RISK-AVERSE OPTIMIZATION: MODELS, ALGORITHMS, AND APPLICATIONS IN MACHINE LEARNING

YASSINE LAGUEL

PhD. Thesis

JURY

RAPPORTRICE	Claudia Sagastizábal
RAPPORTEUR	Joseph Salmon
EXAMINATEUR	Alexandre d'Aspremont
EXAMINATEUR	Mert Gürbüzbalaban
EXAMINATEUR	Panayotis Mertikopoulos
PRÉSIDENTE	Nadia Brauner
DIRECTEUR DE THÈSE	Jérôme Malick
MEMBRE INVITÉ	Zaid Harchaoui
MEMBRE INVITÉ	Wim Van Ackooij

Université Grenoble Alpes
Laboratoire Jean Kuntzmann

PREFACE

This thesis deals with optimization under uncertainty, which has a long history in operations research and mathematical optimization. This field is currently challenged by applications in artificial intelligence and data science, where risk management has become a crucial issue. In this thesis, we consider nonsmooth optimization problems involving risk measures and coming from statistical learning applications. We pay a special attention to the risk measure called the superquantile (also known as the "Conditional Value at Risk") and we show how, in various contexts, it may enforce robustness for decision-making under uncertainty.

First, we consider convex risk measures admitting a representation in terms of superquantiles. We derive first order oracles with optimal computational complexity. These approximate oracles involve different smoothing techniques for which we propose a unified analysis. We also propose an efficient implementation of these oracles, coupled with a series of classical optimization methods, in an open-source software in Python. We show empirically, on classification and regression tasks, that the predictions obtained are robust to data shifts.

We then consider chance-constrained optimization problems. We propose a reformulation of these problems in the form of bilevel programs that involve the superquantile. We propose a (semi-) exact penalization for this reformulation, which we treat with a bundle method. We implement our bilevel approach in an open-source python software, which we illustrate on non-convex problems.

Finally, we investigate the use of the superquantile for federated learning. We consider the case of users with heterogeneous data distributions and we show how the superquantile allows for better performances on non-conforming users. We propose an algorithm adapted to the constraints of federated learning, in terms of communications and data privacy. We prove its theoretical convergence in the convex case by controlling the drift induced by local SGD and the dynamic reweighting induced by superquantiles. We also propose an in-depth numerical study of our algorithm and compare its performance with several established baselines.

RÉSUMÉ

Cette thèse s'inscrit dans le cadre de l'optimisation sous incertitude, qui a une longue tradition en recherche opérationnelle et en optimisation mathématique. Ce domaine trouve aujourd'hui de nouvelles applications en intelligence artificielle et science des données, où la prise en compte du risque est devenu une question cruciale. Dans cette thèse, nous considérons des problèmes d'optimisation, issus d'applications en apprentissage statistique, mettant en jeu des mesures de risque. Nous accordons une attention particulière à la mesure de risque appelée superquantile (également connue sous le nom de "Conditional Value at Risk") et montrons comment, dans divers contextes, elle permet d'obtenir de la robustesse dans la prise de décision.

Dans un premier temps, nous nous intéressons aux mesures de risque convexes admettant une représentation en termes de superquantiles. Nous dérivons des oracles du premier ordre avec une complexité de calcul optimale. Ces oracles approchés font intervenir différentes techniques de lissage pour lesquelles nous proposons une analyse unifiée. Nous proposons aussi une implémentation efficace de ces oracles, couplée à une série de méthodes classiques d'optimisation, dans un logiciel open-source en Python. Nous montrons empiriquement, sur des problèmes de classifications et de régression, que les prédictions obtenues sont robustes aux perturbations des données.

Nous nous penchons ensuite sur les problèmes d'optimisation avec contraintes en probabilités. Nous proposons une reformulation de ces problèmes sous la forme de problèmes bi-niveaux qui font apparaître le superquantile. Nous proposons une pénalisation (semi-)exacte pour cette reformulation, que nous traitons avec une méthode de faisceaux. Nous implémentons notre approche bi-niveau, dans un logiciel open-source, que nous illustrons sur des problèmes non-convexes.

Enfin, nous nous penchons sur l'utilisation du superquantile dans le cadre de l'apprentissage fédéré. Nous considérons le cas d'utilisateurs aux distributions de données hétérogènes et montrons comment le superquantile permet d'obtenir de meilleurs performances sur les utilisateurs les moins privilégiés. Notre algorithme est adapté aux contraintes réelles, en terme de communications et de protection des données. Nous en démontrons la convergence théorique dans le cas convexe en contrôlant simultanément la dérive des modèles locaux induite par la méthode de descente du gradient stochastique locale, ainsi que la redistribution de poids induite par le superquantile. Nous proposons aussi une étude numérique approfondie de notre algorithme en le comparant à un ensemble d'algorithmes constituant l'état de l'art en apprentissage fédéré.

CONTENTS

1

INTRODUCTION

This section serves as an overview of the content of this manuscript. We first briefly introduce the topic of optimization under uncertainty with an emphasis on probabilistic approaches and convex risk measure theory. We then lay down the philosophy we adopt for the writing of this manuscript. We finally give a detailed summary of our contributions.

1.1 OPTIMIZING UNDER UNCERTAINTY

The proliferation of mobile phones, wearables and edge devices has led to an unprecedented growth in the generation of user interaction data. Simultaneously, we observe in various branches of sciences (e.g. biology, astrophysics, behavioural sciences, etc.) a steady commitment in the acquisition of massive datasets to better describe contemporary questions. Tapping into the power of this rich data promises to greatly improve the next generation of intelligent applications, devices, and decision-making systems. However, this upheaval comes with a number of challenges, stemming either from technical limitations (upscaling the learning processes, heterogenous data, dealing with communication constraints) or societal concerns (safety, privacy, fairness, carbon footprint). The data involved are thus for decision-makers an important source of uncertainty but also a great opportunity to tailor their strategy in accordance.

Stochastic optimization

Such uncertainty is typically framed as a random variable $\xi : \Omega \to \mathbb{R}^{\mathrm{m}}$ that is liable to affect a cost function $f(w, \xi)$ (or a constraint $g(w, \xi) \leq 0$) for a given decision w. In many cases, the random function $f : \mathbb{R}^d \times \mathbb{R}^{\mathrm{m}} \to \mathbb{R} \cup \{+\infty\}$ (resp. g) is evaluated in expectation and the decision maker has the task to find an optimal solution $w^\star$ of the associated *stochastic optimization problem*

$$\min_{w \in \mathbb{R}^d} \quad \mathbb{E}\left[f(w, \xi)\right]. \tag{1.1}$$

This approach has proved its relevance and efficiency in a number of applications; see for instance the references in the textbook [149]. However, when high stakes are involved, an estimation in expectation is unlikely to hedge decision makers against risky outcomes. Fortunately, various options may be considered to better handle these worst-case events.

Robust optimization

Robust Optimization. Robust optimization (see e.g. the textbook [12]) has played a significant role in the management of risk for a large portion of optimization problems under uncertainty. The idea is to consider worst-case possible outcomes in a specific uncertainty set and design conservative solution in accordance with them. Formally, assuming that ξ lies almost surely in a subset $\Lambda \subset \mathbb{R}^{\mathrm{m}}$, one typically removes all the stochasticity from (1.1) by considering the minimax variant

$$\min_{w \in \mathbb{R}^{\mathrm{d}}} \max_{z \in \Lambda} f(w, z).$$

Though often computationally appealing, robust reformulations can lead to unnecessarily conservative solutions. We have other modelling options, assuming some a priori information on the randomness of ξ.

Probabilistic Approaches. We may consider for instance a probabilistic recasting of (1.1). That is, given a safety threshold $M \in \mathbb{R}$ set by the modeler, we may try to maximize the probability of landing below M:

$$\max_{w \in \mathbb{R}^d} \mathbb{P}\left[f(w,\xi) \leq M\right]. \tag{1.2}$$

Chance constraints

Probabilistic approaches [158] are also frequently considered for stochastic constraints $g(x,\xi) \leq 0$. Specifically, the modeler fixes an appropriate safety probability threshold $p \in (0,1)$ and consider having the constraint

$$\mathbb{P}\left(g(w,\xi) \leq 0\right) \geq p$$

on his system. These constraints are hard to deal with, due to the non-continuity and non-convexity of the probabilistic function $x \mapsto \mathbb{P}\left(g(w,\xi) \leq 0\right)$. We dedicate Chapter 5 of this thesis to a new approach to handle them in practice.

Quantile-based Optimization. A similar approach to (1.2) consists in minimizing a quantile of the random cost function. Precisely, for a fixed probability threshold $p \in [0,1)$, one can consider in place of (1.2), the problem

$$\min_{w \in \mathbb{R}^d} Q_p(f(w,\xi)),$$

Quantile function

Here, Q_p denotes the *quantile function*, defined as the generalized inverse of the cumulative distribution function:

$$Q_p(X) = \inf\{t \in \mathbb{R} : \ \mathbb{P}[X \leq t] \geq p\}, \quad \forall p \in [0,1). \tag{1.3}$$

This approach has been particularly favoured in the financial industry [51] where the quantile of a given portfolio investment is called the *Value at Risk*. From an optimization perspective, they remain hard to treat due to the potential nonsmoothness and non-convexity of quantile functions.

Price of failure

Superquantile-based Optimization. Both quantile-based and probabilistic programs are designed to ensure the success of a given strategy with high probability. However, they do not account for the price of failure situations and thus open the door to possibly disastrous outcomes. If one wishes instead to tackle risky events, convex risk measures [138] may be a fruitful alternative. Under reasonable assumptions (see recalls in Chapter 3) these risk measures admit a dual formulation leading to problems of the form

$$\min_{w \in \mathbb{R}^d} \max_{q \in \mathcal{A}} \mathbb{E}_q\left(f(w,\xi)\right).$$

Superquantile

Here, the expectation $\mathbb{E}_q$ is taken with respect to the worst distribution q in a set $\mathcal{A}$ of distributions which depends on the risk measure considered. This formulation make appear the distributional robustness brought by the convex risk measures. In this thesis we focus on a popular risk measure: the superquantile – also called the Conditional Value at Risk, Tail Value at Risk, Mean Excess Loss, or Mean Shortfall. Generally speaking, given an integrable random variable $X : \Omega \to \mathbb{R}$, and a safety probability threshold $p \in (0,1)$,

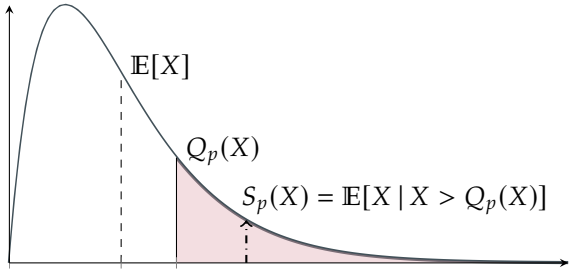


Figure 1.1: For a continuous random variable X, drawing of p-quantile $Q_p(X)$, and p-superquantile $S_p(X)$, defined as an expectation.

the superquantile $S_p(X)$ is defined as the mean of quantiles greater than the p-quantile,

$$S_p(X) = \frac{1}{1-p} \int_{p'=p}^{1} Q_{p'}(X) \mathrm{d}p'. \tag{1.4}$$

Link with quantiles and cumulative distribution function

Superquantiles can be therefore interpreted as a measure of the upper tail of the distribution of X with the parameter p controlling the sensitivity to high losses. One easily deduce from the above definition that the superquantile is a continuous, non-decreasing function of p and that it is a continuous, positively homogeneous function of X. For a given random variable X, as a consequence of [137, Th. 2], one can recover from the cumulative distribution function F_X of X both the p-quantile and the p-superquantile functions as functions of p. Reciprocally, knowing either the p-quantile or the p-superquantile for all $p \in [0, 1]$ suffices to recover F_X.

From a statistical viewpoint, these three notions are also equally consistent [136, Th. 4] in the sense that convergence in distribution for a sequence of random variables $(X_n)_{n\geq 0}$ is equivalent to the pointwise convergence of the two sequences of functions $p \mapsto Q_p(X_n)$, $p \mapsto S_p(X_n)$. This is particularly relevant when the distributions are observed through data sampling. We can use the empirical cumulative distribution functions, quantiles and superquantiles all while upholding asymptotic convergence as the sample size grows.

In this thesis, we will exploit the properties of the superquantile under various angles to tackle challenging data-driven applications.

1.2 ABOUT THIS MANUSCRIPT

Before diving into the contributions I made during my PhD time, I would like to share with the reader the philosophy I adopted for this writing work. The few remarks below clarify the perspective with which I introduced my work as well as the scientific approach I adopted in each chapter of the contributions.

At the beginning of the writing of this thesis, I asked myself which coloration I wanted this work to have. The last three years had indeed given me the opportunity to work on different fields of research related to optimization which could not all be gathered in this manuscript without compromising its unity. Thus, I decided to select three research projects centered around the same topic of interest: the practical minimization of risk measures.

Chapters 2 and 3 of this thesis serve as a preliminary introduction to this work. First, I sought to stress the broad number of applications that probabilistic constraint and convex risk measures may encompass. Although my PhD work was not directly tied to industrial applications, I found important to illustrate

their relevance in various contexts and this led me to dedicate Chapter 2 to this task. Second, I wanted to emphasize in Chapter 3 the interplay between convex analysis and (distributional) robustness for the design and study of risk measures. Besides building upon the foundational works of A. Ruszczyński, A. Shapiro, and T. Rockafellar to provide such interpretation in a general and abstract context, I wished to showcase on a key class of risk measures, with elementary proofs, how dual reasoning could yield a geometric interpretation of risk aversion.

Such geometric considerations are the cornerstone of many rationales in my own contributions. This is why a consistent effort has been put into illustrating the concepts involved in each development of this thesis with figures, examples, and numerical experiments. I genuinely hope those will facilitate the reading of the more technical developments.

My personal contributions are contained in Chapters 4, 5, and 6. In each chapter, I sought to keep an *operational* approach with a balance between theoretical and practical contributions. Specifically, whenever this was possible, I tried to relate the situation considered to an existing concept of mathematical optimization. In this sense, the titles of my sections are chosen to highlight the different concepts in optimization and convex analysis that come into play in this thesis. The practical implementation of these ideas and concepts stands also as an important contribution of my work. This is reflected in particular in Sections 4.3 and 5.4 with the short presentations of my software packages and in Section 6.5 where I present the experimental setting for the large-scale experiments in federated learning.

1.3 OUTLINE AND SPECIFIC CONTRIBUTIONS

Chapters 2 and 3 are preliminary chapters that gather a number of definitions, properties, examples, and illustrations to introduce the main concepts of this thesis. The remaining chapters 4, 5, and 6 contain our personal contributions. We provide now an overview of these contributions with two levels of presentation. We start with a short high-level picture of these contributions. Then, we specify for each chapter how they are organized and give some context.

1.3.1 *Contributions*

We first lay down a brief summary of our contributions.

Minimizing superquantile-based risk measures

- Managing risk has become a central issue in machine learning. Superquantile-based risk measures offer a convenient way to model risk-aversion by focusing on worst-case scenarios. In this thesis, we consider the problem of minimizing such risk measures. We advocate for the use of first-order methods with a special focus on the derivation of efficient oracles. More precisely, we analyze the possible nonsmoothness induced by superquantiles and provide a generic smoothing procedure. We also extend our smoothing approaches to the class of law-invariant comonotone risk measures. Finally, we release a companion software, `SPQR`, which implements in Python the algorithms described and allows practitioners to easily experiments with superquantile-based supervised learning.

Solving chance constrained problems

- In contrast with convex risk measures, chance constrained programs present specific challenges due to the non-convexity induced by the chance constraint. In this thesis, we propose an algorithm for the solving

of a general chance constrained problem. We first propose a reformulation of these problems as bi-level programs. We then propose an approach to solve these problems mixing a variety of techniques and concepts coming from different subdomains of optimization (penalization, weak sharpness, error bounds, DC programming, bundle algorithms, etc.). We finally release an open-source python toolbox implementing the approach, with a special emphasis on fast computational subroutines.

Handling heterogeneity in federated learning

- The emergence of privacy and fairness concerns in distributed optimization has spurred significant interest in the machine learning community. Federated learning is a nascent privacy-friendly paradigm for large-scale networks. In Federated learning, many devices (e.g. mobile phones) collaboratively train a model under the orchestration of a central server (e.g. a service provider), while keeping the data on device throughout the training and coordination processes. In this thesis, we propose a federated learning framework that operates on heterogeneous devices. The proposed approach hinges upon a superquantile-based learning objective to stress the impact of disadvantaged users in the training process. We present a stochastic training algorithm compatible with the privacy concerns of federated frameworks. We analyze its convergence and we support it with extensive numerical experiments.

We move now to a more in-depth overview of our contributions, that we break down by chapters.

CHAPTER 4 : MINIMIZING SUPERQUANTILE-BASED RISK MEASURES.

This chapter is mostly based on our papers [86, 88, 90].

In Chapter 4, we aim at providing efficient minimization procedures for a practical class of convex risk measures provided by the Kusuoka representation [84]. That is, we aim at solving problems of the form

$$\min_{w} \rho\left(f(w,\xi)\right) \tag{1.5}$$

where ρ a risk measure of the form

$$\rho(X) = \int_{p=0}^{1} \beta(p) S_{p'}(X) \mathrm{d}p',$$

with $\beta\colon [0,1] \to \mathbb{R}^+$ an arbitrary weight function and $S_{p'}$ the superquantile (1.4) at level p'.

Context. The superquantile can be traced back to the paper [10]. It stands out as one of prominent examples of risk measures, well established in economics and finance [11, 141]. Superquantiles have been extensively studied from a convex analysis perspective: we refer for instance to [141] for a variational formulation of the superquantile, to [11, 27] for its generalization to a larger class of risk measures, to [50] for a dual formulation (also later generalized in [145] or [139]) and [137] for additional convex properties: for a thorough discussion and many references, we refer to the seminal work [141], the classical textbook [148, Chap. 6], or the tutorial paper [136]. We will come back to these properties in Chapter 3. These nice theoretical properties have given interesting results in various applications, including recent ones in fair learning [174], adversarial classification [64] and reinforcement learning [155]. We will come back to some of these applications in Chapter 2. Regarding computational solving, the idea of using first-order methods to minimize superquantile-based

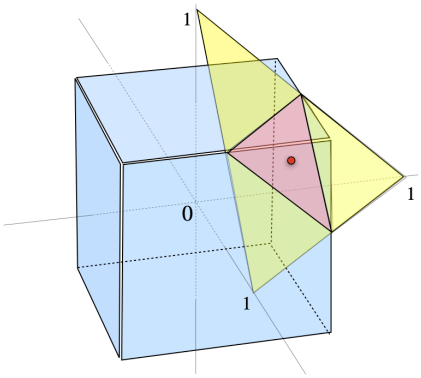


Figure 1.2: Illustration of a dual formulation of the superquantile as the support function of a particular ambiguity set (in red).

risk measures has yet been put aside until recently. Most historical applications considered linear programming approaches. We note among the few exceptions (i) the PhD thesis [113] using subgradient algorithms and (ii) the smoothing approaches laid down in [69, 106] (both applied to specific problems). More recently, gradient-based methods for the minimization of convex risk measures were developed for the large-scale setting [19, 34, 89, 94] and special efforts have been made to show their relevance in deep learning applications [13, 55, 150].

Contributions. This chapter proposes a general and practical framework for the minimization of superquantile-based risk measures.

Non-convex subgradient oracles

1. **We provide explicit and implementable expressions of (non-necessarily convex) subdifferential of superquantile-based cost functions.** Expressions of (convex) subdifferential of superquantile are well-known in general settings; see e.g., [145] for a thorough study. Here we study non-convex subdifferentials to treat cases when the random cost f in (1.5) is not convex. We come up, in the data-driven context, with concrete expressions. We give simple and direct proofs of these results as applications of basic definitions and properties of variational analysis [142].

Smoothing of the superquantile

2. **We provide a unifying analysis of smoothing procedures of nonsmooth superquantile-based functions.** We propose to use the infimal convolution smoothing of [117] and establish precise connections to other existing smoothing approaches. We provide optimal procedures for the computation of first-order oracles for the smoothed superquantile.

Smoothing of the Kusuoka representation

3. **We extend our smoothing procedures of superquantiles to law-invariant comonotone risk measures.** Our analysis leverages both dual properties of the superquantile together with the Kusuoka representation [84]. A special attention is paid to maintaining the optimal log-linear computational complexity for these first-order oracles.

Software package

4. **We release `SPQR`, a publicly-available and easy-to-use Python toolbox for superquantile-based learning.**This toolbox is built off the popular software library `scikit-learn`. We provide computational experiments illustrating (i) the interest of using quasi-Newton algorithms for minimizing superquantile-based objective and (ii) the robustness of superquantile-based models compared to the standard models obtained from empirical risk minimization.

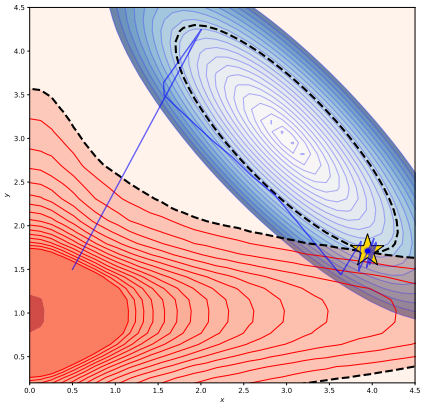


Figure 1.3: Trajectory of the iterates (in blue) of our bundle algorithm on a 2-dimensional chance constrained problem investigated in Chapter 5.

CHAPTER 5 : SOLVING CHANCE CONSTRAINED PROBLEMS.

This chapter is based on our paper [91]

In Chapter 5, we consider the solving of chance constraint problems in data-driven contexts. Specifically, we study problems of the form

$$\begin{cases} \min\limits_{x \in S} & f(x) \\ \text{s.t.} & \mathbb{P}[g(x,\xi) \leq 0] \geq p \end{cases} \tag{1.6}$$

and assume $f\colon \mathbb{R}^d \to \mathbb{R}$ and $S \subset \mathbb{R}^d$ to be convex, $g\colon \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}$ to be convex with respect to x and $\xi : \Omega \to \mathbb{R}^m$ to be observable through data sampling.

Context. Solving (non-convex) chance-constrained problems is notoriously difficult. Several computational methods have been proposed, regardless of any considerations of convexity and smoothness, and under various assumptions on uncertainty, see e.g. [158] for an overview. We note that when distributions are observed through data sampling, two main approaches have been considered. First, mixed-integer approaches [3] were considered to encode the probability constraints via binary activation variables; we refer to [102–104] for a deeper dive into this topic. Second, a considerable interest has been given to approximations of the probabilistic constraints , e.g. via convolutions [4] or by a difference of convex functions [36, 66].

In chapter 5, we propose a third approach based on a reformulation of chance constrained problems into bilevel programs. In recent years, bilevel programming has received considerable interests in the field of machine learning, where it is used for hyperparameter selection procedures [14]. The solving of general bilevel programs via exact penalization was historically proposed in [176] but still remains difficult due to the eventual non-convexity of the induced penalization. In our case, we point out the "Difference of Convex" structure of the obtained penalization and propose to use a bundle method to solve it. While bundle methods [6, 121] were initially considered for convex problems, their relevance on difference of convex objectives was recently showcased in [36, 166].

Contributions. In this chapter, we present a new reformulation of chance constrained programs as bilevel programs. We turn this reformulation into a single-level problem with a Difference of Convex (DC) objective via a semi-exact penalization. More specifically:

Bilevel reformulation

1. **We derive an exact reformulation of** (1.6) **as a bilevel problem** with upper and lower level problems, both convex. Our reformulation strongly relies on a variational formulation of superquantiles.

Double penalization

2. **We derive a double penalization method for the obtained bilevel problem, leading to single level Difference of Convex (DC)-program.** We first bring to light the weak-sharpness properties of the lower-level problem and exploit them to derive an exact penalization of the lower-level constraint. The penalized problem presents a nonsmooth DC objective for which we provide computable first-order oracles.

Solving by bundle methods

3. **We propose to combine a suitable bundle method for the penalized problem together with a dynamic update of the penalization parameters**. In addition we propose to smooth the concave component of the DC objective, using the same tools as in Chapter 4, to ensure better convergence properties of the method on our problem.

Software package

4. **We release `TACO`, an open-source python toolbox which implements our approach.** We numerically illustrate the performances of this toolbox on several toy problems.

CHAPTER 6 : HANDLING HETEROGENEITY IN FEDERATED LEARNING.

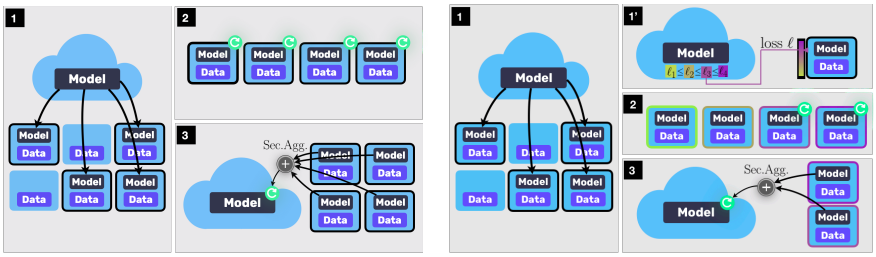
This chapter is based on our papers [89, 126]

The standard algorithm in federated learning is *FedAvg* [109]. It aims at fitting, *in average*, a model to the data distribution of the devices available for training. Formally, given a stochastic objective $F(w, \xi)$ and N training devices with respective data distributions q_i, *FedAvg* aims at minimizing the aggregated objective:

$$\min_{w \in \mathbb{R}^d} \mathbb{E}_{\xi \sim \bar{q}} \left[F(w, \xi) \right] + \frac{\lambda}{2} \|w\|^2 , \tag{1.7}$$

where $\bar{q} = \frac{1}{N} \sum_{i=1}^{N} q_i$ denotes the aggregated distribution of data-points over the whole set of training devices. While this approach works for users who conform to the aggregated distribution $\bar{q}$, it is liable to fail on non-conforming individuals, leading to poor user experience. In Chapter 6, we provide a new federated framework to address the statistical heterogeneity among the devices of the network.

Context. Addressing statistical heterogeneity in federated learning has led to two lines of work. The first develops algorithmic techniques to alleviate the effect of heterogeneity on convergence rates while still minimizing the classical expectation-based objective function (1.7). These techniques include the use of proximal terms [96], control variates [72] or augmenting the server updates [134, 170]; we refer to the recent survey [171] for details. The second line of work addressing heterogeneity involves designing new objective functions by modeling statistical heterogeneity. The *AFL* framework minimizes the worst-case error across all training devices [114]. *q-FFL* framework [97] draws from fair resource allocation literature and proposes to minimize the L^p norm of the per-device losses. A federated optimization algorithm for *AFL* was proposed and its convergence was analyzed in [37]. Finally, a classical risk measure, namely the entropic risk measure, was considered in [98]. We note that no convergence guarantees are currently known for the stochastic optimization algorithms developed for these non-usual frameworks.



(a) *FedAvg*. **(b)** Δ-*FL*.

Figure 1.4: Comparative diagram between the baseline *FedAvg* and our algorithm Δ-*FL* which handles heterogenous devices. Steps 1, 2 and 3 are identical and hold as fundamental components of any practical federated framework: broadcast of server models to a random subselection of devices - running of local stochastic gradient updates - secure aggregation of the selected models. Only step 1′ is specific to Δ-*FL* and can be interpreted as an additional filtering step among selected device to choose which device will run the local updates. Mathematically, this filtering steps interpret as a composition with the superquantile - see more in Chapter 6.

Contributions. In this chapter, we design a practical and theory-backed framework for the handling of non-conforming users in statistically heterogeneous federated environments.

1. **We introduce the Δ-*FL* framework which seeks to provide a minimal level of predictive performance on nonconforming devices.** The framework relies on a superquantile-based objective to minimize the tail statistics of the prediction errors on the client data distributions. Specifically, given a scalar parameter $p \in [0, 1]$, we propose to replace the standard objective in (1.7) by

Δ-FL framework

$$\min_{w \in \mathbb{R}^d} S_{p\, \xi \sim \bar{q}} \left(F(w; \xi) \right) + \frac{\lambda}{2} \|w\|^2 \,. \tag{1.8}$$

2. **We propose a new federated algorithm to solve this problem.** Our algorithm, illustrated in Figure 1.4 builds off the baseline *FedAvg* with an additional filtering step over training devices with poorest performances. We provide a convergence analysis in the strongly convex case.

Theory-backed algorithm

3. **We present extensive numerical results to support our framework.** We use linear models and neural networks, on tasks including image classification and sentiment analysis based on public datasets. The simulations demonstrate superior performances of Δ-*FL* over state-of-the-art baselines on the upper quantiles of the error on test devices, with particular improvements on data-poor devices, while being competitive on the mean error.

Extensive numerical experiments

1.3.2 *Further works non included in this thesis*

During my time as a PhD student, I also had the opportunity to work on two other projects that will not be developed in this thesis. I provide below a brief summary of each. Further information can be found in my webpage

This project led to the paper [92]

EVENTUAL CONVEXITY OF CHANCE CONSTRAINED PROGRAMS. Probabilistic constraints result from taking the probability measure of a given set of random inequalities depending on the decision vector. Even if the original set of inequalities is convex, this favourable property is not immediately transferred to the probabilistically constrained feasible set and may in particular depend on the chosen safety level. In this project, we provided results guaranteeing the convexity of feasible sets to probabilistic constraints when the safety level is greater than a computable threshold. This is often referred to as a situation of "eventual convexity". The key idea in our approach is to reveal the level of underlying convexity in the nominal problem data (e.g., concavity of the probability function) by auxiliary transforming functions. We provided several examples illustrating our theoretical developments.

This project led to the paper [7]

RANDOMIZED AND ASYNCHRONOUS VARIANTS OF PROGRESSIVE HEDGING. Progressive Hedging is a popular decomposition algorithm for solving multi-stage stochastic optimization problems. A computational bottleneck of this algorithm is that all scenario subproblems have to be solved at each iteration. In this project, we introduced randomized versions of the algorithm able to produce new iterates as soon as a single scenario subproblem is solved. Building on its relation with monotone operators, we leveraged recent results on randomized fixed point methods to derive and analyze the proposed methods. Finally, we released the corresponding code as an easy-to-use Julia toolbox and report computational experiments showing the practical interest of randomized algorithms, notably in a parallel context.

1.3.3 *List of publications*

We end up this introduction with a list of our published/submitted papers during our time as PhD candidate.

1. Y. Laguel, J. Malick, and Z. Harchaoui. First-order optimization for superquantile-based learning. *Proceedings of the IEEE International Workshop on Machine learning for signal processing (MLSP)*, 2020.
2. G. Bareilles, Y. Laguel, D. Grishchenko, F. Iutzeler, and J. Malick. Randomized progressive hedging methods for multi-stage stochastic programming. *Annals of Operations Research*, 2020.
3. Y. Laguel*, K. Pillutla*, J. Malick, and Z. Harchaoui.[1] A Superquantile Approach for Federated Learning with Heterogeneous Devices. *55th Annual Conference on Information Sciences and Systems*, 2021.
4. Y. Laguel, W. Van Ackooij, J. Malick, and G. M. Ramalho. On the convexity of level-sets of probability functions. *Journal of Convex Analysis*, 2021.
5. Y. Laguel, K. Pillutla, J. Malick, and Z. Harchaoui. Superquantiles at Work : Machine Learning Applications and Efficient (Sub)gradient Computation. *Set-Valued and Variational Analysis*, 2021.
6. Y. Laguel, J. Malick, and W. Van Ackooij. Chance constrained problems: a bilevel convex optimization perspective. Submitted to *Computational Optimization and Applications*, 2021.
7. Y. Laguel*, K. Pillutla*, J. Malick, and Z. Harchaoui. Federated Learning with Heterogeneous Devices:A Superquantile Optimization Approach. Submitted to the *Journal of Selected Topics in Signal Processing*, 2021.

1 * denotes equal contributions.

2

SELECTED APPLICATIONS

In this section, we review a number of applications where probabilistic and superquantile-based decision problems naturally appears. We first start with traditional applications in operation research (portfolio optimization and energy management). We then move to applications in machine learning. For each application, we keep a high-level point of view: we insist only on the main elements of the modeling process and we omit the technical details. This chapter has only an illustrative purpose. It can be skipped without harming the understanding of our contributions.

2.1 SELECTED APPLICATIONS FROM OPERATIONS RESEARCH

In this section, we review two applications of optimization under uncertainty in operations research. We start with a classical portfolio optimization problem. We introduce then the problem of reservoir-planning in hydro-management.

2.1.1 *Portfolio optimization*

Portfolio management

Since the pioneering work of Markowitz [108], portfolio selection has much evolved, stemming from both the development of risk models and the derivation of efficient computational methods. The objective of a typical asset allocation problem is to maximize the expected return of a portfolio optimization problem while hedging against potential losses. Formally, given a random vector of returns $R = (r_1, \dots, r_n)$, one wishes to find an allocation solution of

$$\begin{cases} \max\limits_{w} & \mathbb{E}[w^\top R] \\ \text{s.t.} & \mathfrak{R}\left(-w^\top R\right) \leq \varepsilon \end{cases}$$

where $\mathfrak{R} \colon \mathcal{L}^1(\Omega) \to \mathbb{R}$ denotes a risk measure aimed at upper-bounding the level risk exposition of the investment $w^\top R$.

Value at Risk

Financial regulations on risk have often been set in terms of quantiles of a distribution of losses or rewards. For instance, given a random variable U that quantifies the losses of an investment, and a safety threshold $p \in (0, 1)$, the Value at Risk is defined as the p-quantile $Q_p(U)$ of U. It is commonly considered for high levels of probability: $p \in \{0.95, 0.98, 0.99\}$. Asset allocation problems involving the Value at Risk - see e.g. [101] - are commonly framed as

$$\begin{cases} \max\limits_{w} & \mathbb{E}_S[w^\top R] \\ \text{s.t.} & Q_p(-w^\top R) \leq \delta \end{cases} \tag{2.1}$$

where δ denotes an arbitrary threshold to be set by the risk modeler.

Conditional Value at Risk

While very popular across the banking institutions, such measures of risk remain hard to optimize due to their non-convex properties. Over the past two decades, the *superquantile*, introduced in Section 1.1, has become a powerful alternative, thanks to its nice convex and regular properties. For a distribution of losses U and a probability threshold p close to 1, the associated superquantile $S_p(U)$ returns the average of the quantiles that are greater than the p-quantile. Hence, for similar probability levels, the superquantile offers a more conservative estimation of risk than the Value at Risk. One may then reconsider problem (2.1) with the superquantile in place of the quantile or even a penalization of this problem yielding for some $\mu > 0$

$$\max_{w} \mathbb{E}_S[w^\top R] - \mu S_p(-w^\top R)$$

Owing to the seminal work of Rockafellar and Uryasev [141], the practical solving of such problems may be achieved by various methods including linear programming and subgradient-based methods. In this thesis, special efforts have been made for the use of the superquantiles in large-scale contexts.

2.1.2 *Reservoir planning*

Hydro valley management

Dealing with uncertainty is a recurrent challenge in energy planning. For instance, when considering the management of a hydro valley, one has to deal with meteorological constraints of the surrounding environment. Such constraints can be prone to unpredictable variations over a short time scale. The hydro valley is composed of a network of hydro powers plants with their associated turbines. Schematically, the aim of a reservoir planning problem is to optimize the usage of the turbines to meet the power demand while ensuring the feasibility of the operations, under the uncertainty of the inflows. Such problems can be modeled as:

$$\begin{aligned} \min_{x \geq 0, x \in \mathbb{R}^n} \quad & c^\top x \\ \text{s.t} \quad & Ax \leq b \\ & u + Cx \leq \xi \leq v + Cx. \end{aligned} \tag{2.2}$$

In the above problem, the variable x denotes the controls on the set of turbines at stake, for instance in terms of cubic meters per hours. The objective $\langle c, . \rangle$ encodes the amount of water processed together with the quantity of electricity power generated. The first polyhedral constraint encodes the modelling of the flow within the network, the bounds on the turbining and the global use of water. The second constraint involves the uncertain amount of water stemming from random inflows modeled by the random vector $\xi : \Omega \to \mathbb{R}^m (m \in \mathbb{N})$. This constraint is present to ensure that no overrun of the maximal reservoir capacity is observed. For more details on the modelling of reservoir planning problems, we orient the reader to the paper [165] and the thesis [164].

Modelling via chance constraints

Satisfaction of the stochastic constraint (2.2) can either be required in expectation or with high probabilities. For the latter option, one thus needs to fix a safety probability threshold p close to 1 and consider the probabilistic constraint:

$$\mathbb{P}\left[u + Cx \leq \xi \leq v + Cx\right] \geq p.$$

Modelling of reservoir planning problems via probabilistic constraints has received considerable interest, e.g. [46, 100, 178] with use of various methods to

solve it including penalty methods [130], supporting hyperplane methods [129] or bundle methods [121]. In the chapter 5 of this thesis, we will consider the solving of a general class of chance constrained problems.

2.2 SELECTED APPLICATIONS FROM MACHINE LEARNING

We move now to a selection of more recent topics in machine learning where superquantiles may be useful. The application from Section 2.2.2, in particular, has motivated the contributions of the whole Chapter 6.

2.2.1 *Safe machine learning*

ERM limitations

Classical supervised learning via empirical risk minimization hinges upon the assumption that the testing distribution coincides with the training distribution. This assumption can be challenged in domain applications of machine learning such as visual systems or dialog systems [133]. Learning machines may indeed operate at prediction time with testing data whose distribution departs from the one of the training data. Thus, in the face of prevalence of worst-case scenarios or unexpected distributions at prediction time, ensuring robust behavior becomes a relevant strategy. This highlights the importance of reconsidering the learning objective used to train learning machines.

Data-driven setting

Formally, consider a data-driven setting where a random function $f(w, \xi)$ can be estimated through a sample of data-points $\xi_1, \ldots, \xi_n, (n \in \mathbb{N})$. This is a standard situation in classical supervised learning where, in the training phase, we have access to n data-points: each data-point is a pair (a, b), where $a \in A$ is a feature vector and $b \in B$ is its corresponding target. For instance, for a binary image classification task, b is a boolean encoding the membership of the image a to one of the two classes. From this training data, the aim is to learn a parameter $w \in W \subset \mathbb{R}^d$ (as "weights") of a given prediction function $\varphi(w, \cdot)$ that produces, for an input $a \in A$, a prediction $z = \varphi(w, a) \in Z$ of the associated target $b \in B$. Typical examples of prediction functions include simple linear models $\varphi(w, a) = w^\top a$, polynomial models (as in Example 2.1 below), or artificial neural networks

$$\varphi(w, a) = w_s^\top \sigma(\cdots \sigma(w_1^\top a)) \,, \tag{2.3}$$

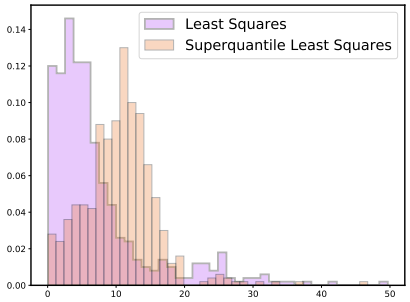
which are successive compositions of linear models w_j and non-linear activations σ. The prediction error is then measured by a loss function $\ell : B \times Z \to \mathbb{R}$. Typical examples of loss functions include the least-squares loss ($B = \mathbb{R}, Z = \mathbb{R}$) or the logistic loss ($B = \{-1, 1\}, Z = \mathbb{R}$), defined respectively as

$$\ell(b, z) = \frac{1}{2}(b - z)^2 \,, \qquad \text{and,} \qquad \ell(b, z) = \log(1 + \exp(-b\, z)) \,. \tag{2.4}$$

In the risk-neutral setting, assuming[1] that the training data are generated from a given distribution P over $A \times B$, the "best" model parameter w solves the optimization problem

$$\min_{w \in W} \left[R(w) = \mathbb{E}_{(a,b)\sim P} \left[\ell(b, \varphi(w, a)) \right] \right] \,. \tag{2.5}$$

1 When the assumption of the existence of an underlying distribution P is not realistic, the usual approach is to still use the empirical risk minimization principal based on the given training dataset $\mathbb{P}_n = \{(a_i, b_i)\}_{1 \le i \le n}$.



Metric	model (2.9)	model (2.10)
Mean	**7.23**	10.62
80^{th} perc.	**9.73**	13.94
90^{th} perc.	17.29	**15.93**
95^{th} perc.	24.00	**17.83**

Figure 2.1: Superquantile regression improves over worst-case datapoints. **Left figure**: histograms of residuals $r_i = |y_i - (w_2 x_i^2 + w_1 x_i + w_0)|$ for model (2.9) (in violet) and model (2.10) (in orange). **Right table**: x^{th} perc. stands for x-th percentile of final distribution of the residuals r_i.

This framework is currently challenged by important domain applications [e.g., 70, 133], in which several of the standard assumptions turn out to be limiting. Indeed classical supervised learning assumes that, at training time, the sampled examples $(a_1, b_1), \ldots, (a_n, b_n)$ are drawn i.i.d. from a given distribution P, and that, at testing time, we face a new example a', also drawn from the same distribution P. However, recent failures of learning systems when operating in unknown environments [77, 111] emphasizes the importance of taking into account that we may not face the same distribution at test/prediction time.

Safe machine learning

A simple way to enforce robustness is then to replace the expectation in (2.5) by a convex risk measure ρ. The resulting objective is:

$$\min_{w \in W} \left[\mathfrak{R}(w) = \rho_{(a,b)\sim P}\left[\ell(b, \varphi(w, a)) \right] \right] . \tag{2.6}$$

In this thesis, we focus on a particular class of risk measures that we will introduce in Section 3.2. A special attention will be given to superquantiles, defined in (1.4), for their fundamental role among such risk measures. We provide a short illustrative example below.

Superquantile least-square regression

Example 2.1. We illustrate the interest of superquantile learning in presence of heterogeneous data, on an elementary regression task. Consider a dataset gathering two different subgroups: for two unknown models $\bar{w}^1$ and $\bar{w}^2$ in R^3 we assume 80% of the points are generated according to

$$y_i = \bar{w}_0^1 + \bar{w}_1^1 \, x_i + \bar{w}_2^1 \, x_i^2 + \varepsilon_i \qquad \text{where } \varepsilon_i \sim \mathcal{N}(0, \sigma^2) , \tag{2.7}$$

and the remaining 20% are generated according to

$$y_i = \bar{w}_0^2 + \bar{w}_1^2 \, x_i + \bar{w}_2^2 \, x_i^2 + \varepsilon_j \qquad \text{where } \varepsilon_j \sim \mathcal{N}(0, \sigma^2) . \tag{2.8}$$

Then we can compare the usual approach using ordinary least-squares

$$\min_{w \in \mathbb{R}^3} \mathbb{E}_{(x,y)\sim \mathbb{P}_n} \left[(y - (w_2 w^2 + w_1 x + w_0))^2 \right] , \tag{2.9}$$

with its superquantile counterpart of the form:

$$\min_{w \in \mathbb{R}^3} [S_p]_{(x,y)\sim \mathbb{P}_n} \left[(y - (w_2 w^2 + w_1 x + w_0))^2 \right] \tag{2.10}$$

using $\rho = S_p$ for $p = 0.9$.

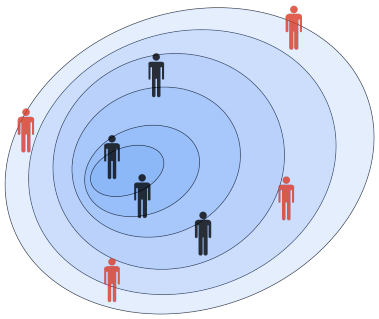


Figure 2.2: Statistical Heterogeneity is a key feature of federated learning where clients with heterogeneous distributions collaborate to learn a single model.

We report on Figure 2.1 the distribution of residuals $r_i = |y_i - (w_2 x_i^2 + w_1 x_i + w_0)|$ for both models. The superquantile model (2.10) shows an improvement of 90/95th quantiles of the distribution of residuals, which appears on histograms as a shift of the upper tail to the left. This comes at the price of a degraded performances on average, which appears on the figure as the shift of the peak of residuals to the right. □

2.2.2 *Federated optimization*

Federated learning is an emerging framework in machine learning where many clients collaboratively train a common model under the orchestration of a central server, while keeping their data decentralized [52, 109, 175]. In a standard federated learning setting, we wish to optimize the performances of a single model $w \in \mathbb{R}^d$ deployed over a large population of clients. We assume that each client has data, drawn i.i.d. from a distribution q. The loss incurred on this device is $F(w;q) := \mathbb{E}_{\xi\sim q}[f(w;\xi)]$, where $f(w;\xi)$ is the chosen loss function (such as the logistic loss) on input-output pair ξ under the model w. The expectation above is assumed to be well-defined and finite. For a given distribution q, smaller values of $F(\cdot;q)$ denote a better fit of the model to the data.

Vanilla Federated learning

Given N client devices available for training, we denote the loss on device k by $F_k(w) := F(w;q_k)$. Vanilla Federated learning aims at minimizing a weighted average of the losses attached to training devices,

$$\min_{w\in\mathbb{R}^d} \sum_{i=1}^{N} \alpha_i F_i(w) \tag{2.11}$$

where the weights α_i are typically set to be proportional to the size of the local dataset in each device. While the minimization of such sums are ubiquitous in machine learning, federated learning comes with a number of specific challenges that we outline below. For a deeper dive into the field, we point the reader to the recent reviews [70, 95, 171].

Federated learning challenges

Privacy constraints. The protection of user's data is a critical issue in federated learning. This is achieved by keeping the processing of the data local to the de-

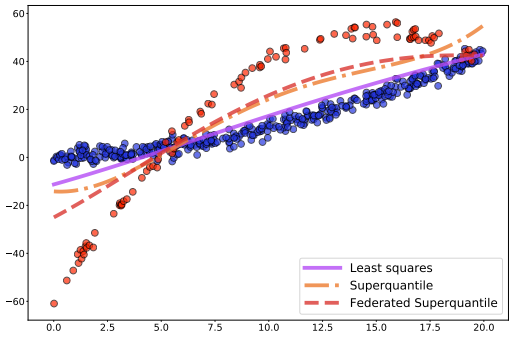


Figure 2.3: Comparison of the three regressions for the toy federated learning setting of Example 2.2. We want commensurate performances among users, which means, graphically, a curve at the same distance from the data-points of the conforming users (in blue) and the non-conforming user (red).

vices while the server coordinates with the devices for a secure aggregation(see e.g. [20]) of model updates.

Communication constraints. In federated learning, the central server is responsible for the orchestration of a huge network with possibly unreliable transmissions. Thus, communications rapidly become a bottleneck calling for the design of communication-efficient algorithms. Efficiency can either be performed through the development of compression algorithms [29, 125, 168] or the reduction of communications via the execution of several *local steps* within devices [72, 75, 153]. In addition, only a small random proportion of the training clients is likely to be available at each stage of the training process.

Statistical heterogeneity. Statistical heterogeneity is also key feature of federated learning: client data distributions are *not* identical. Each user has unique characteristics which are reflected in the data they generate. These characteristics are influenced by personal, cultural, and geographical factors. For instance, the varied use of language contributes to data heterogeneity in a next word prediction task. In this context, the satisfaction of users that do not conform to the average training data distribution and underlying fairness concerns have attracted considerable interest in recent years.

Such limitations have motivated the development of new frameworks to serve more fairly the entire population of users [89, 97, 114]. All aforementioned references share the common feature of replacing the weighted average in the objective (2.11) by a *risk-aware function* $\rho : \mathbb{R}^N \to \mathbb{R}$ meant to better capture non-conforming users. The objective considered becomes:

$$\min_{w \in \mathbb{R}^d} \rho\left[F_{\boldsymbol{\mathcal{K}}}(w)\right] \tag{2.12}$$

where $\boldsymbol{\mathcal{K}} \colon [\![1, N]\!] \to \mathbb{R}$ is a random variable satisfying $\mathbb{P}[\boldsymbol{\mathcal{K}} = k] \propto \alpha_k$.

Federated regression

Example 2.2. Consider a specific instance of Example 2.1 in a federated setting. We consider that 80% of the data corresponds to four devices having the same data distribution following (2.7), while the remaining 20% corresponds to a fifth device having its own distribution following (2.8). In this example, for $k \in [\![1, 5]\!]$, the loss F_k corresponds to the regression objective on the dataset owned by the k^{th} user. Figure 2.3 shows this bivalent dataset: blue points correspond to the data of the four first devices, and red points correspond to the last device. We want a regression that captures worst-cases for both behaviours.

We plot on Figure 2.3 the regression models given by the classical least-squares regression (2.9), the superquantile least-squares regression (2.10), and a *federated* variant of the superquantile least-squares that operates at a user level, i.e. by using the superquantile as ρ in (2.12) . We can make three observations. First the standard model (2.9) (in purple) tends to follow the trend imposed by the first four devices. Second, the superquantile model (2.10) (in orange) has better regression on worst-case data, irrespective of the group of the data point. Finally the federated superquantile model (2.12) finds, in contrast, a compromise between the two trends. Thus, federated superquantile regression better captures the (red) data points of the non-conforming user. □

2.2.3 *Fairness in AI*

Algorithmic fairness has received much attention over the past decade due to the fairness issues raised in a number of machine learning applications (e.g. the pre-screening of job applications). The objective of a fair algorithm is to reduce the bias that standard algorithms are prone to stress on one or several subgroups of users while still maintaining reasonable estimation performances.

As recalled in Section 2.2.1, in classical supervised machine learning, we consider problems of the form

$$\min_{w} \mathbb{E}_{a,b}[\ell(b, \varphi(w, a)]$$

where a, b are drawn from a distribution over the input-output space A, B, $\varphi(w, \cdot)\colon A \to Z$ is a predictor function parametrized by the weights vector $w \in \mathbb{R}^d$ and $\ell\colon B \times Z \to \mathbb{R}$ is a given error function. In a fair machine learning context, one has to deal in addition with the presence of a sensitive feature $s \in S$ (e.g. gender or race) calling for the satisfaction of a *fairness constraint*. Formally, one first needs to design map $\mathfrak{F}\colon B \times Z \times S \to \mathbb{R}$ that permits to gauge the level of unfairness of a given predictor $\varphi(w, \cdot)$ over the extended dataset (a, b, s). A fair formulation of the above problem is then to solve for a given $\varepsilon > 0$:

Fair machine learning

$$\begin{cases} \min\limits_{w} & \mathbb{E}_{a,b}[\ell(b, \varphi(w, a)] \\ \text{s.t.} & \mathfrak{F}(b, \varphi(w, a), s) \leq \varepsilon \end{cases} \tag{2.13}$$

The fairness constraint may be designed in various ways that are not systematically coherent with each other - see e.g. the discussion [76]. While many works have investigated the practical handling of fairness constraints for specific learning tasks [1] such as classification or regression [2, 30], we limit this brief introduction to a general supervised learning context. We adopt below the perspective laid down in [174] to introduce several popular examples of fairness constraints.

