Learning and Optimization in the Face of Data Perturbations

by

Matthew James Staib

B.S., Stanford University (2015)
M.S., Stanford University (2015)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2020

Author .....
Department of Electrical Engineering and Computer Science
May 1, 2020

Certified by .....
Stefanie Jegelka
Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Learning and Optimization in the Face of Data Perturbations

by

Matthew James Staib

Submitted to the Department of Electrical Engineering and Computer Science
on May 1, 2020, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Many problems in the machine learning pipeline boil down to maximizing the expectation of a function over a distribution. This is the classic problem of stochastic optimization. There are two key challenges in solving such stochastic optimization problems: 1) the function is often non-convex, making optimization difficult; 2) the distribution is not known exactly, but may be perturbed adversarially or is otherwise obscured. Each issue is individually so challenging to warrant a substantial accompanying body of work addressing it, but addressing them simultaneously remains difficult.

This thesis addresses problems at the intersection of non-convexity and data perturbations. We study the intersection of the two issues along two dual lines of inquiry: first, we build perturbation-aware algorithms with guarantees for non-convex problems; second, we seek to understand how data perturbations can be leveraged to enhance non-convex optimization algorithms. Along the way, we will study new types of data perturbations and seek to understand their connection to generalization.

Thesis Supervisor: Stefanie Jegelka
Title: Associate Professor of Electrical Engineering and Computer Science

Acknowledgments

A Ph.D. is a long journey, made easier by kind mentors, made more fun by great friends, and supported by loving family. I am indebted to all three.

I will start by thanking my advisor, Stefanie Jegelka. Working with Stefanie over the past few years has been simply wonderful. She has taught me a great deal about both how to do quality research and also how to communicate it. Stefanie has given me countless opportunities, constantly sending me to speak and present and learn, and encouraging me to explore. She has granted me much intellectual freedom, but has also managed to counterbalance that with advice and insight. Working with her has truly been a joy, and I am grateful to have had the opportunity.

Next, I want to thank my other committee members, John Tsitsiklis and Andreas Krause. While John and Andreas came onboard fairly late in the process, nevertheless they provided insightful questions and suggestions which helped shape and contextualize the thesis.

During my Ph.D. I am fortunate to have worked with several great collaborators on projects that are in this thesis or closely related to it. Suvrit Sra and Justin Solomon have been wonderful senior collaborators and mentors. Bryan Wilder and Sebastian Claici have been great student collaborators and also great friends. Thanks also to Sashank Reddi, Satyen Kale, and Sanjiv Kumar for their guidance and for hosting me for a fruitful summer at Google Research, which led to Part III of the thesis.

I also want to thank other mentors who have helped me become the researcher I am today. Thanks to James Kirpes and the rest of the West High math folk for nurturing my love for math in my formative years. At Stanford, thanks to Bernard Widrow and Brad Osgood for helping guide a bushy-tailed undergraduate interested in applying math, and thanks to Jindong Cai and Sheila Melvin for helping that undergraduate stay well-rounded. Thanks to Thomas Moscibroda and Nic Lane for hosting me at MSR Asia, for what was one of my first real doses of academic research. And, many years later, thanks to Div, Jin, and the team at Two Sigma for teaching

me a great deal about applied research (and for having me back full-time!).

I have been fortunate to have had many friends at MIT. First, thanks to LOGSS (Stefanie and Suvrit's research group) for friendships and fun discussions, and thanks more broadly to the MIT ML group. Thanks to the "LIDS++" group that has largely stuck together since the visit days: Joshua, Aidan, Dennis, Deniz, Zhi, and Dogyoon. Thanks to my "Stanford fam" at MIT, among them: Rio, Alfred, Linyi, Marie, Leilani, and Anna. The great musicians of MITSO helped keep me sane. A few other MIT friends are not as easily categorized: Yen-Ling, Candace, the Muscos, Ed Chien, and Sam Park.

Other friends in Cambridge provided support and kept me grounded in the world outside campus: Rio and Avery, Josh and Cat, Ruodi and Vincent, Seb and Cat, Zi and Leon, Hunter and Riley, Ben and Dandan, Sarah and Tom, Brian and Karen (at risk of redundancy, a few of you appear in multiple lists!). And friends from far away served a similar role despite the distance: Jesse and Dawn, Tyler, Gabe, Arun, Garrett, Boris, and Anna.

Next I thank my family. Thanks to Ginat, Steve, Michael, and David, for treating me like one of their own. Thanks to my in-laws for always checking in and wishing 注意安全、身体健康! Thanks to my extended family for constant encouragement and dealing with my too-long explanations of too-mathy topics. Thanks to Catherine, Becca, Cassie, and Anna, for keeping childhood zany and remaining good friends as we ~~age~~ gain life experience. And, of course, thanks mom and dad for always supporting me, encouraging me, nurturing my curiosity and sense of self-confidence, and giving me every opportunity to explore.

Finally, thank you to my wife 思孜, who has been my constant source of strength and inspiration. Without her radiance, intellect, insight, and open mind, this thesis would have suffered, and my worldview would not be as broad. And without her support and love, it would have been much harder to finish. 感谢你傻鹅。

Contents

Acknowledgements **5**

List of Figures **13**

1 Introduction **15**

1.1 Motivation 15

1.2 Thesis outline 17

1.3 Additional related publications 18

1.4 Notation 19

I Understanding the link between DRO and generalization **20**

2 Background on generalization, data perturbations, and DRO **21**

2.1 Data perturbations and generalization 22

2.1.1 Random perturbations 22

2.1.2 Adversarial perturbations 22

2.2 Distributionally Robust Optimization (DRO) 23

2.2.1 Far-reaching relevance to machine learning 25

3 DRO, MMD, kernels, and generalization **27**

3.1 Introduction 27

3.2 Background and related work 29

3.3 Generalization bounds via MMD DRO 31

3.3.1 Bounding the DRO adversary's problem . . . 32
3.4 Connections to kernel ridge regression . . . 34
3.4.1 Bounding norms of products . . . 35
3.4.2 Implications: kernel ridge regression . . . 36
3.4.3 Algorithmic implications . . . 38
3.5 Approximation and connections to variance regularization . . . 39
3.6 Experiments . . . 41
3.6.1 Alternate regularizer . . . 41
3.6.2 Conjecture: generalizing beyond Gaussian kernels . . . 42
3.7 Discussion and future work . . . 46

II Algorithms for distributionally robust subset selection 47

4 Submodularity background 49
4.1 Submodular set functions . . . 49
4.1.1 Definitions . . . 50
4.1.2 Optimization . . . 50
4.2 General submodular functions . . . 52
4.2.1 Definitions: submodular functions and DR functions . . . 52
4.2.2 Optimization . . . 53
4.3 Submodular DRO . . . 54
4.3.1 Robust and risk-averse submodular optimization . . . 55
4.3.2 Submodular optimization with errors . . . 56

5 Distributionally robust submodular maximization 57
5.1 Introduction . . . 57
5.1.1 Related work . . . 59
5.2 Stochastic submodular functions and distributional robustness . . . 61
5.2.1 Stochastic submodular functions . . . 61
5.2.2 Optimization and empirical approximation . . . 62
5.2.3 Variance regularization via distributionally robust optimization 64

5.3 Exact algorithm for χ^2-DRO . . . 66
5.4 Algorithmic approach . . . 70
5.5 Experiments . . . 75
5.5.1 Facility Location . . . 76
5.5.2 Influence maximization . . . 77
5.5.3 Rounding . . . 78
5.6 Discussion and future work . . . 79

6 Robust Budget Allocation **81**
6.1 Introduction . . . 81
6.1.1 Background and related work . . . 84
6.2 Robust and stochastic Budget Allocation . . . 85
6.2.1 Stochastic optimization . . . 85
6.2.2 Robust optimization . . . 86
6.3 Robust Budget Allocation: main ideas . . . 88
6.4 Constrained continuous submodular function minimization . . . 90
6.4.1 Forming an equivalent convex problem . . . 91
6.4.2 Bounding solution quality for the constrained problem . . . 95
6.5 Simple examples where our approach is optimal . . . 101
6.5.1 Separable problems . . . 101
6.5.2 Non-separable quadratics and SDP relaxations . . . 105
6.5.3 Evaluation of suboptimality bounds . . . 106
6.6 Robust Budget Allocation experiments . . . 107
6.6.1 Synthetic . . . 108
6.6.2 Yahoo! data . . . 110
6.6.3 Comparison to first-order methods . . . 111
6.7 Discussion and future work . . . 112

III The reverse: leveraging perturbations for better non-

convex optimization algorithms **114**

7 Escaping saddle points with Adaptive Gradient Methods and perturbations **115**

7.1 Introduction . . . 115
7.1.1 Adaptive gradient methods (AGMs) . . . 116
7.1.2 Related work . . . 119
7.2 Notation and definitions . . . 120
7.3 The RMSProp preconditioner . . . 121
7.3.1 What is the purpose of the preconditioner? . . . 122
7.3.2 Reddi et al. (2018b) counterexample resolution . . . 123
7.4 Main results: gluing estimation and optimization . . . 124
7.4.1 Estimating from moving sequences . . . 124
7.4.2 Convergence results . . . 127
7.5 Discussion . . . 132
7.5.1 How to set the regularization parameter ε . . . 132
7.5.2 Comparison to SGD . . . 133
7.5.3 Alternative preconditioners . . . 134
7.5.4 Tuning the EMA parameter β . . . 134
7.6 Experiments . . . 135
7.7 Further discussion and future work . . . 136

IV Conclusion **139**

8 Conclusion **141**

8.1 High-level summary . . . 141
8.2 Future directions . . . 142
8.2.1 Perturbations and generalization . . . 142
8.2.2 Perturbation-aware optimization . . . 142
8.2.3 Perturbations for optimization . . . 143

V Bibliography and Appendix **144**

Bibliography **145**

A DRO, MMD, kernels, and generalization **165**

A.1 Proofs of main structural results 165

A.2 Gaussian kernel bounds 167

A.2.1 Trace inequality 172

A.2.2 Extensions of Proposition 3.4.1 173

A.3 Proofs for Section 3.5 173

B Distributionally robust submodular maximization **177**

B.1 Tail Bound 177

B.2 Equivalence of Variance Regularization and Distributionally Robust Optimization 178

B.3 Exact Linear Oracle 180

B.3.1 Unique solutions 185

B.3.2 Lipschitz gradient 186

B.4 Convergence analysis for MFW 188

B.5 Rounding to a distribution over subsets 191

C Robust Budget Allocation **193**

C.1 Worst-Case Approximation Ratio versus True Worst-Case 193

C.2 DR-submodularity and $L^\natural$-convexity 194

C.3 Constrained Continuous Submodular Function Minimization 195

C.3.1 Solving the Optimization Problem 195

C.3.2 Runtime 197

D Escaping saddle points with Adaptive Gradient Methods and perturbations **199**

D.1 More Insights from Idealized Adaptive Methods (IAM) 199

D.2 Algorithm Details 200

D.3 Curvature and noise constants for different preconditioners 201
D.3.1 Constants for identity preconditioner 202
D.3.2 Constants for full matrix IAM 203
D.3.3 Constants for diagonal IAM 205
D.4 Convergence results for the diagonal case 207
D.5 Main Proof 208
D.5.1 Definitions 209
D.5.2 High level picture 210
D.5.3 Amortized increase due to large stepsize iterations 212
D.5.4 Bound on possible increase when $\mathcal{E}_t^c$ occurs 213
D.5.5 Bound on decrease (progress) when $\mathcal{E}_t$ occurs 214
D.5.6 Auxiliary lemmas 228
D.5.7 Descent lemmas 229
D.6 Convergence to First-Order Stationary Points 231
D.6.1 Generic Preconditioners: Proof of Theorem 7.4.2 231
D.6.2 Generic Preconditioners with Errors: Proof of Theorem 7.4.3 . 233
D.7 Online Matrix Estimation 234
D.8 Converting Noise Estimates into Preconditioner Estimates 238

List of Figures

1-1 Block diagram showing topics addressed by the the thesis and how they relate to each other. 17

3-1 Pictorial representation of the DRO Generalization Principle 3.1.1. . . 28

3-2 Comparison of standard kernel ridge regression regularizer $\|h\|_\sigma^2$ versus our proposed regularizer $\|h^2\|_{\sigma/\sqrt{2}}$. 41

3-3 Estimates of the ratio σ'/σ needed so that $\|k_\sigma(0,\cdot)^2\|_{\sigma'} \leq \|k_\sigma(0,\cdot)\|_\sigma^2$. 43

3-4 Study of $\|h^2\|_{\sigma'}$ versus $\|h\|_\sigma^2$ for functions $h = \cos\theta\, k_\sigma(0,\cdot) + \sin\theta\, k_\sigma(1,\cdot)$. 43

3-5 Distribution of ratio $\|h^2\|_{\sigma'}/\|h\|_\sigma^2$ for randomly sampled functions h, for Laplace kernels with different bandwidths σ. 44

3-6 Distribution of ratio $\|h^2\|_{\sigma'}/\|h\|_\sigma^2$ for randomly sampled functions h, for Matérn kernels with different bandwidths σ. 45

5-1 Algorithm comparison and generalization performance on last.fm dataset. 77

5-2 Influence maximization on political blogs dataset. 78

6-1 Bipartite graph demonstrating the setup of (Robust) Budget Allocation. 82

6-2 Comparison of submodular optimization solution versus optimal SDP solution for non-convex quadratic programs. 107

6-3 Empirical study of when Theorem 6.4.2 can certify optimality of our constrained submodular minimization solution. 109

6-4 Comparison of robust versus non-robust solutions for Budget Allocation on synthetic data. 110

6-5 Convergence properties of our algorithm on a Robust Budget Allocation problem with real data. 111
6-6 Convergence properties of Frank-Wolfe (FW), versus the optimal value attained with our scheme (SFM). 111

7-1 SGD vs RMSProp performance escaping a saddle point with poorly conditioned gradient noise. 137
7-2 Performance on MNIST logistic regression of RMSProp with different choices of β and decreasing stepsize. 138

Chapter 1

Introduction

1.1 Motivation

Machine learning systems are increasingly prevalent, as are decision systems that make use of learned models of the world. Though the types of models, decisions, and application areas are manifold, one fundamental tool is key to all of them: stochastic optimization, i.e. the problem $\min_w \mathbb{E}_{z\sim\mathbb{P}}[f(w,z)]$. For example, in supervised learning, $\mathbb{P}$ is a distribution of pairs $z = (x, y)$ of datapoints x together with labels y. Our goal is to choose a model w that we can use to predict y from x, e.g. $y \approx w^T x$. For a single sample pair $z = (x, y)$, the function $f(w, z)$ might capture the prediction error when w is used to predict y from x. It is then natural to choose a model w with minimum prediction error $\mathbb{E}_{z\sim\mathbb{P}}[f(w,z)]$ on the data generating distribution $\mathbb{P}$.

Beyond statistical learning, stochastic optimization problems capture many decision problems we may need to make downstream. Indeed, the classic work of von Neumann and Morgenstern (1944) shows that, under certain assumptions, any rational actor making decisions in a stochastic environment can be thought of as optimizing $\mathbb{E}_{z\sim\mathbb{P}}[f(w,z)]$ for an appropriately chosen utility function f.

There are two key challenges to solving stochastic optimization problems. First, $\mathbb{P}$ is often not known exactly; instead we might only have samples from $\mathbb{P}$, or noisy parameter estimates for $\mathbb{P}$ if $\mathbb{P}$ is a parametric model. In statistical learning, one typically replaces $\mathbb{P}$ with the empirical distribution $\hat{\mathbb{P}}_n$, and minimizes $\mathbb{E}_{z\sim\hat{\mathbb{P}}_n}[f(w,z)] =$

$\frac{1}{n}\sum_{i=1}^{n} f(w, z_i)$. This is called empirical risk minimization (ERM). While ERM provides a good starting point, other approaches can often do better, especially in more challenging settings. For example, in extreme cases, $\mathbb{P}$ might be perturbed adversarially, and more specialized algorithms can yield substantial performance improvements (see e.g. (Madry et al., 2018)). A major challenge in machine learning, generalization, is to achieve good performance on $\mathbb{P}$ despite these difficulties.

Second, f is often non-convex, making optimization challenging even if $\mathbb{P}$ were known exactly. This thesis addresses both these challenges from multiple facets.

One approach to improving generalization (despite the effect of perturbations on $\mathbb{P}$) is to explicitly encode these perturbations via an uncertainty set $\mathcal{U}$ that captures how $\mathbb{P}$ might be altered. Then we can solve the distributionally robust optimization (DRO) problem $\min_w \max_{\mathbb{Q}\in\mathcal{U}} \mathbb{E}_{\mathbb{Q}}[f(w)]$. We try to choose a point w that works as well as possible in spite of an adversary that perturbs $\mathbb{P}$ within $\mathcal{U}$. This approach has seen success in the operations research literature and is a promising technique for machine learning. The first part of this thesis develops a new variant of DRO with particularly strong connections to generalization in machine learning. However, DRO problems can be challenging for some uncertainty sets $\mathcal{U}$, especially when f is non-convex. The second part of this thesis develops new algorithms for solving DRO problems in the special case when f is submodular. In the final part of this thesis, we investigate a different type of noise in $\mathbb{P}$ due to subsampling a dataset, as is common in practice when e.g. the dataset cannot fit in memory. This is the problem of stochastic non-convex optimization. We show how the noise due to subsampling, normally a hindrance to optimization, can be reshaped to yield better performance when f is non-convex.

There are three fundamental questions we ask, each of which we study in one of the three parts in the thesis:

1. Precisely how are DRO and generalization linked? And how can this bond be strengthened via e.g. new DRO problems?

2. When and how is it possible to solve DRO problems when f is not convex?

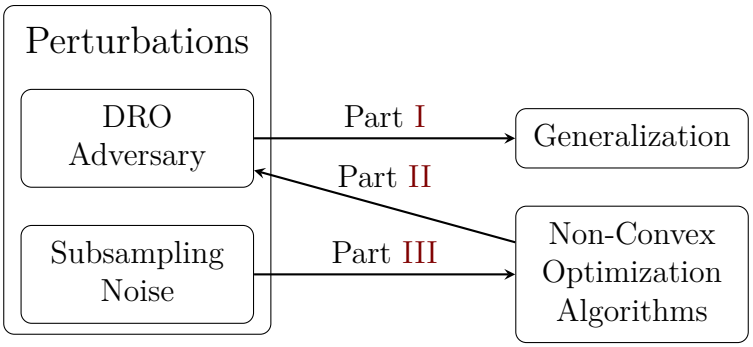


Figure 1-1: Block diagram showing how this thesis addresses topics at the interfaces of three main areas of study: data perturbations, generalization, and non-convex optimization algorithms.

3. How can data perturbations be leveraged to create better non-convex optimization algorithms?

1.2 Thesis outline

These questions and their relationship with data perturbations and non-convex optimization can be understood via Figure 1-1. The key players are generalization, perturbations, and non-convex optimization algorithms, and each part of the thesis studies the relationship between two of these.

Part I asks how perturbations (via DRO) relate to generalization. We start by giving background in Chapter 2 on the interplay of generalization and perturbations. We specifically focus on DRO, and introduce the types of DRO problems considered in the literature thus far. Then in Chapter 3, which is based on (Staib and Jegelka, 2019a), we develop a new kind of DRO problem with strong ties to generalization.

Part II asks how to construct optimization algorithms for non-convex DRO problems, specifically, submodular DRO problems. In Chapter 4 we provide background on submodular optimization. We highlight positive results in submodular maximization and minimization, and also discuss potential difficulties, such as hardness results for robust and risk-averse submodular optimization. In Chapter 5, based on (Staib et al., 2019b), we demonstrate how to solve a certain class of submodular DRO problem, with broad applications to problems such as influence maximization and facility

location. As a byproduct, we tighten results relating certain DRO problems and variance regularization. Then in Chapter 6, based on (Staib and Jegelka, 2019b), we focus on a specific combinatorial problem called Budget Allocation. We introduce a robust version of the problem, and develop new submodular optimization techniques in order to solve it.

Finally, Part III asks how perturbations can help non-convex optimization. In Chapter 7, which is based on (Staib et al., 2019a), we study a class of optimization algorithms called adaptive gradient methods (AGMs). These are poorly understood in the non-convex setting. We observe that AGMs reshape subsampling noise in a way that is useful for non-convex optimization. This insight, coupled with other new observations, allows us to give the first (non-convex) second-order convergence result for any AGM, and understand when AGMs work well compared to alternative algorithms.

1.3 Additional related publications

Here we list other work completed during the author's program. These also address other aspects of robustness, uncertainty, geometry, and optimization. While these papers are strongly related to the thesis, they were left out to maintain a clearer story.

- Matthew Staib, Sebastian Claici, Justin M Solomon, and Stefanie Jegelka. Parallel streaming Wasserstein barycenters. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2644–2655. Curran Associates, Inc., 2017.

- Matthew Staib and Stefanie Jegelka. Distributionally robust deep learning as a generalization of adversarial training. In *NIPS Machine Learning and Computer Security Workshop*, 2017a.

- Matthew Staib and Stefanie Jegelka. Wasserstein k-means++ for cloud regime histogram clustering. In *Proceedings of the Seventh International Workshop on Climate Informatics: CI 2017*, 2017c.

1.4 Notation

As the thesis includes work in several areas with their own notational conventions, it is difficult to wholly unify the notation; we have made efforts to do so where possible. We refer to a model or function we want to learn by h; this is most relevant in Part I. We will refer to generic objective functions by f. Their arguments are context-dependent: in learning problems, considered in the background sections and in Part III, the arguments are weights w to be learned. In Part I we focus specifically on problems where the objective is the loss ℓ_h incurred by our model h. Elsewhere, particularly in Part II, the arguments are more varied, and may be subsets S or generalizations thereof x.

We refer to small probabilities by δ. The radius of an uncertainty set we refer to by ε, e.g. $\{x : \|x\| \leq \varepsilon\}$, except in Part III, where ε is an algorithm parameter. When presenting optimization results, τ is used to denote convergence tolerances. By $\mathbf{1}$ and $\mathbf{0}$ we mean the all-ones and all-zeros vectors, respectively. Throughout the thesis, δ_x is a point mass distribution at x. We reserve $\mathbb{P}$ and $\mathbb{Q}$ to refer to distributions, and by $\hat{\mathbb{P}}_n$ we mean an empirical distribution $\frac{1}{n}\sum_{i=1}^n \delta_{x_i}$ consisting of x_i that are sampled iid from $\mathbb{P}$.

Other notation that is more specific to individual chapters is introduced as needed.

Part I

Understanding the link between DRO and generalization

Chapter 2

Background on generalization, data perturbations, and DRO

One of the key challenges in machine learning is generalization. In the traditional statistical setup, a model trained only on samples from a distribution $\mathbb{P}$ generalizes if it performs similarly well on the samples and on $\mathbb{P}$ itself. Formally, we assume there is an underlying *population* distribution $\mathbb{P}$ of interest: in machine learning, for example, $\mathbb{P}$ could be a distribution of all possible images of objects, together with labels for the objects. Our goal is to learn a model h that performs well on this entire distribution, e.g. h can accurately predict the correct label for each object. However, we cannot possibly have access to *every* possible image. Instead, we have a finite dataset, called the training set, composed of n samples $z_1, \ldots, z_n$ that are assumed to be drawn i.i.d. from $\mathbb{P}$. These samples form the *empirical* distribution $\hat{\mathbb{P}}_n$. In lieu of seeking good performance on $\mathbb{P}$, we seek a model h that performs well on the available dataset $\hat{\mathbb{P}}_n$. We say the learned model h *statistically generalizes* if its performance on the available data $\hat{\mathbb{P}}_n$ is similar to its performance on the population distribution $\mathbb{P}$.

The above setup is convenient for statistical analysis, but is not always realistic. Future unseen examples need not come from the same distribution as our training set. Instead, when we deploy our model and evaluate it on new examples, any number of changes may have occured. Perhaps the new example images have different lighting, or are rotated differently. Maybe the relative frequencies of different types of objects

have changed. The new images could be noisier, or even adversarially perturbed. We say *generalization* in the broadest sense means our learned model h can perform well on new unseen examples, regardless of whether those samples come from the same distribution $\mathbb{P}$ as our test set.

We will see that data perturbations and distributionally robust optimization (DRO) give us tools to understand and improve both *statistical generalization* as well as *generalization* in the broadest sense.

2.1 Data perturbations and generalization

2.1.1 Random perturbations

Practitioners often have some prior idea of what kinds of variations occur in the population distribution $\mathbb{P}$ but may not necessarily arise in the training set. One especially clear case of this is in image classification. In curated training sets of images of objects, typically the object is centered in the image and right-side up. But while most pictures of e.g. cars have the car centered and oriented normally, a photographer could conceivably choose to take a picture off-center and at a strange angle. Such pictures, while potentially rare, may occur in the population distribution $\mathbb{P}$ even if they do not occur in the training set.

A simple way to encode such knowledge about the population distribution is to take examples from the training set, perturb them, and add them to the training set. For example, for each image in the training set, add randomly flipped, rotated and cropped versions of that image. This practice is called data augmentation, and it is extremely popular and successful in improving generalization performance.

2.1.2 Adversarial perturbations

Random perturbations are not the only type of perturbations that have found use in machine learning. Substantial recent work considers augmenting the training set with *adversarially* perturbed examples (Goodfellow et al., 2015; Szegedy et al., 2014;

Madry et al., 2018). For example, we may seek a model w that performs well even when each training example z is perturbed adversarially in a ball $B_\varepsilon(z)$ of radius ε around z:

$$\inf_w \mathop{\mathbb{E}}_{z\sim\hat{\mathbb{P}}_n} \left[\sup_{\tilde{z}\in B_\varepsilon(z)} f(w,\tilde{z}) \right]. \tag{2.1}$$

Training a model against adversarially perturbed examples is known as *adversarial training*. Adversarial training is actually also a specific example of *robust optimization* (Ben-Tal et al., 2009; Bertsimas et al., 2011), in which we want to make a decision that performs well despite errors in the objective function and even the constraints, e.g.:

$$\inf_w \sup_{u,A,b\in\mathcal{U}} \{f(w;u) \text{ s.t. } Aw \le b\}. \tag{2.2}$$

Though originally motivated due to concerns about security and robustness of learned models, adversarial training can also improve models in qualitative ways that are not captured by test performance (Tsipras et al., 2019, Section 3).

2.2 Distributionally Robust Optimization (DRO)

The previous section focused on the effect of perturbing individual examples from the training set, and how such perturbations can improve generalization. More generally, we could jointly perturb the entire training set, or equivalently, perturb the empirical distribution $\hat{\mathbb{P}}_n$ within some uncertainty set $\mathcal{U}$. This is an instance of distributionally robust optimization (DRO) (Goh and Sim, 2010; Bertsimas et al., 2018). DRO, introduced by Scarf (1958), asks to not only perform well on a fixed problem instance (parameterized by a distribution), but simultaneously for a range of problems, each determined by a distribution in an *uncertainty set* $\mathcal{U}$. Concretely, we want to make a decision w that solves

$$\text{(DRO)} \qquad \inf_w \sup_{\mathbb{Q}\in\mathcal{U}} \mathbb{E}_{z\sim\mathbb{Q}}[f(w,z)]. \tag{2.3}$$

The uncertainty set plays a key role: it implicitly defines the induced notion of robustness.

As an example, Scarf (1958) is concerned with an inventory management problem that depends on the distribution of demand. Here, the decision maker knows the mean and variance of the demand distribution – perhaps she has accurate measurements of these quantitites – but wishes to avoid making further assumptions. Scarf sets up a DRO problem where the uncertainty set $\mathcal{U}$ consists of all distributions with that mean and variance, and derives a closed form solution.

Many DRO problems studied by the operations research community are similarly defined by moment constraints. More recently, Delage and Ye (2010) have shown for a wide class of problems how to allow the moments themselves to be uncertain. We refer to all of this body of work as DRO with *moment-based* uncertainty sets.

While moment-based DRO has a rich history, it is not well suited to many machine learning problems, where the distributions are not well captured by moments alone. One could hardly capture the distribution of natural images, for example, by its mean and variance. Instead there has been extensive study of DRO with *discrepancy-based* uncertainty sets. Here, the uncertainty set $\mathcal{U}$ is given as a ball centered on some nominal distribution $\mathbb{P}_0$, i.e. $\mathcal{U} = \{\mathbb{Q} : D(\mathbb{Q}, \mathbb{P}_0) \leq \varepsilon\}$, where D is some discrepancy measure.

In this thesis, as in most DRO work in machine learning, we focus on *data-driven DRO*, where the nominal distribution $\mathbb{P}_0$ at the center of $\mathcal{U}$ is taken to be an empirical sample $\hat{\mathbb{P}}_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{z_i}$. In other words, we consider perturbations of our empirical distribution, e.g. our dataset of images. In data-driven DRO, the size or radius ε of the uncertainty set $\mathcal{U}$ is also determined in a data-dependent way, and may depend on the number of samples n, how spread out the data is, and the particular choice of discrepancy D. The choice of discrepancy D also determines how tractable the DRO problem is, and is therefore critical.

In machine learning, two choices of the discrepancy D are prevalent: ϕ-divergences (Ben-Tal et al., 2013; Duchi et al., 2016; Lam, 2016), and Wasserstein distance (Mohajerin Esfahani and Kuhn, 2018; Shafieezadeh Abadeh et al., 2015; Blanchet et al.,

2019). The first option, ϕ-divergences, have the form $D_\phi(\mathbb{P}||\mathbb{Q}) = \int \phi(d\mathbb{P}/d\mathbb{Q})\, d\mathbb{Q}$, and include χ^2 divergence and Kullback-Leibler divergence. DRO with χ^2-divergence is roughly equivalent to regularizing by variance (Maurer and Pontil, 2009; Gotoh et al., 2018; Lam, 2016; Duchi et al., 2016; Namkoong and Duchi, 2017), and, as we will see in Chapter 5, the worst case distribution $\mathbb{Q} \in \mathcal{U}$ can be computed exactly in $O(n \log n)$.

The second option, Wasserstein distance, is defined in terms of a distance metric g on the data space. The p-Wasserstein distance W_p between measures μ, ν is given by $W_p(\mu, \nu) = \inf\{\int g(x,y)^p \, d\gamma(x,y) : \gamma \in \Pi(\mu,\nu)\}^{1/p}$, where $\Pi(\mu,\nu)$ is the set of couplings of μ and ν (Villani, 2008). DRO with Wasserstein distance is asymptotically equivalent to certain common norm penalties (Gao et al., 2017), and the worst case $\mathbb{Q} \in \mathcal{U}$ can be computed approximately in several cases (Mohajerin Esfahani and Kuhn, 2018; Gao and Kleywegt, 2016). While most results for Wasserstein DRO focus on the case when g is Euclidean, there are extensions available to e.g. Mahalanobis distances (Blanchet et al., 2017, 2018). Concentration results bounding $W_p(\mathbb{P}, \hat{\mathbb{P}}_n)$ with high probability, which are needed to determine the radius ε of the uncertainty set $\mathcal{U}$, are available for many settings, e.g. (Fournier and Guillin, 2015; Lei, 2020; Singh and Póczos, 2018; Weed et al., 2019).

