the constraints $A\lambda = p$, hence the objective is constant. Thus the LP becomes a feasibility problem

$$\text{find} \quad \lambda \in \Delta_{n!} \quad \text{s.t.} \quad A\lambda = p. \tag{B.1}$$

That said, (B.1) is still computationally intractable even for moderate values of n because it involves $n!$ variables. Nonetheless, the dual of (B.1) admits the following robust LP interpretation:

$$\max_{\beta,\nu} \left\{ \langle \beta, p\rangle - \nu : \; \max_{\sigma\in S_n} \langle \beta, a(\sigma)\rangle \le \nu \right\}. \tag{B.2}$$

Note that verifying the feasibility of a solution with respect to the robust constraint in (B.2), i.e.,

$$\max_{\sigma\in S_n} \langle \beta, a(\sigma)\rangle = \max_{\sigma} \left\{ \sum_{j\in[m]} \sum_{i\in\mathcal{A}_j} \beta_{ij} a_{ij}(\sigma) : \; \sigma \in S_n \right\} \le \nu \tag{B.3}$$

is a combinatorial problem of the exact same form as (5.10). Farias et al. [62] suggests solving (B.2) either using the constraint sampling technique [46] or by building an approximation to its robust counterpart obtained from approximating the uncertainty sets with an efficiently representable polyhedron.

In fact, (B.1) can be seen as choosing $\lambda \in \Delta_{n!}$ to minimize a (very harsh) distance measure:

$$\min_{\lambda \in \Delta_{n!}} D(A\lambda, p), \quad D(A\lambda, p) = \begin{cases} 0, & A\lambda = p \\ \infty, & \text{otherwise.} \end{cases} \tag{B.4}$$

In general, and specifically when the observations are noisy, there is no guarantee that there exists $\lambda \in \Delta_{n!}$ to fit the data p exactly, i.e., $A\lambda = p$. To remedy this, van Ryzin and Vulcano [147] and Bertsimas and Mišic [27] examine approaches that use less harsh distance measures $D(\cdot, \cdot)$.

B.1.2 Maximum Likelihood Estimation Approach

van Ryzin and Vulcano [147] propose the following method to learn λ via maximum likelihood estimation (MLE). We next describe their method and provide an alternative interpretation of their approach as the minimization of a particular distance measure, namely Kullback-Leibler (KL) divergence, between the true distributions $A_j\lambda$ and their empirical estimates p_j.

By (5.1), each item-assortment pair $i \in \mathcal{A}_j$ is seen Kq_{ij} times amongst the observations $\left\{i^k, \mathcal{A}^k\right\}_{k=1}^K$. Based on this, the log-likelihood of the observation set $\left\{i^k, \mathcal{A}^k\right\}_{k=1}^K$ is $\sum_{j\in[m]} \sum_{i\in\mathcal{A}_j} Kq_{ij} \log\left(\langle a_{ij}, \lambda\rangle\right)$. Thus, ignoring the constant K factor, the MLE problem is

$$\max_{\lambda} \left\{ \sum_{j\in[m]} \sum_{i\in\mathcal{A}_j} q_{ij} \log\left(\langle a_{ij}, \lambda\rangle\right) : \ \lambda \in \Delta_{n!} \right\}. \tag{B.5}$$

Throughout, we use the convention that when $q_{ij} = \langle a_{ij}, \lambda\rangle = 0$, we set $q_{ij} \log(\langle a_{ij}, \lambda\rangle) = 0$. This implies that if the optimal solution λ to (B.5) has $\mathbb{P}_\lambda[i \mid \mathcal{A}_j] = \langle a_{ij}, \lambda\rangle = 0$, then we must have $q_{ij} = 0$ also, i.e., we did not observe any choices of i from $\mathcal{A}_j$ in our data either.

Like (B.1), the problem (B.5) is very large, with $n!$ variables. A column generation technique is suggested in van Ryzin and Vulcano [147] to get around this, i.e., solve (B.5) on a subset of the variables, and use the optimality conditions to add variables as needed. The MLE column generating subproblem is constructed as

$$\max_{\sigma} \left\{ \sum_{j\in[m]} \sum_{i\in\mathcal{A}_j} \frac{q_{ij} a_{ij}(\sigma)}{\langle a_{ij}, \lambda(S)\rangle} : \ \sigma \in S_n \right\}. \tag{B.6}$$

The solution $\lambda(S)$ is optimal if (B.6) $\leq K$, otherwise the column σ^* maximizing (B.6) is added to the set S, and the process is repeated. Note that (B.6) has the same form as (5.10) and (B.3).