Perfect and approximate Fairness. Situations of perfect fairness often allude to a condition of statistical independence between predictions and sensitive features. For instance, given weights $w \in \mathbb{R}^d$, *demographic parity* [45] denotes complete statistical independence between the predictor $\varphi(w, \cdot)$ and the sensitive attributes s,

Perfect fairness

$$\varphi(w, a) \perp\!\!\!\perp s$$

where $\perp\!\!\!\perp$ denotes the statistical independance. Alternatively, the concept of *equality of opportunity* [58] allows for the prior knowledge of the outputs

$$\varphi(w, a) \perp\!\!\!\perp s \mid b.$$

Model	$L_1(w)$ (blue subgr.)	$L_2(w)$ (red subgr.)
least-square (2.9)	4.59	17.76
superquantile (2.10)	9.88	13.62
federated superquantile (2.12)	10.87	11.46

Table 2.1: Average performances of each model over both subgroups.

Approximate fairness

Such assumptions, considered as ideal, often lead to singularly difficult optimization problems [1]. Instead one may consider approximate variants by bounding for instance the level of dependence between s and $\varphi(w, \cdot)$. In [71], the authors consider the Kullback-Leibler divergence to measure such independence:

$$\mathfrak{F}(b, \varphi(w, a), s) = KL\left(\mathbb{P}_{\varphi(w,a),s} \,|\, \mathbb{P}_{\varphi(w,a)} \times \mathbb{P}_s\right).$$

Solving such problems remains difficult in general due to the non-convexity induced by the fairness constraint.

Subgroup risk

Fairness via Subgroup Losses. The notion of subgroup risk, introduced in [173], consists in measuring the fairness of a given predictor with respect to the sequence of losses evaluated among the subgroups defined by the sensitive feature s. Specifically, given a sensitive attribute $\bar{s} \in S$ and a predictor $\varphi(w, \cdot)$, the subgroup risk of $\varphi(w, \cdot)$ is defined by

$$L_{\bar{s}}(w) := \mathbb{E}_{a,b|s=\bar{s}}\, \ell(b, \varphi(w, a)).$$

The fairness constraint is then defined in terms of deviations of the conditional expectation $L_s(w)$, turning (2.13) into

$$\begin{cases} \min\limits_{w} & \mathbb{E}_s[L_s(w)] \\ \text{s.t.} & \mathfrak{D}(L_s(w)) \leq \varepsilon \end{cases}$$

where $\mathfrak{D}\colon L^1(\Omega) \to \mathbb{R}$ denotes an arbitrary *measure of deviation*. Penalization of the above deviation constraint amounts to the minimization of a single objective that holds as a composition of a *risk measure* $\rho\colon L^1(\Omega) \to \mathbb{R}$ with the subgroup risks involved

$$\min_{w \in \mathbb{R}^d} \rho\left(L_s(w)\right).$$

In this context, [174] advocate for the use of the superquantile and show empirically how it may enforce fairness on illustrative examples. We will show in this thesis that such approach may be extended to large-scale situations thanks to the nice properties of the superquantile.

Fair regression

Example 2.3. Let us come back to the toy Example 2.2. We look at it with the perspective of fairness between the predominant group (the four blue users) and the minority group (the fifth "red" user). Table 2.1 compares (i) the average performance over the predominant group and (ii) the average performance on the minority group. We observe that the difference between these performance is minimal for the user-level superquantile model provided by (2.12), achieving better approximate group fairness. □

3

TECHNICAL PRELIMINARIES

In this chapter, we formally introduce some of the main concepts involved in this thesis. We first provide a general framework for the measurement of random losses with a particular focus on the role that convexity plays to enforce distributional robustness. We then present the class of law-invariant comonotone risk measures and lay out their nice geometric and functional properties. We make a special emphasis on the superquantile which is at the core of this thesis, especially in Chapters 4 and 6. We finally move to chance constrained problems and present some of their specific challenges that will be addressed in Chapter 5. Experienced readers may skip this chapter: next chapters provide precise pointers to the results below if needed.

3.1 RISK-AVERSE OPTIMIZATION THROUGH THE LENS OF DUALITY THEORY

We present in this section a general framework to model risk aversion in a stochastic environment. We draw on the setup laid down in [145] and the general topological and convexity results from the textbooks [144, 169, 177]. To lighten the presentation, we will give precise pointers to references only for the fundamental results. We will implicitly refer to these general textbooks for intermediate or secondary results and developments.

3.1.1 *Duality in probability spaces*

We start this section with brief reminders on duality in probability spaces through a rather abstract setting that will be specialized later in this chapter.

Topological dual of a set of random variables

Pairing random variables and measures. Let $(\Omega, \mathcal{F})$ be a fixed measurable space and consider a linear space $\mathcal{X}$ of random variables from Ω to $\mathbb{R}$. We pair $\mathcal{X}$ with an arbitrary linear space $\mathcal{Y}$ of finite signed measures satisfying:

$$\int_\Omega |X| \, \mathrm{d}|\mu| < \infty \quad \forall X \in \mathcal{X}, \ \forall \mu \in \mathcal{Y}$$

where $|\mu| := \mu_+ + \mu_-$ and (μ_+, μ_-) denotes the Jordan decomposition of μ.

Furthermore, we assume that $\mathcal{X}$ and $\mathcal{Y}$ are locally convex topological vector spaces such that the scalar product is *compatible* with the respective topologies on $\mathcal{X}$ and $\mathcal{Y}$. That is, $\mathcal{Y}$ is the topological dual of $\mathcal{X}$ and no linear functional of the form $\langle ., \mu \rangle$ nor $\langle X, . \rangle$ is identically null for any $X \in \mathcal{X}$ and $\mu \in \mathcal{Y}$.

Example 3.1 (Basic examples of pairings)**.** A standard setting is to have, for any σ-finite measure $\boldsymbol{\mu}$ on $(\Omega, \mathcal{F})$ and $p \in [1, \infty)$, $\mathcal{X}$ as the quotient space $\boldsymbol{L}_p(\Omega, \mathcal{F}, \boldsymbol{\mu})$. A natural choice for $\mathcal{Y}$ is then the set of absolutely continuous measures μ with respect to $\boldsymbol{\mu}$ admitting a Radon-Nikodym derivative in $\boldsymbol{L}_q$, where $\frac{1}{p} + \frac{1}{q} = 1$. □

Fenchel conjugate and bi-conjugate

The Fenchel conjugate. Given such pairing, we may define for any proper

functional $\rho\colon \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$ the *Fenchel conjugate* of ρ, denoted $\rho^*\colon \mathcal{Y} \to \mathbb{R} \cup \{+\infty\}$ as:

$$\rho^*(\mu) = \sup_{X \in \mathcal{X}} \langle X, \mu \rangle - \rho(X)$$

and the *Fenchel biconjugate* $\rho^{**}\colon \mathcal{X} \to \mathbb{R} \cup \{\infty\}$ of ρ as

$$\rho^{**}(X) = \sup_{\mu \in \mathcal{Y}} \langle X, \mu \rangle - \rho^*(\mu).$$

As the maximum of linear functions, the fenchel biconjugate is a convex function with respect to X. Note that we always have $\rho^{**}(X) = (\rho^*)^*(X)$ for all $X \in \mathcal{X}$ but the equality $\rho^{**} = (\rho^*)^*$ does not necessarily hold, since $\mathcal{X}$ is not necessarily equal to its bidual. It is however valid whenever $\mathcal{X}$ is a reflexive Banach space (e.g. $\mathcal{X} = \boldsymbol{L}_p$ for $p > 1$).

Measuring random losses

From random variables to random losses. In many applications, the random variable X is of the form $f(w, \xi)$ and characterizes the cost of a given model $w \in \mathbb{R}^d$ while exposed to a random variable $\xi\colon \Omega \to \mathbb{R}^m$. Thus, a straightforward way to measure the loss incurred by w is to fix a measure $\mu \in \mathcal{Y}$ and consider:

$$F(w) := \rho(f(w, \xi)) = \int_\Omega f(w, \xi(\omega)) \, \mathrm{d}\, \mu(\omega). \tag{3.1}$$

Under suitable conditions, the expected value function $F : \mathbb{R}^d \to \mathbb{R}$ may inherit the properties satisfied by the integrand $w \mapsto f(w, \xi)$. For instance, if (i) f is lower semi-continuous with respect to w and (ii) for w' locally around w, $f(w', \cdot)$ can be uniformly dominated by an integrable random variable, then F is known to be lower semi-continuous as well (see for instance proposition 14 from [144, Chapter 1]). If one assumes in addition f to be convex with respect to w, then one may subdifferentiate F by interchanging the integral and the subdifferential:

$$\partial F(w) = \int_\Omega \partial f(w, \omega) \, \mathrm{d}\, \mu(\omega) := \left\{ \int_\Omega g \, \mathrm{d}\, \mu, s.t. \ \begin{smallmatrix} g:\Omega \to \mathbb{R}^d \text{ measurable} \\ g(\omega) \in \partial f(w, \omega), \ \forall \omega \in \Omega \end{smallmatrix} \right\}. \tag{3.2}$$

According to the terminology of [142], such functions $w \mapsto f(w, \xi)$ are called *normal convex integrands*. For a proof of such standard result, see for instance the chapter 2 of [144].

3.1.2 *Convex risk measures and Fenchel-Moreau's theorem*

Convex risk measures

Relying only on a single measure μ to evaluate the cost of a given model w via (3.1) may be insufficient to hedge against risky events. Instead, we are looking for some *risk functional* $\rho\colon \mathcal{X} \to \mathbb{R}$ that can stress the impact of worst-case possible outcomes *under various distributional shifts*. With this in mind, one may note that convex risk measures, i.e, functionals satisfying

$$\rho(\lambda X + (1-\lambda)Y) \leq \lambda \, \rho(X) + (1-\lambda) \, \rho(Y), \quad \forall X, Y \in \mathcal{X}, \forall \lambda \in [0, 1],$$

offer a natural interpretation of robustness. This a consequence of the Fenchel-Moreau theorem:

The Fenchel-Moreau theorem

Proposition 3.1. *Let $\mathcal{X}$ be a locally compact Haussdorf space, paired with its topological*

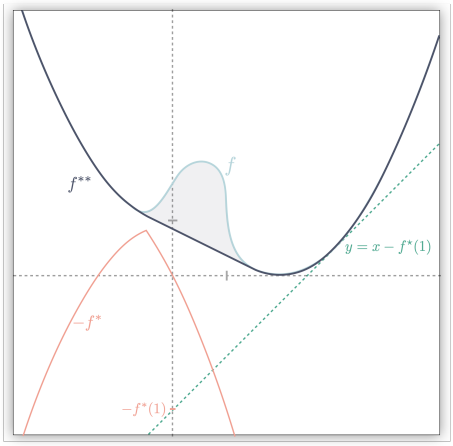


Figure 3.1: The Fenchel conjugate and the bi-conjugate of a real function f. The Fenchel conjugate is always convex. The Fenchel biconjugate f^{**} satisfies $\mathrm{epi} f^{**} = \overline{\mathrm{conv}}\ \mathrm{epi} f$.

dual $\mathcal{X}^*$, and $\rho\colon \mathcal{X} \to \mathbb{R}$ a proper, convex, lower-semi-continuous functional. Then, for all $X \in \mathcal{X}$, we have:

$$\rho(X) = \rho^{**}(X) = \sup_{\mu \in \mathcal{X}^*} \langle X, \mu \rangle - \rho^*(\mu). \tag{3.3}$$

As a consequence, proper l.s.c. convex risk measures are pointwise solutions of robust optimization problems over the space of measures $\mathcal{X}^*$.

A proof of this standard result can be found in [177, Theorem 2.3.3]. This duality result has been the cornerstone of many works seeking to model risk-aversion (see in particular [145]): the correspondence between ρ and ρ^* enables to characterize robustness via desirable properties on ρ^*. In this thesis, we will mostly focus on the cases where ρ^* is the indicator of a convex set of measures.

3.1.3 *Supporting distributionally robust optimization*

Support functions on $\mathcal{X}$

When the Fenchel conjugate ρ^* of ρ is the indicator of a closed convex set $\mathcal{A} \subset \mathcal{Y}$, ρ is a support function and satisfies two additional properties:

$$\text{(Sub-additivity)} \quad \rho(X + X') \leq \rho(X) + \rho(X'), \quad \text{for all } X, X' \in \mathcal{X} \tag{H1}$$

$$\text{(Positive homogeneity)} \quad \rho(\lambda X) = \lambda\, \rho(X), \quad \text{for all } X \in \mathcal{X}, \lambda \geq 0. \tag{H2}$$

Additional desirable properties may be set on ρ to model risk aversion. Among such, the *monotonicity* property defined as:

$$\text{(Monotonicity)} \quad \rho(X) \leq \rho(X'), \quad \text{for all } X \leq X' \ (a.s.). \tag{H3}$$

ensures that the risk decreases when $X := f(w, \xi)$ gets decreased for almost all values of ξ. It is also convenient whenever $w \mapsto f(w, z)$ is convex for all fixed $z \in \mathbb{R}^m$ as the composition $w \mapsto \rho(f(w, \xi)$ becomes also convex. From a dual perspective, this translates into $\mathcal{A} \subset \{\mu, \mu \geq 0\}$. We will provide an elementary proof of this statement in Proposition 3.4 of this chapter.

One last property we also wish to mention is the *translation invariance*, set as:

$$\text{(Translation Invariance)} \quad \rho(X + c) = \rho(X) + c, \quad \text{for all } X \in \mathcal{X}, c \in \mathbb{R}. \quad \text{(H4)}$$

Distributionally robust optimization

Intuitively, this means that the risk functional ρ should not be perturbed by deterministic outcomes. Interestingly, we will observe soon in Proposition 3.6 that support functions satisfying this property necessarily have their support included in the hyperplan $\{\mu \in \mathcal{Y}, \text{ s.t. } \mu(\Omega) = 1\}$. Hence, combined with (H3), this property ensures that all measures in the support $\mathcal{A}$ are necessarily probability measures. In such case, the support $\mathcal{A}$ is often called an *ambiguity set* and the problem of minimizing the composition $w \mapsto \rho(f(w, \xi))$ is referred to as a *distributionally robust optimization* (DRO) problem:

$$\rho(X) = \sup_{\mu \in \mathcal{A}} \mathbb{E}_\mu[X]. \tag{3.4}$$

Divergence-based ambiguity sets

Example 3.2. A popular approach for the modelling of a support goes through the design of a divergence function [11, 33]. A closed proper convex map $\varphi : \mathbb{R}_+ \to \mathbb{R}_+ \cup \{+\infty\}$ is called a *divergence function* if $1 \in \operatorname{int}\operatorname{dom}\varphi$ and $\min \varphi = \varphi(1) = 0$. Then given some ground measure $\boldsymbol{\mu} \in \mathcal{Y}$, the φ-divergence of a measure $\nu \ll \boldsymbol{\mu}$ that is absolutely continuous with respect to $\boldsymbol{\mu}$, is defined as:

$$\mathcal{D}_\varphi(\nu, \boldsymbol{\mu}) = \int_\Omega \varphi\left(\frac{d\nu}{d\boldsymbol{\mu}}\right) \mathrm{d}\boldsymbol{\mu}.$$

where $\frac{d\nu}{d\boldsymbol{\mu}}$ denotes the Radon-Nikodym derivative of ν with respect to $\boldsymbol{\mu}$. For a fixed threshold $r > 0$ designed to model the level of aversion to risk of the decision maker, the associated distributionally robust optimization problem writes:

$$\min_{w \in \mathbb{R}^d} \sup_{\substack{\nu \ll \boldsymbol{\mu} \\ \mathcal{D}_\varphi(\nu, \boldsymbol{\mu}) \leq r}} \mathbb{E}_\nu[f(w, \xi)]$$

Wasserstein ambiguity sets

Example 3.3. The support $\mathcal{A}$ in (3.4) can also be a ball with respect to a given metric in the space of measures. For example, Wasserstein ambiguity sets have been receiving much attention, e.g. [64, 83, 146]. For any couple of measures $\mu, \nu \in \mathcal{Y}$, the *Wasserstein distance* between μ and ν is defined for $p \geq 1$ as:

$$\mathcal{W}(\mu, \nu) := \inf_{\pi \in \Pi(\mu, \nu)} \left(\int_{\Omega^2} \|x - y\|^2 \, \mathrm{d}\pi(x, y) \right)^{1/p}.$$

where $\Pi(\mu, \nu)$ denotes the set of product measures having marginals μ and ν. Given a sample of values $\xi_1, \ldots, \xi_n$ and the associated empirical distribution $\mathbb{P}_n$, for a safety radius $r > 0$, the data-driven Wassertein DRO problem writes

$$\min_{w \in \mathbb{R}^d} \sup_{\substack{\mu \in \mathcal{Y} \\ \mathcal{W}(\mu, \mathbb{P}_n) \leq r}} \mathbb{E}_\mu[f(x, \xi)].$$

Support functions satisfying the properties (H1) to (H4) are called *coherent* risk measures and were introduced in [5]. A growing body of literature has examined their properties from various angles, including structural properties [138, 147], statistical properties [94, 112] and applications [28, 89, 106]. This thesis may be seen as following the same line of works, with a focus on a particular class of coherent risk measures that will be specified in the forthcoming Section 3.2.

3.1.4 *Subdifferential formula*

Given a random loss $w \mapsto f(w, \xi)$ and coherent risk measure $\rho : \mathcal{X} \to \mathbb{R}$, we consider

$$\min_{w \in \mathbb{R}^d} F(w) := \rho(f(w, \xi)). \tag{3.5}$$

Various methods may be used for the minimization of such functions, depending on the regularity properties of f, the statistical properties of ξ, and the geometry of the ambiguity set $\mathcal{A}$. When f is a convex normal integrand, one may solve (3.5) via subgradient-based methods [145]. Let us first recall the subdifferentiation rules for ρ.

Subdifferential of a risk measure

We define an *algebraic subgradient* of a proper convex l.s.c. risk functional $\rho : \mathcal{X} \to \mathbb{R}$ at X as any linear functional ℓ satisfying:

$$\rho(X') \geq \rho(X) + \ell(X' - X), \quad \forall X' \in \mathcal{X}\,. \tag{3.6}$$

While such linear functionals need not be continuous, we call the subset of those that are continuous the *subdifferential* of ρ and we denote it $\partial\rho(X)$. Since $\mathcal{Y}$ is assumed to be the topological dual of $\mathcal{X}$, any element of this set is of the form $\langle \mu, . \rangle$ for some $\mu \in \mathcal{Y}$.

For $\mu \in \partial\,\rho(X)$, we have by definition of the Fenchel conjugate:

$$\rho^*(\mu) \geq \langle \mu, X \rangle - \rho(X).$$

Since relation (3.6) is equivalent to

$$\langle X, \mu \rangle - \rho(X) \geq \langle X', \mu \rangle - \rho(X'), \quad \forall X' \in \mathcal{X},$$

taking the supremum on the right-hand side yields:

$$\langle X, \mu \rangle - \rho(X) = \rho^*(\mu)$$

By Fenchel-Moreau's theorem, we thus have:

$$\sup_{\nu \in \mathcal{Y}} \langle X, \nu \rangle - \rho^*(\nu) = \rho(X) = \langle X, \mu \rangle - \rho^*(\mu)$$

which means

$$\mu \in \arg\max_{\nu \in \mathcal{Y}} \langle X, \nu \rangle - \rho^*(\nu)$$

Conversely, if $\mu \in \arg\max_{\nu \in \mathcal{Y}} \langle X, \nu \rangle - \rho^*(\nu)$, then using again Fenchel-Moreau's theorem yields for any $X' \in \mathcal{X}$:

$$\begin{cases} \rho(X) & = \langle X, \mu \rangle - \rho^*(\mu) \\ \rho(X') & \geq \langle X', \mu \rangle - \rho^*(\mu) \end{cases}$$

which implies

$$\rho(X') \geq \langle X', \mu \rangle + \big(\rho(X) - \langle X, \mu \rangle\big)\,.$$

which yields $\arg\max_{\nu \in \mathcal{Y}} \langle X, \nu \rangle - \rho^*(\nu) \subset \partial\rho(X)$. Hence, we obtain a description of the subdifferential of ρ as:

$$\partial(X) = \arg\max_{\nu \in \mathcal{Y}} \langle X, \nu \rangle - \rho^*(\nu).$$

Finally, we may then invoke the *chain formula* [145] to subdifferentiate the composition.

The subdifferential formula

Proposition 3.2. *Let $w \mapsto f(w,\xi)$ be a finite-valued convex normal integrand that is continuous at $\bar{w}$. Then $w \mapsto F(w) = \rho(f(w,\xi))$ is subdifferentiable at $\bar{w}$ with*

$$\partial F(\bar{w}) = \overline{\text{conv}} \left\{ \sup_{\mu \in \partial \rho(f(\bar{w},\xi))} \int_\Omega \partial_w f(\bar{w},\xi)\, \mathrm{d}\,\mu \right\}. \tag{3.7}$$

In the Section 4.2.1, we will focus on the special case where ρ is the superquantile risk measure. We will extend (3.7), to the case where F is not convex together with an explicit reformulation.

3.2 COHERENCY OF A CLASS OF SUPPORT FUNCTIONS

We present now a special class of risk measures that will be central in this thesis. We first precise the concept of coherency for risk measures on finite support. We consider then the class of risk measures of interest by introducing the concepts of *law-invariance* and *comonotonicity*. We provide then a brief overview of the superquantile risk measures. We finally explain how the superquantile enables to recover any risk measures of interest via *Kusuoka's representation*.

In this section, we provide proofs of all the results, lemmas and propositions, though these results are known and actually hold in general setups (see e.g. [120, 147]). Our proofs rely on basic results from linear algebra and convex analysis that we highlight along the developments.

Risk measures in the discrete setting

Coherency in Risk-averse Optimization. To simplify the presentation, assume the universe to be finite $\Omega = \{\xi_1, \dots, \xi_n\}$. The sigma-algebra is fixed as $\mathcal{F} = 2^\Omega$, and the probability measure is set to be the empirical distribution $\mathbb{P}_n(\xi_i) = 1/n$ for all $i \in [\![1, n]\!]$. Random variables $X \in \mathcal{X}$ and measures $\mu \in \mathcal{Y}$ can both be identified with vectors x, q in $\mathbb{R}^n$:

$$x_i = X(\xi_i), \quad q_i = \mu(\{\xi_i\}), \quad \forall\ 1 \le i \le n,$$

and the pairing reduces to the usual dot product:

$$\mathbb{E}_\mu[X] = q^\top x = \sum_{i=1}^n X(\xi_i)\mu(\{\xi_i\}), \quad \forall X \in \mathcal{X}, \forall \mu \in \mathcal{Y}.$$

In the previous section, we saw with Fenchel-Moreau's theorem 3.1 that any closed proper convex risk measure ρ may be written as the solution of a robust problem (3.3). When the Fenchel conjugate ρ^* of ρ is the indicator of a convex compact subset $\mathcal{C} \subset \mathbb{R}^n$, ρ turns out to be a support function:

$$\rho(x) = \sup_{q \in \mathcal{C}} q^\top x. \tag{3.8}$$

We actually have the following correspondence.

Standard properties of support functions

Proposition 3.3. *The function ρ defined in (3.8) satisfies the following assertions:*

(i) [Normalization] $\rho(0) = 0$.

(ii) [Positive Homogeneity] For any $\lambda > 0$, $\rho(\lambda x) = \lambda\, \rho(x)$.

(iii) [Proper l.s.c convexity] ρ *is proper, closed and convex on* $\mathbb{R}^n$.

(iv) [Finiteness] $\operatorname{dom} \rho = \mathbb{R}^n$

Conversely any function $\varphi : \mathbb{R}^n \to \mathbb{R}$ satisfying the above four assumptions is the support function of a convex compact of $\mathbb{R}^n$.

Proof. The first three assertions fall with Prop. 2.1.2 and 2.1.3 from the chapter C of [62]. Conversely any function satisfying the first three assumptions is the support function of a closed convex subset, by Prop. 3.1.1 from the Chapter C of [62]. Fourth assumption falls from compactness of $\mathcal{C}$. Conversely, we just have to show the compactness of the support set $\mathcal{C}$. We already know that $\mathcal{C}$ is closed. It remains then to show that it is bounded to get compactness, since $\mathbb{R}^n$ has finite dimension. We first note that since ρ is convex and $\operatorname{dom}\rho = \mathbb{R}^n$, ρ is continuous. By contradiction, if $\mathcal{C}$ is not bounded, let $(q_n)_{n\geq 0} \in (\mathcal{C}\backslash\{0\})^{\mathbb{N}}$ such that $|q_n| \to +\infty$ and $x_n = \frac{q_n}{\|q_n\|}$. Since $\mathbb{R}^n$ has finite dimension, its unit ball is compact we may extract a converging sequence $(x_{\psi(n)})_{n\geq 0} \to \bar{x}$ of $(x_n)_{n\geq 0}$. Thus, for all $n \in \mathbb{N}$, we have: $\rho(x_{\psi(n)}) \geq q_{\psi(n)}^\top x_{\psi(n)} = \|q_{\psi(n)}\| \to \infty$ which brings the contradiction. □

In this section, we investigate the relationship between additional properties on the support function ρ and the geometry of the corresponding support $\mathcal{C}$.

Growth and inclusion in the positive cone

Proposition 3.4. *Let ρ be a function defined as in (3.8). Then, the following assertions are equivalent:*

(i) The function ρ is increasing, i.e.: for any, $x, y \in \mathbb{R}^n$ such that $x_i \leq y_i$ for all $i \in \{1, \dots, n\}$ we have $\rho(x) \leq \rho(y)$

(ii) The support $\mathcal{C}$ belongs to the positive cone $\mathbb{R}^n_+$.

Proof. $(i) \implies (ii)$: Let $(e_1, \dots, e_n)$ denote the canonical basis of $\mathbb{R}^n$. For any $i \in \{1, \dots, n\}$, we have: $\rho(-e_i) \leq \rho(0) \Leftrightarrow \inf_{q\in\mathcal{C}} q_i \geq 0$ which yields $\mathcal{C} \subset \mathbb{R}^n_+$. $(ii) \implies (i)$: Let x, y satisfying assumption (i). Then, since $\mathcal{C} \subset \mathbb{R}^n_+$, we have: $q^\top(y - x) \geq 0, \forall q \in \mathcal{C} \implies q^\top x \leq q^\top y, \forall q \in \mathcal{C} \implies \rho(x) = \max_{q\in\mathcal{C}} q^\top x \leq \max_{q\in\mathcal{C}} q^\top y = \rho(y)$. □

Translation invariance and inclusion in the simplex

Proposition 3.5. *Let ρ be a function defined as in (3.8). Then, the following assertions are equivalent:*

(i) The function ρ is translation invariant: $\rho(x + \lambda e) = \rho(x) + \lambda$ for any $\lambda \in \mathbb{R}$, where $e = (1, \dots, 1)^\top$

(ii) The support belongs to the hyperplane $\{q \in \mathbb{R}^n, \sum_{i=1}^n q_i = 1\}$.

Proof. The implication $(i) \implies (ii)$ goes as follows. Note that $\rho(e) = \sup_{q\in\mathcal{C}} \sum_{i=1}^n q_i = \rho(0) + 1 = 1$ and $\rho(-e) = -\inf_{q\in\mathcal{C}} \sum_{i=1}^n q_i = \rho(0) - 1 = -1$. Hence $\mathcal{C} \subset \{q \in \mathbb{R}^n, \sum_{i=1}^n q_i = 1\}$. The reverse implication is immediate. □

Hence, if we assume our support function ρ to satisfy both the growth and the translation invariance properties, its support belongs then to the unit simplex $\Delta := \{q \in \mathbb{R}^n, \sum_{i=1}^n q_i = 1, 0 \leq q_i \ \forall i \in \{1, \dots, n\}\}$, which spans the space of probability measures q on the discrete space $\Omega = \{\xi_1, \dots, \xi_n\}$. We recover in this discrete setting the concept of *coherent* risk measure introduced in Section 3.2.

Law-invariance

Law-invariant risk measures. We propose now to focus, in this discrete setting, on risk measures that only depends on the cumulative distribution function of the input random variables $x \in \mathbb{R}^n$. Such risk measures are called *law invariant*. As shown in the next lemma, law invariance is directly linked to the symmetry properties of the support. Throughout, we will denote $\mathfrak{S}_n$ the permutation set of $[\![1, n]\!]$.

Law invariance and symmetries of the support

Proposition 3.6. *Let ρ be defined as in (3.8). The two following assertions are then equivalent:*

(i) The function ρ is invariant with respect to symmetries, i.e.:

$$\rho(x) = \rho \circ \sigma(x) \quad \forall \sigma \in \mathfrak{S}_n, \forall x \in \mathbb{R}^n$$

(ii) The support $\mathcal{C}$ is invariant with respect to symmetries: $\mathcal{C} = \sigma(\mathcal{C})$ for any $\sigma \in \mathfrak{S}_n$.

Proof. For any $\sigma \in \mathfrak{S}_n$, let us denote by T_σ, the associated permutation matrix defined by: $T_{\sigma\, i,j} = 1$ if $j = \sigma(i)$ and 0 otherwise. We note then for any $x \in \mathbb{R}^n$, $\sigma \in \mathfrak{S}_n$ that:

$$\rho(\sigma(x)) = \max_{q \in \mathcal{C}} \langle q, T_\sigma x \rangle = \max_{q \in \mathcal{C}} \langle T_\sigma^\top q, x \rangle = \max_{q' \in T_\sigma^\top(\mathcal{C})} \langle q', x \rangle = \max_{q' \in \sigma^{-1}(\mathcal{C})} \langle q', x \rangle \,, \tag{3.9}$$

since any permutation matrix is orthogonal. Hence $\rho \circ \sigma$ is the support function of the set $\sigma^{-1}(\mathcal{C})$. This gives $(ii) \implies (i)$. Conversely, we know from [62] that two closed convex sets are equal if and only if their support functions are equal, which yields with (3.9) the result. □

We can actually leverage the symmetry properties of the support of law-invariant risk measures to recover information the distribution $q \in \mathcal{C}$ which achieves the maximum of the support function for a given input $x \in \mathbb{R}^n$. To do so, we define for any distribution $q \in \mathcal{C}$ the *orbit* of q, denoted $\mathcal{O}(q)$ as: $\mathcal{O}(q) = \{\sigma(q), \sigma \in \mathfrak{S}_n\}$. The relation $q' \in \mathcal{O}(q)$ defines then an equivalence relation on $\mathcal{C}$. For any $q \in \mathcal{C}$, we will denote $\bar{q} = \mathcal{O}(q)$ its associated equivalence class. For any $u \in \mathbb{R}^n$, we call *sorting permutation* any permutation $\sigma_u \in \mathfrak{S}_n$, such that:

$$u_{\sigma_u(1)} \leq \cdots \leq u_{\sigma_u(n)}$$

Law invariance and the rearrangement inequality

Lemma 3.7. *Let $x \in \mathbb{R}^n$ and $q \in \mathcal{C}$ be fixed. Let σ_x and σ_q be two associated sorting permutations:*

$$x_{\sigma_x(1)} \leq \cdots \leq x_{\sigma_x(n)} \quad \text{and} \quad q_{\sigma_q(1)} \leq \cdots \leq q_{\sigma_q(n)}$$

Then, for any $q' \in \mathcal{O}(q)$, we have:

$$q'^\top x \leq \sigma_x^{-1}(\sigma_q(q))^\top x$$

Proof. By the rearrangement inequality, we have, for any $q' \in \mathcal{O}(q)$:

$$q'^\top x \leq \sum_{i=1}^{n} q_{\sigma_q}(i) x_{\sigma_x}(i) \leq \langle T_{\sigma_q} q, T_{\sigma_x} x \rangle \leq \langle T_{\sigma_x}^{-1} T_{\sigma_q} q, x \rangle$$

which yields the result. □

Comonotonicity

Comonotonicity and generation of the support. We note that for a supporting set $\mathcal{C}$ that is generated by a single distribution, i.e. $\mathcal{C} = \overline{\text{conv}}(\mathcal{O}(q))$ for some $q \in \Delta$, the extreme points of $\mathcal{C}$ constitute the orbit $\mathcal{O}(q)$. Subsequently, this last lemma ensures that evaluating the support function at x boils down to a sorting of the elements of x. Let us now characterize risk measures supported on such sets. For any $x, y \in \mathbb{R}^n$, we say that x and y are *comonotone* if:

$$x_i \leq x_j \iff y_i \leq y_j \quad \forall i, j \in [\![1, n]\!]$$

Comonotonicity yields a second equivalence relation. We note from lemma 3.7, that for any law-invariant support function and for any $x \in \mathbb{R}^n$, $\arg\max_{q\in\mathcal{C}} q^\top x$ is comonotone to x.

A risk measure ρ is said to be *comonotone* if for all $x, y \in \mathbb{R}^n$ that are comonotone, ρ satisfies $\rho(x+y) = \rho(x) + \rho(y)$. As next Proposition shows, comonotonicity can also be tied to the geometry of the support $\mathcal{C}$.

Comonotonicity and generation of the support

Proposition 3.8. *Let ρ be defined as in (3.8) and law-invariant. Then, the two following assertions are equivalent:*

(i) The support function ρ is comonotone.

(ii) The support $\mathcal{C}$ is generated by a single distribution, i.e. there exists $q \in \mathcal{C}$ such that $\mathcal{C} = \overline{\text{conv}}(\mathcal{O}(q))$.

This well-known results was first established in atomless spaces within the seminal paper [84]. Its extension to finite probability spaces was proposed in [17] with a proof relying on submodularity considerations. Here we provide a different proof based on elementary results from convex analysis.

Proof. $(i) \implies (ii)$: Let $q \in \mathcal{C}$ be an exposed point of $\mathcal{C}$. By law-invariance of ρ, we have $\mathcal{O}(q) \subset \mathcal{C}$ and thus convexity of C ensures $\overline{\text{conv}}(\mathcal{O}(q)) = \mathcal{C}$. We want to show that $\overline{\text{conv}}(\mathcal{O}(q)) = \mathcal{C}$. Let us assume by contradiction that there exists an extreme point $\tilde{q}$ of $\mathcal{C}$ such that $\tilde{q} \notin \overline{\text{conv}}(\mathcal{O}(q))$. By Straszewicz's theorem [154] and closedness of $\overline{\text{conv}}(\mathcal{O}(q))$, we may actually assume $\tilde{q}$ to be an exposed point of $\mathcal{C}$. It is then clear that $\mathcal{O}(\tilde{q}) \cap \mathcal{O}(q) = \emptyset$. Let then $q_1, q_2 \in \mathcal{O}(q) \times \mathcal{O}(\tilde{q})$ such that q_1 and q_2 are comonotone. Let x_1 and x_2 two arbitrary points respectively in $\mathcal{N}_\mathcal{C}(q_1)$ and $\mathcal{N}_\mathcal{C}(q_2)$. By the proposition 3.1.4 of the chapter C of [62], $\rho(x_1) = q_1^\top x_1$ and $\rho(x_2) = q_2^\top x_2$. Hence by lemma 3.7 and transitivity of the comonotone equivalence relation x_1 is comonotone to x_2. However, for any $q \in \mathcal{C}$, since q_1 and q_2 are exposed, if $q \neq q_1$, $q^\top x_1 < \rho(x_1)$ and if $q \neq q_2, q^\top x_2 < \rho(x_2)$. Since $q_1 \neq q_2$, we have:

$$q^\top(x_1 + x_2) < \rho(x_1) + \rho(x_2) \quad \forall q \in \mathcal{C})$$

By compactness of $\mathcal{C}$, we get $\rho((x_1 + x_2) < \rho(x_1) + \rho(x_2)$ which yields the contradiction.

$(ii) \implies (i)$: By definition of support functions, $x, y \in \mathbb{R}^n$, $\rho(x+y) \leq \rho(x) + \rho(y)$. Let $x, y \in \mathbb{R}^n$ be comonotone. Then the permutation $\sigma_x \in \mathfrak{S}_n$ that sorts the coordinates of x is also a sorting permutation for y, i.e. we have:

$$\begin{aligned} & x_{\sigma_x(1)} \leq \cdots \leq x_{\sigma_x(n)} \\ \text{and} \quad & y_{\sigma_x(1)} \leq \cdots \leq y_{\sigma_x(n)} \end{aligned}$$

Let $q \in \mathcal{C}$ such that $\mathcal{C} = \overline{\text{conv}}(\mathcal{O}(q))$. Then by lemma 3.7, we have for any $q' \in \mathcal{O}(q)$, $q'^\top x \leq \sigma_x^{-1}\sigma_q(q)^\top x$ and $q'^\top y \leq \sigma_x^{-1}\sigma_q(q)^\top y$. Now since $\arg\max_{q'\in\mathcal{C}} q'^\top x$ and $\arg\max_{q'\in\mathcal{C}} q'^\top y$ both lie in the set of extreme points of $\mathcal{C}$ which is by assumption the orbit of q, we have: $\rho(x) = \sigma_x^{-1}\sigma_q(q)^\top x$ and $\rho(y) = \sigma_x^{-1}\sigma_q(q)^\top y$. Hence, we have:

$$\rho(x) + \rho(y) = \sigma_x^{-1}\sigma_q(q)^\top(x+y) \leq \sup_{q'\in\mathcal{C}} q'^\top(x+y) = \rho(x+y)$$

which finishes the proof. □

Combining the two previous lemmas yields the following corollary.

Risk profile

Corollary 3.9. *Let ρ be defined as in 3.8, law-invariant and comonotone. Let $q \in \mathcal{C}$ such that $\mathcal{C} = \overline{\text{conv}}(\mathcal{O}(q))$. Then, for any $x \in \mathbb{R}^n$, we have:*

$$\rho(x) = \langle \sigma_q(q), \sigma_x(x) \rangle = \langle \sigma_x^{-1}(\sigma_q(q)), x \rangle$$

In other words, this result shows that when the support function ρ satisfies the law-invariance and comonotonicity properties (and assuming that we know an extreme point q of the support $\mathcal{C}$), evaluating ρ at a given $x \in \mathbb{R}^n$ boils down to a sorting of the coordinates of x. This is much less demanding than solving the linear program (3.8) in general. Moreover, we note that designing a law-invariant and comonotone support function amounts to designing a *risk profile*, i.e. an increasing sequence $q^\star$ that sums to one. The support is then recovered through the formula $\mathcal{C} = \overline{\text{conv}}(\mathcal{O}(q^\star))$.

3.3 SUPERQUANTILES AND THE KUSUOKA REPRESENTATION

In this section, we present in detail the superquantile, which is a special example of law-invariant comonotone risk measure. This risk measure was introduced in the introductory Section 1.1 of this thesis. We insist here on its convex and geometric properties, based on the above *coherent* framework.

3.3.1 *The superquantile risk measure*

Superquantile of discrete random variables

In the discrete setting described in Section 3.2, the general formula (1.4) defining the superquantile reduces to a sum that can be further split as

$$S_p(x) = \frac{1}{n(1-p)} \sum_{i \in I_>} x_i + \frac{\delta}{1-p} Q_p(x) \quad \text{with } I_> = \{i : x_i > Q_p(x)\}. \tag{3.10}$$

This expression involves the distance from p to the next discontinuity point of the quantile function: $\delta = F_x(Q_p(x)) - p = \frac{1}{n}(n - |I_>|) - p$,that we illustrate in Figure 3.2. Hence, (3.10) gives an efficient way to compute superquantiles from the following three step procedure: (a) compute the p-quantile with the specialized algorithm (called `quickfind`) of complexity $\mathcal{O}(n)$; (b) select all values greater or equal than the quantile; (c) average values along (3.10). We note also that for probability values p in $\left\{\frac{i}{n}, 1 \leq i \leq n\right\}$, this expression simplifies even more.

Quantiles, superquantiles, and coordinates of a vector

Lemma 3.10. *Let $x \in \mathbb{R}^n$ be a fixed vector and σ_x a sorting permutation for x. For any $i \in [\![1, \dots, n-1]\!]$, we have:*

$$Q_{\frac{i}{n}}(x) = x_{\sigma_x(i)} \qquad S_{\frac{i}{n}}(x) = \frac{1}{n-i} \sum_{k=i+1}^{n} x_{\sigma_x(k)}$$

Proof. Let $i \in \{1, \dots, n-1\}$ be fixed. Then, $\frac{1}{n} \sum_{k=1}^{n} \mathbb{1}_{x_k \leq x_{\sigma_x(i)}} \geq \frac{i}{n}$ so $x_{\sigma_x(i)} \geq Q_{\frac{i}{n}}(x)$. But for any $t < x_{\sigma_x(i)}$, it is also clear that: $\frac{1}{n} \sum_{k=1}^{n} \mathbb{1}_{x_k \leq t} < \frac{i}{n}$ which yields

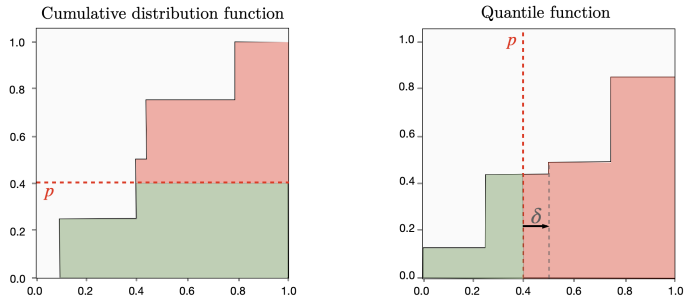


Figure 3.2: Illustration of the integral expression of the superquantile. Cumulative distribution function (on the left) and quantile function (on the right) are each other's inverse. The superquantile is obtained by averaging the quantiles greater than the p-quantile (red section on graph on the right).

the first result. It is in addition clear that for any $i \in [\![1, n-1]\!]$, and for any $p \in (\frac{i}{n}, \frac{i+1}{n}]$, $Q_p(x) = Q_{\frac{i+1}{n}}(x) = x_{\sigma_x(i+1)}$. Hence,

$$S_{\frac{i}{n}}(x) = \frac{1}{1-\frac{i}{n}} \int_{p'=\frac{i}{n}}^{1} Q_{p'}(x)\mathrm{d}p' = \frac{n}{n-i} \sum_{k=i}^{n-1} \int_{p'=\frac{k}{n}}^{\frac{k+1}{n}} Q_{p'}(x)\mathrm{d}p$$
$$= \frac{n}{n-i} \sum_{k=i}^{n-1} \frac{1}{n} Q_{\frac{k+1}{n}}(x)\mathrm{d}p' = \frac{1}{n-i} \sum_{k=i+1}^{n} x_{\sigma_x(i+1)}$$

which ends the proof. □

DUAL FORMULATIONS.

On top of expression (3.10), the superquantile offers two other useful formulations dual of each other:

- The superquantile is the support function of the intersection of the simplex with a box (see Figure 3.3). *Distributionally robust representation*

$$S_p(x) = \max_{q \in \Delta_p} q^\top x \quad \text{with } \Delta_p = \left\{ q \in \mathbb{R}^n_+ : \sum_{i=1}^{n} q_i = 1, q_i \leq \frac{1}{n(1-p)} \right\}. \quad (3.11)$$

  This problem corresponds to a classical optimization problem, called the fractional knapsack problem, which is solved, after sorting the x_i's, by a simple greedy strategy of the associated q_i's [35], yielding back the discrete formulation (3.10).

- Another expression, which initially appeared in [139], is *Variational representation*

$$S_p(x) = \inf_{\eta \in \mathbb{R}} \left\{ \eta + \frac{1}{n(1-p)} \sum_{i=1}^{n} \max(x_i - \eta, 0) \right\}. \quad (3.12)$$

  This formulation can be derived through Lagrangian duality from (3.11): it suffices to dualize the simplex constraint $\sum_{i=1}^{n} q_i = 1$. It will be central in the developments of Chapter 5 of this thesis.

As a support function given in (3.11), we see that the superquantile is a *Extreme points of the support*

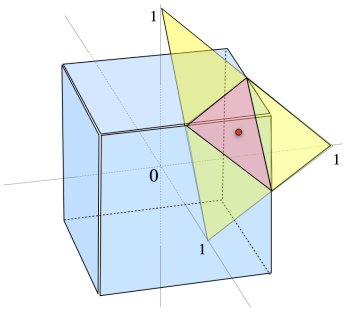


Figure 3.3: Illustration of the dual expression of the superquantile (recall of Figure 1.2 from Section 1.1). $x \mapsto S_p(x)$ is the support function of the red polytope. The red point at the center represents the uniform distribution.

coherent risk measure. Moreover, we note that the support Δ_p is the convex hull of the orbit $\mathcal{O}(q_p)$ with

$$q_p := \left\{ 0, \dots, 0, \delta, \underbrace{\frac{1}{n(1-p)}, \dots, \frac{1}{n(1-p)}}_{n\times(1-p-\delta) \text{ times}} \right\}.$$

We therefore deduce from Propositions 3.6 and 3.8 that it is law-invariant and comonotone. In practice, superquantile have been shown experimentally to produce models more robust to distributional shifts in various contexts; we refer to [34, 74, 85, 94]. We will provide numerical illustrations in Section 4.4.

3.3.2 *The Kusuoka representation*

As we have seen with Corollary 3.9, any law-invariant comonotone risk measures can be written as a convex combination of quantiles of the input vector x. However, quantiles are know to be non-convex and nonsmooth functions. In this section, we present a standard representation of law-invariant comonotone risk measures known as the *Kusuoka representation* [84, 120]. We will leverage this representation in Section 4.2 to produce efficient first-order oracles for the minimization of such risk measures.

Kusuoka Representation

Proposition 3.11. *Let ρ be defined as in (3.8), law-invariant and comonotone. Let $q \in \mathcal{C}$ such that $\mathcal{C} = \overline{\text{conv}}(\mathcal{O}(q))$ and $\alpha := \sigma_q(q)$. Then for any $x \in \mathbb{R}^n$, we have:*

$$\rho(x) = \sum_{k=1}^{n} \beta_k S_{\frac{k-1}{n}}(x) \tag{3.13}$$

where $(\beta_k)_{1\le k\le n}$ satisfies : $\beta_1 = n\alpha_1$, $\beta_i = (n+1-i)(\alpha_i - \alpha_{i-1})$ for $i \in \{2, \dots, n\}$.

Proof. Let $x \in \mathbb{R}^n$ be fixed. For any $i \in [\![1, \dots, n-1]\!]$, let $\bar{X}_i := \sum_{k=i}^n x_{\sigma_x(k)} = (n-i+1)S_{\frac{i-1}{n}}(x)$. By corollary 3.9, we have:

$$\begin{aligned}
\rho(x) &= \sum_{i=1}^n \alpha_i x_{\sigma_x(i)} = \sum_{i=1}^n \alpha_i \left(\sum_{k=i}^n x_{\sigma_x(k)} - \sum_{k'=i+1}^n x_{\sigma_x(k')} \right) \\
&= \sum_{i=1}^n \alpha_i (\bar{X}_i - \bar{X}_{i+1}) \quad \big(\text{with convention } \bar{X}_{n+1} = 0\big) \\
&= \sum_{i=1}^n \alpha_i \bar{X}_i - \sum_{i=2}^n \alpha_{i-1} \bar{X}_i = \alpha_1 \bar{X}_1 + \sum_{i=2}^n (\alpha_i - \alpha_{i-1}) \bar{X}_i \\
&= n\alpha_1 S_0(x) + \sum_{i=2}^n (n+1-i)(\alpha_i - \alpha_{i-1}) S_{\frac{i-1}{n}}(x)
\end{aligned}$$

which yields the result. □

One may observe from the above proof that the Kusuoka representation stems from an elementary "integration by part". In particular, we did not assume ρ to satisfy the growth condition from Proposition 3.4. Hence the Kusuoka representation, usually given for coherent risk measures, holds de facto for any linear combination of quantiles functions. In particular, the developments of Section 4.2 will remain valid for other classes of functions alike such as cardinality-based submodular functions.

3.4 MINIMIZATION BY FIRST-ORDER METHODS

Minimization of superquantile-based objectives comes with a number of technical challenges on the structure of the problem tackled, the size of the dataset or the nonsmoothness of the objective. Standard works on minimizing superquantiles considered linear programming or convex programming techniques, including interior point algorithms; see the review of [135]. Surprisingly, the use of first-order algorithm for superquantile-based optimization is quite recent and seems to be driven by applications in machine learning.

In this section, we provide an overview of the range of first-order methods to minimize superquantile-based objective functions. Precisely, we consider problems of the form:

$$\min_{w \in \mathbb{R}^d} S_p[f(w, \xi)] \tag{3.14}$$

where $\xi : \Omega \to \mathbb{R}$ is uniform over a a batch of training samples $\mathcal{D} = \{\xi_1, \dots, \xi_n\}$. Our discussion focuses on practical considerations; we give pointers to references presenting more details and theoretical analysis.

Batch algorithms. The first approach for minimizing superquantile-based objective functions is to use standard subgradient-based methods (subgradient and dual averaging) or gradient-based methods (gradient, accelerated gradient, Quasi-Newton). More precisely, we have two cases:

- If $w \mapsto f(w, z)$ is convex for all z, then composition 3.14 is convex and we may compute a subgradient according to (3.7) (we will make this more concrete in the forthcoming Proposition 4.1) with the same complexity as the one for computing a quantile. We can use standard convex nonsmooth optimization methods, such as subgradient methods and dual averaging. These algorithms satisfy ergodic convergence guarantees in objective values [15]. *Convex case*

Smooth case

- If $w \mapsto f(w, z)$ is smooth for all z, then we can smooth the superquantile (see forthcoming Section 4.2.2) to get a gradient oracle that approximates the composition 3.14. We can use standard methods for smooth optimization: gradient method, accelerated gradient method, and quasi-Newton (L-BFGS). If furthermore we have convexity, these algorithms satisfy convergence guarantees in objective values [15, 16].

For small to medium-size datasets, such batch methods are shown to be simple and efficient; see [85, Sec. 4] and forthcoming developments of Chapter 4. For large-scale problems though, the oracles become too costly as they require sorting loss values on the whole data set. We turn to the other formulations to introduce stochastic and mini-batch algorithms, that usually are the methods of choice for the case of standard learning using empirical risk minimization.

Mini-batch algorithms. Mini-batch algorithms for the minimization of the superquantile have received much attention in recent years. Given the diverse formulations of the superquantile, several approaches have been considered.

Smoothing the variational representation (3.12)

From the perspective of the formulation (3.12) of the objective, the superquantile-based learning problem writes

$$\min_{w \in \mathbb{R}^d} \min_{\eta \in \mathbb{R}} \left\{ \frac{1}{n(1-p)} \sum_{i=1}^{n} \max\{f(w, \xi_i) - \eta, 0\} + \eta \right\}. \tag{3.15}$$

When the loss is assumed to be smooth, one may again smooth the inner $\max\{\cdot, 0\}$ term to get a smooth approximation of this joint objective. One can then perform a joint minimization with respect to the model w and the dual variable η. In other words, superquantile learning reduces to a standard empirical risk minimization with a modified loss function truncated by the max-term. In practice batch methods may not be interesting here, since they would not leverage the fact that the minimization over η can be performed explicitly. Thus [89] proposes, in a context of federated learning, to rather perform independent minimization over x and η alternatively. The min-min approach paves the way to stochastic and mini-batch algorithms.