2.2.1 Far-reaching relevance to machine learning

DRO sheds light on both notions of generalization. For the broader notion of generalization, note that DRO generalizes adversarial training (Sinha et al., 2018; Staib and Jegelka, 2017a). DRO has also been applied to fairness across groups (Hu et al., 2018; Oren et al., 2019; Sagawa* et al., 2020; Hashimoto et al., 2018). Causal inference can be understood in terms of DRO, e.g. see the paper of Meinshausen (2018). In theory one could encode many other desiderata via DRO, e.g. invariance to dataset shifts. Statistical generalization also enjoys many connections to DRO, in particular via regularization. We will explore new such connections in Chapter 3.

Overall, DRO is a useful tool in machine learning now. As optimization technology improves, we expect it will become easier to encode more sophisticated desiderata

with the language of DRO. At present, however, DRO with these rich uncertainty sets remains mostly impractical. Instead, DRO with discrepancy-based uncertainty sets are the most tractable and most relevant to machine learning. As we will see in Chapter 5, even discrepancy-based DRO problems can be challenging to solve, especially for large scale problems and non-convex objectives. Hence, in this thesis, we focus on DRO with discrepancy-based uncertainty sets. Nevertheless, we remain optimistic about the future of DRO in machine learning.

Chapter 3

DRO, MMD, kernels, and generalization

3.1 Introduction

In Chapter 2 we gave background on DRO and, in particular, discrepancy-based DRO. The two prototypical cases are ϕ-divergence and Wasserstein DRO. There has been much fruitful effort towards understanding and solving these DRO problems, and connecting them to statistical generalization. But these two classes of DRO problems are not a panacea: in this chapter we argue that there is a third type of discrepancy measure worth considering.

Our motivation comes from an extremely direct way of linking DRO and statistical generalization. One of the main objects of study in statistical generalization are generalization bounds, i.e. certified upper bounds on the population loss or error of a learned model. We propose the following straightforward path towards using DRO as a tool to prove such generalization bounds:

Principle 3.1.1 (DRO Generalization Principle)**.** Suppose we have learned a candidate model h to predict y from x, i.e. $y \approx h(x)$. Let $\mathcal{U}$ be a set of distributions containing the empirical distribution $\hat{\mathbb{P}}_n$. Suppose $\mathcal{U}$ is large enough so that, with probability $1 - \delta$, $\mathcal{U}$ contains the population $\mathbb{P}$. Then with probability $1 - \delta$, the

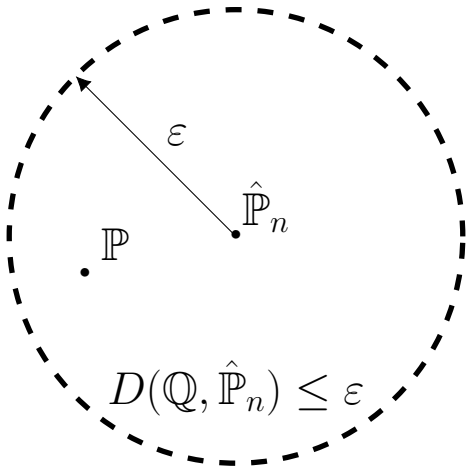


Figure 3-1: A pictorial representation of the DRO Generalization Principle. If the population distribution is in the uncertainty set $\mathcal{U}$, then the worst case performance over all elements of $\mathcal{U}$ bounds the population performance.

population loss $\mathbb{E}_{x\sim\mathbb{P}}[\ell_h(x)]$ is bounded by

$$\mathbb{E}_{x\sim\mathbb{P}}[\ell_h(x)] \;\leq\; \sup_{\mathbb{Q}\in\mathcal{U}} \mathbb{E}_{x\sim\mathbb{Q}}[\ell_h(x)]. \tag{3.1}$$

We focus on uncertainty sets $\mathcal{U}$ defined by $\mathcal{U} = \{\mathbb{Q} : D(\mathbb{Q}, \hat{\mathbb{P}}_n) \leq \varepsilon\}$ for some divergence measure D. Principle 3.1.1 is described pictorially in Figure 3-1. While Principle 3.1.1 is simple and intuitive, it is difficult to use with current DRO technology. This is due to drawbacks to the ϕ-divergence and Wasserstein uncertainty sets currently used. Any ϕ-divergence uncertainty set $\mathcal{U}$ around $\hat{\mathbb{P}}_n$ contains only distributions with the same (finite) support as $\hat{\mathbb{P}}_n$. Hence, the population $\mathbb{P}$ is typically *not* in $\mathcal{U}$, and so the DRO objective value cannot directly certify out of sample performance. Wasserstein uncertainty sets do not suffer from this problem. But, they are more computationally expensive, and the key results on equivalences (to regularization) and computation are typically limited to convex objectives or are only asymptotic bounds.

We introduce and develop a new class of DRO problems, where the uncertainty set $\mathcal{U}$ is defined with respect to maximum mean discrepancy (MMD) (Gretton et al., 2012), a kernel-based distance between distributions. MMD DRO complements existing approaches and avoids some of their drawbacks. In particular, with MMD DRO

we can directly apply Principle 3.1.1 because unlike ϕ-divergences, the uncertainty set $\mathcal{U}$ will contain $\mathbb{P}$ if the radius is large enough.

First, we show that MMD DRO is roughly equivalent to regularizing by the Hilbert norm $\|\ell_h\|_{\mathcal{H}}$ of the loss ℓ_h (not the model h). While, in general, $\|\ell_h\|_{\mathcal{H}}$ may be difficult to compute, we show settings in which it is tractable. Specifically, for kernel ridge regression with a Gaussian kernel, we prove a bound on $\|\ell_h\|_{\mathcal{H}}$ that, as a byproduct, yields generalization bounds that match (up to a small constant) the standard ones. Along the way, we prove bounds on the Hilbert norm of products of functions that may be of independent interest. These bounds also suggest an alternate regularizer for kernel ridge regression.

Second, beyond kernel methods, we show how MMD DRO can be efficiently approximate empirically. This approximation leads to another insight: MMD DRO generalizes variance-based regularization.

Overall, our results offer deeper insights into the landscape of regularization and robustness approaches, and a more complete picture of the effects of different divergences for defining robustness.

3.2 Background and related work

Background information on DRO is given in Chapter 2. Here we briefly discuss other related areas of work, namely MMD and penalization by the Hilbert norm $\|\cdot\|_{\mathcal{H}}$.

Maximum Mean Discrepancy (MMD). MMD is a distance metric between distributions that leverages kernel embeddings. Let $\mathcal{H}$ be a reproducing kernel Hilbert space (RKHS) with kernel k and norm $\|\cdot\|_{\mathcal{H}}$. MMD is defined as follows:

Definition 3.2.1. The *maximum mean discrepancy (MMD)* between distributions $\mathbb{P}$ and $\mathbb{Q}$ is

$$d_{\text{MMD}}(\mathbb{P}, \mathbb{Q}) := \sup_{g \in \mathcal{H}: \|g\|_{\mathcal{H}} \leq 1} \mathbb{E}_{x \sim \mathbb{P}}[g(x)] - \mathbb{E}_{x \sim \mathbb{Q}}[g(x)]. \tag{3.2}$$

Fact 3.2.1. Define the mean embedding $\mu_{\mathbb{P}}$ of the distribution $\mathbb{P}$ by $\mu_{\mathbb{P}} = \mathbb{E}_{x\sim\mathbb{P}}[k(x,\cdot)]$. Then the MMD between distributions $\mathbb{P}$ and $\mathbb{Q}$ can be equivalently written

$$d_{\mathrm{MMD}}(\mathbb{P},\mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}}. \tag{3.3}$$

MMD and (more generally) kernel mean embeddings have been used in many applications, particularly in two- and one-sample tests (Gretton et al., 2012; Jitkrittum et al., 2017; Liu et al., 2016; Chwialkowski et al., 2016) and in generative modeling (Dziugaite et al., 2015; Li et al., 2015; Sutherland et al., 2017; Bikowski et al., 2018). We refer the interested reader to the monograph by Muandet et al. (2017). MMD admits efficient estimation, as well as fast convergence properties, which are of chief importance in our work.

Further related work. In Chapter 2 we discussed work in operations research that has considered DRO problems that capture uncertainty in moments of the distribution, e.g. (Delage and Ye, 2010). These approaches typically focus on first- and second-order moments; in contrast, an MMD uncertainty set allows high order moments to vary, depending on the choice of kernel.

Beyond DRO, Xu et al. (2009) study the connection between robustness and regularization in SVMs, and perturbations within a (possibly Hilbert) norm ball. Unlike our work, their results are limited to SVMs instead of general loss minimization. Moreover, they consider only perturbation of individual data points instead of shifts in the entire *distribution*. Bietti et al. (2019) show that many regularizers used for neural networks can also be interpreted in light of an appropriately chosen Hilbert norm (Bietti and Mairal, 2019).

3.3 Generalization bounds via MMD DRO

The main focus of this chapter is Distributionally Robust Optimization where the uncertainty set is defined via the MMD distance d_{MMD}:

$$\inf_{h} \sup_{\mathbb{Q}: d_{\text{MMD}}(\mathbb{Q}, \hat{\mathbb{P}}_n) \leq \varepsilon} \mathbb{E}_{x \sim \mathbb{Q}}[\ell_h(x)]. \tag{3.4}$$

One motivation for considering MMD in this setting are its possible implications for Generalization. Recall that for the DRO Generalization Principle 3.1.1 to apply, the uncertainty set $\mathcal{U}$ must contain the population distribution with high probability. To ensure this, the radius of $\mathcal{U}$ must be large enough. But, the larger the radius, the more pessimistic is the DRO minimax problem, which may lead to over-regularization. This radius depends on how quickly $d_{\text{MMD}}(\mathbb{P}, \hat{\mathbb{P}}_n)$ shrinks to zero, i.e., on the empirical accuracy of the divergence.

In contrast to Wasserstein distance, which converges at a rate of $O(n^{-1/d})$ (Fournier and Guillin, 2015), MMD between the empirical sample $\hat{\mathbb{P}}_n$ and population $\mathbb{P}$ shrinks as $O(n^{-1/2})$:

Lemma 3.3.1 (Modified from (Muandet et al., 2017), Theorem 3.4)**.** *Suppose that $k(x, x) \leq M$ for all x. Let $\hat{\mathbb{P}}_n$ be an n sample empirical approximation to $\mathbb{P}$. Then with probability $1 - \delta$,*

$$d_{\text{MMD}}(\mathbb{P}, \hat{\mathbb{P}}_n) \leq 2\sqrt{\tfrac{M}{n}} + \sqrt{\tfrac{2\log(1/\delta)}{n}}. \tag{3.5}$$

The constant M is dimension-independent for many common universal kernels, e.g. Gaussian, Laplace, and Matern kernels. With Lemma 3.3.1 in hand, we conclude a simple high probability bound on out-of-sample performance:

Corollary 3.3.1. *Suppose that $k(x, x) \leq M$ for all x. Set the uncertainty set radius ε to $\varepsilon = 2\sqrt{M/n} + \sqrt{2\log(1/\delta)/n}$. Then with probability $1 - \delta$, we have the following*

bound on population risk:

$$\mathbb{E}_{x\sim\mathbb{P}}[\ell_h(x)] \leq \sup_{\mathbb{Q}:d_{\mathrm{MMD}}(\mathbb{Q},\hat{\mathbb{P}}_n)\leq\varepsilon} \mathbb{E}_{x\sim\mathbb{Q}}[\ell_h(x)]. \tag{3.6}$$

We refer to the right hand side as the DRO adversary's problem. In the next section we develop results that enable us to bound its value, and consequently bound the DRO problem (3.4).

3.3.1 Bounding the DRO adversary's problem

The DRO adversary's problem seeks the distribution $\mathbb{Q}$ in the MMD ball so that $\mathbb{E}_{x\sim\mathbb{Q}}[\ell_h(x)]$ is as high as possible. Reasoning about the optimal worst-case $\mathbb{Q}$ is the main difficulty in DRO. With MMD, we take two steps for simplification. First, instead of directly optimizing over distributions, we optimize over their mean embeddings in the Hilbert space (described in Fact 3.2.1). Second, while the adversary's problem (3.6) makes sense for general ℓ_h, we assume that the loss ℓ_h is in $\mathcal{H}$. In case $\ell_h \notin \mathcal{H}$, often k is a universal kernel, meaning under mild conditions ℓ_h can be approximated arbitrarily well by a member of $\mathcal{H}$ (Muandet et al., 2017, Definition 3.3).

With the additional assumption that $\ell_h \in \mathcal{H}$, the risk $\mathbb{E}_{x\sim\mathbb{P}}[\ell_h(x)]$ can also be written as $\langle \ell_h, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$. Then we obtain

$$\sup_{\mathbb{Q}:d_{\mathrm{MMD}}(\mathbb{Q},\mathbb{P})\leq\varepsilon} \mathbb{E}_{x\sim\mathbb{Q}}[\ell_h(x)] \leq \sup_{\mu_{\mathbb{Q}}\in\mathcal{H}:\|\mu_{\mathbb{Q}}-\mu_{\mathbb{P}}\|_{\mathcal{H}}\leq\varepsilon} \langle \ell_h, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}}, \tag{3.7}$$

where we have an inequality because not every function in $\mathcal{H}$ is the mean embedding of some probability distribution. If k is a characteristic kernel (Muandet et al., 2017, Definition 3.2), the mapping $\mathbb{P} \mapsto \mu_{\mathbb{P}}$ is injective. In this case, the only looseness in the bound is due to discarding the constraints that $\mathbb{Q}$ integrates to one and is nonnegative. However it is difficult to constrain the mean embedding $\mu_{\mathbb{Q}}$ in this way as it is a function.

The mean embedding form of the problem is simpler to work with, and leads to

further interpretations.

Theorem 3.3.1. *Let $\ell_h, \mu_{\mathbb{P}} \in \mathcal{H}$. We have the following equality:*

$$\sup_{\mu_{\mathbb{Q}} \in \mathcal{H}: \|\mu_{\mathbb{Q}} - \mu_{\mathbb{P}}\|_{\mathcal{H}} \leq \varepsilon} \langle \ell_h, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} = \langle \ell_h, \mu_{\mathbb{P}} \rangle_{\mathcal{H}} + \varepsilon \|\ell_h\|_{\mathcal{H}} = \mathbb{E}_{x \sim \mathbb{P}}[\ell_h(x)] + \varepsilon \|\ell_h\|_{\mathcal{H}}. \tag{3.8}$$

In particular, the adversary's optimal solution is $\mu^*_{\mathbb{Q}} = \mu_{\mathbb{P}} + \frac{\varepsilon}{\|\ell\|_{\mathcal{H}}} \ell_h$.

Combining Theorem 3.3.1 with equation (3.7) yields our main result for this section:

Corollary 3.3.2. *Let $\ell_h \in \mathcal{H}$, let $\mathbb{P}$ be a probability distribution, and fix $\varepsilon > 0$. Then,*

$$\sup_{\mathbb{Q}: d_{\mathrm{MMD}}(\mathbb{P}, \mathbb{Q}) \leq \varepsilon} \mathbb{E}_{x \sim \mathbb{Q}}[\ell_h(x)] \;\leq\; \mathbb{E}_{x \sim \mathbb{P}}[\ell_h(x)] + \varepsilon \|\ell_h\|_{\mathcal{H}} \qquad \textit{and therefore} \tag{3.9}$$

$$\inf_h \sup_{\mathbb{Q}: d_{\mathrm{MMD}}(\mathbb{P}, \mathbb{Q}) \leq \varepsilon} \mathbb{E}_{x \sim \mathbb{Q}}[\ell_h(x)] \;\leq\; \inf_h \mathbb{E}_{x \sim \mathbb{P}}[\ell_h(x)] + \varepsilon \|\ell_h\|_{\mathcal{H}}. \tag{3.10}$$

Combining Corollary 3.3.2 with Corollary 3.3.1 shows that minimizing the empirical risk plus a norm on ℓ_h leads to a high probability bound on out-of-sample performance. This result is similar to results that equate Wasserstein DRO to norm regularization. For example, Gao et al. (2017) show that under appropriate assumptions on ℓ_h, DRO with a p-Wasserstein ball is asymptotically equivalent to $\mathbb{E}_{x \sim \hat{\mathbb{P}}_n}[\ell_h(x)] + \varepsilon \|\nabla_x \ell_h\|_{\hat{\mathbb{P}}_n, q}$, where $\|\nabla_x \ell_h\|_{\hat{\mathbb{P}}_n, q} = \left(\frac{1}{n} \sum_{i=1}^n \|\nabla_x \ell_h(x_i)\|_*^q\right)^{1/q}$ measures a kind of q-norm average of $\|\nabla_x \ell_h(x_i)\|_*$ at each data point x_i (here q is such that $1/p + 1/q = 1$, and $\|\cdot\|_*$ is the dual norm of the metric defining the Wasserstein distance).

There are a few key differences between our result and that of Gao et al. (2017). First, the norms are different. Second, their result penalizes only the gradient of ℓ_h, while ours penalizes ℓ_h directly. Third, except for certain special cases, the Wasserstein results cannot serve as a true upper bound; there are higher order terms that only shrink to zero as $\varepsilon \to 0$. These higher order terms may not be so small: in high dimension d, the radius ε of the uncertainty set needed so that $\mathbb{P} \in \mathcal{U}$ shrinks very slowly, as $O(n^{-1/d})$ (Fournier and Guillin, 2015).

Remark 3.3.1. Theorem 3.3.1 and Corollary 3.3.2 require that ℓ_h is in the RKHS $\mathcal{H}$. Though this may seem restrictive, if the kernel k is universal, as is the case for many kernels used in practice such as Gaussian and Laplace kernels, we can readily extend our results to all bounded continuous functions. Suppose ℓ_h is a bounded continuous function on a compact metric space $\mathcal{X}$. By definition (e.g. (Muandet et al., 2017), Definition 3.3), if k is a universal kernel on $\mathcal{X}$, then for any $\varepsilon > 0$, there is some $\ell' \in \mathcal{H}$ with $\sup_{x \in \mathcal{X}} |\ell_h(x) - \ell'(x)| < \varepsilon$. It follows that for any measure $\mathbb{P}$, we can bound the expectation of $\ell_h(x)$ by that of ℓ': $\mathbb{E}_{x \sim \mathbb{P}}[\ell_h(x)] < \mathbb{E}_{x \sim \mathbb{P}}[\ell'(x)] + \varepsilon$. Then, we can apply our results to $\ell' \in \mathcal{H}$.

3.4 Connections to kernel ridge regression

After applying Corollary 3.3.2, we are interested in solving:

$$\inf_h \mathbb{E}_{x \sim \hat{\mathbb{P}}_n}[\ell_h(x)] + \varepsilon \|\ell_h\|_{\mathcal{H}}. \tag{3.11}$$

Here, we penalize our model h by $\|\ell_h\|_{\mathcal{H}}$. This looks similar to but is very different from the usual penalty $\|h\|_{\mathcal{H}}$ in kernel methods. In fact, Hilbert norms of function compositions such as ℓ_h pose several challenges. For example, h and ℓ_h may not belong to the same RKHS – it is not hard to construct counterexamples, even when ℓ is merely quadratic. So, the objective (3.11) is not yet computational.

Despite these challenges, we next develop tools that will allow us to bound $\|\ell_h\|_{\mathcal{H}}$ and use it as a regularizer. These tools may be of independent interest to bound RKHS norms of composite functions (e.g., for settings as in (Bietti et al., 2019)). Due to the difficulty of this task, we specialize to Gaussian kernels $k_\sigma(x, y) = \exp(-\|x - y\|^2/(2\sigma^2))$. In this setting, our results apply pretty generally: the norm inside the expression for k_σ can be any norm that has an associated inner product; for example, it can be a Mahalanobis norm, in which case k_σ can be interpreted as a Gaussian kernel with general covariance. Since we will need to take care regarding the bandwidth σ, we explicitly write it out for the inner product $\langle \cdot, \cdot \rangle_\sigma$ and norm $\|\cdot\|_\sigma$, of the corresponding

RKHS H_σ.

To make the setting concrete, consider kernel ridge regression, with Gaussian kernel k_σ. As usual, we assume there is a simple target function h^* that fits our data: $h^*(x_i) = y_i$. Then the loss ℓ_h of h is $\ell_h(x) = (h(x) - h^*(x))^2$, so we wish to solve

$$\inf_h \mathbb{E}_{x\sim\hat{\mathbb{P}}_n}[(h(x) - h^*(x))^2] + \varepsilon\|(h - h^*)^2\|_\sigma. \tag{3.12}$$

3.4.1 Bounding norms of products

To bound $\|(h - h^*)^2\|_\sigma$, it will suffice to bound RKHS norms of products. The key result for this subsection is the following deceptively simple-looking bound:

Theorem 3.4.1. *Let $f, g \in \mathcal{H}_\sigma$, that is, the RKHS corresponding to the Gaussian kernel k_σ of bandwidth σ. Then, $\|fg\|_{\sigma/\sqrt{2}} \le \|f\|_\sigma \|g\|_\sigma$.*

Indeed, there are already subtleties: if $f, g \in \mathcal{H}_\sigma$, then, to discuss the norm of the product fg, we need to decrease the bandwidth from σ to $\sigma/\sqrt{2}$.

We prove Theorem 3.4.1 via two steps. First, we represent the functions f, g, and fg *exactly* in terms of traces of certain matrices. This step is highly dependent on the specific structure of the Gaussian kernel. Then, we can apply standard trace inequalities. Proofs of both results are given in Appendix A.2.

Proposition 3.4.1. *Let $f, g \in \mathcal{H}_\sigma$ have expansions $f = \sum_i a_i k_\sigma(x_i, \cdot)$ and $g = \sum_j b_j k_\sigma(x_j, \cdot)$. For shorthand denote by $z_i = \phi_{\sqrt{2}\sigma}(x_i)$ the (possibly infinite) feature expansion of x_i in $\mathcal{H}_{\sqrt{2}\sigma}$. Then,*

$$\|fg\|^2_{\sigma/\sqrt{2}} = \operatorname{tr}(A^2B^2), \quad \|f\|^2_\sigma = \operatorname{tr}(A^2), \quad \text{and} \quad \|g\|^2_\sigma = \operatorname{tr}(B^2),$$

where $A = \sum_i a_i z_i z_i^T$ and $B = \sum_j a_j z_j z_j^T$.

Lemma 3.4.1. *Let X, Y be symmetric and positive semidefinite. Then $\operatorname{tr}(XY) \le \operatorname{tr}(X)\operatorname{tr}(Y)$.*

With these intermediate results in hand, we can prove the main bound of interest:

Proof of Theorem 3.4.1. By Proposition 3.4.1, we may write

$$\|fg\|_{\sigma/\sqrt{2}}^2 = \operatorname{tr}(A^2B^2), \quad \|f\|_\sigma^2 = \operatorname{tr}(A^2), \quad \text{and} \quad \|g\|_\sigma^2 = \operatorname{tr}(B^2),$$

where $A = \sum_i a_i z_i z_i^T$ and $B = \sum_j b_j z_j z_j^T$ are chosen as described in Proposition 3.4.1. Since A and B are each symmetric, it follows that A^2 and B^2 are each symmetric and positive semidefinite. Then we can apply Lemma 3.4.1 to conclude that

$$\|fg\|_{\sigma/\sqrt{2}}^2 = \operatorname{tr}(A^2B^2) \le \operatorname{tr}(A^2)\operatorname{tr}(B^2) = \|f\|_\sigma^2\|g\|_\sigma^2. \qquad \square$$

3.4.2 Implications: kernel ridge regression

With the help of Theorem 3.4.1, we can develop DRO-based bounds for actual learning problems. In this section we develop such bounds for Gaussian kernel ridge regression, i.e. problem (3.12).

For shorthand, we write $R_{\mathbb{Q}}(h) = \mathbb{E}_{x\sim\mathbb{Q}}[\ell_h(x)] = \mathbb{E}_{x\sim\mathbb{Q}}[(h(x) - h^*(x))^2]$ for the risk of h on a distribution $\mathbb{Q}$. Generalization amounts to proving that the population risk $R_{\mathbb{P}}(h)$ is not too different than the empirical risk $R_{\hat{\mathbb{P}}_n}(h)$.

Theorem 3.4.2. *Assume the target function h^* satisfies $\|(h^*)^2\|_{\sigma/\sqrt{2}} \le \Lambda_{(h^*)^2}$ and $\|h^*\|_\sigma \le \Lambda_{h^*}$. Then, for any $\delta > 0$, with probability $1-\delta$, the following holds for all functions h satisfying $\|h^2\|_{\sigma/\sqrt{2}} \le \Lambda_{h^2}$ and $\|h\|_\sigma \le \Lambda_h$:*

$$R_{\mathbb{P}}(h) \le R_{\hat{\mathbb{P}}_n}(h) + \tfrac{2}{\sqrt{n}}\left(1 + \sqrt{\tfrac{\log(1/\delta)}{2}}\right)\left(\Lambda_{h^2} + \Lambda_{(h^*)^2} + 2\Lambda_h\Lambda_{h^*}\right). \tag{3.13}$$

Proof. We utilize the DRO Generalization Principle 3.1.1. By Lemma 3.3.1 we know that with probability $1-\delta$, $d_{\mathrm{MMD}}(\hat{\mathbb{P}}_n, \mathbb{P}) \le \varepsilon$ for $\varepsilon = (2 + \sqrt{2\log(1/\delta)})/\sqrt{n}$, since $k_\sigma(x,x) \le M = 1$. Note the bandwidth σ does not affect the convergence result. As

a result of Lemma 3.3.1, with probability $1-\delta$:

$$R_{\mathbb{P}}(h) = \mathbb{E}_{x\sim\mathbb{P}}[(h(x)-h^*(x))^2] \tag{3.14}$$

$$\overset{(a)}{\leq} \mathbb{E}_{x\sim\hat{\mathbb{P}}_n}[(h(x)-h^*(x))^2] + \varepsilon\|(h-h^*)^2\|_{\sigma/\sqrt{2}} \tag{3.15}$$

$$\overset{(b)}{\leq} R_{\hat{\mathbb{P}}_n}(h) + \varepsilon\left(\|h^2\|_{\sigma/\sqrt{2}} + \|(h^*)^2\|_{\sigma/\sqrt{2}} + 2\|hh^*\|_{\sigma/\sqrt{2}}\right) \tag{3.16}$$

$$\overset{(c)}{\leq} R_{\hat{\mathbb{P}}_n}(h) + \varepsilon\left(\Lambda_{h^2} + \Lambda_{(h^*)^2} + 2\Lambda_h\Lambda_{h^*}\right), \tag{3.17}$$

where (a) is by Corollary 3.3.2, (b) is by the triangle inequality, and (c) follows from Theorem 3.4.1 and our assumptions on h and h^*. Plugging in the bound on ε yields the result. □

We placed different bounds on each of $h, h^*, h^2, (h^*)^2$ to emphasize the dependence on each. Since each is bounded separately, the DRO based bound in Theorem 3.4.2 allows finer control of the complexity of the function class than is typical. Since, by Theorem 3.4.1, the norms of h^2, $(h^*)^2$ and hh^* are bounded by those of h and h^*, we may also state Theorem 3.4.2 just with $\|h\|_\sigma$ and $\|h^*\|_\sigma$.

Corollary 3.4.1. *Assume the target function h^* satisfies $\|h^*\|_\sigma \leq \Lambda$. Then, for any $\delta > 0$, with probability $1-\delta$, the following holds for all functions h satisfying $\|h\|_\sigma \leq \Lambda$:*

$$R_{\mathbb{P}}(h) \leq R_{\hat{\mathbb{P}}_n}(h) + \tfrac{8\Lambda^2}{\sqrt{n}}\left(1+\sqrt{\tfrac{\log(1/\delta)}{2}}\right). \tag{3.18}$$

Proof. We reduce to Theorem 3.4.2. By Theorem 3.4.1, we know that $\|h^2\|_{\sigma/\sqrt{2}} \leq \|h\|_\sigma^2$, which may be bounded above by Λ^2 (and similarly for h^*). Therefore we can take $\Lambda_{h^2} = \Lambda_h^2 = \Lambda$ and $\Lambda_{(h^*)^2} = \Lambda_{h^*}^2 = \Lambda$ in Theorem 3.4.2. The result follows by bounding

$$\Lambda_{h^2} + \Lambda_{(h^*)^2} + 2\Lambda_h\Lambda_{h^*} \leq \Lambda^2 + \Lambda^2 + 2\Lambda\cdot\Lambda = 4\Lambda^2.$$ □

Generalization bounds for kernel ridge regression are of course not new; we emphasize that the DRO viewpoint provides an intuitive approach that also grants finer control over the function complexity. Moreover, our results take essentially the same

form as the typical generalization bounds for kernel ridge regression, reproduced below:

Theorem 3.4.3 (Specialized from (Mohri et al., 2018), Theorem 10.7)**.** *Assume the target function h^* satisfies $\|h^*\|_\sigma \leq \Lambda$. Then, for any $\delta > 0$, with probability $1-\delta$, it holds for all functions h satisfying $\|h\|_\sigma \leq \Lambda$ that*

$$R_{\mathbb{P}}(h) \leq R_{\hat{\mathbb{P}}_n}(h) + \frac{8\Lambda^2}{\sqrt{n}}\left(1 + \frac{1}{2}\sqrt{\frac{\log(1/\delta)}{2}}\right). \tag{3.19}$$

Hence, our DRO-based Theorem 3.4.2 evidently recovers standard results up to a universal constant.