We next demonstrate that the MLE problem (B.5) admits a nice interpretation between the empirical estimates $\{p_j\}_{j\in[m]}$ and the distributions $\{A_j\lambda\}_{j\in[m]}$. To observe this, let us rewrite the

objective in (B.5) as

$$\sum_{j\in[m]}\sum_{i\in\mathcal{A}_j} q_{ij}\log\left(\langle a_{ij},\lambda\rangle\right) = \sum_{j\in[m]} q_j \sum_{i\in\mathcal{A}_j} p_{ij}\log\left(\langle a_{ij},\lambda\rangle\right)$$
$$= -\sum_{j\in[m]} q_j \underbrace{\sum_{i\in\mathcal{A}_j} p_{ij}\log\left(\frac{p_{ij}}{\langle a_{ij},\lambda\rangle}\right)}_{=\mathrm{KL}(p_j,A_j\lambda)} + \underbrace{\sum_{j\in[m]} q_j \sum_{i\in\mathcal{A}_j} p_{ij}\log(p_{ij})}_{=\text{constant}}$$

where $\mathrm{KL}(a,b)$ is the KL divergence between two probability distributions a and b. Hence, (B.5) is equivalent to solving

$$\min_{\lambda}\left\{\sum_{j\in[m]} q_j\,\mathrm{KL}(p_j,A_j\lambda):\ \lambda\in\Delta_{n!}\right\}. \tag{B.7}$$

Thus, by defining $D(A\lambda,p)=\sum_{j\in[m]} q_j\,\mathrm{KL}(p_j,A_j\lambda)$, we see that the MLE approach is equivalent to (B.4) but with a different distance measure $D(\cdot,\cdot)$.

B.1.3 Norm-Minimization Approach

As opposed to the approaches outlined in Sections B.1.1 and B.1.2, in order to estimate a non-parametric choice model λ, Bertsimas and Mišic [27] suggest minimizing the ℓ_1-norm of $p-A\lambda$ by solving

$$\min_{\lambda}\left\{\|p-A\lambda\|_1:\ \lambda\in\Delta_{n!}\right\}. \tag{B.8}$$

In fact, (B.8) can be cast as an LP, but it is still computationally intractable since the dimension of λ is $n!$. Similar to van Ryzin and Vulcano [147], Bertsimas and Mišic [27] addresses this computational difficulty via a column generation approach. Again, (B.8) is of the same form as (B.4) where the distance measure $D(\cdot,\cdot)$ is selected to be $D(A\lambda,p)=\|p-A\lambda\|_1$. Furthermore, the resulting column generating subproblem is of the form

$$\max_{\sigma}\left\{\sum_{j\in[m]}\sum_{i\in\mathcal{A}_j}\beta_{ij}(S)a_{ij}(\sigma)-\nu(S):\ \sigma\in S_n\right\}, \tag{B.9}$$

where $\beta(S)$ and $\nu(S)$ are from the dual solution to solving (B.8) on a subset of columns $\sigma\in S\subset S_n$. Again, this subproblem has the same form as (5.10), (B.3) and (B.6).

B.2 Supplementary Computational Results

We present plots for our performance metrics under different ground truth choice model generation parameters $K=1,5,10$ and $L=5,10,100$, as well as using a different number of training subsets $m=10,20,50$. For each parameter combination, we generated 100 ground truth choice models.

B.2.1 Comparison of different K and L

Figures B.1, B.2 and B.3 show plots for the ground truth models generated for $L=5,10,100$ respectively, with different $K=1,5,10$, while keeping $m=20$ fixed. In almost all of the cases, we observe the same behavior as before: there are only minor differences in test MAE, but for model sparsity and algorithm efficiency the non-smooth dual approach outperforms the others. The one exception is the pure MNL setup with large L value, i.e., Figure B.3a for $K=1$, $L=100$, and the

distance measure $D(\cdot, \cdot)$ is based on ℓ_∞-norm; in such a case both smoothed primal and smoothed dual approaches slightly outperform the non-smooth dual approach in terms of the model sparsity and the average number of iterations. We note that this setting with $K = 1$, $L = 100$ may be considered as an unrealistic ground truth choice model in practice: having only $K = 1$ multinomial logit segment is rare in practice (hence the wide literature on alternative choice models), and a very large L value, e.g., $L = 100$, means that the utilities for top items are quite exaggerated. Furthermore, when comparing Figures B.1a, B.2a and B.3a (i.e., $K = 1, L = 5, 10, 100$) we notice that the number of iterations for the non-smooth dual approach stays relatively consistent as we increase L, but the smooth approaches decrease for higher L. In general, the algorithm efficiency of the non-smooth dual approach is relatively consistent across the different parameter regimes, whereas higher variations are observed for the others.

B.2.2 Comparison of different m

In Figure B.4, we test the effect of m by varying $m \in \{10, 20, 50\}$ while using a similar ground truth model from the main paper with $K = L = 5$. We observe that as m increases, the test MAE goes down, but the model sparsity and the number of iterations to convergence increases across all different approaches and norms used. This is as expected, since having more training subsets should allow us to fit better models, but increases the dimension of the choice probability set X. Our conclusions regarding the comparison of different approaches remain essentially the same: the non-smooth dual approach still outperforms the others.