Stochastic approaches to (3.12)

Several works , including [152] and [174] (as well as [48] without mentioning superquantile), use successfully standard stochastic optimization algorithms on this modified objective. Observe though that, if a mini-batch of data is sampled uniformly at random from the data, only a fraction $(1-p)$ carry (sub)gradient information. Furthermore, the (sub)gradients of these examples are scaled by $\frac{1}{1-p}$, leading to exploding directions. Thus mini-batch estimates of (sub)gradients of superquantile-based objectives may suffer from high variance. A solution proposed by [34] is to perform an adaptive sampling rather than a uniform one. This algorithm gradually adjusts its sampling distribution to increasingly sample tail events, until it eventually minimizes the superquantile. This approach has a nice two-player interpretation related to the third formulation, recalled below.

Mini-batch approaches for the DRO formulation (3.11)

The DRO expression (3.11) of f leads to the following formulation

$$\min_{w \in \mathbb{R}^d} \max_{q \in \Delta_n} \left\{ \sum_{i=1}^{n} q_i f(w, \xi_i) : 0 \leq q_i \leq \frac{1}{n(1-p)} \right\}. \tag{3.16}$$

This min-max formulation offers several ways to solve the superquantile-based learning. A first approach would consist in considering it as a generic saddle point problem and using standard (extra-)gradient algorithms or recent extensions exploiting some aspects of the problem (see e.g. [107] for a variance-

reduced min-max with strongly concave max). In our specific case, computing the max can be done systematically by a greedy algorithm with quasi-linear time complexity (see Section 4.2). This key feature is exploited by the stochastic algorithm of [44], and also by the one of [74] without relating it to superquantile. This algorithm uses a biased sampling approximation to f or f_μ which has nice guarantees. We briefly describe below this approach.

Mini-batch stochastic estimator of the superquantile

We sample a mini-batch $\mathcal{S}$ of size s, uniformly in $\mathcal{D}$ and we consider the restriction

$$\begin{aligned}\widetilde{f}(w) &= [S_p]_{(a,b)\sim\mathcal{S}}\big(f(w,\xi)\big)\\ &= \max_{q\in\Delta_s}\left\{\sum_{i\in\mathcal{S}} q_i\, f(w,\xi_i) : 0\le q_i\le\frac{1}{s(1-p)}\right\}.\end{aligned}$$

Alternatively, one may smooth this stochastic estimator. We will discuss possible smoothing procedures in the upcoming Section 4.2.2. Specialized here for a given smoothing parameter $\nu>0$ and a strongly convex function D, this gives

$$\widetilde{f}^{\nu}(w) = \max_{q\in\Delta_s}\left\{\sum_{i\in\mathcal{S}} q_i\, f(w,\xi_i) - \nu D(q) : 0\le q_i\le\frac{1}{s(1-p)}\right\}. \tag{3.17}$$

Using the (sub)gradient oracles that will be presented in Section 4.2 on $\widetilde{f}^{(\nu)}$, we can run stochastic gradient methods. These methods require a number of gradient evaluations independent of training set size and number of parameters, making them suitable for large-scale applications. However, one should note that $\widetilde{f}$ happens to be a biased estimator of f, in view of the two following results [94]. We will use them in Chapter 6.

Bias of $\widetilde{f}$ [94]

Proposition 3.12. *Let U_s denote the uniform distribution over all subsets of $[n]$ of size s. Assume that loss f is bounded: $f(w,\xi_i)\le B$, $\forall w\in\mathbb{R}^{\mathrm{d}}, \forall i\in[\![1,n]\!]$. Then, for any $w\in\mathbb{R}^{\mathrm{d}}$, we have*

$$\left|\mathbb{E}_{\mathcal{S}\sim U_s}[\widetilde{f}(w)] - f(w)\right| \le \frac{B}{\sqrt{(1-p)s}}$$

Under similar assumptions, one may also bound the bias of the smoothed stochastic estimator (3.17) as well as the variance of its gradient.

Bias and variance of $\nabla\widetilde{f}^{\nu}$ [94]

Proposition 3.13. *Let U_s denote the uniform distribution over all subsets of $[n]$ of size s. Assume that loss f is Lipschitz with respect to w: $f(w,\xi_i)-f(w',\xi_i)|\le G$, $\forall w,w'\in\mathbb{R}^{\mathrm{d}}$. Then, for any $w\in\mathbb{R}^{\mathrm{d}}$, we have*

$$\begin{aligned}\left|\mathbb{E}_{\mathcal{S}\sim U_s}[\widetilde{f}^{\nu}(w)] - f(w)\right| &\le \frac{B}{\sqrt{(1-p)s}} + 2\nu\log s\,,\\ \mathbb{E}_{S\sim U_m}\left\|\nabla\widetilde{f}^{\nu}(w) - \nabla\,\mathbb{E}_{\mathcal{S}\sim U_s}\widetilde{f}^{\nu}(w)\right\|^2 &\le \frac{8G^2}{(1-p)s}\,.\end{aligned}$$

A comparison between batch and stochastic methods will be presented in the Section 4.4 of this thesis.

3.5 CHALLENGES IN CHANCE CONSTRAINED PROGRAMMING

We end this chapter with a related topic that will be explored in Chapter 5. Chance constraints appear as a versatile way to model the exposure to uncertainty

in optimization. Introduced in [25], they have been used in many applications, such as in energy [129, 162] and telecommunications [110]. We refer to the seminal paper [127], the book chapter [38] for an introduction to the theory, and to the recent article [158] for a discussion covering recent developments.

Chance constraints

Chance constraints result from taking the probability measure of a given set of random inequalities depending on the decision vector. Formally speaking, given a map $g\colon E \times \mathbb{R}^m \to \mathbb{R}^k$, where E is a (reflexive) Banach space and a random vector $\xi \in \mathbb{R}^m$ defined on an appropriate probability space, we first define the probabilistic constraint function $\varphi\colon X \to [0,1]$ as:

$$\varphi(w) := \mathbb{P}[g(w,\xi) \leq 0]\,. \tag{3.18}$$

Chance constrained problems

As its name suggests, a chance constrained problem is an optimization problem of the form:

$$\begin{cases} \min\limits_{w \in S} & f(w) \\ \text{s.t.} & \varphi(w) \geq p \end{cases} \tag{3.19}$$

where $f\colon \mathbb{R}^{\mathrm{d}} \to \mathbb{R}$ is a given objective function, $S \subset E$ is an arbitrary set of deterministic constraints and $p \in (0,1)$ a user-defined safety level. The interpretation of (3.19) is simple: one requires the decision w to minimize the cost function f while satisfying both the deterministic constraint $w \in S$, and, the random inequality system $g(w,\xi) \leq 0$ with a probability at least greater than p. Letting $M(p)$ be the set of admissible points for the probabilistic constraint,

$$M(p) = \left\{w, \varphi(w) \geq p\right\},$$

this problem rewrites

$$\min_{w \in S \cap M(p)} f(w).$$

From now on, we will assume f, g, and S to be convex in order to bring to light the inherent difficulties of chance constraints. Indeed, such problems are still designated as especially difficult for two reasons in particular, that we summarize below.

Nonsmoothness of chance constraints

Nonsmoothness. When considering a parametric distribution in place of the random variable ξ, one may deal with the above chance constraint through the (generalized) differentiation of the probabilistic function φ. Indeed, the development of readily implementable oracles for probability functions has been studied in a growing body of literature [143, 156, 160].

Nonetheless, when ξ follows a discrete distribution, which typically occurs with sample-based approximations approaches, working through the subdifferentiation of φ becomes impracticable and alternative methods may be considered. Popular numerical methods for dealing with such constraints may be boolean approaches, e.g., [79, 80], p-efficient point based concepts, e.g., [39, 40, 163], robust optimization [9], penalty approach [47], scenario approximation [23, 132], convex approximation [115], or yet other approximations [53, 67].

Non-convexity of chance constraints

Non-convexity. The possible non-convexity induced by the chance constraint can considerably complicate the handling of the problem. Understanding when $M(p)$ is a convex set is important for the point of view of optimization, to guarantee that local solutions are also globally optimal and to use numerical solution methods that exploit this convexity. A first result of the convexity of $M(p)$ follows from Prékopa's celebrated log-concavity theorem (see [49, Proposition 4] for its infinite dimensional version and [43] for generalizations):

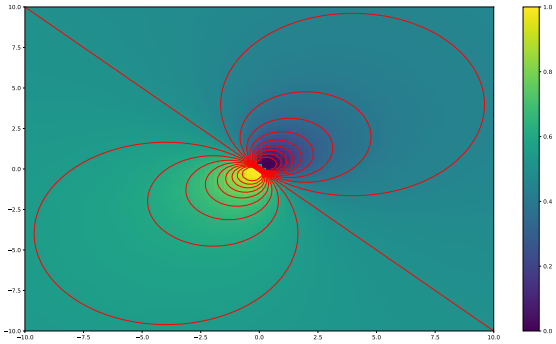


Figure 3.4: Kataoka's example of eventually convex chance constraint. Here ξ follows a 2-dimensional gaussian distribution with parameters $\mu = (1, 1)$ and $\Sigma = I_2$. Even in simple cases, chance constraints are not garanteed to be convex for all values of p.

the convexity of $M(p)$ is guaranteed for all $p \in [0, 1]$, when $-g$ is jointly quasi-concave in both arguments and ξ an appropriate random vector. However, joint-quasi-concavity of $-g$ is rather exceptional and fails in many basic situations.

Kataoka's example
See e.g. [73]

Example 3.4. Take $g(x, \xi) = x^\top \xi$ and ξ to be a multi-variate Gaussian with mean $\mu \in \mathbb{R}^n$ and covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$. Then $M(p)$ is known to be convex only whenever $p \geq \frac{1}{2}$. Indeed, denoting z an arbitrary 1-dimensional standard gaussian variable and ϕ its cumulative distribution function, we observe that for any $x \in \mathbb{R}^n$, one has

$$\mathbb{P}[x^\top \xi] = \mathbb{P}\left[x^\top \mu + \sqrt{x^\top \Sigma x} z \leq 0\right] = \phi\left(\frac{-x^\top \mu}{\sqrt{x^\top \Sigma x}}\right).$$

Hence,

$$\mathbb{P}[x^\top \xi] \geq p \Leftrightarrow \phi\left(\frac{-x^\top \mu}{\sqrt{x^\top \Sigma x}}\right) \geq p \Leftrightarrow x^\top \mu + \sqrt{x^\top \Sigma x}\phi^{-1}(p) \leq 0$$

Thus, in this example, the comvexity of the set $M(p)$ boils down to quasi-convexity of the function $x \mapsto x^\top \mu + \sqrt{x^\top \Sigma x}\phi^{-1}(p)$ which clearly occurs only when $\phi^{-1}(p) \geq 0$, that is when p is greater than 0.5. □

In the above example, we thus observe that if the convexity of $M(p)$ does not hold for all p, there still exists a (computable) threshold $p^* \in [0, 1]$ such that the set $M(p)$ is convex for all $p \geq p^*$. This property is called *eventual convexity* as observed by [128] and coined by [59] (which studies the case where g is separable and ξ has independent components). Eventual convexity results are further generalized in [60] by allowing for the components of ξ to be coupled through a copulæ dependency structure. These results are refined, by allowing for more copulæ and with sharper bounds for p^* in [157], and extended to all Archimedian copulæ in [159], where also an appropriate solution algorithm is provided. When the mapping g is non-separable, eventual convexity results are provided in [161] for the special case where ξ is elliptically symmetrically distributed and generalized in [92].

In the Chapter 5 of this thesis, we will focus on the practical solving of such chance constrained problems in the case where ξ is observable through data sampling.

4

MINIMIZING SUPERQUANTILE-BASED RISK MEASURES

This chapter is devoted to the practical minimization of a given class of convex risk measures and their applications in machine learning. Namely, we consider *law-invariant comonotone* convex risk measures and we show how they can be efficiently minimized via first-order methods. An important focus is given to the superquantile risk measures for its central role within the class. The developments laid down below build upon the following works:

- Y. Laguel, J. Malick, and Z. Harchaoui. First-order optimization for superquantile-based learning. *Proceedings of the IEEE International Workshop on Machine learning for signal processing (MLSP)*,2020.
- Y. Laguel, K. Pillutla, J. Malick, and Z. Harchaoui. Superquantiles at work: machine learning applications and efficient subgradient computation. *Accepted in Set-Valued and Variational Analysis.*
- Y. Laguel, J. Malick, and Z. Harchaoui. Superquantile-based learning: a direct approach using gradient-based optimization. *Under review.*

4.1 INTRODUCTION

In view of the recalls of chapter 3 and the Fenchel-Moreau Theorem 3.1, convexity is a natural assumption for the design of risk measures to enforce robustness in uncertain environments. When the risk measure considered is coherent, (i.e. sub-additive, positively homogeneous, monotone and translation-invariant), it may be represented as the support function of a set in the space of measures, called the *ambiguity set*. The study of the correspondence between the functional properties of a given coherent risk measure and the associated ambiguity set has drawn much attention, see [131] for an recent overview.

General problem

In this chapter, we focus on the practical solving of problems of the form:

$$\min_{w \in \mathbb{R}^d} \rho(f(w, \xi)) \tag{4.1}$$

where ρ denotes a law-invariant coherent comonotone risk measures – see Section 3.2 for a brief overview. A special attention is given to the superquantile risk measure which was first introduced in Section 1.1 of this thesis and further developed in Section 3.3. Our developments rely on a careful analysis of the dual properties of superquantile functions to produce first-order oracles with optimal computational complexity. We also advertise our open-source python software `SPQR` for the minimization of such risk measures and show how it enforce (distributional) robustness in a sequence of numerical experiments.

Outline

In Section 4.2, we study the (sub)-gradient calculus of superquantile-based functions with a focus on computational efficiency. In particular, Section 4.2.1 specifies the general subgradient formula from Eq. (3.7) for superquantile-based

functions and extend it to the case of non-convex losses. Section 4.2.2 considers gradients of smooth approximations of the superquantile by inf-convolution. We establish an equivalence result between smoothing various smoothing procedures of superquantiles in Section 4.2.3. We generalize these smoothing procedures to law-invariant comonotone risk measures in 4.2.4. We provide in Section 4.3 a short presentation of the toolbox SPQR. Finally, we present in Section 4.4 numerical experiments showing the interest of superquantile-based risk measures.

4.2 EFFICIENT (SUB)-GRADIENTS COMPUTATIONS

We propose to solve problems of the form (4.1) by first-order methods. We first focus on superquantile-based learning objectives for which we provide easy-to-implement expressions of subgradients in Section 4.2.1, and of gradients of smoothed approximations of them in Section 4.2.2. For both oracles, we provide efficient subroutines to implement them in linear time. We make connections between several smoothing procedures of the superquantile in Section 4.2.3. We finally generalize in section 4.2.4 these procedures to law-invariant comonotone risk measures with a special care on maintaining the optimal linear time complexity.

4.2.1 *Subdifferentiation via the chain rule*

Superquantile-based losses

In this section, we provide explicit and implementable expressions of the subdifferential of a general superquantile-based loss:

$$f(w) = S_p(L(w)). \tag{4.2}$$

where S_p denotes the p-superquantile defined in (1.4) and $L : w \mapsto L(w, \xi_i)_{1\leq i\leq N}$ denotes a differentiable loss. In this chapter, we do not assume the components of L, i.e. the terms $L_i(w) := L(w, \xi_i)$ to be convex.

Expressions of (convex) subdifferential of superquantiles are well-known in general settings; see e.g., [145] for a thorough study and Proposition 3.2 for a general result. Here we study non-convex subdifferentials and derive concrete expressions in the data-driven context; we give direct proofs as applications of basic definitions and properties of nonsmooth analysis.

We start by recalling the standard notions of subgradients for nonsmooth functions (in finite dimension), following the terminology of [140]. For a function $\psi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$, the regular (or Fréchet) subdifferential of ψ at $\bar{w}$ (such that $\psi(\bar{w}) < +\infty$) is defined by

Fréchet subdifferential

$$\partial^R \psi(\bar{w}) = \left\{ s \in \mathbb{R}^d : \ \psi(w) \geq \psi(\bar{w}) + s^\top (w - \bar{w}) + o(\|w - \bar{w}\|) \right\}.$$

The regular subdifferential thus corresponds to the set of gradients of smooth functions that are below ψ and coincide with it at $\bar{w}$. The limiting subdifferential is the set of all limits produced by regular subgradients

Limiting subdifferential

$$\partial^L \psi(\bar{w}) = \limsup_{w \to \bar{w}, \psi(w) \to \psi(\bar{w})} \partial^R \psi(w).$$

These notions generalize (sub)gradients of both smooth functions and convex functions: for these functions indeed, the two subdifferentials coincide, and

they reduce to $\{\nabla\psi(\bar{w})\}$ when ψ is smooth and to the standard subdifferential from convex analysis when ψ is convex.

For the function (4.2), which is the composition of a convex function and a continuously differentiable function, we get from basic chain rules that the two subdifferentials coincide; we simply denote it by $\partial f(w)$. Moreover the dual representation (3.11) expressing S_p as a support function allows to obtain readily an expression of the subdifferential of ∂S_p and, as a result, of the one of f. We formalize all this in the following proposition.

Explicit subdifferential of superquantile-based functions

Proposition 4.1. *Consider the superquantile-based function (4.2) with L continuously differentiable. We have*

$$\partial f(\bar{w}) = \left(\partial^L f(\bar{w}) = \partial^R f(\bar{w}) = \right) \nabla L(\bar{w})^* \partial\, S_p(L(\bar{w})) \tag{4.3}$$

where $\nabla L(\bar{w})^*$ is the adjoint of the Jacobian of L at $\bar{w}$ and $\partial\, S_p(L(\bar{w}))$ the (convex) subdifferential of S_p taken at $L(\bar{w})$. Moreover, for $w \in \mathbb{R}^d$, compute $L(w) \in \mathbb{R}^n$ and $Q_p(L(w)) \in \mathbb{R}$. Consider $I_>$ the set of indices such that $L_i(w) > Q_p(L(w))$ and $I_=$ the set of indices such that $L_i(w) = Q_p(L(w))$. Then the subdifferential of f at w can be written with the gradients $\nabla L_i(w)$ for $i \in I_> \cup I_=$, as follows

$$\partial f(w) \;=\; \frac{1}{n(1-p)} \sum_{i \in I_>} \nabla L_i(w) \;+\; \frac{\delta}{1-p}\, \mathrm{conv}\left\{\nabla L_i(w) : i \in I_=\right\}. \tag{4.4}$$

Proof. We apply the chain rule of [140, 10.6] to the composition $S_p \circ L$: we have that S_p is convex with full domain, which implies that the two subdifferentials[1] of f coincide (i.e., f is regular in the terminology of [140]) and we have (4.3).

Since S_p is the support function of the set Δ_p, standard subdifferential calculus [61, Cor. 4.4.4] gives that $\partial S_p(L(w))$ is the set of optimal solutions of (3.11) with $L(w) = (L_i(w))_{1 \le i \le n}$. Knowing $I_>$ and $I_=$, the so-called fractional knapsack problem (3.11) can be solved by the simple greedy strategy [35] of taking the largest q_i for $i \in I_>$ and completing to 1 with the q_i for $i \in I_=$. Thus

$$q \text{ solution of (3.11)} \iff \begin{cases} q_i = \frac{1}{n(1-p)} & \text{if } i \in I_> \\ 0 \le q_i \le \frac{1}{n(1-p)} & \text{if } i \in I_= \text{ s.t. } \sum_{i \in I_=} q_i = \frac{\delta}{1-p} \\ q_i = 0 & \text{otherwise.} \end{cases}$$

By (4.3), this gives:

$$\partial f(w) = \frac{1}{n(1-p)} \sum_{i \in I_>} \nabla L_i(w) + \left\{ \sum_{i \in I_=} q_i \nabla L_i(w),\ \text{s.t.} \ \left\{ \begin{array}{l} 0 \le q_i\ \forall i \in I_= \\ \sum_{i \in I_=} q_i = \frac{\delta}{1-p} \end{array} \right\} \right.$$

Finally, introducing weights $\alpha_i = \frac{q_i(1-p)}{\delta}$ for $i \in I_=$, the right-hand term can be written as the convex hull of $\nabla L_i(w)$ for $i \in I_=$, which gives the expression. □

We observe that the expression of $\partial f(w)$ does not involve the gradients of all the L_i's, but only of those associated to the largest values. We also see that f is differentiable at w if and only if $I_=$ is reduced to a singleton. The objective function is not differentiable in general, which poses a problem for a direct application of machine-learning gradient-based algorithms.

1 Remark on the Clarke subdifferential: As another by-product of the chain rule [140, 10.6], the set of horizon subgradients of f is reduced to 0 since so is the one of S_p (convex and defined on $\mathbb{R}^n$). As a consequence, the Clarke subdifferential is the convex hull of the limiting subdifferential [140, 8.49]. Thus we have, in our case, that the three subdifferentials (regular, limiting and Clarke) coincide.

4.2.2 *Efficient Smoothing*

In this paragraph, we study a smoothing of nonsmooth superquantile-based functions (4.2). We propose to use the infimal convolution smoothing of [117]; the comparison to other smoothing approaches is postponed to the next section. We follow the guidelines of [8] : we smooth only the superquantile S_p rather than the whole function f. Thus we consider

$$f_\nu(w) = S_{p,\nu}(L(w)) \quad \text{for } S_{p,\nu} \text{ a smooth approximation of } S_p. \tag{4.5}$$

Inf-convolution of the superquantile

Regularizing the dual representation (3.11) of superquantile, we consider the function, parameterized by the smoothing parameter ν,

$$S_{p,\nu}(x) = \max_{q \in \Delta_p} \left\{ q^\top x - \nu D(q) \right\}, \forall x \in \mathbb{R}^n, \tag{4.6}$$

for a given strongly convex function D. The following proposition establishes that the resulting function f_ν as (4.5) is a smooth approximation of f, as a direct application of e.g., [8, Theorem 4.1, Lemma 4.2], or [117, Theorem 1].

Smoothed approximation

Proposition 4.2. *In the above setting, the function f_ν provides a global approximation of f, i.e.*

$$f_\nu(w) \le f(w) \le f_\nu(w) + \frac{\nu}{2} \qquad \text{for all } w \in \mathbb{R}^d.$$

Moreover $S_{p,\nu}$ is differentiable, with $\nabla S_{p,\nu}(x)$ being the argmax of (4.6), unique by strong convexity of D. When L is differentiable, f_ν is differentiable as well, with

$$\nabla f_\nu(w) = \nabla L(w)^* \nabla S_{p,\nu}(L(w)). \tag{4.7}$$

Strongly convex regularization

In our quest for simple and implementable expressions, we study in the rest of this section the case of separable strongly functions of the form:

$$D(q) = \sum_{i=1}^n d(q_i) \qquad \text{given a strongly convex function } d : [0,1] \to \mathbb{R}. \tag{4.8}$$

We provide in Corollary 4.4 a general scheme to compute the gradient with explicit expressions in Examples 4.1 and 4.2 for special choices of d. Finally we discuss the role of the smoothing parameter ν on a numerical illustration.

A special conjugate function

We start with a lemma gathering the nice duality properties of (4.6). A one-dimensional convex function plays a special role: it is the convex conjugate of the sum of νd and the indicator of the segment $\left[0, 1/n(1-p)\right]$

$$g_\nu(s) = \left(\nu d + i_{\left[0, \frac{1}{n(1-p)}\right]} \right)^* (s) = \max_{0 \le t \le \frac{1}{n(1-p)}} \left\{ s\, t - \nu\, d(t) \right\}. \tag{4.9}$$

Since d is strongly convex, standard (one-dimensional) convex analysis gives (see e.g., [61, Prop.I.6.2.2]) that g_ν is continuously differentiable with derivative $g'_\nu(s)$ being the (unique) t achieving the above max. Simple calculus yields

$$g'_\nu(s) = \begin{cases} 0 & \text{if } s \le \nu\, d'_+(0) \\ \frac{1}{n(1-p)} & \text{if } s \ge \nu\, d'_-(1/(n(1-p))) \\ (d^*)'\left(\frac{s}{\nu}\right) & \text{otherwise.} \end{cases} \tag{4.10}$$

where $d'_+(0) \in [-\infty, +\infty)$ and $d'_-(1/(n(1-p))) \in [-\infty, +\infty)$ are respectively the right-derivative of d at 0 and the left-derivative of d at $1/(n(1-p))$. Note finally that $g_\nu{}'$ is a non-decreasing function.

Lemma 4.3. *The dual problem of (4.6) can be expressed as the (smooth convex) one-dimensional problem:* *Duality*

$$\min_{\eta} \quad \theta(\eta) = \eta + \sum_{i=1}^{n} g_\nu(x_i - \eta). \tag{4.11}$$

Moreover, there is no duality gap between (4.6) and (4.11). There exists a primal-dual solution $(q_\nu^\star, \eta^\star)$ and the unique primal solution can be written $q_\nu^\star = (g'_\nu(x_i - \eta^\star))_{i=1,\dots,n}$ with the help of (4.10).

Proof. This lemma could be proved by applying a sequence of results from abstract Lagrangian duality [61, Chap. XII]. Instead, we provide a simple proof from the direct calculus developed so far. Consider the dualization of the constraint $\sum_{i=1}^n q_i - 1 = 0$ in Δ_p. For a primal variable $q \in B_p = \left[0, \frac{1}{n(1-p)}\right]^n$ and a dual variable $\eta \in \mathbb{R}$, we write the Lagrangian

$$L(q, \eta) = \sum_{i=1}^{n} q_i x_i - \nu d_i(q_i) - \eta\Big(\sum_{i=1}^{n} q_i - 1\Big) = \eta + \sum_{i=1}^{n} q_i(x_i - \eta) - \nu d_i(q_i)\,,$$

and the associated dual function

$$\theta(\eta) = \max_{q \in B_p} L(q, \eta) = \eta + \sum_{i=1}^{n} \max_{0 \le q_i \le \frac{1}{n(1-p)}} \left\{ q_i(x_i - \eta) - \nu\, d_i(q_i) \right\},$$

which gives the expression of the dual function (4.11) from (4.9). Note for later that we have, by construction, the so-called weak duality inequality

$$\theta(\eta) \ge L(q, \eta) = \sum_{i=1}^{n} q_i x_i - \nu d_i(q_i) \quad \text{for all } \eta \text{ and all feasible } q \in \Delta_p. \tag{4.12}$$

Now recall that g_ν in (4.9) is differentiable and so is the dual function with

$$\theta'(\eta) = 1 - \sum_{i=1}^{n} g'_\nu(x_i - \eta)\,. \tag{4.13}$$

The above expression also shows that

$$\lim_{\eta \to +\infty} \theta'(\eta) = 1 \qquad \text{and} \qquad \lim_{\eta \to -\infty} \theta'(\eta) = 1 - \sum_{i=1}^{n} \frac{1}{n(1-p)} = \frac{-p}{1-p}.$$

By continuity of g'_ν and θ', this implies that there exists $\eta^\star$ such that $\theta'(\eta^\star) = 0$, i.e., there exists a dual solution $\eta^\star$. On the primal side, the compactness of B_p and strong convexity of d gives existence and uniqueness of the primal solution, denoted $q_\nu^\star$. Observe now that (4.13) means that the vector $(g'_\nu(x_i - \eta^\star))_{i=1,\dots,n}$, which lies in B_p by construction, is in fact primal feasible. From (4.12) and uniqueness of the primal solution, this implies that *Primal solution of (4.6)*

$$q_\nu^\star = (g'_\nu(x_i - \eta^\star))_{i=1,\dots,n} \tag{4.14}$$

and that there is no duality gap. □

From Lemma 4.3, we get an almost explicit expressions of values and gradients of the smooth approximation f_ν.

Oracle for smooth approximation

Corollary 4.4. *Consider f_ν defined by (4.5) with L differentiable. With $\eta^\star$ an optimal solution of (4.11) with $x_i = L_i(w)$,*

$$f_\nu(w) = \eta^\star + \sum_{i=1}^n g_\nu(L_i(w) - \eta^\star)\,, \tag{4.15}$$

$$\nabla f_\nu(w) = \sum_{i=1}^n g'_\nu(L_i(w) - \eta^\star)\,\nabla L_i(w) \tag{4.16}$$

where g_ν and g'_ν are given by (4.9) and (4.10).

Proof. The no-gap result of Lemma 4.3 gives that $S_{p,\nu}(x)$ is equal to the optimal value of (4.11). This gives directly the above expression of $f_\nu(w) = S_{p,\nu}(L(w))$ with $\eta^\star$ an optimal solution of (4.11) with $x_i = L_i(w)$. Regarding the expression of the gradient, Proposition 4.2 states that $\nabla S_{p,\nu}(x)$ is the optimal solution of (4.6), and Lemma 4.3 expresses it as $(g'_\nu(x_i - \eta^\star))_{i=1,\dots,n}$. We then get the expression of $\nabla f_\nu(w)$ from (4.7). □

Thus the computation of the first-order oracle of f_ν boils down to solving the one-dimensional convex problem (4.11) with $x_i = L_i(w)$. This easy task can be done in general by bisection or higher-order schemes. Here Lemma 4.3 allows us to make an additional simplification with an initial interval tightening. We can indeed shrink the segment where to find $\eta^\star$ to two consecutive points in

$$N = \left\{ x_i - \nu\, d'_+(0),\, x_i - \nu\, d'_-\Big(\frac{1}{(n(1-p)}\Big)\;\; i = 1,\dots,n \right\} \tag{4.17}$$

which is a set of special points regarding the structure of the dual function (recall (4.10) and (4.11)). Denoting $\underline{\eta}$ and $\bar{\eta}$, defined respectively as the largest point in N such that $\theta'(\underline{\eta}) \le 0$ and the smallest point in N such that $\theta'(\bar{\eta}) \ge 0$, we get $\eta^\star$ by testing three cases:

- if $\theta'(\underline{\eta}) = 0$, take $\eta^\star = \underline{\eta}$; if $\theta'(\bar{\eta}) = 0$, take $\eta^\star = \bar{\eta}$;
- otherwise, compute $\eta^\star$ in the small interval $[\underline{\eta}, \bar{\eta}]$.

The initial interval tightening thus boils down to having sorted points in N, which is obtained directly from sorting the given data.

Finally we emphasize that we can sometimes go one step further ahead and obtain explicit expressions of $\eta^\star$ and thus, readily implementable expressions of $\nabla f_\nu(w)$. In the next two examples, we illustrate this for two cases of interest, when we smooth the superquantile by a divergence to the uniform probability (which is at the center of Δ_p; recall Figure 3.3). In particular the smoothing detailed in the forthcoming Example 4.1 was used in the numerical illustrations of Examples 2.1, 2.2, and 2.3 (where the resulting smoothed superquantile optimization problems were solved by L-BFGS).

Euclidean smoothing

Example 4.1. We suggest to smooth the superquantile with the Euclidean distance to the uniform distribution

$$D(q) = \frac{1}{2}\|q - \bar{q}\|^2 \quad \text{with} \quad \bar{q} = \left(\frac{1}{n}, \dots, \frac{1}{n}\right), \tag{4.18}$$

which consists in taking in (4.8)

$$d(t) = \frac{1}{2}\left(t - \frac{1}{n}\right)^2.$$

In this case, elementary calculus gives

$$d'_-(0) = -\frac{1}{n}, \quad d'_+\left(\frac{1}{n(1-p)}\right) = \frac{p}{n(1-p)}, \quad \text{and} \quad (d^*)'\left(\frac{t}{\nu}\right) = \frac{t}{\nu} + \frac{1}{n}$$

so that we get from (4.10) the following expression

$$g'_\nu(x_i - \eta) = \begin{cases} 0 & \text{if } \eta \geq x_i + \frac{\nu}{n} \\ \frac{1}{n(1-p)} & \text{if } \eta \leq x_i - \frac{\nu}{n}\frac{p}{1-p} \\ \frac{x_i - \eta}{\nu} + \frac{1}{n} & \text{otherwise.} \end{cases} \tag{4.19}$$

We also have that θ' is piecewise linear in this case and that

$$N = \left\{ x_i + \frac{\nu}{n}, x_i - \frac{\nu}{n}\frac{p}{1-p} \;\; i = 1, \ldots, n \right\}.$$

Therefore from $\underline{\eta}$ and $\bar{\eta}$ in N, finding $\eta^\star$ in the interval $[\underline{\eta}, \bar{\eta}]$ simply reduces to interpolating linearly as *Linear interpolation*

$$\eta^\star = \underline{\eta} - \frac{\theta'(\underline{\eta})(\bar{\eta} - \underline{\eta})}{\theta'(\bar{\eta}) - \theta'(\underline{\eta})}. \tag{4.20}$$

We can apply Corollary 4.4 to get an efficiently implemented expression of the gradient. Note that the obtained expression of $\nabla f_\nu(w)$ involves only the gradients $\nabla L_i(w)$ for largest values of $L_i(w)$ (comparable to the expression of $\partial L(w)$ in Proposition 4.1). The overall procedure is laid down in 1 □

Example 4.2. We use here the Kullback-Lieber divergence to the uniform probability *Entropic smoothing*

$$D(q) = \sum_{i=1}^n q_i \log(q_i/\bar{q}_i) \quad \text{with} \quad \bar{q} = \left(\frac{1}{n}, \ldots, \frac{1}{n}\right).$$

which consists in taking $d(t) = t\log(t)$ in (4.8). Elementary calculus then gives

$$d'_+(0) = -\infty, \quad d'_-\left(\frac{1}{n(1-p)}\right) = 1 - \log(n(1-p)), \quad \text{and} \quad (d^*)'\left(\frac{t}{\nu}\right) = \exp\left(\frac{t}{\nu} - 1\right)$$

which in turn yields

$$g'_\nu(x_i - \eta) = \begin{cases} \frac{1}{n(1-p)} & \text{if } \eta \leq x_i + \nu\left(\log(n(1-p)) - 1\right) \\ \exp\left(\frac{x_i - \eta}{\nu} - 1\right) & \text{otherwise} \end{cases}$$
$$N = \left\{ x_i + \nu\left(\log(n(1-p)) - 1\right) \;\; i = 1, \ldots, n \right\}.$$

On the interval $[\underline{\eta}, \bar{\eta}]$, we have that

$$\theta'(\eta) = 1 - \sum_{i \in I} \frac{1}{n(1-p)} - \sum_{i \notin I} \exp\left(\frac{x_i - \eta}{\nu} - 1\right)$$

Algorithm 1: Fast subroutine for smoothed oracle in the euclidean setting

Initialization: $e = (1, \dots, 1)^\top$, $x = L(w)$, $\ell = \frac{1}{n(1-p)}$, $q_\mu = 0 \in \mathbb{R}^n$

Find in the points of non-differentiability $\mathcal{P}$, a and b such that,
$\mathcal{P} := \{x_i + \frac{\nu}{n}, x_i - \frac{\nu p}{n(1-p)}, i \in \{1, \dots, n\}\}$
$a := \max\{s \in \mathcal{P}, \theta'(s) \leq 0\}$
$b := \min\{s \in \mathcal{P}, \theta'(s) > 0\}$;
Set the dual optimal solution $\eta^\star := a - \frac{\theta'(a)(b-a)}{\theta'(b)-\theta'(a)}$;
Construct the primal solution component-wise:
for $1 \leq k \leq n$ **do**
  if $\eta^\star < x_k - \frac{\nu p}{n(1-p)}$ **then**
    $[q_\nu]_k = \frac{1}{n(1-p)}$;
  else if $x_k - \frac{\nu p}{n(1-p)} \leq \eta^\star < x_k + \frac{\nu}{n}$ **then**
    $[q_\nu]_k = \frac{x_k - \eta^\star}{\nu} + \frac{1}{n}$;
  else
    $[q_\nu]_k = 0$
  end
end
Output: $q_\nu \in \mathbb{R}^n$: solution of (4.6)

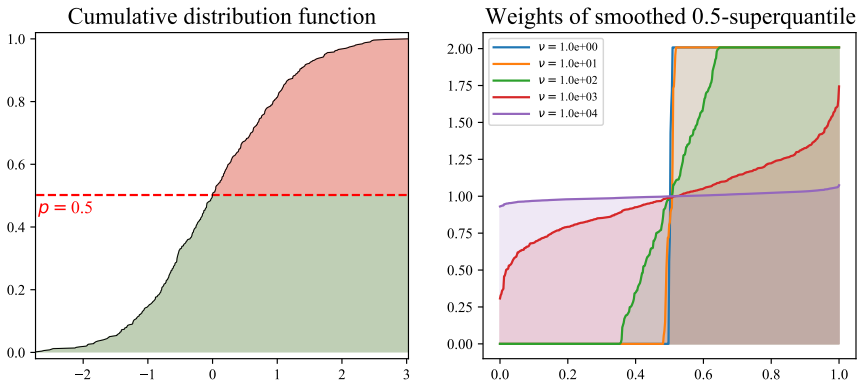


Figure 4.1: Impact of the smoothing parameter ν on the relative weighing between data points. **Left:** empirical cumulative distribution of $n = 500$ points sampled from a standard Gaussian distribution. **Right:** distribution of weights, i.e., the optimal solution of (4.6) for $p = 0.5$, with respect to sorted data points (i.e., value at abscissa t is the weight attached to the t-quantile). Different colours correspond to different values of ν.

Entropic interpolation

with $I = \{i, x_i + \nu\left(\log(n(1-p)) - 1\right) \leq \underline{\eta}\}$ the set of indices of points in N smaller than $\underline{\eta}$. This yields

$$\eta^\star = \nu \log\left(\frac{\sum_{i \notin I} \exp(x_i/\nu - 1)}{1 - |I|/\left(n(1-p)\right)}\right) \tag{4.21}$$

We can then apply Corollary 4.4 to get the smoothed gradient. □

We conclude this section on the infimal-smoothing of the superquantile with two remarks illustrating the impact of the smoothing parameter ν.

Impact of the smoothing parameter on the weights

Remark 4.1. We illustrate the impact of the smoothing parameter ν on the relative weights given to the data. We consider the Euclidean smoothing of Example 4.1 with $p = 0.5$; we sample $n = 500$ points from a Gaussian distribution; and

we compute, for different values of ν, the distribution of weights q_i of (4.19), solutions to smoothed problem (4.6). The right-hand side of Figure 4.1 displays the impact of ν of the obtained weights. In particular, we note that as ν grows, the distribution q_i tends to spread uniformly over all data-points, so that the smoothed superquantile acts like the expectation. In contrast, when ν is close to 0, the distribution approximates the uniform distribition over the interval $[p, 1]$, so that the smoothed superquantile acts like the superquantile. This approximation is further discussed in the next remark. □

Impact of the smoothing parameter on the approximation

Remark 4.2. We briefly illustrate here the impact of the smoothing parameter ν: we fix a vector $\bar{w}$ and we observe the values of smoothed approximations of a superquantile-based function for different values of ν. More precisely, we consider a logistic regression problem on the Australian credit dataset from the UCI ML repository. We use the quadratic smoothing of Example 4.1 with $\nu = 0.1$; and we solve the problem by L-BFGS to get the reference point $\bar{w}$. Then we compute, at this point, the values of:

- the underlying superquantile-based objective (4.2) which corresponds to the case $\nu = 0$;
- the smoothed approximations (which corresponds to (4.5) with $S_{p,\nu}$ replacing S_p) for a sequence of ν evenly spread on a log scale;
- the usual empirical risk minimization objective, which corresponds to the case $\nu = +\infty$. Indeed, in this regime $\nu \to +\infty$, the impact of the quadratic penalization term $(q - \bar{q})$ increases so that the solution of (4.6) eventually becomes the uniform distribution $\bar{q}$, in which case $S_{p,\nu}$ coincides with the expectation.

We observe on Figure 4.2 what is expected: for small values of ν, the difference between the superquantile-based objective and its smooth approximations vanishes; for large values of ν, the smoothed superquantile loss tends to the average loss and does not approximate the nonsmooth superquantile loss well.

A key benefit of smoothing the superquantile is to leverage efficient smooth optimization algorithms, such as L-BFGS, for superquantile learning. When ν is too small, the problem is almost nonsmooth, which leads to numerical issues with convergence (on this instance, L-BGFS fails to converge when ν too small or when used with the nonsmooth oracle of Proposition 4.1 due to a line search failure). When ν is too large, the smoothed superquantile gets close to the expectation and the interest of using a superquantile approach disappears. This illustrates the interest of having a moderate ν for superquantile learning, where the smoothed objective is an reasonable approximation of the nonsmooth superquantile, while still being smooth enough to leverage fast optimization algorithms. □

4.2.3 *Comparison to other smoothing schemes*

We compare the proposed infimal convolution smoothing of the superquantile (4.6) to other possible smoothing schemes. Classical smoothing techniques are based either on convolution or infimal convolution. For superquantile, one could either smooth the dual representation (3.11) or the variational representation (3.12). Together, this yields four natural ways to smooth the superquantile.

We first formalize the equivalence between the two infimal convolution smoothings: indeed, smoothing the dual representation considered in the

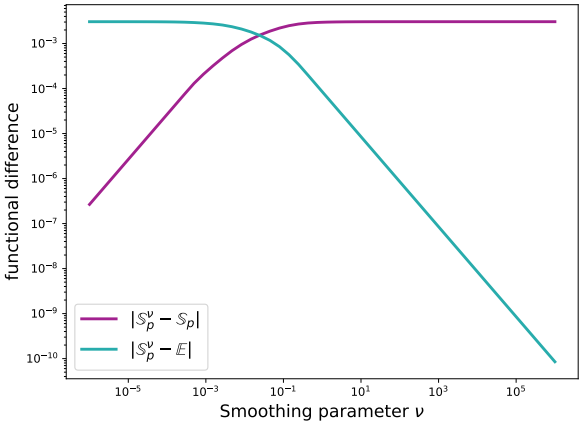


Figure 4.2: Impact of the smoothing parameter ν solving a superquantile logistic regression on a classical dataset (Australian Credit dataset).

preceding section corresponds to a smoothing of $\max\{\cdot, 0\}$ in the variational formulation.

Equivalence of smoothings with infimal convolution

Corollary 4.5. *The infimal convolution smoothing of S_p with a separable strongly convex function (4.8) is equivalent to the infimal convolution smoothing of the positive part $\max\{\cdot, 0\}$ as*

$$m_\nu(\eta) = \max_{0 \le t \le 1} \left\{ \eta\, t - \nu\, \tilde{d}(t) \right\} \qquad \text{with } \tilde{d}(t) = n(1-p)d\left(\frac{t}{n(1-p)} \right). \tag{4.22}$$

More precisely, we have the following equality (to be compared with (3.12))

$$S_{p,\nu}(x) = \min_\eta \left\{ \eta + \frac{1}{n(1-p)} \sum_{i=1}^n m_\nu(x_i - \eta) \right\}.$$

Proof. A direct change of variable in (4.9) gives $g_\nu(x_i - \eta) = \frac{1}{n(1-p)} m_\nu(x_i - \eta)$. The proof is direct from the expression of the dual problem (4.11) and the no-gap result stated in Lemma 4.3. □

Smoothing by convolution

Next, we show an equivalence between the smoothing by infimal convolution (4.22), and by convolution, as considered in [26, 106]. Given a continuous probability density $h\colon \mathbb{R} \to \mathbb{R}_+$, (such that $\int_{-\infty}^{\infty} |s| h(s) \mathrm{d}s$ is finite) the smoothing by convolution of the function $\max\{\cdot, 0\}$ with smoothing parameter $\nu > 0$ is defined by[2]

$$\bar{m}_\nu(\eta) = \frac{1}{\nu} \int_{-\infty}^{\infty} \max\{\eta - s, 0\} h\left(\tfrac{s}{\nu}\right) \mathrm{d}s = \frac{1}{\nu} \int_{-\infty}^{\eta} (\eta - s) h\left(\tfrac{s}{\nu}\right) \mathrm{d}s\,. \tag{4.23}$$

The function $\bar{m}_\nu$ is convex and smooth, with derivative

$$\bar{m}'_\nu(\eta) = \frac{1}{\nu} \int_{-\infty}^{\eta} h\left(\tfrac{s}{\nu}\right) \mathrm{d}s\,. \tag{4.24}$$

2 Applied to $\max\{x, \cdot\}$, the general smoothing by convolution as defined in (4.23) coincides with the double integral representation used in [26, 106]. Indeed, integrating (4.24) yields

$$\bar{m}_\nu(\eta) = \tfrac{1}{\nu} \int_{-\infty}^{\eta} \int_{-\infty}^{\eta'} h\left(\tfrac{s}{\nu}\right) \mathrm{d}s\, \mathrm{d}\eta'\,.$$

The next proposition, relating this smoothing to the previous one, involves $Q_t(h)$ the quantile function of a random variable with density h.

Equivalence of convolution/inf-convolution smoothings

Proposition 4.6. *With the above notation, the convolution smoothing $\bar{m}_\nu$ of (4.23) for $\nu = 1$ can be written as the infimal-convolution smoothing (to be compared with (4.22))*

$$\bar{m}_1(\eta) = \max_{0\le t\le 1} \left\{\eta\, t - \bar{d}(t)\right\} \quad \text{where} \quad \bar{d}(t) = tQ_t(h) - \bar{m}_1(Q_t(h)). \tag{4.25}$$

Conversely, the infimal convolution smoothing m_ν of (4.22) for $\nu = 1$ can be written as the convolution smoothing (to be compared with (4.23))

$$m_1(\eta) = \lim_{s\to-\infty} m_1(s) + \int_{-\infty}^{\eta} (\eta - s)\tilde{h}(s)\mathrm{d}s \quad \text{where} \quad \tilde{h}(s) = m_1''(s) \text{ a.e.} \tag{4.26}$$

Proof. For the first part, we consider the convex conjugate of $\bar{m}_1$

$$\bar{m}_1^*(t) = \sup_{\eta\in\mathbb{R}} \left\{\eta\, t - \bar{m}_1(\eta)\right\} .$$

If $t \notin [0,1]$, the supremum is $+\infty$ since $|\bar{m}_1(\eta) - \max\{\eta, 0\}|$ is bounded by an absolute constant. For $t \in [0,1]$, the concave function $\eta \mapsto \eta t - \bar{m}_1(\eta)$ is maximized at $\eta^\star$ if and only if it satisfies the first-order optimality condition

$$t = \bar{m}_1'(\eta^\star) = \int_{-\infty}^{\eta^\star} h(s)\mathrm{d}s.$$

Since the latter is the cumulative distribution function, $\eta^\star = Q_t(h)$ is the corresponding quantile function (well-defined since h is continuous). This yields

$$\bar{m}_1^* = \bar{d} + i_{[0,1]}, \tag{4.27}$$

which in turn gives (4.25). Finally to establish the strong convexity of $\bar{d}$, we use again (4.27) together with the smoothness of $\bar{m}_1$. Thus $\bar{m}_1$ corresponds to the infimal-convolution smoothing with $\bar{d}$.

For the second part, we start by noting that since m_1' is Lipschitz, m_1'' exists almost everywhere, and $\tilde{h}$ is well-defined. Since m_1 is convex, it also holds that $m_1''(s) \ge 0$, and then that we have the normalization

$$\int_{-\infty}^{\infty} \tilde{h}(s)\mathrm{d}s = \int_{-\infty}^{\infty} \tilde{m}_1''(s)\mathrm{d}s = \lim_{\eta\to\infty} m_1'(\eta) - \lim_{\eta\to-\infty} m_1'(\eta) = 1 - 0 = 1\,,$$

where we use $m'(\eta)$ is the (unique) optimal solution of (4.22). Then the proof follows from the next two claims.

Claim 1: m_1 admits a limit at $-\infty$. Convexity of m_1 gives that m_1' is non-decreasing. Since $\lim_{s\to-\infty} m_1'(s) = 0$, we get that m_1' is non-negative. Thus, m_1 is non-decreasing and, since it is bounded from below, this implies that m_1 admits a limit at $-\infty$ (that we denote $m_1(-\infty)$).

Claim 2: $\lim_{s\to-\infty} s\, m_1'(s) = 0$. For a given s, we write:

$$s\; m_1'(2s) \le \int_{2s}^{s} m_1'(t)\mathrm{d}t = m_1(s) - m_1(2s),$$

where the inequality comes from the fact that m_1' is non-decreasing. Using that m_1 admits a limit at $-\infty$ (Claim 1), we then get Claim 2.

Finally, we can conclude the proof with integration by parts:

$$\begin{aligned} m_1(\eta) &= m_1(-\infty) + \int_{-\infty}^{\eta} m_1'(s)\mathrm{d}s \\ &= m_1(-\infty) + [(s-\eta)m_1'(s)]_{-\infty}^{\eta} + \int_{-\infty}^{\eta} (\eta - s)\tilde{h}(s)\mathrm{d}s \\ &= m_1(-\infty) + \int_{-\infty}^{\eta} (\eta - s)\tilde{h}(s)\mathrm{d}s. \end{aligned}$$

This establishes (4.26) and ends the proof. □

Finally, we mention the smoothing of the dual representation (3.11) using convolution, which would write:

$$\bar{S}_p^{\nu}(x) = \frac{1}{\nu}\int_{\mathbb{R}^n} S_p(x-z)h\left(\tfrac{z}{\nu}\right)\mathrm{d}z = \mathbb{E}_{Z\sim h}[S_p(x-\nu Z)]\,,$$

for the density $h\colon \mathbb{R}^n \to \mathbb{R}$ and the parameter $\nu > 0$. We do not consider this smoothing approach because it suffers from two drawbacks in view of practical implementation. First, it usually cannot be computed in closed form, unlike the other smoothing approaches considered here. Second, the Lipschitz constant of the gradient (appearing in condition numbers, constant scalings, and rates of convergence of first-order methods [116]) scales badly: as $O(\sqrt{n}/\nu)$ for the Lipschitz constant of $\nabla\bar{S}_p^{\nu}$ [118, Lemma 2], as opposed to the dimension-independent $O(1/\nu)$ for the one of ∇S_p^{ν} [117, Theorem 1].

4.2.4 *Generalization to law-invariant comonotone risk-measures*

In this section, we provide fast smoothing procedures for the class of law-invariant comonotone risk measures. These results generalize the one given in Section 4.2.2 while maintaining (up to a log factor) the linear time complexity of oracle computations. In order to achieve this optimal complexity, we leverage the Kusuoka representation recalled in Section 3.3.2 and extend the dual properties of the superquantile given in Section 4.2.2. The developments of this section are of algorithmic nature. We consider the euclidean smoothing (see Example 4.1) for its specific properties that we highlight thereafter.

SMOOTHING LAW-INVARIANT COMONOTONE RISK MEASURES.

Consider a discrete risk measure $\rho : \mathbb{R}^n \to \mathbb{R}$ as in (3.8) satisfying the law-invariant and comonotone properties. Let $(\beta_k)_{1\le k\le n}$ be the sequence yielding the Kusuoka representation (3.13) of ρ. We propose to smooth ρ by smoothing the sequence of terms $\beta_k S_{\frac{k-1}{n}}(x)$. For a fixed smoothing parameter $\nu > 0$, we thus consider the smooth counterpart ρ_ν defined as:

$$\rho_\nu(x) = \sum_{k=1}^{n} \beta_k S_{\frac{k-1}{n},\nu}(x). \tag{4.28}$$

with $S_{\frac{k-1}{n},\nu}$ the smooth approximation (4.6) of the superquantile provided with the euclidean smoothing from Example 4.1. The following proposition is a direct generalization of Proposition 4.2.