3.4.3 Algorithmic implications

The generalization result in Theorem 3.4.3 is often used to justify penalizing by the norm $\|h\|_\sigma$, since it is the only part of the bound (other than the risk $R_{\hat{\mathbb{P}}_n}(h)$) that depends on h. In contrast, our DRO-based generalization bound in Theorem 3.4.2 is of the form

$$R_{\mathbb{P}}(h) - R_{\hat{\mathbb{P}}_n}(h) \leq \varepsilon\left(\|h^2\|_{\sigma/\sqrt{2}} + \|(h^*)^2\|_{\sigma/\sqrt{2}} + 2\|h\|_\sigma\|h^*\|_\sigma\right), \tag{3.20}$$

which depends on h through both norms $\|h\|_\sigma$ and $\|h^2\|_{\sigma/\sqrt{2}}$. This bound motivates the use of both norms as regularizers in kernel regression, i.e. we would instead solve

$$\inf_{h\in\mathcal{H}_\sigma} \mathbb{E}_{(x,y)\sim\hat{\mathbb{P}}_n}[(h(x)-y)^2] + \lambda_1\|h\|_\sigma + \lambda_2\|h^2\|_{\sigma/\sqrt{2}}. \tag{3.21}$$

Given data $(x_i, y_i)_{i=1}^n$, for kernel ridge regression, the Representer Theorem implies that it is sufficient to consider only h of the form $h = \sum_{i=1}^n a_i k_\sigma(x_i, \cdot)$. Here this is not in general possible due to the norm of h^2. However, it is possible to evaluate and compute gradients of $\|h^2\|^2_{\sigma/\sqrt{2}}$: let K be the matrix with $K_{ij} = k_{\sqrt{2}\sigma}(x_i, x_j)$, and let $D = \mathrm{diag}(a)$. Using Proposition 3.4.1, we can prove $\|h^2\|^2_{\sigma/\sqrt{2}} = \mathrm{tr}((DK)^4)$ A complete proof is given in Corollary A.2.1 in the appendix.

3.5 Approximation and connections to variance regularization

In the previous section we studied bounding the MMD DRO problem (3.4) via Hilbert norm penalization. Going beyond kernel methods where we search over $h \in \mathcal{H}$, it is even less clear how to evaluate the Hilbert norm $\|\ell_h\|_{\mathcal{H}}$. To circumvent this issue, next we approach the DRO problem from a different angle: we directly search for the adversarial distribution $\mathbb{Q}$. Along the way, we will build connections to variance regularization (Maurer and Pontil, 2009; Gotoh et al., 2018; Lam, 2016; Namkoong and Duchi, 2017), where the empirical risk is regularized by the empirical variance of ℓ_h: $\mathrm{Var}_{\hat{\mathbb{P}}_n}(\ell_h) = \mathbb{E}_{x\sim\hat{\mathbb{P}}_n}[\ell_h(x)^2] - \mathbb{E}_{x\sim\hat{\mathbb{P}}_n}[\ell_h(x)]^2$. In particular, we show in Theorem 3.5.1 that MMD DRO yields stronger regularization than variance.

Searching over all distributions $\mathbb{Q}$ in the MMD ball is intractable, so we restrict our attention to those with the same support $\{x_i\}_{i=1}^n$ as the empirical sample $\hat{\mathbb{P}}_n$. All such distributions $\mathbb{Q}$ can be written as $\mathbb{Q} = \sum_{i=1}^n w_i \delta_{x_i}$, where w is in the n-dimensional simplex. By restricting the set of candidate distributions $\mathbb{Q}$, we make the adversary weaker:

$$\begin{array}{ll} \sup_{\mathbb{Q}} & \mathbb{E}_{x\sim\mathbb{Q}}[\ell_h(x)] \\ \text{s.t.} & d_{\mathrm{MMD}}(\mathbb{Q}, \hat{\mathbb{P}}_n) \le \varepsilon \end{array} \quad \geq \quad \begin{array}{ll} \sup_{w} & \sum_{i=1}^n w_i \ell_h(x_i) \\ \text{s.t.} & d_{\mathrm{MMD}}(\sum_{i=1}^n w_i \delta_{x_i}, \hat{\mathbb{P}}_n) \le \varepsilon \\ & \sum_{i=1}^n w_i = 1 \\ & w_i \ge 0 \; \forall i = 1, \ldots, n. \end{array} \tag{3.22}$$

By restricting the support of $\mathbb{Q}$, it is no longer possible to guarantee out of sample performance, since it typically will have different support. Yet, as we will see, problem (3.22) has nice connections.

The d_{MMD} constraint is a quadratic penalty on $v = w - \frac{1}{n}\mathbf{1}$, as one may see via

the mean embedding definition of MMD:

$$d_{\text{MMD}}\left(\sum_{i=1}^n w_i\delta_{x_i}, \hat{\mathbb{P}}_n\right)^2 = \left\|\sum_{i=1}^n w_i k(x_i,\cdot) - \frac{1}{n}\sum_{i=1}^n k(x_i,\cdot)\right\|_{\mathcal{H}}^2 = \left\|\sum_{i=1}^n v_i k(x_i,\cdot)\right\|_{\mathcal{H}}^2. \tag{3.23}$$

The last term is $v^T K v = (w - \frac{1}{n}\mathbf{1})^T K (w - \frac{1}{n}\mathbf{1})$, where K is the kernel matrix with $K_{ij} = k(x_i, x_j)$. If the radius ε of the uncertainty set is small enough, the constraints $w_i \geq 0$ are inactive, and can be ignored. By dropping these constraints, we can solve the adversary's problem in closed form:

Lemma 3.5.1. *Let $\vec{\ell}$ be the vector with i-th element $\ell_h(x_i)$. If ε is small enough that the constraints w_i are not active, then the optimal value of problem (3.22) is given by*

$$\mathbb{E}_{x\sim\hat{\mathbb{P}}_n}[\ell_h(x)] + \varepsilon\sqrt{\vec{\ell}^T K^{-1}\vec{\ell} - \frac{(\vec{\ell}^T K^{-1}\mathbf{1})^2}{\mathbf{1}^T K^{-1}\mathbf{1}}}. \tag{3.24}$$

In other words, fitting a model to minimize the support-constrained approximation of MMD DRO is equivalent to penalizing by the nonconvex regularizer in Lemma 3.5.1. To better understand this regularizer, consider, for instance, the case that the kernel matrix K equals the identity I. This will happen e.g. for a Gaussian kernel as the bandwidth σ approaches zero. Then, the regularizer equals

$$\varepsilon\sqrt{\vec{\ell}^T K^{-1}\vec{\ell} - \frac{(\vec{\ell}^T K^{-1}\mathbf{1})^2}{\mathbf{1}^T K^{-1}\mathbf{1}}} = \varepsilon\sqrt{\vec{\ell}^T\vec{\ell} - \frac{(\vec{\ell}^T\mathbf{1})^2}{\mathbf{1}^T\mathbf{1}}} = \varepsilon\sqrt{n}\sqrt{\text{Var}_{\hat{\mathbb{P}}_n}(\ell_h)}. \tag{3.25}$$

In fact, this equivalence holds a bit more generally:

Lemma 3.5.2. *Let $K = aI + b\mathbf{1}\mathbf{1}^T$, so that K_{ij} equals a if $i = j$, and $b + a$ otherwise. Then,*

$$\sqrt{\vec{\ell}^T K^{-1}\vec{\ell} - \frac{(\vec{\ell}^T K^{-1}\mathbf{1})^2}{\mathbf{1}^T K^{-1}\mathbf{1}}} = a^{-1/2}\sqrt{n}\sqrt{\text{Var}_{\hat{\mathbb{P}}_n}(\ell_h)}. \tag{3.26}$$

As a consequence, we conclude that with the right choice of kernel k, MMD DRO is a stronger regularizer than variance:

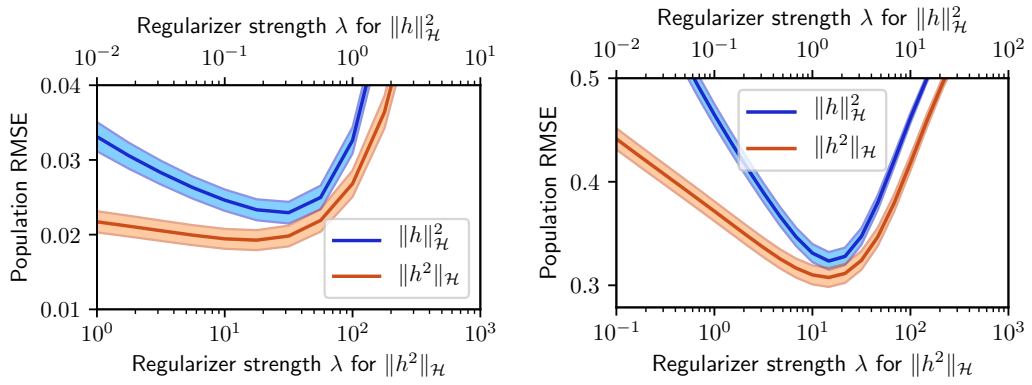


Figure 3-2: Comparison of the two regularizers $\|h\|_\sigma^2$ and $\|h^2\|_{\sigma/\sqrt{2}}$ in both the easy (left) and hard (right) settings, across a parameter sweep of λ. The x-axis is shifted to make comparison easier.

Theorem 3.5.1. *There exists a kernel k so that MMD DRO bounds the variance regularized problem:*

$$\mathbb{E}_{x\sim\hat{\mathbb{P}}_n}[\ell_h(x)] \quad \leq \quad \mathbb{E}_{x\sim\hat{\mathbb{P}}_n}[\ell_h(x)] + \varepsilon\sqrt{n}\sqrt{\mathrm{Var}_{\hat{\mathbb{P}}_n}(\ell_h)} \quad \leq \quad \sup_{\mathbb{Q}:d_{\mathrm{MMD}}(\mathbb{Q},\hat{\mathbb{P}}_n)\leq\varepsilon} [\ell_h(x)]. \tag{3.27}$$

3.6 Experiments

3.6.1 Alternate regularizer

In subsection 3.4.3 we proposed an alternate regularizer for kernel ridge regression, specifically, penalizing $\|h^2\|_{\sigma/\sqrt{2}}$ instead of $\|h\|_\sigma^2$. Here we probe the new regularizer on a synthetic problem where we can precisely compute the population risk $R_{\mathbb{P}}(h)$. Consider the Gaussian kernel k_σ with $\sigma = 1$. Fix the ground truth $h = k_\sigma(1,\cdot) - k_\sigma(-1,\cdot) \in \mathcal{H}_\sigma$. Sample 10^4 points from a standard one dimensional Gaussian, and set this as the population $\mathbb{P}$. Then subsample n points $x_i = h(x_i) + \varepsilon_i$, where ε_i are Gaussian. We consider both an easy regime, where $n = 10^3$ and $\mathrm{Var}(\varepsilon_i) = 10^{-2}$, and a hard regime where $n = 10^2$ and $\mathrm{Var}(\varepsilon_i) = 1$. On the empirical data, we fit $h \in \mathcal{H}_\sigma$ by minimizing square loss plus either $\lambda\|h\|_\sigma^2$ (as is typical) or $\lambda\|h^2\|_{\sigma/\sqrt{2}}$ (our proposal). We average over 10^2 resampling trials for the easy case and 10^3 for the hard

case, and report 95% confidence intervals. Figure 3-2 shows the result in each case for a parameter sweep over λ. If λ is tuned properly, the tighter regularizer $\|h^2\|_{\sigma/\sqrt{2}}$ yields better performance in both cases. It also appears the regularizer $\|h^2\|_{\sigma/\sqrt{2}}$ is less sensitive to the choice of λ: performance decays slowly when λ is too low.

3.6.2 Conjecture: generalizing beyond Gaussian kernels

One limitation of our DRO-based kernel ridge regression bounds is that they apply only for Gaussian kernels. Our proof relies on the fact that, for any bandwidth σ, we can find a smaller bandwidth σ' so that $\|h^2\|_{\sigma'} \leq \|h\|_\sigma^2$. For Gaussian kernels, Theorem 3.4.1 implies we can choose $\sigma' = \sigma/\sqrt{2}$, but at present, we lack similar theoretical results for other kernels.

In this subsection we give empirical evidence that such results may hold for popular kernels. In particular, we study the Laplace and Matérn kernels. The Laplace kernel is defined by $k_\sigma(x, y) = \exp(-\|x - y\|/\sigma)$; the Matérn kernel is defined by $k_\sigma(x, y) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{\|x-y\|}{\sigma}\right)^\nu K_\nu \left(\sqrt{2\nu}\frac{\|x-y\|}{\sigma}\right)$, where K_ν is a modified Bessel function, and $\nu \geq 0$ is a parameter most commonly set to 1.5 or 2.5.

One major difficulty in proving such a bound is computing the norm $\|h^2\|_{\sigma'}$. Computing the RKHS norm $\|f\|_\mathcal{H}$ of a function f is easiest when we are given the expansion $f = \sum_i a_i k(x_i, \cdot)$, in which case we can compute the norm in closed form. But for $f = h^2$ we do not in general have such an expansion. Instead, we numerically estimate the norm $\|h^2\|_{\sigma'}$[1] and empirically validate the bound.

Scaling the bandwidth, and functions with two term expansions

We focus on one-dimensional functions defined on $\mathbb{R}$. First, for each kernel (Laplace; Matérn with $\nu = 1.5, 2.5$), we numerically compute the scaling factor $C = \sigma'/\sigma$ so that $\|k_\sigma(0, \cdot)^2\|_{\sigma'} \leq \|k_\sigma(0, \cdot)\|_\sigma^2$; these results are summarized in Figure 3-3. For Matérn kernels, we suspect that, as ν increases, the required scaling factor should

[1]This is done via approximating the Fourier transform $\mathcal{F}[h^2]$ of h^2, and using the fact that, for shift-invariant kernels k, the norm $\|f\|_\mathcal{H}^2 = \int_\omega |\mathcal{F}[f(\omega)]|^2/\mathcal{F}[k(\omega)]\, d\omega$, up to a constant that depends on how the Fourier transform is defined. Numerically estimating this integral leads to an estimate of $\|h^2\|_{\sigma'}$.

Figure 3-3: Estimates of the ratio σ'/σ needed so that $\|k_\sigma(0,\cdot)^2\|_{\sigma'} \le \|k_\sigma(0,\cdot)\|_\sigma^2$.

	Laplace	Matérn, $\nu = 1.5$	Matérn, $\nu = 2.5$
Bandwidth scale factor	0.5	≈ 0.648	≈ 0.678

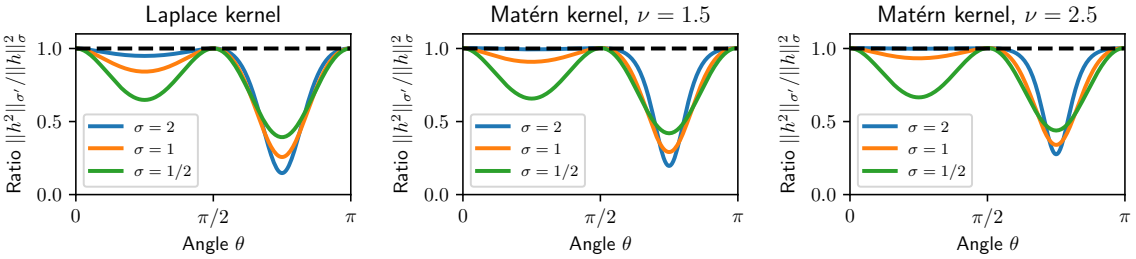


Figure 3-4: Study of the ratio $\|h^2\|_{\sigma'}/\|h\|_\sigma^2$ across different kernels and bandwidths σ, for functions of the form $h = \cos\theta\, k_\sigma(0,\cdot) + \sin\theta\, k_\sigma(1,\cdot)$. The ratio never exceeds 1, implying $\|h^2\|_{\sigma'} \le \|h\|_\sigma^2$ for all functions of this form.

approach $1/\sqrt{2} \approx 0.707$; this is because as ν increases, the Matérn kernel approaches the Gaussian kernel, for which we already proved that $1/\sqrt{2}$ suffices. Through the rest of this section, we fix the scaling factor for each kernel in accordance with Figure 3-3, e.g. $\sigma' = \sigma/2$ for the Laplace kernel.

Then, we study the inequality $\|h^2\|_{\sigma'} \le \|h\|_\sigma^2$ for all functions h with two terms in their expansion: $h = a_1 k_\sigma(x_1,\cdot) + a_2 k_\sigma(x_2,\cdot)$. By shift-invariance, instead of varying x_1 and x_2, we can set $x_1 = 0$ and $x_2 = 1$ and vary the bandwidth σ. And because norms are positively homogeneous, it suffices to study weights $a = (a_1, a_2)$ with $\|a\|_2 = 1$. We parameterize such functions in terms of an angle θ: $h = \cos\theta\, k_\sigma(0,\cdot) + \sin\theta\, k_\sigma(1,\cdot)$. In Figure 3-4 we plot the ratio $\|h^2\|_{\sigma'}/\|h\|_\sigma^2$ as a function of θ. The numerical estimates of this ratio never exceed 1 for any of the three kernels, meaning the inequality empirically holds for all.

Random functions with ten term expansions

The next logical step is to study functions h with more terms in their expansion. As the number of terms increases, it becomes more difficult to systematically check every such function (as we did for the two term case). Instead, we randomly sample functions. Specifically, here we study functions with ten terms centered at $\{1,\dots,10\}$: $h = \sum_{i=1}^{10} a_i k_\sigma(i,\cdot)$. For each kernel, we randomly sample 10^4 such functions by

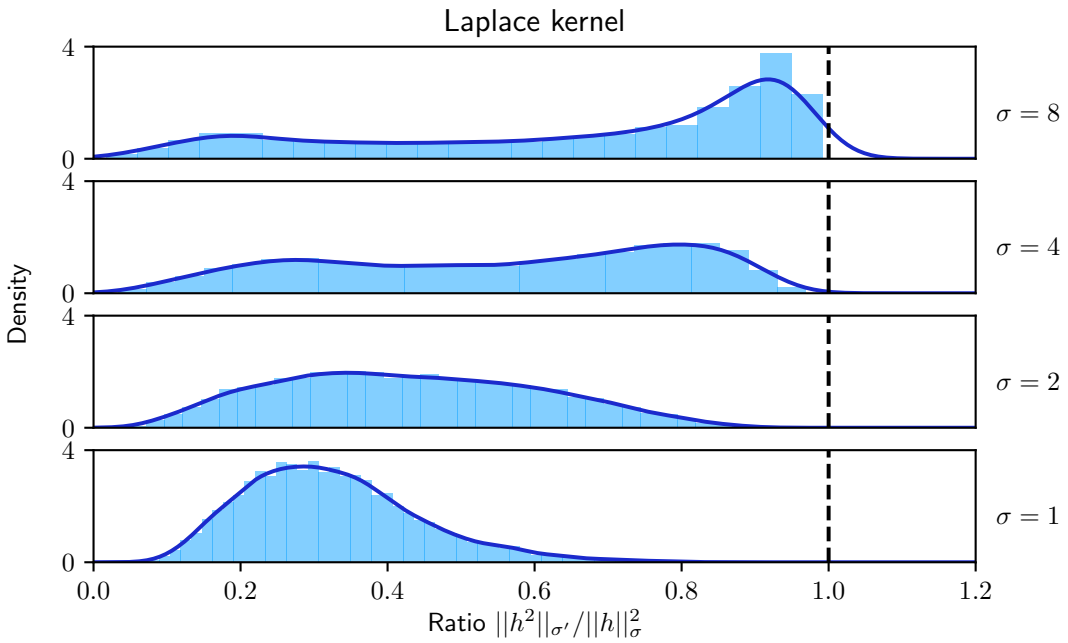


Figure 3-5: Distribution of ratio $\|h^2\|_{\sigma'}/\|h\|^2_\sigma$ for randomly sampled functions h, for Laplace kernels with different bandwidths $\sigma \in \{1, 2, 4, 8\}$.

sampling each coefficient a_i from a standard Gaussian. We repeat for several different bandwidths σ, and plot histograms of the ratio $\|h^2\|_{\sigma'}/\|h\|^2_\sigma$.

Our histograms for the Laplace kernel are displayed in Figure 3-5, and those for the Matérn kernels are in Figure 3-6. For the Laplace kernel, the maximum value of the ratio observed was about 0.99, meaning that the inequality always held. Between the two Matérn kernels, we observe a maximum ratio value of about 1.006. While this ratio is slightly greater than 1, the discrepancy could be attributed to error in our numerical estimate either of $\|h^2\|_{\sigma'}$ or of the scaling factor σ'/σ.

One notable trend present in Figures 3-5 and 3-6 is that, as the bandwidth σ increases, the distribution of the ratio $\|h^2\|_{\sigma'}/\|h\|^2_\sigma$ skews higher towards 1. In other words, $\|h^2\|_{\sigma'}$ and $\|h\|^2_\sigma$ tend to be more similar when the bandwidth σ is larger. This pattern suggests that using the new regularizer $\|h^2\|_{\sigma'}$ should result in more different behavior when σ is *small*. However, in practice, when data is limited and better regularization would matter most, *large* bandwidths σ tend to work better due to their regularizing effect. The interplay between the new regularizer and the bandwidth is an interesting direction for future study.

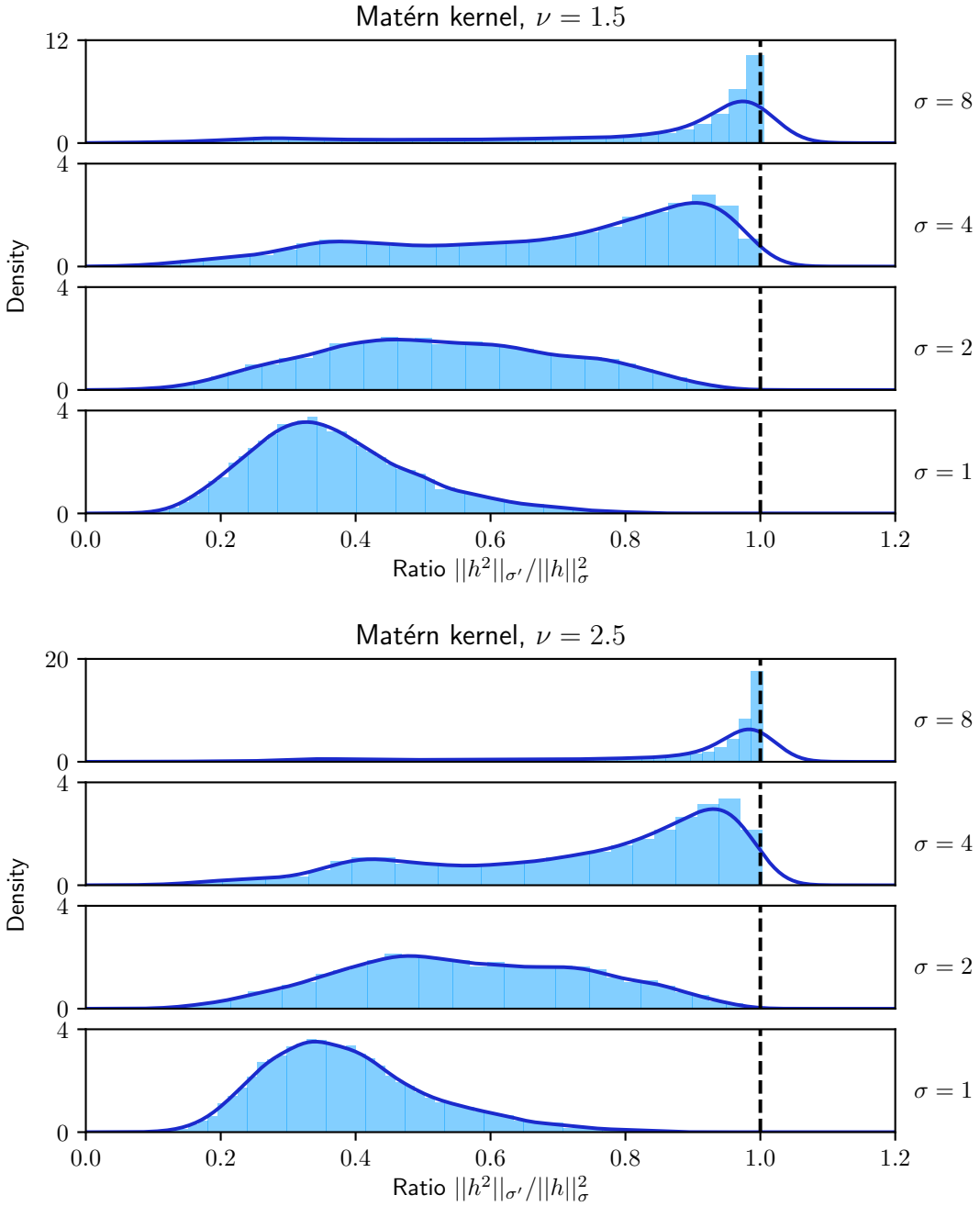


Figure 3-6: Distribution of ratio $\|h^2\|_{\sigma'}/\|h\|^2_\sigma$ for randomly sampled functions h, for Matérn kernels with different bandwidths $\sigma \in \{1, 2, 4, 8\}$. Top and bottom are plots for Matérn kernels with ν set to 1.5 and 2.5, respectively.

3.7 Discussion and future work

In this chapter we introduce MMD DRO, distributionally robust optimization with maximum mean discrepancy uncertainty sets. We prove fundamental structural results and upper bounds for MMD DRO, and unearth deep connections, in particular to Gaussian kernel ridge regression and variance regularization.

Several open questions remain. In terms of theory, our MMD DRO approach to generalization bounds leaves much new ground to explore. In particular, we conjecture that our approach might also work for ridge regression with non-Gaussian kernels; this conjecture is supported by our experiments in Section 3.6.2. Practically, there is also much left to do to make MMD DRO a general purpose tool. We have presented two approximations of MMD DRO, each with strengths and drawbacks: the upper bound in Corollary 3.3.2 enables our kernel ridge regression generalization bound, but is potentially loose, and is difficult to use more generally because the Hilbert norm is tricky to compute; the discrete approximation in Section 3.5 is more practical, and in fact has already seen application in Bayesian Optimization (Kirschner et al., 2020), but is not an upper bound on the MMD DRO problem. Future work could address these drawbacks, or potentially develop a tractable exact reformulation of the DRO problem.

Part II

Algorithms for distributionally robust subset selection

Chapter 4

Submodularity background

In work described in Part I we strengthen the bond between DRO and generalization in machine learning. Now, we take this bond as motivation to build algorithms for solving DRO problems. For convex objectives, there are classic convex reformulations available for many robust optimization problems; see e.g. (Bertsimas et al., 2011). DRO problems are more challenging because often the adversary searches over the infinite dimensional space of distributions; still, for convex objectives, there are tractable finite reformulations (Shafieezadeh Abadeh et al., 2015; Blanchet et al., 2019; Mohajerin Esfahani and Kuhn, 2018). But for non-convex problems, even when there is a tractable reformulation available e.g. (Sinha et al., 2018), it is not possible to guarantee solution quality due to non-convexity.

Instead of trying to tackle general non-convex DRO, we will focus on a special, promising subcase: DRO problems involving submodular objective functions.

4.1 Submodular set functions

Submodular set functions have natural applications in many facets of machine learning and related areas, e.g. dictionary learning (Das and Kempe, 2011), influence maximization (Kempe et al., 2003; Domingos and Richardson, 2001), data summarization (Lin and Bilmes, 2011), probabilistic modeling (Djolonga and Krause, 2014) and diversity (Kulesza and Taskar, 2012).

4.1.1 Definitions

Submodular *set functions* in particular are the most studied. Given a ground set of items V, we say a set function is a function f defined on the powerset 2^V of V. For set functions, submodularity can be captured by two equivalent definitions:

Definition 4.1.1 (Submodular set functions)**.** A set function $f : 2^V \to \mathbb{R}$ is *submodular* if, for all $S, T \subseteq V$, it holds that

$$f(S) + f(T) \geq f(S \cap T) + f(S \cup T). \tag{4.1}$$

Equivalently, f is *submodular* if, for all S and T satisfying $S \subseteq T \subseteq V$, and for all $i \in V \setminus T$, it holds that

$$f(S \cup \{i\}) - f(S) \geq f(T \cup \{i\}) - f(T). \tag{4.2}$$

Equation (4.2) is known as the *diminishing returns (DR)* property. As the set S grows, the marginal gain of adding a new item i decreases, i.e. there are diminishing returns. Additionally, if $-f$ is submodular, we say that f is *supermodular*.

4.1.2 Optimization

Many optimization problems involving submodular set functions admit efficient algorithms with theoretical guarantees on performance. Here, we briefly highlight some such guarantees, for submodular set function maximization and minimization.

Maximization

Perhaps the most common example of a submodular maximization problem is monotone submodular maximization under a cardinality constraint, i.e. $\max_{S:|S|\leq k} f(S)$. Here, f is monotone, i.e. $f(S) \leq f(T)$ whenever $S \subset T$. The greedy algorithm admits a $(1 - 1/e)$ approximation ratio, shown in classic work by Nemhauser et al. (1978). Much work has gone into speeding up the greedy algorithm in practice, e.g.

accelerated variants (Minoux, 1978), stochastic approximations (Mirzasoleiman et al., 2015) and distributed variants (Mirzasoleiman et al., 2016).

The same $(1 - 1/e)$ approximation ratio can also be attained with more general matroid constraints (Vondrak, 2008), by using a continuous version of the greedy algorithm (Calinescu et al., 2011). This continuous algorithm is more expensive but can also be accelerated, as we will review in Chapter 5. Many other submodular maximization problems also admit guarantees. For example, Feige et al. (2011) gave a deterministic 1/3 approximation for non-monotone unconstrained submodular maximization, while Buchbinder et al. (2015) gave a randomized 1/2 approximation for the same problem.

Minimization

Submodular minimization has rich connections to convex optimization. Given a set function $f : 2^V \to \mathbb{R}$, one can define an extension $\hat{f} : \mathbb{R}^{|V|} \to \mathbb{R}$, known as the Lovász extension, so that f is submodular if and only if $\hat{f}$ is convex (Lovász, 1983, Proposition 4.1). Among other foundational contributions, Edmonds (1970) showed how to efficiently compute subgradients of the Lovász extension. Grötschel et al. (1981) gave the first polynomial time algorithm for submodular set function minimization, based on applying the ellipsoid method to the Lovász extension. Combinatorial algorithms with polynomial time guarantees came later (Schrijver, 2000; Iwata et al., 2001). Recent work (Lee et al., 2015; Chakrabarty et al., 2017; Axelrod et al., 2020) has pushed the theoretical time complexity down even further.

In practice, the Fujishige-Wolfe algorithm (Wolfe, 1976; Fujishige, 2005) is a popular choice despite worse guarantees, though these have been improved recently (Chakrabarty et al., 2014). Much work has sought faster practical performance for special cases, e.g. separable problems (Jegelka et al., 2013). For further background, we recommend the monograph by Bach (2013).

4.2 General submodular functions

While submodular set functions are the best studied, the concepts of submodularity and diminishing returns (DR) can also be generalized beyond set functions. These concepts extend readily to functions on the integer lattice, and even to continuous domains. We will introduce two generalizations, namely (general) submodular and DR-submodular functions.

These generalizations are actually of great practical relevance for set function optimization. Many algorithms for submodular set function maximization actually depend on DR-submodular maximization. And many algorithms for submodular set function minimization readily extend to more general domains.