B.2.3 Dynamic experiments with different norms

Figures B.5, B.6 and B.7 show the performance metrics in the dynamic data setting from varying the norm used between ℓ_1, ℓ_2, and ℓ_∞ respectively. The conclusions are the same as before, except when ℓ_∞ is used, the smoothed dual approach has comparable efficiency to the primal approach.

Figure B.1: Performance metrics in the static setup for different K, fixing $L = 5, m = 20$.

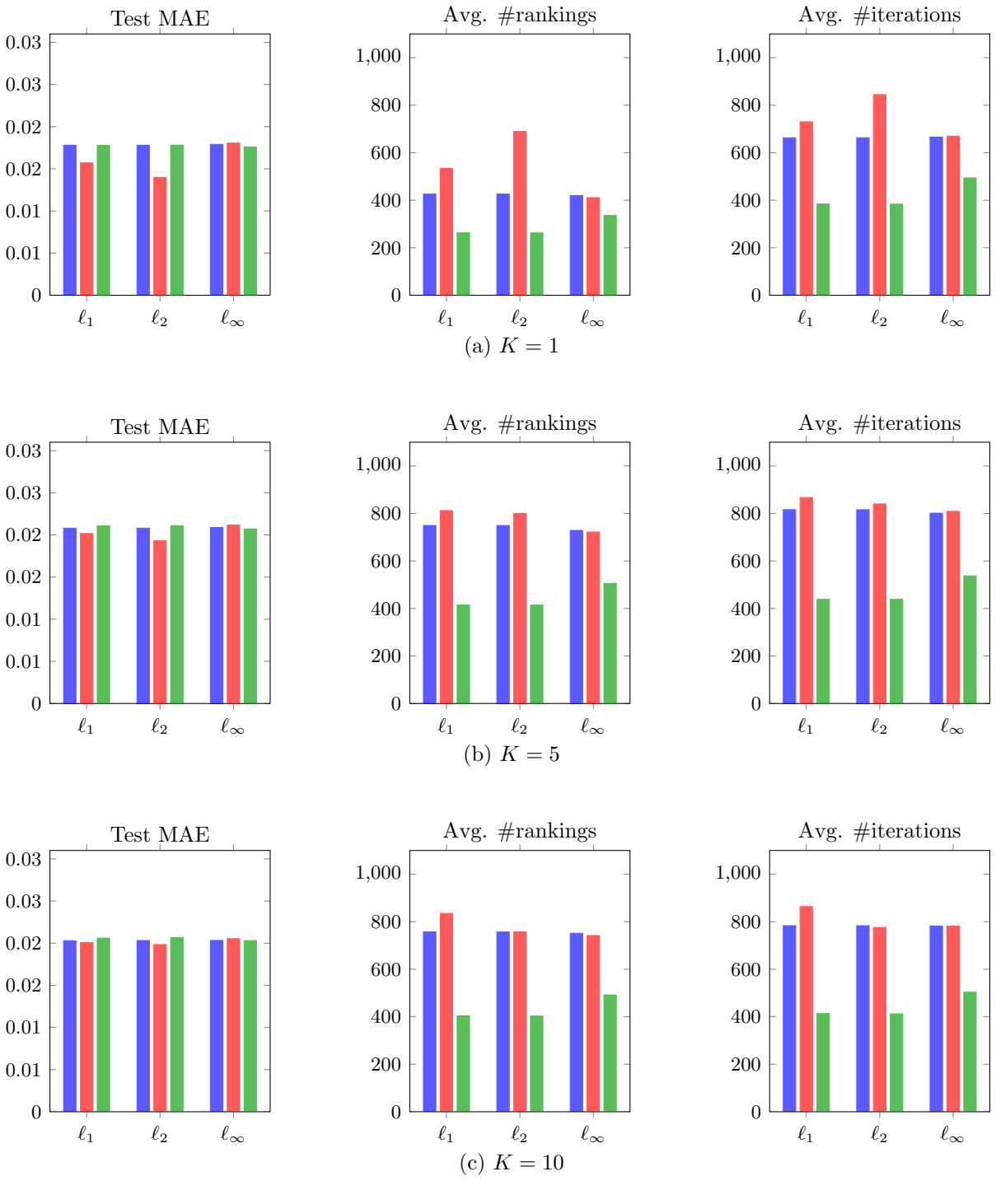
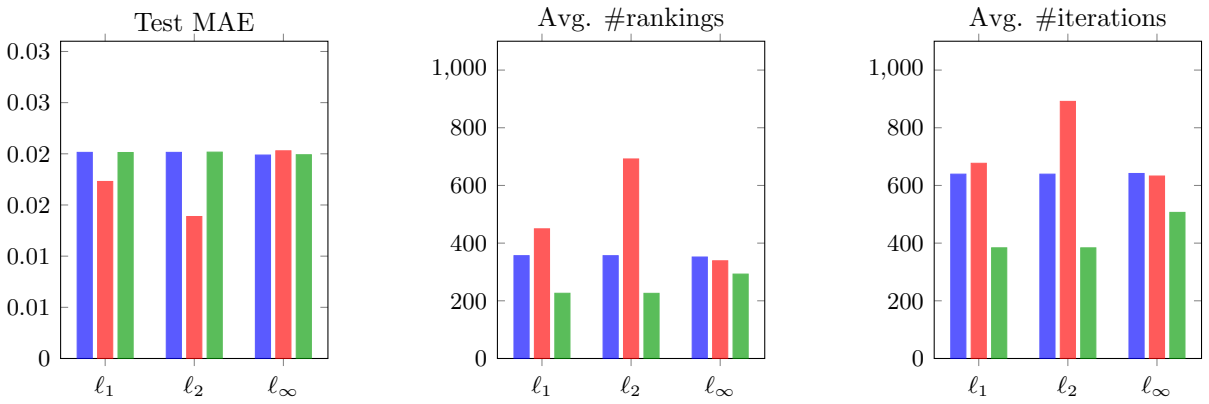
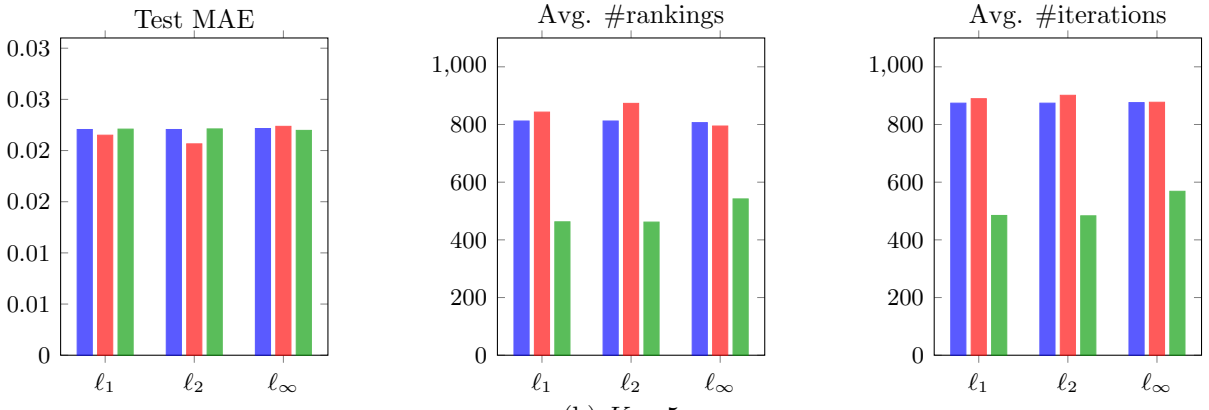