Smoothing of ρ

Proposition 4.7. *For any $\nu > 0$, the function ρ_ν defined in (4.28) is a C^1 approximation*

of ρ*:*

$$|\rho_\nu(x) - \rho(x)| \leq \frac{\nu}{2}\sum_{k=2}^{n}(\alpha_k - \alpha_1). \tag{4.29}$$

where the sequence (α_i) *is as defined in Proposition 3.11.*

Proof. Let $\nu > 0$ be fixed and α be defined as in Proposition (3.11). For any $x \in \mathbb{R}^d$, since Δ_0 is reduced to the singleton $\{1/n, \ldots, 1/n\}$, we have

$$S_{0,\nu}(x) = S_0(x).$$

Thus,

$$\begin{aligned}
|\rho_\nu(x) - \rho(x)| &\leq \left|\sum_{k=2}^{n}\beta_k S_{\frac{k-1}{n},\nu}(x) - S_{\frac{k-1}{n}}(x)\right| \leq \sum_{k=2}^{n}\beta_k\left|S_{\frac{k-1}{n},\nu}(x) - S_{\frac{k-1}{n}}(x)\right| \\
&\leq \frac{\nu}{2}\sum_{k=2}^{n}\beta_k \leq \frac{\nu}{2}\sum_{k=2}^{n}(n+1-k)\,(\alpha_k - \alpha_{k-1}) \\
&\leq \frac{\nu}{2}\sum_{k=2}^{n}(\alpha_k - \alpha_1),
\end{aligned}$$

where the last inequality resulted from an integration by part and the previous one from the non-decrease of the sequence (α_i). Smoothness is a direct consequence of the smoothness of the terms $S_{\frac{k-1}{n},\nu}$ given in Proposition 4.2. □

In particular, we know from proposition 3.5 that if the support $\mathcal{C}$ is included in the simplex, then we necessarily have:

$$|\rho_\nu(x) - \rho(x)| \leq \frac{\nu}{2}.$$

Computation of the smoothed gradient $\nabla\rho_\nu(x)$. Let the smooth parameter $\nu > 0$ be fixed. Let us assume for the moment that we have at our disposal the whole sequence of dual solutions $(\eta_k^\star)_{0\leq k\leq n-1}$ of the problem (4.11) for the values of p in $\left\{\frac{k}{n}, 0 \leq k \leq n-1\right\}$. Then we have a gradient of ρ_ν: *Smoothed gradient of* ρ

$$\nabla\rho_\nu(x) = \sum_{k=1}^{n}\beta_k q^{k-1} \tag{4.30}$$

where q^k denotes the associated primal solution given by (4.14). Let us explicit this expression.

For $k \in [\![0, n-1]\!]$, let us denote θ_k the objective of the dual problem (4.11) with probability value $p = \frac{k}{n}$:

$$\theta_k(\eta) = \eta + \sum_{i=1}^{n} g_{\nu,k}(x_i - \eta)$$

where $g_{\nu,k} : s \mapsto \max_{0\leq t\leq\frac{1}{n(1-\frac{k}{n})}}\{st - \nu d(t)\}$ with $d\colon t \mapsto \frac{1}{2}\left(t - \frac{1}{n}\right)^2$. Recall that by strong convexity of d, the functions $g_{\nu,k}$, $k \in [\![0, n-1]\!]$), are all continuously differentiable and convex. Thus, for all $k \in [\![0, n-1]\!]$, θ_k is continuously differentiable and θ_k' admits everywhere left and right derivatives. We denote

$\theta_k''^{+}(\eta)$ the second-order right-derivative at η. Finally, we also recall from Example 4.1 and the equation (4.10) that $g_{\nu,k}$ has for derivative:

$$g'_{\nu,k}(s) = \begin{cases} 0 & \text{if } s \leq -\frac{\nu}{n} \\ \frac{1}{n-k} & \text{if } s > \frac{\nu k}{n(n-k)} \\ \frac{s}{\nu} + \frac{1}{n} & \text{otherwise.} \end{cases} \tag{4.31}$$

We note that for any $k \in [\![2, n-1]\!]$, we have by (4.19):

$$q^k = \sum_{i=1}^{n} \frac{1}{n-k} \mathbb{1}_{\eta_k^\star < x_i - \frac{\nu k}{n(n-k)}} + \left(\frac{x_i - \eta_k^\star}{\nu} + \frac{1}{n} \right) \mathbb{1}_{x_i - \frac{\nu k}{n(n-k)} \leq \eta_k^\star < x_i + \frac{\nu}{n}} \varepsilon_i \tag{4.32}$$

Decomposition of the smoothed gradient

which yields:

$$\begin{aligned} \nabla \rho_\nu(x) = \sum_{k=1}^{n} \beta_k q^{k-1} = & \underbrace{\sum_{k=1}^{n} \beta_k \sum_{i=1}^{n} \frac{1}{n-(k-1)} \mathbb{1}_{\eta_{k-1}^\star < x_i - \frac{\nu(k-1)}{n-(k-1)}} \varepsilon_i}_{A} \\ & + \underbrace{\sum_{k=1}^{n} \beta_k \left(\sum_{i=1}^{n} \frac{x_i - \eta_{k-1}^\star}{\nu} + \frac{1}{n} \right) \mathbb{1}_{x_i - \frac{\nu(k-1)}{n(n-k+1)} \leq \eta_{k-1}^\star < x_i + \frac{\nu}{n}} \varepsilon_i}_{B} \end{aligned} \tag{4.33}$$

Note that if one computes naively for a given $x \in \mathbb{R}^n$, $\nabla \rho_\nu(x)$ by computing separately the primal solutions q^k from (4.32), it would boil down to a $\mathcal{O}(n^2)$ complexity. That would compromise the quasi-linear time complexity we aim to achieve. In the rest of this section, we show how to compute $\nabla \rho_\nu(x)$ in quasi-linear time via the expression (4.33). More precisely, we show in the following order:

1. how to compute the whole sequence of $(\eta_k^\star)_{0 \leq k \leq n-1}$ in quasi-linear time.
2. how to compute the term A from (4.33) in linear time.
3. how to compute the term B from (4.33) in linear time.

We denote by $(x_{(i)})_{1 \leq i \leq n}$ be the sequence of coordinates of the input vector $x \in \mathbb{R}^n$ ranged in an (arbitrary) non-decreasing order.

CHASING THE DUAL VARIABLE $\eta_k^\star$.

We saw in Lemma 4.3 that finding the minimizer $\eta_k^\star$ of θ_k can be done in linear time. In this paragraph, we show the stronger claim that finding the complete sequence of roots $(\eta_k^\star)_{0 \leq k \leq n-1}$ can be done in at most $\mathcal{O}(n \log n)$ operations. We start with two simple observations formalized as lemmas.

First root

Lemma 4.8. *$\eta_0^\star := x_{(0)} - 1$ is a minimizer of θ_0 and $\theta_k''^{+}(x_{(0)} - 1) = 0$.*

Proof. Noting that for all $s \leq x_{(0)}$ and $j \in [\![1, n]\!]$ we have $x_j - s \geq 0$, we have in view of (4.31)

$$\theta_0'(s) = 1 - \sum_{i=1}^{n} g'_{\nu,0}(x_i - s) = 0.$$

In particular, $\theta_0'(s) = \theta_0''^{+}(s) = 0$ □

Lemma 4.9. *The sequence $(\theta'_k)_{0\leq k\leq n-1}$ is non-increasing on $\mathbb{R}$.*

Non-increase of the sequence $(\theta_k)_{0\leq k\leq n-1}$

Proof. For any $k \in [\![0, n-1]\!]$, we note that

$$\theta'_k(\eta) = 1 - \sum_{i=1}^{n} g'_{\nu,k}(x_i - \eta) \quad , \forall \eta \in \mathbb{R},$$

so it suffices to show the non-decrease of $(g'_{\nu,k})_{0\leq k\leq n-1}$. Let us fix $k \in [\![0, n-2]\!]$. For all $s \leq \frac{\nu k}{n(n-k)}$, we have by (4.31) $g_{\nu,k}(s) = g_{\nu,k}(s)$. If $s \geq \frac{\nu(k+1)}{n(n-k-1)}$, we also have $g_{\nu,k}(s) = \frac{1}{n-k} < \frac{1}{n-k-1} = g_{\nu,k+1}(s)$. Finally, if $s \in [\frac{\nu k}{n(n-k)}, \frac{\nu(k+1)}{n(n-k-1)}]$, we have

$$\begin{aligned} g_{\nu,k}(s) &= \frac{1}{n-k} = \frac{k+n-k}{n(n-k)} = \frac{k}{n(n-k)} + \frac{1}{n} \\ &= \frac{1}{\nu}\frac{\nu k}{n(n-k)} + \frac{1}{n} \leq \frac{s}{\nu} + \frac{1}{n} = g_{\nu,k+1}(s), \end{aligned}$$

which ends the proof. □

Running through θ'_k. Let $k \in [\![0, n-1]\!]$ be fixed and $\eta \in \mathbb{R}$ be such that $\theta'_k(\eta) \leq 0$. As a specificity of the Euclidean prox function d, we obtain that θ'_k is a piece-wise affine non-decreasing function. The discontinuity points of θ''_k all belong to the set

Set of discontinuity points N_k

$$N_k := \left\{ x_{(i)} - \frac{\nu k}{n(n-k)},\ x_{(i)} + \frac{\nu}{n}, i \in [\![1, n]\!] \right\}.$$

Assume for now that we know the values of $\theta'_k(\eta)$, $\theta''^{+}_k(\eta)$ and the two following indices:

$$\begin{aligned} i^-(\eta, k) &:= \begin{cases} \min\{i,\ x_{(i)} + \frac{\nu}{n} > \eta\} & \text{if } \eta < x_{(n)} + \frac{\nu}{n} \\ n+1 & \text{otherwise,} \end{cases} \\ i^+(\eta, k) &:= \begin{cases} \min\{i,\ x_{(i)} - \frac{\nu k}{n(n-k)} > \eta\} & \text{if } \eta < x_{(n)} - \frac{\nu k}{n(n-k)} \\ n+1 & \text{otherwise.} \end{cases} \end{aligned} \tag{4.34}$$

Intuitively, $i^-(\eta, k)$ and $i^+(\eta, k)$ are the indices of the first points in N_k that are strictly greater than η.

Let us then set λ to be the first point in N_k that is strictly greater than η[3]

$$\lambda := \min_i N_k \cap (\eta, \infty) = \min \left\{ x_{(i^-(\eta,k))} + \frac{\nu}{n}, x_{(i^+(\eta,k))} - \frac{\nu k}{n(n-k)} \right\}. \tag{4.35}$$

3 It is well defined since (4.31) ensures $\theta'_k(x_{(n)} + \frac{\nu}{n}) = 1 - \sum_{i=1}^{n} g'_{\nu,k}(x_{(i)} - x_{(n)} - \frac{\nu}{n}) = 1 > \theta(\eta)$ ensures that $x_{(n)} + \frac{\nu}{n} > \eta$ by non-decrease of θ'_k.

One can get efficiently as follows the quantities $\theta'_k(\lambda)$, $\theta''^+_k(\lambda)$, $i^+(\lambda,k)$ and $i^-(\lambda,k)$. First observe that since $\lambda > \eta$ implies $i^-(\lambda,k) \geq i^-(\eta,k)$ and $i^+(\lambda,k) \geq i^+(\eta,k)$, we have:

$$\begin{aligned} i^-(\lambda,k) &= i^-(\eta,k) + \sum_{i=i^-(\eta,k)}^{n} \mathbb{1}_{\lambda = x_{(i)} + \frac{\nu}{n}} \\ \text{and } i^+(\lambda,k) &= i^+(\eta,k) + \sum_{i=i^+(\eta,k)}^{n} \mathbb{1}_{\lambda = x_{(i)} - \frac{\nu k}{n(n-k)}} \,. \end{aligned} \tag{4.36}$$

Then, by definition of λ, the function θ''^+_k has no discontinuity point on the interval $[\eta,\lambda)$. Thus,

$$\theta'_k(\lambda) = \theta'_k(\eta) + (\lambda - \eta)\theta''^+_k(\eta). \tag{4.37}$$

Finally, noting that $(d^*)'' = 1$, simple derivations of the function θ_k yield

$$\theta''^+_k(\lambda) = \frac{1}{\nu} \sum_{i=i^-(\lambda,k)}^{i^+(\lambda,k)} (d^*)'' \left(\frac{x_{(i)} - \lambda}{\nu} \right) = \frac{1}{\nu} \left(i^+(\lambda,k) - i^-(\lambda,k) \right). \tag{4.38}$$

First Complexity remark

Remark 4.3. We observe that the computation of $i^-(\lambda,k)$ requires a single pass on the points $x_{(i)}$ such that $i \in [\![i^-(\eta,k), i^-(\lambda,k)]\!]$. Similarly, the computation of $i^+(\lambda,k)$ requires a single pass on the points $x_{(i)}$ such that $i \in [\![i^+(\eta,k), i^+(\lambda,k)]\!]$. Once these two values are found, according to (4.37) and (4.38), one may compute $\theta'_k(\lambda)$ and $\theta''^+_k(\lambda)$ in constant time. □

From θ'_k to θ'_{k+1}. Assuming now that we know $\theta'_k(\eta)$, $\theta''^+_k(\eta)$, $i^-(\eta,k)$ and $i^+(\eta,k)$ as defined in 4.34, we can compute efficiently these indices at $k+1$ - $i^-(\eta,k+1)$, $i^+(\eta,k+1)$ - as well as $\theta'_{k+1}(\eta)$ as well $\theta''^+_{k+1}(\eta)$.

Indeed, we first note by definition of $i^-(\eta,k+1)$ and since $\frac{k+1}{n(n-k-1)} > \frac{k}{n(n-k)}$ that,

$$\begin{aligned} &i^-(\eta,k+1) = i^-(\eta,k)\,, \\ &i^+(\eta,k+1) = 1 + \sum_{i=1}^{n} \mathbb{1}_{\eta \geq x_i - \frac{\nu(k+1)}{n(n-k-1)}} = i^+(\eta,k) + \sum_{i=i^+(\eta,k)}^{n} \mathbb{1}_{x_{(i)} - \frac{\nu k}{n(n-k)} > \eta \geq x_{(i)} - \frac{\nu(k+1)}{n(n-k-1)}} \,. \end{aligned} \tag{4.39}$$

We also note in view of (4.10) that

$$\begin{aligned} \theta'_k(\eta) &= 1 - \sum_{i=1}^{n} g'_{\nu,k}(x_i - \eta) \\ &= 1 - \sum_{i=1}^{n} \frac{1}{n-k} \mathbb{1}_{\eta < x_i - \frac{\nu k}{n(n-k)}} + (d^*)' \left(\frac{x_i - \eta}{\nu} \right) \mathbb{1}_{x_i - \frac{\nu k}{n(n-k)} \leq \eta < x_i + \frac{\nu}{n}} \\ &= 1 - \sum_{i=i^-(\eta,k)}^{i^+(\eta,k)-1} (d^*)' \left(\frac{x_{(i)} - \eta}{\nu} \right) - \sum_{i \geq i^+(\eta,k)} \frac{1}{n-k}. \end{aligned}$$

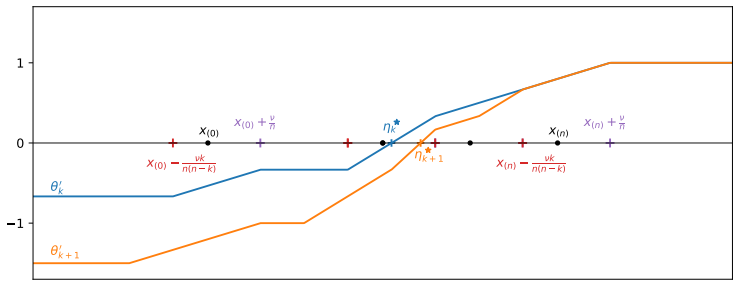


Figure 4.3: Illustration of the roots of the dual functions θ_k. Once the x_i's are sorted, a single pass over them suffices to compute the whole sequence $(\eta_k^\star)_{0\le k\le n-1}$

Since $i^-(\eta,k) = i^-(\eta,k+1)$ and $i^+(\eta,k+1) \ge i^+(\eta,k)$, we obtain

$$\begin{aligned}\theta'_k(\eta) - \theta'_{k+1}(\eta) = &\sum_{i=i^+(\eta,k)}^{i^+(\eta,k+1)-1} \left(\frac{x_{(i)}-\eta}{\nu} + \frac{1}{n} - \frac{1}{n-k}\right) \\ &+ \left(n+1-i^+(\eta,k+1)\right)\left(\frac{1}{n-(k+1)} - \frac{1}{n-k}\right).\end{aligned} \tag{4.40}$$

Finally, we also have by (4.38):

$$\theta''^+_{k+1}(\eta) = \frac{1}{\nu}\left(i^+(\eta,k+1) - i^-(\eta,k+1)\right). \tag{4.41}$$

Second Complexity remark

Remark 4.4. We observe that the computation of $i^-(\eta,k+1)$ is immediate and the computation of $i^+(\eta,k+1)$ requires a single pass on the points $x_{(i)}$ such that $i \in [\![i^+(\eta,k), i^+(\eta,k+1)]\!]$. Similarly, the computation of $\theta'_{k+1}(\eta)$ requires a single pass on the points $x_{(i)}$ such that $i \in [\![i^+(\eta,k), i^+(\eta,k+1)]\!]$. Finally, still in view of (4.38), one may compute $\theta''^+_k(\lambda)$ in constant time. □

Getting the whole sequence $(\eta_k^\star)_{0\le k\le n-1}$ in quasi-linear time. Previous observations offer a guideline for the computation of the sequence $(\eta_k^\star)_{0\le k\le n-1}$. We explain it here, we detail it Algorithm 2, and we illustrate it in Figure 4.3.

Getting $(\eta_k^\star)_{0\le k\le n-1}$ in $\mathcal{O}(n\log n)$ operations

Theorem 4.10. *Assuming the data points $(x_{(i)})_{1\le i\le n}$ are sorted, Algorithm 2 computes the sequence of dual solutions $(\eta_k^\star)_{0\le k\le n-1}$ in quasi-linear time.*

We begin with several intermediate lemmas to bound the complexity of a single iteration k of our Algorithm 2.[4] We first note that for each rank $k \in [\![0, n-1]\!]$, running lines 2 to 3 clearly requires at most 2 operations.

Let us now bound the complexity for running lines 5 to 16. We introduce for $k \in [\![1, n-1]\!]$, the sets

$$\begin{aligned} M_k &= (\eta_{k-1}^\star, \eta_k^\star) \cap N_k \\ I_k^+ &= \left\{ i \in [\![1,n]\!], x_{(i)} - \frac{\nu k}{n(n-k)} \in M_k \right\} \\ I_k^- &= \left\{ i \in [\![1,n]\!], x_{(i)} + \frac{\nu}{n} \in M_k \right\} \end{aligned}$$

4 We count as unitary operations any elementary operation between two float numbers (addition, subtraction, multiplication, division, equality, inequality) and the reading or writing of a float variable.

The sets I_k do not intersect

Lemma 4.11. *For any $k, k' \in [\![1, n-1]\!]$ such that $k \neq k'$, we have:*

$$I_k^- \cap I_{k'}^- = \emptyset \quad \text{and} \quad I_k^+ \cap I_{k'}^+ = \emptyset.$$

Proof. In view of Lemma 4.9 the sequence $(\eta_k^\star)_{0\leq k\leq n-1}$ is non-decreasing. Hence, for any $k, k' \in [\![1, n-1]\!]$ such that $k \neq k'$, we have

$$M_k \cap M_{k'} = \emptyset.$$

Thus, it is clear that the sets $I_k^-, k \in [\![1, n-2]\!]$ do not intersect each other. Let us fix now $k \in [\![1, n-2]\!]$ and $i \in [\![1, n]\!]$. If $i \in I_k^+$, then since the sequence $(\nu k/(n(n-k)))_{0\leq k\leq n-1}$ is increasing, we have

$$x_{(i)} - \frac{\nu(k+1)}{n(n-k-1)} < x_{(i)} - \frac{\nu k}{n(n-k)} < \eta_k^\star$$

so $i < \min I_{k+1}^+$. Reciprocally, if $i \in I_{k+1}^+$, then we still have

$$x_{(i)} - \frac{\nu k}{n(n-k)} > x_{(i)} - \frac{\nu(k+1)}{n(n-k-1)} > \eta_k^\star$$

so $i > \max I_k^+$. This also implies, that the sets $I_k^+, k \in [\![1, n-2]\!]$ do not intersect each other. □

We can bound then the number of operations required to run the lines 5 to 16 in Algorithm 2.

Upper bound on the complexity of lines 5 to 16

Lemma 4.12. *There exists a constant $\mathcal{C}_1$ such that for all $k \in [\![1, n-1]\!]$, the number D_k of operations required to run the lines 5 to 16 in Algorithm 2 satisfies*

$$D_k \leq \mathcal{C}_1 \, (\mathrm{card}(M_k) + 1). \tag{4.42}$$

Proof. Observe that during the execution of the lines 5 to 16, we visit only points in M_k and the first point in N_k (according to the sorting of the x_i established) that is greater or equal to $\eta_k^\star$. Using the sorting of the x_i's previously computed we can visit these points a single time in an increasing order if we rely on the indices $i^-(\cdot, k)$ and $i^+(\cdot, k)$ and formula (4.35) during this visit. The price of visiting each point is constant: it is either a single equality test (which happens in (4.36) when several points in M_k are equal to λ), or a finite number of operations, in view of formulas (4.37), (4.38) and the inequality tests in lines 7, 10 and 13. Letting $\mathcal{C}_1$ be an upper bound of this finite number, which is independant of k, we obtain (4.42). □

We note that for each rank $k \in [\![0, n-1]\!]$, running lines 17 to 24 clearly requires at most constant number of operations, independant of k, that we will denote $\mathcal{C}_2$. Finally we can upper-bound the number of operations required to run lines 26 to 29 with the following lemma. Let us now introduce for $k \in [\![0, n-2]\!]$,

$$\begin{aligned} J_k &= [\![i^+(\eta_k^\star, k) i^+(\eta_k^\star, k+1) - 1]\!] \\ &= \left\{ i \in [\![1, n]\!], x_{(i)} - \frac{\nu k}{n(n-k)} > \eta \geq x_{(i)} - \frac{\nu(k+1)}{n(n-k-1)} \right\} \end{aligned}$$

The sets J_k do not intersect

Lemma 4.13. *For any $k, k' \in [\![0, n-1]\!]$ such that $k \neq k'$, we have $J_k \cap J_{k'} = \emptyset$.*

Algorithm 2: Computation of the sequence $(\eta_k^\star)_{0\le k\le n-1}$ in $\mathcal{O}(n\log n)$

Require : $(x_{(i)})_{1\le i\le n}$: sorted sequence of coordinates of x
Initialize: $\eta := x_{(0)} - 1$, $k := 0$ $i^-(\eta,0) := 1$, $i^+(\eta,0) = 1$, $\theta_0'(\eta) = 0$, $\theta_0''^+(\eta) = 0$;

while $k \le n-1$ **do**
  if $\theta_k'(\eta) = 0$ **then**
    Store $\eta_k^\star = \eta$;
  else
    Set λ as in (4.35) ;
    Compute $\theta_k'(\lambda)$ using to (4.37);
    if $\theta_k'(\lambda) \le 0$ **then**
      Compute $i^-(\lambda,k)$, $i^+(\lambda,k)$, and $\theta_k''^+(\lambda)$ using to (4.36) and (4.38);
    end
    while $\theta_k'(\lambda) < 0$ **do**
      $\eta := \lambda$;
      Set λ as in (4.35) and compute $\theta_k'(\lambda)$ using 4.37;
      if $\theta_k'(\lambda) \le 0$ **then**
        Compute the values of $i^-(\lambda,k)$, $i^+(\lambda,k)$ and $\theta_k''^+(\lambda)$ according to (4.36), (4.37) and (4.38);
      end
    end
    if $\theta_k'(\lambda) = 0$ **then**
      Store $\eta_k^\star := \lambda$;
    else if $\theta_k'(\eta) = 0$ **then**
      Store $\eta_k^\star := \eta$;
    else
      Store $\eta_k^\star = \eta - \frac{\theta_k'(\eta)(\lambda-\eta)}{\theta_k'(\lambda)-\theta_k'(\eta)}$;
    end
    Set $\eta = \eta_k^\star$;
  end
  $k := k+1$;
  if $k < n$ **then**
    Compute $i^-(\eta,k)$, $i^+(\eta,k)$, $\theta_k'(\eta)$ and $\theta_k''^+(\eta)$ according to (4.39), (4.40) and (4.41);
  end
end
return The sequence $(\eta_k^\star)_{0\le k\le n-1}$

Proof. Let $i \in [\![1,n]\!]$ be fixed. If $i \in J_k$, then since the sequence $(\eta_k^\star)_{0\le k\le n-1}$ is non-decreasing (in view of Lemma 4.9) and since the sequence $(\nu k/(n(n-k)))_{0\le k\le n-1}$ increases, we have

$$x_{(i)} - \frac{\nu(k+1)}{n(n-k-1)} \le \eta_k^\star \le \eta_{k+1}^\star$$

which implies $i < \min J_{k+1}$. Reciprocally, using the same properties, if $i \in J_{k+1}$, then

$$x_{(i)} - \frac{\nu(k+1)}{n(n-k-1)} > \eta_{k+1}^\star \ge \eta_k^\star$$

and we cannot have $i > \max J_k$. □

Upper bound on the complexity of lines 26 to 29

Lemma 4.14. *There exists a constant C_3 such that for all rank $k \in [\![0, n-1]\!]$, the number R_k of operations required to run the lines 26 to 29 in Algorithm 2 satisfies*

$$R_k \leq C_3 \ \mathrm{card}(J_k) + 6. \tag{4.43}$$

Proof. Line 26 and 27 both require 1 operation. Computing $i^-(\eta_k^\star, k+1)$ according to (4.39) also requires 1 operation. Computation of $i^+(\eta_k^\star, k+1)$ using (4.39) clearly requires 2 card $J_k + 1$ iterations (2 inequalities per term in $J_k + 1$ inequality for the term indexed by $i^+(\eta_k^\star, k+1)$). Computation of $\theta'_{k+1}(\eta)$ using (4.40) also requires at most a constant multiple m of card J_k operations (in view of (4.40), $m < 10$). Finally once $i^+(\eta_k^\star, k+1)$ is known, computing $\theta''^+_{k+1}(\eta_k^\star)$ using (4.41) requires 3 operations. Overall, we obtain that

$$R_k \leq (m+2)\,\mathrm{card}\, J_k + 3 + 1 + 1 + 1 + 1 = (m+2)\,\mathrm{card}\, J_k + 7$$

which gives (4.43). □

We can finally prove Theorem 4.10, which is the main result of this section.

Proof of Theorem (4.10)

Proof. Initialization of Algorithm 2 can clearly be done in constant time (assuming the x_i's are already sorted). Let $\mathcal{T}_n$ denotes the number of operations required to run lines 1 to 30 of Algorithm 2. For the stage $k = 0$, we know that the initial value of η, $\eta_0^\star - 1$ is already a root of θ_0 by Lemma 4.8, which implies that lines 4 to 25 are not run during this iteration. Thus, the total number of iterations $\mathcal{T}_n$ satisfies

$$\mathcal{T}_n \leq \underbrace{2 + R_0}_{\text{first stage}} + \underbrace{\sum_{k=1}^{n-1} (2 + D_k + C_2 + R_k)}_{\text{other stages}} \leq (2 + C_2)\, n + \sum_{k=1}^{n-1} D_k + \sum_{k=0}^{n-1} R_k$$

By Lemma 4.12, we have

$$\begin{aligned}\sum_{k=0}^{n} D_k &\leq C_1 n + C_1 \sum_{k=0}^{n-1} \mathrm{card}\, M_k \\ &\leq C_1 n + C_1 \sum_{k=0}^{n-1} \mathrm{card}\, I_k^- + \mathrm{card}\, I_k^+, \text{(by definition of } M_k, I_k^- \text{ and } I_k^+)\end{aligned}$$

by definition of M_k, I_k^-, and I_k^+. Now, by Lemma 4.11 and since all sets I_k^+ and I_k^- are in $[\![1, n]\!]$, we have $\sum_{k=0}^{n-1} \mathrm{card}\, I_k^- \leq n$, and $\sum_{k=0}^{n-1} \mathrm{card}\, I_k^+ \leq n$. Hence,

$$\sum_{k=0}^{n} D_k \leq 3C_1 n.$$

Finally, given Lemma 4.14 we have:

$$\sum_{k=0}^{n} R_k \leq 7n + C_3 \sum_{k=0}^{n-1} \mathrm{card}\, J_k$$

and by Lemma 4.13 we also have $\sum_{k=0}^{n} J_k \leq n$ which yields

$$\sum_{k=0}^{n} R_k \leq (7 + \mathcal{C}_3)n.$$

Thus,

$$\mathcal{T}_n \leq (9 + 3\mathcal{C}_1 + \mathcal{C}_2 + \mathcal{C}_3))n$$

which finishes the proof. □

BACK TO THE COMPUTATION OF (4.33).

Let us show now how the terms A and B from (4.33) may be computed in linear time.

Let $(\varepsilon_1, \ldots, \varepsilon_n)$ denote the canonical basis of $\mathbb{R}^n$. Let $(x_{(i)})_{1\leq i\leq n}$ be again the sequence of coordinates of x sorted in an arbitrary non-decreasing order. If σ denotes the permutation that sorts the coordinates of x (i.e. $x_{(i)} = x_{\sigma(i)}$), we will accordingly transpose the notation by defining for $i \in [\![1, n]\!]$, $\varepsilon_{(i)} := \varepsilon_{\sigma(i)}$, etc...

We show in the next two paragraphs how the expressions A and B from (4.33) may be computed in linear time.

Computing A from (4.33). We first show how the computation of A may be performed in linear time. Indeed, observe that

$$\begin{aligned}
A &= \sum_{k=1}^{n} \beta_k \sum_{i=i^+(\eta^\star_{k-1},k-1)}^{n} \frac{1}{n-(k-1)} \varepsilon_i \\
&= \beta_1 \frac{1}{n} \sum_{i=i^0_{\eta^\star_0}}^{n} \varepsilon_{(i)} + \beta_2 \frac{1}{n-1} \sum_{i=i^1_{\eta^\star_1}}^{n} \varepsilon_{(i)} + \cdots + \beta_n \sum_{i=i^{n-1}_{\eta^\star_{n-1}}}^{n} \varepsilon_{(i)} \\
&= \sum_{k=1}^{n} \left(\sum_{s=1}^{k} \beta_s \frac{1}{n-(s-1)} \right) \sum_{i=i^+(\eta^\star_{k-1},k-1)}^{i^+(\eta^\star_k,k)-1} \varepsilon_{(i)}.
\end{aligned}$$

Hence, *Computing A in linear time*

$$A = \sum_{k=1}^{n} c_k \sum_{i=i^+(\eta^\star_{k-1},k-1)}^{i^+(\eta^\star_k,k)-1} \varepsilon_{(i)} \tag{4.44}$$

where $c_k := \sum_{i=1}^{k} \beta_i \frac{1}{n-i+1}$. Finally, through direct considerations, one may observe that the sequence $(i^+(\eta^\star_k, k))_{0\leq k\leq n-1}$ is non-decreasing. Hence, the (possibly empty) intervals $[\![i^+(\eta^\star_{k-1}, k-1), i^+(\eta^\star_k, k)-1]\!]$ do not intersect each other. Based on (4.44) , we propose in Algorithm 3 a computation of A in linear time.

Computing B from (4.33). We note that expression B from (4.33) writes: *Computing B in linear time*

$$B = \sum_{i=1}^{n} \left(\sum_{k\in K_i} \beta_k \frac{x_{(i)} - \eta^\star_{k-1}}{\nu} + \frac{1}{n} \right) \varepsilon_{(i)} = \sum_{i=1}^{n} (a_i x_{(i)} + b_i)\varepsilon_{(i)} \tag{4.45}$$

Algorithm 3: Computation of the term A from (4.33)

Require : Sorted sequence $(x_{(i)})_{1\leq i\leq n}$,
Sequence of dual solutions $(\eta_k^\star)_{0\leq k\leq n-1}$,
Sequence of weights $(\beta_k)_{1\leq k\leq n}$

Initialize: $k := 1$, $i := 1$, $c := \frac{\beta_1}{n}$, $A := (0,\dots,0)^\top \in \mathbb{R}^n$.

```
while x_(i) − v 1/(n−(k−1)) ≤ η*_{k−1} do
    i = i + 1 ;
end
while i ≤ n do
    while x_(i) − v 1/(n−k) ≤ η*_k do
        A_(i) = A_(i) + c ;
        i = i + 1 ;
        if i = n + 1 then
            Break;
    end
    k = k + 1, c = c + β_k 1/(n−(k−1)) ;
end
return A.
```

where for $i \in [\![1, n]\!]$, we introduced

$$\begin{aligned} K_i &:= \left\{ k,\ \eta_{k-1}^\star - \frac{\nu}{n} < x_{(i)} \leq \eta_{k-1}^\star + \frac{\nu(k-1)}{n(n-k+1)} \right\} \\ a_i &:= \sum_{k\in K_i} \frac{\beta_k}{\nu} \\ b_i &:= \sum_{k\in K_i} \frac{-\beta_k \eta_{k-1}^\star}{\nu} + \frac{1}{n}. \end{aligned} \tag{4.46}$$

Algorithm 4 uses this formulation to reach a linear time complexity for the computation of B. We introduce there the sequence $(t_i)_{1\leq i\leq 3n}$ which is a concatenation the sequences

$$\begin{aligned} S_0 &:= (x_i)_{1\leq i\leq n} \\ S_1 &:= (\eta_k^\star - \frac{\nu}{n})_{0\leq k\leq n-1} \\ S_2 &:= (\eta_k^\star + \frac{\nu k}{n(n-k)})_{0\leq k\leq n-1} \end{aligned} \tag{4.47}$$

in an non-decreasing order. We assume also without harming the time complexity of such sorting that in case of equality between any terms of these three sequences, the terms in S_0 always appear before the ones in S_1 and S_2. In Algorithm 4, we perform a pass over the sequence $(t_i)_{1\leq i\leq 3n}$ and update the variable B every time a point t_i initially in S_0 is encountered. Variables a and b are designed to track the sequences $(a_i)_{1\leq i\leq n}$ and $(b_i)_{1\leq i\leq n}$ as defined in (4.46).

4.3 SPQR: A PYTHON-TOOLBOX FOR SUPERQUANTILE-BASED RISK MEASURES

We provide a Python software package for superquantile-based learning; it is named SPQR for SuPer Quantile Risk optimization. The software package includes modeling tools and optimization algorithms to minimize objectives of

Algorithm 4: Computation of the term B from (4.33)

Require : Sequence of weights $(\beta_k)_{1\leq k\leq n}$
$(t_i)_{1\leq i\leq 3n}$: sorted union of S_0, S_1 and S_2 as defined in (4.47)
Initialize: $a := 0$, $b := 0$, $i := 1$, $B := (0,\dots,0)^\top \in \mathbb{R}^n$

while $i \leq 3n$ **do**
  if $t_i \in S_1$ **then**
    $k :=$ initial position of t_i in the sequence S_1 ;
    $a := a + \frac{\beta_k}{\nu}$;
    $b := b - \frac{\beta_k \eta^\star_{k-1}}{\nu} + \frac{1}{n}$;
  else if $t_i \in S_2$ **then**
    $k :=$ initial position of t_i in the sequence S_2 ;
    $a := a - \frac{\beta_k}{\nu}$;
    $b := b + \frac{\beta_k \eta^\star_{k-1}}{\nu} - \frac{1}{n}$;
  else
    $k :=$ initial position of t_i in the sequence S_0 ;
    $B_{(k)} := B_{(k)} + a \times x_{(k)} + b$;
  end
  $i = i + 1$;
end
return B

the form (4.2) with just a few lines of code. The implementation builds off basic structures of `scikit-learn` [122] the popular python machine learning library. SPQR routines rely on just-in-time compilation [93] to ensure efficient running times. The software package is publicly available at

https://github.com/yassine-laguel/spqr.

We now walk the reader through the toolbox SPQR.

BASIC USAGE: INPUT FORMAT AND EXECUTION.

Input format

The user provides a dataset modeled as a couple $(A, B) \in \mathbb{R}^{n\times p} \times \mathbb{R}^p$ and a first-order oracle for the function L^i. The dataset is stored into two numpy arrays `A` and `B`; for instance, for realizations of random variables:

```
import numpy as np
A = np.random.rand(100, 2)
alpha = np.array([1., 2.])
B = np.dot(A, alpha) + np.random.rand(100)
```

The two python functions `L` and `L_prime` are assumed to be functions of the triplet `(w,a,b)` where `w` is the variable and `(a,b)` a datapoint. For instance, the oracle for superquantile linear regression are the following one.

```
# Define the loss and its derivative
def L(w,a,b):
    return 0.5 * np.linalg.norm(b - np.dot(a,w))**2
def L_prime(w,a,b):
    return -1.0 * (b - np.dot(a,w)) * a
```

The `RiskOptimizer` object

Before minimizing (4.2), we instantiate the `RiskOptimizer` object with the

oracles, following the standard usage of `scikit-learn`. The basic instantiation is:

```
from SPQR import RiskOptimizer
# Instantiate a risk optimizer object
optimizer = RiskOptimizer(L, L_prime)
```

`RiskOptimizer` inherits from `scikit-learn`'s estimators: we use the `fit` method to run the optimization algorithm on the provided data, to get a minimizer of (4.2).

```
#  Running the algorithm
optimizer.fit(A,B)
sol = optimizer.solution
```

ADVANCED USE: PARAMETERS AND SPQR OBJECTS.

Customization

Options and parameters. The customizable parameters are stored in a python dictionary, called `params`, which is designed as an attribute of the `RiskOptimizer` class. The main parameters to tune are: the choice of the oracle, the choice of the algorithm, the safety probability level `p`, the starting point of the algorithm `w_start`, the maximum number of iterations `max_iter`. The user can specify some of these parameters as an input and the others will be filled with defaults values when instantiating a `RiskOptimizer`. For example:

```
custom_params = {'algorithm': 'dualaveraging', # selected algorithm
    'p': 0.2 } # safety probability level
custom_optimizer = RiskOptimizer(loss, loss_prime,
    params=custom_params)
```

Some important parameters (such as the safety probability level, the algorithm chosen, or the smoothing parameter μ) can be given directly to the constructor of the class `RiskOptimizer` when instantiating the object. For example:

```
other_custom_optimizer = RiskOptimizer(loss, loss_prime, p=0.95,
    algorithm='bfgs', mu=0.1)
```

Available oracles

Oracle classes. The selection of the oracle is automatically done when the user instantiate the `RiskOptimizer` object. Four different oracles are implemented as python objects. The first two oracles, `OracleSubgradient` and `OracleSmoothGradient` are designed for batch methods. The subgradient oracle `OracleSubgradient` is the one instantiated when the algorithm chosen by the user is `'subgradient'` or `'dual_averaging`. The smoothed oracle is instantiated when the algorithm chosen is `'gradient'`, `'nesterov'` or `'bfgs'`. The last two oracles are designed for mini-batch methods: `OracleStochasticSubgradient` and `OracleStochasticGradient` To avoid the treatment of optional parameters when instantiating an oracle, we advise to go through the instantiation of a `RiskOptimizer` first.

```
custom_params = {'algorithm': 'nesterov', # selected algorithm
    'p': 0.5 # safety probability level
}
#  Instantiation of the Risk Optimizer
custom_optimizer = RiskOptimizer(loss, loss_prime,
    params=custom_params)
```

```
# Recovery of the oracle
smooth_oracle = custom_optimizer.oracle
```

Available algorithms

Algorithms class. The algorithm chosen is a parameter for the instantiation of the `RiskOptimizer` class. This parameter can either be given in the input dictionary `params` or directly to the constructor of `RiskOptimizer`. The user has the choice among `'subgradient'`, `'dual_averaging'`, `'gradient'`, `'nesterov'`, `'bfgs'` and `'sgd'`.

```
# Risk Optimizer class with nesterov accelerated gradient algorithm
custom_optimizer = RiskOptimizer(loss, loss_prime,
    algorithm='nesterov')
```

Each algorithm is implemented as a python class that stores the oracle, together with relevant parameters for the optimization process. The main method of each implemented algorithm class is `run`, which is run when `RiskOptimizer.fit` is called. The parameters of the algorithm selected are stored in the dictionary `params` that is an input of the class `RiskOptimizer`. Hence, in a standard usage, there is no need to interact with the algorithm python object.

4.4 NUMERICAL EXPERIMENTS

In this section, we report two types of numerical experiments:

- In the first paragraph, we consider “Optimization" experiments. There are many algorithmic options within the toolbox `SPQR`; we provide here a comparison of batch vs. mini-batch algorithms and a discussion of the tuning of the smoothness parameter.
- In the first paragraph, we consider “Learning" experiments. The interest of using superquantile in learning has been shown empirically in several recent papers, including [34, 89, 94, 152, 174]. We provide here complementary experiments highlighting the robustness of superquantile-learnt models.

All experiments are run using `SPQR`. The optimization algorithms are initialized at $w = 0 \in \mathbb{R}^d$. For these experiments, we use a bunch of standard datasets from the UCI repository, which scale from 352 to 94644 datapoints. For each dataset, categorical features were one-hot encoded so that the total number of features ranges from 3 to 287. For one experiment, we report the agreggated results for all the datasets. For the other experiments, we report in the main text the detailed results for one representative dataset, and we provide in appendix complementary results for others datasets.

4.4.1 *Solving superquantile-based learning*

In this section, we illustrate two different aspects of the optimization methods available in `SPQR`. First, we compare the two families of algorithms available batch vs. mini-batch (more precisely SGD with momentum and L-BFGS) showing the interest of using batch algorithms for superquantile-based learning within `scikitlearn/SPQR`. Second, we experiment with all the range of the smoothing parameter, advocating to avoid extreme values.

Batch vs. mini-batch. We compare on a standard problem stochastic gradient optimization (denoted SGD) and batch quasi-Newton optimization using low-memory BFGS [119] (denoted BFGS).

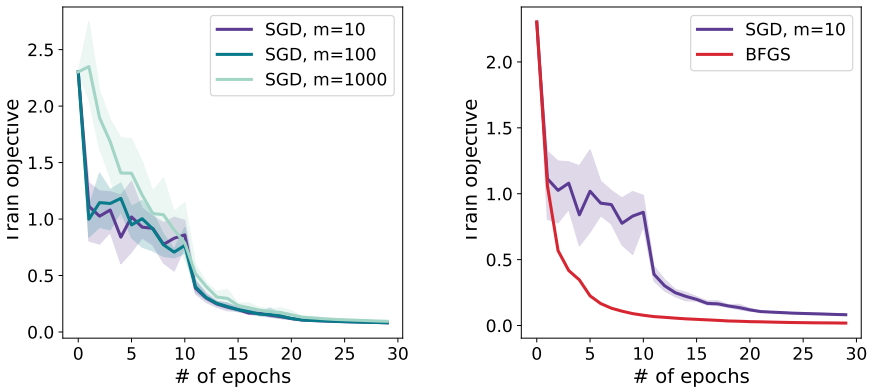


Figure 4.4: A comparison between batch/mini-batch algorithms in `SPQR` on a superquantile logistic regression problem with `MNIST`. Left: comparison of the runs of SGD with different batch sizes. Right: best SGD vs. batch quasi-Newton.

Setup

For this experiment, the set-up is similar to the one of [94]. We consider a supervised multi-class classification task with the superquantile multinomial logistic loss on the `MNIST` dataset. We perform feature extraction from the images using a pre-trained convolutional network similarly to [94]. For a fixed probability threshold set to $p = 0.8$, we then train a linear multi-class classifier on top of the transformed data. For SGD, we use a momentum term of 0.9 and we use a step decay scheme $\eta_t = \eta_0 d^{-\lfloor t/t_0 \rfloor}$, where η_0 is tuned with respect to the size of the mini-batch m, and where $d = 0.5$ and $t_0 = 10$ epochs are fixed throughout all the experiments. For each mini-batch size $m \in \{10, 100, 1000\}$, we tune η_0 via a grid-search and take the highest initial value yielding a non-diverging sequence of iterates. In constrast with SGD, the quasi-Newton algorithm does not require specific tuning as it automatically calibrates stepsizes by line-searches at each iteration.

Comparison between L-BFGS and SGD

On the left part of Figure 4.4, we compare the performance of SGD for the different mini-batch sizes. Each color corresponds to a mini batch size $m \in \{10, 100, 1000\}$. Along iterates, the bold line represents the mean value over the five seeds of the functions and the shaded region represent the difference between the min and max values across the seeds. We observe that there is no substantial difference among the sizes of the mini-batches: all curves show a noisy behaviour(caused by the stochastic approximation of the gradient at each step) and eventually converge to a suboptimal value. Unlike SGD, L-BFGS (right part of Figure 4.4) presents a stable convergence. We observe also that a large number of epochs is necessary for SGD to catch up with BFGS for superquantile-based training. This is to be contrasted with the usually small number of epochs necessary for SGD to catch with BFGS for expectation-based training or ERM. Note that a a final bias remains visible between the stochastic methods and the deterministic BFGS, as expected by the theory laid down in [94].

Impact of the smoothing parameter on L-BFGS

Impact of the smoothing parameter on L-BFGS. We consider a logistic regression on the `Australian Credit` dataset. For a sequence of smoothing parameters ν evenly spread on a log scale, we train $w_\nu^\star$ by solving the superquantile learning objective with L-BFGS and $p = .99$.

On Figure 4.5, we report both the value of the smoothed .99-superquantile (purple) and the nonsmoothed .99-superquantile (dashed green) at the $w_\nu^\star$.

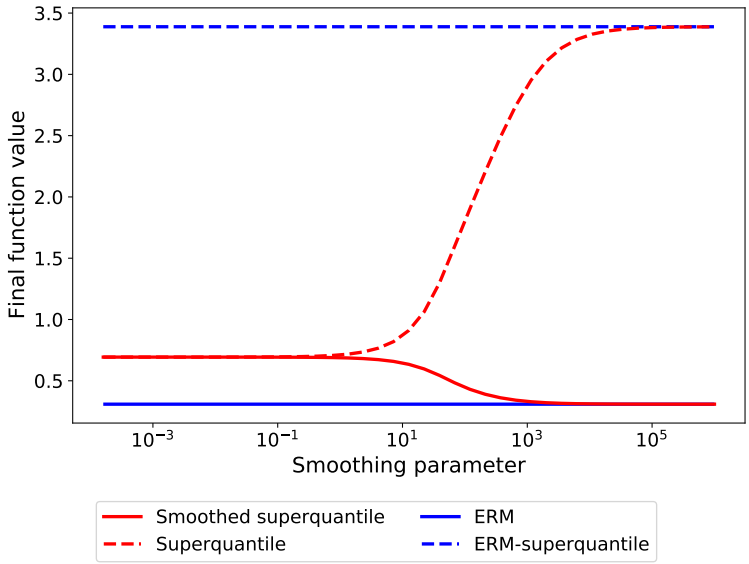


Figure 4.5: Impact of the smoothing parameter ν on the results obtained by the quasi-Newton algorithm solving a superquantile logistic regression on the `Australian Credit` dataset. Medium values are preferable: small values compromise convergence and large values give solutions close to the standard ERM.

We also train the standard empirical risk minimizer $w^\star$ and we report both the average loss (solid black line) and the nonsmoothed .99-superquantile loss (dashed black line) at $w^\star$.

For very small values of ν ($< 10^{-3}$), we observe unsuccessful termination of the L-BFGS algorithm, due to the failure of the line-search. For medium values of ν (< 1), the value of smooth superquantile-based function at $w_\nu^\star$ roughly coincides the nonsmooth one. Finally for high values of $\nu (> 10^3)$, we observe that the smooth superquantile tends to the same optimal function value of the empirical risk minimizer $w^\star$, as expected from Section 4.2.

4.4.2 *Superquantile brings robustness against distributional shifts*

In the second part of the numerical experiments, we show the interests of superquantile for worst-case learning by comparing superquantile-based minimization vs empirical risk minimization, similarly to [34]. For the three next standard regression or classification tasks, we proceed as follows. For each dataset, we first perform a 80%-20% train-test split. Second, we minimize with respect to the train set a regularized objective, both in expectation and with respect to the superquantile:

$$\begin{aligned} &\min_{w \in \mathbb{R}^d} \ \mathbb{E}_{(x,y)\sim D_{\text{train}}} \ell(y, w^\top x) + \frac{\lambda}{2}\|w\|_2^2 \\ &\min_{w \in \mathbb{R}^d} \ [S_p]_{(x,y)\sim D_{\text{train}}} \ell(y, w^\top x) + \frac{\lambda}{2}\|w\|_2^2 \end{aligned} \tag{4.48}$$

We set the regularization parameter λ to be the inverse of the number of training data-points: $\lambda = 1/n_{\text{train}}$. The above problems are solved with `SPQR` using L-BGFS. Then we perform three different types of distributional shifts on the

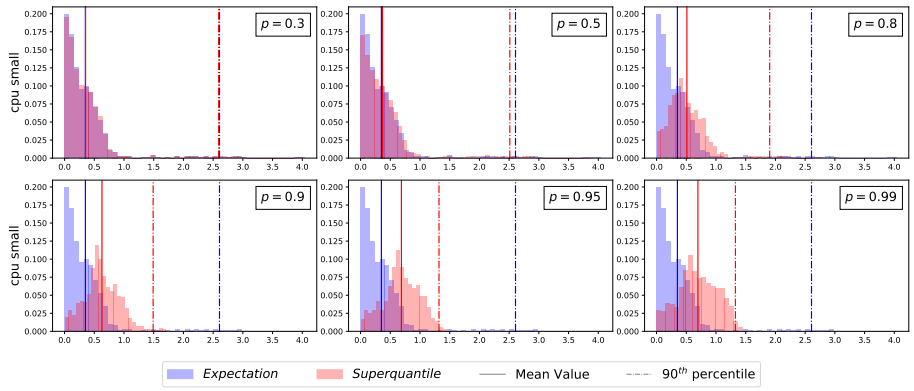


Figure 4.6: Reshaping of the histogram of testing losses for superquantile regression models (in red) as p grows. We observe a shift to the left of the 90^{th} quantile of losses, at the price of degrading the average value.

testing set and we compare the behaviour of the superquantile-based models and the ERM models. We develop this approach in the next three settings.

Setup

Superquantile ridge-regression. We consider a ridge regression problem, that is (4.48) with $\ell(y, w^\top x) = (y - w^\top x)^2$, on the dataset `Cpu-small`. We minimize the two problems in expectation and (ii) with respect to the superquantile with several safety thresholds $p \in \{0.3, 0.5, 0.7, 0.8, 0.9, 0.95, 0.99\}$.

Superquantiles reduce worst-case losses

We report in Figure 4.6 the histogram of losses on the test set and compare each trained superquantile model (in red) with the ERM model (in blue). We observe that as the probability threshold p grows, the right tail distribution of losses on the test set gets shifted to the left. In particular, a drastic decrease of the 90^{th} quantile of the losses can be observed. Thus superquantile learning allows us to reduce worst-case losses. This comes with the price of lower performances on the left tail distribution.