4.2.1 Definitions: submodular functions and DR functions

When we generalize beyond sets, the situation becomes more complicated, because there are multiple ways to generalize submodularity. Recall that there are two equivalent ways of defining submodular set functions: submodularity, i.e. Equation (4.1); and diminishing returns (DR), i.e. Equation (4.2). Though these coincide for set functions, for more general domains, DR is actually more restrictive than submodularity. Some optimization results apply for all (general) submodular functions, but some only hold for DR functions. We must therefore give two separate definitions.

First, we generalize submodularity, i.e. Equation (4.1). To do this, we need a suitable generalization of union and intersection. *Distributive lattices* are structures that admit such generalizations, where union ($\cup$) is replaced by join ($\vee$), and intersection ($\cap$) is replaced by meet ($\wedge$). For this thesis, we restrict our attention to domains that are subsets of the integer lattice or of $\mathbb{R}^d$, and where $x \vee y$ denotes the coordinate-wise maximum and $x \wedge y$ the coordinate-wise minimum:

Definition 4.2.1 (Submodular functions)**.** A function $f : \mathcal{X} \to \mathbb{R}$ is *submodular* if for all $x, y \in \mathcal{X}$, it holds that

$$f(x) + f(y) \geq f(x \vee y) + f(x \wedge y). \tag{4.3}$$

Note that set functions are a special case: set $\mathcal{X} = \{0,1\}^{|V|}$ and encode sets S by their indicator vectors $\mathbf{1}_S \in \mathcal{X}$. If f is defined on a subset of $\mathbb{R}^d$ and is twice-differentiable, the property of submodularity means that all off-diagonal entries of the Hessian are nonpositive, i.e., $\frac{\partial f(x)}{\partial x_i \partial x_j} \leq 0$ for all $i \neq j$ (Topkis, 1978, Theorem 3.2). These functions may be convex, concave, or neither. We will return in Chapter 6 to general submodular functions.

Second, we consider functions which instead satisfy a general version of the DR property. These are called DR-submodular functions:

Definition 4.2.2 (DR-submodular functions)**.** A function $f : \mathcal{X} \to \mathbb{R}$ is DR-submodular if, for all $x \leq y \in \mathcal{X}$, $i \in [n]$, and $\gamma > 0$ so that $x + \gamma e_i$ and $y + \gamma e_i$ are still in $\mathcal{X}$, we have $f(x + \gamma e_i) - f(x) \geq f(y + \gamma e_i) - f(y)$.

Note that DR-submodularity implies submodularity, so DR-submodular functions are a subclass of submodular functions. If f is twice-differentiable, DR-submodularity implies that *all* entries of the Hessian are nonpositive, i.e. $\frac{\partial f(x)}{\partial x_i \partial x_j} \leq 0$ for all i and j. DR-submodularity will be crucial to the optimization results in Chapter 5.

There are a number of other connections and alternate characterizations of submodular and DR-submodular functions; we recommend the thesis of Bian (2019) for more detail.

4.2.2 Optimization

Submodular functions on lattices can be minimized by a reduction to set functions, more precisely, ring families (Birkhoff, 1937). Combinatorial algorithms for submodular optimization on lattices are discussed in (Khachaturov et al., 2012). More recently, Bach (2019) extended results based on the convex Lovász extension, by building on connections to optimal transport, in order to give a continuous algorithm for general submodular minimization. Based on the connections made by Bach (2019), Axelrod et al. (2020) recently gave a faster continuous algorithm for general submodular minimization. The subclass of $L^\natural$-convex functions admits strongly polynomial time minimization (Murota, 2003; Kolmogorov and Shioura, 2009; Murota and Shioura,

2014), but does not apply to the problems discussed in this thesis.

Similarly, results for submodular maximization extend to integer lattices, e.g. (Gottschalk and Peis, 2015). Stronger results are possible for DR-submodular functions: many approximation results for the set function case extend (Bian et al., 2017; Soma and Yoshida, 2015; Soma et al., 2014). Ene and Nguyen (2016) show how to reduce DR-submodular optimization to set function optimization, for certain constraint sets.

4.3 Submodular DRO

As detailed above, submodular functions can be efficiently optimized in many cases, despite their potential non-convexity. We take this as inspiration towards studying DRO problems with submodular objectives as a first step towards general non-convex DRO. Accordingly, we present below a prototypical distributionally robust submodular optimization problem:

$$\text{(Submodular DRO)} \qquad \sup_{S} \inf_{\mathbb{Q} \in \mathcal{U}} \mathbb{E}_{z \sim \mathbb{Q}}[f(S, z)]. \tag{4.4}$$

Problem (4.4) is just one possible variation: as there are many tractable submodular optimization problems, so too are there many submodular DRO problems worthy of study. We might focus on discrete or continuous domains, or maximization or minimization. The problems we study in Chapters 5 and 6 represent just a small fraction of these.

However, it is worth emphasizing that, while many submodular optimization problems are tractable, it need not follow that submodular DRO problems are too. Submodular DRO problems can be substantially more challenging. One way to understand why is that, unlike convexity, submodularity is not in general preserved under pointwise maximum or minimum. Moreover, submodular DRO is a special case of robust submodular optimization, which in some cases is known to be hard to approximate (Krause et al., 2008a). In spite of these challenges, many positive results exist

for robust and risk-averse submodular optimization; we review some of these below.

4.3.1 Robust and risk-averse submodular optimization

Robust submodular optimization

Most work on robust submodular optimization has focused on maximization, where we seek a set S, perhaps subject to some constraints, that solves:

$$\max_{S} \min_{f \in \mathcal{F}} f(S). \tag{4.5}$$

An adversary chooses a submodular objective function f from a set $\mathcal{F}$, which is typically finite. Though this problem is hard to approximate to any polynomial factor of $|V|$ (Krause et al., 2008a), positive results are possible with appropriate relaxations. In particular, there are many algorithms for choosing a near-optimal *distribution* of subsets S (Krause et al., 2011; Chen et al., 2017; Anari et al., 2019; Wilder, 2018a).

Moreover, the above worst-case results due to Krause et al. (2008a) may not apply when there is more structure in the adversary's set $\mathcal{F}$ of candidate functions. One example is *deletion-robust submodular maximization*, where we wish to solve:

$$\max_{S:|S| \leq k} \min_{T \subset S:|T| \leq t} f(S \setminus T). \tag{4.6}$$

Here the adversary can only delete elements from the chosen set S. Studying this special case separately is crucial, as it actually admits efficient algorithms with constant factor approximation guarantees (Orlin et al., 2016; Bogunovic et al., 2017; Mirzasoleiman et al., 2016; Kazemi et al., 2018). Another specific problem that has attracted attention is robust influence maximization (Chen et al., 2016; He and Kempe, 2016; Lowalekar et al., 2016; Kalimeris et al., 2019).

Risk-averse submodular optimization

Besides robust optimization, there are other approaches to encourage risk-aversion in stochastic decision making. For example, instead of optimizing an expectation, one might optimize value-at-risk (VaR) or conditional value-at-risk (CVaR) (Rockafellar and Uryasev, 2000, 2002), which capture tail performance. While risk-averse versions of modular (linear) set function optimization admit tractable optimization, in particular for robust (Bertsimas and Sim, 2003) and CVaR (Nikolova, 2010) variants, submodular objectives do not in general admit such optimization (Maehara, 2015). However, variants and relaxations of submodular CVaR problems admit approximations (Ohsaka and Yoshida, 2017; Wilder, 2018b).

4.3.2 Submodular optimization with errors

Beyond robust and risk-averse formulations of submodular problems, other work has focused on the sensitivity of submodular optimization algorithms to noise and errors. Hassidim and Singer (2017) give positive results for submodular maximization with stochastic errors, but also prove hardness in the case of adversarial errors. Balkanski et al. (2017) prove hardness results for general submodular maximization in a more difficult setting, where we see the function value only on randomly sampled subsets. Conversely, Balkanski et al. (2016) give positive results for this problem for functions with bounded curvature. There is also work studying submodular minimization with errors (Halabi and Jegelka, 2019; Ito, 2019).

Chapter 5

Distributionally robust submodular maximization

5.1 Introduction

In Chapter 4 we have seen that submodular functions have wide application and have good properties for optimization. Most of these results are for generic submodular functions that admit an evaluation oracle. However, in many settings the submodular function we wish to optimize has additional structure, which may present both challenges and an opportunity to do better.

In particular, we focus on the problem of *stochastic submodular optimization* (SSO), which has recently gained much attention. In SSO, we wish to choose a set S, subject to e.g. a cardinality constraint $|S| \leq k$, in order to maximize $f_{\mathbb{P}}(S) := \mathbb{E}_{f\sim\mathbb{P}}[f(S)]$ for some distribution $\mathbb{P}$. Note that SSO is a particular instantiation of the fundamental problem of stochastic optimization discussed in Chapter 1. Stochastic submodular optimization encompasses many problems, e.g. influence maximization and facility location. Much recent work has focused on more computationally efficient gradient-based algorithms for SSO (Karimi et al., 2017; Mokhtari et al., 2018a; Hassani et al., 2017; Karbasi et al., 2019; Zhang et al., 2019). However, none of this work accounts for uncertainty in the distribution $\mathbb{P}$. We typically lack direct access to $\mathbb{P}$, and instead may have only a fixed set of samples $f_1, \ldots, f_n$ from $\mathbb{P}$ that form

an empirical distribution $\hat{\mathbb{P}}_n$. Prior work on SSO does not address this gap.

The problem we consider is: how can we, given only $\hat{\mathbb{P}}_n$, efficiently select a subset S that has good performance on $\mathbb{P}$? Or, more generally, how can we solve SSO problems given obscured or perturbed access to $\mathbb{P}$? This challenge strongly resembles statistical learning, as studied e.g. in Chapter 3. Work on SSO so far has largely ignored this aspect of SSO, and instead focused on optimizing the empirical estimate $f_{\hat{\mathbb{P}}_n} = \frac{1}{n}\sum_{i=1}^n f_i$, akin to empirical risk minimization. While this works well when n is large and $f_{\hat{\mathbb{P}}_n} \approx f_{\mathbb{P}}$, for small n, it may be possible to do better. In statistical learning, when n is small, one can often achieve better performance on the population $\mathbb{P}$ by regularizing the model (analagous in this setting to the decision S). However, it is not obvious how to select a regularizer in this setting. In statistical learning, norm penalties (e.g. the norm of the weights of a linear model) are perhaps the most common regularizer. But it is not clear how to port those over to our SSO setting, since our decision S is already constrained: $|S| \leq k$.

The astute reader may already (correctly) suspect that we will propose to use DRO to protect against this gap between $\mathbb{P}$ and $\hat{\mathbb{P}}_n$. For generic stochastic optimization, we have strongly advocated in Chapters 2 and 3 for DRO as a way of dealing with such perturbations. It especially makes sense for SSO, since it is not clear how otherwise to regularize the problem. Of the DRO options available, DRO with χ^2 uncertainty sets seems the most sensible. The other alternatives, namely Wasserstein and MMD, would force us to develop geometry on the space of submodular functions f; this may be fruitful for special cases, but our focus here is general SSO.

DRO with χ^2 uncertainty sets is especially popular due to its connection with variance regularization (per section 2.2). By variance regularization, we mean that we use the variance of $f(S)$ on $\hat{\mathbb{P}}_n$ as a data-dependent regularizer, and optimize $f_{\hat{\mathbb{P}}_n}(S) - C_1\sqrt{\mathrm{Var}_{\hat{\mathbb{P}}_n}(f(S))/n}$. In studying SSO, both the DRO viewpoint and the variance regularization viewpoint will be crucial, because both are intuitive but also challenging. Variance regularization is easy to motivate: when the variance is high, it dominates a standard high-probability lower bound on the population performance $f_{\mathbb{P}}(S)$, which is the quantity we actually want to optimize. However, it seems difficult

to give optimization guarantees for the variance regularized problem, since even if all f_i are submodular, their variance need not be (see Fact 5.2.1; Fact 6.2.1 in the next chapter is also related). On the other hand, for DRO, we have provided substantial motivation in the preceding chapters. However, a priori, the DRO reformulation of variance regularization could also be difficult, due to the hardness of robust submodular maximization discussed in Section 4.3.1.

In this chapter we show, perhaps surprisingly, that variance-regularized submodular maximization is both tractable and scalable. We give a theoretically-backed algorithm for distributionally robust submodular optimization with χ^2 uncertainty sets. Our algorithm substantially improves over a naive application of previous approaches for robust submodular problems. Along the way, we develop improved technical results for general (non-submodular) DRO problems, including both improved algorithmic tools and more refined structural characterizations of the problem. For instance, we give a more complete characterization of the relationship between χ^2 DRO and variance regularization. We verify empirically that in many real-world settings, variance regularization enables better generalization from fixed samples of a stochastic submodular function, particularly when the variance is high.

5.1.1 Related work

We build on and significantly extend a recent line of research in statistical learning and optimization that develops a relationship between distributional robustness and variance-based regularization (Maurer and Pontil, 2009; Gotoh et al., 2018; Lam, 2016; Duchi et al., 2016; Namkoong and Duchi, 2017). While previous work has uniformly focused on the continuous (and typically convex) case, here we address *combinatorial* problems with submodular structure, requiring further technical developments. As a byproduct, we better characterize the behavior of the DRO problem under low sample variance (which was left open in previous work), show conditions under which the DRO problem becomes smooth, and provide improved algorithmic tools which apply to general DRO problems.

Another related area is robust submodular optimization, which we discuss in Sec-

tion 4.3.1. Existing work aims to maximize the minimum of a set of submodular functions, but does not address the *distributionally* robust optimization problem where an adversary perturbs the empirical distribution. We develop scalable algorithms, accompanied by approximation guarantees, for this case. Our algorithms improve both theoretically and empirically over naive application of previous robust submodular optimization algorithms to DRO. Further, our work is motivated by the connection between distributional robustness and generalization in learning, which has not previously been studied for submodular functions. Stan et al. (2017) study generalization in a related combinatorial problem, but they do not explicitly balance bias and variance, and the goal is different: they seek a smaller ground set which still contains a good subset for each user in the population.

A complementary line of work concerns *stochastic* submodular optimization (Karimi et al., 2017; Mokhtari et al., 2018a; Hassani et al., 2017; Karbasi et al., 2019; Zhang et al., 2019) that, as opposed to our setting, requires a sampling oracle for the underlying function. We draw from stochastic optimization tools, but assume only a fixed dataset is available.

Our motivation also relates to optimization from samples, where we have access to values of a fixed unknown function on inputs sampled from a distribution. Balkanski et al. (2017, 2016) prove hardness results for general submodular maximization from samples, with positive results for functions with bounded curvature. We address a different model where the underlying function itself is stochastic and we observe realizations of it. Hence, it is possible to well-approximate the optimization problem from polynomially many samples. The challenge is to construct algorithms that make more effective use of data.

5.2 Stochastic submodular functions and distributional robustness

Refer to Chapter 4 for background on submodular functions and optimization. We call a set function $f : 2^V \to \mathbb{R}$ *monotone* if $S \subseteq T$ implies $f(S) \leq f(T)$. Let $\mathbb{P}$ be a distribution over monotone submodular functions f. We assume that each function is normalized and bounded, i.e., $f(\emptyset) = 0$ and $f(S) \in [0, B]$ almost surely for all subsets S. We seek a subset S that maximizes

$$f_{\mathbb{P}}(S) := \mathbb{E}_{f \sim \mathbb{P}}[f(S)] \tag{5.1}$$

subject to some constraints, e.g., $|S| \leq k$. We call the function $f_{\mathbb{P}}(S)$ a *stochastic submodular function*. Such functions arise in many domains; we begin with two specific motivating examples.

5.2.1 Stochastic submodular functions

Influence Maximization. Consider a graph $G = (V, E)$ on which influence propagates. We seek to choose an initial seed set $S \subseteq V$ of influenced nodes to maximize the expected number subsequently reached. Each edge can be either active, meaning that it can propagate influence, or inactive. A node is influenced if it is reachable from S via active edges. Common diffusion models specify a distribution of active edges, e.g., the Independent Cascade Model (ICM), the Linear Threshold Model (LTM), and generalizations thereof. Regardless of the specific model, each can be described by the distribution of "live-edge graphs" induced by the active edges $\mathcal{E}$ (Kempe et al., 2003). Hence, the expected number of influenced nodes $f(S)$ can be written as an expectation over live-edge graphs: $f_{\text{IM}}(S) = \mathbb{E}_{\mathcal{E}}[f(S; \mathcal{E})]$. The distribution over live-edge graphs induces a distribution $\mathbb{P}$ over functions f as in equation (5.1).

Facility Location. Fix a ground set V of possibile facility locations j. Suppose we have a (possibly infinite as in (Stan et al., 2017)) number of demand points i

drawn from a distribution $\mathcal{D}$. For example, each i may correspond to a user sampled from a population $\mathcal{D}$. The goal of *facility location* is to choose a subset $S \subset V$ that covers the demand points as well as possible. Each demand point i is equipped with a vector $r^i \in \mathbb{R}^{|V|}$ describing how well point i is covered by each facility j. We wish to maximize: $f_{\text{facloc}}(S) = \mathbb{E}_{i\sim\mathcal{D}}\left[\max_{j\in S} r^i_j\right]$. Each $f(S) = \max_{j\in S} r_j$ is submodular, and $\mathcal{D}$ induces a distribution P over the functions $f(S)$ as in equation (5.1).

5.2.2 Optimization and empirical approximation

Two main issues arise with stochastic submodular functions. First, simple techniques such as the greedy algorithm become impractical since we must accurately compute marginal gains. Recent alternative algorithms (Karimi et al., 2017; Mokhtari et al., 2018a; Hassani et al., 2017) make use of additional, specific information about the function, such as efficient gradient oracles for the multilinear extension. A second issue has so far been neglected: the degree of access we have to the underlying function (and its gradients). In many settings, we only have access to a limited, fixed number of samples, either because these samples are given as observed data or because sampling the true model is computationally prohibitive.

Formally, instead of the full distribution $\mathbb{P}$, we have access to an empirical distribution $\hat{\mathbb{P}}_n$ composed of n samples $f_1, \ldots, f_n \sim \mathbb{P}$. One approach is to optimize

$$f_{\hat{\mathbb{P}}_n} = \mathbb{E}_{f\sim\hat{\mathbb{P}}_n}[f(S)] = \frac{1}{n}\sum_{i=1}^{n} f_i(S), \tag{5.2}$$

and hope that $f_{\hat{\mathbb{P}}_n}$ adequately approximates $f_{\mathbb{P}}$. This is guaranteed when n is sufficiently large. E.g., in influence maximization, for $f_{\hat{\mathbb{P}}_n}(S)$ to approximate $f_{\mathbb{P}}(S)$ within error ε with probability $1-\delta$, Kempe et al. (2015) show that $O\left(\frac{|V|^2}{\varepsilon^2}\log\frac{1}{\delta}\right)$ samples suffice. To our knowledge, this is the tightest general bound available. Still, it easily amounts to thousands of samples even for small graphs; in many applications we would not have so much data.

The problem of maximizing $f_{\mathbb{P}}(S)$ from samples greatly resembles statistical learn-

ing. Namely, if the f_i are drawn iid from $\mathbb{P}$, then we can write

$$f_{\mathbb{P}}(S) \geq f_{\hat{\mathbb{P}}_n}(S) - C_1\sqrt{\frac{\mathrm{Var}_{\mathbb{P}}\left(f(S)\right)}{n}} - \frac{C_2}{n} \tag{5.3}$$

for each S with high probability, where C_1 and C_2 are constants that depend on the problem. For instance, if we want this bound to hold with probability $1-\delta$, then applying the Bernstein bound (see Appendix B.1) yields $C_1 \leq \sqrt{2\log\frac{1}{\delta}}$ and $C_2 \leq \frac{2B}{3}\log\frac{1}{\delta}$ (recall B is an upper bound on $f(S)$). Given that we have only finite samples, it would then be logical to directly optimize

$$f_{\hat{\mathbb{P}}_n}(S) - C_1\sqrt{\mathrm{Var}_{\hat{\mathbb{P}}_n}\left(f(S)\right)/n}, \tag{5.4}$$

where $\mathrm{Var}_{\hat{\mathbb{P}}_n}$ refers to the empirical variance over the sample. This would allow us to directly optimize the tradeoff between bias and variance. However, even when each f is submodular, the variance-regularized objective (5.4) need not be:

Fact 5.2.1. There exists a distribution $\mathbb{P}$ of functions f so that each $f \sim \mathbb{P}$ is submodular, but neither $-\mathrm{Var}_{\mathbb{P}}(f(S))$ nor $-\sqrt{\mathrm{Var}_{\mathbb{P}}(f(S))}$ are submodular.

Proof. Our counterexample uses a uniform distribution over two such functions $g, h : 2^V \to \mathbb{R}$. In particular, we choose $h = 0$. Then the variance of the uniform distribution over g and h is given by:

$$\begin{aligned}
\mathrm{Var}_{\mathbb{P}}(f(S)) &= \mathbb{E}_{f\sim\mathbb{P}}[f(S)^2] - \mathbb{E}_{f\sim\mathbb{P}}[f(S)]^2 \\
&= \frac{1}{2}g(S)^2 + \frac{1}{2}\cdot 0^2 - \left(\frac{1}{2}g(S) + \frac{1}{2}\cdot 0\right)^2 \\
&= \frac{1}{4}g(S)^2.
\end{aligned}$$

Since submodularity is invariant to positive rescaling, it suffices to study g^2 in lieu of $\mathrm{Var}_{\mathbb{P}}(f)$ and $|g|$ in lieu of $\sqrt{\mathrm{Var}_{\mathbb{P}}(f)}$. In other words, it suffices to find a submodular function g so that neither $-g^2$ nor $-|g|$ are submodular. For $w = (1, -1)$, the modular function $g(S) = \sum_{i\in S} w_i$ will do the trick, defined on the ground set $V = \{0, 1\}$. The function g is modular and therefore also submodular. However, $|g|$ is not submodular,

because

$$-|g(\{0,1\})| + |g(\{0\})| = 1 \not\leq -1 = -|g(\{1\})| + |g(\emptyset)|. \tag{5.5}$$

Moreover, since the range of $g(S)$ is $\{-1, 0, 1\}$, we have $g(S)^2 = |g(S)|$, so $-g^2$ is not submodular either. □

5.2.3 Variance regularization via distributionally robust optimization

While the optimization problem (5.4) is not directly solvable via submodular optimization, we will see next that distributionally robust optimization (DRO) can help provide a tractable reformulation. In DRO, we seek to optimize our function in the face of an adversary who perturbs the empirical distribution within an uncertainty set $\mathcal{P}$:

$$\max_{S} \min_{\mathbb{Q} \in \mathcal{P}} \mathbb{E}_{f \sim \mathbb{Q}}[f(S)]. \tag{5.6}$$

We focus on the case when the adversary set $\mathcal{P}$ is a χ^2 ball around the empirical distribution:

Definition 5.2.1. The χ^2 divergence between distributions $\mathbb{P}$ and $\tilde{\mathbb{P}}$ is $D_\phi(\mathbb{P}||\tilde{\mathbb{P}}) = \frac{1}{2}\int \left(d\mathbb{P}/d\tilde{\mathbb{P}} - 1\right)^2 d\tilde{\mathbb{P}}$. The χ^2 uncertainty set around an empirical distribution $\hat{\mathbb{P}}_n$ is $\mathcal{P}_{\rho,n} = \{\mathbb{Q} : D_\phi(\mathbb{Q}||\hat{\mathbb{P}}_n) \leq \rho/n\}$. When $\hat{\mathbb{P}}_n$ corresponds to an empirical sample $Z_1, \ldots, Z_n$, we encode $\mathbb{Q}$ by a vector p in the simplex Δ_n and equivalently write $\mathcal{P}_{\rho,n} = \left\{p \in \Delta_n : \frac{1}{2}\|np - \mathbf{1}\|_2^2 \leq \rho\right\}$.

In particular, maximizing the variance-regularized objective (5.4) is equivalent to solving a distributionally robust problem when the sample variance is high enough. The intuition behind this equivalence is that the χ^2 ball is a quadratic ball in the simplex, and the variance penalty is also quadratic. More formally:

Theorem 5.2.1 (modified from (Namkoong and Duchi, 2017))**.** *Fix $\rho \geq 0$, and let $Z \in [0, B]$ be a random variable (i.e. $Z = f(S)$). Write $s_n^2 = \mathrm{Var}_{\hat{\mathbb{P}}_n}(Z)$ and let*

$OPT = \inf_{\mathbb{Q}\in\mathcal{P}_{\rho,n}} \mathbb{E}_{\mathbb{Q}}[Z]$. *Then*

$$\max\left\{0, \sqrt{\frac{2\rho}{n}s_n^2} - \frac{2B\rho}{n}\right\} \leq \mathbb{E}_{\hat{\mathbb{P}}_n}[Z] - OPT \leq \sqrt{\frac{2\rho}{n}s_n^2}.$$

Moreover, if $s_n^2 \geq 2\rho(\max_i z_i - \overline{z}_n)^2/n$, *then* $OPT = \mathbb{E}_{\hat{\mathbb{P}}_n}[Z] - \sqrt{2\rho s_n^2/n}$, *i.e.,* χ^2*-DRO is exactly equivalent to variance regularization.*

In several settings, Namkoong and Duchi (2017) show this holds with high probability, by requiring high population variance $\mathrm{Var}_{\mathbb{P}}(Z)$ and applying concentration results to show the empirical variance $\mathrm{Var}_{\hat{\mathbb{P}}_n}(Z)$ is high enough. While Theorem 5.2.1 is a direct port from the convex setting, the corresponding high probability result for submodular functions is more involved:

Lemma 5.2.1. *Fix parameters* δ, ρ, $|V|$ *and* $k \geq 1$, *and define the constant* M *by:*

$$M = \max\left\{\sqrt{32\rho/7}, \sqrt{36\left(\log\left(1/\delta\right) + |V|\log(1+24k)\right)}\right\}.$$

For all S *with* $|S| \leq k$ *and* $\mathrm{Var}_{\mathbb{P}}(f_{\mathbb{P}}(S)) \geq \frac{B}{\sqrt{n}}M$, χ^2*-DRO is exactly equivalent to variance regularization with combined probability at least* $1-\delta$.

This result is obtained as a byproduct of a more general argument that applies to all points in a fractional relaxation of the submodular problem (see Appendix B.2) and shows equivalence of the two problems when the population variance is sufficiently high. However, it is not clear what the DRO problem yields when the sample variance is too small. We give a more precise characterization of how the DRO problem behaves under arbitrary variance:

Lemma 5.2.2. *Let* $\rho < n(n-1)/2$. *Suppose all* $z_1, \ldots, z_n$ *are distinct, with* $z_1 < \cdots < z_n$. *Define* $\alpha(m,n,\rho) = 2\rho m/n^2 + m/n - 1$, *and let* $\mathcal{I} = \{m \in \{1,\ldots,n\} :$

$\alpha(m,n,\rho) > 0\}$. *Then,* $\inf_{\mathbb{Q}\in\mathcal{P}_{\rho,n}} \mathbb{E}_{\mathbb{Q}}[Z]$ *is equal to*

$$\min_{m\in\mathcal{I}} \left\{ \overline{z}_m - \min\left\{ \sqrt{\alpha(m,n,\rho)s_m^2},\ \frac{s_m^2}{z_m - \overline{z}_m} \right\} \right\} \\ \leq \mathbb{E}_{\hat{\mathbb{P}}_n}[Z] - \min\left\{ \sqrt{\frac{2\rho s_n^2}{n}},\ \frac{s_n^2}{z_n - \overline{z}_n} \right\},$$

where $\hat{\mathbb{P}}_m$ *denotes the uniform distribution on* $z_1,\ldots,z_m$, $\overline{z}_m = \mathbb{E}_{\hat{\mathbb{P}}_m}[Z]$, *and* $s_m^2 = \mathrm{Var}_{\hat{\mathbb{P}}_m}(Z)$.

The inequality holds since n is always in $\mathcal{I}$ and $\alpha(n,n,\rho) = 2\rho/n$. As in Theorem 5.2.1, when the variance $s_n^2 \geq 2\rho/n \cdot (z_n - \overline{z}_n)^2$, we recover the exact variance expansion. We show Lemma 5.2.2 by developing an exact algorithm for linear optimization over the χ^2 ball, which we overview in the next section.

Finally, we apply the equivalence of DRO and variance regularization to obtain a surrogate optimization problem. Fix the set S, and let Z be the random variable induced by $f(S)$ with $f \sim \mathbb{P}$. Theorem 5.2.1 in this setting suggests that instead of directly optimizing equation (5.4), we can instead solve

$$\max_S \min_{\mathbb{Q}\in\mathcal{P}_{\rho,n}} \mathbb{E}_{f\sim\mathbb{Q}}[f(S)] = \max_S \min_{p\in\mathcal{P}_{\rho,n}} \sum_{i=1}^n p_i f_i(S). \tag{5.7}$$

We will return to this problem in Section 5.4. First, though, we discuss how to solve the DRO adversary's problem efficiently and exactly.

5.3 Exact algorithm for χ^2-DRO

In this section we show how to construct an $O(n\log n)$ time *exact* oracle for linear optimization in the χ^2 ball:

$$\begin{array}{ll} \min_p & \langle z, p\rangle \\ \text{s.t.} & \frac{1}{2}\|np - \mathbf{1}\|_2^2 \leq \rho \\ & \mathbf{1}^T p = 1 \\ & p_i \geq 0,\ i = 1,\ldots,n. \end{array} \tag{5.8}$$

Without loss of generality, assume $z_1 \leq z_2 \leq \cdots \leq z_n$. This can be done by sorting in $O(n \log n)$ time.