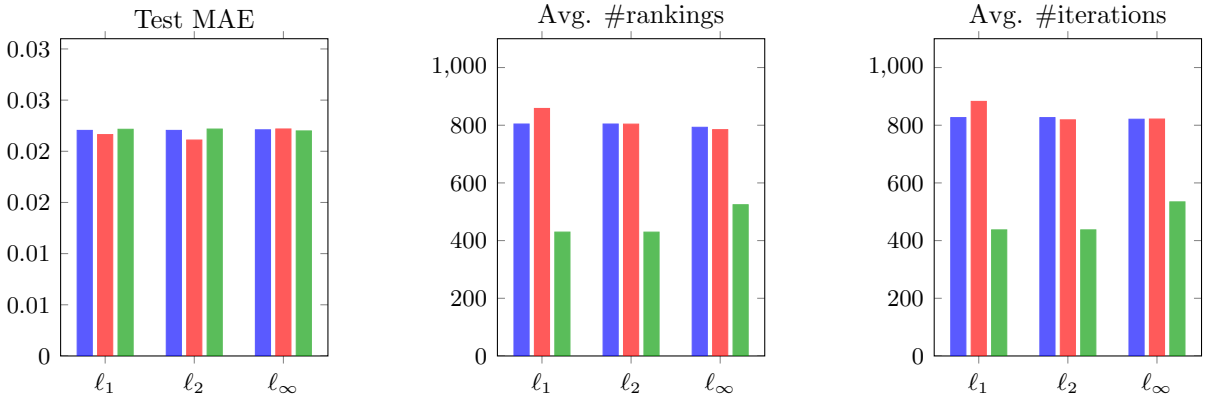
Figure B.2: Performance metrics in the static setup for different K, fixing $L = 10, m = 20$.