Setup

Superquantile logistic regression. We consider a regularized logistic regression problem, that is (4.48) with $\ell(y, w^\top x) = -y\sigma(w^\top x) - (1-y)\sigma(w^\top x)$ (where $\sigma(z) := \frac{1}{1+e^{-z}}$ denotes the sigmoid function). We use 10 classification datasets from the UCI repository library and we perform a distributional shift on the train sets: we subsample the majority class so that it accounts for only 10% of the minority class. Then we train a ERM and superquantile models. The safety parameter p is tuned via a cross validation procedure on the shifted train set. We finally compute, for the best parameter obtained, the test accuracy and the test loss.

Robustness of superquantile for a given distributional shifts

We report our results in Table 4.1. For most datasets, we note a significant decrease of the test loss with the superquantile model, when compared to ERM model. In terms of accuracy, the superquantile model offers better performance for this particular distributional shift.

Setup

Robustness to all possible distributional shifts. We take the same setting as before, focusing on the `splice` dataset, and now we perform a sequence of distributional shifts on the training set by rebalancing all the proportions of the two classes. More precisely, for a fixed $\alpha \in (0, 1)$, we compute the number $n_{\min}$ of sample from the minority class; we randomly select $\lceil \alpha n_{\min} \rceil$ points from the majority class and $\lceil (1-\alpha) n_{\min} \rceil$ from the minority class. We train on the shifted train set the two logistic regression models of (4.48). We repeat this experiment for 5 different seeds and we compute the average test loss and test accuracies of

	Superquantile		Expectation	
Dataset	Accuracy	Loss	Accuracy	Loss
Adult	53.2 ± 0.67	0.693 ± 0.00	$\mathbf{55.4 \pm 0.48}$	1.072 ± 0.01
Monks	$\mathbf{64.4 \pm 2.65}$	0.714 ± 0.05	54.0 ± 1.57	1.207 ± 0.08
Splice	$\mathbf{82.7 \pm 0.62}$	0.681 ± 0.05	81.7 ± 0.78	0.557 ± 0.04
Diabetes	42.5 ± 4.72	0.694 ± 0.00	$\mathbf{45.1 \pm 4.51}$	1.325 ± 0.12
Spambase	$\mathbf{78.4 \pm 1.23}$	0.761 ± 0.15	77.1 ± 0.87	0.635 ± 0.07
Mammography	$\mathbf{39.1 \pm 7.59}$	0.730 ± 0.01	$\mathbf{39.1 \pm 6.90}$	1.293 ± 0.09
Electricity	42.8 ± 0.40	0.693 ± 0.00	$\mathbf{47.5 \pm 0.63}$	1.060 ± 0.01
Phoneme	37.3 ± 5.38	0.737 ± 0.01	$\mathbf{50.5 \pm 3.10}$	1.292 ± 0.04
Nomao	$\mathbf{87.5 \pm 0.22}$	0.413 ± 0.03	$\mathbf{87.4 \pm 0.23}$	0.394 ± 0.02
Skin-segmentation	$\mathbf{92.1 \pm 0.11}$	0.420 ± 0.00	$\mathbf{91.9 \pm 0.05}$	0.537 ± 0.01

Table 4.1: Comparison of performances between a superquantile model and a risk-neutral model for a logistic regression on a distributionally shifted dataset.

both models. The experiment is conducted for 100 values of α evenly spread on $(0, 1)$.

Robustness of superquantiles to adversarial distributional shifts

The histograms of Figure 4.7 depicts the performances, as α varies, of ERM against the superquantile for a fixed probability threshold p. In terms of losses, the superquantile model brings better performances for almost all values of p. In particular, the 90^{th} quantile of the losses over all considered shifts gets notably decreased for p.In terms of accuracy, the superquantile models brings better performance with respect to distributional shifts for all values of p.

4.5 CONCLUSION

In this chapter, we addressed the problem of minimizing superquantile-based risk measures by first-order methods. We provided explicit expressions of (sub)gradients of (smoothed) superquantiles where we went down to the details of computations in order to get efficient first-order oracles for superquantile-based functions. We provided a generalization of such smoothing procedures to coherent law-invariant comonotone risk measures with a special care on maintaining a minimal complexity. We released an open-source python toolbox for the minimization of superquantile-based objectives. The first one, built on top of `scikit-learn` aims at tackling general convex problems of low or middle size. We finally provided an overview of the recent applications of superquantiles in machine learning and illustrated on real datasets, the performances of our methods through several numerical experiments.

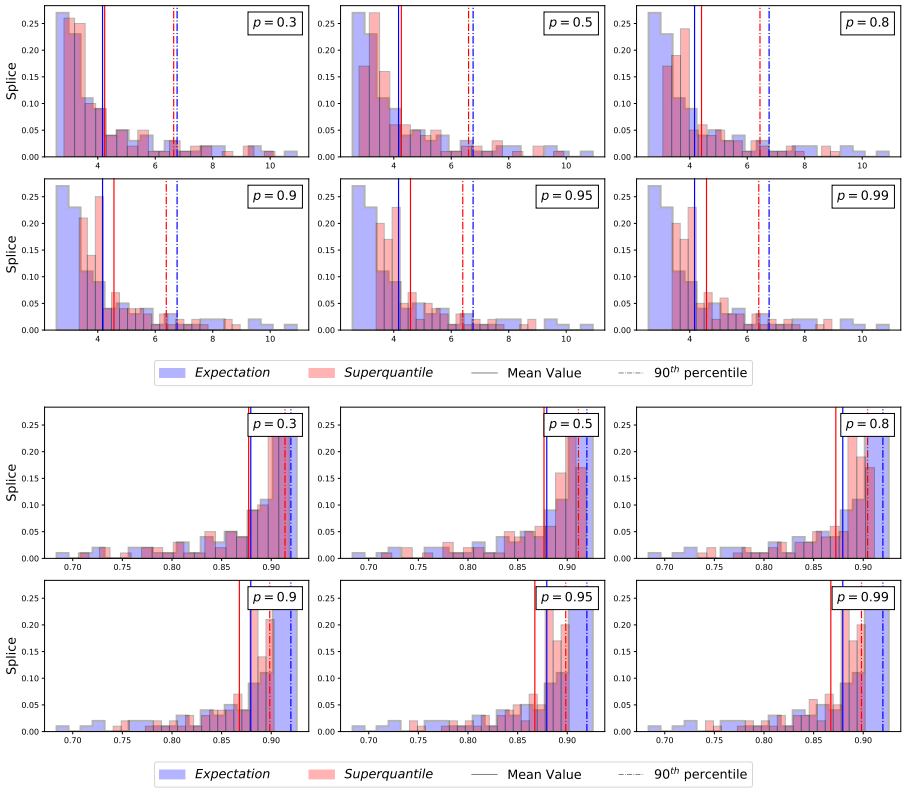


Figure 4.7: Reshaping of histograms of test losses (top) and test accuracies (bottom) over all class imbalances (for a classification task with logistic regression and the `splice` dataset).

5

SOLVING CHANCE CONSTRAINED PROBLEMS

This chapter is devoted to a new algorithm for solving chance constrained (convex) programs.

The developments laid down below build upon the following work:

- Y. Laguel, W. Van Ackooij and J. Malick. Chance constrained convex problems: a bilevel convex optimization perspective. Preprint: https://arxiv.org/pdf/2103.10832.pdf, 2021.

5.1 INTRODUCTION

General chance constrained problem

In this chapter, we address the solving of the general chance-constrained optimization problem introduced in Section 3.5. That is, for a fixed safety probability level $p \in [0, 1)$, we consider:

$$\begin{aligned} \min_{x \in \mathcal{C}} \quad & f(w) \\ \text{s.t.} \quad & \mathbb{P}[g(w, \xi) \leq 0] \geq p, \end{aligned} \tag{5.1}$$

where $f\colon \mathbb{R}^d \to \mathbb{R}$ and $g\colon \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}$ are two given functions, ξ is a random vector valued in $\mathbb{R}^m$ and $\mathcal{C} \subset \mathbb{R}^d$ is a (deterministic) closed constraint set. We consider the case of underlying convexity: we assume that f and g are convex (with respect to w). For our practical developments, we also assume that we have first-order oracles for f and g and that the $\mathcal{C}$ is a box constraint on the decision variable x. As discussed in Section 3.5, even with underlying convexity, chance-constrained problems remain difficult to solve: the chance constraint feasible can remain non-convex even in simple standard cases as illustrated in Example 3.4.

In this chapter, we propose an original approach for solving chance-constrained optimization problems. First, we present an exact reformulation of (nonconvex) chance-constrained problems as (convex) bilevel optimization problems. This reformulation is simple and natural, involving superquantiles. Second, exploiting this bilevel reformulation, we propose a general algorithm for solving chance-constrained problems, and we release an open-source python toolbox implementing it. In the case where we make no assumption on the underlying uncertainty and have only samples of ξ, we propose and analyse a double penalization method, leading to an unconstrained single level DC (Difference of Convex) programs. Our approach enables to deal with a fairly large sample of data-points in comparison with state-of-the-art methods based on mixed-integer reformulations, e.g. [3]. Thus our work mixes a variety of techniques coming from different subdomains of optimization: penalization, error bounds, DC programming, bundle algorithm, Nesterov's smoothing; relevant references are given along the discussion.

Outline

This chapter is structured as follows. In Section 5.2, we leverage the known

link with (super)quantiles and chance-constraint to establish a novel bilevel reformulation of general chance constrained problems. In Section 5.3, we propose and analyse a penalty approach revealing the underlying DC structure. In Section 5.4, we discuss implementation of this approach in our publicly available toolbox. In Section 5.5, we provide illustrative numerical experiments, as a proof of concept, showing the interest of the method. Technical details on secondary theoretical points and on implementation issues are postponed to appendices.

5.2 CHANCE CONSTRAINED PROBLEMS SEEN AS BILEVEL PROBLEMS

In this section, we derive the reformulation of a chance constraint as a bilevel program wherein both the upper and lower level problems, when taken individually, are convex.

By definition, the chance constraint in (5.1) involves the cumulative distribution function: we have for any fixed $w \in \mathbb{R}^d$, $\mathbb{P}[g(w,\xi) \leq 0] \geq p \Leftrightarrow F_{g(w,\xi)}(0) \geq p$. We rewrite this constraint using quantiles, as formalized in the next lemma.

From chance constraints to quantile constraints

Lemma 5.1. *For any $w \in \mathbb{R}^d$ and $p \in [0,1)$, we have:*

$$\mathbb{P}[g(w,\xi) \leq 0] \geq p \iff Q_p(g(w,\xi)) \leq 0.$$

Proof. By definition of the quantile and continuity on the right of the cumulative distribution function, we always have $p \leq \mathbb{P}[g(w,\xi) \leq Q_p(g(w,\xi))]$. Thus, since cumulative distribution functions are increasing, if $Q_p(g(w,\xi)) \leq 0$, then $\mathbb{P}[g(w,\xi) \leq Q_p(g(w,\xi))] \leq \mathbb{P}[g(w,\xi) \leq 0]$ which implies that $\mathbb{P}[g(w,\xi) \leq 0] \geq p$. Conversely, since $Q_p(g(w,\xi))$ is the infimum of $\{t \in \mathbb{R} \ : \mathbb{P}[X \leq t] \geq p\}$, if $Q_p(g(w,\xi)) > 0$, then necessarily we have $\mathbb{P}[g(w,\xi) \leq 0] < p$. □

Such reformulation has already been investigated in previous works (see e.g. [123]). Compared to the empirirical cumulative distribution function, the p-quantile benefits from more regularity: $x \mapsto Q_p(g(w,\xi))$ is continuous with respect to x whenever g is. However, the quantile remains a non-convex and non-smooth function. We propose to reformulate once again this problem in terms of superquantiles. To that end, we build upon the variational formulation (3.12) of the superquantile in its general form.

Quantile are solution to the variational problem (3.12)

Lemma 5.2. *For an integrable random variable X and a probability level p, the superquantile $S_p(X)$ is the optimal value of the convex one-dimensional problem*

$$\inf_{\eta \in \mathbb{R}} \ \eta + \frac{1}{1-p}\,\mathbb{E}[\max(X-\eta, 0)]. \tag{5.2}$$

Moreover, the quantile $Q_p(X)$ is the left end-point of the solution interval of this problem

Together with (5.2), we obtain from the previous easy lemma a bilevel formulation of the general chance-constrained problem (5.4). The idea is simple: introducing an auxiliary variable $\eta \in \mathbb{R}^d$ to recast the potentially non-convex chance constraint of (5.1) as two constraints, a simple bound constraint and a difficult optimality constraint, forming a lower subproblem. Introducing the lower objective function $G : \mathcal{C} \times \mathbb{R} \to \mathbb{R}$

$$G(w,s) = s + \frac{1}{1-p}\,\mathbb{E}[\max(g(w,\xi)-s, 0)], \tag{5.3}$$

we have the following exact reformulation of chance-constrained problems.

Theorem 5.3. *Problem (5.1) is equivalent to the bilevel problem:*

Bi-level reformulation of chance constraints

$$\begin{cases} \min_{w\in\mathcal{C},\eta\in\mathbb{R}} & f(w) \\ s.t. & \eta \leq 0 \\ & \eta \in S(w) = \arg\min_{s\in\mathbb{R}} G(w,s). \end{cases} \tag{5.4}$$

More precisely, if $w^\star$ is an optimal solution of (5.1), then $(w^\star, Q_p(g(w^\star,\xi)))$ is an optimal solution of the above bilevel problem, and conversely.

Proof. It is clear with Lemma 5.1 that problem (5.1) is equivalent to

$$\begin{cases} \min\limits_{w\in\mathcal{C},\eta\in\mathbb{R}} & f(w) \\ \text{s.t.} & \eta \leq 0 \\ & \eta = Q_p(g(w,\xi)) \end{cases} . \tag{5.5}$$

By Lemma 5.2, $Q_p(g(w,\xi)) \in S(w)$ for any $w \in \mathbb{R}^d$. Hence, any solution (w,η) of (5.5) is feasible for (5.4). Conversely, any solution $(w^\star,\eta^\star)$ of (5.4) satisfies: $Q_p(g(w^\star,\xi)) \leq \eta^\star \leq 0$ which implies that $(w^\star, Q_p(g(w^\star,\xi))$ is a feasible point of (5.5). Since both problems have the same objective, they are equivalent. □

The first constraint $\eta \leq 0$ is an easy one-dimensional bound constraint which does not involve the decision variable w. The second constraint, which constitutes the lower level problem is more difficult; when this constraint is satisfied, η is an upper-bound on the p-quantile of $g(w,\xi)$. We readily see the joint convexity of the objective function of the lower level problem in (5.4) with respect to s and w. Note finally that the extension to joint chance constrained programs is straightforward: if $g : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^k$ has all its components $g_i, (1 \leq i \leq k)$ convex with respect to x, we have:

$$\begin{aligned} \mathbb{P}\left[g(w,\xi) \leq 0\right] \geq p \Leftrightarrow \quad & \mathbb{P}\left[\max_{1\leq i\leq k} g_i(w,\xi) \leq 0\right] \geq p \\ \Leftrightarrow \quad & Q_p\left(\max_{1\leq i\leq k} g_i(w,\xi)\right) \leq 0 \\ \Leftrightarrow \quad & \exists\, \eta \leq 0 \\ & \text{s.t. } \eta \in \arg\min_{s\in\mathbb{R}} s + \frac{1}{1-p}\,\mathbb{E}\left[\max_{0\leq i\leq k} g_i(w,\xi)\right] \text{ (with } g_0 := 0) \end{aligned}$$

and the lower-level problem remains convex.

This bilevel reformulation is nice, natural and seemingly new; we believe that it opens the door to new approaches for solving chance-constrained convex problems. In the next section, we propose such an approach based on the reformulation.

5.3 A DOUBLE PENALIZATION SCHEME

In this section, we explore one possibility offered by the bilevel formulation of chance-constrained problems, presented in the previous section. We propose a (double) penalization approach for solving the bilevel optimization problem, with a different treatment of the two constraints: a basic penalization of the easy

constraint together with an exact penalization of the hard constraint formalized as the lower problem.

We first exhibit and analyze in Section 5.3.1, the weak sharpness properties of the lower-level problem in (5.4). We show then in Section 5.3.2 to what extent these properties help to provide an exact penalization of the "hard" constraint. We finally present the double penalty scheme in section 5.3.3.

From the bilevel problem (5.4), we derive the two following penalized problems, associated with two penalization parameters $\mu, \lambda > 0$ and

First penalization

$$(P_\mu) \quad \begin{cases} \min\limits_{(w,\eta)\in\mathcal{C}\times\mathbb{R}} & f(w) + \mu \max(\eta, 0) \\ \text{s.t.} & \eta \in \arg\min_{s\in\mathbb{R}} G(w,s) \end{cases} \tag{5.6}$$

and

Second penalization

$$(P_{\lambda,\mu}) \quad \min_{(w,\eta)\in\mathcal{C}\times\mathbb{R}} f(w) + \lambda \left(G(w,\eta) - \min_{s\in\mathbb{R}} G(w,s) \right) + \mu \max(\eta, 0). \tag{5.7}$$

We consider a general data-driven situation where the uncertainty ξ is just known through a sample (or, said alternatively, follows an equiprobable discrete distribution over $n \in \mathbb{N}$ arbitrary values): we assume that there exists $\xi_1, \xi_2, \ldots, \xi_n \in \mathbb{R}^m$ such that $\mathbb{P}[\xi = \xi_i] = \frac{1}{n}$ for all $i \in \{1, \ldots, n\}$. The set $\mathcal{I}_n$ defined as

$$\mathcal{I}_n = \left\{ \frac{i}{n}, \quad i \in \{0, ..., n-1\} \right\} \tag{5.8}$$

plays a special role in our developments. In particular, we denote by $\text{dist}_{\mathcal{I}_n}(p) := \inf\{\|p - \frac{i}{m}\|, 0 \leq i \leq n-1\}$, the distance to $\mathcal{I}_n$,, to define a key quantity appearing in the variational results of this section: we introduce

$$\delta_p = \begin{cases} \dfrac{1}{n(1-p)} & \text{if } p \in \mathcal{I}_n \\ \dfrac{\text{dist}_{\mathcal{I}_n}(p)}{(1-p)} & \text{otherwise,} \end{cases} \tag{5.9}$$

which depends implicitly on the number of samples n and the fixed safety parameter p.

5.3.1 *Weak sharpness and analysis of the value function*

In chapter 4 of this thesis, we uncovered the nonsmoothness of the superquantile function based on a study of its distributionally robust formulation (3.11). We extend here the analysis by bringing to light the weak sharpness properties of the superquantile, based on the variational formulation (5.2). Given an optimization problem

$$\min_{w\in\mathcal{X}} \varphi(w),$$

we recall that its solution set $\mathcal{X}^\star$ is said to be *weakly sharp* (or equivalently that it satisfies the *first-order growth condition*) if there exists a constant $\delta > 0$ such that

Weak sharpness

$$\varphi(x) \geq \varphi(y) + \delta \, \text{dist}_{\mathcal{X}^\star}(x), \quad \forall x \in \mathcal{X}, \forall y \in \mathcal{X}^\star.$$

The constant δ, specific to each problem is often called the *weak-sharpness modulus* of this problem. Now, one may observe that weak sharpness of the problem (5.2) is a direct consequence of its linearity, in view of the next Proposition.

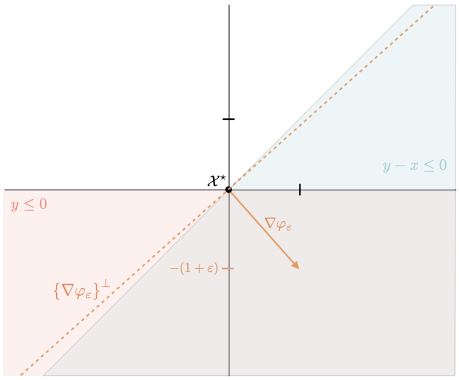


Figure 5.1: Illustration of Example 5.1. A linear problem with a vanishing weak sharpness modulus.

Proposition 5.4. *If a linear programming problem in a Banach space has a nonempty set of optimal solutions, then such solution set is weakly-sharp.*

See proposition 3.125 of [21]

The proof of such result is based on Hoffman's lemma (see e.g. [21][Proposition 2.200]) and guarantees the existence of a weak-sharpness modulus without suggesting a way to derive the modulus of a specific problem. One should also note that such modulus may be arbitrarily close to 0, in view of Example 5.1.

Vanishing Weak Sharpness

Example 5.1. For any $\varepsilon > 0$, consider the 2-dimensional problem $(\mathcal{P}_\varepsilon)$ defined as:

$$(\mathcal{P}_\varepsilon) \quad \begin{cases} \min\limits_{(x,y)\in\mathbb{R}^2} & \varphi_\varepsilon(x,y) = x - (1+\varepsilon)\, y \\ \text{s.t.} & y \leq 0 \\ & y - x \leq 0 \end{cases} \tag{5.10}$$

This problem is illustrated in Figure 5.1. Note that for any $\varepsilon > 0$ the solution set of (P_ε) is reduced to the singleton 0. Observe now that the point $(-1,-1)$ is feasible for all $\varepsilon > 0$ and $\varphi_\varepsilon(-1,-1) = \varepsilon$ while $\text{dist}_{\mathcal{X}^\star} = \sqrt{2}$ does not depend on ε. Hence taking ε arbitrarily small lead to a problem $\mathcal{P}_\varepsilon$ with arbitrarily small weak sharp modulus. □

Fortunately, the superquantile problem (5.2) admits for any $p \in [0,1)$ a weak-sharpness modulus which will be key in the forthcoming exact penalization procedure. We study here the value function $h : \mathcal{C} \times \mathbb{R} \to \mathbb{R}$ defined, from G in (5.3), as

$$h(w,\eta) = G(w,\eta) - \min_{s\in\mathbb{R}} G(w,s). \tag{5.11}$$

The next result relates h to $\text{dist}_{S(w)}(\cdot)$, the distance function to $S(w)$, the solution set of the lower level problems in (5.4). This is our main technical result, on which next propositions are based.

Weak sharpness of the superquantile

Theorem 5.5. *Let $p \in [0,1)$ be fixed but arbitrary. The function h defined in (5.11) satisfies for any $(w,\eta) \in \mathcal{C} \times \mathbb{R}$*

$$h(w,\eta) \geq \delta_p \ \text{dist}_{S(w)}(\eta) \qquad \textit{with } \delta_p \textit{ defined by (5.9).}$$

In other word, the variational problem (5.2) is weakly sharp with a weak-sharpness modulus higher than δ_p.

Proof. Let us fix $w \in \mathcal{C}$ and denote by q_p the p-quantile of $g(w,\xi)$. We first note that by the arguments in the proof of Lemma 5.1, we have

$$p \leq \mathbb{P}[g(w,\xi) \leq q_p],$$

with equality holding in the left inequality, if and only if p belongs to $\mathcal{I}_n$.

For any fixed but arbitrary $\eta \in \mathbb{R}$, we have the following identity:

$$\begin{aligned} h(w,\eta) &= \eta + \frac{1}{1-p}\mathbb{E}[\max(g(w,\xi)-\eta,0)] - \left(q_p + \frac{1}{1-p}\mathbb{E}[\max(g(w,\xi)-q_p,0)]\right) \\ &= (\eta - q_p) + \frac{1}{1-p}\mathbb{E}\left[\max(g(w,\xi),\eta) - \eta - (\max(g(w,\xi),q_p) - q_p)\right] \\ &= (\eta - q_p)(1 - \frac{1}{1-p}) + \frac{1}{1-p}\mathbb{E}\left[\max(g(w,\xi),\eta) - \max(g(w,\xi),q_p)\right]. \end{aligned}$$

Now, by employing a case distinction on the location of η with respect to q_p, we will derive the desired inequalities. Let us first consider that $\eta > q_p$, then we have:

$$\begin{aligned} h(w,\eta) &= (\eta - q_p)(1 - \frac{1}{1-p}) + \frac{1}{1-p}\mathbb{E}\left[(\eta - g(w,\xi))\,\mathbb{1}_{q_p < g(w,\xi) \le \eta} + (\eta - q_p)\,\mathbb{1}_{g(w,\xi)\le q_p}\right] \\ &= (\eta - q_p)(1 - \frac{1}{1-p} + \frac{1}{1-p}\,\mathbb{P}[g(w,\xi) \le q_p]) + \frac{1}{1-p}\,\mathbb{E}[(\eta - g(w,\xi))\,\mathbb{1}_{q_p < g(w,\xi)\le\eta}] \end{aligned}$$

which finally gives:

$$h(w,\eta) = \frac{(\eta - q_p)}{1-p}\left(\mathbb{P}[g(w,\xi) \le q_p] - p + \mathbb{E}\left[\frac{\eta - g(w,\xi)}{\eta - q_p}\,\mathbb{1}_{q_p < g(w,\xi)\le\eta}\right]\right). \quad (5.12)$$

Now if $p \notin \mathcal{I}_n$, since $\mathbb{P}[g(w,\xi) \le q_p] \in \mathcal{I}_n$, we have:

$$h(w,\eta) \ge (\eta - q_p)\frac{1}{1-p}\left(\mathbb{P}[g(w,\xi) \le q_p] - p\right) \ge \frac{\mathrm{dist}_{\mathcal{I}_n}(p)}{1-p}\ \mathrm{dist}_{S(w)}(\eta).$$

Here we use that clearly $|\eta - q_p| \ge \mathrm{dist}_{S(w)}(\eta)$, since $q_p \in S(w)$ as already recalled.

If $p \in \mathcal{I}_n$, we have two cases. First, if $\mathbb{P}[g(w,\xi) \le q_p] > p$, then since $\mathbb{P}[g(w,\xi) \le q_p] \in \mathcal{I}_n$, we have by definition of $\mathcal{I}_n$, $\mathbb{P}[g(w,\xi)] - p \ge \frac{1}{n}$ and consequently

$$h(w,\eta) \ge (\eta - q_p)\frac{1}{1-p}\left(\mathbb{P}[g(w,\xi) \le q_p] - p\right) \ge \frac{1}{n(1-p)}\ \mathrm{dist}_{S(w)}(\eta).$$

Second, if $\mathbb{P}[g(w,\xi) \le q_p] = p$, then $h(w,\eta) = \frac{1}{1-p}\,\mathbb{E}[(\eta - g(w,\xi))\,\mathbb{1}_{q_p < g(w,\xi)\le\eta}]$. We let q_p^+ be the successor quantile, i.e.,

$$q_p^+ = \inf\{t \ge \mathbb{R} :\ \mathbb{P}[g(w,\xi) \le t] > p\}.$$

Now if $\eta \in (q_p, q_p^+)$, we have $h(w,\eta) = 0$. If $\eta \ge q_p^+$, then

$$\begin{aligned} h(w,\eta) &= \frac{1}{1-p}\,\mathbb{E}\left[(\eta - g(w,\xi))\,\mathbb{1}_{q_p^+ \le g(w,\xi)\le\eta}\right] \\ &\ge (\eta - q_p^+)\frac{\mathbb{P}[g(w,\xi) = q_p^+]}{1-p} \ge \frac{1}{n(1-p)}\,\mathrm{dist}_{S(w)}(\eta) \end{aligned}$$

where the last inequality results from our earlier estimates.

The second case to consider involves the situation $\eta < q_p$. Here, we have:

$$h(w,\eta) = (\eta - q_p)(1 - \frac{1}{1-p}) + \frac{1}{1-p}\,\mathbb{E}\left[(g(w,\xi) - q_p)\,\mathbb{1}_{\eta < g(w,\xi) \le q_p} + (\eta - q_p)\,\mathbb{1}_{g(w,\xi)\le\eta}\right]$$
$$= (\eta - q_p)\left(1 - \frac{1}{1-p} + \frac{1}{1-p}\,\mathbb{P}[g(w,\xi)\le\eta]\right) + \frac{1}{1-p}\,\mathbb{E}\left[(g(w,\xi) - q_p)\,\mathbb{1}_{\eta < g(w,\xi)\le q_p}\right]$$

which leads us to

$$h(w,\eta) = \frac{(q_p - \eta)}{1-p}\left(p - \mathbb{P}[g(w,\xi)\le\eta] - \mathbb{E}\left[\frac{q_p - g(w,\xi)}{q_p - \eta}\mathbb{1}_{\eta < g(w,\xi)\le q_p}\right]\right). \quad (5.13)$$

Let us define the antecessor quantile q_p^- as

$$q_p^- = \max\left\{\sup\{t \ge \mathbb{R} :\ \mathbb{P}[g(w,\xi)\le t] < p\}, \min\{g(w,\xi_i)\}_{i=1}^n - 1\right\}.$$

and note that: $\mathbb{P}[g(w,\xi)\le q_p^-] < p \le \mathbb{P}[g(w,\xi)\le q_p]$. Hence, we have:

$$h(w,\eta) = \frac{(q_p - \eta)}{1-p}\left(p - \mathbb{P}[g(w,\xi)\le\eta] - \mathbb{E}\left[\frac{q_p - g(w,\xi)}{q_p - \eta}\mathbb{1}_{\eta < g(w,\xi)\le q_p^-}\right]\right).$$
$$\ge \frac{(q_p - \eta)}{1-p}\left(p - \mathbb{P}[g(w,\xi)\le \max(\eta, q_p^-)]\right).$$

If $p \notin \mathcal{I}_n$, we get, since $\mathbb{P}[g(w,\xi)\le\max(\eta,q_p^-)] \in \mathcal{I}_n$:

$$h(w,\eta) \ge \frac{\mathrm{dist}_{\mathcal{I}_n}(p)}{1-p}\ \mathrm{dist}_{S(w)}(\eta).$$

If $p \in \mathcal{I}_n$ we note subsequently that $\mathbb{P}[g(w,\xi)\le\max(q_p^-,\eta)] = \mathbb{P}[g(w,\xi)\le q_p^-]$ since $\mathbb{1}\, q_p^- \le g(w,\xi) \le \eta = 0$ under the assumption $\eta \le q_p$. Thus, since $p > \mathbb{P}[g(w,\xi)\le q_p^-] \in \mathcal{I}_n$ we have necessarily $p - \mathbb{P}[g(w,\xi)\le q_p^-] \ge \frac{1}{n}$ which gives:

$$h(w,\eta) \ge \frac{1}{n(1-p)}\ \mathrm{dist}_{S(w)}(\eta).$$

□

Following the terminology of [176], this theorem shows that h is a uniform parametric error bound. We note that the quality of this bound is altered by the number n of data points considered. This drawback actually passes to the limit in the sense that $(w,\eta) \mapsto h(w,\eta)$ fails to be a uniform parametric error bound when ξ follows a continuous distribution; this is an interesting but secondary result that we prove in Section 5.3.4.

5.3.2 *Exact penalization for the hard constraint*

We show here that $(P_{\lambda,\mu})$ is an exact penalization of (P_μ), when λ is large enough. The proof of this result follows usual rationale (see e.g., [31, Prop. 2.4.3]); the main technicality is the sharp growth of h established in Theorem 5.5.

Exact Penalization of the hard constraint

Proposition 5.6. *Let $\mu > 0$ be given and assume that there is a solution to (P_μ) defined in (5.6). Then for any $\lambda > \mu/\delta_p$ with δ_p defined in (5.9), the solution set of (P_μ) coincides with the one of $(P_{\lambda,\mu})$ defined in (5.7).*

Proof. Take $\mu > 0$, define $\lambda_\mu = \mu/\delta_p$, and take $\lambda > \lambda_\mu$ arbitrary but fixed. Let us first take a solution $(w^\star, \eta^\star) \in \mathcal{C} \times \mathbb{R}$ of (P_μ) and show by contradiction that it is also a solution of $(P_{\lambda,\mu})$. Indeed, to the contrary, assume there exists some $\varepsilon > 0$ and $(w', \eta') \in \mathcal{C} \times \mathbb{R}$ such that:

$$f(w') + \mu \max(0, \eta') + \lambda h_{w'}(\eta') \leq f(w^\star) + \mu \max(0, \eta^\star) + \lambda \; h_{w^\star}(\eta^\star) - \varepsilon.$$

Let then $\eta'_p \in S(w')$ be such that : $|\eta'_p - \eta'| \leq \text{dist}_{S(w')}(\eta') + \frac{\varepsilon}{2\mu}$. Then the point (w', η'_p) is a feasible for P_μ (recall $\eta'_p \in S(w')$) and since $\eta \mapsto \mu \max(0, \eta)$ is μ-Lipschitz, we first have

$$\begin{aligned} f(w') + \mu \max(0, \eta'_p) &\leq f(w') + \mu \max(\eta', 0) + \mu |\eta'_p - \eta'| \\ &\leq f(w') + \mu \max(\eta', 0) + \mu \left(\text{dist}_{S(w')}(\eta') + \frac{\varepsilon}{2\mu} \right). \end{aligned}$$

Using Theorem 5.5, we then have

$$\begin{aligned} f(w') + \mu \max(0, \eta'_p) &\leq f(w') + \mu \max(\eta', 0) + \mu \left(\frac{1}{\delta_p} h(w', \eta') + \frac{\varepsilon}{2\mu} \right) \\ &\leq f(w') + \mu \max(\eta', 0) + \lambda_\mu \; h(w', \eta') + \frac{\varepsilon}{2} \\ &\leq f(w^\star) + \mu \max(\eta^\star, 0) - \frac{\varepsilon}{2} \end{aligned}$$

which gives the contradiction. Hence any solution of (P_μ) is also a solution to problem $(P_{\lambda,\mu})$.

Let now $(\bar{w}, \bar{\eta})$ be a solution of $(P_{\lambda,\mu})$ and let us show that it is actually a solution for P_μ. Let again $(w^\star, \eta^\star)$ be an arbitrary solution of (P_μ). We first note that, by the optimality result of $(\bar{x}, \bar{\eta})$ for $(P_{\lambda,\mu})$, we have:

$$f(\bar{w}) + \mu \max(0, \bar{\eta}) + \lambda \underbrace{h(\bar{w}\bar{\eta})}_{\geq 0} \leq f(w^\star) + \mu \max(0, \eta^\star) + \lambda \underbrace{h(w^\star, \eta^\star)}_{=0},$$

which by positivity of the function h and feasibility for (P_μ), i.e., $h(w^\star, \eta^\star) = 0$ of $(w^\star, \eta^\star)$ yields:

$$f(\bar{w}) + \mu \max(0, \bar{\eta}) \leq f(w^\star) + \mu \max(0, \eta^\star).$$

It remains to show that $(\bar{w}\bar{\eta})$ is a feasible point for (P_μ). By the first point, $(w^\star, \eta^\star)$ is both a solution of $(P_{\lambda,\mu})$ and $(P_{\frac{\lambda+\lambda_\mu}{2},\mu})$. Hence, we have:

$$\begin{aligned} f(\bar{w}) + \mu \max(0, \bar{\eta}) + \lambda h(\bar{w}, \bar{\eta}) &\leq f(w^\star) + \mu \max(0, \eta^\star) \\ &= f(w^\star) + \mu \max(0, \eta^\star) + \frac{\lambda + \lambda_\mu}{2} h(w^\star, \eta^\star) \\ &\leq f(\bar{w}) + \mu \max(0, \bar{\eta}) + \frac{\lambda + \lambda_\mu}{2} h(\bar{w}, \bar{\eta}) \end{aligned}$$

But since $\lambda > \lambda_\mu$ we necessarily have: $h(\bar{w}, \bar{\eta}) = 0$ which implies by the properties of the value function that $(\bar{w}, \bar{\eta})$ is a feasible point for (P_μ). □

We note that the above result is a special case of theorem [176, Theorem 2.6] which is meant for generalized bilevel programs. Based on the terminology of [176], we have shown that P_μ satisfies the partial calmness property, as the value function h was shown to be a uniform parametric error bound.

5.3.3 *Double penalization scheme*

From the previous results, we get that solving the sequence of penalized problems gives approximations of the solution of the initial problem. We formalize this in the next proposition suited for our context of double penalization. The proof of this result follows standard arguments; see e.g. [105, Ch. 13.1].

Penalization of the easy constraint

Proposition 5.7. *Assume that Problem (5.4) has a non-empty feasible set. Let $(\mu_k)_{k\geq 0}$ be an increasing sequence such that $\mu_k \nearrow \infty$, and $(\lambda_k)_{k\geq 0}$ be taken such that $\lambda_k > \frac{\mu_k}{\delta_p}$ with δ_p as defined in (5.9). If, for all k, there exists a solution of (P_{λ_k,μ_k}) (denoted by (w_k, η_k)), then any cluster point of the sequence (w_k, η_k) is an optimal solution of (5.1).*

Proof. The fact that (w_k, η_k) is an optimal solution of (P_{λ_k,μ_k}) implies that

$$\begin{aligned} f(w_k) + \mu_k \max(0, \eta_k) + \lambda_k h(w_k, \eta_k) \\ \leq f(w_{k+1}) + \mu_k \max(0, \eta_{k+1}) + \lambda_k h(w_{k+1}, \eta_{k+1}) \end{aligned} \tag{5.14}$$

Similarly for (w_{k+1}, η_{k+1}), we get

$$\begin{aligned} f(w_{k+1}) + \mu_{k+1} \max(0, \eta_{k+1}) + \lambda_{k+1} h(w_{k+1}, \eta_{k+1}) \\ \leq f(w_k) + \mu_{k+1} \max(0, \eta_k) + \lambda_{k+1} h(w_k, \eta_k). \end{aligned}$$

By Proposition 5.6, η_k (resp. η_{k+1}) is feasible for (P_{μ_k}) (resp. $(P_{\mu_{k+1}})$); in other words, we have $h(w_k, \eta_k) = h(w_{k+1}, \eta_{k+1}) = 0$. Hence summing up these two inequalities yields

$$\max(\eta_k, 0) \geq \max(\eta_{k+1}, 0).$$

Using this last inequality with (5.14) gives:

$$f(w_k) - f(w_{k+1}) \leq \mu_k \left(\max(\eta_{k+1}, 0) - \max(\eta_k, 0)\right) \leq 0,$$

and as a consequence the sequence $\{f(w_k)\}_{k\geq 0}$ increases. Let (w', η') be an arbitrary feasible solution for (P). By definition of the sequence (w_k, η_k), for any $k \in \mathbb{N}$, we have:

$$f(w_k) \leq f(w_k) + \mu_k \max(\eta_k, 0) \leq f(w') + \mu_k \max(\eta', 0) \leq f(w'). \tag{5.15}$$

Therefore for any cluster point $(\bar{w}, \bar{\eta})$ of the sequence $\{(w_k, \eta_k)\}_{k\geq 0}$, we have $f(\bar{w}) \leq f(w')$. In order to show that $(\bar{w}, \bar{\eta})$ is a a solution of (5.4), it remains to show its feasibility. With the right hand side inequality of (5.15), we obtain

$$\max(\eta_k, 0) \leq \frac{f(w') - f(w_k)}{\mu_k} \leq \frac{f(w') - f(w_0)}{\mu_k} \xrightarrow[k\to\infty]{} 0,$$

so that we may deduce that, $\bar{\eta} \leq 0$. Moreover, continuity of h ensures that $h(\bar{x}, \bar{\eta}) = 0$ which completes the proof. □

In words, cluster points of a sequence of solutions obtained as μ grows to $+\infty$ are feasible solutions of the initial chance-constrained problem. In practice though, we have observed that taking a fixed μ is enough for reaching good approximations of the solution with increasing λ's; see in particular the numerical experiments of Section 5.5. In the next section, we discuss further the practical implementation of the conceptual double penalization scheme.

5.3.4 *Uniform bound at the limit*

We show here that the uniform error bound derived in Section 5.3.1 vanishes at the limiting case of continuous distributions. We assume that, for a fixed $x \in \mathbb{R}^{\mathrm{d}}$, the random variable $g(w,\xi)$ has a continuous density $f_{w,\xi} : \mathbb{R} \to \mathbb{R}$ denoted by $f_{w,\xi}$: we have, for all $a \leq b$,

$$\mathbb{P}[a \leq g(w,\xi) \leq b] = \int_a^b f_{w,\xi}(t)\,\mathrm{d}\,t.$$

Weak sharpness vanishes with sample size

Proposition 5.8. *Fix $x \in \mathbb{R}^d$ and denote by q_p the p-quantile of the distribution followed by the random variable $g(w,\xi)$. If $g(w,\xi)$ has a continuous density, then the value function $\eta \mapsto h(w,\eta)$ defined in (5.11) is differentiable at $\eta = q_p$ (with $h'(w,q_p) = 0$).*

Proof. We first note that the existence of a density ensures the continuity of the cumulative distribution function of $g(w,\xi)$, which in turns implies $\mathbb{P}[g(w,\xi) \leq q_p] = p$. Let us now come back to expressions established in the proof of Theorem 5.5. From (5.12), we have, for $\eta > q_p$,

$$\begin{aligned} h(w,\eta) &= (\eta - q_p)\frac{1}{1-p}\left(\mathbb{P}[g(w,\xi) \leq q_p] - p + \mathbb{E}\left[\frac{\eta - g(w,\xi)}{\eta - q_p}\,\mathbb{1}_{q_p < g(w,\xi) \leq \eta}\right]\right) \\ &= \frac{1}{1-p}\,\mathbb{E}\left[(\eta - g(w,\xi))\,\mathbb{1}_{q_p < g(w,\xi) \leq \eta}\right] \;=\; \frac{1}{1-p}\int_{q_p}^{\eta}(\eta - t) f_{w,\xi}(t)\,\mathrm{d}\,t \\ &= \frac{1}{1-p}\left(\eta\int_{q_p}^{\eta} f_{w,\xi}(t)\,\mathrm{d}\,t - \int_{q_p}^{\eta} t\, f_{w,\xi}(t)\,\mathrm{d}\,t\right). \end{aligned}$$

By continuity of the above integrands, we can use the fundamental theorem of calculus to get that $h(w,\cdot)$ admits a right derivative at $\eta = q_p$ such that

$$\begin{aligned} h'_+(w,\eta) &= \lim_{\substack{\eta \to q_p \\ \eta > q_p}} \frac{h(w,\eta) - h(w,q_p)}{\eta - q_p} \\ &= \lim_{\substack{\eta \to q_p \\ \eta > q_p}} \frac{1}{1-p}\left(\eta\frac{\int_{q_p}^{\eta} f_{w,\xi}(t)\,\mathrm{d}\,t}{\eta - q_p} - \frac{\int_{q_p}^{\eta} t\, f_{w,\xi}(t)\,\mathrm{d}\,t}{\eta - q_p}\right) \\ &= \lim_{\substack{\eta \to q_p \\ \eta > q_p}} \frac{1}{1-p}\left(\eta f_{w,\xi}(q_p) - q_p f_{w,\xi}(q_p)\right) \;=\; 0\,. \end{aligned}$$

For the case $\eta < q_p$, we have from (5.13), together with $\mathbb{P}[g(w,\xi) = q_p] = 0$:

$$\begin{aligned} h(w,\eta) &= (q_p - \eta)\frac{1}{1-p}\left(\mathbb{E}\left[1 - \frac{(q_p - g(w,\xi))}{q_p - \eta}\,\mathbb{1}_{\eta < g(w,\xi) < q_p}\right] + \mathbb{P}[g(w,\xi) = q_p]\right) \\ &= \frac{1}{1-p}\left((\eta - q_p)\int_{\eta}^{q_p} f_{w,\xi}(t)\,\mathrm{d}\,t - \int_{\eta}^{q_p}(q_p - t) f_{w,\xi}(t))\,\mathrm{d}\,t\right). \end{aligned}$$

Using again to the fundamental theorem of calculus, we get that $h(w,\cdot)$ admits a left derivative at $\eta = q_p$ with:

$$h'_-(w,\eta) = \lim_{\substack{\eta \to q_p \\ \eta < q_p}} \frac{h(w,\eta) - h(w,q_p)}{\eta - q_p}$$
$$= \lim_{\substack{\eta \to q_p \\ \eta < q_p}} \frac{1}{1-p}\left((\eta - q_p)\frac{\int_\eta^{q_p} f_{w,\xi}(t)\,\mathrm{d}\,t}{\eta - q_p} - \frac{\int_\eta^{q_p}(q_p - t) f_{w,\xi}(t)\,\mathrm{d}\,t}{\eta - q_p}\right) = 0\,.$$

We can conclude that $h(w,\cdot)$ is differentiable at q_p with zero as derivative. □

5.4 DOUBLE PENALIZATION IN PRACTICE

In this section, we propose a practical version of the double penalization scheme for solving chance-constrained optimization problems. First, we present in Section 5.4.1 how to tackle the inner penalized problem $(P_{\lambda,\mu})$ by leveraging its difference-of-convex (DC) structure. Then we quickly describe, in Section 5.4.2, the python toolbox that we release, implementing this bundle algorithm and efficient oracles within the double penalization method.

5.4.1 *Solving penalized problems by a bundle algorithm*

We discuss here an algorithm for solving $(P_{\lambda,\mu})$ by revealing the DC structure of the objective function. Notice indeed that, introducing the two convex functions

$$\varphi_1(w,\eta) = f(w) + \lambda G(w,\eta) + \mu \max(\eta, 0) \quad \text{and} \quad \varphi_2(w,\eta) = \lambda \min_{s\in\mathbb{R}} G(w,s)$$

we can write $(P_{\lambda,\mu})$ as the DC problem

$$\min_{(w,\eta)\in\mathcal{C}\times\mathbb{R}} \varphi(w,\eta) = \varphi_1(w,\eta) - \varphi_2(w,\eta). \tag{5.16}$$

We then propose to solve this problem by the bundle algorithm of [36], which showed to be a method of choice for DC problems. This bundle algorithm interacts with first-order oracles for φ_1 and φ_2; in our situation, there exist computational procedures to compute subgradients of φ_1 and φ_2 from output of oracles of f and g, as formalized in the next proposition. Note that at the price of more heavy expressions, we could derive the whole subdifferential.

Oracle derivations

Proposition 5.9. *Let $(w,\eta) \in \mathcal{C}\times\mathbb{R}$ be fixed. Let s_f be a subgradient of f at w and $s_{g_1},\dots,s_{g_n}$ be respective subgradients of $g(\cdot,\xi_1),\dots,g(\cdot,\xi_n)$ at w. For a given $t\in\mathbb{R}$, denote by $I_{>t}$ the set of indices such that $g(w,\xi_i) > t$ and by $I_{=t}$ the set of indices such that $g(w,\xi_i) = t$. Let finally $\alpha = \frac{\mathbb{P}[g(w,\xi)\le Q_p(g(w,\xi))]-p}{\#(I_{=Q_p(g(w,\xi))})}$. Then, s_{φ_1} and s_{φ_2} defined as:*

$$s_{\varphi_1} = \left(s_f + \frac{\lambda}{n(1-p)} \sum_{i\in I_{>\eta}}^{n} s_{g_i}\,,\ 1 + \mu\,\mathbb{1}_{\eta>0} - \lambda\frac{\#(I_{>\eta})}{n(1-p)} \right)$$

$$s_{\varphi_2} = \left(\frac{\lambda}{n(1-p)} \left(\sum_{i\in I_{>Q_p(g(w,\xi))}} s_{g_i} + \alpha \sum_{i\in I_{=Q_p(g(w,\xi))}} s_{g_i} \right),\ 0 \right)$$

are respectively subgradients of φ_1 and φ_2 at (w,η).

Notice now that the convergence result for the bundle algorithm [36, Th. 1] guarantees convergence towards a point $\bar{u} = (\bar{x}, \bar{\eta})$ satisfying

$$\partial\varphi_2(\bar{u}) \cap \partial\varphi_1(\bar{u}) \neq \emptyset, \tag{5.17}$$

which is a weak notion of criticality. Thus, we propose to furthermore replace φ_2 in (5.16) by a smooth approximation of it, denoted by $\widetilde{\varphi}_2$. The reason is that the bundle method minimizing $\widetilde{\varphi} = \varphi_1 - \widetilde{\varphi}_2$ then reaches a Clarke-stationary point: indeed, (5.17) reads $\nabla\widetilde{\varphi}_2(\bar{u}) \subset \partial\varphi_1(\bar{u})$, which gives $0 \in \partial\varphi(\bar{u}) = \partial\varphi_1(\bar{u}) + \nabla\widetilde{\varphi}_2(\bar{u})$, i.e., that $\bar{u}$ is Clarke-stationary (for the smoothed problem). To smooth φ_2, we use the efficient smoothing procedure provided in the Section 4.2.2 of this thesis with the euclidean smoothing. This yields a smooth approximation $\widetilde{\varphi}_2$ of the form,

$$\widetilde{\varphi}_2(w,\eta) = \lambda \sup_{\substack{0\le q_i\le \frac{1}{n(1-p)} \\ q_1+\cdots+q_n=1}} \sum_{i=1}^{n} \left\{ q_i\, g(w,\xi_i) - \frac{\rho}{2}(q_i - \tfrac{1}{n})^2 \right\} \tag{5.18}$$

We provide now a direct proof of the subgradient expressions of Proposition 5.9.

Oracles subgradient computations. Let $(w,\eta) \in \mathcal{C} \times \mathbb{R}$ be fixed, and consider first the case of φ_1. For $i \in \{1,\dots,n\}$, by successive applications of Theorems 4.1.1 and 4.4.2 from [63, Chap. D] to the functions

$$\varphi_1^{(i)} : (w,\eta) \mapsto \frac{1}{n}\left[f(w) + \mu \max(\eta, 0) + \lambda \left(\eta + \frac{1}{1-p} \max(g(w,\xi_i) - \eta, 0) \right) \right]$$

we get for any $i \in \{1,\dots,n\}$

$$\begin{aligned} \frac{1}{n} s_f + \frac{\lambda}{n(1-p)} \mathbb{1}_{g(w,\xi_i)>\eta}\, s_g &\in \partial_w \varphi_1^i(w,\eta) \\ \frac{\mu}{n}\mathbb{1}_{\eta>0} + \frac{\lambda}{n} - \frac{\lambda}{n(1-p)} \mathbb{1}_{g(w,\xi_i)>\eta} &\in \partial_\eta \varphi_1^i(w,\eta). \end{aligned}$$

Since $\varphi_1 = \sum_{i=1}^n \varphi_1^{(i)}$, we thus have

$$\left(s_f + \frac{\lambda}{n(1-p)} \sum_{i \in I_{>\eta}} s_{g_i}\,,\ \mu\,\mathbb{1}_{\eta>0} + \lambda - \lambda \frac{\#(I_{>\eta})}{n(1-p)} \right) \in \partial\varphi_1(w,\eta)$$

For φ_2 we need first the whole subdifferential of the function G, which, using above mentioned properties, writes

$$\begin{aligned} \partial G(w,\eta) = \Bigg\{ \Bigg(& \frac{1}{1-p} \sum_{i=1}^{n} \frac{s_{g_i}}{n} (\mathbb{1}_{g(w,\xi_i)>\eta} + \beta_i\, \mathbb{1}_{g(w,\xi_i)=\eta}), \\ & 1 - \frac{1}{1-p} \sum_{i=1}^{n} \frac{1}{n} (\mathbb{1}_{g(w,\xi_i)>\eta} + \beta_i\, \mathbb{1}_{g(w,\xi_i)=\eta}) \Bigg),\ \beta_i \in [0,1], \quad \forall i \in \{1,\dots,n\} \Bigg\}. \end{aligned}$$

By taking $\beta_i = \alpha$ (for all $i \in \{1,\dots,n\}$) with the specific α given in the statement, we can zero the second term in the above expression.