First, we wish to discard the case where the χ^2 constraint is not tight:

Lemma 5.3.1. *Let ℓ be the largest integer so that $z_1 = z_\ell$, i.e. $z_1 = \cdots = z_\ell < z_{\ell+1}$. If $\rho \geq n(n-\ell)/(2\ell)$, then problem (5.8) is solved by $p_i^\star = 1/\ell$ for $i = 1, \ldots, \ell$. Otherwise, the χ^2 (quadratic) constraint must be tight.*

The proof for Lemma 5.3.1 and all other Lemmas in this section can be found in section B.3 in the appendix. Before proceeding, we define several auxiliary variables which can all be computed from the problem data in $O(n)$ time:

$$\overline{z}_j = \sum_{i=1}^{j} z_i, \; j = 1, \ldots, n \tag{5.9}$$

$$b_j = \sum_{i=1}^{j} z_i^2, \; j = 1, \ldots, n \tag{5.10}$$

$$s_j^2 = \frac{b_j}{j} - (\overline{z}_j)^2, \; j = 1, \ldots, n. \tag{5.11}$$

Note that $\overline{z}_j$ and s_j^2 are the mean and variance of $\{z_1, \ldots, z_j\}$. Now, we begin by writing down the Lagrangian of problem (5.8):

$$\mathcal{L}(p, \lambda, \theta, \eta) = \langle z, p \rangle + \lambda \left(\frac{1}{2} \|np - \mathbf{1}\|_2^2 - \rho \right) + \theta \left(\sum_{i=1}^{n} p_i - 1 \right) - \langle \eta, p \rangle, \tag{5.12}$$

with dual variables $\lambda \in \mathbb{R}_+$, $\theta \in \mathbb{R}$, and $\eta \in \mathbb{R}_+^n$. By the KKT conditions, an optimal assignment $p^\star, \lambda^\star, \theta^\star, \eta^\star$ must satisfy

$$0 = \nabla_p \mathcal{L}(p^\star, \lambda^\star, \theta^\star, \eta^\star) = z + \lambda^\star n(np^\star - \mathbf{1}) + \theta^\star \mathbf{1} - \eta^\star, \tag{5.13}$$

or, equivalently,

$$\lambda^\star n^2 p_i^\star = \lambda^\star n - z_i - \theta^\star + \eta_i^\star. \tag{5.14}$$

By complementary slackness, either $\eta_i^\star > 0$, in which case $p_i^\star = 0$, or $\eta_i^\star = 0$ and

$$\lambda^\star n^2 p_i^\star = \lambda^\star n - z_i - \theta^\star. \tag{5.15}$$

Since $z_1 \leq \cdots \leq z_n$, it follows that $p_i^\star$ decreases as i increases until eventually $p_i^\star = 0$. Hence there exists m so that for $i = 1, \ldots, m$ we have $p_i^\star > 0$ and thereafter $p_i^\star = 0$. Combining this observation with equation (5.15), we can solve for $p^\star$:

$$p_i^\star = \left(1 - \frac{(z_i + \theta^\star)}{\lambda^\star n}\right) \cdot \frac{1}{n} \text{ for } i = 1, \ldots, m, \tag{5.16}$$

$$\text{and } p_i^\star = 0 \text{ otherwise.} \tag{5.17}$$

Note we can divide by $\lambda^\star$ as we have already determined via Lemma 5.3.1 that the corresponding constraint is tight (and therefore $\lambda^\star > 0$).

It is challenging to compute the value of m in closed form, but fortunately we do not need to. We can instead search for the best choice of m. For each guess of m, we can compute in closed form the corresponding optimal values of $p^\star, \lambda^\star, \theta^\star$ and $\eta^\star$. There are only n possible choices for m, and each can be checked in constant time; at the end, we can simply pick the best one. As such, for the rest of the results in this section, we can fix m and assume it is optimal.

Our first step is to solve for $\theta^\star$ as a function of $\lambda^\star$:

Lemma 5.3.2. *Suppose that $\lambda^\star$ is optimal. Then the optimal value of $\theta^\star$ is:*

$$\theta^\star = \left(1 - \frac{n}{m}\right) \lambda^\star n - \overline{z}_m. \tag{5.18}$$

Since we have solved for $\theta^\star$ as a function of the other variables, we can eliminate it and express $p^\star$ purely as a function of $\lambda^\star$. As a consequence, we can derive the objective value in terms of $\lambda^\star$:

Lemma 5.3.3. *Let $\lambda^\star$ be optimal. The optimal solution $p^\star$ obtains the objective value*

$$\langle z, p^\star \rangle = \overline{z}_m - \frac{m s_m^2}{\lambda^\star n^2}.$$

Algorithm 1 Linear optimization in χ^2 ball

Input: pre-sorted vector z with $z_1 \le \cdots \le z_n$
Output: optimal vector p

▷ Check if χ^2 constraint is tight
Compute maximum ℓ s.t. $z_1 = z_\ell$
if $n(n-\ell)/(2\ell) \le \rho$ **then**
  return p with $p_i = 1/\ell$ for $i \le \ell$ and $p_i = 0$ otherwise
end if

▷ Since χ^2 constraint is tight, now we search for optimal m
$\overline{z}_j \leftarrow \frac{1}{j}\sum_{i=1}^{j} z_i,\ j = 1, \ldots, n$
$b_j \leftarrow \sum_{i=1}^{j} z_i^2,\ j = 1, \ldots, n$
$s_j^2 \leftarrow b_j/j - (\overline{z}_j)^2,\ j = 1, \ldots, n$
$m_{\min} \leftarrow \min\{m \in \{1, \ldots, n\} : \alpha(m, n, \rho) > 0\}$
$\lambda_m = \frac{1}{n^2} \cdot \max\left\{\sqrt{\frac{m^2 s_m^2}{\alpha(m,n,\rho)}},\ (z_m - \overline{z}_m)m\right\},\ m = m_{\min}, \ldots, n$
$v_m \leftarrow \overline{z}_m - m s_m^2/(\lambda_m n^2),\ m = m_{\min}, \ldots, n$
$m_{\text{opt}} \leftarrow \operatorname{argmin}_m\{v_m : m = m_{\min}, \ldots, n\}$
$\theta \leftarrow \left(1 - \frac{n}{m_{\text{opt}}}\right)\lambda_{m_{\text{opt}}} n - \overline{z}_{m_{\text{opt}}}$
return $p = \frac{1}{n}\max\left(0, 1 - \frac{z_{m_{\text{opt}}} + \theta}{\lambda_{m_{\text{opt}}} n}\right)$

Since $ms_m^2 \ge 0$, $\lambda^\star$ will be the minimum value of λ such that the induced p is still feasible. Since the $\mathbf{1}^T p = 1$ constraint is guaranteed by the choice of $\theta^\star$, to compute $\lambda^\star$ we need only check the χ^2 and nonnegativity constraints. As a byproduct, we derive a simple check to help prune out suboptimal m:

Lemma 5.3.4. *If $\alpha(m, n, p) > 0$, the optimal feasible $\lambda^\star$ for this value of m is given by*

$$\lambda^\star = \frac{1}{n^2} \cdot \max\left\{\sqrt{\frac{m^2 s_m^2}{\alpha(m, n, \rho)}},\ m(z_m - \overline{z}_m)\right\}. \tag{5.19}$$

Otherwise, if $\alpha(m, n, p) \le 0$, then m cannot be optimal.

With Lemmas 5.3.3 and 5.3.4 in hand, our strategy is clear: for each m with $\alpha(m, n, \rho) > 0$, we compute the optimal dual variables, and then use them to compute the objective value. At the end, we choose the best value of m, and compute $p^\star$ via equation (5.16). Our algorithm is given more formally in Algorithm 1.

5.4 Algorithmic approach

Even though each $f_i(\cdot)$ is submodular, it is not obvious how to solve Problem (5.7): robust submodular maximization is in general inapproximable, i.e. no polynomial-time algorithm can guarantee a positive fraction of the optimal value unless P = NP (Krause et al., 2008b). Recent work has sought tractable relaxations (Staib and Jegelka, 2017b; Krause et al., 2008b; Wilder, 2018a; Anari et al., 2019; Orlin et al., 2016; Bogunovic et al., 2017), but these either do not apply or yield much weaker results in our setting. We consider a relaxation of robust submodular maximization that returns a near-optimal *distribution* over subsets S (as in (Chen et al., 2017; Wilder, 2018a)). That is, we solve the robust problem $\max_{\mathcal{D}} \min_{i\in[m]} \mathbb{E}_{S\sim\mathcal{D}}[f_i(S)]$ where $\mathcal{D}$ is a distribution over sets S. It is not immediately clear how to represent a distribution over exponentially many subsets. We will later see that optimizing a product distribution (i.e. via the multilinear extension) is enough. Our strategy, based on "continuous greedy" ideas, extends the set function f to a continuous function F, then maximizes a robust problem involving F via continuous optimization.

Multilinear extension. One canonical extension of a submodular function f to the continuous domain is the *multilinear extension*. The multilinear extension $F : [0,1]^{|V|} \to \mathbb{R}$ of f is defined as $F(x) = \sum_{S\subseteq V} f(S) \prod_{i\in S} x_i \prod_{j\notin S}(1-x_j)$. That is, $F(x)$ is the expected value of $f(S)$ when each item i in the ground set is included in S independently with probability x_i. A crucial property of F (that we later return to) is that it is a continuous *DR-submodular* function, in the sense of Definition 4.2.2.

Efficient algorithms are available for maximizing DR-submodular functions over convex sets (Calinescu et al., 2011; Feldman et al., 2011; Bian et al., 2017). Specifically, we take $\mathcal{X}$ to be the convex hull of the indicator vectors of feasible sets. The robust continuous optimization problem we wish to solve is then

$$\max_{x\in\mathcal{X}} \min_{p\in\mathcal{P}_{\rho,n}} \sum_{i=1}^{n} p_i F_i(x). \tag{5.20}$$

It remains to address two questions: (1) how to efficiently solve Problem (5.20) –

existing algorithms only apply to the max, not the maximin version – and (2) how to then obtain a solution for Problem (5.7).

We address the former question in the next section. For the latter question, given a maximizing distribution $\mathcal{D}$ over subsets, existing techniques (e.g., swap rounding) can be used to round $\mathcal{D}$ to a deterministic subset S with no loss in solution quality (Chekuri et al., 2010). But our variable x in Problem (5.20) can only express a limited class of distributions with independent marginals $\Pr(i \in S)$, not all distributions $\mathcal{D}$. Fortunately, this discrepancy does not cost us much:

Lemma 5.4.1. *Suppose x is an α-optimal solution to Problem (5.20). Then x induces a distribution $\mathcal{D}$ over subsets so that $\mathcal{D}$ is $(1 - 1/e)\alpha$-optimal for Problem (5.7).*

Our proof involves the *correlation gap* (Agrawal et al., 2010). It is also possible to eliminate the $(1 - 1/e)$ gap altogether by using multiple copies of the decision variables to optimize over a more expressive class of distributions (Wilder, 2018a), but empirically we find this unnecessary.

Next, we address algorithms for solving Problem (5.20). Since a convex combination of submodular functions is still submodular, we can see each p as inducing a submodular function, so Problem (5.20) asks to maximize the minimum of a set of continuous submodular functions.

Frank-Wolfe algorithm and complications. In the remainder of this section, we show how Problem (5.20) can be solved with optimal approximation ratio (as in Lemma 5.4.1) by Algorithm 2, which is based on Frank-Wolfe (FW) (Frank and Wolfe, 1956; Jaggi, 2013). FW algorithms iteratively move toward the feasible point that maximizes the inner product with the gradient. Instead of a projection step, each iteration uses a single linear optimization over the feasible set $\mathcal{X}$; this is very cheap for the feasible sets we are interested in (e.g., a simple greedy algorithm for matroid polytopes). Indeed, FW is currently the best approach for maximizing DR-submodular functions in many settings. While there are FW algorithms designed for convex-concave games (Gidel et al., 2017), it is not possible to directly apply these to the submodular setting while maintaining approximation guarantees.

Algorithm 2 Momentum Frank-Wolfe (MFW) for DRO

Input: functions F_i, time T, batch size c, parameter ρ, stepsizes $\rho_t > 0$
$x^{(0)} \leftarrow \mathbf{0}$
for $t = 1, \ldots, T$ **do**
  $p^{(t)} \leftarrow \operatorname{argmin}_{p \in \mathcal{P}_{\rho,n}} \sum_{i=1}^{n} p_i F_i(x^{(t-1)})$
  Draw $i_1, \ldots, i_c$ from $\{1, \ldots, n\}$
  $\tilde{\nabla}^{(t)} \leftarrow \frac{1}{c} \sum_{\ell=1}^{c} p_{i_\ell}^{(t)} \nabla F_{i_\ell}(x^{(t-1)})$
  $d^{(t)} \leftarrow (1 - \rho_t) d^{(t-1)} + \rho_t \tilde{\nabla}^{(t)}$
  $v^{(t)} \leftarrow \operatorname{argmax}_{v \in \mathcal{X}} \langle d^{(t)}, v \rangle$
  $x^{(t)} \leftarrow x^{(t-1)} + v^{(t)}/T$
end for
return $x^{(T)}$

Instead, observe that, since the pointwise minimum of concave functions is concave, the robust objective $G(x) = \min_{p \in \mathcal{P}_{\rho,n}} \sum_{i=1}^{n} p_i F_i(x)$ is also DR-submodular. However, a naive application of FW to $G(x)$ faces several difficulties. First, to evaluate and differentiate $G(x)$, we require an exact oracle for the inner minimization problem over p, whereas past work (Namkoong and Duchi, 2017) gave only an approximate oracle. In comparison, our submodular setting is more delicate, so an inexact oracle does not suffice: the issue is that two solutions to the inner problem can have arbitrarily *close solution value* while also providing arbitrarily *different gradients*. Hence, gradient steps with respect to an approximate minimizer may not actually improve the solution value. To resolve this issue, we provide an *exact* $O(n \log n)$ time subroutine in Appendix B.3. Compared to previous techniques, our algorithm rests on a more precise characterization of solutions to linear optimization over the χ^2 ball, which is often helpful in deriving structural results for general DRO problems (e.g., Lemmas 5.2.2 and 5.4.2).

Second, especially when the amount of data is large, we would like to use stochastic gradient estimates instead of requiring a full gradient computation at every iteration. This introduces additional noise and standard Frank-Wolfe algorithms will require $O(1/\tau^2)$ gradient samples per iteration to cope. Accordingly, we build on a recent algorithm of Mokhtari et al. (2018a) that accelerates Frank-Wolfe by re-using old gradient information; we refer to their algorithm as *Momentum Frank-Wolfe (MFW)*.

For smooth DR-submodular functions, MFW achieves a $(1 - 1/e)$-optimal solution with additive error τ in $O(1/\tau^3)$ time. Building off MFW is advantageous versus other stochastic first-order algorithms for DR-submodular maximization, e.g. Hassani et al. (2017) achieve suboptimal approximation ratio, and Karimi et al. (2017) focus only on a subclass of problems. Accordingly, we focus on MFW, and generalize MFW to the DRO problem by solving the next challenge.

Third, Frank-Wolfe (and MFW) require a smooth objective with Lipschitz-continuous gradients; this does *not* hold in general for pointwise minima. Wilder (2018a) gets around this issue in the context of other robust submodular optimization problems by replacing $G(x)$ with the stochastically smoothed function $G_\mu(x) = \mathbb{E}_{z\sim\mu}[G(x+z)]$ as in (Duchi et al., 2012; Lan, 2013), where μ is a uniform distribution over a ball of size u. Combined with our exact inner minimization oracle, this yields a $(1 - 1/e)$ optimal solution to Problem (5.20) with τ error using $O(1/\tau^4)$ stochastic gradient samples. However, this approach results in poor empirical performance for the DRO problem (as we demonstrate later). We obtain faster convergence, in both theory and practice, through a better characterization of the DRO problem: we show that in many cases, we actually obtain a smooth problem,

Smoothness of the robust problem. While general theoretical bounds rely on smoothing $G(x)$, in practice, MFW without any smoothing performs the best. This behavior suggests that for real-world problems, the robust objective $G(x)$ may actually be smooth with Lipschitz-continuous gradient. Via our exact characterization of the worst-case distribution, we can make this intuition rigorous:

Lemma 5.4.2. *Define $h(z) = \min_{p\in\mathcal{P}_{\rho,n}}\langle z, p\rangle$, for $z \in [0, B]^n$, and let s_n^2 be the sample variance of z. On the subset of z's satisfying the high sample variance condition $s_n^2 \geq (2\rho B^2)/n$, $h(z)$ is smooth and has L-Lipschitz gradient with constant $L \leq \frac{2\sqrt{2\rho}}{n^{3/2}} + \frac{2}{Bn}$.*

Combined with the smoothness of each F_i, this yields smoothness of G.

Corollary 5.4.1. *Suppose each F_i is L_F-Lipschitz. Under the high sample variance condition, ∇G is L_G-Lipschitz for $L_G = L_F + \frac{2b\sqrt{2\rho|V|}}{n} + \frac{2b\sqrt{|V|}}{B\sqrt{n}}$.*

For submodular functions, $L_F \leq b\sqrt{k}$, where b is the largest value of a single item (Mokhtari et al., 2018a). However, Corollary 5.4.1 is a general property of DRO (not specific to the submodular case), with broader implications. For instance, in the convex case, we immediately obtain a $O(1/\tau)$ convergence rate for the gradient descent algorithm proposed by Namkoong and Duchi (2017) (previously, the best possible bound would be $O(1/\tau^2)$ via nonsmooth techniques). Our result follows from more general properties that guarantee smoothness with fewer assumptions (see Appendices B.3.1, B.3.2). For example:

Fact 5.4.1. For $\rho \leq \frac{1}{2}$, the robust objective $h(z) = \min_{p \in \mathcal{P}_{\rho,n}} \langle z, p \rangle$ is smooth when $\{z_i\}$ are not all equal.

Combined with reasonable assumptions on the distribution of F_i, this means $G(x)$ is nearly always smooth. Native smoothness of the robust problem yields a significant runtime improvement over the general minimum-of-submodular case. In particular, instead of $O(1/\tau^4)$, we achieve the $O(1/\tau^3)$ rate of the simpler, non-robust submodular maximization:

Theorem 5.4.1. *When the high sample variance condition holds, MFW with no smoothing satisfies*

$$\mathbb{E}[G(x^{(T)})] \geq (1 - 1/e)\, OPT - \frac{2\sqrt{kQ}}{T^{1/3}} - \frac{Lk}{T}$$

where $Q = \max\{9^{2/3}\|\nabla G(x^0) - d^0\|, 16\sigma^2 + 3L_G^2 k\}$*;* σ *is the variance of the stochastic gradients.*

This convergence rate for DRO is almost the same as for a single submodular function (the non-robust case) (Mokhtari et al., 2018a); only the Lipschitz constant is different, but this gap vanishes as n grows. It is perhaps surprising that we can obtain this rate for the robust problem, especially using an algorithm like MFW which was originally intended for the nonrobust setting. Indeed, previous work on robust submodular optimization has relied on different techniques; MFW is not an obvious candidate for DRO. However, as surveyed below, our better characterization

of the DRO problem and subsequent ability to leverage MFW yields theoretical and empirical benefits.

Comparison with previous algorithms Two recently proposed algorithms for robust submodular maximization could also be used in DRO, but have drawbacks compared to MFW. Here, we compare their theoretical performance with MFW (we also show how MFW improves empirically in Section 5.5).

First, Chen et al. (2017) view robust optimization as a zero-sum game and apply no-regret learning to compute an approximate equilibrium. Their algorithm applies online gradient descent from the perspective of the adversary, adjusting the distributional parameters p. At each iteration, an α-approximate oracle for submodular optimization (e.g., the greedy algorithm or a Frank-Wolfe algorithm) is used to compute a best response for the maximizing player. In order to achieve an α-approximation up to additive loss τ, the no-regret algorithm requires $O(1/\tau^2)$ iterations. However, each iteration requires a full invocation of an algorithm for submodular maximization. Our MFW algorithm requires runtime close to a *single* submodular maximization call. This results in substantially faster runtime to achieve the same solution solution quality, as we demonstrate experimentally.

Second, Wilder (2018a) proposes the EQUATOR algorithm, which also applies a Frank-Wolfe approach to the multilinear extension but uses randomized smoothing. Our analysis shows smoothing is unnecessary for the DRO problem, allowing our algorithm to converge using $O(1/\tau^3)$ stochastic gradients, while EQUATOR requires $O(1/\tau^4)$. This theoretical gap is reflected in empirical performance: EQUATOR converges much more slowly, and to lower solution quality, than MFW.

5.5 Experiments

To probe the strength and practicality of our methods, we empirically study the two motivating problems from Section 5.2: influence maximization and facility location. We first report performance of distributions x^* that optimize the multilinear extension

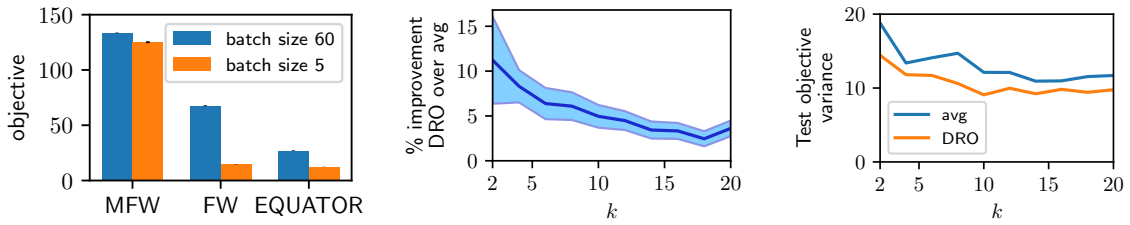
or its DRO variant (5.20), and later demonstrate high performance is maintained even after rounding. DRO improves test performance in all cases. All error bars are 95% confidence intervals.

5.5.1 Facility Location

Similar to (Mokhtari et al., 2018a) we consider a facility location problem motivated by recommender systems. We use a music dataset from last.fm (las) with roughly 360000 users, 160000 bands, and over 17 million total records. For each user i, record r_j^i indicates how many times they listened to a song by band j. We seek a subset of bands so that the average user likes at least one of the bands, as measured by the playcounts. More specifically, we fix a collection of bands, and observe a *sample* of users; we seek a subset of bands that performs well on the *entire population* of users. Here, we randomly sample a subset of 1000 "train" users from the dataset, solve the DRO and ERM problems for k bands, and evaluate performance on the remaining ≈ 360000 "test" users from the dataset.

Optimization. We first compare MFW to previously proposed robust optimization algorithms, applied to the DRO problem with $k = 3$. Figure 5-1a compares **1.** MFW, **2.** Frank-Wolfe (FW) with no momentum and **3.** EQUATOR (Wilder, 2018a). Naive FW handles noisy gradients poorly (especially with small batches), while EQUATOR underperforms since its randomized smoothing is not necessary for our natively smooth problem. We also compared to the online gradient descent (OGD) algorithm of Chen et al. (2017). OGD achieved slightly worse objective value than MFW with an order of magnitude greater runtime: OGD required 53.23 minutes on average, compared to 4.81 for MFW. EQUATOR and FW had equivalent runtime to MFW since all used the same batch size and number of iterations. MFW dominates the alternatives in both runtime and solution quality.

Generalization. Next, we evaluate the effect of DRO on test set performance across varying set sizes k. Results are averaged over 64 trials for $\rho = 10$ (corresponding to probability of failure $\delta = e^{-10}$ of the high probability bound). In Figure 5-1b we plot the mean percent improvement in test objective of DRO versus optimizing the



(a) Algorithm comparison (b) % improvement via DRO (c) Test performance variance

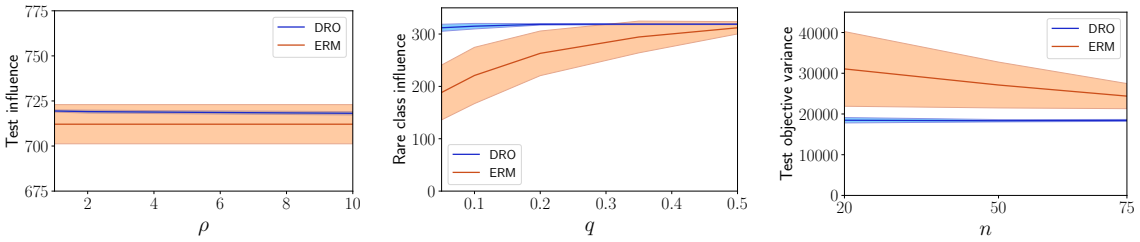
Figure 5-1: Algorithm comparison and generalization performance on last.fm dataset.

average. DRO achieves clear gains, especially when the set size k is small. In Figure 5-1c we show the variance of test performance achieved by each method. DRO achieves lower variance, meaning that overall DRO achieves better test performance, and with better consistency.

5.5.2 Influence maximization

As described in Section 5.2, we study an influence maximization problem where we observe samples of live-edge graphs $\mathcal{E}_1, \ldots, \mathcal{E}_n \sim \mathbb{P}$. Our setting is challenging for learning: the number of samples is small and P has high variance. Specifically, $\mathbb{P}$ is a mixture of two different independent cascade models (ICM). In the ICM, each edge e is (independently) live with probability p_e. In our mixture, each edge has $p_e = 0.025$ with probability q and $p_e = 0.1$ with probability $1 - q$, mixing between settings of low and high influence spread. This models the realistic case where some messages are shared more widely than others. The mixture is *not* an ICM, as observing the state of one edge gives information about the propagation probability for other edges. Handling such cases is an advantage of our DRO approach over ICM-specific robust influence maximization methods (Chen et al., 2016).

We use the political blogs dataset, a network with 1490 nodes representing links between blogs related to politics (Adamic and Glance, 2005). Figure 5-2 compares the performance of DRO and ERM. Figure 5-2a shows that DRO generalizes better, achieving higher performance on the test set. Each algorithm was given $n = 20$ training samples, $k = 10$ seeds, and we set q (the frequency of low influence) to be 0.1. Test influence was evaluated via a held-out set of 3000 samples from P.



(a) Influence on held-out set (b) Rare class held-out perf. (c) Test performance variance

Figure 5-2: Influence maximization on political blogs dataset.

Figure 5-2b shows that DRO's improved generalization stems from greatly improved performance on the rare class in the mixture (low propagation probabilities). For these instances, DRO obtains a greater than *40% improvement over ERM* in held-out performance for $q = 0.1$. As q increases (i.e., the rare class becomes less rare), ERM's performance on these instances converges towards DRO. A similar pattern is reflected in Figure 5-2c, which shows the variance in each algorithm's influence spread on the test set as a function of the number of training samples. *DRO's variance is lower by 25-40%.* As expected, DRO's advantage is greatest for small n, the most challenging setting for learning.

5.5.3 Rounding

Above, we report results achieved by the optimal distribution x^* on the multilinear extension $F(x^*)$ of the relevant stochastic submodular function. But to use our methods in practice, we eventually need to round x^* to a single subset S. One might worry that variabilty from the rounding procedure could erase DRO's gains. This is not the case: DRO still performs better empirically, even after rounding.

On the earlier Facility Location problem for $k = 4$, we compared the optimal distributions x^*_{ERM} and x^*_{DRO} for ERM and DRO. For each, we rounded 500 times to deterministic sets via swap rounding (Chekuri et al., 2010) and compared the resulting distributions of test objective values $\mathbb{E}_{f\sim\mathbb{P}}[f(S)]$ (on a large subsample from the test set P). Over 64 trials (the stochasticity of MFW leads to different $x^*_{\text{ERM}}, x^*_{\text{DRO}}$), we observed that: 1. DRO always achieved better mean performance, on average by 9.3%; 2. DRO achieved lower variance in 88% of trials; 3. *for every quantile*, DRO

was better on at least 73% of trials. We conclude DRO leads to better performance on the test set, both on $F(x^*)$ and on the original problem after rounding.

5.6 Discussion and future work

In this chapter we address *stochastic submodular optimization* (SSO), where we wish to optimize $f_{\mathbb{P}}(S) = \mathbb{E}_{f\sim\mathbb{P}}[f(S)]$. Unlike prior work, we focus on the setting where only a finite number of samples $f_1, \ldots, f_n \sim \mathbb{P}$ is available. Instead of simply maximizing the empirical mean $\frac{1}{n}\sum_i f_i$, we directly optimize a variance-regularized version which **1.** gives a high probability lower bound for $f_{\mathbb{P}}(S)$ (generalization) and **2.** allows us to trade off bias and variance in estimating $f_{\mathbb{P}}$. We accomplish this via an equivalent reformulation as a *distributionally robust* submodular optimization problem. Along the way, we show new results for the relation between distributionally robust optimization (DRO) and variance regularization. We further give conditions for the uniqueness of the DRO solution: these are broadly useful for guaranteeing that DRO problems are smooth. Even though robust submodular maximization is hard in general, as discussed in Chapter 4, we are able to give efficient approximation algorithms for our reformulation. Empirically, our approach yields notable improvements for influence maximization and facility location problems.

The intersection of DRO and submodular optimization is ripe with possibility, and there are many interesting directions to pursue, one of which we will explore in Chapter 6. Other types of DRO problems, such as those discussed in Chapters 2 and 3, may also prove tractable and effective in the SSO setting. Other submodular problems, such as submodular minimization, may admit useful and tractable DRO reformulations or algorithms. Weaker versions of submodularity, that retain good optimization properties, may also admit tractable DRO algorithms with guarantees. We are hopeful that the notion of submodularity will enable the field to further expand the set of non-convex functions that are amenable to DRO.

Chapter 6

Robust Budget Allocation

6.1 Introduction

In Chapter 5 we studied a relatively general class of problems, namely stochastic submodular optimization, and we showed how to improve performance via DRO. While those results apply to many submodular objectives, in this chapter, we instead focus on a specific problem, *(Robust) Budget Allocation*, which is of interest to the machine learning and data mining communities. While the problem itself is more narrow, in solving it, we develop techniques in section 6.4 that have wide applicabilty in constrained submodular minimization.

Our motivation stems from the optimal allocation of resources for maximizing influence, or spread of information or coverage. These problems have gained attention in the past few years (Domingos and Richardson, 2001; Kempe et al., 2003; Chen et al., 2009; Gomez Rodriguez et al., 2012; Borgs et al., 2014). Formally, in the *Budget Allocation Problem*, one is given a bipartite influence graph between channels S and people T, and the task is to assign a budget $y(s)$ to each channel s in S with the goal of maximizing the expected number of influenced people $\mathcal{I}(y)$. We illustrate the setup in Figure 6-1. Each edge $(s,t) \in E$ between channel s and person t is weighted with a probability p_{st} that, e.g., an advertisement on radio station s will influence person t to buy some product. The budget $y(s)$ controls how many independent attempts are made via the channel s to influence the people in T. The probability

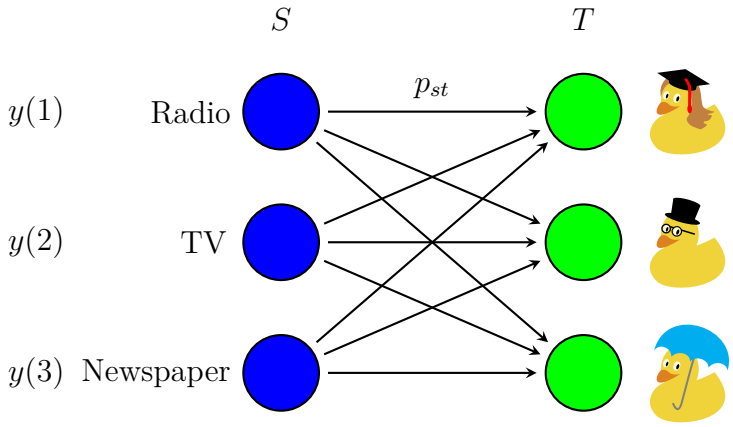


Figure 6-1: Bipartite graph demonstrating the setup of the (Robust) Budget Allocation problem. Each individual on the right can be influenced via any of the channels on the left. Each edge is weighted with a probability p_{st} representing how susceptible person $t \in T$ is to channel $s \in S$.

that a customer t is influenced when the advertising budget is y is

$$I_t(y) = 1 - \prod_{(s,t)\in E} [1 - p_{st}]^{y(s)}, \tag{6.1}$$

and hence the expected number of influenced people is $\mathcal{I}(y) = \sum_{t\in T} I_t(y)$. We write $\mathcal{I}(y; p) = \mathcal{I}(y)$ to make the dependence on the probabilities p_{st} explicit. The total budget y must remain within some feasible set $\mathcal{Y}$ which may encode e.g. a total budget limit $\sum_{s\in S} y(s) \leq C$. We allow the budgets y to be continuous, as in (Bian et al., 2017).