(a) $K = 1$



(b) $K = 5$



(c) $K = 10$

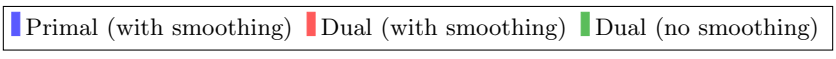
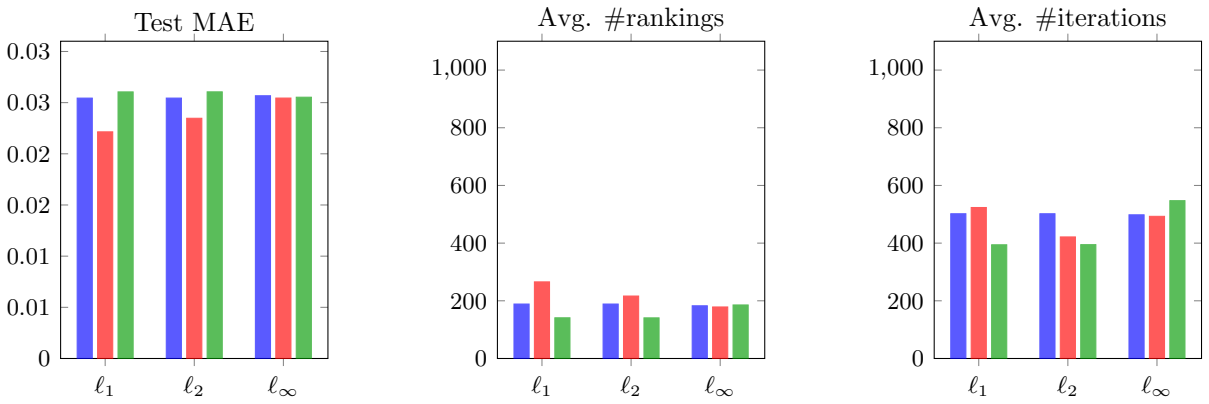
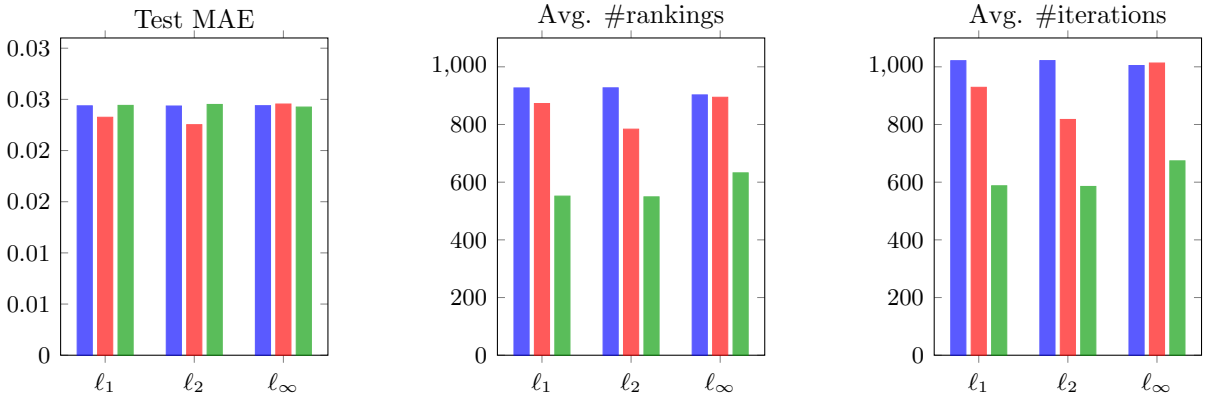


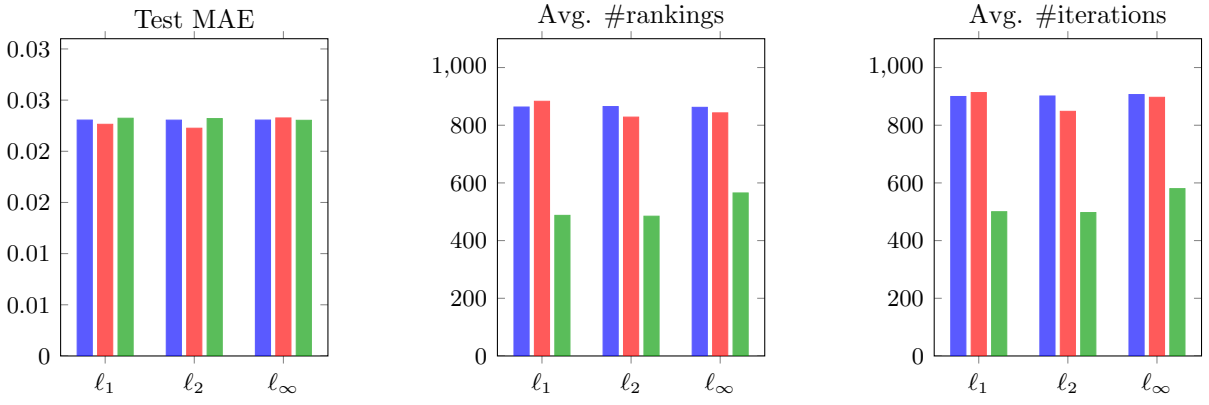
Figure B.3: Performance metrics in the static setup for different K, fixing $L = 100, m = 20$.