Now since $\varphi_2(w,\eta) = \lambda \min_{s\in\mathbb{R}} G(w,s)$ with $Q_p(g(w,\xi)) \in \arg\min_{s\in\mathbb{R}} G(w,s)$, we apply Corollary 4.5.3 of [63, Chap. D] to obtain a subgradient of φ_2:

$$s_{\varphi_2} = \left(\frac{\lambda}{n(1-p)} \left(\sum_{i\in I_{>Q_p(g(w,\xi)}} s_{g_i} + \alpha \sum_{i\in I_{=Q_p(g(w,\xi))}} s_{g_i} \right), 0 \right)$$

which completes the proof. □

5.4.2 *TACO: A python toolbox for chance-constrained problems*

We release `TACO`, an open-source python toolbox for solving chance constrained optimization problems (5.1). The toolbox implements the penalization approach outlined in Section 5.3 together with the bundle method [36] for the inner penalized subproblems. `TACO` routines rely on just-in-time compilation supported by Numba [93]. The routines are optimized to provide fast performances on reasonably large datasets. Documentation is available at

https://yassine-laguel.github.io/taco.

We provide here basic information on `TACO`; for further information, we refer to the online documentation.

The python class `Problem` wraps up all information about the problem to be solved. This class possesses an attribute `data` which contains the values of ξ and is formatted as a `numpy` array in 64-bit float precision. The class also implements two methods giving first-order oracles: `objective_func` and `objective_grad` for the objective function f, and `constraint_func` and `constraint_grad` for the constraint function g.

Input format

Let us take a simple quadratic problem in $\mathbb{R}^2$ to illustrate the instantiation of a problem. We consider

$$\begin{aligned} &\min_{w\in\mathbb{R}^2} \ \|w-a\|^2 && a = [1.0, 2.0]^\top \\ &\text{s.t.} \ \ \mathbb{P}[w^\top \xi \le 0] \ge 0.9, && \text{with 1000 samples of } \xi \sim \mathcal{N}(0,1). \end{aligned}$$

The instance of `Problem` is in this case:

```
import numpy as np
class Problem:
    def __init__(self, dim=2, sample_size=1000):
        self.data = np.random.normal(size=(sample_size, dim),
            dtype=np.float64)
        self.a = np.array([1.0, 2.0], dtypte=np.float64)
    def objective_fun(self,\variablemodel):
        return np.dot(\variablemodel-self.a,x-self.a)
    def objective_grad(self, \variablemodel):
        return \variablemodel
    def constraint_func(self, \variablemodel, z):
        return np.dot(\variablemodel,z)
    def constraint_grad(self, \variablemodel, z)
        return z
problem = Problem()
```

The `Optimizer` object

`TACO` handles the optimization process with a python class named `Optimizer`. Given an instance of Problem and hyper-parameters provided by the user, the class Optimizer runs an implementation of the bundle method of [36] on the

penalized problem (5.7). The toolbox gives the option to update the penalization parameters μ, λ along the running process to escape possible stationary points for the DC objective that are non-feasible for the chance constraint.

```
from taco import Optimizer
problem = Problem()
optimizer = Optimizer(problem, p=0.9, starting_point=np.zeros(2,
    dtype=np.float64), pen1=1.0, pen2=10.0)
sol = optimizer.run()
```

Customization

Customizable parameters are stored in a python dictionary, called `params`, and designed as an attribute of the class `Optimizer`. The main parameters to tune are: the safety level of probability `p`, the starting penalization parameters $\mu =$ `pen1` and $\lambda =$ `pen2`, the starting point of the algorithm and the starting value for the proximal parameter of the bundle method. We note that often a tuning of both starting penalization parameters may be required to get a satisfying solution for the problem considered. See for instance the experimental setup of our numerical illustations in 5.5.2. Others parameters are filled with default values when instantiating an `Optimizer`; for instance:

```
custom_options = {
    'p': 0.9,
    'pen1': 1.0,
    'pen2': 10.0,
    'bund_mu_start': 50.0,
    'bund_max_size_bundle_set': 30,
}
custom_optimizer = Optimizer(problem, params=custom_options)
```

Some important parameters (such as the safety probability level, or the starting penalization parameters) may also be given directly to the constructor of the class `Optimizer`, when instantiating the object; as in the first example.

IMPLEMENTATION DETAILS ON TACO.

Penalization procedure

Further customization. `TACO` relies on a set of hyperparameters to be provided by the user and specified in a single dictionnary passed as an argument of the class `Optimizer`. There are two families of parameters to be specified. First, the parameters concerning the oracles φ_1 and φ_2. These are the starting penalization parameters λ and μ, the multiplicative factors to increment them along the penalization process, and the smoothing parameter of $\tilde{\varphi}_2$.

Parameters of the bundle method

The second family of parameters concerns the bundle method. It gathers the proximal parameters of the bundle method, the precision targeted, the starting point of the algorithm, the maximal size of the bundle information, and parameters related used when restarting the bundle method (see more in the following section). Overall the most important parameters to specify are the starting penalization parameters μ and λ with respective keys `'pen1'` and `'pen2'` and the starting proximal parameter of the bundle algorithm. In the toolbox, we provide the set of parameters used in our numerical experiments. In addition of the final solution, it is possible to log the iterates, function values and time values, by calling the method with the option `logs=True`. The `verbose=True` option also allows the user to observe in real time the progression of the algorithm along the iterations.

Finally we underline that `TACO` subroutines rely on just-in-time compilation supported by `Numba`, which consistently improves the running time. Further

improvements can be achieved when the instance considered can be cast as a `Numba jitclass`. The parameter `'numba'` in the input dictionnary of the associated `Optimizer` object should then be set to `True`.

On the bundle algorithm. We give now some information on our implementation of the bundle algorithm of [36] to tackle the double penalized problem $(P_{\lambda,\mu})$ written as a DC problem. We discuss the parameters used at various steps of the procedure. We refer to [36] for more details.

- **Overall run**: The starting point, the maximum number of iterations as well as the precision tolerance for termination may be set by the user.

- **Subproblems**: Each iteration of the bundle algorithm requires solving a quadratic subproblem (see [36, Equation (9)]), for which we use the solver `cvxopt` [167] by simplicity.

- **Stabilization center**: Whenever the solution of a subproblem satisfies a sufficient decrease in terms a function value, it is considered as a new stability center. The condition to qualify sufficient decrease is given in [36, Equation (12)]. It involves a constant κ which may be tuned by the user.

- **Proximal parameters**: The initial value of the proximal parameter involved in quadratic subproblems can be set by the user. The user can also specify upper and lower acceptance bounds for it. After each iteration, the prox-parameter is updated: it is increased by a constant factor in case of serious step, and decreased otherwise. Both factors can be tuned by the user.

- **Bundle information**: The bundle of cutting-planes is augmented after each null step with new linearization, and emptied after each serious step. We fix a maximum size for the bundle: above this parameter, the bundle is emptied and proximal parameter is restarted to a specified restarting value. When the bundle is emptied, we have the chance of a specific improvement: if the stability center is feasible in the chance-constraint, we replace the coordinate playing the role of η by the p-quantile of $g(w, \xi)$, thus decrease the objective function.

- **Termination Criteria**: We use a simple stopping criteria: we stop when the euclidean distance between the current iterate and the current stability center falls below a certain threshold specified by the user.

5.5 NUMERICAL ILLUSTRATIONS

We illustrate our double penalisation approach implemented in the toolbox `TACO` on two problems: a 2-dimensional quadratic problem with a non-convex chance constraint (in Section 5.5.1), and a family of problems with explicit solutions (in Section 5.5.2). These proof-of-concept experiments are not meant to be extensive but to show that our approach is viable. These experiments are reproducible: the experimental framework is available on the toolbox's website.

5.5.1 *Visualization of convergence on a 2D problem*

Instance of eventually convex problem

We consider a two-dimensional toy quadratic problem in order to track the

convergence of the iterates on the sublevel sets. We take [92, Ex. 4.1] which considers an instance of problem (5.1) with

$$
\begin{aligned}
f(w) &= \frac{1}{2}(w-a)^\top Q(w-a) \quad \text{with } a = \begin{pmatrix} 2. \\ 2. \end{pmatrix}, Q = \begin{pmatrix} 5.5 & 4.5 \\ 4.5 & 5.5 \end{pmatrix} \\
g(w,z) &= z^\top W(w) z + w^\top z \quad \text{with } W(w) = \begin{pmatrix} w_1^2 + 0.5 & 0. \\ 0. & |w_2 - 1|^3 + 1 \end{pmatrix} \\
\xi &\sim \mathcal{N}(\mu, \Sigma) \qquad 10^4 \text{ samplings with } \mu = \begin{pmatrix} 1. \\ 1. \end{pmatrix}, \Sigma = \begin{pmatrix} 20. & 0. \\ 0. & 20. \end{pmatrix}.
\end{aligned}
\tag{5.19}
$$

For this example, [92] shows that the chance constraint is convex for large enough probability levels, but here we take a low probability level $p = 0.008$ to have a non-convex chance-constraint. We can see this on Figure 5.2, ploting the level sets of the objective function and the constraint function: the chance-constrained region for $p = 0.008$ is delimited by a black dashed line; the optimal value of this problem is located at the star.

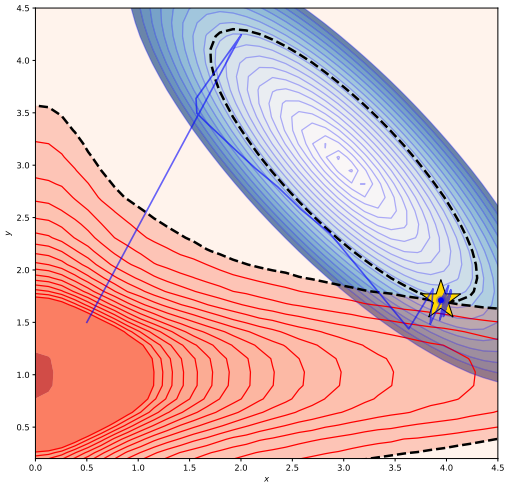


Figure 5.2: Trajectory of the iterates (in blue) on the plot of the level sets of the chance-constraint and the objective for the 2d problem with data (5.19).

Setup

Experimental setting. For this 2d problem, we use the starting point $x = (0.5, 1.5)$ (and $\eta = 0.01$) well-inside the chance-constraint. The initial penalization parameters μ and λ are respectively initialized to 400 and 600. The initial proximal parameter is fixed to 38.0 with lower and upper acceptance bounds set to 10^{-3} and 10^3. Increasing and decreasing factors for this parameter are fixed to 1.05 and 0.95. The classification rule parameter is set to 10^{-4}. The maximal size of the information bundle is set to 20 and the threshold of the termination criteria is set to 10^{-7}.

Visualization of the convergence

Results. We plot on the sublevel sets of Figure 5.2 the path (in deep blue) taken by the sequence of iterates starting from the point $[0.5, 1.5]$ moving towards the solution. We observe that the sequence of iterates, after a exploration of the functions landscape, gets rapidly close to the optimal solution. At the end of the convergence, we also see a zigzag behaviour around the frontier of the chance constraint.This can be explained by the penalization term which is activated asymptotically whenever the sequence gets out of the chance constraint.

5.5.2 *Experiments on a family of problems*

Problems in larger dimension

We consider the family of d-dimensional norm problems of [66, section 5.1]. For a given dimension d, the problem writes as an instance of (5.1) with

$$f(w) = -\|x\|_1 \qquad \text{and} \qquad g(w, Z) = \max_{i \in \{1,\dots,10\}} \sum_{j=1}^{d} Z_{i,j}^2 w_j^2 - 100 \tag{5.20}$$

and ξ is random matrix of dimensions $10 \times d$ statisfying for all i, j, $\xi_{i,j} \sim \mathcal{N}(0, 1)$. The interest of this family of problems is that they have explicit solutions: for given d, the optimal value is

$$f^\star = -\frac{10\, d}{\sqrt{F^{(-1)}_{\chi^2_d}(p^{\frac{1}{10}})}}$$

where $F_{\chi^2_d}$ is χ^2 cumulative distribution with d degrees of freedom. We consider four instances of this problems with dimension d from 2 to 200 and the safety probability threshold p set to 0.8. We consider the case of the rich information on uncertainty: ξ is sampled 10000 times. In this case, a direct approach consisting in solving the standard mixed-integer quadratic reformulations (see e.g. [3]) with efficient MINLP solvers (we used `Juniper` [82]) does not provide reasonable solutions; see basic information in forthcoming Section 5.5.3.

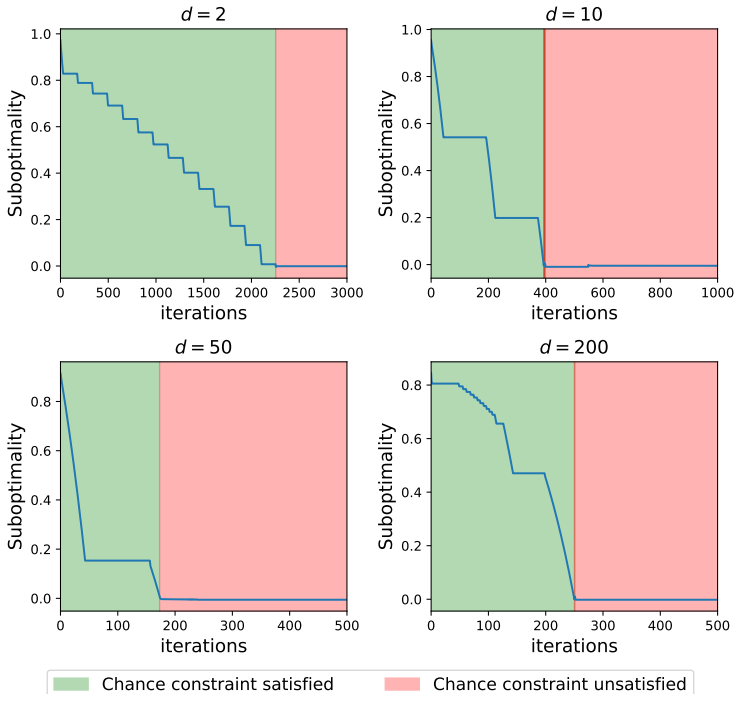


Figure 5.3: Convergence of the algorithm on four norm problems (5.20) with $d = 2, 10.50, 200$.

Setup

Experimental Setting 5.5.2. For any fixed dimension d comprised in $\{2, 10, 50, 200\}$, the algorithm is run from the starting point $(0.1, \dots, 0.1) \in \mathbb{R}^{d+1}$. The starting penalization parameter μ, constant for the 4 instances, is set to $\mu = 10.0$. We

tuned the second penalization parameter λ along problems: we observed that $\lambda = \{1.75, 1.5, 1.5, 2.0\}$ give good performances for the considered problems. The starting proximal parameters is fixed to 60.0 with lower and upper acceptance bounds set to 10^{-4} and 10^{5} respectively. Increasing and decreasing factors for the proximal parameter are fixed to 1.01 and 0.99. The classification rule parameter is set to 10^{-4}. The maximal size of the information bundle is set to 300.

Analysis of the convergence

Results. Figure 5.3 plots the relative suboptimality

$$(f(w_k) - f^\star)/|f^\star|$$

along iterations. The green (resp. red) regions represent iterates that, respectively, satisfy (resp. do not satisfy) the chance constraint.

In the four instances, we take a first iterate well inside the feasible region. We observe an initial decrease of the objective function down to optimal value. Then the chance constraint starts to be violated only when this threshold is reached, and the last part of convergence deals with local improvement of precision and feasibility.

Table 5.1 reports the final suboptimality and satisfaction of the probabilistic constraint. The probability constraint is evaluated for 100 sampled points from of the total $N = 10000$ points. We give the resulting probability; the standard deviation is 0.004 for the four instances.

Dimension	Suboptimality	$\mathbb{P}[g(w,\xi) \leq 0]$
$d = 2$	8.9×10^{-4}	0.799
$d = 10$	5.0×10^{-3}	0.787
$d = 50$	5.6×10^{-3}	0.769
$d = 200$	1.8×10^{-3}	0.781

Table 5.1: Final suboptimality and feasibility for (5.20) (where $p = 0.8$).

We observe that the algorithm reaches an accuracy of order of 10^{-3}. Regarding satisfaction of the constraint $\mathbb{P}[g(w,\xi) \leq 0] \geq 0.8$, it is achieved to a 10^{-4} precision for $d = 2$ but it slightly degrades as the dimension grows.

5.5.3 *Limitations of MINLP approach*

Standard MINLP reformulation

We finish this chapter with a remark on Mixed Integer Non-Linear Programming (MINLP) approaches. Mixed-integer reformulation approaches (see e.g. [3]) are often considered as the state-of-the-art to solve chance constrained optimization problems by sample average approximation. Applying directly such a reformulation to Problem (5.20) in Section 5.5.2 leads to the equivalent mixed integer quadratic program:

$$\begin{aligned} \min_{w\in\mathbb{R}^d,\, z\in\{0,1\}^N} \quad & -\sum_{i=1}^{d} w_i \\ \text{s.t.} \quad & \sum_{k=1}^{d} (\xi_i)_{j,k}^2 w_k^2 - 100 \leq M\, z_i, \quad \forall i \in [\![1, N]\!],\ \forall j [\![1, 10]\!] \\ & \sum_{i=1}^{N} z_i \leq pN, \ \ w \geq 0. \end{aligned}$$

where M is a large "big-M" constant. In our setting, such formulation involves $10 \times N = 100000$ quadratic constraint involving binary variables. We were not able to solve the resulting mixed-integer problem in reasonable times using the MINLP solver `Juniper` [82] (that is based on Ipopt and JuMP). This shows that a direct application of reformulation techniques combined with reliable software failed on this problem – in contrast with our approach.

5.6 CONCLUSION

In this chapter, we uncovered a new formulation of chance-constrained problems based on the variational formulation of the superquantile. This reformulation takes the form of a bilevel program with convex upper-level and lower-level problems. We proposed a double penalization procedure to solve this bilevel program. We analyzed and exploited the weak sharpness property of the superquantile to show that exact penalization may be achieved for the hard constraint of this problem. The objective of the penalized problem being a Difference of Convex (DC) function, we proposed to address it with a recent proximal bundle method. In order to ensure better convergence properties of the method, we proposed to smooth the superquantile term involved in the DC objective. We made available an open-source python toolbox with fast computational procedures to solve chance-constrained problems with this approach together with numerical illustrations showing its interest.

6

HANDLING HETEROGENEITY IN FEDERATED LEARNING

This chapter is devoted to the learning of models in heterogeneous distributed environments.

The developments laid down below build upon the following work:

- Y. Laguel*, K. Pillutla*, J. Malick and Z. Harchaoui. A superquantile approach to Federated Learning with heterogenuous devices. Proceedings of the 55^{th} Annual Conference on Information Sciences and Systems (CISS 2021), 2021.

- Y. Laguel*, K. Pillutla*, J. Malick and Z. Harchaoui. Federated Learning with Heterogeneous Devices:A Superquantile Optimization Approach. Submitted to the Journal of Selected Topics in Signal Processing, for the special issue on distributed machine learning for wireless communication, 2021.

* *denotes equal contributions.*

6.1 INTRODUCTION

Federated learning is a distributed optimization paradigm with a central server and a large number of clients. An introduction and references are provided in the Section 2.2.2 of this thesis.

In federated learning statistical heterogeneity is a key feature: client data distributions are *not* identical. Each user has unique characteristics which are reflected in the data they generate. These characteristics are influenced by personal, cultural, and geographical factors. For instance, the varied use of language contributes to data heterogeneity in a next word prediction task.

We present in this chapter a robust approach to federated learning that guarantees a minimal level of predictive performance to all devices even in situations where the population is heterogeneous. Usual algorithms for federated learning such as *FedAvg* [109] seek to minimize the prediction error on average over the population of devices available for training. While this approach can be effective in terms of predictive performance for each device whose local data is close to the average distribution, it is liable to fail on devices which are far from this distribution. The approach we develop addresses these issues by minimizing a learning objective based on the superquantile, which was introduced in the Section 3.3 of this thesis.

This approach, coined Δ-*FL*, allows one to control higher percentiles of the distribution of errors over the (possibly heterogeneous) population of devices. We shall show in the experiments that our approach is more efficient than a direct approach simply seeking to minimize the worst error over the population of devices. Compared to *FedAvg*, Δ-*FL* delivers improved prediction to data-poor or non-conforming devices. We present theoretical convergence guarantees for

the Δ-*FL* algorithm and show how to implement it in a way that is compatible with basic federated learning ingredients.

Outline

In Section 6.2, we start with a formal description of the vanilla federated framework and introduce the concept of *conformity* to capture its potential statistical heterogeneity. Section 6.3 describes Δ-*FL*, the new framework we propose to better handle non-conforming users together with a practical privacy-compliant algorithm. In Section 6.4, we provide a convergence analysis of our algorithm with a convergence rate in the strongly convex setting. Section 6.5 presents experimental results, comparing our proposed approach to existing ones, on benchmark datasets for federated learning.

6.2 PROBLEM SETTING

The federated learning paradigm, introduced in the Section 2.2.2 of this thesis, calls for optimization methods that are compliant with its specific constraints. We begin this Section with a brief presentation of *FedAvg*, the standard baseline in federated learning. We introduce then in Section 6.2.2 the concept of conformity to describe the statistical heterogeneity of a given federated framework.

6.2.1 *Vanilla federated learning and FedAvg*

Federated learning consists of heterogeneous client devices which collaboratively train a machine learning model under the orchestration of a central server. The model is then deployed on all devices, including those not seen during training.

Vanilla federated learning objective

Analogous to the classical expectation-based objective function in empirical risk minimization approach, the standard objective in federated learning is to minimize the average loss on the training devices

$$\min_{w\in\mathbb{R}^d} \sum_{k=1}^{N} \alpha_k F_k(w) + \frac{\lambda}{2}\|w\|^2 \,, \tag{6.1}$$

where the weight α_k of training device k is chosen and $\lambda > 0$ is a regularization parameter. As mentioned in Section 2.2.2, each loss $F_k : w \mapsto \mathbb{E}_{\xi\sim\xi_k}\left[f(w,\xi)\right]$ evaluates the performance of the model w on the device k, hosting the distribution q_k. We assume that $\sum_{k=1}^{N} \alpha_k = 1$ w.l.o.g. We call this objective the *vanilla FL* objective.

FedAvg

The standard training algorithm to solve (6.1) is *FedAvg* [109]. We illustrate it in Figure 6.2. Each round of the algorithm consists in following steps:

1- The server samples a set S of m devices from $[\![1, N]\!]$ and broadcasts the current model $w^{(t)}$ to these devices.

2- Staring from $w_{k,0}^{(t)} = w^{(t)}$, each device $k \in S$ makes τ local gradient or stochastic gradient descent steps[1] with a learning rate γ:

$$w_{k,j+1}^{(t)} = w_{k,j}^{(t)} - \gamma \nabla F_k(w_{k,j}^{(t)}) \,.$$

3- The models from the selected devices are sent to the server and aggregated to update the server model

$$w^{(t+1)} = \frac{\sum_{k\in S} \alpha_k w_{k,\tau}^{(t)}}{\sum_{k\in S} \alpha_k} \,.$$

1 For simplicity, we consider full gradient steps on each device.

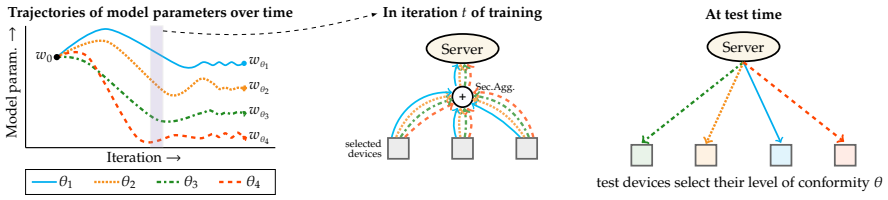


Figure 6.1: Schematic summary of the Δ-*FL* framework. **Left**: The server maintains multiple models w_{θ_j}, one for each level of conformity θ_j. **Middle**: During training, selected devices participate in training *each* model w_{θ_j}. Individual updates are securely aggregated to update the server model. **Right**: Each test user is allowed to select their level of conformity θ, and are served the corresponding model w_θ.

FedAvg addresses the communication bottleneck by using $\tau > 1$ local computation steps as opposed to $\tau = 1$ local steps in minibatch SGD. It also performs the averaging step (c) securely to enhance data privacy. However, the vanilla FL objective places a limit on how well statistical heterogeneity can be addressed. By minimizing the average training loss, the resulting model w can sacrifice performance on "difficult" devices in order to perform well on average. In other words, it is not guaranteed to perform well on *individual* test devices, whose distribution p might be quite different from the average training distribution $\sum_{k=1}^N \alpha_k q_k$. Our goal in this work is to design an objective function, different from the vanilla FL objective (6.1) to better handle statistical heterogeneity, and design a federated optimization algorithm similar to *FedAvg* to optimize it.

6.2.2 *Problem formulation: conformity and heterogeneity*

In this work, we consider test devices whose distribution p can be written as a mixture $p_\pi := \sum_{k=1}^N \pi_k q_k$ of the training distribution q_k of the devices with weights $\pi \in \Delta^{N-1}$. Here, Δ^{N-1} denotes the probability simplex in $\mathbb{R}^N$. The test distribution p_π is different from the average training distribution $p_\alpha = \sum_{k=1}^N \alpha_k q_k$ if the mixture weights π are different from the training weights $\alpha = (\alpha_1, \ldots, \alpha_N)$.

We now define the *conformity* of a mixture p_π to the training distribution p_α, as a measure of the degree of similarity between p_π and p_α.

Conformity

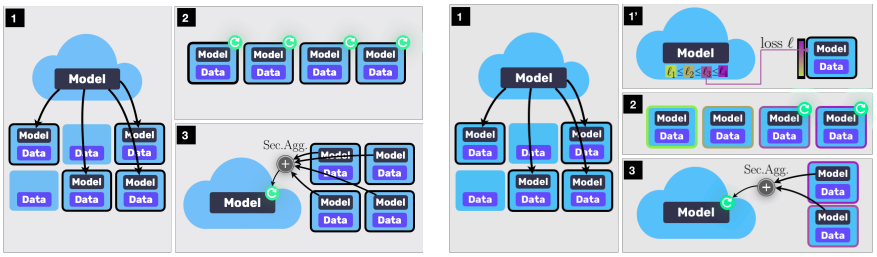
Definition 6.1. The *conformity* $\text{conf}(p_\pi) \in (0, 1]$ of a mixture p_π with weights π is defined as $\min_{k \in [\![1,N]\!]} \alpha_k / \pi_k$. The conformity of a device refers to the conformity of its data distribution.

A mixture distribution p_π with $\text{conf}(p_\pi) = \theta$ must satisfy $\pi_k \leq \alpha_k / \theta$ for each k. Since $\sum_k \pi_k = 1$, we also get that $\pi_k \geq \max\{0, \alpha_k - (1-\theta)\}/\theta$[2].

Interpretation

Assuming that the training devices are a representative sample of the population of devices, every device's distribution can be well-approximated by a mixture p_π for some $\pi \in \Delta^{N-1}$. The conformity of a device is *a scalar summary of how close it is to the trend*, modeled by the aggregated distribution $p_\alpha = \sum_k \alpha_k q_k$. A test device with conformity $\theta \approx 1$ closely conforms to the trend. For such device, a model trained on p_α is expected to have a high predictive power. In contrast, a test device with $\theta \approx 0$ can have a distribution that vastly differ from p_α, and the predictive power of a model trained on p_α can be poor.

2 We do not directly impose a lower bound on π_k, apart from the implied bound $\pi_k \geq \max\{0, \alpha_k - (1-\theta)\}/\theta$, because it is not realistic to assume that the distribution on a test device must necessarily contain a component of every training distribution q_k.



(a) *FedAvg*. **(b)** Δ-*FL*.

Figure 6.2: Comparative diagram between the baseline *FedAvg* and our algorithm Δ-*FL* (recall of Figure 1.4). Both algorithms consist of the following steps (note difference in step 1'). **Step 1**: Server selects m client devices and broadcasts the model to each selected device. **Step 1' (Δ-*FL* only)**: Each selected device computes the loss (a scalar) incurred by the model on its local data and sends it to the server. Based on these losses, the server computes a threshold loss. It only keeps devices whose losses are larger than this threshold, and un-selects the other devices. **Step 2**: Each selected device computes an update to the server model based on its local data. **Step 3**: Updates from selected devices are securely aggregated to update the server model.

There is a trade-off between fitting to the trend and supporting non-conforming test devices. The conformity θ presents a natural way to encapsulate this trade-off in a scalar parameter. That is, given a conformity $\theta \in (0, 1)$, we choose to only support test distributions p_π with $\text{conf}(p_\pi) \geq \theta$.

6.3 THE Δ-FL FRAMEWORK

In this section, we define the Δ-*FL* framework in Section 6.3.1, and propose an algorithm to optimize in the federated setting in Section 6.3.2. Finally, we illustrate our approach on a toy example in Section 6.3.3.

6.3.1 *The framework*

The Δ-*FL* framework aims to supply each test device with a model appropriate to its conformity. Given a discretization $\{\theta_1, \ldots, \theta_r\}$ of $(0, 1]$, Δ-*FL* maintains r models, one for each conformity level θ_j. The local data is not allowed to leave a device due to privacy restrictions; hence, the conformity of a test device cannot be measured. Instead, we allow each test device to tune their conformity. See the schematic in Figure 6.1 for an illustration.

Δ-FL's objective

To train a model for a conformity level θ, we aim to do well on *all* distributions p_π with $\text{conf}(p_\pi) \geq \theta$. Thus, we propose to solve:

$$\min_{w \in \mathbb{R}^d} \left[F_\theta(w) := \max_{\pi \in \mathcal{P}_\theta} F(w; p_\pi) + \frac{\lambda}{2} \|w\|^2 \right], \quad \text{where,} \quad \mathcal{P}_\theta := \left\{ \pi \in \Delta^{N-1} \,:\, \text{conf}(p_\pi) \geq \theta \right\}. \tag{6.2}$$

In contrast, the vanilla FL objective optimizes $F(w; p_\alpha)$, which is defined on the basis of the training distribution p_α. We observe that Δ-*FL* is designed to be robust on all test devices with conformity greater than θ.

Connection to the Superquantile

Noting that for any $\pi \in \Delta^{N-1}$, we have by definition

$$F(w, p_\pi) = \sum_{i=1}^{N} \pi_i F(w, q_i).$$

One may recognize in the objective function of (6.2) the $(1-\theta)$-superquantile over the distribution of losses $(F_k(w))_{k\in[\![1,N]\!]}$ weighted with respective probabilities $(\alpha_k)_{k\in[\![1,N]\!]}$. Thus, the Δ-*FL* framework enforces robustness at the scale of the network by dynamically re-weighting the importance of the devices involved in the training process.

6.3.2 *Federated optimization for Δ-FL*

We propose a federated optimization algorithm for the Δ-*FL* objective. While there could be many approaches to optimizing (6.2), we consider algorithms similar to *FedAvg* for their ability to avoid communication bottlenecks and preserve the privacy of user data. A practical federated learning algorithm cannot assume that all the devices are always available; it must be able to work with a subset of devices in each round. To this end, we define the counterpart of the constraint set $\mathcal{P}_\theta$ from (6.2) defined on a subset $S \subset [\![1, N]\!]$ of m devices as:

$$\mathcal{P}_{\theta,S} = \left\{\pi \in \Delta^{|S|-1} \,:\, \pi_k \leq \alpha_k/(\theta A_S), \text{ for } k \in S\right\}, \tag{6.3}$$

where $A_S = \sum_{k\in S} \alpha_k$ is the cumulative weight in S and we denote $(\pi_k)_{k\in S} \in \mathbb{R}^{|S|}$ by π with slight abuse of notation.

Algorithm for Δ-FL

The optimization algorithm for the Δ-*FL* objective (6.2) is given in Algorithm 5, and illustrated in Figure 6.2. It has the following four steps:

1- *Model Broadcast* (line 2): The server samples a set S of m devices from $[\![1, N]\!]$ and sends the current model $w^{(t)}$.

1'- *Device Reweighting* (line 3): Each device $k \in S$ sends its current loss $F_k(w^{(t)})$ to the server; the server computes and sends back new weights $\pi_k^{(t)}$ to device k.

2- *Local Updates* (loop of line 4): Staring from $w_{k,0}^{(t)} = w^{(t)}$, each device $k \in S$ makes τ local gradient or stochastic gradient descent steps with a learning rate γ.

3- *Update Aggregation* (line 10): The models from the selected devices are sent to the server and aggregated to update the server model (with the weights $\pi^{(t)}$ computed in Line 3).

Compliance to federated constraints

Compared to *FedAvg*, Δ-*FL* has the additional step of computing new weights $\pi_k^{(t)}$ for each selected device $k \in S$ in Line 3. Let us consider Δ-*FL* in relation to the three keys aspects of federated learning.

(a) *Communication Bottleneck*: Identical to *FedAvg*, Δ-*FL* algorithm performs multiple computation rounds per communication round.

(b) *Statistical Heterogeneity*: The Δ-*FL* objective (6.2) is designed to better handle the statistical heterogeneity by minimizing the worst-case over all test distributions with conformity at least θ, while the vanilla FL objective cannot handle non-conforming devices.

(c) *Privacy*: Identical to *FedAvg*, Δ-*FL* does not require any data transfer and the aggregation of line 10 can be securely performed using secure multiparty communication. However, as Algorithm 5 is currently stated, reweighting the selected devices (line 3) requires the per-device losses $F_k(w^{(t)})$ to be sent visible to the server. We show in Section 6.5.4 how to compute the weights $\pi^{(t)}$ using a secure aggregation oracle.

Algorithm 5: The Δ-*FL* Algorithm

Require: Initial iterate $w^{(0)}$, Number of communication rounds T, Number of devices per round m, Number of local updates τ, Local step size γ

for $t = 0, 1, \cdots, T-1$ **do**
  Sample m devices from $[\![1, N]\!]$ without replacement in S;
  Compute $\pi^{(t)} = \arg\max_{\pi \in \mathcal{P}_{\theta,S}} \sum_{k \in S} \pi_k F_k(w^{(t)})$;
  for *each selected device* $k \in S$ *in parallel* **do**
    Initialize $w_{k,0}^{(t)} = w^{(t)}$;
    for $j = 0, \cdots, \tau - 1$ **do**
      $w_{k,j+1}^{(t)} = (1 - \gamma\lambda) w_{k,j}^{(t)} - \gamma \nabla F_k(w_{k,j}^{(t)})$;
    end
  end
  $w^{(t+1)} = \sum_{k \in S} \pi_k^{(t)} w_{k,\tau}^{(t)}$;
end
return w_T

6.3.3 *Illustration on a toy example*

A mean estimation problem

We now illustrate the Δ-*FL* objective on a simple example of a mixture of Gaussians. Consider a mixture of $N = 3$ Gaussian distributions in $\mathbb{R}^2$, with uniform weights ($\alpha_k = 1/N$), identity covariance and respective means μ_1, μ_2, μ_3 which form a scalene triangle – see Figure 6.3. We assume that each distribution represents a training device. Consider the task of mean estimation where $f(w; \xi) = \|\xi - w\|_2^2$ so that $F(w; p)$ is minimized by the mean of the mixture p_α. Suppose in our toy federated learning scenario that a model w is trained on the 3 available training devices before being deployed on a test device with distribution p_π. Vanilla federated learning, which is a special case of the Δ-*FL* framework with conformity $\theta = 1$, aims to minimize $F(\cdot\,; p_\alpha)$ over the training distribution p_α. The minimizer w_1 of the loss $F(\cdot\,; p_\alpha)$ on the training distribution is simply the mean $w_1 := (\mu_1 + \mu_2 + \mu_3)/3$.

Performance on test clients

Now consider a conformity level of $\theta = 2/3$. In this case, a simple calculation shows that the Δ-*FL* objective is a piecewise quadratic, which is minimized at the midpoint of the longest side of the triangle formed by μ_1, μ_2, μ_3. In the example of Figure 6.3, this is $w_{2/3} = (\mu_1 + \mu_3)/2$.

Next, consider the set $\mathcal{P}_{2/3}$ of all mixture weights π such that $\text{conf}(p_\pi) \geq 2/3$. We see from Figure 6.3 (middle) that there are mixtures for which $w_{2/3}$ is better than w_1 and vice-versa. However, from the histogram of losses in Figure 6.3, we see that the *worst loss* $F(\cdot\,; p_\pi)$ over all such mixtures is lower for the Δ-*FL* model $w_{2/3}$. In practical terms, Δ-*FL* presents an improvement on devices with the worst user experience. Moreover, by optimizing the superquantile, Δ-*FL* aims for good performance on *all* test devices with a given conformity, *irrespective of their distribution*. Note that while we use a uniform distribution in the illustration of Figure 6.3 (right), this distribution is unknown in practice.

6.4 CONVERGENCE ANALYSIS

In this section, we provide a convergence analysis of our algorithm. The principal result of this section is Theorem 6.1 which establishes a convergence rate in the strongly convex setting. The main difficulties raised concern the

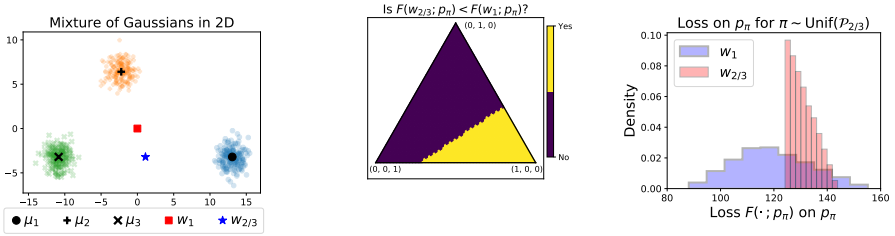


Figure 6.3: Illustration of Δ-*FL* with a uniform mixture of Gaussians. **Left**: Positions in $\mathbb{R}^2$ of the means μ_1, μ_2, μ_3 of Gaussians q_1, q_2, q_3 resp., the vanilla federated learning model w_1, and the Δ-*FL* model $w_{2/3}$ at conformity $\theta = 2/3$. **Middle**: Comparison of the loss $F(\cdot\,; p_\pi)$ for each possible mixture p_π with weights $\pi = (\pi_1, \pi_2, \pi_3)$. **Right**: Histogram of losses $F(\cdot; p_\pi)$ for p_π drawn uniformly from the set of all mixtures of q_1, q_2, q_3 with conformity at least $\theta = 2/3$.

extension of the recent local SGD framework [42, 56, 81, 99, 153] to the dynamic reweighting that the superquantile induces in line (3) of our Algorithm 5. Alternatively, this analysis can be seen as an extension of the stochastic gradient descent method [94] for superquantiles to federated frameworks.

6.4.1 *Assumptions and main result*

We first detail the assumptions under which our convergence analysis holds.

Assumptions. First, we assume without loss of generality that the weight of each training device is $\alpha_k = 1/N$[3]. We make some assumptions on the per-device losses F_k, which are assumed to hold throughout this section. For each device $k \in [N]$, the objective F_k is

(a) B-bounded, i.e., $0 \le F_k(w) \le B$ for all $w \in \mathbb{R}^d$,
(b) G-Lipschitz, i.e., $|F_k(w) - F_k(w')| \le G \|w - w'\|$ for all $w, w' \in \mathbb{R}^d$, and,
(c) L-smooth, i.e., F_k is continuously differentiable and its gradient ∇F_k is L-Lipschitz.

Convergence result for Δ-FL

Theorem 6.1. *Suppose each function F_k is convex and that the three above assumptions hold. Fix a time horizon T and consider the sequence $(w^{(t)})_{t=0}^T$ of iterates produced by the Δ-FL algorithm with smoothing. Define $L' = \lambda + L + G^2\sqrt{T}$ and assume that the learning rate γ satisfies*

$$\gamma \le \min\left\{\frac{1}{4\tau L'}, \frac{\sqrt{\lambda}}{18\tau(L+\lambda)\sqrt{L'}}\right\}.$$

Define the averaged iterate

$$\overline{w}^{(t)} = \frac{\sum_{i=0}^t \beta_i w^{(i)}}{\sum_{i=0}^t \beta_i}, \quad \text{where} \quad \beta_i = \left(1 - \frac{\gamma\lambda\tau}{2}\right)^{-(1+i)}.$$

Then, letting $w^\star = \arg\min_{w\in\mathbb{R}^d} F_\theta^\nu(w)$, we have the bound

$$\mathbb{E}\left[F_\theta(\overline{w}^{(T)}) - F_\theta(w^\star)\right] \le 4\exp(-\gamma\tau\lambda T)\left\|w^{(0)} - w^\star\right\|^2 + c\,\mathcal{T},$$

3 This assumption is made to avoid technicalities with random sums $\sum_{k\in S}\alpha_k$. We can extend the convergence analysis to the case of unequal α_k's by performing a standard reduction of replacing $F_k(w)$ with $F'_k(w) := N\alpha_k F_k(w)$. The proofs hold in this case if B, G, L are multiplied by a factor of $N\max_k \alpha_k$.

where c is a universal constant and

$$\mathcal{T} = \frac{G^2 L^2 \gamma^2 \tau(\tau-1)}{\lambda} + \frac{B}{\sqrt{\theta m}} + \frac{G^2 \gamma \tau}{\theta m} + \frac{\log m}{\sqrt{T}}.$$

Constant terms in $\mathcal{T}$ and the final rate

The first term in $\mathcal{T}$ is called client drift [72, 99] and decays as $O(\gamma^2)$ and vanishes if $\tau = 1$. The second and third terms stem from the bias and variance from sampling devices in each round, and vanish if $m = N$. See Proposition 3.13 for further details. The final term appears due to considering smoothed objective F_θ^ν for the minimization of the nonsmooth F_θ. These terms can be balanced by taking $\gamma = O(1/\sqrt{\tau^2 T})$ to get $O(1/\sqrt{T})$ convergence up to the bias term. Finally, the bias and variance due to partial participation encourage having θm large enough for the bound to be meaningful.

Proof Synopsis.

- In Section 6.4.2, we first recall and adapt the bias and variance of the stochastic estimator of the superquantile from Section 3.4 together with a few basic convex inequalities.
- In Section 6.4.3, we show how to bound the gradient dissimilarity within the robust selection of devices performed in Lines 2-3 of Algorithm 5 and the *client drift* caused by the local steps.
- In Section 6.4.4, we provide a descent lemma for one complete round of communication and gather all the results to finish the proof.

6.4.2 *Preliminary results*

We first recall a few standard inequalities:

- *Rob Peter to pay Paul*: For any $x, y \in \mathbb{R}^d$ and $\alpha > 0$ we have:

$$\|x+y\|^2 \leq (1+\alpha)\|x\|^2 + \left(1+\frac{1}{\alpha}\right)\|y\|^2. \tag{6.4}$$

- *Pythagoras theorem*: For any $\mathbb{R}^d$-valued random vector X such that $\mathbb{E}\|X\|^2 < \infty$,

$$\mathbb{E}\|u\|^2 = \mathbb{E}\|u - \mathbb{E}[u]\|^2 + \|\mathbb{E}[u]\|^2. \tag{6.5}$$

- *Strong convexity*: Let $F : \mathbb{R}^{\mathrm{d}} \to \mathbb{R}$ be μ-strongly convex. Then for any $x, y \in \mathbb{R}^{\mathrm{d}}$, we have:

$$\langle \nabla F(x), x - y \rangle \geq F(x) - F(y) + \frac{\mu}{2}\|x-y\|^2. \tag{6.6}$$

- *Smoothness*: Let $F : \mathbb{R}^{\mathrm{d}} \to \mathbb{R}$ be L-smooth and let $F^\star$ be the minimum value of F (assuming it exists). Then for any $x \in \mathbb{R}^{\mathrm{d}}$, we have:

$$\|\nabla F(x)\|^2 \leq 2L\left(F(x) - F^\star\right). \tag{6.7}$$

Smoothing

We handle the nonsmoothness of the objective due to the superquantile with the smoothing procedure from Section 4.2.2. More precisely, we consider the

entropic smoothing from Example 4.2 on the sequence of losses over the set $S \subset [N]$ of selected devices:

$$D_S(\pi) = \sum_{k \in S} \pi_k \log(m\, \pi_k)\,.$$

We simply write $D(\pi)$ when $S = [N]$. We define the smooth counterpart to (6.2) as [8, 41, 117]

$$F_\theta^\nu(w) = \max_{\pi \in \mathcal{P}_\theta} \left\{ \sum_{k=1}^N \pi_k F_k(w) - \nu D(\pi) \right\} + \frac{\lambda}{2} \|w\|^2\,, \tag{6.8}$$

where $\nu > 0$ is a fixed smoothing parameter. We have that $|F_\theta^\nu(w) - F_\theta(w)| \leq 2\nu \log N$. We modify Line 3 of Algorithm 5 as

$$\pi^{(t)} = \arg\max_{\pi \in \mathcal{P}_{\theta,S}} \left\{ \sum_{k \in S} \pi_k F_k(w^{(t)}) - \nu D_S(\pi) \right\}\,.$$

As pointed out in the Section 4.2.2 of this thesis, for any $S \subset [N]$ of size m, the partial superquantile is differentiable at w with :

$$\nabla F_{\theta,S}^\nu(w) = \sum_{k \in S} \pi_k^\star \nabla \tilde{F}_k(w) \tag{6.9}$$

where $\tilde{F}_k$, introduced to lighten the notations, denotes $\tilde{F}_k(w) = F_k(w) + (\lambda/2)\|w\|^2$, and $\pi^\star$ denotes the solution to the maximization problem

$$F_{\theta,S}^\nu(w) = \max_{\pi \in \mathcal{P}_{\theta,S}} \sum_{k \in S} \pi_k \tilde{F}_k(w) - \nu D_S(\pi).$$

In view of the assumptions on the losses F_k, the following result falls from standard [8, Theorem 4.1, Lemma 4.2].

Smoothing and smoothness constants

Proposition 6.2. *For every $\nu > 0$, we have that $F_{\theta,S}^\nu$ and $\overline{F}_{\theta,S}^\nu$ are L'-smooth with $L' = L + \lambda + \frac{G^2}{\nu}$.*

6.4.3 *Adversarial gradient dissimilarity and client drift*

In this section, we show how to bound the client drift that is accumulated during the local steps of our Algorithm 5. We first analyze the client dissimilarity among the training clients in our specific risk-averse setting. We derive then

Bounding gradient dissimilarity. Bounding of the variance of gradient estimators is a key assumption in the analysis of stochastic gradients methods (see e.g. the textbook [22]). In the centralized setting, when considering a stochastic objective $\mathbb{E}_\xi[f(w,\xi)]$, it is standard to assume for a given estimator g_w of $\nabla_w \mathbb{E}\left[f(w,\xi\right]$ that there exists some constants $M_1, M_2 > 0$ such that for all $w \in \mathbb{R}^{\mathrm{d}}$,

$$\mathbb{E}\left[\left\|g_w\right\|^2\right] \leq M_1 \quad \text{or} \quad \mathbb{E}\left[\left\|g_w\right\|^2\right] \leq M_1 + M_2 \left\|\nabla_w \mathbb{E}\left[f(w,\xi\right]\right\|^2$$

In the federated setting, the use of a subset $S \subset [N]$ of devices in each round induces noise on the estimation of the average gradient over the whole network.

Bounded on gradient dissimilarity assumption in federated learning

Thus, such assumption translates into a *bound on the gradient dissimilarity* among the agents [72, 172]:

$$\frac{1}{N}\sum_{k\in[N]}\left\|\nabla\tilde{F}_k(w)\right\|^2 \leq M_1 \quad \text{or} \quad \frac{1}{N}\sum_{k\in[N]}\left\|\nabla\tilde{F}_k(w)\right\|^2 \leq M_1 + M_2\left\|\frac{1}{N}\sum_{k\in[N]}\nabla\tilde{F}_k(w)\right\|^2$$

Here, we also consider the minimization of the global loss F_θ^ν by a stochastic algorithm based on a partial participation of the devices in the network, with the additional difficulties that we only have access to a biased estimator $\overline{F}_\theta^\nu$ of the loss F_θ^ν and its gradient. In particular, the adaptive reweighting of the clients selected at each round does not permit the direct use of such assumption. We show instead in the next lemma that the variance of stochastic gradient estimator can also be bounded, thanks to the Lipschitz assumption.