Since its introduction by Alon et al. (2012), several works have extended the formulation of Budget Allocation and provided algorithms (Bian et al., 2017; Hatano et al., 2015; Maehara et al., 2015; Soma et al., 2014; Soma and Yoshida, 2015). Budget Allocation may also be viewed as influence maximization on a bipartite graph, where information spreads as in the Independent Cascade model. For integer y, Budget Allocation and Influence Maximization are NP-hard. Yet, constant-factor approximations are possible, and build on the fact that the influence function is submodular in the binary case, and *DR-submodular* in the integer case (Soma et al., 2014; Hatano et al., 2015). If y is continuous, the problem is a concave maximization

problem.

The formulation of Budget Allocation assumes that the transmission probabilities are known exactly. But this is rarely true in practice. Typically, the probabilities p_{st}, and possibly the graph itself, must be inferred from observations (Gomez Rodriguez et al., 2010; Du et al., 2013; Narasimhan et al., 2015; Du et al., 2014; Netrapalli and Sanghavi, 2012). In Section 6.6 we will see that a misspecification or point estimate of parameters p_{st} can lead to much reduced outcomes. A more realistic assumption is to know *confidence intervals* for the p_{st}. Realizing this severe deficiency, recent work studied robust versions of Influence Maximization, where a budget y must be chosen that maximizes the worst-case approximation ratio over a set of possible influence functions (He and Kempe, 2016; Chen et al., 2016; Lowalekar et al., 2016). The resulting optimization problem is hard but admits bicriteria approximations.

In this work, we revisit Budget Allocation under uncertainty from the perspective of robust optimization (Bertsimas et al., 2011; Ben-Tal et al., 2009). We maximize the worst-case influence – not approximation ratio – for p in a confidence set centered around the "best guess" (e.g., posterior mean). This avoids pitfalls of the approximation ratio formulation (which can be misled to return poor worst-case budgets, as demonstrated in Appendix C.1), while also allowing us to formulate the problem as a max-min game:

$$\max_{y \in \mathcal{Y}} \min_{p \in \mathcal{P}} \mathcal{I}(y; p), \tag{6.2}$$

where an "adversary" can arbitrarily manipulate p within the confidence set $\mathcal{P}$. With p fixed, $\mathcal{I}(y; p)$ is concave in y. However, the influence function $\mathcal{I}(y; p)$ is not convex, and not even quasiconvex, in the adversary's variables p_{st}.

The new, key insight we exploit in this work is that $\mathcal{I}(y; p)$ has the property of *continuous submodularity* in p – in contrast to previously exploited submodular maximization in y – and can hence be minimized by generalizing techniques from discrete submodular optimization (Bach, 2019). The techniques in (Bach, 2019), however, are restricted to box constraints, and do not directly apply to our confidence sets. In fact, general constrained submodular minimization is hard (Svitkina and

Fleischer, 2011; Goel et al., 2009; Iwata and Nagano, 2009). We make the following contributions:

1. We provide the first results for continuous submodular minimization with box constraints and one more "nice" constraint, and checkable conditions under which the algorithm is guaranteed to return a global optimum. In other words, we have a provable algorithm for a new class of constrained nonconvex minimization problems that should be of interest more broadly.
2. Leveraging the above result, we present an algorithm with optimality bounds for Robust Budget Allocation in the nonconvex adversarial scenario (6.2).

6.1.1 Background and related work

For background on submodularity and diminishing returns, refer to Chapter 4. Particularly relevant are section 4.1.2 on submodular minimization, section 4.2 on general submodular functions, and section 4.3.1 on risk-averse submodular optimization. We will draw on all of these in both our approach to the problem, as well as in our algorithmic solution. In the rest of this background section we focus on discussing problems most related to Budget Allocation.

A sister problem of Budget Allocation is *Influence Maximization* on general graphs, where a set of seed nodes is selected to start a propagation process. The influence function is still monotone submodular and amenable to the greedy algorithm (Kempe et al., 2003), but it cannot be evaluated explicitly and requires approximation (Chen et al., 2010).

Stochastic Coverage (Goemans and Vondrák, 2006) is a version of Set Cover where the covering sets $S_i \subset V$ are random. A variant of Budget Allocation can be written as stochastic coverage with multiplicity. Stochastic Coverage has mainly been studied in the online or adaptive setting, where logarithmic approximation factors can be achieved (Golovin and Krause, 2011; Deshpande et al., 2016; Adamczyk et al., 2016).

Our objective function (6.2) is a *signomial* in p, i.e., a linear combination of monomials of the form $\prod_i x_i^{c_i}$. General signomial optimization is NP-hard (Chiang, 2005),

but certain subclasses are tractable: *posynomials* with all nonnegative coefficients can be minimized via Geometric Programming (Boyd et al., 2007), and signomials with a single negative coefficient admit sum of squares-like relaxations (Chandrasekaran and Shah, 2016). Our problem, a constrained posynomial maximization, is not in general a geometric program. Some work addresses this setting via monomial approximation (Pascual and Ben-Israel, 1970; Ecker, 1980), but, to our knowledge, our algorithm is the first that solves this problem to arbitrary accuracy.

6.2 Robust and stochastic Budget Allocation

The unknown parameters in Budget Allocation are the transmission probabilities p_{st} or edge weights in a graph. If these are estimated from data, we may have posterior distributions or, a weaker assumption, confidence sets for the parameters. For ease of notation, we will work with the failure probabilities $x_{st} = 1 - p_{st}$ instead of the p_{st} directly, and write $\mathcal{I}(y; x)$ instead of $\mathcal{I}(y; p)$.

6.2.1 Stochastic optimization

If a (posterior) distribution of the parameters is known, a simple strategy is to use expectations. For example, we can place a uniform prior on x_{st}, and observe n_{st} independent observations drawn from $\text{Ber}(x_{st})$. If we observe α_{st} failures and β_{st} successes, the resulting posterior distribution on the variable X_{st} is $\text{Beta}(1 + \alpha_{st}, 1 + \beta_{st})$. Given such a posterior, we may optimize

$$\max_{y \in \mathcal{Y}} \ \mathcal{I}(y; \mathbb{E}[X]) \qquad (6.3) \qquad \text{or} \qquad \max_{y \in \mathcal{Y}} \ \mathbb{E}[\mathcal{I}(y; X)]. \qquad (6.4)$$

Proposition 6.2.1. *Problems (6.3) and (6.4) are concave maximization problems over the (convex) set $\mathcal{Y}$ and can be solved exactly.*

Concavity of (6.4) follows since it is an expectation over concave functions, and it can be solved by stochastic gradient ascent or by explicitly computing gradients.

Merely maximizing expectation does not explicitly account for volatility and hence risk. One option is to penalize variance (Ben-Tal and Nemirovski, 2000; Bertsimas

et al., 2011; Atamtürk and Narayanan, 2008):

$$\min_{y \in \mathcal{Y}} \; -\mathbb{E}[\mathcal{I}(y; X)] + \varepsilon \sqrt{\mathrm{Var}(\mathcal{I}(y; X))}. \tag{6.5}$$

This approach is strongly related to optimizing CVaR, discussed in section 4.3.1. However, for submodular objectives, these approaches to risk-aversion can be challenging, and our case is no exception:

Fact 6.2.1. For y in the nonnegative orthant, the term $\sqrt{\mathrm{Var}(\mathcal{I}(y; X))}$ need not be convex or concave, and need not be submodular or supermodular.

This observation does not rule out a solution, but the apparent difficulties further motivate a robust formulation that, as we will see, is amenable to optimization.

6.2.2 Robust optimization

The focus of this work is the robust version of Budget Allocation, where we allow an adversary to arbitrarily set the parameters x within an uncertainty set $\mathcal{X}$. This uncertainty set may result, for instance, from a known distribution, or simply from assumed bounds. Formally, we solve

$$\max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} \; \mathcal{I}(y; x), \tag{6.6}$$

where $\mathcal{Y} \subset \mathbb{R}_+^S$ is a convex set with an efficient projection oracle, and $\mathcal{X}$ is an uncertainty set containing an estimate $\hat{x}$. In the sequel, we use uncertainty sets $\mathcal{X} = \{x \in \mathrm{Box}(l, u) : R(x) \leq B\}$, where R is a distance (or divergence) from the estimate $\hat{x}$, and $\mathrm{Box}(l, u)$ is the box $\prod_{(s,t) \in E} [l_{st}, u_{st}]$. The intervals $[l_{st}, u_{st}]$ can be thought of as either confidence intervals around $\hat{x}$, or, if $[l_{st}, u_{st}] = [0, 1]$, they enforce that each x_{st} is a valid probability.

Common examples of uncertainty sets used in robust optimization are *Ellipsoidal* and *D-norm uncertainty sets* (Bertsimas et al., 2011). Our algorithm in Section 6.4 applies to both.

Ellipsoidal uncertainty. The ellipsoidal or quadratic uncertainty set is defined by

$$\mathcal{X}^Q(\gamma) = \{x \in \text{Box}(0,1) : (x - \hat{x})^T \Sigma^{-1} (x - \hat{x}) \leq \gamma\},$$

where Σ is the covariance of the random vector X of probabilities distributed according to our Beta posteriors. In our case, since the distributions on each x_{st} are independent, Σ^{-1} is actually diagonal. Writing $\Sigma = \text{diag}(\sigma^2)$, we have

$$\mathcal{X}^Q(\gamma) = \Big\{x \in \text{Box}(0,1) : \sum_{(s,t)\in E} R_{st}(x_{st}) \leq \gamma\Big\},$$

where $R_{st}(x) = (x_{st} - \hat{x}_{st})^2 \sigma_{st}^{-2}$.

D-norm uncertainty. The D-norm uncertainty set is similar to an ℓ_1-ball around $\hat{x}$, and is defined as

$$\mathcal{X}^D(\gamma) = \Big\{x : \exists c \in \text{Box}(0,1) \text{ s.t. } x_{st} = \hat{x}_{st} + (u_{st} - \hat{x}_{st})c_{st}, \sum_{(s,t)\in E} c_{st} \leq \gamma\Big\}.$$

Essentially, we allow an adversary to increase $\hat{x}_{st}$ up to some upper bound u_{st}, subject to some total budget γ across all terms x_{st}. The set $\mathcal{X}^D(\gamma)$ can be rewritten as

$$\mathcal{X}^D(\gamma) = \Big\{x \in \text{Box}(\hat{x}, u) : \sum_{(s,t)\in E} R_{st}(x_{st}) \leq \gamma\Big\},$$

where $R_{st}(x_{st}) = (x_{st} - \hat{x}_{st})/(u_{st} - \hat{x}_{st})$ is the fraction of the interval $[\hat{x}_{st}, u_{st}]$ we have used when increasing x_{st}.

Comparison to DRO. In the context of the preceding chapters, it is important to note that, in our approach to Robust Budget Allocation, we technically depart from DRO. This is because the uncertainty we allow in the probabilities x (or p) affects both the distribution as well as the objective function – whereas in DRO, only the distribution changes. This technicality manifests because the decision maker in Budget Allocation actually alters the relevant distribution: by changing y, the

decision maker changes the distribution of which edges (s,t) are active. Nevertheless, the spirit of Robust Budget Allocation is the same: we want to do well in expectation, namely to maximize expected influence, even when the underlying distribution is perturbed.

Overall, the min-max formulation $\max_{y\in\mathcal{Y}} \min_{x\in\mathcal{X}} \mathcal{I}(y;x)$ has several benefits: the model is not tied to a specific learning algorithm for the probabilities x as long as we can choose a suitable confidence set. Moreover, this formulation allows to fully hedge against a worst-case scenario.

6.3 Robust Budget Allocation: main ideas

Next, we address in two main steps how to solve Problem (6.6), first the outer and then the inner optimization problem. As noted above, the function $\mathcal{I}(y;x)$ is concave as a function of y for fixed x. As a pointwise minimum of concave functions, $F(y) := \min_{x\in\mathcal{X}} \mathcal{I}(y;x)$ is concave. Hence, if we can compute subgradients of $F(y)$, we can solve our max-min-problem via the subgradient method, as outlined in Algorithm 3.

A subgradient $g_y \in \partial F(y)$ at y is given by the gradient of $\mathcal{I}(y;x^*)$ for the minimizing $x^* \in \arg\min_{x\in\mathcal{X}} \mathcal{I}(y;x)$, i.e., $g_y = \nabla_y \mathcal{I}(y;x^*)$. Hence, we must be able to compute x^* for any y. We also obtain a duality gap: for any x', y' we have

$$\min_{x\in\mathcal{X}} \mathcal{I}(y';x) \le \max_{y\in\mathcal{Y}} \min_{x\in\mathcal{X}} \mathcal{I}(y;x) \le \max_{y\in\mathcal{Y}} \mathcal{I}(y;x'). \tag{6.7}$$

This means we can estimate the optimal value $\mathcal{I}^*$ and use it in Polyak's stepsize rule for the subgradient method (Polyak, 1987).

What remains to be addressed is how to compute x^*. $\mathcal{I}(y;x)$ is not convex in x, and not even quasiconvex. For example, standard methods (Wainwright and Chiang, 2004, Chapter 12) imply that $f(x_1,x_2,x_3) = 1 - x_1x_2 - \sqrt{x_3}$ is not quasiconvex on $\mathbb{R}^3_+$. Moreover, the above-mentioned signomial optimization techniques do not apply for an exact solution either. So, it is not immediately clear that we can solve the inner optimization problem.

Algorithm 3 Subgradient Ascent

Input: suboptimality tolerance $\tau > 0$, initial feasible budget $y^{(0)} \in \mathcal{Y}$
Output: τ-optimal budget y for Problem (6.6)
repeat
  $x^{(k)} \leftarrow \arg\min_{x \in \mathcal{X}} \mathcal{I}(y^{(k)}; x)$ ▷ Find worst-case x for $y^{(k)}$
  $g^{(k)} \leftarrow \nabla_y \mathcal{I}(y^{(k)}; x^{(k)})$ ▷ Gradient with respect to y of $\min_{x \in \mathcal{X}} \mathcal{I}(y; x)$ at $y = y^{(k)}$
  $L^{(k)} \leftarrow \mathcal{I}(y^{(k)}; x^{(k)})$ ▷ Lower bound on optimal value
  $U^{(k)} \leftarrow \max_{y \in \mathcal{Y}} \mathcal{I}(y; x^{(k)})$ ▷ Upper bound on optimal value
  $\gamma^{(k)} \leftarrow (U^{(k)} - L^{(k)})/\|g^{(k)}\|_2^2$ ▷ Polyak's stepsize rule
  $y^{(k+1)} \leftarrow \text{proj}_{\mathcal{Y}}(y^{(k)} + \gamma^{(k)} g^{(k)})$
  $k \leftarrow k + 1$
until $U^{(k)} - L^{(k)} \leq \tau$

The key insight we will be using is that $\mathcal{I}(y; x)$ has a different beneficial property: while not convex, $\mathcal{I}(y; x)$ as a function of x is *continuous submodular*.

Lemma 6.3.1. *Suppose we have $n \geq 1$ differentiable functions $f_i : \mathbb{R} \to \mathbb{R}_+$, for $i = 1, \ldots, n$, either all nonincreasing or all nondecreasing. Then, $f(x) = \prod_{i=1}^n f_i(x_i)$ is a continuous supermodular function from $\mathbb{R}^n$ to $\mathbb{R}_+$.*

Proof. For $n = 1$, the resulting function is modular and therefore supermodular. In the case $n \geq 2$, we simply need to compute derivatives. The mixed derivatives are

$$\frac{\partial f}{\partial x_i \partial x_j} = f_i'(x_i) f_j'(x_j) \cdot \prod_{k \neq i,j} f_k(x_k). \tag{6.8}$$

By monotonicity, f_i' and f_j' have the same sign, so their product is nonnegative, and since each f_k is nonnegative, the entire expression is nonnegative. Hence, $f(x)$ is continuous supermodular by Theorem 3.2 of Topkis (1978). □

Corollary 6.3.1. *The influence function $\mathcal{I}(y; x)$ defined in Section 6.2 is continuous submodular in x over the nonnegative orthant, for each $y \geq 0$.*

Proof. Since submodularity is preserved under summation, it suffices to show that each function $I_t(y)$ is continuous submodular. By Lemma 6.3.1, since $f_s(z) = z^{y(s)}$ is nonnegative and monotone nondecreasing for $y(s) \geq 0$, the product $\prod_{(s,t) \in E} x_{st}^{y(s)}$ is

continuous supermodular in x. Flipping the sign and adding a constant term yields $I_t(y)$, which is hence continuous submodular. □

We further conjecture that the functions $\mathcal{I}(y; x)$ enjoy another beneficial property, beyond submodularity:

Conjecture 6.3.1. *Strong duality holds, i.e.,*

$$\max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} \mathcal{I}(y; x) = \min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \mathcal{I}(y; x). \tag{6.9}$$

If strong duality holds, then the duality gap $\max_{y \in \mathcal{Y}} \mathcal{I}(y; x^*) - \min_{x \in \mathcal{X}} \mathcal{I}(y^*; x)$ in Equation (6.7) is zero at optimality. If $\mathcal{I}(y; x)$ were quasiconvex in x, strong duality would hold by Sion's min-max theorem, but this is not the case. In practice, we observe that the duality gap always converges to zero.

We have seen the functions $\mathcal{I}(y; x)$ enjoy nice structural properties, but it is still not clear how to solve the inner problem. Bach (2019) demonstrates how to minimize a continuous submodular function $H(x)$ subject to box constraints $x \in \text{Box}(l, u)$, up to an arbitrary suboptimality gap $\tau > 0$. The constraint set $\mathcal{X}$ in our Robust Budget Allocation problem, however, has box constraints with an additional constraint $R(x) \leq B$. This case is not addressed in any previous work. Fortunately, for a large class of functions R, there is still an efficient algorithm for continuous submodular minimization, which we present in the next section.

6.4 Constrained continuous submodular function minimization

The previous section shows that, to solve Robust Budget Allocation, we need an algorithm for minimizing a monotone continuous submodular function $H(x)$ subject

to box constraints $x \in \text{Box}(l, u)$ and a constraint $R(x) \leq B$:

$$\begin{aligned} \text{minimize} \quad & H(x) \\ \text{s.t.} \quad & R(x) \leq B \\ & x \in \text{Box}(l, u). \end{aligned} \tag{6.10}$$

If H and R were convex, the constrained problem would be equivalent to solving, with the right Lagrange multipler $\lambda^* \geq 0$:

$$\begin{aligned} \text{minimize} \quad & H(x) + \lambda^* R(x) \\ \text{s.t.} \quad & x \in \text{Box}(l, u). \end{aligned} \tag{6.11}$$

Although H and R are not necessarily convex here, it turns out that a similar approach indeed applies. The property of submodularity then enables a special relaxation that allows one to obtain a solution for all possible values of λ via a single convex optimization problem. The main idea of our approach bears similarity with (Nagano et al., 2011) for the set function case, but our setting with continuous functions and various uncertainty sets is more general, and requires more argumentation. We present our theoretical results here, and defer implementation details to the appendix.

6.4.1 Forming an equivalent convex problem

Following Bach (2019), we discretize the problem; for a sufficiently fine discretization, we will achieve arbitrary accuracy. This discretization will in turn lead to a convex relaxation. Let A be an *interpolation mapping* that maps the discrete set $\prod_{i=1}^{n}[k_i]$ into $\text{Box}(l, u) = \prod_{i=1}^{n}[l_i, u_i]$ via the componentwise interpolation functions $A_i : [k_i] \to [l_i, u_i]$. We say A_i is δ-*fine* if $A_i(x_i + 1) - A_i(x_i) \leq \delta$ for all $x_i \in \{0, 1, \ldots, k_i - 2\}$. We will further say the full interpolation function A is δ-fine if each A_i is δ-fine.

This mapping yields functions $H^\delta : \prod_{i=1}^{n}[k_i] \to \mathbb{R}$ and $R^\delta : \prod_{i=1}^{n}[k_i] \to \mathbb{R}$ via $H^\delta(x) = H(A(x))$ and $R^\delta(x) = R(A(x))$. H^δ is submodular on the integer lattice. This construction reduces Problem (6.11) to a submodular minimization problem over

the integer lattice:

$$\begin{aligned} &\text{minimize} && H^\delta(x) + \lambda R^\delta(x) \\ &\text{s.t.} && x \in \textstyle\prod_{i=1}^n [k_i]. \end{aligned} \tag{6.12}$$

Motivated by convex optimization, one may hope that there exists a λ whose associated minimizer $x(\lambda)$ yields a nearly optimal solution for the corresponding constrained Problem (6.10) in the lattice case, where H^δ and R^δ replace H and R. Theorem 6.4.2 below states that, under a condition, this is indeed the case. Moreover, a second benefit of submodularity is that we can find the entire solution path for Problem (6.12) by solving a single optimization problem.

Lemma 6.4.1. *Suppose H is continuous submodular, and suppose the regularizer R is strictly increasing and separable: $R(x) = \sum_{i=1}^n R_i(x_i)$. Then we can recover a minimizer $x(\lambda)$ for the induced discrete Problem (6.12) for any $\lambda \in \mathbb{R}$ by solving a single convex optimization problem.*

To formally prove Lemma 6.4.1, we need to go into more detail. The convex optimization problem arises from a relaxation $h_\downarrow$ that is an analogue of the *Lovász extension* of set functions to continuous submodular functions (Bach, 2019). The basic idea for the extension $h_\downarrow$ is: instead of fixing a value for each coordinate of x, we give a distribution over values, and $h_\downarrow$ is the expected function value under that distribution. As a corollary, $h_\downarrow$ coincides with H^δ on lattice points.

Instead of specifying a full joint distribution over all coordinates, we will only need to give coordinatewise marginals μ_i. It is also convenient to represent the distributions μ_i via their (reversed) cumulative distributions functions ρ_i. The best joint distribution follows directly from these marginals: it is the solution to a multi-marginal optimal transport problem between the marginals, where the transport cost is the original submodular function H or H^δ. Formally $h_\downarrow$ can be defined as:

Definition 6.4.1 (Bach (2019))**.** Write $\mathcal{X} = \prod_{i=1}^n \mathcal{X}_i$, and let $H : \mathcal{X} \to \mathbb{R}$ be a submodular function (discrete or continuous). We define the *generalized Lovász extension*

of H by:

$$h_\downarrow(\rho_1, \ldots, \rho_n) = h_\downarrow(\mu_1, \ldots, \mu_n) := \inf_{\gamma \in \mathcal{P}(\mathcal{X}, \{\mu_i\})} \int_{\mathcal{X}} H(x)\, d\gamma(x) \tag{6.13}$$

where $\mathcal{P}(\mathcal{X}, \{\mu_i\})$ is the set of measures γ whose marginals match the μ_i, for all coordinates i.

Importantly, $h_\downarrow$ is convex if and only if H is submodular (Bach, 2019, Theorem 1). This makes optimizing $h_\downarrow$ tractable. To prove Lemma 6.4.1 we will use a specific correspondence between a discrete submodular function H^δ and its extension $h_\downarrow$:

Theorem 6.4.1 (Theorem 4 from Bach (2019))**.** *Let $H^\delta : \prod_{i=1}^n [k_i] \to \mathbb{R}$ be a submodular function with generalized Lovász extension $h_\downarrow$. Also let a_{iy_i} be strictly convex functions for all $i = 1, \ldots, n$ and each $y_i \in [k_i]$. The set $\mathbb{R}^k_\downarrow$ refers to the set of ordered vectors $z \in \mathbb{R}^k$ that satisfy $z_1 \geq z_2 \geq \cdots \geq z_k$, and the notation $\rho_i(x_i)$ denotes the x_i-th coordinate of the vector ρ_i. The vector ρ_i should still be understood as a discrete reverse cumulative distribution function, as stated earlier. For convenience we also write $\rho = \rho_1, \ldots, \rho_n$. Then the two problems*

$$\begin{array}{ll} \text{minimize} & H^\delta(x) + \sum_{i=1}^n \sum_{y_i=1}^{x_i} a'_{iy_i}(\lambda) \\ \text{s.t.} & x \in \prod_{i=1}^n [k_i]. \end{array} \tag{6.14}$$

and

$$\begin{array}{ll} \text{minimize} & h_\downarrow(\rho) + \sum_{i=1}^n \sum_{x_i=1}^{k_i - 1} a_{ix_i}[\rho_i(x_i)] \\ \text{s.t.} & \rho \in \prod_{i=1}^n \mathbb{R}^{k_i - 1}_\downarrow \end{array} \tag{6.15}$$

are equivalent. Specifically, one recovers a solution to Problem (6.14) for any λ: find ρ^* which solves Problem (6.15) and, for each component i, choose x_i to be the maximal value for which $\rho^*_i(x_i) \geq \lambda$.

With Theorem 6.4.1 in hand, we are finally ready to prove Lemma 6.4.1. Our high-level strategy is to convert Problem (6.12) into the form of Problem (6.14). Per Theorem 6.4.1, we can solve Problem (6.14) and hence Problem (6.12) simultaneously for all λ, simply by solving the single convex Problem (6.15).

Lemma 6.4.1. The discretized form of the regularizer R^δ is also separable and can be written $R^\delta(x) = \sum_{i=1}^n R_i^\delta(x)$. For each $i = 1, \dots, n$ and each $y_i \in [k_i]$ with $y_i \geq 1$, define $a_{iy_i}(t) = \frac{1}{2}t^2 \cdot [R_i^\delta(y_i) - R_i^\delta(y_i - 1)]$, so that $a'_{iy_i}(t) = t \cdot [R_i^\delta(y_i) - R_i^\delta(y_i - 1)]$. Since we assumed $R(x)$ is strictly increasing, the coefficient of t^2 in each $a_{iy_i}(t)$ is strictly positive, so that each $a_{iy_i}(t)$ is strictly convex. Then,

$$\lambda R_i^\delta(x_i) = \lambda \cdot \left[R_i^\delta(0) + \sum_{y_i=1}^{x_i} \left(R_i^\delta(y_i) - R_i^\delta(y_i - 1) \right) \right] \tag{6.16}$$

$$= \lambda R_i^\delta(0) + \sum_{y_i=1}^{x_i} a'_{iy_i}(\lambda), \tag{6.17}$$

so that the discretized version of the minimization problem (6.12) can be written as

$$\begin{array}{ll} \text{minimize} & H^\delta(x) + \lambda R^\delta(0) + \sum_{i=1}^n \sum_{y_i=1}^{x_i} a'_{iy_i}(\lambda) \\ \text{s.t.} & x \in \prod_{i=1}^n [k_i]. \end{array} \tag{6.18}$$

Since the term $R^\delta(0)$ does not depend on the variable x, this minimization is equivalent to

$$\begin{array}{ll} \text{minimize} & H^\delta(x) + \sum_{i=1}^n \sum_{y_i=1}^{x_i} a'_{iy_i}(\lambda) \\ \text{s.t.} & x \in \prod_{i=1}^n [k_i]. \end{array} \tag{6.19}$$

This problem is in the precise form where we can apply Theorem 6.4.1 to show equivalence between Problems (6.14) and (6.15), so we are done. □

Problem (6.15) can be solved by Frank-Wolfe methods (Frank and Wolfe, 1956; Dunn and Harshbarger, 1978; Lacoste-Julien, 2016; Jaggi, 2013). This is because the greedy algorithm for computing subgradients of the Lovász extension can be generalized, and yields a linear optimization oracle for the dual of Problem (6.15). We detail the relationship between Problems (6.12) and (6.15), as well as how to implement the Frank-Wolfe methods, in Appendix C.3.1.

6.4.2 Bounding solution quality for the constrained problem

We now have a tractable convex formulation, Equation (6.15), of the *regularized* problem. But it is not yet clear if we can also recover a good solution to the original *constrained* problem.

Let ρ^* be the optimal solution for Problem (6.15). For any λ, we obtain a rounded solution $x(\lambda)$ for Problem (6.12) by thresholding: we set $x(\lambda)_i = \max\{j \mid 1 \leq j \leq k_i - 1,\ \rho_i^*(j) \geq \lambda\}$, or zero if $\rho_i^*(j) < \lambda$ for all j. Each $x(\lambda')$ is the optimal solution for Problem (6.12) with $\lambda = \lambda'$. We use the largest parameterized solution $x(\lambda)$ that is still feasible, i.e. the solution $x(\lambda^*)$ where λ^* solves

$$\begin{aligned} \min \quad & H^\delta(x(\lambda)) \\ \text{s.t.} \quad & \lambda \geq 0 \\ & R^\delta(x(\lambda)) \leq B. \end{aligned} \tag{6.20}$$

This λ^* can be found efficiently via binary search or a linear scan.

Theorem 6.4.2. *Let H be continuous submodular and monotone decreasing, with ℓ_∞-Lipschitz constant G, and let R be strictly increasing and separable. Assume all entries $\rho_i^*(j)$ of the optimal solution ρ^* of Problem (6.15) are distinct. Let $x' = A(x(\lambda^*))$ be the thresholding corresponding to the optimal solution λ^* of Problem (6.20), mapped back into the original continuous domain $\mathcal{X}$. Then x' is feasible for the continuous Problem (6.10), and is a $2G\delta$-approximate solution:*

$$H(x') \leq 2G\delta + \min_{x \in \text{Box}(l,u),\ R(x) \leq B} H(x).$$

Theorem 6.4.2 implies an algorithm for solving Problem (6.10) to τ-optimality: (1) set $\delta = \tau/G$, (2) compute ρ^* that solves Problem (6.15), (3) find the optimal thresholding of ρ^* by determining the smallest λ^* for which $R^\delta(x(\lambda^*)) \leq B$, and (4) map $x(\lambda^*)$ back into continuous space via the interpolation mapping A.

The general idea of this proof is to first show that the best integer-valued point

x_d^* that solves

$$x_d^* \in \operatorname*{argmin}_{x \in \prod_{i=1}^n [k_i] : R^\delta(x) \leq B} H^\delta(x)$$

is also nearly a minimizer of the continuous version of the problem, due to the fineness of the discretization. Then, we show that the solutions traced out by $x(\lambda)$ get very close to x_d^*. These two results are simply combined via the triangle inequality.