(a) $K = 1$



(b) $K = 5$



(c) $K = 10$

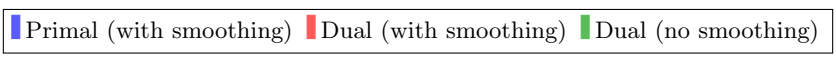
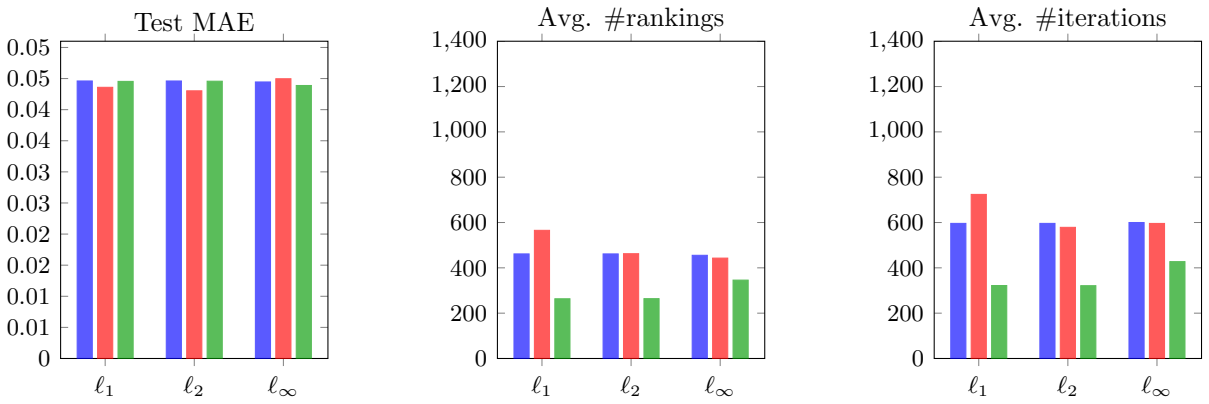
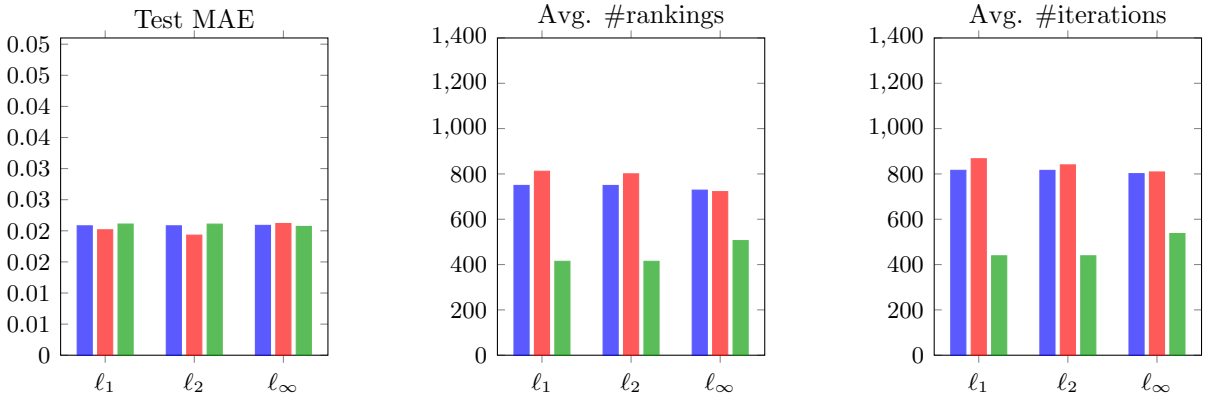