Adversarial Gradient Dissimilarity

Proposition 6.3. *Consider the quantities $\pi^{(t)}, w^{(t)}$ from Algorithm 5. We have,*

$$\mathbb{E}\left[\sum_{k\in S}\pi_k^{(t)}\left\|\nabla\tilde{F}_k(w^{(t)})\right\|^2\middle|\mathcal{F}_t\right] \leq \left(4+\frac{8}{\theta m}\right)G^2 + \left\|\nabla\overline{F}_\theta^\nu(w^{(t)})\right\|^2.$$

Proof. By (6.5), we have:

$$\begin{aligned}\sum_{k\in S}\pi_k^{(t)}\left\|\nabla\tilde{F}_j(w^{(t)})\right\|^2 &= \sum_{k\in S}\pi_k^{(t)}\left\|\nabla\tilde{F}_k(w^{(t)}) - \nabla F_{\theta,S}^\nu(w^{(t)})\right\|^2 + \left\|\nabla F_{\theta,S}^\nu(w^{(t)})\right\|^2 \\ &= \sum_{k\in S}\pi_k^{(t)}\left\|(\nabla F_k(w^{(t)}) - \sum_{i\in S}\pi_i^{(t)}\nabla F_i(w^{(t)}))\right\|^2 + \left\|\nabla F_{\theta,S}^\nu(w^{(t)})\right\|^2.\end{aligned}$$

Now since the weights $\pi_k^{(t)}$ sum to one, we may use the convexity of $\|\cdot\|^2$ to get:

$$\sum_{k\in S}\pi_k^{(t)}\left\|\nabla\tilde{F}_j(w^{(t)})\right\|^2 \leq \sum_{k,i\in S}\pi_k^{(t)}\pi_i^{(t)}\left\|\nabla F_i(w^{(t)}) - \nabla F_k(w^{(t)})\right\|^2 + \left\|\nabla F_{\theta,S}^\nu(w^{(t)})\right\|^2.$$

The squared triangle inequality (cf. (6.4)) together with the Lipschitz assumption on the functions F_k yields:

$$\begin{aligned}\sum_{k\in S}\pi_k^{(t)}\left\|\nabla\tilde{F}_k(w^{(t)})\right\|^2 &\leq 2\sum_{k,i\in S}\pi_k^{(t)}\pi_i^{(t)}\left(\left\|\nabla F_k(w^{(t)})\right\|^2 + \left\|\nabla F_i(w^{(t)})\right\|^2\right)\left\|\nabla F_{\theta,S}^\nu(w^{(t)})\right\|^2 \\ &\leq 4\,G^2 + \left\|\nabla F_{\theta,S}^\nu(w^{(t)})\right\|^2.\end{aligned}$$

Thus, taking an expectation over $S\sim U_m$ gives

$$\mathbb{E}\left[\sum_{k\in S}\pi_k^{(t)}\left\|\nabla\tilde{F}_j(w^{(t)})\right\|^2\middle|\mathcal{F}_t\right] \leq 4\,G^2 + \mathbb{E}_{S\sim U_m}\left[\left\|\nabla F_{\theta,S}^\nu(w^{(t)})\right\|^2\right].$$

By Pythagoras theorem (6.5), we get,

$$\begin{aligned}&\mathbb{E}\left[\sum_{k\in S}\pi_k^{(t)}\left\|\nabla\tilde{F}_k(w^{(t)})\right\|^2\middle|\mathcal{F}_t\right] \\ &\qquad\leq 4\,G^2 + \mathbb{E}\left[\left\|\nabla F_{\theta,S}^\nu(w^{(t)}) - \nabla\overline{F}_\theta^\nu(w^{(t)})\right\|^2\middle|\mathcal{F}_t\right] + \left\|\nabla\overline{F}_\theta^\nu(w^{(t)})\right\|^2.\end{aligned} \tag{6.10}$$

Finally, substituting the variance bound from Lemma 3.13 into (6.10) yields the stated result. □

Bounding the Client Drift. During federated learning, each device takes multiple local steps. This causes the resulting update to be a biased estimator of a descent direction for the global objective. This phenomenon has been referred to as "client drift" [72, 99]. Current proof techniques rely on treating this as a "noise" term which is to be controlled. In the context of this work, the reweighting by $\pi^{(t)}$ requires us to adapt this typical definition of client drift to our setting. We thus define the client drift $d^{(t)}$ in outer iteration t of the algorithm as

Client drift

$$d^{(t)} := \mathbb{E}_{S\sim U_m}\left[\sum_{k\in S}\pi_k^{(t)}\sum_{j=0}^{\tau-1}\left\|w_{k,j}^{(t)}-w^{(t)}\right\|^2\middle|\mathcal{F}_t\right]. \tag{6.11}$$

Proposition 6.4. *If $\gamma \leq \frac{1}{4\tau(L+\lambda)}$, we have the following bounds for any $t \geq 0$:*

Upper bound on client drift

$$d^{(t)} \leq \tau^2(\tau-1)\gamma^2 e^2\left(\left(4+\frac{8}{\theta m}\right)G^2+\left\|\overline{F}_\theta^\nu(w^{(t)})\right\|^2\right) \text{ and,}$$
$$d^{(t)} \leq \tau^2(\tau-1)\gamma^2 e^2\left(\left(4+\frac{8}{\theta m}\right)G^2+2L'\left(\overline{F}_\theta^\nu(w^{(t)})-\overline{F}_\theta^\nu(\overline{w}^\star)\right)\right).$$

Proof. If $\tau = 1$, there is nothing to prove as both sides of the inequality are 0. We assume now that $\tau > 1$. Let us first fix $S \subset [N]$ of size $|S| = m$. For any $k \in S$ and $j \in [\![1, \tau-1]\!]$, by (6.4), we have:

$$\begin{aligned}\left\|w_{k,j}^{(t)}-w^{(t)}\right\|^2 &= \left\|w_{k,j-1}^{(r)}-\gamma\nabla\tilde{F}_k(w_{k,j-1}^{(t)})-w^{(t)}\right\|^2\\ &\leq \left(1+\frac{1}{\tau-1}\right)\left\|w_{k,j-1}^{(t)}-w^{(t)}\right\|^2+\tau\gamma^2\left\|\nabla\tilde{F}_k(w_{k,j-1}^{(t)})\right\|^2.\end{aligned}$$

Using(6.4) and the smoothness of the local losses, this gives:

$$\begin{aligned}\left\|w_{k,j}^{(t)}-w^{(t)}\right\|^2 &\leq \left(1+\frac{1}{\tau-1}\right)\left\|w_{k,j-1}^{(t)}-w^{(t)}\right\|^2\\ &\quad+2\tau\gamma^2\left(\left\|\nabla\tilde{F}_k(w_{k,j-1}^{(t)})-\nabla\tilde{F}_k(w^{(t)})\right\|^2+\left\|\nabla\tilde{F}_k(w^{(t)})\right\|^2\right)\\ &\leq \left(1+\frac{1}{\tau-1}\right)\left\|w_{k,j-1}^{(t)}-w^{(t)}\right\|^2\\ &\quad+2\tau\gamma^2(L+\lambda)^2\left\|w_{k,j-1}^{(t)}-w^{(t)}\right\|^2+2\tau\gamma^2\left\|\nabla\tilde{F}_k(w^{(t)})\right\|^2.\end{aligned}$$

Hence, for $\gamma \leq \frac{1}{4\tau(L+\lambda)}$, we get:

$$\begin{aligned}\left\|w_{k,j}^{(t)}-w^{(t)}\right\|^2 &\leq \left(1+\frac{1}{\tau-1}+2\tau\gamma^2(L+\lambda)^2\right)\left\|w_{k,j-1}^{(t)}-w^{(t)}\right\|^2+2\tau\gamma^2\left\|\nabla\tilde{F}_k(w^{(t)})\right\|^2\\ &\leq \left(1+\frac{2}{\tau-1}\right)\left\|w_{k,j-1}^{(t)}-w^{(t)}\right\|^2+2\tau\gamma^2\left\|\nabla\tilde{F}_k(w^{(t)})\right\|^2.\end{aligned}$$

Unrolling this recursion yields for any $j \leq \tau - 1$

$$\left\|w_{k,j}^{(t)} - w^{(t)}\right\|^2 \leq \sum_{i=0}^{j-1} \left(1 + \frac{2}{\tau - 1}\right)^i \left(2\tau\gamma^2 \left\|\nabla \tilde{F}_k(w^{(t)})\right\|^2\right)$$
$$\leq \frac{\tau - 1}{2} \left(1 + \frac{2}{\tau - 1}\right)^j \left(2\tau\gamma^2 \left\|\nabla \tilde{F}_k(w^{(t)})\right\|^2\right).$$

Thus, for any $j \leq \tau - 1$,

$$\left\|w_{k,j}^{(t)} - w^{(t)}\right\|^2 \leq \frac{\tau - 1}{2} \left(1 + \frac{2}{\tau - 1}\right)^{\tau - 1} \left(2\tau\gamma^2 \left\|\nabla \tilde{F}_k(w^{(t)})\right\|^2\right)$$
$$\leq \tau(\tau - 1)\gamma^2 e^2 \left\|\nabla \tilde{F}_k(w^{(t)})\right\|^2,$$

where we use $(1 + 1/x)^x \leq e$ for any $x > 0$. Therefore, by Lemma 6.3, we get

$$d^{(t)} \leq \tau^2(\tau - 1)\gamma^2 e^2 \, \mathbb{E}_{S \sim U_m} \left[\sum_{k \in S} \pi_k^{(t)} \left\|\nabla \tilde{F}_k(w^{(t)})\right\|^2 \middle| \mathcal{F}^{(t)}\right]$$
$$\leq \tau^2(\tau - 1)\gamma^2 e^2 \left(\left(4 + \frac{8}{\theta m}\right) G^2 + \left\|\overline{F}_\theta^\nu(w^{(t)})\right\|^2\right).$$

This gives the first bound. The second bound follows by smoothness (6.7). □

Bound on the Norm of Each Update. We bound the expected squared norm of each update $w^{(t+1)} - w^{(t)}$, which has the closed form expression:

$$w^{(t+1)} - w^{(t)} = -\gamma \sum_{k \in S} \pi_k^{(t)} \sum_{j=0}^{\tau-1} \nabla \tilde{F}_k(w_{k,j}^{(t)}). \tag{6.12}$$

Bound on updates' norm

Proposition 6.5. *We have the bounds,*

$$\gamma^2 \, \mathbb{E}\left[\left\|\sum_{k \in S} \pi_k^{(t)} \sum_{j=0}^{\tau-1} \nabla \tilde{F}_k(w_{k,j}^{(t)})\right\|^2 \middle| \mathcal{F}_t\right] \leq 2\gamma^2\tau \, (L + \lambda)^2 \; d^{(t)} + \frac{16\tau^2\gamma^2 G^2}{\theta m}$$
$$+ 4\tau^2\gamma^2 L' \left(\overline{F}_\theta^\nu(w^{(t)}) - \overline{F}_\theta^\nu(\overline{w}^\star)\right),$$

where $d^{(t)}$ is the client drift term defined in (6.11).

Proof. Using (6.4) together with the gradient formula (6.9), we get:

$$\left\|\sum_{k \in S} \pi_k^{(t)} \sum_{j=0}^{\tau-1} \nabla \tilde{F}_k(w_{k,j}^{(t)})\right\|^2 \leq 2\left\|\sum_{k \in S} \pi_k^{(t)} \sum_{j=0}^{\tau-1} \left(\nabla \tilde{F}_k(w_{k,j}^{(t)}) - \nabla \tilde{F}_k(w^{(t)})\right)\right\|^2 + 2\left\|\sum_{k \in S} \pi_k^{(t)} \sum_{j=0}^{\tau-1} \nabla \tilde{F}_k(w^{(t)})\right\|^2$$
$$\leq 2\tau \sum_{k \in S} \pi_k^{(t)} \sum_{j=0}^{\tau-1} \left\|\nabla \tilde{F}_k(w_{k,j}^{(t)}) - \nabla \tilde{F}_k(w^{(t)})\right\|^2 + 2\tau^2 \left\|\nabla F_{\theta,S}^\nu(w^{(t)})\right\|^2.$$

For the first term, we invoke $(L+\lambda)$-smoothness of $\tilde{F}_k$ and take the expectation to get $2\tau(L+\lambda)^2 d^{(t)}$. For the second term, we use (6.5) followed by the variance bound of Proposition 3.13 to get:

$$\mathbb{E}\left[\left\|\nabla F^{\nu}_{\theta,S}(w^{(t)})\right\|^2 \middle| \mathcal{F}_t\right] = \mathbb{E}\left[\left\|\sum_{k\in S}\pi_k^{(t)}\nabla\tilde{F}_k(w^{(t)}) - \nabla\overline{F}^{\nu}_{\theta}(w^{(t)})\right\|^2 \middle| \mathcal{F}_t\right] + \left\|\nabla\overline{F}^{\nu}_{\theta}(w^{(t)})\right\|^2$$
$$\leq \frac{8G^2}{\theta m} + \left\|\nabla\overline{F}^{\nu}_{\theta}(w^{(t)})\right\|^2 .$$

This gives

$$\gamma^2\,\mathbb{E}\left[\left\|\sum_{k\in S}\pi_k^{(t)}\sum_{j=0}^{\tau-1}\nabla\tilde{F}_k(w_{k,j}^{(t)})\right\|^2 \middle| \mathcal{F}_t\right] \leq 2\gamma^2\tau\,(L+\lambda)^2\;d^{(t)} + \frac{16\tau^2\gamma^2G^2}{\theta m} + 2\tau^2\gamma^2\left\|\nabla\overline{F}^{\nu}_{\theta}(w^{(t)})\right\|^2$$

The stated result follows by smoothness (6.7). □

6.4.4 *Effect of one round*

The crux of the proof of Theorem 6.1 is the following statement.

Proposition 6.6. *Consider the setting of Theorem 6.1. Let $(w^{(t)})_{t\geq 0}$ the sequence of global models generated by Algorithm 5. For any $t \geq 0$, we have:* *Descent Lemma*

$$\overline{F}^{\nu}_{\theta}(w^{(t)}) - \overline{F}^{\nu}_{\theta}(\overline{w}^{\star}) \leq \frac{1}{\gamma\tau}\left(1 - \frac{\lambda\gamma\tau}{2}\right)\left\|w^{(t)} - \overline{w}^{\star}\right\|^2 - \frac{1}{\gamma\tau}\left\|w^{(t+1)} - \overline{w}^{\star}\right\|^2$$
$$+ \frac{16\tau G^2\gamma}{\theta m} + \frac{9\,(L+\lambda)^2}{\tau\lambda}d^{(t)},$$

where $d^{(t)}$ denotes the client drift, defined in (6.11).

Proof. We denote $\mathbb{E}_t[\cdot] := \mathbb{E}[\cdot|\mathcal{F}_t]$. We expand the update 6.12 to get

$$\mathbb{E}_t\left\|w^{(t+1)} - \overline{w}^{\star}\right\|^2 = \left\|w^{(t)} - \overline{w}^{\star}\right\|^2 \underbrace{- 2\gamma\mathbb{E}_t\left[\sum_{k\in S}\pi_k^{(t)}\sum_{j=0}^{\tau-1}\left\langle\nabla\tilde{F}_k(w_{k,j}^{(t)}), w^{(t)} - \overline{w}^{\star}\right\rangle\right]}_{=:A}$$
$$\underbrace{+\,\gamma^2\,\mathbb{E}_t\left\|\sum_{k\in S}\pi_k^{(t)}\sum_{j=0}^{\tau-1}\nabla\tilde{F}_k(w_{k,j}^{(t)})\right\|^2}_{=:B} .$$

Let us first bound A. We use λ-strong convexity (6.6) of $\tilde{F}_k$ to get

$$\begin{aligned}\left\langle \sum_{k\in S} \pi_k^{(t)} \nabla \tilde{F}_k(w_{k,j}^{(t)}), w^{(t)} - \overline{w}^\star \right\rangle &= \left\langle \sum_{k\in S} \pi_k^{(t)} \nabla \tilde{F}_k(w^{(t)}), w^{(t)} - \overline{w}^\star \right\rangle \\ &\quad + \left\langle \sum_{k\in S} \pi_k^{(t)} \left(\nabla \tilde{F}_k(w_{k,j}^{(t)}) - \nabla \tilde{F}_k(w^{(t)}) \right), w^{(t)} - \overline{w}^\star \right\rangle \\ &\geq F_{\theta,S}^{\nu}(w^{(t)}) - F_{\theta,S}^{\nu}(\overline{w}^\star) + \frac{\lambda}{2} \left\| w^{(t)} - \overline{w}^\star \right\|^2 \\ &\quad - \left| \left\langle \sum_{k\in S} \pi_k^{(t)} \left(\nabla \tilde{F}_k(w_{k,j}^{(t)}) - \nabla \tilde{F}_k(w^{(t)}) \right), w^{(t)} - \overline{w}^\star \right\rangle \right| .\end{aligned}$$

Next, using successively the triangle inequality, the Cauchy-Schwartz inequality and $(L+\lambda)$-smoothness of the $\tilde{F}_k$ yields:

$$\begin{aligned}\left| \left\langle \sum_{k\in S} \pi_k^{(t)} \left(\nabla \tilde{F}_k(w_{k,j}^{(t)}) - \nabla \tilde{F}_k(w^{(t)}) \right), w^{(t)} - \overline{w}^\star \right\rangle \right| &\leq \sum_{k\in S} \pi_k^{(t)} \left| \left\langle \nabla \tilde{F}_k(w_{k,j}^{(t)}) - \nabla \tilde{F}_k(w^{(t)}), w^{(t)} - \overline{w}^\star \right\rangle \right| \\ &\leq \sum_{k\in S} \pi_k^{(t)} \| \nabla \tilde{F}_k(w_{k,j}^{(t)}) - \nabla \tilde{F}_k(w^{(t)}) \| \left\| w^{(t)} - \overline{w}^\star \right\| \\ &\leq \sum_{k\in S} \pi_k^{(t)} (L+\lambda) \| w_{k,j}^{(t)} - w^{(t)} \| \left\| w^{(t)} - \overline{w}^\star \right\| .\end{aligned}$$

Finally, using $2|ab| \leq a^2/c^2 + c^2 b^2$ and the convexity of $t \mapsto t^2$,

$$\begin{aligned}&\left| \left\langle \sum_{k\in S} \pi_k^{(t)} \left(\nabla \tilde{F}_k(w_{k,j}^{(t)}) - \nabla \tilde{F}_k(w^{(t)}) \right), w^{(t)} - \overline{w}^\star \right\rangle \right| \\ &\qquad \leq \frac{4}{\lambda} \left(\sum_{k\in S} \pi_k^{(t)} (L+\lambda) \| w_{k,j}^{(t)} - w^{(t)} \| \right)^2 + \frac{\lambda}{4} \left\| w^{(t)} - \overline{w}^\star \right\|^2 \\ &\qquad \leq \frac{\lambda}{4} \left\| w^{(t)} - \overline{w}^\star \right\|^2 + \frac{4(L+\lambda)^2}{\lambda} \sum_{k\in S} \pi_k^{(t)} \left\| w_{k,j}^{(t)} - w^{(t)} \right\|^2 .\end{aligned}$$

Overall, we bound A as

$$\begin{aligned}A &\geq 2\gamma \, \mathbb{E}_t \left[\left[\sum_{j=0}^{\tau-1} \left(F_{\theta,S}^{\nu}(w^{(t)}) - F_{\theta,S}^{\nu}(\overline{w}^\star) + \frac{\lambda}{4} \left\| w^{(t)} - \overline{w}^\star \right\|^2 - \frac{4(L+\lambda)^2}{\lambda} \sum_{k\in S} \pi_k^{(t)} \left\| w_{k,j}^{(t)} - w^{(t)} \right\|^2 \right) \right] \right] \\ &\geq 2\gamma\tau \left(\overline{F}_\theta^{\nu}(w^{(t)}) - \overline{F}_\theta^{\nu}(\overline{w}^\star) \right) + \frac{\lambda\gamma\tau}{2} \left\| w^{(t)} - \overline{w}^\star \right\|^2 - \frac{8\gamma(L+\lambda)^2}{\lambda} d^{(t)} ,\end{aligned}$$

where we use the definition of $d^{(t)}$ from (6.11). We bound B using Proposition 6.5. Putting these together, we get,

$$\begin{aligned}\mathbb{E}_t \left\| w^{(t+1)} - \overline{w}^\star \right\|^2 \leq \left(1 - \frac{\lambda\gamma\tau}{2} \right) \left\| w^{(t)} - \overline{w}^\star \right\|^2 - (2\gamma\tau - 4\gamma^2\tau^2 L')(\overline{F}_\theta^{\nu}(w^{(t)}) - \overline{F}_\theta^{\nu}(\overline{w}^\star)) \\ + \frac{16\tau^2 G^2 \gamma^2}{\theta m} + 2 \left(\gamma^2 \tau (L+\lambda)^2 + 4\gamma \frac{(L+\lambda)^2}{\lambda} \right) d^{(t)} .\end{aligned}$$

With $\gamma \leq (4\tau L')^{-1}$ we have $2\gamma\tau - 4\gamma^2\tau^2 L' \geq \gamma\tau$. Likewise, the same condition on γ also implies $2(\gamma\,(L+\lambda)^2 + 4(L+\lambda)^2/(\tau\lambda)) \leq 9\,(L+\lambda)^2\,/(\tau\lambda)$. Rearranging completes the proof. □

We have now all the ingredients to establish the proof of Theorem 6.1. *Time for a conclusion*!

Proof of Theorem 6.1. Plugging in the client drift bound of Proposition 6.4 into the bound of Proposition 6.6 and rearranging, we get

$$\left(1 - \frac{18L'\,(L+\lambda)^2\,\tau^2\gamma^2 e^2}{\lambda}\right)\left(\overline{F}_\theta^\nu(w^{(t)}) - \overline{F}_\theta^\nu(\overline{w}^\star)\right) \\ \leq \frac{1}{\gamma\tau}\left(1 - \frac{\lambda\gamma\tau}{2}\right)\left\|w^{(t)} - \overline{w}^\star\right\|^2 - \frac{1}{\gamma\tau}\,\mathbb{E}_t\left\|w^{(t+1)} - \overline{w}^\star\right\|^2 \\ + \frac{16\tau G^2\gamma}{\theta m} + \frac{9G^2\,(L+\lambda)^2\,\tau^2\gamma^2 e^2}{\lambda}\left(4 + \frac{8}{\theta m}\right).$$

Since $36e^2 \leq 18^2$ for $\gamma \leq \sqrt{\lambda}(18\tau\,(L+\lambda)\,\sqrt{L'})^{-1}$, we have $18L'\,(L+\lambda)^2\,\tau^2\gamma^2 e^2/\lambda \leq \frac{1}{2}$ which implies:

$$\overline{F}_\theta^\nu(w^{(t)}) - \overline{F}_\theta^\nu(\overline{w}^\star) \leq \frac{2}{\gamma\tau}\left(1 - \frac{\lambda\gamma\tau}{2}\right)\left\|w^{(t)} - \overline{w}^\star\right\|^2 - \frac{2}{\gamma\tau}\,\mathbb{E}_t\left\|w^{(t+1)} - \overline{w}^\star\right\|^2 \\ + \underbrace{\frac{32\tau G^2\gamma}{\theta m} + \frac{18G^2\,(L+\lambda)^2\,\tau^2\gamma^2 e^2}{\lambda}\left(4 + \frac{8}{\theta m}\right)}_{=:\mathcal{T}_1}.$$

Next, we use convexity to get

$$\mathbb{E}\left[\overline{F}_\theta^\nu(\overline{w}^{(T)}) - \overline{F}_\theta^\nu(\overline{w}^\star)\right] \leq \frac{1}{\sum_{t=0}^T\left(1 - \frac{\lambda\gamma\tau}{2}\right)^{-(1+t)}}\sum_{t=0}^T\left(1 - \frac{\lambda\gamma\tau}{2}\right)^{-(1+t)}\mathbb{E}\left[\overline{F}_\theta^\nu(w^{(t)}) - \overline{F}_\theta^\nu(\overline{w}^\star)\right] \\ \leq \frac{2\sum_{t=0}^T\left(1 - \frac{\lambda\gamma\tau}{2}\right)^{-t}\mathbb{E}\left[\left\|w^{(t)} - \overline{w}^\star\right\|^2\right] - \left(1 - \frac{\lambda\gamma\tau}{2}\right)^{-(1+t)}\mathbb{E}\left[\left\|w^{(t+1)} - \overline{w}^\star\right\|^2\right]}{\gamma\tau\sum_{t=0}^T\left(1 - \frac{\lambda\gamma\tau}{2}\right)^{-(1+t)}} + \mathcal{T}_1,$$

so that telescoping the sum yields

$$\mathbb{E}\left[\overline{F}_\theta^\nu(\overline{w}^{(T)}) - \overline{F}_\theta^\nu(\overline{w}^\star)\right] \leq \frac{2\left\|w^{(0)} - \overline{w}^\star\right\|^2}{\gamma\tau\sum_{t=0}^T\left(1 - \frac{\lambda\gamma\tau}{2}\right)^{-(1+t)}} + \mathcal{T}_1\,.$$

Now, we can lower bound the denominator with

$$\sum_{t=0}^T\left(1 - \frac{\lambda\gamma\tau}{2}\right)^{-(1+t)} \geq \frac{1}{\gamma\tau\lambda}e^{T\gamma\tau\lambda},$$

to get the bound

$$\mathbb{E}\left[\overline{F}_\theta^\nu(\overline{w}^{(T)}) - \overline{F}_\theta^\nu(\overline{w}^\star)\right] \leq 2\lambda e^{-T\gamma\tau\lambda}\left\|w^{(0)} - \overline{w}^\star\right\|^2 + \mathcal{T}_1\,. \tag{6.13}$$

It remains to translate the results on $\overline{F}_\theta^\nu$ into F_θ. For the left hand side, we use the bias bound of Property 3.13. For the right hand side, we use the λ-strong convexity of F_θ and Property 3.13 we have:

$$\begin{aligned}\left\|w^{(0)} - \overline{w}^\star\right\|^2 &\leq 2\left\|w^{(0)} - w^\star\right\|^2 + 2\left\|\overline{w}^\star - w^\star\right\|^2 \leq 2\left\|w^{(0)} - w^\star\right\|^2 + \frac{4}{\lambda}\left(F_\theta(\overline{w}^\star) - F_\theta(w^\star)\right) \\ &\leq 2\left\|w^{(0)} - w^\star\right\|^2 + \frac{4}{\lambda}\left(F_\theta(\overline{w}^\star) - \overline{F}_\theta^\nu(\overline{w}^\star) + \overline{F}_\theta^\nu(\overline{w}^\star) - \overline{F}_\theta^\nu(w^\star) + \overline{F}_\theta^\nu(w^\star) - F_\theta(w^\star)\right) \\ &\leq 2\left\|w^{(0)} - w^\star\right\|^2 + \frac{4}{\lambda}\left(\frac{2B}{\sqrt{\theta m}} + 4\nu\log m\right)\end{aligned}$$

since $\overline{F}_\theta^\nu(\overline{w}^\star) - \overline{F}_\theta^\nu(w^\star) \leq 0$. Plugging this into (6.13) completes the proof. □

6.5 NUMERICAL EXPERIMENTS

In this section, we describe in details the experimental setup and the results. Here is its outline:

- Section 6.5.1 describes the datasets and tasks.
- Section 6.5.2 gives a detailed description of the hyperparameters used and the evaluation methodology.
- Section 6.5.3 details the experimental results we obtain for our algorithm Δ-*FL* in comparison to a broad selection of baselines.

The code and the scripts to reproduce results are made publicly available at

`https://github.com/krishnap25/simplicial-fl`.

6.5.1 *Datasets and tasks*

We conduct our experiments on two datasets from computer vision and natural language processing, described in detail below. These datasets contain a natural, non-iid split of data which is reflective of data heterogeneity encountered in federated learning. In these two examples, each device has a finite number of datapoints. Thus, we let its probability distribution q_k to be the empirical distribution over the available examples, and the weight α_k to be proportional to the number of datapoints available on the device. The data was preprocessed using LEAF [24].

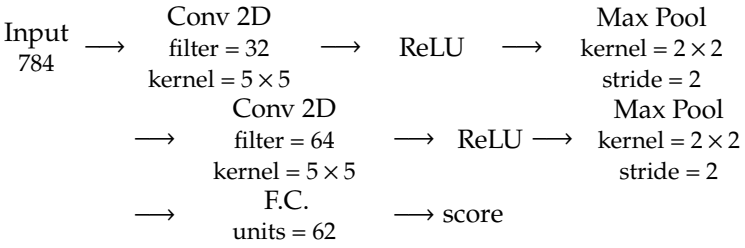
EMNIST FOR HANDWRITTEN-LETTER RECOGNITION.

Dataset. EMNIST [32] is a character recognition dataset. This dataset contains images of handwritten digits or letters, labeled with their identification (a-z,A-Z, 0-9). The images are grey-scaled pictures of $28 \times 28 = 784$ pixels.

Train and Test Devices. Each image is also annotated with the "writer" of the image, i.e., the human subject who hand-wrote the digit/letter during the data collection process. A device consists in all the images supplied by the same writer. From this set of devices, we discard all devices containing less than 100 images. The remaining devices were partitioned into two groups — 1730 training and 1730 testing devices. For each experiment we subsampled 865 training and 865 testing devices for computational tractability.

Model. We consider the following models for this task.

- **Linear Model**: We use a linear softmax regression model. In this case each F_k is convex. We train parameters $w \in \mathbb{R}^{62\times 784}$. Given an input image $x \in \mathbb{R}^{784}$, the score of each class $c \in [\![1, 62]\!]$ is the dot product $\langle w_c, x\rangle$. The probability p_c assigned to each class is then computed as a softmax: $p_c = \exp\langle w_c, x\rangle / \sum_{c'} \exp\langle w_{c'}, x\rangle$. The prediction for a given image is then the class with the highest probability.
- **ConvNet**: We also consider a convolutional neural network. Its architecture satisfies the following scheme:



  In other words, it contains two convolutional layers with max-pooling and one fully connected layer (F.C) which outputs a vector in $\mathbb{R}^{62}$. The outputs of the ConvNet are scores with respect to each class. They are also used with a softmax operation to compute probabilities.

The loss used to train both models is the multinomial logistic loss $L(p, y) = -\log p_y$, where p denotes the vector of probabilities computed by the model, and p_y denotes its y^{th} component. In the convex case we add a quadratic regularization term of the form $(\lambda/2)\|w\|_2^2$.

SENT140 FOR SENTIMENT ANALYSIS.

Dataset. Sent140 [54] is a text dataset of 1,600,498 tweets produced by 660,120 Twitter accounts. Each tweet is represented by a character string with emojis redacted. Each tweet is labeled with a binary sentiment reaction (i.e., positive or negative), which is inferred based on the emojis in the original tweet.

Train and Test Devices. Each device represents a twitter account and contains only tweets published by this account. From this set of devices we discarded all devices containing less that 50 tweets, and split the 877 remaining devices into a train set and a test set of sizes 438 and 439 respectively. This split was held fixed for all experiments. Each word in the tweet is encoded by its 50-dimensional GloVe embedding [124].

Model. We consider the following models.

- **Linear Model**: We consider a l_2-regularized linear logistic regression model where the parameter vector w is of dimension 50. In this case, each F_k is convex. We summarize each tweet by the average of the GloVe embeddings of the words of the tweet.
- **RNN**: The nonconvex model is a Long Short Term Memory (LSTM) model [65] built on the GloVe embeddings of the words of the tweet. The hidden dimension of the LSTM is same as the embedding dimension, i.e., 50. We refer to it as "RNN".

The loss function is the binary logistic loss. In the convex case, we also add a quadratic regularization term of the form $(\lambda/2)\|w\|_2^2$

6.5.2 *Algorithms, hyperparameters and evaluation strategy*

ALGORITHM AND BASELINES.
The proposed Δ-*FL* is run for three values of $\theta \in \{0.8, 0.5, 0.1\}$. We compare it to the following baselines:

- *FedAvg* [109]: It is the de facto standard for the vanilla federated learning objective.
- *FedAvg*, θ: We also consider *FedAvg* with a random device filtering step: local updates are run on a fraction of the initial number of devices randomly selected per round. For each dataset, we try three fraction values, corresponding to the average number of devices selected by Δ-*FL* for the three values of θ used (cf. Figure 6.11). We report as *FedAvg-Sub* the performance of *FedAvg*, θ with $\theta \in \{0.8, 0.5, 0.1\}$ which gives the best performance on Δ-*FL* (i.e., lowest 90^{th} percentile of test misclassification error).
- *FedProx* [96]: It augments *FedAvg* with a proximal term but still minimizes the vanilla federated learning objective.
- q-*FFL* [97]: It raises the per-device losses to the power $(1 + q)$, where $q \geq 0$ is a parameter, in order to focus on devices with higher loss.
- *AFL* [114]: It aims to minimize the worst per-device loss. We implement it as an asymptotic version of q-FFL, using a large value of q, as this was found to yield better convergence with comparable performance [97]. In the experiments we take $q = 10.0$.
- *Tilted-ERM* [98]: It aims at minimizing a parameterized variant of *logsum-exp* function over the per-device losses.

The experiments are conducted on the datasets described in Section 6.5.1.

HYPERPARAMETERS.

Rounds. We measure the progress of each algorithm by the number of calls to secure aggregation routine for weight vectors, i.e., the number of communication rounds.

For the experiments, we choose the number of communication rounds depending on the convergence of the optimization for *FedAvg*. For the EMNIST dataset, we run the algorithm for 3000 communication rounds with the linear model and 1000 for the ConvNet. For the Sent140 dataset, we run the 1000 communication rounds for the linear model and 600 for the RNN.

Devices per Round. We choose the same number of devices per round for each method, with the exception of *FedAvg*, θ. All devices are assumed to be available and selections are made uniformly at random. In particular, we select 100 devices per round for all experiments with the exception of Sent140 RNN for which we used 50 devices per round.

Local Updates and Minibatch Size. Each selected device locally runs 1 epoch of mini-batch stochastic gradient descent. The effect of this choice of local epochs is explored further at the end of Section 6.5. We used the default mini-batch of 10 for all experiments [109], except for 16 for EMNIST ConvNet. This is

because the latter experiments were run using on a GPU, as we describe in the forthcoming paragraph on the hardware.

Learning rate scheme. We now describe the learning rate γ_t used during *LocalUpdate*. For the linear model we used a constant fixed learning rate $\gamma_t \equiv \gamma_0$, while for the neural network models, we used a step decay scheme $\gamma_t = \gamma_0 c^{-\lfloor t/t_0 \rfloor}$ where γ_0 and $0 < c \leq 1$ are tuned. We tuned these parameters only for the baseline *FedAvg* and used the same learning rate for the other baselines and Δ-*FL* at all values of θ.

For the neural network models, we fixed t_0 so that the learning rate was decayed once or twice during the fixed time horizon T. In particular, we used $t_0 = 400$ for EMNIST ConvNet (where $T = 1000$), and $t_0 = 200$ for Sent140 RNN (where $T = 600$). We tuned c from the set $\{2^{-3}, 2^{-2}, 2^{-1}, 1\}$, while the choice of the range of γ_0 depended on the dataset-model pair. The tuning criterion we used was the mean of the loss distribution over the training devices (with device k weighted by α_k) at the end of the time horizon. That is, we chose the γ_0, c which gave the best terminal training loss.

Tuning of the regularization parameter. The regularization parameter λ for linear models was tuned with cross validation from the set $\{10^{-k} : k \in \{3, \cdots, 8\}\}$. This was performed as described below.

For each dataset, we held out half the training devices as validation devices. Then, for different values of the regularization parameter, we trained a model with the (smaller subset of) training devices and evaluate its performance on the validation devices. We selected the value of the regularization parameter as the one which gave the smallest 90^{th} percentile of the misclassification error on the validation devices.

Baselines Parameters. We tune the proximal parameter of *FedProx* with cross validation. The procedure we followed is identical to the procedure we described above for the regularization parameter λ. The set of parameters tested is $\{10^{-j}, j \in \{0, \ldots, 3\}\}$. We adopt the same strategy for q-FFL, where the set of parameters q tested is $\{10^j, j \in \{-3, \cdots, 1\}\}$, and *Tilted-ERM*, where the set of temperatures t tested is $\{0.1, 0.5, 1., 5., 10., 50., 100., 200\}\}$.

EVALUATION STRATEGY AND OTHER DETAILS.

Evaluation metrics. We record the loss of each training device and the misclassification error of each testing device, as measured on its local data.

The evaluation metrics noted in Section 6.5.3 are the following : the weighted mean of the loss distribution over the training devices, the (unweighted) mean misclassification error over the testing devices, the weighted p-percentile of the loss over the training device and the (unweighted) p-percentile of the misclassification error over the testing devices for values of p among $\{20\%, 50\%, 60\%, 80\%, 90\%, 95\%\}$. The weight α_k used for training device k was set proportional to the number of datapoints on the device.

Evaluation times. We evaluate the model during training process for once every l communication rounds. The value of l used was $l = 50$ for EMNIST linear model, $l = 10$ for EMNIST ConvNet, $l = 20$ for Sent140 linear model and $l = 25$ for Sent140 RNN.

Hardware. We run each experiment as a simulation as a single process. The linear models were trained on m5.8xlarge AWS instances, each with an Intel Xeon Platinum 8000 series processor with 128 GB of memory running at most 3.1 GHz. The neural network experiments were trained on workstation with an Intel i9 processor with 128 GB of memory at 1.2 GHz, and two Nvidia Titan

Xp GPUs. The Sent140 RNN experiments were run on a CPU while the other neural network experiments were run using GPUs.

Software Packages. Our implementation is based on NumPy using the Python language. In the neural network experiments, we use PyTorch to implement the *LocalUpdate* procedure, i.e., the model itself and the automatic differentiation routines provided by PyTorch to make SGD updates.

Randomness. Since several sampling routines appear in the procedures such as the selection of devices or the local stochastic gradient, we carry our experiments with five different seeds and plot the average metric value over these seeds. Each simulation is run on a single process. Where appropriate, we report one standard deviation from the mean.

6.5.3 *Experimental results*

We now present the experimental results of the paper.

- We present different metrics on the distribution of test misclassification error over the devices, comparing Δ-*FL* to baselines.
- We study the convergence of Algorithm 5 for Δ-*FL* over the course of the optimization, and compare it with *FedAvg*.
- We plot the histograms of the distribution of losses over train devices as well as the test misclassification errors over test devices at the end of the training process.
- We present in the form of scatter plots the training loss and test misclassification error across devices achieved at the end of training, versus the number of local data points on the device.
- We present the number of devices selected at each communication round for Δ-*FL* (after device filtering).
- We finally present the impact of the number of local epochs on the convergence of Δ-*FL*.

Comparison to Baselines. In Tables 6.1 to 6.4, we present a comparison of various statistics of the test misclassification error distribution for different methods. For each column, the smallest mean over five random runs is highlighted in bold. Further, if no other method is within one standard deviation of this method, the entire entry (i.e., mean $\pm$ std) is highlighted in bold. Our main findings are summarized below.

Δ-FL consistently achieves the smallest 90^{th} percentile error

Δ-*FL* achieves a 3.3% absolute (12% relative) improvement over any vanilla FL objective on EMNIST-ConvNet. Among the heterogeneity aware objectives, Δ-*FL* achieves 1.8% improvement over the next best objective, which is *Tilted-ERM*. We note that q-*FFL* marginally outperforms Δ-*FL* on Sent140-Linear, but the difference 0.05% is much smaller than the standard deviation across runs.

Δ-FL is competitive at multiple values of θ

For EMNIST-ConvNet, Δ-*FL* with $\theta \in \{0.5, 0.8\}$ is better in 90^{th} percentile error than *all* other methods we compare to, and Δ-*FL* with $\theta = 0.1$ is tied with *Tilted-ERM*, the next best method. We also empirically confirm that Δ-*FL* interpolates between *FedAvg* ($\theta \to 1$) and *AFL* ($\theta \to 0$).

Δ-FL works best for larger values of conformity levels

We observe that Δ-*FL* with $\theta = 0.1$ is unstable for Sent140-RNN. This is consistent with our discussion following Theorem 6.1, where we advocate for values m larger than $1/\theta$. Indeed, this can be explained by Δ-*FL*'s sparse re-weighting, which only gives non-zero weights to $\theta m = 5$ devices on average in each round.

Δ-FL is yet competitive in terms of average error

Perhaps surprisingly, Δ-*FL* actually gets the best test error performance on

EMNIST-ConvNet and Sent140-Linear. This suggests that the average test distribution is shifted relative to the average training distribution p_α. In the other cases, we find that the reduction in mean error is small relative to the gains in the 90^{th} percentile error compared to Vanilla FL methods.

Minimizing superquantile loss over all devices performs better than minimizing worst error over all devices

Specifically, *AFL* which aims to minimize the worst error among all devices, as well as other objectives which approximate it (Δ-*FL* with $\theta \to 0$, q-*FFL* with $q \to \infty$, *Tilted-ERM* with $\nu \to 0$) tend to achieve poor performance. We find that *AFL* achieves the highest error both in terms of 90^{th} percentile and the mean. Δ-*FL* offers a more nuanced and more effective approach via the constraint set $\text{conf}(p_\pi) \geq \theta$ than the straight pessimistic approach minimizing the worst error among all devices.

Method	Mean	Standard Deviation	10^{th} Percentile	Median	90^{th} Percentile
FedAvg	34.38 ± 0.38	18.39 ± 0.33	21.54 ± 0.35	32.61 ± 0.39	49.65 ± 0.67
FedAvg $\theta = 0.5$	34.51 ± 0.47	18.21 ± 0.30	21.40 ± 0.36	32.36 ± 0.59	50.28 ± 0.77
FedProx	33.82 ± 0.30	18.25 ± 0.23	21.37 ± 0.35	31.75 ± 0.20	49.15 ± 0.74
q-FFL (Best $q = 1.0$)	34.71 ± 0.27	19.34 ± 0.30	22.33 ± 0.41	32.80 ± 0.23	49.90 ± 0.58
Tilted-ERM (Best $t = 1.0$)	34.15 ± 0.25	10.78 ± 0.30	22.43 ± 0.29	32.36 ± 0.23	48.59 ± 0.62
AFL	39.32 ± 0.27	25.42 ± 0.27	28.64 ± 0.43	38.16 ± 0.34	51.62 ± 0.28
Δ-FL $\theta = 0.8$	34.48 ± 0.26	19.16 ± 0.32	22.24 ± 0.32	32.85 ± 0.31	49.10 ± 0.24
Δ-FL $\theta = 0.5$	35.01 ± 0.20	20.46 ± 0.34	23.64 ± 0.22	33.83 ± 0.34	48.44 ± 0.38
Δ-FL $\theta = 0.1$	38.32 ± 0.48	23.86 ± 0.59	27.27 ± 0.64	37.52 ± 0.67	50.34 ± 0.95

Table 6.1: Metrics for the test misclassification error for EMNIST (Linear Model).

Method	Mean	Standard Deviation	10^{th} Percentile	Median	90^{th} Percentile
FedAvg	16.63 ± 0.50	4.94 ± 0.14	6.43 ± 0.24	15.34 ± 0.37	28.46 ± 1.07
FedAvg $\theta = 0.5$	16.22 ± 0.23	5.06 ± 0.17	6.47 ± 0.28	15.05 ± 0.25	27.56 ± 0.81
FedProx	16.01 ± 0.54	5.16 ± 0.32	6.68 ± 0.44	14.88 ± 0.29	27.01 ± 1.86
q-FFL (Best $q = 0.001$)	16.58 ± 0.30	5.05 ± 0.21	6.53 ± 0.20	15.40 ± 0.43	28.02 ± 0.80
Tilted-ERM (Best $t = 1.0$)	15.69 ± 0.38	7.31 ± 0.68	7.26 ± 0.51	**14.66** ± 0.16	25.46 ± 1.49
AFL	33.00 ± 0.37	20.38 ± 0.23	22.92 ± 0.23	31.58 ± 0.27	45.07 ± 1.00
Δ-FL $\theta = 0.8$	16.08 ± 0.40	5.60 ± 0.14	7.31 ± 0.29	14.85 ± 0.48	26.23 ± 1.15
Δ-FL $\theta = 0.5$	15.48 ± 0.30	6.13 ± 0.15	8.08 ± 0.16	14.73 ± 0.22	23.69 ± 0.94
Δ-FL $\theta = 0.1$	16.37 ± 1.03	6.61 ± 0.42	8.28 ± 0.65	15.49 ± 1.03	25.45 ± 2.77

Table 6.2: Metrics for the test misclassification error for EMNIST (ConvNet Model).

Method	Mean	Standard Deviation	10^{th} Percentile	Median	90^{th} Percentile
FedAvg	34.74 ± 0.31	12.16 ± 0.15	21.89 ± 0.24	34.81 ± 0.38	46.83 ± 0.54
FedAvg $\theta = 0.8$	34.47 ± 0.03	12.08 ± 0.16	21.69 ± 0.26	34.62 ± 0.17	46.59 ± 0.38
FedProx	34.74 ± 0.31	12.16 ± 0.15	21.89 ± 0.24	34.82 ± 0.39	46.83 ± 0.54
q-FFL (Best $q = 1.0$)	34.48 ± 0.06	11.96 ± 0.14	21.61 ± 0.24	34.57 ± 0.16	46.38 ± 0.40
Tilted-ERM (Best $t = 1.0$)	34.71 ± 0.31	12.00 ± 0.14	21.83 ± 0.34	34.91 ± 0.39	46.70 ± 0.50
AFL	35.97 ± 0.08	11.83 ± 0.09	23.58 ± 0.28	36.09 ± 0.17	47.51 ± 0.32
Δ-FL $\theta = 0.8$	34.41 ± 0.22	12.17 ± 0.11	21.77 ± 0.34	34.64 ± 0.25	46.44 ± 0.38
Δ-FL $\theta = 0.5$	35.28 ± 0.25	11.68 ± 0.40	23.03 ± 0.38	35.55 ± 0.53	46.64 ± 0.41
Δ-FL $\theta = 0.1$	37.78 ± 0.89	12.86 ± 0.52	23.93 ± 0.99	37.80 ± 1.30	51.38 ± 1.07

Table 6.3: Metrics for the test misclassification error for Sent140 (Linear Model).

Method	Mean	Standard Deviation	10^{th} Percentile	Median	90^{th} Percentile
FedAvg	30.16 ± 0.44	4.36 ± 1.26	10.06 ± 2.06	29.51 ± 0.33	49.66 ± 3.95 1
FedAvg $\theta = 0.8$	29.85 ± 0.46	5.39 ± 1.32	11.90 ± 2.27	29.57 ± 0.31	46.93 ± 3.84 1
FedProx	30.20 ± 0.48	4.35 ± 1.23	10.37 ± 2.08	29.51 ± 0.32	49.85 ± 4.07
q-FFL (Best $q = 0.01$)	29.99 ± 0.56	4.90 ± 1.66	10.98 ± 2.88	29.56 ± 0.39	48.65 ± 4.68
Tilted-ERM (Best $t = 1.0$)	30.13 ± 0.49	14.17 ± 2.10	13.18 ± 3.33	29.96 ± 0.84	46.54 ± 3.27
AFL	37.74 ± 0.65	9.90 ± 1.46	18.19 ± 1.99	36.95 ± 1.03	57.78 ± 1.19
Δ-FL $\theta = 0.8$	30.30 ± 0.33	6.75 ± 2.68	13.05 ± 3.87	29.92 ± 0.38	46.46 ± 4.39
Δ-FL $\theta = 0.5$	33.58 ± 2.44	8.74 ± 3.98	16.77 ± 6.62	33.28 ± 2.27	50.47 ± 8.24
Δ-FL $\theta = 0.1$	51.97 ± 11.81	9.11 ± 5.47	16.67 ± 9.15	52.44 ± 13.21	86.44 ± 10.95

Table 6.4: Metrics for the test misclassification error for Sent140 (RNN Model).

Performance Across Iterations. We present our results only for the EMNIST dataset. For the other datasets, we point the reader to our papers [87, 88]. We group plots by models and datasets. The x axis of the plots below represents the number of communication rounds along the simulation. The y-axis represents either the training loss or the testing accuracy (either the mean or some percentile).

Overall, Δ-*FL* exhibits better convergence properties for the high percentiles of the distribution of test misclassification errors over the devices: see for instance the 99^{th}, 95^{th} and 90^{th} percentiles in figure 6.5. This comes with the price of lower performance than the baseline *FedAvg* on low percentiles of the distribution.

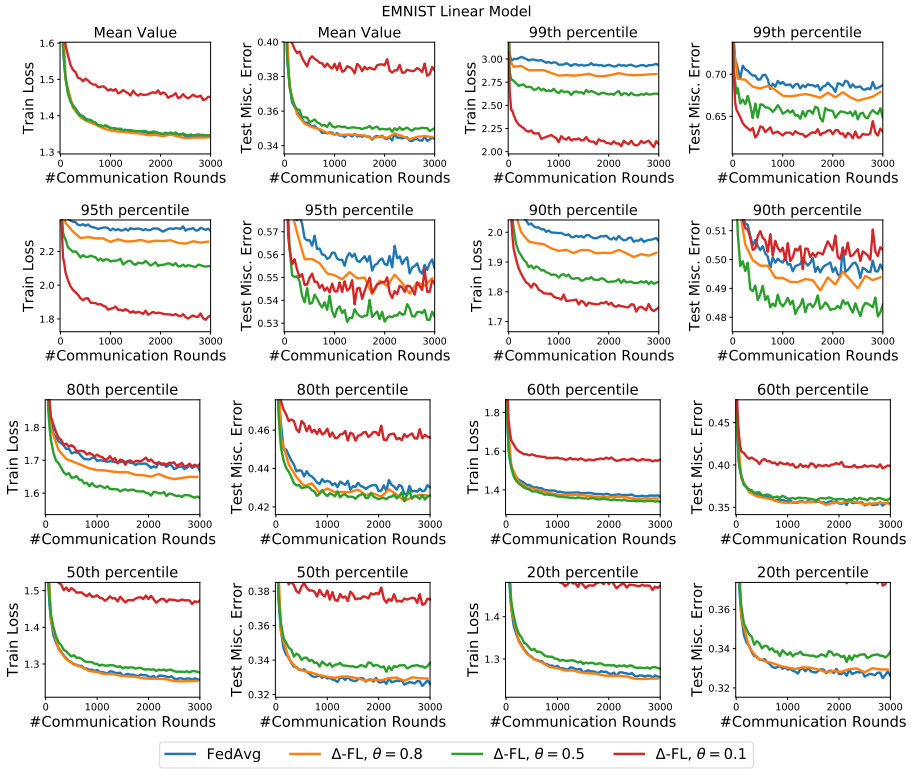


Figure 6.4: Performance across iterations of EMNIST linear model.

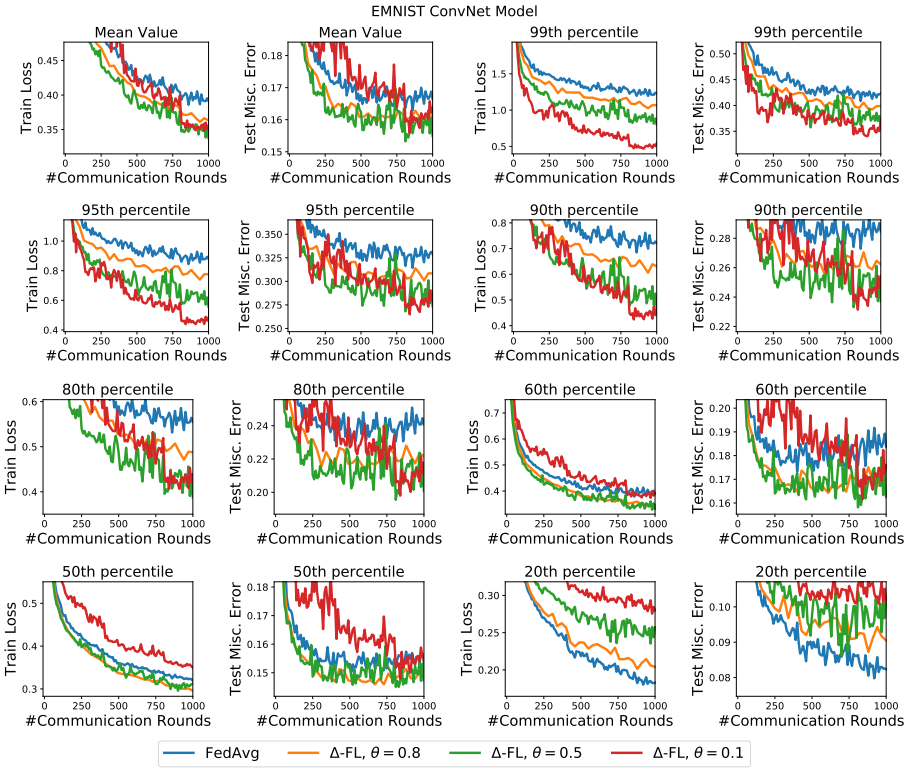


Figure 6.5: Performance across iterations of EMNIST ConvNet model.

Histograms of Loss and Test Misc. Error over Devices. Here, we plot in Figure 6.6 the histograms of the loss distribution over training devices and in Figure 6.7 the histograms of the misclassification error distribution over testing devices for the dataset EMNIST (for other datasets, see [87]). We report the losses and errors obtained at the end of the training process. Each metric is averaged per device over 5 runs of the random seed. We note that Δ-*FL* tends to exhibit thinner upper tails at at multiple values of θ and a lower variance of the distribution in most of the cases. This is also confirmed by the figures in Tables 6.1 to 6.4. This shows the benefit of using Δ-*FL* over vanilla *FedAvg*.