We begin with a Lemma bounding the optimal discrete solution by the optimal continuous solution:

Lemma 6.4.2. *With x_d^* defined as above,*

$$H^\delta(x_d^*) \leq G\delta + \min_{x \in \mathcal{X} : R(x) \leq B} H(x). \tag{6.21}$$

Proof. Consider $x^* \in \arg\min_{x \in \mathcal{X} : R(x) \leq B} H(x)$. If x^* corresponds to an integral point in the discretized domain, then $H(x^*) = H^\delta(x_d^*)$ and we are done. Else, since our discretization is δ-fine, we can find a discrete point x_{floor} with $x^* - \delta \leq A(x_{\text{floor}}) \leq x^*$ elementwise. Algorithmically, x_{floor} is a kind of elementwise floor of x^* with respect to the discretization. There are two implications of the bound between $A(x_{\text{floor}})$ and x^*: first, by monotonicity, $R^\delta(x_{\text{floor}}) \leq B$, i.e. $A(x_{\text{floor}})$ is feasible for the original continuous problem; second, we must have $\|x^* - A(x_{\text{floor}})\|_\infty \leq \delta$. Applying the Lipschitz property of H and then the optimality of x_d^*, we have

$$G\delta \geq H(A(x_{\text{floor}})) - H(x^*) = H^\delta(x_{\text{floor}}) - H(x^*) \geq H^\delta(x_d^*) - H(x^*),$$

from which (6.21) follows. □

The next step in proving our suboptimality bound is to bound the suboptimality of our thresholded solutions relative to the true discrete solution:

Lemma 6.4.3. *Define λ_- and λ_+ by*

$$\lambda_- \in \operatorname*{argmin}_{\lambda \geq 0 : R^\delta(x(\lambda)) \leq B} H^\delta(x(\lambda)) \quad \text{and} \quad \lambda_+ \in \operatorname*{argmax}_{\lambda \geq 0 : R^\delta(x(\lambda)) \geq B} H^\delta(x(\lambda)).$$

Then, we can bound the discrete optimal value $H^\delta(x_d^*)$ on both sides by

$$H^\delta(x(\lambda_+)) \le H^\delta(x_d^*) \le H^\delta(x(\lambda_-)). \tag{6.22}$$

Proof. Note that

$$\min_{x\in\prod_{i=1}^n[k_i]:R^\delta(x)\le B} H^\delta(x) = \min_{x\in\prod_{i=1}^n[k_i]} \max_{\lambda\ge 0} \left\{H^\delta(x) + \lambda(R^\delta(x) - B)\right\}, \tag{6.23}$$

since either the term $R^\delta(x)-B$ does not contribute, or it blows up when x is infeasible. Continuing, we can bound:

$$\begin{aligned}
\min_{x\in\prod_{i=1}^n[k_i]:R^\delta(x)\le B} H^\delta(x) &= \min_{x\in\prod_{i=1}^n[k_i]} \max_{\lambda\ge 0} \left\{H^\delta(x) + \lambda(R^\delta(x) - B)\right\} && (6.24)\\
&\overset{(a)}{\ge} \max_{\lambda\ge 0} \min_{x\in\prod_{i=1}^n[k_i]} \left\{H^\delta(x) + \lambda(R^\delta(x) - B)\right\} && (6.25)\\
&\overset{(b)}{=} \max_{\lambda\ge 0} \left\{H^\delta(x(\lambda)) + \lambda(R^\delta(x(\lambda)) - B)\right\} && (6.26)\\
&\overset{(c)}{\ge} \max_{\lambda\ge 0:R^\delta(x(\lambda))\ge B} \left\{H^\delta(x(\lambda)) + \lambda(R^\delta(x(\lambda)) - B)\right\} && (6.27)\\
&\overset{(d)}{\ge} \max_{\lambda\ge 0:R^\delta(x(\lambda))\ge B} H^\delta(x(\lambda)) && (6.28)\\
&\overset{(e)}{=} H^\delta(x(\lambda_+)), && (6.29)
\end{aligned}$$

where (a) uses weak duality, (b) plugs in the definition of $x(\lambda)$, (c) shrinks the set of candidate λ, (d) bounds the regularizing term by zero, and (e) is the definition of $x(\lambda_+)$. We can also bound the optimal value of $H^\delta(x_d^*)$ from the other side:

$$H^\delta(x_d^*) = \min_{x\in\prod_{i=1}^n[k_i]:R^\delta(x)\le B} H^\delta(x) \le \min_{\lambda\ge 0:R^\delta(x(\lambda))\le B} H^\delta(x(\lambda)) = H^\delta(x(\lambda_-)) \tag{6.30}$$

because the set of $x(\lambda)$ parameterized by λ is a subset of the full set $\{x \in \prod_{i=1}^n[k_i] : R^\delta(x) \le B\}$. □

Corollary 6.4.1. *In the same setting as Lemma 6.4.3, it holds that*

$$H^\delta(x(\lambda_-)) \le G\delta + H^\delta(x_d^*).$$

Proof. Via Lemma 6.4.3 we can bound the optimal value of $H^\delta(x_d^*)$ on either side by optimization problems where we seek an optimal $\lambda \geq 0$ for the parameterization $x(\lambda)$:

$$H^\delta(x(\lambda_+)) \leq H^\delta(x_d^*) \leq H^\delta(x(\lambda_-)). \tag{6.31}$$

Recall that $x(\lambda)$ comes from thresholding the values of ρ^* by λ, and that we assume that the elements of ρ^* are unique. Hence, as we increase λ, the components of x decrease, in steps of one. Combining this with the strict monotonicity of R, we see that $\|x(\lambda_+) - x(\lambda_-)\|_\infty \leq 1$. By the Lipschitz properties of H^δ, it follows that $\left|H^\delta(x(\lambda_+)) - H^\delta(x(\lambda_-))\right| \leq G\delta$. Since $H^\delta(x_d^*)$ lies in the interval between $H^\delta(x(\lambda_+))$ and $H^\delta(x(\lambda_-))$, it follows that $\left|H^\delta(x_d^*) - H^\delta(x(\lambda_-))\right| \leq G\delta$. □

With the above technical results in place, we can easily prove Theorem 6.4.2:

Theorem 6.4.2. We now combine Lemma 6.4.2 and Corollary 6.4.2. We have that

$$\begin{aligned} H(x') &\overset{(a)}{=} H^\delta(x(\lambda_-)) && (6.32)\\ &\overset{(b)}{\leq} G\delta + H^\delta(x_d^*) && (6.33)\\ &\overset{(c)}{\leq} G\delta + \left(G\delta + \min_{x \in \text{Box}(l,u),\ R(x) \leq B} H(x)\right) && (6.34)\\ &= 2G\delta + \min_{x \in \text{Box}(l,u),\ R(x) \leq B} H(x), && (6.35) \end{aligned}$$

where (a) is the definition of x', (b) follows from Lemma 6.4.2 and (c) follows from Corollary 6.4.2. □

Computable Optimality Bounds

Beyond the theoretical guarantee of Theorem 6.4.2, for any problem instance and candidate solution x', we can compute bounds on the gap between $H(x')$ and $H^\delta(x_d^*)$:

1. The discrete point $x(\lambda_+)$ yields the bound

$$H(x') \leq [H(x') - H^\delta(x(\lambda_+))] + H^\delta(x_d^*). \tag{6.36}$$

2. The Lagrangian yields the bound

$$H(x') \leq \lambda^*(B - R(x')) + H^\delta(x_d^*). \tag{6.37}$$

The first bound is a simple consequence of Lemma 6.4.3:

$$H^\delta(x(\lambda_+)) \leq H^\delta(x_d^*) \tag{6.38}$$

$$\implies 0 \leq -H^\delta(x(\lambda_+)) + H^\delta(x_d^*) \tag{6.39}$$

$$\implies H(x') \leq H(x') - H^\delta(x(\lambda_+)) + H^\delta(x_d^*). \tag{6.40}$$

As for the Lagrangian bound, since $x(\lambda^*)$ is a minimizer for the regularized function $H^\delta(x) + \lambda^*(R^\delta(x) - B)$, it follows that

$$H^\delta(x(\lambda^*)) + \lambda^*(R^\delta(x(\lambda^*)) - B) \leq H^\delta(x_d^*) + \lambda^*(R^\delta(x_d^*) - B). \tag{6.41}$$

Rearranging, and observing that $R^\delta(x_d^*) \leq B$ because x_d^* is feasible, it holds that

$$H(x') = H^\delta(x(\lambda^*)) \tag{6.42}$$

$$\leq H^\delta(x_d^*) + \lambda^*(R^\delta(x_d^*) - R^\delta(x(\lambda^*))) \tag{6.43}$$

$$\leq H^\delta(x_d^*) + \lambda^*(B - R(x')). \tag{6.44}$$

One can also combine either of these bounds with the result from the proof of Theorem 6.4.2 that $H^\delta(x_d^*) \leq G\delta + H(x^*)$ yielding e.g.

$$H(x') \leq G\delta + \lambda^*(B - R(x')) + H^\delta(x^*). \tag{6.45}$$

Improvements

The requirement in Theorem 6.4.2 that the elements of ρ^* be distinct may seem somewhat restrictive, but as long as ρ^* has distinct elements in the neighborhood of our particular λ^*, this bound still holds. We see in Section 6.6.1 that in practice, ρ^*

almost always has distinct elements in the regime we care about, and the bounds of Remark 6.4.2 are very good.

If H is DR-submodular and R is affine in each coordinate, then Problem (6.12) can be represented more compactly via the reduction of Ene and Nguyen (2016), and hence problem (6.10) can be solved more efficiently. In particular, the influence function $\mathcal{I}(y; x)$ is DR-submodular in x when for each s, $y(s) = 0$ or $y(s) \geq 1$.

Application to Robust Budget Allocation

The above algorithm directly applies to Robust Allocation with the uncertainty sets in Section 6.2.2. The ellipsoidal uncertainty set $\mathcal{X}^Q$ corresponds to the constraint that $\sum_{(s,t)\in E} R_{st}(x_{st}) \leq \gamma$ with $R_{st}(x) = (x_{st} - \hat{x}_{st})^2 \sigma_{st}^{-2}$, and $x \in \text{Box}(0, 1)$. By the monotonicity of $\mathcal{I}(x, y)$, there is never incentive to reduce any x_{st} below $\hat{x}_{st}$, so we can replace $\text{Box}(0, 1)$ with $\text{Box}(\hat{x}, 1)$. On this interval, each R_{st} is strictly increasing, and Theorem 6.4.2 applies.

For D-norm sets, we have $R_{st}(x_{st}) = (x_{st} - \hat{x}_{st})/(u_{st} - \hat{x}_{st})$. Since each R_{st} is monotone, Theorem 6.4.2 applies.

Runtime and Alternatives

The core part of the algorithm uses Frank-Wolfe to optimize the regularized convex extension, Problem (6.15). This convex problem can be solved to τ-suboptimality in time $O(\tau^{-1} n^2 \delta^{-3} \alpha^{-1} |T|^2 \log n\delta^{-1})$, where α is the minimum derivative of the functions R_i (see Appendix C.3.2 for details). Suppose that the optimal solution ρ^* to Problem (6.15) has distinct elements separated by η; then choosing $\tau = \eta^2 \alpha \delta / 8$ results in an exact solution to the discrete regularized Problem (6.12) in total time $O(\eta^{-2} n^2 \delta^{-4} \alpha^{-2} |T|^2 \log n\delta^{-1})$.

Noting that $H^\delta + \lambda R^\delta$ is submodular for all λ, one could instead perform binary search over λ, each time converting the objective into a submodular *set* function via Birkhoff's theorem and solving submodular minimization e.g. via a fast, recent method (Chakrabarty et al., 2017; Lee et al., 2015). However, we are not aware of a practical implementation of the algorithm in (Lee et al., 2015). The algorithm

in (Chakrabarty et al., 2017) yields a solution only in expectation. This approach also requires care in the precision of the search over λ, whereas our approach solves for all λ simultaneously, and picks directly from the $O(n\delta^{-1})$ elements of ρ^*.

A host of alternate approaches are also possible, e.g. a generalization of the minimum norm point algorithm (Wolfe, 1976; Fujishige, 2005) which is also suggested by Bach (2019). However such development is out of scope for this work: our focus is on developing a convex formulation, rather than optimizing the algorithm.

6.5 Simple examples where our approach is optimal

Next, we take a view beyond Budget Allocation, and theoretically and empirically evaluate the optimality of our constrained submodular minimization algorithm on two classes of nonconvex problem where the optimal solution can be computed. For one class, the algorithm provably yields the global optimal solution. For the other class, the algorithm empirically yields solutions that are very close to globally optimal solutions from a specialized SDP relaxation.

6.5.1 Separable problems

First, we assume that the objective function $H(x)$ and constraint function $R(x)$ are continuous submodular and *separable*. Some such problems admit simple analytic solutions despite nonconvexity. Our approach will recover these solutions. To understand how our method behaves when the objective and constraints are separable, the following structural result about the convex extension will be useful.

Lemma 6.5.1 (also appears informally in (Bach, 2019))**.** *Suppose the objective is separable:* $H(x) = \sum_{i=1}^{n} H_i(x_i)$. *Then the convex extension* $h_\downarrow(\mu_1, \ldots, \mu_n)$ *is also separable:*

$$h_\downarrow(\mu_1, \ldots, \mu_n) = \sum_{i=1}^{n} h_{i\downarrow}(\mu_i), \tag{6.46}$$

where $h_{i\downarrow}$ is the extension for H_i.

Note that this separability also holds for any block-separable structure.

Proof. First we write out the definition of the convex extension $h_\downarrow(\mu)$:

$$h_\downarrow(\mu_1, \ldots, \mu_n) = \inf_{\gamma \in \mathcal{P}(\mathcal{X}, \{\mu_i\})} \int_{\mathcal{X}} H(x)\, d\gamma(x) \tag{6.47}$$

$$= \inf_{\gamma \in \mathcal{P}(\mathcal{X}, \{\mu_i\})} \int_{\mathcal{X}} \sum_{i=1}^{n} H_i(x_i)\, d\gamma(x) \tag{6.48}$$

$$= \inf_{\gamma \in \mathcal{P}(\mathcal{X}, \{\mu_i\})} \sum_{i=1}^{n} \int_{\mathcal{X}} H_i(x_i)\, d\gamma(x). \tag{6.49}$$

Each integral of $H_i(x_i)$ only depends on the marginal of γ, which by definition is μ_i. Since this is the only dependence on γ, the infimum is now unnecessary:

$$h_\downarrow(\mu_1, \ldots, \mu_n) = \sum_{i=1}^{n} \int_{\mathcal{X}_i} H_i(x_i)\, d\mu_i(x_i) = \sum_{i=1}^{n} h_{i\downarrow}(\mu_i). \qquad \square$$

Write $\Delta H_i(x_i) = H_i(x_i) - H_i(x_i - 1)$ and similarly for $R_i(x_i)$. Using Lemma 6.5.1 we can prove the following structural result:

Proposition 6.5.1. *Suppose $H(x)$ and $R(x)$ are both separable as above. Consider the regularized problem:*

$$\begin{array}{ll} \text{minimize} & h_\downarrow(\rho) + \sum_{i=1}^{n} \sum_{x_i=1}^{k_i - 1} a_{i x_i}(\rho_i(x_i)) \\ \text{s.t.} & \rho \in \prod_{i=1}^{n} \mathbb{R}_\downarrow^{k_i - 1}. \end{array} \tag{6.50}$$

If $Q_i(x_i) := \frac{\Delta H_i(x_i)}{\Delta R_i(x_i)}$ is nondecreasing in x_i for all i, then the optimal solution for Problem (6.50) is given by $\rho_i^*(x_i) = -Q_i(x_i)$.

Proof. By Lemma 6.5.1 we have $h_\downarrow(\rho) = \sum_{i=1}^{n} h_{i\downarrow}(\rho_i)$. In the single dimensional case, $h_{i\downarrow}$, the extension is very easy to compute, as it is given by $h_{i\downarrow}(\mu_i) = \int_{\mathcal{X}_i} H_i(x_i)\ d\mu_i(x_i)$. We instead use the alternative characterization of $h_{i\downarrow}$ in terms of the reversed cumulative distribution function ρ_i. In the discrete case, we write $\rho_i(x_i) = \mu_i(x_i) + \mu_i(x_i +$

$1) + \cdots + \mu_i(k_i - 1)$, and so $h_{i\downarrow}(\rho_i)$ is given by:

$$h_{i\downarrow}(\rho_i) = H_i(0) + \sum_{x_i=1}^{k_i-1} \rho_i(x_i)(H_i(x_i) - H_i(x_i - 1)) \tag{6.51}$$

$$= H_i(0) + \sum_{x_i=1}^{k_i-1} \rho_i(x_i)\Delta H_i(x_i). \tag{6.52}$$

We assumed the constraint functions $R(x) = \sum_{i=1}^{n} R_i(x_i)$ are separable over i. As in the proof of Lemma 6.4.1, we convert each R_i into strongly convex functions $a_{ix_i}(t) = \frac{1}{2}t^2\Delta R_i(x_i)$. Since the convex extension $h_\downarrow$ is now also separable, as are the monotonicity constraints, we may separately consider n problems of the form:

$$\begin{array}{ll} \text{minimize} & h_{i\downarrow}(\rho_i) + \sum_{x_i=1}^{k_i-1} a_{ix_i}(\rho_i(x_i)) \\ \text{s.t.} & \rho_i \in \mathbb{R}_{\downarrow}^{k_i-1} \end{array} \tag{6.53}$$

The first term $h_{i\downarrow}(\rho_i)$ is also a sum over x_i, so we may rewrite the objective as:

$$\sum_{x_i=1}^{k_i-1} \rho_i(x_i)\Delta H_i(x_i) + \frac{1}{2}\sum_{x_i=1}^{k_i-1}[\rho_i(x_i)]^2\Delta R_i(x_i) \tag{6.54}$$

$$= \sum_{x_i=1}^{k_i-1} \left\{ \rho_i(x_i)\Delta H_i(x_i) + \frac{1}{2}[\rho_i(x_i)]^2\Delta R_i(x_i) \right\}. \tag{6.55}$$

Completing the square, we may write

$$\rho_i(x_i)\Delta H_i(x_i) + \frac{1}{2}[\rho_i(x_i)]^2\Delta R_i(x_i) \tag{6.56}$$

$$= \frac{\Delta R_i(x_i)}{2}\left(\rho_i(x_i) + \frac{\Delta H_i(x_i)}{\Delta R_i(x_i)}\right)^2 - \frac{\Delta R_i}{2}\cdot\left(\frac{\Delta H_i(x_i)}{\Delta R_i(x_i)}\right)^2. \tag{6.57}$$

The last term is a constant that does not depend on ρ_i, so we ignore it. Using, in addition, the identity $Q_i(x_i) = \frac{\Delta H_i(x_i)}{\Delta R_i(x_i)}$, the problem we wish to solve is:

$$\begin{array}{ll} \text{minimize} & \sum_{x_i=1}^{k_i-1} \Delta R_i(x_i)\left(\rho_i(x_i) + Q_i(x_i)\right)^2 \\ \text{s.t.} & \rho_i \in \mathbb{R}_{\downarrow}^{k_i-1} \end{array}. \tag{6.58}$$

This is a weighted isotonic regression problem. We try to fit $\rho_i(x_i)$ to the value $-Q_i(x_i)$, with associated weight $\Delta R_i(x_i)$. We assumed $Q_i(x_i)$ is nondecreasing, so $-Q_i(x_i)$ is nonincreasing. Therefore setting $\rho_i^*(x_i) = -Q_i(x_i)$ is feasible and obtains optimal objective value (zero). □

There are many situations in which $Q_i(x_i)$ is nondecreasing, and therefore $\rho_i^*(x_i) = -Q_i(x_i)$. We focus on a particularly simple one: let $H_i(x_i) = a_i x_i^p$ and $R_i(x_i) = r_i x_i^p$, so that the overall optimization problem is

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n a_i x_i^p \\ \text{s.t.} & \sum_{i=1}^n r_i x_i^p \leq B \\ & \mathbf{0} \leq x \leq \mathbf{1}. \end{array} \tag{6.59}$$

The problem might be nonconvex in its current form, but the transformation $y_i = r_i x_i^p$ gives a convex problem:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n \frac{a_i}{r_i} \cdot y_i \\ \text{s.t.} & \mathbf{1}^T y \leq B \\ & \mathbf{0} \leq y \leq r. \end{array} \tag{6.60}$$

The constraint here is a scaled version of the simplex. An optimal solution can be found by sorting the indices so $\frac{a_1}{r_1} \leq \cdots \leq \frac{a_n}{r_n}$, and saturating $y_1, y_2, \ldots$ in order until the total budget B is achieved.

This procedure is precisely equivalent to what our algorithm will do, even without the convex reparameterization. In our case,

$$Q_i(x_i) = \frac{a_i}{r_i} \cdot \frac{x_i^p - (x_i - 1)^p}{x_i^p - (x_i - 1)^p} = \frac{a_i}{r_i} \tag{6.61}$$

is constant, hence nondecreasing. Therefore $\rho_i^*(x_i) = -\frac{a_i}{r_i}$ solves Problem (6.50). Consider now the thresholding step of our algorithm, where we search over λ and take $\rho_i^*(x_i)$ only if $\rho_i^*(x_i) \geq \lambda$. Since $\rho_i^*(x_i) = \rho_i^*$ is constant, we will take entire coordinates at a time: we will first find the coordinate i with $-Q_i = -\frac{a_i}{r_i}$ maximized,

i.e. $\frac{a_i}{r_i}$ minimized. We set $x_i = k_i - 1$ (which maps back to $x_i = 1$ when we undiscretize). Then we move onto the next best coordinate, and so on.

Proposition 6.5.1 confirms at least for this special case that our algorithm finds the optimal solution. Moreover, we can find the optimal solution without using a more specialized approach that depends on e.g. quadratic problem structure.

6.5.2 Non-separable quadratics and SDP relaxations

Reasoning about guaranteed performance gets more difficult as we move away from separable problems. Even empirical evaluation becomes problematic in nonconvex settings where all we can know about the globally optimal solution is the suboptimality bound returned by our algorithm. Before addressing these harder regimes, we study a harder class of nonconvex problems that still admit global optimality guarantees. Specifically we look at a certain class of nonconvex quadratically-constrained quadratic problems:

$$\begin{aligned} \min_x \quad & \frac{1}{2}x^T Ax + c^T x \\ \text{s.t.} \quad & x \in \text{Box}(0,1) \\ & \frac{1}{2}x^T \text{diag}(r)x \leq B \end{aligned} \qquad \Leftrightarrow \qquad \begin{aligned} \min_x \quad & \frac{1}{2}x^T Ax + c^T x \\ \text{s.t.} \quad & x_i^2 \leq x_i \; \forall i \\ & \frac{1}{2}x^T \text{diag}(r)x \leq B, \end{aligned}$$

where A has nonpositive off-diagonal entries. These problems are a useful benchmark because they can be solved globally via an SDP relaxation (Kim and Kojima, 2003). Concretely:

Theorem 6.5.1 (Theorem 4 from Kim and Kojima (2003))**.** *Let A have nonpositive*

off-diagonal entries. Let (X^*, x^*) *be a solution to the SDP*

$$\begin{aligned} \min_{X,x} \quad & \frac{1}{2}\operatorname{tr}(AX) + c^T x \\ \text{s.t.} \quad & \operatorname{diag}(X) \leq x \\ & \frac{1}{2}\operatorname{tr}(\operatorname{diag}(r)X) \leq B \\ & \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0. \end{aligned}$$

Taking z^* *to be the elementwise square root of* $\operatorname{diag}(X^*)$ *yields an optimal solution to the original nonconvex problem.*

We emphasize that this is a very special subclass of constrained submodular problems: the SDP approach only applies when both the objective and constraint are quadratic, while our algorithm applies more broadly.

We compare our constrained submodular optimization algorithm to the SDP relaxation on random problems. For our algorithm, we discretize each coordinate into $k = 1000$ pieces, and run only 300 iterations of Frank-Wolfe on the convex relaxation (6.15). The matrix A is set to $M + M^T$, where each entry of M is sampled uniformly in $[-1, 0]$. The linear term c is set to zero, the constraint vector r is set to the all ones vector, and we vary B. Figure 6-2 shows histograms of the gap in performance between the two algorithms. For most instances our approach does nearly as well as the globally optimal SDP solution.

In fact, if A is restricted to be a diagonal matrix (we take only its diagonal part), our algorithm always achieved a relative suboptimality gap below 10^{-4}. As predicted by Proposition 6.5.1, for separable problems our algorithm is essentially optimal.

6.5.3 Evaluation of suboptimality bounds

In Section 6.4.2 we give solution-dependent suboptimality bounds for the algorithm. These require only the discrete solution, (possibly) the optimal Lagrange multiplier, the granularity δ (chosen to be 0.001 in these experiments), and the ℓ_∞ Lipschitz

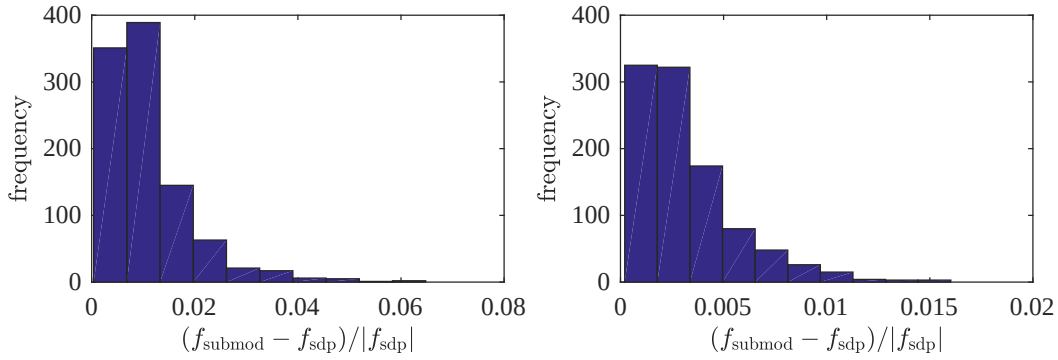


Figure 6-2: Relative suboptimality of the submodular optimization solution (objective value f_{submod}) vs the globally optimal SDP solution (objective value f_{sdp}). Quadratic constraints $\|x\|^2 \leq B$ with $B = 0.1$ (left) and $B = 1$ (right).

constant of $H(x) = \frac{1}{2}x^T Ax$. Since $x \leq 1$ elementwise, we can equivalently bound the ℓ_∞ norm of the linear map $x \mapsto \frac{1}{2}\mathbf{1}^T Ax$, which is just $\frac{1}{2}\|\mathbf{1}^T A\|_1$. Since the bounds do assume access to the optimal ρ^* for Problem (6.15), it is important to run Frank-Wolfe for enough iterations to get a good approximation to ρ^*.

For each quadratic experiment from the previous section, we compute the best available suboptimality bound. Here, even 300 iterations were easily sufficient to approximate ρ^* for this purpose. For the quadratic problems, the bounds from Section 6.4.2 were typically ≈ 0.1 for $B = 0.1$ and ≈ 0.01 for $B = 1$. The computed bounds were always an upper bound on the true suboptimality in our experiments.

6.6 Robust Budget Allocation experiments

After testing the core submodular minimization subroutine, we return to the motivating application of Robust Budget Allocation. We evaluate our algorithm on both synthetic test data and a real-world bidding dataset from Yahoo! Webscope yah to demonstrate that our method yields real improvements. For all experiments, we used Algorithm 3 as the outer loop. For the inner submodular minimization step, we implemented the pairwise Frank-Wolfe algorithm of Lacoste-Julien and Jaggi (2015). In all cases, the feasible set of budgets $\mathcal{Y}$ is $\{y \in \mathbb{R}^S_+ : \sum_{s \in S} y(s) \leq C\}$ where the specific budget C depends on the experiment. Our code is available at `git.io/vHXkO`.

6.6.1 Synthetic

On the synthetic data, we probe two questions: (1) how often does the distinctness condition of Theorem 6.4.2 hold, so that we are guaranteed an optimal solution; and (2) what is the gain of using a robust versus non-robust solution in an adversarial setting? For both settings, we set $|S| = 6$ and $|T| = 2$ and discretize with $\delta = 0.001$. We generated true probabilties p_{st}, created Beta posteriors, and built both Ellipsoidal uncertainty sets $\mathcal{X}^Q(\gamma)$ and D-norm sets $\mathcal{X}^D(\gamma)$.

Optimality

Theorem 6.4.2 and Remark 6.4.2 demand that the values $\rho_i^*(j)$ be distinct at our chosen Lagrange multiplier λ^* and, under this condition, guarantee optimality. We illustrate this in four examples: for Ellipsoidal or a D-norm uncertainty set, and a total influence budget $C \in \{0.4, 4\}$. Figure 6-3 shows all elements of ρ^* in sorted order, as well as a horizontal line indicating our Lagrange multiplier λ^* which serves as a threshold. Despite some plateaus, the entries $\rho_i^*(j)$ are distinct in most regimes, in particular around λ^*, the regime that is needed for our results. Moreover, in practice (on the Yahoo data) we observe later in Figure 6-5 that both solution-dependent bounds from Remark 6.4.2 are very good, and all solutions are optimal within a very small gap.

Robustness and Quality

Next, we probe the effect of a robust versus non-robust solution for different uncertainty sets and budgets γ of the adversary. We compare our robust solution with using a point estimate for x, i.e., $y_{\text{nom}} \in \text{argmax}_{y \in \mathcal{Y}} \mathcal{I}(y; \hat{x})$, treating estimates as ground truth, and the stochastic solution $y_{\text{expect}} \in \text{argmax}_{y \in \mathcal{Y}} \mathbb{E}[\mathcal{I}(y; X)]$ as per Section 6.2.1. These two optimization problems were solved via standard first-order methods using TFOCS (Becker et al., 2011).