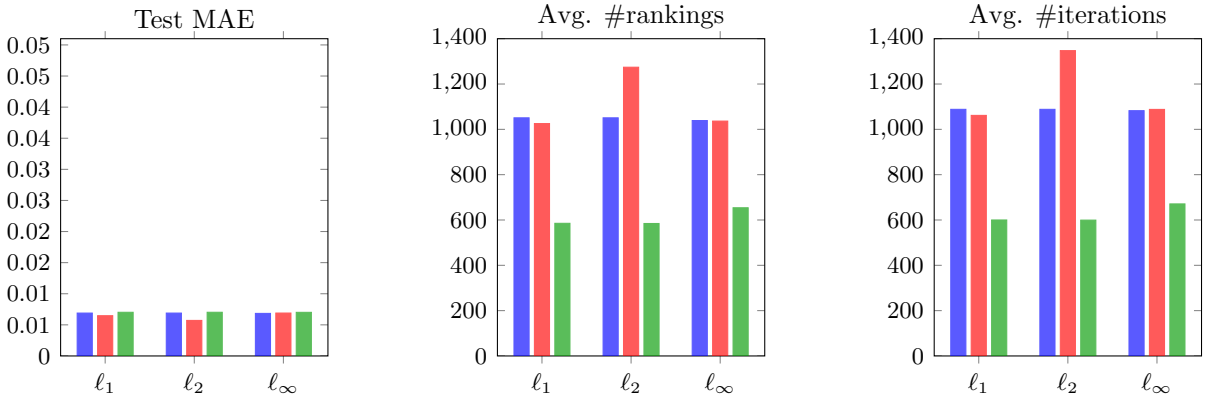
Figure B.4: Performance metrics in the static setup for different m, fixing $K = L = 5$.



(a) $m = 10$



(b) $m = 20$



(c) $m = 50$

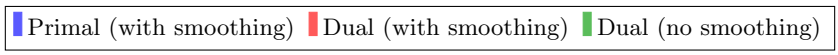


Figure B.5: Performance metrics using ℓ_1-norm for dynamic data.