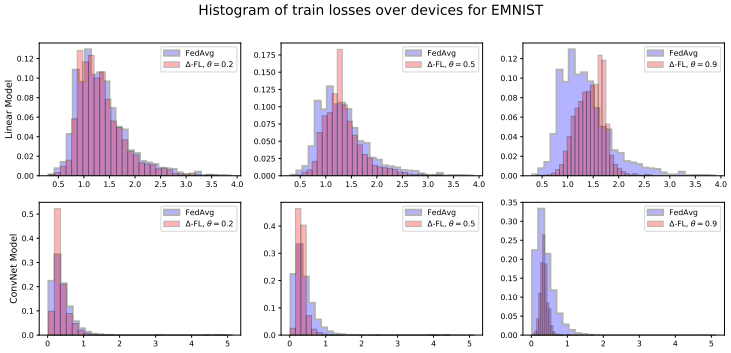


Figure 6.6: Histogram of loss distribution over training devices from EMNIST. Top:linear – Bottom:ConNet).

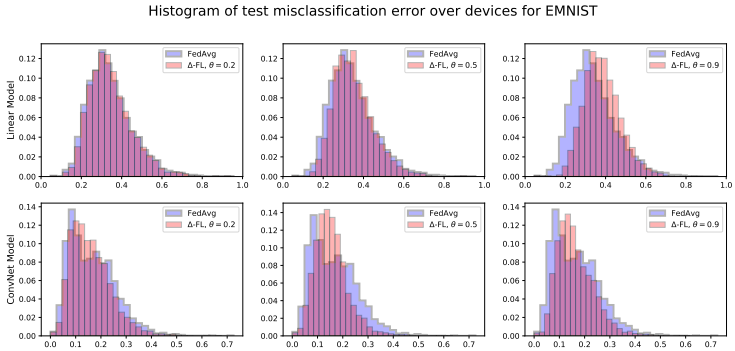


Figure 6.7: Histogram of misclassification error distribution over testing devices for EMNIST Top:linear – Bottom:ConNet).

Performance compared to local data size. Next, we plot the loss on training devices versus the amount of local data on the device and the misclassification error on the test devices versus the amount of local data on the device. See Figures 6.8 and 6.9 for EMNIST and Figure 6.10 for Sent140.

Observe firstly that improvement over the worst cases is achieved regardless of the local data size of the devices. Indeed, the device filtering step operates a sorting of the loss of the devices which does not prevent small devices from being selected. In contrary, *FedAvg*, by averaging with respect to the weights of the devices is likely to put more the accent on the devices with larger local data size. Secondly, Δ-*FL* appears to reduce the variance of the losses on the train devices. Lastly, note that amongst test devices with a small number of data points (e.g., < 200 for EMNIST or < 100 for Sent140), Δ-*FL* reduces the variance of the misclassification error. Both effects are more pronounced on the neural network models.

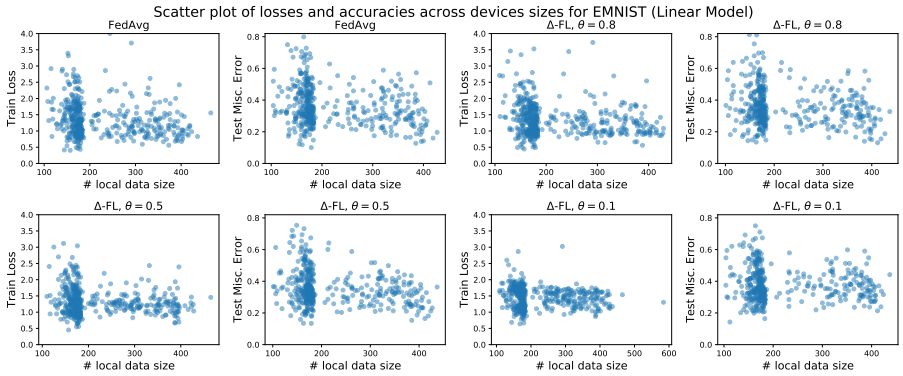


Figure 6.8: For *FedAvg* and Δ-*FL* with values of θ in $\{0.1, 0.5, 0.8\}$, scatter plot of (left) loss on training device vs. amount of local data, and (right) misclassification error on testing device vs. amount of local data for EMNIST (Linear Model)

Number of Devices Selected per Communication Rounds. Next, we plot the number of devices selected per round (after device filtering, if applicable). The shaded area denotes the maximum and minimum over 5 random runs. We see from Figure 6.11 that device-filtering is stable in the number of devices filtered out.

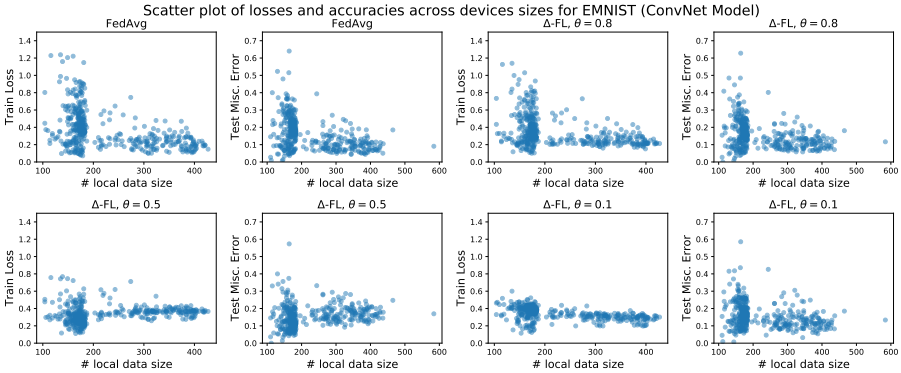


Figure 6.9: For *FedAvg* and Δ-*FL* with values of θ in $\{0.1, 0.5, 0.8\}$, scatter plot of (left) loss on training device vs. amount of local data, and (right) misclassification error on testing device vs. amount of local data for EMNIST (ConvNet Model)

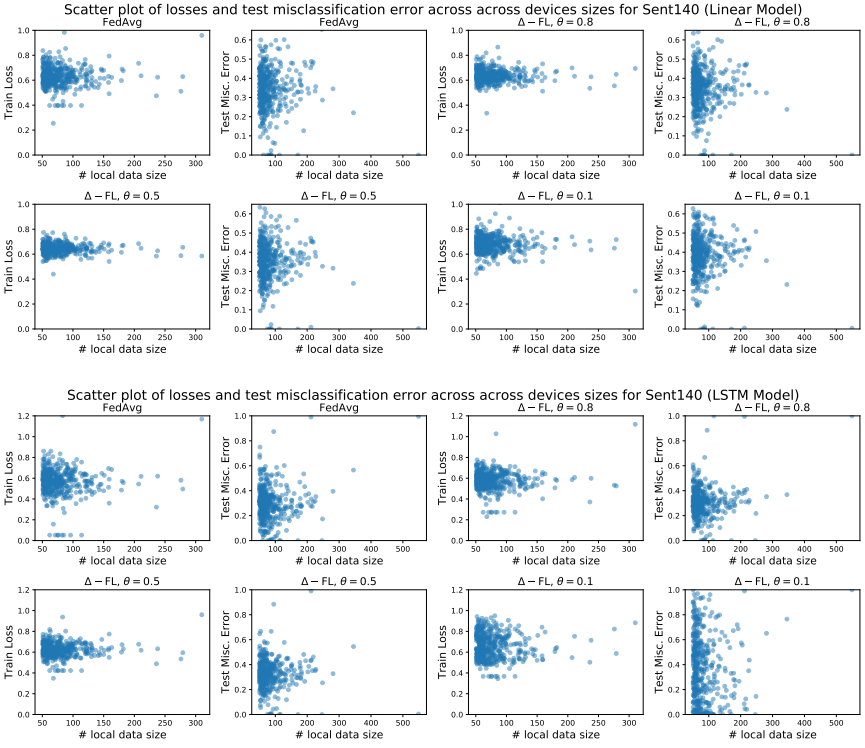


Figure 6.10: For *FedAvg* and Δ-*FL* with values of θ in $\{0.1, 0.5, 0.8\}$, scatter plot of (left) loss on training device vs. amount of local data, and (right) misclassification error on testing device vs. amount of local data for Sent140.

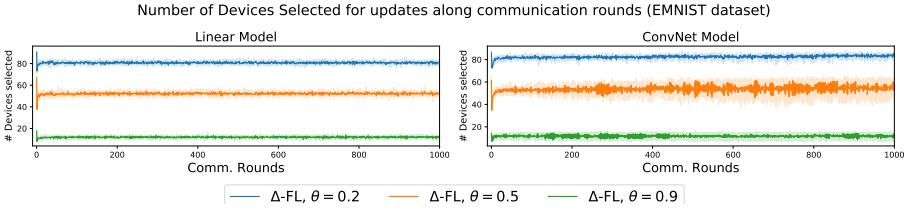


Figure 6.11: Number of devices selected per round (after device filtering) for the EMNIST dataset. The shaded region denotes the maximum and minimum over 5 random runs.

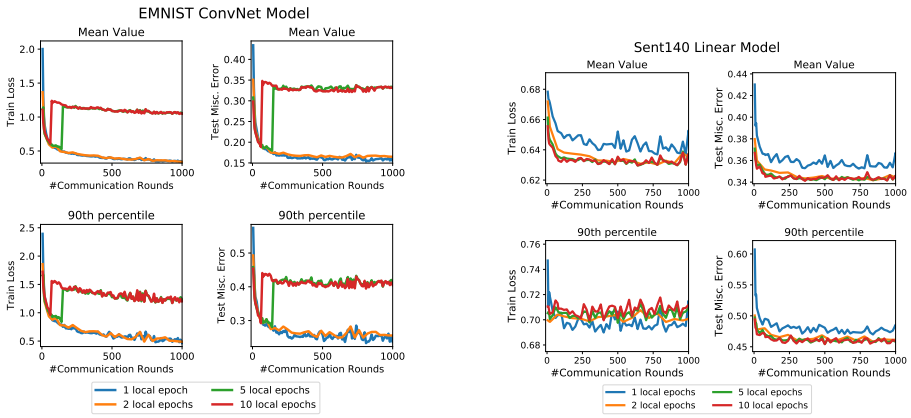


Figure 6.12: Effect of the number of local epochs.

Effect of number of local epochs. We present in Figure 6.12, the effect of the number of epochs in *FedAvg*, on the overall convergence of Δ-*FL*. We observe for EMNIST ConvNet that a large number of local epochs leads to poor convergence. A similar behaviour was observed in [109, Figure 3] for *FedAvg*. On Sent140 linear model, we see a minor improvement for a larger number of local epochs.

6.5.4 *Performing secure aggregation*

In this final section, we show how Δ-*FL* can address privacy concerns in Section 6.5.4.

Recall that Algorithm 5 as stated, requires each selected client device to send its loss to the server for the client filtering step (see line 3). We now present a way to perform this step without any reduction in privacy from directly sending client losses to the server. As noted earlier, line 3 consists in the computation of the superquantile of the sequence of losses $F_k(w^{(t)})$. We saw in the Section 3.3 of this thesis that such computation boils down to computing the $(1-\theta)$-quantile of theses losses. We show now how to implement this quantile computation using secure aggregation.

Setup. Suppose we wish to find the p-quantile of $x_1, \cdots, x_m \in \mathbb{R}$ with respective weights $\alpha_1, \cdots, \alpha_m > 0$. It is known [78, e.g.,] that q_p is a p-quantile iff it it minimizes $H_p : \mathbb{R} \to \mathbb{R}$ defined as

H_p is often referred to as the "pinball" loss

$$H_p(q) := \sum_{k=1}^{m} \alpha_k \varphi_p(x_k - q), \quad \text{where,} \quad \varphi_p(\rho) := \begin{cases} p\rho, & \text{if } \rho \geq 0, \\ -(1-p)\rho, & \text{if } \rho < 0. \end{cases}$$

Algorithm. Recall that secure aggregation can find a weighted mean of vectors (and hence, scalars) distributed across m devices without revealing each device's vector to other devices or the server. We now show how to compute a quantile as an iterative weighted mean, making it amenable to implementation via secure aggregation. The underlying algorithm, based on the principle of majorization-minimization was used, e.g., in [68].

For any $\widetilde{q} \notin \{x_1, \cdots, x_m\}$, define

$$\widetilde{H}_p(q;\widetilde{q}) := \frac{1}{4}\sum_{k=1}^{m} \alpha_k \left[\frac{|x_k - q|^2}{|x_k - \widetilde{q}|} + (4p-2)(x_k - q) + |x_k - \widetilde{q}| \right],$$

as a majorizing surrogate for H_p at $\widetilde{q}$, i.e., $\widetilde{H}_p(\cdot\,;\widetilde{q}) \geq H_p$ and $\widetilde{H}_p(\widetilde{q};\widetilde{q}) = H_p(\widetilde{q})$. Note that $\widetilde{H}_p(q;\widetilde{q})$ is an isotropic quadratic in q.

A majorization-minimization algorithm to minimize H and hence find the p-quantile can thus be given as

$$\begin{aligned} q_{t+1} &= \begin{cases} \arg\min_q \widetilde{H}_p(q;q_t) & \text{if } q_t \notin \{x_1, \cdots, x_m\} \\ x_k & \text{if } q_t = x_k \text{ for some } k \in [m] \end{cases} \\ &= \begin{cases} \frac{\sum_{k=1}^m \beta_{k,t} x_k + (2p-1)}{\sum_{k=1}^m \beta_{k,t}} & \text{if } q_t \notin \{x_1, \cdots, x_m\} \\ x_k & \text{if } q_t = x_k \text{ for some } k \in [m], \end{cases} \end{aligned} \tag{6.14}$$

where

$$\beta_{k,t} = \frac{\alpha_k}{|x_k - q_t|}.$$

Communication Cost of Secure Aggregation For Quantile. Here we show in regular cases that the communication cost of secure aggregation of quantile is typically in the range $0.1 - 2.5\%$ of the cost of secure aggregation of weight vectors (table 6.5 contains a summary of the following discussion). The asymptotic total communication cost of securely aggregating m vectors in $\mathbb{R}^d$ is $\mathcal{O}(m^2 + dm)$ bits [20] or $\mathcal{O}(m \log m + dm \log m)$ bits [151]. In typical cross-device federated learning applications, $m \sim 100 - 500$, while the model dimension is $d \sim 10^6$ [57, 175]. Typically, $n \sim 10 - 50$ iterations of (6.14) will suffice to recover exact filtering (see benchmarking below).

Overhead of secure quantile computation

We now quantify the ratio C_q/C_w of the communication cost of secure aggregation for quantile C_q to that of secure aggregation of weight vectors C_w. In the case of [20], this ratio is

$$\frac{C_q}{C_w} = \frac{n(m^2+m)}{m^2+md} \sim \frac{nm}{d},$$

while for the case of [151], it is

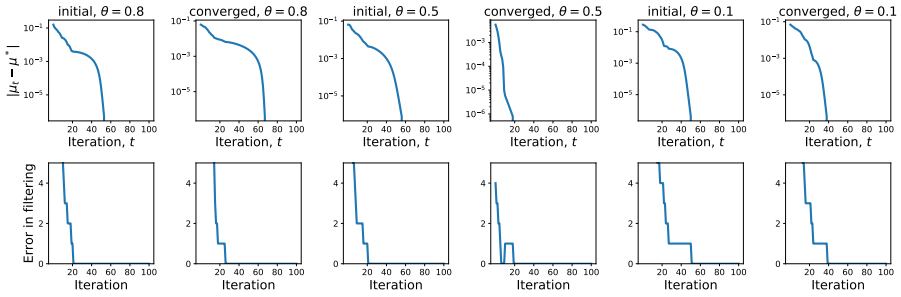
$$\frac{C_q}{C_w} = \frac{n(m\log m)}{m\log m + md\log m} \sim \frac{n}{d}.$$

Thus, the additional overhead of secure aggregation of the quantile is in the range $0.1 - 2.5\%$ under typical range of values for the protocol of [20], while it is in the range $0.01 - 0.05\%$ for the protocol of [151].

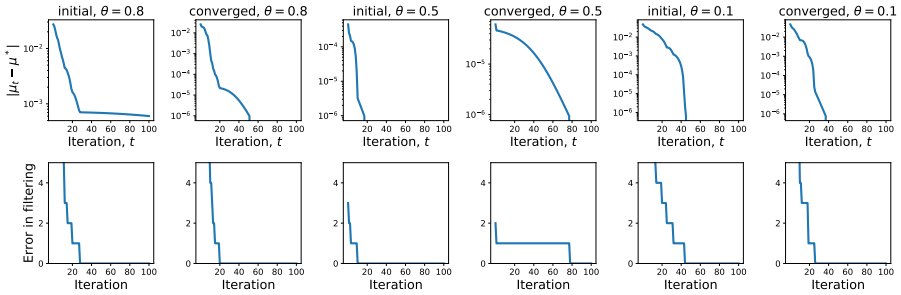
Secure Quantile Computation on Real Data. We now plot the convergence of the secure quantile iterations in (6.14). We follow the experimental setup that will be detailed in the forthcoming Section 6.5 and compute the $(1-\theta)$-quantile with (6.14). We repeat this experiment for $\theta \in \{0.2, 0.5, 0.9\}$ at the 100^{th} iteration ("initial") of Algorithm 5 as well as the the last iteration ("converged"). We plot the difference of the iteration q_t of (6.14) to the quantile q^* as well as the error in filtering, i.e., $|\sum_k \mathbb{I}(x_k \geq q_t) - \sum_k \mathbb{I}(x_k \geq q^*)|$. See Figure 6.13 for plots. We see that the secure update(6.14) converges rapidly and sometimes even finds an

Table 6.5: Comparison of communication cost of (i) secure aggregation for quantile of m scalars with n rounds of (6.14), versus, (ii) secure aggregation of m weight vectors in d dimension. Typical values used are $m \in \{100, 500\}$, $n \in \{10, 50\}$ and $d \sim 10^6$.

Sec.Agg. Protocol	Comm. Cost of quantile, C_q	Comm. Cost of weight vector, C_w	Ratio C_q/C_w	Range of C_q/C_w
[20]	$nm^2 + nm$	$n^2 + dm$	nm/d	0.1% to 2.5%
[151]	$nm \log m$	$m \log m + dm \log m$	n/d	0.01% to 0.05%



(a) EMNIST ConvNet.



(b) Sent140 Linear Model.

Figure 6.13: Convergence of Secure $(1-\theta)$-Quantile using Iteration (6.14). The first row denotes the error in the quantile estimate q_t of (6.14), while the second wrong row gives the error in filtering, i.e., difference in the number $x_k \geq q_t$ of devices filtered out q_t vs. the same number $x_k \geq q^*$ for q^*.

exact quantile. Moreover, it can find an estimate with zero filtering errors in $10-50$ updates.

6.6 CONCLUSION

In this chapter, we presented the Δ-*FL* framework that operates with heterogeneous client devices while still guaranteeing a minimal level of predictive performance to each individual device. We modeled the similarity between client data distributions using the conformity, which is a scalar summary of how closely a client device conforms to the population. Δ-*FL* relies on a superquantile-based objective, parameterized by the conformity, to minimize the tail statistics of the prediction errors on the client data distributions. We presented a federated optimization algorithm compatible with secure aggregation, which interleaves device reweighting steps with local stochastic gradient methods. We derived finite time convergence guarantees in the convex setting.

Experimental results on federated learning benchmarks demonstrate superior performances of Δ-*FL* over state-of-the-art baselines on the upper quantiles of the error on test devices, with particular improvements on data-poor devices, while being competitive on the mean error.

7

CONCLUSION AND PERSPECTIVES

Conclusion

In this manuscript, we considered two related topics, in optimization with (super)quantiles. The first topic is the minimization of superquantile-based objectives for machine learning in both centralized and federated settings. The second topic is the solving of non-convex problems with quantile constraints, for which we proposed a bilevel reformulation. Our work leverages the strong interplay between convex analysis and risk aversion to derive new optimization procedures in practical data-driven contexts.

In Chapter 4, we highlighted the benefits of using the superquantile in machine learning. Our software offers the possibility to seamlessly optimize a learning loss with the superquantile in place of the expectation, for classical learning tasks. We provided various numerical experiments that illustrate two features of superquantile-based learning: its impact on worst-case scenarios and its robustness to adversarial distributional shifts. Our procedures are based on a thorough analysis of the smoothing techniques applied to the superquantile. We also proposed an extension of such smoothing to law-invariant coherent comonotone risk measures to provide decision makers with a broader set of risk modeling tools without computational overhead.

In Chapter 5, we considered the solving of chance-constrained problems. We leveraged the dual formulation of the superquantile to propose a bilevel reformulation with convex upper and lower levels. In this second line of work (and contrary to the previous contribution) we exploited the nonsmoothness of the superquantile to derive a (semi-)exact penalisation procedure for our bilevel reformulation. Our numerical experiments show that this approach can scale on datasets which are difficulet to solve by direct (MINLP) approaches.

Finally, in Chapter 6, we proposed a new federated learning framework to address statistical heterogeneity among users. We provided a theoretical analysis that puts into perspective the properties of stochastic estimators of the superquantile and their impact on the convergence of local gradient schemes. We produced extensive numerical experiences showing the benefits of a superquantile approach over other state of the art methods, for handling of non-conforming users.

Further Perspectives

Each project presented above is just one step ahead and calls for further developments. They indeed open to some fundamental problems as well as practical incremental developments. Here are a few interesting research questions that triggered my attention.

How to fix the safety parameter p in superquantile-based learning ? Knowing how to set the parameter p in the definition of the superquantile is a question that I have often seen myself asked when presenting my work. First I want to say that this is essentially a modeling problem. In federated learning, for example, setting a parameter p amounts to focus training only on the $(1-p)$-proportion of worst-performing users. Should we focus only on the worst 5% ? the worst

10% ? Should this be decided based on the proportion of each user's community in the network? One could thus argue that the choice of p is mainly a political issue. For now, state-of-the-art works run a cross-validation on a grid of values of p, typically between 0.8 and 0.99 and select the value giving the best accuracy on the validation set. However, with the rise of quantitative methods to measure the fairness of a given prediction models, it may be interesting to propose *automatic* tuning procedures for tuning p. This is the main question I would like to tackle over the next months.

Can superquantiles help to model risk-aversion beyond law-invariant comonotone risk measures ? Actually, this question has already been partially answered in [18], where the authors propose a duality result which links Wasserstein ambiguity on the probability of misclassification for a given model to the superquantile of a computable associated quantity. Based on the dual properties of the superquantiles which were exposed in Chapter 3, I believe this could help to develop tractable approaches for a broad class of adversarial problems based on Wasserstein ambiguity.

Can we derive better convergence guarantees for chance-constrained problems ? The bilevel reformulation of chance-constrained programs I proposed in this thesis paves the way for several future directions. First the characterization of optimality of the solutions of our algorithm may be further investigated. For now, we are able only to build a sequence of critical points of the penalized problems that we derived based on our (semi)-exact penalization procedures. Whether or not cluster points of this sequence exhibit some notions of criticality for the initial chance constrained program remains an intriguing question to solve. Second, the procedure for the partial update of the penalization parameters of such problems remains hard in practice since no prior information (beyond the lower bound of Theorem 5.5) is available. In other words, our algorithm is primal only. If we managed to turn it into a primal-dual procedure with an automatic update of the penalization (i.e. dual) parameters, we can hope for a better convergence in practice. Finally, the derivation of (local) convergence rate for chance constrained programs is a direction I'd like to think of in medium term.

How to extend federated learning to decentralized frameworks ? The challenges raised by federated learning (see Section 2.2.2) have given rise to a number of works in the distributed optimization community. I would be interested in addressing these challenges in a fully decentralized setting, i.e. in the absence of a central server. On top of the existing theoretical challenges of federated optimization, the establishment of a consensus among the nodes of the network raises additional questions which may be tackled with a unified analysis. This would open the door to more resilient frameworks for a number of applications where communication possibilities between the nodes are scarce and cannot be handled by a central entity (e.g., for wireless sensor networks, IoT-enabled edge devices, etc.).

BIBLIOGRAPHY

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69. PMLR, 2018.

[2] Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. Fair regression: Quantitative definitions and reduction-based algorithms. In *International Conference on Machine Learning*, pages 120–129. PMLR, 2019.

[3] Shabbir Ahmed and Alexander Shapiro. Solving chance-constrained stochastic programs via sampling and integer programming. In *State-of-the-art decision-making tools in the information-intensive age*, pages 261–269. Informs, 2008.

[4] Laetitia Andrieu, Guy Cohen, and Felisa J Vázquez-Abad. Gradient-based simulation optimization under probability constraints. *European Journal of Operational Research*, 212(2):345–351, 2011.

[5] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent Measures of Risk. *Mathematical finance*, 9(3):203–228, 1999.

[6] Léonard Bacaud, Claude Lemaréchal, Arnaud Renaud, and Claudia Sagastizábal. Bundle methods in stochastic optimal power management: A disaggregated approach using preconditioners. *Computational Optimization and Applications*, 20 (3):227–244, 2001.

[7] Gilles Bareilles, Yassine Laguel, Dmitry Grishchenko, Franck Iutzeler, and Jérôme Malick. Randomized progressive hedging methods for multi-stage stochastic programming. *Annals of Operations Research*, 295(2):535–560, 2020.

[8] Amir Beck and Marc Teboulle. Smoothing and First Order Methods: A Unified Framework. *SIAM Journal on Optimization*, 22(2):557–580, 2012.

[9] A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming, series A*, 88:411–424, 2000.

[10] Aharon Ben-Tal and Marc Teboulle. Expected utility, penalty functions, and duality in stochastic nonlinear programming. *Management Science*, 32(11):1445–1466, 1986.

[11] Aharon Ben-Tal and Marc Teboulle. An Old-New Concept of Convex Risk Measures: The Optimized Certainty Equivalent. *Mathematical Finance*, 17(3):449–476, 2007.

[12] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization*, volume 28. Princeton University Press, 2009.

[13] Leonard Berrada, Andrew Zisserman, and M. Pawan Kumar. Smooth loss functions for deep top-k classification. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[14] Quentin Bertrand, Quentin Klopfenstein, Mathieu Blondel, Samuel Vaiter, Alexandre Gramfort, and Joseph Salmon. Implicit differentiation of lasso-type models for hyperparameter optimization. In *International Conference on Machine Learning*, pages 810–821. PMLR, 2020.

[15] D.P. Bertsekas. *Convex Optimization Algorithms*. Athena Scientific, 2015.

[16] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2016.

[17] Dimitris Bertsimas and David B Brown. Constructing uncertainty sets for robust linear optimization. *Operations research*, 57(6):1483–1495, 2009.

[18] Jose Blanchet and Karthyek Murthy. Quantifying distributional model risk via optimal transport. *Mathematics of Operations Research*, 44(2):565–600, 2019.

[19] Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. Fast differentiable sorting and ranking. In *International Conference on Machine Learning*, pages 950–959. PMLR, 2020.

[20] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[21] J Frédéric Bonnans and Alexander Shapiro. *Perturbation analysis of optimization problems*. Springer Science & Business Media, 2013.

[22] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization Methods for Large-Scale Machine Learning. *Siam Review*, 60(2):223–311, 2018.

[23] G. C. Calafiore and M. C. Campi. The scenario approach to robust control design. *IEEE Trans. Automat. Control*, 51:742–753, 2006.

[24] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. LEAF: A benchmark for federated settings. *CoRR*, abs/1812.01097, 2018.

[25] Abraham Charnes and William Wager Cooper. Chance-constrained programming. *Management science*, 6(1):73–79, 1959.

[26] Chunhui Chen and Olvi L Mangasarian. A class of smoothing functions for nonlinear and mixed complementarity problems. *Computational Optimization and Applications*, 5(2):97–138, 1996.

[27] Emilie Chouzenoux, Henri Gérard, and Jean-Christophe Pesquet. General risk measures for robust machine learning. *arXiv preprint arXiv:1904.11707*, 2019.

[28] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research*, 18(1):6070–6120, 2017.

[29] Sélim Chraibi, Ahmed Khaled, Dmitry Kovalev, Peter Richtárik, Adil Salim, and Martin Takác. Distributed fixed point methods with compressed iterates. *CoRR*, abs/1912.09925, 2019.

[30] Evgenii Chzhen, Christophe Denis, Mohamed Hebiri, Luca Oneto, and Massimiliano Pontil. Fair regression with wasserstein barycenters. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[31] Frank H Clarke. *Optimization and Nonsmooth Analysis*, volume 5. SIAM, 1990.

[32] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: extending MNIST to handwritten letters. In *2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017*, pages 2921–2926. IEEE, 2017. doi: 10.1109/IJCNN.2017.7966217.

[33] Imre Csiszár. Information-type measures of difference of probability distributions and indirect observation. *studia scientiarum Mathematicarum Hungarica*, 2:229–318, 1967.

[34] Sebastian Curi, Kfir Y Levy, Stefanie Jegelka, and Andreas Krause. Adaptive sampling for stochastic risk-averse learning. *NeurIPS*, 2020.

[35] George B. Dantzig. Discrete-variable extremum problems. *Oper. Res.*, 5(2):266–288, 1957. ISSN 0030-364X. doi: 10.1287/opre.5.2.266.

[36] Welington de Oliveira. Proximal bundle methods for nonsmooth dc programming. *Journal of Global Optimization*, 2019.

[37] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Distributionally Robust Federated Averaging. In *Neural Information Processing Systems*, 2020.

[38] D. Dentcheva. Optimisation models with probabilistic constraints. In A. Shapiro, D. Dentcheva, and A. Ruszczyński, editors, *Lectures on Stochastic Programming. Modeling and Theory*, volume 9 of *MPS-SIAM series on optimization*. 2009.

[39] D. Dentcheva and G. Martinez. Regularization methods for optimization problems with probabilistic constraints. *Mathematical Programming (series A)*, 138(1-2):223–251, 2013.

[40] D. Dentcheva, A. Prékopa, and A. Ruszczyński. Concavity and efficient points for discrete distributions in stochastic programming. *Mathematical Programming*, 89: 55–77, 2000.

[41] Olivier Devolder, François Glineur, and Yurii E. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Program.*, 146(1-2): 37–75, 2014.

[42] Aymeric Dieuleveut and Kumar Kshitij Patel. Communication Trade-offs for Local-SGD with Large Step Size. In *Advances in Neural Information Processing Systems*, pages 13579–13590, 2019.

[43] A. L. Diniz and R. Henrion. On probabilistic constraints with multivariate truncated gaussian and lognormal distributions. *Energy Systems*, 8(1):149–167, 2017.

[44] John C Duchi and Hongseok Namkoong. Variance-based Regularization with Convex Objectives. *Journal of Machine Learning Research*, 20(68):1–55, 2019.

[45] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[46] NCP Edirisinghe, EI Patterson, and Nasreddine Saadouli. Capacity planning model for a multipurpose water reservoir with target-priority operation. *Annals of Operations Research*, 100(1):273–303, 2000.

[47] Y.M. Ermoliev, T.Y. Ermolieva, G.J. Macdonald, and V.I. Norkin. Stochastic optimization of insurance portfolios for managing exposure to catastrophic risk. *Annals of Operations Research*, 99:207–225, 2000.

[48] Yanbo Fan, Siwei Lyu, Yiming Ying, and Bao-Gang Hu. Learning with average top-k loss. In *NIPS*, 2017.

[49] M. H. Farshbaf-Shaker, R. Henrion, and D. Hömberg. Properties of chance constraints in infinite dimensions with an application to pde constrained optimization. *Set Valued and Variational Analysis*, 26(4):821–841, 2018.

[50] Hans Föllmer and Alexander Schied. Convex measures of risk and trading constraints. *Finance and stochastics*, 6(4):429–447, 2002.

[51] Hans Föllmer and Alexander Schied. *Stochastic finance. An introduction in discrete time.* Berlin: de Gruyter, 2016. doi: 10.1515/9783110463453.

[52] Tomer Gafni, Nir Shlezinger, Kobi Cohen, Yonina C. Eldar, and H. Vincent Poor. Federated learning: A signal processing perspective. *CoRR*, abs/2103.17150, 2021.

[53] A. Geletu, A. Hoffmann, M. Klöppel, and P. Li. A tractable approximation of non-convex chance constrained optimization with non-gaussian uncertainties. *Engineering Optimization*, 47(4):495–520, 2015.

[54] Alec Go, Richa Bhayani, and Lei Huang. Twitter Sentiment Classification using Distant Supervision. *CS224N Project Report, Stanford*, page 2009, 2009.

[55] Mert Gürbüzbalaban, Andrzej Ruszczynski, and Landi Zhu. A stochastic subgradient method for distributionally robust non-convex learning. *CoRR*, abs/2006.04873, 2020.

[56] Farzin Haddadpour, Mohammad Mahdi Kamani, Mehrdad Mahdavi, and Viveck Cadambe. Local SGD with Periodic Averaging: Tighter Analysis and Adaptive Synchronization. In *Advances in Neural Information Processing Systems*, pages 11080–11092, 2019.

[57] Andrew Hard, Kanishka Rao, Rajiv Mathews, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *CoRR*, abs/1811.03604, 2018.

[58] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29:3315–3323, 2016.

[59] R. Henrion and C. Strugarek. Convexity of chance constraints with independent random variables. *Computational Optimization and Applications*, 41:263–276, 2008.

[60] R. Henrion and C. Strugarek. Convexity of chance constraints with dependent random variables: the use of copulae. In M. Bertocchi, G. Consigli, and M.A.H. Dempster, editors, *Stochastic Optimization Methods in Finance and Energy: New Financial Products and Energy Market Strategies*, International Series in Operations Research and Management Science, pages 427–439. Springer-Verlag New York, 2011.

[61] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms*. Springer Verlag, Heidelberg, 1993. Two volumes.

[62] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of convex analysis*. Springer Science & Business Media, 2012.

[63] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex analysis and minimization algorithms I: Fundamentals*, volume 305. Springer science & business media, 2013.

[64] Nam Ho-Nguyen and Stephen J. Wright. Adversarial classification via distributional robustness with wasserstein ambiguity. *CoRR*, abs/2005.13815, 2020.

[65] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural computation*, 9(8):1735–1780, 1997.

[66] L Jeff Hong, Yi Yang, and Liwei Zhang. Sequential convex approximations to joint chance constrained programs: A monte carlo approach. *Operations Research*, 59(3), 2011.

[67] L.J. Hong, Y. Yang, and L. Zhang. Sequential convex approximations to joint chance constrained programed: A monte carlo approach. *Operations Research*, 3 (59):617–630, 2011.

[68] David R Hunter and Kenneth Lange. Quantile Regression via an MM Algorithm. *Journal of Computational and Graphical Statistics*, 9(1):60–77, 2000.

[69] Garud Iyengar and Alfred Ka Chun Ma. Fast gradient descent method for mean-cvar optimization. *Annals of Operations Research*, 205(1):203–212, 2013.

[70] Peter Kairouz et al. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1-2):1–210, 2021. doi: 10.1561/2200000083.

[71] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-Aware Classifier with Prejudice Remover Regularizer. In *ECML PKDD*, pages 35–50. Springer, 2012.

[72] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

[73] S. Kataoka. A stochastic programming model. *Econometrica*, 31:181–196, 1963.

[74] Kenji Kawaguchi and Haihao Lu. Ordered sgd: A new stochastic optimization framework for empirical risk minimization. In *International Conference on Artificial Intelligence and Statistics*, pages 669–679. PMLR, 2020.

[75] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pages 4519–4529. PMLR, 2020.

[76] Jon Kleinberg. Inherent trade-offs in algorithmic fairness. In *Abstracts of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems*, pages 40–40, 2018.

[77] Will Knight. A self-driving Uber has killed a pedestrian in Arizona. *Ethical Tech*, March 2018.

[78] Roger Koenker and Gilbert Bassett Jr. Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50, 1978.

[79] A. Kogan and M. A. Lejeune. Threshold boolean form for joint probabilistic constraints with random technology matrix. *Mathematical Programming*, 147(1-2): 391–427, 2014.

[80] A. Kogan, M. A. Lejeune, and J. Luedtke. Erratum to: Threshold boolean form for joint probabilistic constraints with random technology matrix. *Mathematical Programming*, 155(1):617–620, 2016.

[81] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U. Stich. A unified theory of decentralized SGD with changing topology and local updates. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5381–5393. PMLR, 2020.

[82] Ole Kröger, Carleton Coffrin, Hassan Hijazi, and Harsha Nagarajan. Juniper: An open-source nonlinear branch-and-bound solver in julia. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Springer International Publishing, 2018. ISBN 978-3-319-93031-2.

[83] Daniel Kuhn, Peyman Mohajerin Esfahani, Viet Anh Nguyen, and Soroosh Shafieezadeh-Abadeh. Wasserstein Distributionally Robust Optimization: Theory and Applications in Machine Learning. In *Operations Research & Management Science in the Age of Analytics*, pages 130–166. INFORMS, 2019.

[84] Shigeo Kusuoka. On law invariant coherent risk measures. In *Advances in mathematical economics*, pages 83–95. Springer, 2001.

[85] Yassine Laguel, Jérôme Malick, and Zaid Harchaoui. First-order optimization for superquantile-based supervised learning. In *IEEE MLSP*, 2020.

[86] Yassine Laguel, Jérôme Malick, and Zaid Harchaoui. First-order optimization for superquantile-based supervised learning. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2020.

[87] Yassine Laguel, Krishna Pillutla, Jérôme Malick, and Zaid Harchaoui. Device heterogeneity in federated learning: A superquantile approach. *arXiv preprint arXiv:2002.11223*, 2020.

[88] Yassine Laguel, Jérôme Malick, and Zaid Harchaoui. Superquantile-based learning: a direct approach using gradient-based optimization. *Under review for the Journal of Signal Processing Systems*, 2021.

[89] Yassine Laguel, Krishna Pillutla, Jérôme Malick, and Zaid Harchaoui. A Superquantile Approach to Federated Learning with Heterogeneous Devices. In *IEEE CISS*, 2021.

[90] Yassine Laguel, Krishna Pillutla, Jérôme Malick, and Zaid Harchaoui. Superquantiles at work : Machine learning applications and efficient (sub)gradient computation. *Set-Valued and Variational Analysis*, 2021.

[91] Yassine Laguel, Wim Van Ackooij, and Jérôme Malick. Chance constrained problems: a bilevel convex optimization perspective. *Under review for Computational Optimization and Applications*, 2021.

[92] Yassine Laguel, Wim Van Ackooij, Jérôme Malick, and Guilherme Matiussi-Ramalho. On the convexity of level-sets of probability functions. *Journal of convex analysis*, 2021.

[93] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. Numba: A llvm-based python jit compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, LLVM '15, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450340052.

[94] Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. *NeurIPS*, 2020.

[95] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[96] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated Optimization in Heterogeneous Networks. In *Proceedings of Machine Learning and Systems*, pages 429–450. 2020.

[97] Tian Li, Maziar Sanjabi, and Virginia Smith. Fair Resource Allocation in Federated Learning. In *International Conference on Learning Representations*, 2020.

[98] Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. Tilted Empirical Risk Minimization. In *International Conference on Learning Representations*, 2021.

[99] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the Convergence of FedAvg on Non-IID Data. In *ICLR*, 2020.

[100] Daniel P Loucks, Jery R Stedinger, Douglas A Haith, et al. *Water resource systems planning and analysis*. Prentice-Hall., 1981.

[101] Andre Lucas and Pieter Klaassen. Extreme returns, downside risk, and optimal asset allocation. *Journal of Portfolio Management*, 25(1):71, 1998.

[102] James Luedtke. An integer programming and decomposition approach to general chance-constrained mathematical programs. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 271–284. Springer, 2010.

[103] James Luedtke. A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. *Mathematical Programming*, 146(1):219–244, 2014.

[104] James Luedtke, Shabbir Ahmed, and George L Nemhauser. An integer programming approach for linear programs with probabilistic constraints. *Mathematical programming*, 122(2):247–272, 2010.

[105] David G Luenberger and Yinyu Ye. *Linear and nonlinear programming*, volume 2. Springer, 1984.

[106] Juan Pablo Luna, Claudia Sagastizábal, and Mikhail Solodov. An approximation scheme for a class of risk-averse stochastic equilibrium problems. *Mathematical Programming*, 157(2):451–481, 2016.

[107] Luo Luo, Haishan Ye, Zhichao Huang, and Tong Zhang. Stochastic recursive gradient descent ascent for stochastic nonconvex-strongly-concave minimax problems. *Advances in Neural Information Processing Systems*, 33, 2020.

[108] Harry Markowitz. Portfolio selection. *Journal of Finance*, 7, 1952.

[109] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[110] Elena Medova. Chance-constrained stochastic programming forintegrated services network management. *Annals of Operations Research*, 81:213–230, 1998.

[111] Rachel Metz. Microsoft's neo-Nazi sexbot was a great lesson for makers of AI assistants. *Artificial Intelligence*, March 2018.

[112] Zakaria Mhammedi, Benjamin Guedj, and Robert C. Williamson. PAC-Bayesian Bound for the Conditional Value at Risk. In *NeurIPS*, 2020.

[113] Sofia I Miranda. Superquantile regression: theory, algorithms, and applications. Technical report, Naval postgraduate school Monterey ca, 2014.

[114] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic Federated Learning. In *International Conference on Machine Learning*, 2019.

[115] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal of Optimization*, 17(4):969–996, 2006.

[116] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[117] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[118] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2), 2017.

[119] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[120] Nilay Noyan and Gábor Rudolf. Kusuoka representations of coherent risk measures in general probability spaces. *Annals of Operations Research*, 229(1):591–605, 2015.

[121] Welington de Oliveira and Claudia Sagastizábal. Bundle methods in the xxist century: A bird's-eye view. *Pesquisa Operacional*, 34(3):647–670, 2014.

[122] F. Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2011.

[123] Alejandra Peña-Ordieres, James R Luedtke, and Andreas Wächter. Solving chance-constrained problems via a smooth sample-based nonlinear approximation. *SIAM Journal on Optimization*, 30(3):2221–2250, 2020.

[124] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.

[125] Constantin Philippenko and Aymeric Dieuleveut. Preserved central model for faster bidirectional compression in distributed settings. *CoRR*, abs/2102.12528, 2021.

[126] Krishna Pillutla, Yassine Laguel, Jérôme Malick, and Zaid Harchaoui. Chance constrained problems: a bilevel convex optimization perspective. *Under review for the IEEE Journal of Selected Topics in Signal Processing*, 2021.

[127] A. Prékopa. *Stochastic Programming*. Kluwer, Dordrecht, 1995. doi: 10.1007/978-94-017-3087-7.

[128] A. Prékopa. On the concavity of multivariate probability distributions functions. *Operations Research Letters*, 29:1–4, 2001.

[129] András Prékopa and Tamás Szántai. Flood control reservoir system design using stochastic programming. In *Mathematical programming in use*, pages 138–151. Springer, 1978.

[130] András Prékopa, Tamás Rapcsák, and István Zsuffa. Serially linked reservoir system design using stochastic programing. *Water Resources Research*, 14(4):672–678, 1978.

[131] Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

[132] F. A. Ramponi. Consistency of the scenario approach. *SIAM Journal on Optimization*, 28(1):135–162, 2018.

[133] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5389–5400. PMLR, 2019.

[134] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive Federated Optimization. In *International Conference on Learning Representations*, 2021.

[135] R Terry Rockafellar, Johannes O Royset, and Sofia I Miranda. Superquantile regression with applications to buffered reliability, uncertainty quantification, and conditional value-at-risk. *European Journal of Operational Research*, 234(1):140–154, 2014.

[136] R Tyrrell Rockafellar and Johannes O Royset. Superquantiles and their applications to risk, random variables, and regression. In *Theory Driven by Influential Applications*, pages 151–167. INFORMS, 2013.

[137] R Tyrrell Rockafellar and Johannes O Royset. Random variables, monotone relations, and convex analysis. *Mathematical Programming*, 148(1-2), 2014.

[138] R Tyrrell Rockafellar and Stan Uryasev. The Fundamental Risk Quadrangle in Risk Management, Optimization and Statistical Estimation. *Surveys in Operations Research and Management Science*, 18(1-2):33–53, 2013.

[139] R Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of banking & finance*, 26(7):1443–1471, 2002.

[140] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

[141] R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

[142] R.T. Rockafellar and R.J.-B. Wets. *Variational Analysis*. Springer Verlag, Heidelberg, 1998.

[143] J.O. Royset and E. Polak. Extensions of stochastic optimization results to problems with system failure probability functions. *Journal of Optimization Theory and Applications*, 133(1):1–18, 2007.

[144] A Ruszczynski and A Shapiro. Stochastic programming (handbooks in operations research and management science), 2003.

[145] Andrzej Ruszczyński and Alexander Shapiro. Optimization of convex risk functions. *Mathematics of operations research*, 31(3):433–452, 2006.

[146] Soroosh Shafieezadeh-Abadeh, Daniel Kuhn, and Peyman Mohajerin Esfahani. Regularization via mass transportation. *Journal of Machine Learning Research*, 20 (103):1–68, 2019.

[147] Alexander Shapiro. On kusuoka representation of law invariant risk measures. *Mathematics of Operations Research.*, 38:142–152, 2013.

[148] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczynski. *Lectures on Stochastic Programming: Modeling and Theory, Second Edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2014. ISBN 1611973422, 9781611973426.

[149] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczynski. *Lectures on stochastic programming: modeling and theory*. SIAM, 2021.

[150] Samarth Sinha, Zhengli Zhao, Anirudh Goyal, Colin Raffel, and Augustus Odena. Top-k training of gans: Improving GAN performance by throwing away bad samples. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[151] Jinhyun So, Basak Güler, and Amir Salman Avestimehr. Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning. *IEEE J. Sel. Areas Inf. Theory*, 2(1):479–489, 2021. doi: 10.1109/JSAIT.2021.3054610.

[152] Tasuku Soma and Yuichi Yoshida. Statistical learning with conditional value at risk. *CoRR*, abs/2002.05826, 2020.

[153] Sebastian U. Stich. Local SGD Converges Fast and Communicates Little. In *International Conference on Learning Representations*, 2019.

[154] Stefan Straszewicz. Über exponierte punkte abgeschlossener punktmengen. *Fundamenta Mathematicae*, 24(1):139–143, 1935.

[155] Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Policy gradient for coherent risk measures. In *Advances in Neural Information Processing Systems*, pages 1468–1476, 2015.

[156] S. Uryas'ev. Derivatives of probability functions and some applications. *Annals of Operations Research*, 56:287–311, 1995.

[157] W. van Ackooij. Eventual convexity of chance constrained feasible sets. *Optimization (A Journal of Mathematical Programming and Operations Research)*, 64(5):1263–1284, 2015. doi: 10.1080/02331934.2013.855211.

[158] W. van Ackooij. A discussion of probability functions and constraints from a variational perspective. *Set-Valued and Variational Analysis (online)*, 28:585–609, 2020.

[159] W. van Ackooij and W. de Oliveira. Convexity and optimization with copulæ structured probabilistic constraints. *Optimization: A Journal of Mathematical Programming and Operations Research*, 65(7):1349–1376, 2016.

[160] W. van Ackooij and R. Henrion. (Sub-) Gradient formulae for probability functions of random inequality systems under Gaussian distribution. *SIAM Journal on Uncertainty Quantification*, 5(1):63–87, 2017.

[161] W. van Ackooij and J. Malick. Eventual convexity of probability constraints with elliptical distributions. *Mathematical Programming*, 175(1):1–27, 2019. doi: 10.1007/s10107-018-1230-3.

[162] W. van Ackooij, R. Henrion, A. Möller, and R. Zorgati. Joint chance constrained programming for hydro reservoir management. *Optimization and Engineering*, 15, 2014.

[163] W. van Ackooij, V. Berge, W. de Oliveira, and C. Sagastizábal. Probabilistic optimization via approximate p-efficient points and bundle methods. *Computers & Operations Research*, 77:177–193, 2017.

[164] Wim Van Ackooij. *Chance constrained programming: with applications in Energy Management*. PhD thesis, Ecole Centrale Paris, 2013.

[165] Wim van Ackooij, René Henrion, Andris Möller, and Riadh Zorgati. Joint chance constrained programming for hydro reservoir management. *Optimization and Engineering*, 15(2):509–531, 2014.

[166] Wim van Ackooij, Sophie Demassey, Paul Javal, Hugo Morais, Welington de Oliveira, and Bhargav Swaminathan. A bundle method for nonsmooth dc programming with application to chance-constrained problems. *Computational Optimization and Applications*, 78(2):451–490, 2021.

[167] Lieven Vandenberghe. The cvxopt linear and quadratic cone program solvers. *Online: http://cvxopt. org/documentation/coneprog. pdf*, 2010.

[168] Thijs Vogels, Sai Praneeth Karinireddy, and Martin Jaggi. Powersgd: Practical low-rank gradient compression for distributed optimization. *Advances In Neural Information Processing Systems 32 (Nips 2019)*, 32(CONF), 2019.

[169] Rudin Walter. Real and complex analysis. 1987.

[170] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization. In *Neural Information Processing Systems*, 2020.

[171] Jianyu Wang et al. A field guide to federated optimization. *CoRR*, abs/2107.06917, 2021.

[172] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K. Leung, Christian Makaya, Ting He, and Kevin Chan. Adaptive federated learning in resource constrained edge computing systems. *IEEE J. Sel. Areas Commun.*, 37(6):1205–1221, 2019. doi: 10.1109/JSAC.2019.2904348.

[173] Robert Williamson and Aditya Menon. Fairness risk measures. In *International Conference on Machine Learning*, pages 6786–6797. PMLR, 2019.

[174] Robert C. Williamson and Aditya Krishna Menon. Fairness Risk Measures. In *International Conference on Machine Learning*, 2019.

[175] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. Applied federated learning: Improving google keyboard query suggestions. *CoRR*, abs/1812.02903, 2018.

[176] Jane J Ye, Daoli Zhu, and Qiji Jim Zhu. Exact penalization and necessary optimality conditions for generalized bilevel programming problems. *SIAM Journal on optimization*, 7(2):481–507, 1997.

[177] Constantin Zalinescu. *Convex analysis in general vector spaces*. World scientific, 2002.

[178] Riadh Zorgati, Wim van Ackooij, and Romain Apparigliato. Supply shortage hedging: estimating the electrical power margin for optimizing financial and physical assets with chance-constrained programming. *IEEE Transactions on Power Systems*, 24(2):533–540, 2009.

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

COLOPHON

This manuscript was typeset with LaTeX 2$_\varepsilon$ using Hermann Zapf's Palatino type face (the actual Type 1 PostScript fonts used were URW Palladio L and FPL). The monospaced text (hyperlinks, etc.) was typeset in *Bera Mono*, originally developed by Bitstream, Inc. as "Bitstream Vera" (with Type 1 PostScript fonts by Malte Rosenau and Ulrich Dirr).

The typographic style of this dissertation was inspired by the authoritative genius of Bringhurst's *Elements of Typographic Style*, ported to LaTeX by André Miede, the original designer of the `classicthesis` template. Any unsightly deviations from these works should be attributed solely to the author's (not always successful) efforts to conform to the awkward A4 paper size.

Risk-averse Optimization: Models, Algorithms, and Applications in Machine Learning

Yassine Laguel