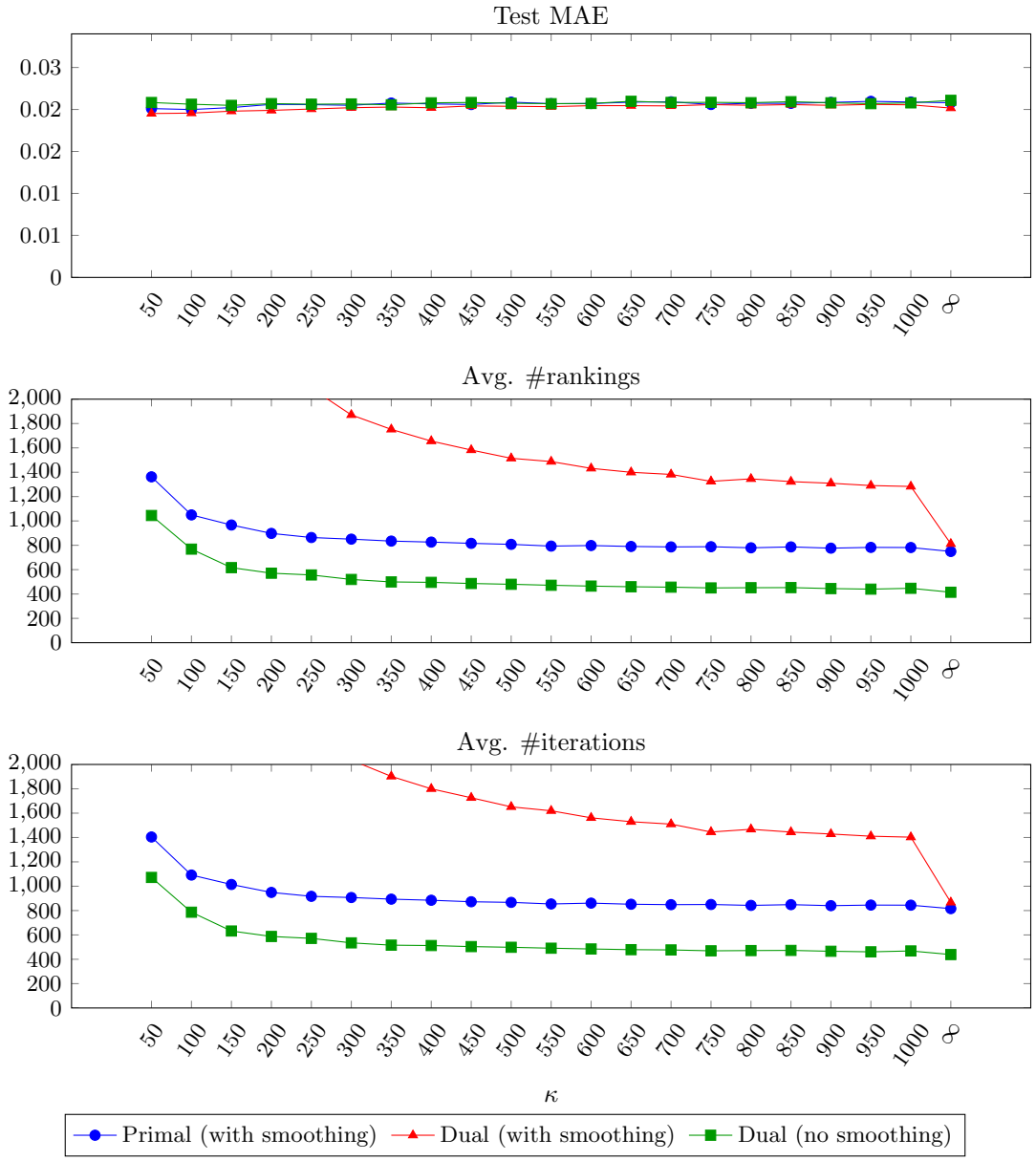


Figure B.6: Performance metrics using ℓ_2-norm for dynamic data.

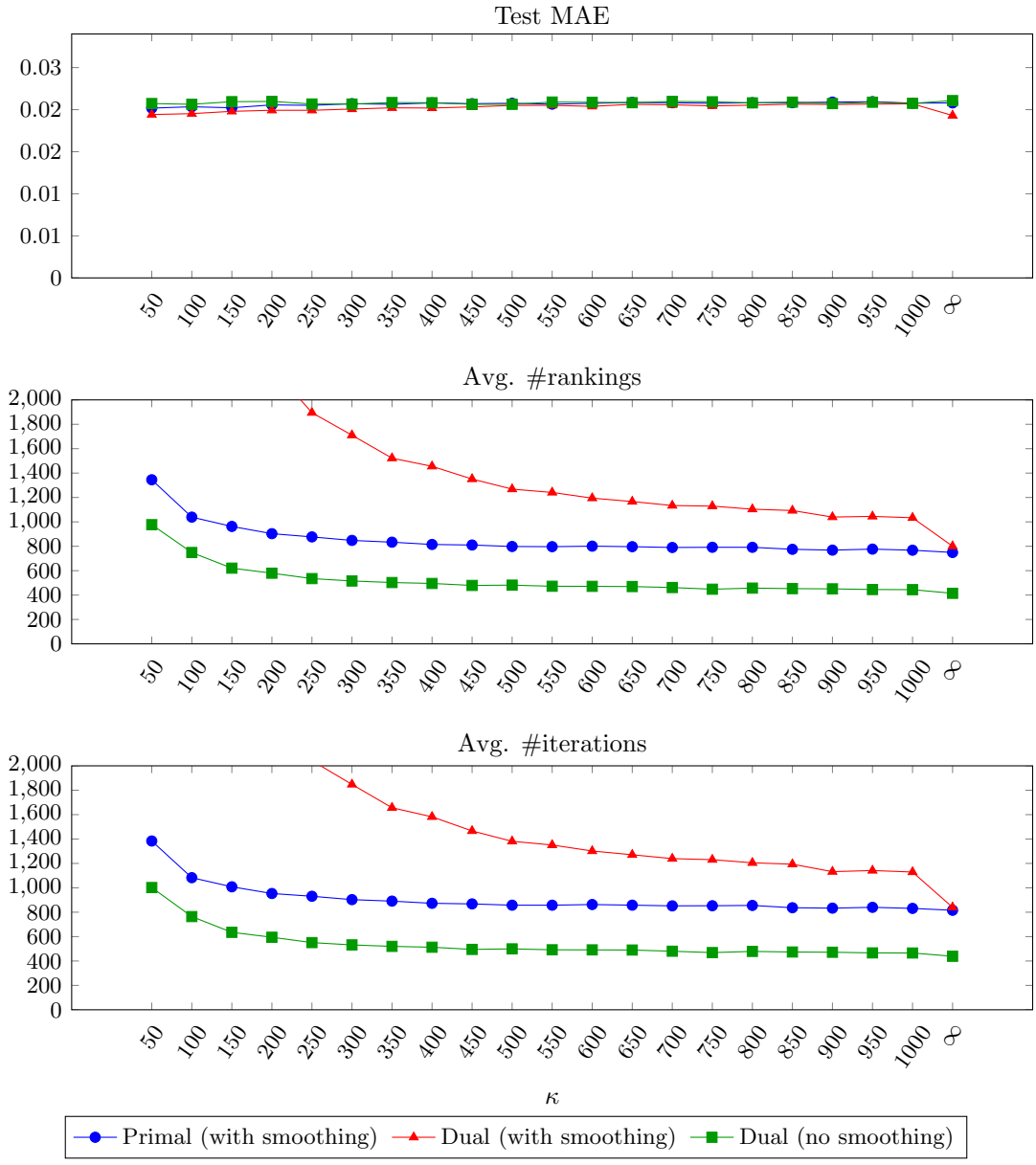


Figure B.7: Performance metrics using ℓ_∞-norm for dynamic data.