Figure 6-4 demonstrates that indeed, the alternative budgets are sensitive to the adversary and the robustly-chosen budget y_{robust} performs better, even in cases where

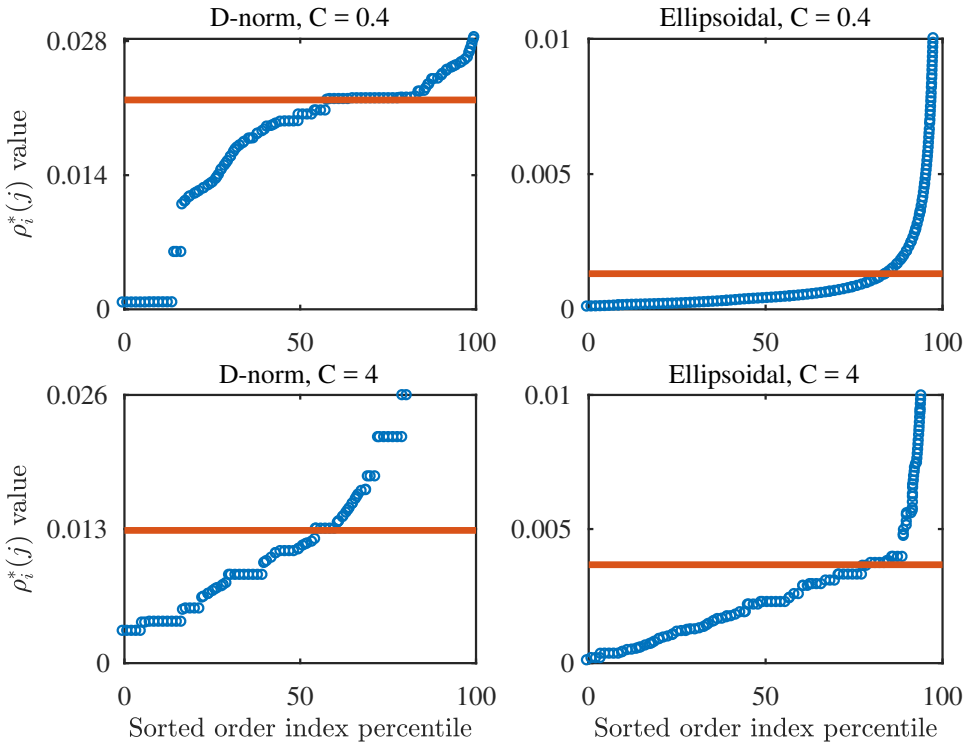


Figure 6-3: Visualization of the sorted values of $\rho_i^*(j)$ (blue dots) with comparison to the particular Lagrange multiplier λ^* (orange line). In most regimes there are no duplicate values, so that Theorem 6.4.2 applies. The theorem only needs distinctness at λ^*.

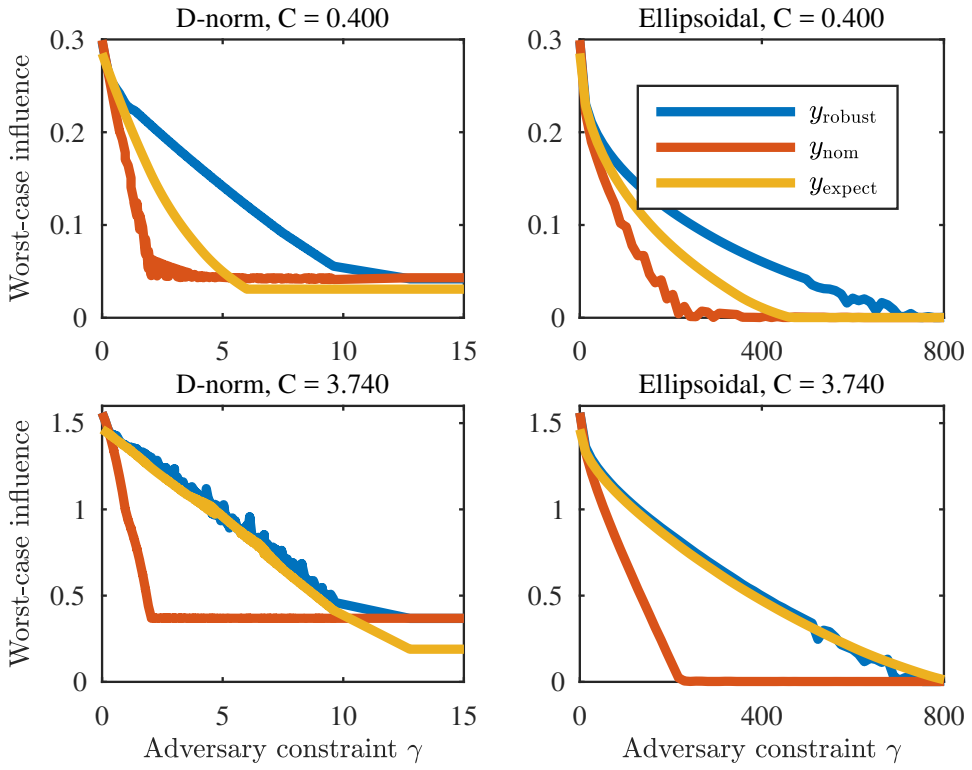


Figure 6-4: Comparison of worst-case expected influences for D-norm uncertainty sets $\mathcal{X}^D(\gamma)$ (left) and ellipsoidal uncertainty sets $\mathcal{X}^Q(\gamma)$ (right), for different total budget bounds C. For any particular adversary budget γ, we compare $\min_{x \in \mathcal{X}(\gamma)} \mathcal{I}(y; x)$ for each candidate allocation y.

the other budgets achieve zero influence. When the total budget C is large, y_{expect} performs nearly as well as y_{robust}, but when resources are scarce (C is small) and the actual choice seems to matter more, y_{robust} performs far better.

6.6.2 Yahoo! data

To evaluate our method on real-world data, we formulate a Budget Allocation instance on advertiser bidding data from Yahoo! Webscope (yah). This dataset logs bids on 1000 different phrases by advertising accounts. We map the phrases to channels S and the accounts to customers T, with an edge between s and t if a corresponding bid was made. For each pair (s, t), we draw the associated transmission probability p_{st} uniformly from $[0, 0.4]$. We bias these towards zero because we expect people not to be easily influenced by advertising in the real world. We then generate an estimate

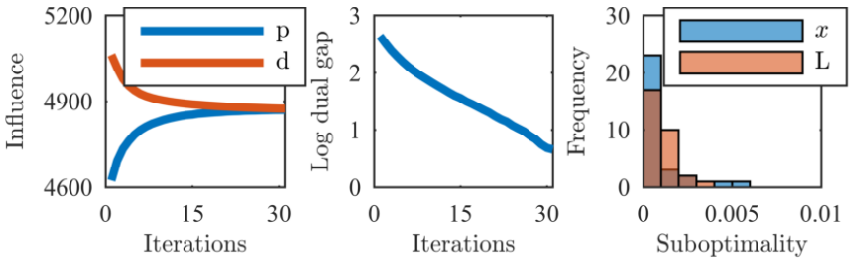


Figure 6-5: Convergence properties of our algorithm on real data. In the first plot, 'p' and 'd' refer to primal and dual values, with dual gap shown on the second plot. The third plot demonstrates that the problem-dependent suboptimality bounds of Remark 6.4.2 (x for $x(\lambda_+)$ and L for Lagrangian) are very small (good) for all inner iterations of this run.

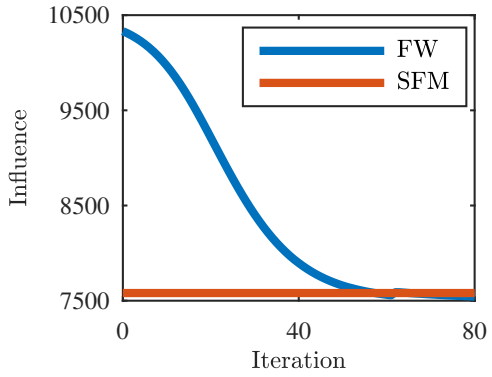


Figure 6-6: Convergence properties of Frank-Wolfe (FW), versus the optimal value attained with our scheme (SFM).

$\hat{p}_{st}$ and build up a posterior by generating n_{st} samples from $\text{Ber}(p_{st})$, where n_{st} is the number of bids between s and t in the dataset.

This transformation yields a bipartite graph with $|S| = 1000$, $|T| = 10475$, and more than 50,000 edges that we use for Budget Allocation. In our experiments, the typical gap between the naive y_{nom} and robust y_{robust} was 100-500 expected influenced people. We plot convergence of the outer loop in Figure 6-5, where we observe fast convergence of both primal influence value and the dual bound.

6.6.3 Comparison to first-order methods

Given the success of first-order methods on nonconvex problems in practice, it is natural to compare these to our method for finding the worst-case vector x. On

one of our Yahoo problem instances with D-norm uncertainty set, we compared our submodular minimization scheme to Frank-Wolfe with fixed stepsize as in (Lacoste-Julien, 2016), implementing the linear oracle using MOSEK (MOSEK ApS, 2015). Interestingly, from various initializations, Frank-Wolfe finds an optimal solution, as verified by comparing to the guaranteed solution of our algorithm. Note that, due to non-convexity, there are no formal guarantees for Frank-Wolfe to be optimal here, motivating the question of global convergence properties of Frank-Wolfe in the presence of submodularity.

It is important to note that there are many cases where first-order methods are inefficient or do not apply to our setup. These methods require either a projection oracle onto or linear optimization oracle over the feasible set $\mathcal{X}$ defined by ℓ, u and $R(x)$. The D-norm set admits a linear optimization oracle via linear programming, but we are not aware of any efficient linear optimization oracle for Ellipsoidal uncertainty, nor projection oracle for either set, that does not require quadratic programming. Even more, our algorithm applies for nonconvex functions $R(x)$ which induce nonconvex feasible sets $\mathcal{X}$. Such nonconvex sets may not even admit a unique projection, while our algorithm achieves provable solutions.

6.7 Discussion and future work

In this chapter, we address the issue of uncertain parameters (or, model misspecification) in Budget Allocation or Bipartite Influence Maximization (Alon et al., 2012) via robust optimization. The resulting *Robust Budget Allocation* is a nonconvex-concave saddle point problem. Although the inner optimization problem is nonconvex, we show how continuous submodularity can be leveraged to solve the problem to arbitrary accuracy τ, as can be verified with the proposed bounds on the duality gap. In particular, our approach extends continuous submodular minimization methods (Bach, 2019) to more general constraint sets, introducing a mechanism to solve a new class of constrained nonconvex optimization problems. Our method provably performs well on a class of separable nonconvex problems, and empirically well on

nonconvex quadratics. For Robust Budget Allocation, we confirm on synthetic and real data that our method finds high-quality solutions that are robust to parameters varying arbitrarily in an uncertainty set, and scales up to graphs with over 50,000 edges.

There are many compelling directions for further study. The uncertainty sets we use are standard in the robust optimization literature, but have not been applied to e.g. Robust Influence Maximization; it would be interesting to generalize our ideas to general graphs. Finally, despite the inherent nonconvexity of our problem, first-order methods are often able to find a globally optimal solution. Explaining this phenomenon requires further study of the geometry of constrained monotone submodular minimization.

Part III

The reverse: leveraging perturbations for better non-convex optimization algorithms

Chapter 7

Escaping saddle points with Adaptive Gradient Methods and perturbations

7.1 Introduction

In the previous sections, we focused on how to work around data perturbations, by designing algorithms that explicitly guard against them. Our approach has been to explicitly inject perturbations – we will call these *intentional perturbations* – into the learning or decision problem, and then design an algorithm that can handle these perturbations. We have seen that performance is highly dependent on selecting the appropriate type or shape of perturbations: in DRO, MMD, Wasserstein and ϕ-divergence uncertainty sets all have vastly different performance characteristics.

In contrast, in machine learning problems such as empirical risk minimization (ERM), there are often already perturbations – *unintentional perturbations* – in the learning algorithm itself. This is because stochastic first-order methods such as stochastic gradient descent (SGD) subsample. Instead of computing full gradients of the empirical risk $\mathbb{E}_{z\sim\hat{\mathbb{P}}_n}[f(w,z)] = \frac{1}{n}\sum_{i=1}^{n} f(w,z_i)$, which requires summing over n terms, it is much cheaper to sample a smaller subset S of the datapoints and use

the gradient of $\frac{1}{|S|}\sum_{i\in S} f(w, z_i)$ as an approximation. Instead of the full gradient ∇ of the empirical risk, we receive a stochastic gradient $g = \nabla + \xi$, where ξ captures the noise due to subsampling. We say ξ is an *unintentional perturbation* because we do not add it ourselves, thinking it can help performance; rather, it arises due to computational convenience.

We are of course not the first to study subsampling noise: it is a classic issue, and is a core motivation for online and stochastic optimization research. There is an immense body of work on stochastic optimization, and a wide variety of approaches to dealing with the computation versus noise level tradeoff for ERM specifically. We highlight variance reduction techniques such as (Shalev-Shwartz and Zhang, 2013; Schmidt et al., 2017; Defazio et al., 2014; Johnson and Zhang, 2013); for more thorough background, see e.g. (Bottou et al., 2018).

In this chapter we specifically explore how the nature of the subsampling noise ξ affects optimization performance. In the same way that the type of DRO uncertainty set is critical to achieving good performance, so too is managing the shape of the noise ξ crucial to optimization. In this way we draw a connection between intentional perturbations – the focus of Parts I and II – and unintentional perturbations.

7.1.1 Adaptive gradient methods (AGMs)

Adaptive gradient methods (AGMs) are one attempt to improve performance in this noisy stochastic optimization setting. The intuition is that some coordinates of the stochastic gradients are noisier than others, so it may help to have a different learning rate for each coordinate.[1] AGMs set these learning rates adaptively based on past observed stochastic gradients. Though the history of AGMs dates back decades (see e.g. (Jacobs, 1988)), at present, Adagrad (McMahan and Streeter, 2010; Duchi et al., 2011) is perhaps the best known AGM. Adagrad uses the square root of the sum of the outer product of the past gradients to achieve adaptivity. At time step t, Adagrad

[1]More generally, AGMs precondition the stochastic gradient update direction. Most AGMs use per-coordinate learning rates, equivalent to preconditioning by a diagonal matrix. In our work it is actually simpler to first study full-matrix preconditioning, and then to modify the results for the more common diagonal case.

updates the parameters in the following manner:

$$w_{t+1} = w_t - G_t^{-1/2} g_t,$$

where g_t is a noisy stochastic gradient at w_t and $G_t = \sum_{i=1}^{t} g_i g_i^T$. More often, a diagonal version of Adagrad is used due to practical considerations, which effectively yields a per parameter learning rate. In the convex setting, Adagrad achieves provably good performance, especially when the gradients are sparse.

The non-convex case is more mysterious. Adagrad in particular does not work as well as in the convex case: this performance degradation is often attributed to the rapid decay of the learning rate in Adagrad over time, which is a consequence of rapid increase in eigenvalues of the matrix G_t. Instead, two variants of Adagrad, namely Adam (Kingma and Ba, 2015) and its special case, RMSProp (Tieleman and Hinton, 2012), are extremely popular for non-convex problems in deep learning. We focus on RMSProp, a variant of which is outlined in Algorithm 5; RMSProp replaces the sum of the outer products with an exponential moving average (EMA) i.e., $G_t = (1-\beta)\sum_{i=1}^{t} \beta^{t-i} g_i g_i^T$ for some constant $\beta \in (0, 1)$. RMSProp often enjoys better empirical performance than SGD, but we lack a clear idea as to why. In contrast to the convex setting where we know Adagrad can converge faster than SGD, in the non-convex setting there is little theory for AGMs. There is limited work showing convergence of AGMs to first-order stationary points, e.g. (Ward et al., 2019). But there are also counterexamples showing that RMSProp need not converge even for convex problems (Reddi et al., 2018b). Moreover, RMSProp is challenging to analyze due to the EMA and its many terms.

In this chapter, we introduce a much simpler way of thinking about adaptive methods such as Adam and RMSProp. Roughly, adaptive methods try to precondition SGD by some matrix A, e.g. when A is diagonal, A_{ii} corresponds to the effective stepsize for coordinate i. For some choices of A the algorithms do not have oracle access to A, but instead form an estimate $\hat{A} \approx A$. We separate out these two steps, by 1) giving convergence guarantees for an idealized setting where we have access to

A, then 2) proving bounds on the quality of the estimate $\hat{A}$. Our approach makes it possible to effectively intuit about the algorithms, prove convergence guarantees (including second-order convergence), and give insights about how to choose algorithm parameters. It also leads to a number of surprising results, including an understanding of why the Reddi et al. (2018b) counterexample is hard for adaptive methods, why adaptive methods tend to escape saddle points faster than SGD (observed empirically in (Reddi et al., 2018a)), insights into how to tune Adam's parameters, and (to our knowledge) the first *second-order* convergence proof for any adaptive method.

More significantly for the theme of this thesis: our convergence results hinge on the observation that the RMSProp preconditioner A rescales the subsampling noise ξ to enable better optimization.

Contributions: In addition to the aforementioned novel viewpoint, we also make the following key contributions:

- We develop a new approach for analyzing convergence of adaptive methods leveraging the preconditioner viewpoint and by way of disentangling estimation from the behavior of the *idealized* preconditioner.

- We provide *second-order convergence* results for adaptive methods, and as a byproduct, first-order convergence results. To the best of our knowledge, ours is the first work to show second order convergence for any adaptive method.

- We provide theoretical insights on how adaptive methods escape saddle points quickly. In particular, we show that the preconditioner used in adaptive methods leads to isotropic noise near stationary points, which helps escape saddle points faster.

- Our analysis also provides practical suggestions for tuning the exponential moving average parameter β.

7.1.2 Related work

There is an immense amount of work studying nonconvex optimization for machine learning, which is too much to discuss here in detail. Thus, we only briefly discuss two lines of work that are most relevant to our paper here. First, the recent work e.g. (Chen et al., 2019; Reddi et al., 2018b; Zou et al., 2019) to understand and give theoretical guarantees for adaptive methods such as Adam and RMSProp. Second, the technical developments in using first-order algorithms to achieve nonconvex second-order convergence (see Definition 7.2.1) e.g. (Ge et al., 2015; Allen-Zhu and Li, 2018; Jin et al., 2017; Lee et al., 2016).

Nonconvex convergence of adaptive methods. Many recent works have investigated convergence properties of adaptive methods. However, to our knowledge, all these results either require convexity or show only first-order convergence to stationary points. Reddi et al. (2018b) showed non-convergence of Adam and RMSProp in simple convex settings and provided a variant of Adam, called AMSGrad, with guaranteed convergence in the convex setting; Zhou et al. (2018) generalized this to a nonconvex first-order convergence result. Zaheer et al. (2018) showed first-order convergence of Adam when the batch size grows over time. Chen et al. (2019) bound the nonconvex convergence rate for a large family of Adam-like algorithms, but they essentially need to assume the effective stepsize is well-behaved (as in AMSGrad). Agarwal et al. (2019) give a convex convergence result for a full-matrix version of RMSProp, which they extend to the nonconvex case via iteratively optimizing convex functions. Their algorithm uses a fixed sliding window instead of an exponential moving average. Mukkamala and Hein (2017) prove improved convergence bounds for Adagrad in the online strongly convex case; they prove similar results for RMSProp, but only in a regime where it is essentially the same as Adagrad. Ward et al. (2019) give a nonconvex convergence result for a variant of Adagrad which employs an adaptively decreasing single learning rate (not per-parameter). Zou et al. (2019) give sufficient conditions for first-order convergence of Adam.

Nonconvex second order convergence of first order methods. Starting with Ge et al. (2015) there has been a resurgence in interest in giving first-order algorithms that find *second* order stationary points of nonconvex objectives, where the gradient is small and the Hessian is nearly positive semidefinite. Most other results in this space operate in the deterministic setting where we have exact gradients, with carefully injected isotropic noise to escape saddle points. Levy (2016) show improved results for normalized gradient descent. Some algorithms rely on Hessian-vector products instead of pure gradient information e.g. (Agarwal et al., 2017; Carmon et al., 2018); it is possible to reduce Hessian-vector based algorithms to gradient algorithms (Xu et al., 2018; Allen-Zhu and Li, 2018). Jin et al. (2017) improve the dependence on dimension to polylogarithmic. Mokhtari et al. (2018b) work towards adapting these techniques for constrained optimization. Most relevant to our work is that of Daneshmand et al. (2018), who prove convergence of SGD with better rates than Ge et al. (2015). Concurrent with our paper, Fang et al. (2019) give even better rates for SGD. Our work differs in that we provide second-order results for *preconditioned* SGD.

7.2 Notation and definitions

The objective function is f, and the gradient and Hessian of f are ∇f and $H = \nabla^2 f$, respectively. Denote by $w_t \in \mathbb{R}^d$ the iterate at time t, by g_t an unbiased stochastic gradient at w_t and by ∇_t the expected gradient at t. The matrix G_t refers to $\mathbb{E}[g_t g_t^T]$. Denote by $\lambda_{\max}(G)$ and $\lambda_{\min}(G)$ the largest and smallest eigenvalues of G, and $\kappa(G)$ is the condition number $\lambda_{\max}(G)/\lambda_{\min}(G)$ of G. For a vector v, its elementwise p-th power is written v^p. The objective $f(w)$ has global minimizer w^*, and we write $f^* = f(w^*)$. The Euclidean norm of a vector v is written as $\|v\|$, while for a matrix M, $\|M\|$ refers to the operator norm of M. The matrix I is the identity matrix, whose dimension should be clear from context.

Definition 7.2.1 (Second-order stationary point)**.** A (τ_g, τ_h)-stationary point of f is a point w so that $\|\nabla f(w)\| \leq \tau_g$ and $\lambda_{\min}(\nabla^2 f(w)) \geq -\tau_h$, where $\tau_g, \tau_h > 0$.

Algorithm 4 Preconditioned SGD

Input: initial w_0, time T, stepsize η, preconditioner $A(w)$
for $t = 0, \dots, T$ **do**
  $g_t \leftarrow$ stochastic gradient at w_t
  $A_t \leftarrow A(w_t)$ ▷ e.g. $A_t = \mathbb{E}[g_t g_t^T]^{-1/2}$
  $w_{t+1} \leftarrow w_t - \eta A_t g_t$
end for

Algorithm 5 Full-matrix RMSProp

Input: initial w_0, time T, stepsize η, small number $\varepsilon > 0$ for stability
for $t = 0, \dots, T$ **do**
  $g_t \leftarrow$ stochastic gradient
  $\hat{G}_t = \beta \hat{G}_{t-1} + (1-\beta) g_t g_t^T$
  $A_t = (\hat{G}_t + \varepsilon I)^{-1/2}$
  $w_{t+1} \leftarrow w_t - \eta A_t g_t$
end for

As is standard (e.g. Nesterov and Polyak (2006)), we will discuss only $(\tau, \sqrt{\rho\tau})$-stationary points, where ρ is the Lipschitz constant of the Hessian.

7.3 The RMSProp preconditioner

Recall that methods like Adam and RMSProp replace the running sum $\sum_{i=1}^t g_i g_i^T$ used in Adagrad with an exponential moving average (EMA) of the form $(1-\beta)\sum_{i=1}^t \beta^{t-i} g_i g_i^T$, e.g. full-matrix RMSProp is described formally in Algorithm 5. One key observation is that $\hat{G}_t = (1-\beta)\sum_{i=1}^t \beta^{t-i} g_i g_i^T \approx \mathbb{E}[g_t g_t^T] =: G_t$ if β is chosen appropriately; in other words, at time t, the accumulated $\hat{G}_t$ can be seen as an approximation of the true second moment matrix $G_t = \mathbb{E}[g_t g_t^T]$ at the current iterate. Thus, RMSProp can be viewed as preconditioned SGD (Algorithm 4) with the preconditioner being $A_t = G_t^{-1/2}$. In practice, it is too expensive to compute G_t exactly since it requires summing over all training samples. Practical adaptive methods (see Algorithm 5) estimate this preconditioner (or a diagonal approximation) on-the-fly via an EMA.

Before developing our formal results, we will build intuition about the behavior of adaptive methods by studying an idealized adaptive method (IAM) with perfect access to G_t. In the rest of this section, we make use of idealized RMSProp to

answer some simple questions about adaptive methods that we feel have not yet been addressed satisfactorily.

7.3.1 What is the purpose of the preconditioner?

Why should preconditioning by $A = \mathbb{E}[gg^T]^{-1/2}$ help optimization? The original Adam paper (Kingma and Ba, 2015) argues that Adam is an approximation to natural gradient descent, since if the objective f is a log-likelihood, $\mathbb{E}[gg^T]$ approximates the Fisher information matrix, which captures curvature information in the space of distributions. There are multiple issues with comparing adaptive methods to natural gradient descent, which we discuss in Appendix D.1. Instead, Balles and Hennig (2018) argue that the primary function of adaptive methods is to equalize the stochastic gradient noise in each direction. But it is still *not* clear why or how equalized noise should help optimization.

Our IAM abstraction makes it easy to explain precisely how rescaling the gradient noise helps. Specifically, we manipulate the update rule for idealized RMSProp:

$$w_{t+1} \leftarrow w_t - \eta A_t g_t \tag{7.1}$$

$$= w_t - \eta A_t \nabla_t - \eta \underbrace{A_t(g_t - \nabla_t)}_{=:\xi_t} \tag{7.2}$$

The $A_t \nabla_t$ term is deterministic; only ξ_t is stochastic, with mean $\mathbb{E}[A_t(g_t - \nabla_t)] = A_t \mathbb{E}[g_t - \nabla_t] = 0$. Take $\varepsilon = 0$ and assume $G_t = \mathbb{E}[g_t g_t^T]$ is invertible, so that $\xi_t = G_t^{-1/2}(g_t - \nabla_t)$. Now we can be more precise about how RMSProp rescales gradient noise. Specifically, we compute the covariance of the noise ξ_t:

$$\mathrm{Cov}(\xi_t) = I - G_t^{-1/2} \nabla_t \nabla_t^T G_t^{-1/2}. \tag{7.3}$$

The key insight is: near stationary points, ∇_t will be small, so that the noise covariance $\mathrm{Cov}(\xi_t)$ is approximately the identity matrix I. In other words, at stationary points, the gradient noise is approximately isotropic. This observation hints at why adaptive methods are so successful for nonconvex problems, where one of the main

challenges is to escape saddle points (Reddi et al., 2018a). Essentially all first-order approaches for escaping saddlepoints rely on adding carefully tuned isotropic noise, so that regardless of what the escape direction is, there is enough noise in that direction to escape with high probability.

7.3.2 Reddi et al. (2018b) counterexample resolution

Recently, Reddi et al. (2018b) provided a simple *convex* stochastic counterexample on which RMSProp and Adam do not converge. Their reasoning is that RMSProp and Adam too quickly forget about large gradients from the past, in favor of small (but poor) gradients at the present. In contrast, for RMSProp with the idealized preconditioner (Algorithm 4 with $A = \mathbb{E}[gg^T]^{-1/2}$), there is no issue, but the preconditioner A cannot be computed in practice. Rather, for this example, the exponential moving average estimation scheme fails to adequately estimate the preconditioner.

The counterexample is an optimization problem of the form

$$\min_{w\in[-1,1]} F(w) = pf_1(w) + (1-p)f_2(w), \tag{7.4}$$

where the stochastic gradient oracle returns ∇f_1 with probability p and ∇f_2 otherwise. Let $\zeta > 0$ be "small," and $C > 0$ be "large." Reddi et al. (2018b) set $p = (1+\zeta)/(C+1)$, $f_1(w) = Cw$, and $f_2(w) = -w$. Overall, then, $F(w) = \zeta w$ which is minimized at $w = -1$, however Reddi et al. (2018b) show that RMSProp has $\mathbb{E}[F(w_t)] \geq 0$ and so incurs suboptimality gap at least ζ. In contrast, the idealized preconditioner is a function of

$$\mathbb{E}[g^2] = p\left(\frac{\partial f_1}{\partial w}\right)^2 + (1-p)\left(\frac{\partial f_2}{\partial w}\right)^2 = C(1+\zeta) - \zeta$$

which is a constant independent of w. Hence the preconditioner is constant, and, up to the choice of stepsize, idealized RMSProp on this problem is the same as SGD, which of course will converge.

The difficulty for practical adaptive methods (which estimate $\mathbb{E}[g^2]$ via an EMA)

is that as C grows, the variance of the estimate of $\mathbb{E}[g^2]$ grows too. Thus Reddi et al. (2018b) break Adam by making estimation of $\mathbb{E}[g^2]$ harder.

7.4 Main results: gluing estimation and optimization

The key enabling insight of this chapter is to separately study the preconditioner and its estimation via EMA, then combine these to give proofs for practical adaptive methods. We will prove a formal guarantee that the EMA estimate $\hat{G}_t$ is close to the true G_t. By combining our estimation results with the underlying behavior of the preconditioner, we will be able to give convergence proofs for practical adaptive methods that are constructed in a novel, modular way.

Separating these two components enables more general results: we actually analyze preconditioned SGD (Algorithm 4) with oracle access to an arbitrary preconditioner $A(w)$. Idealized RMSProp is but one particular instance. Our convergence results depend only on specific properties of the preconditioner $A(w)$, with which we can recover convergence results for many RMSProp variants simply by bounding the appropriate constants. For example, $A = (\mathbb{E}[gg^T]^{1/2}+\varepsilon I)^{-1}$ corresponds to full-matrix Adam with $\beta_1 = 0$ or RMSProp as commonly implemented. For cleaner presentation, we instead focus on the variant $A = (\mathbb{E}[gg^T] + \varepsilon I)^{-1/2}$, but our proof technique can handle either case or its diagonal approximation.

7.4.1 Estimating from moving sequences

The above discussion about IAM is helpful for intuition, and as a base algorithm for analyzing convergence. But it remains to understand how well the estimation procedure works, both for intuition's sake and for later use in a convergence proof. In this section we introduce an abstraction we name "estimation from moving sequences." This abstraction will allow us to guarantee high quality estimates of the preconditioner, or, for that matter, any similarly constructed preconditioner. Our

results will moreover make apparent how to choose the β parameter in the exponential moving average: β should increase with the stepsize η. Increasing β over time has been supported both empirically (Shazeer and Stern, 2018) as well as theoretically (Mukkamala and Hein, 2017; Zou et al., 2019; Reddi et al., 2018b), though to our knowledge, the precise pinning of β to the stepsize η is new.

Suppose there is a sequence of states $w_1, w_2, \ldots, w_T \in \mathcal{W}$, e.g. the parameters of our model at each time step. We have access to the states x_t, but more importantly we know the states are not changing too fast: $\|w_t - w_{t-1}\|$ is bounded for all t. There is a Lipschitz function $G : \mathcal{W} \to \mathbb{R}^{d\times d}$, which in our case is the second moment matrix of the stochastic gradients, but could be more general. We would like to estimate $G(w)$ for each $w = w_t$, but we have only a noisy oracle $Y(w)$ for $G(w)$, which we assume is unbiased and has bounded variance. Our goal is, given noisy reads $Y_1, \ldots, Y_T$ of $G(w_1), \ldots, G(w_T)$, to estimate $G(w_T)$ at the current point w_T as well as possible.

We consider estimators of the form $\sum_{t=1}^{T} p_t Y_t$. For example. setting $p_T = 1$ and all others to zero would yield an unbiased (but high variance) estimate of $G(w_T)$. We could assign more mass to older samples Y_t, but this will introduce bias into the estimate. By optimizing this bias-variance tradeoff, we can get a good estimator. In particular, taking p to be an exponential moving average (EMA) of $\{Y_t\}_{t=1}^{T}$ will prioritize more recent and relevant estimates, while placing enough weight on old estimates to reduce the variance. The tradeoff is controlled by the EMA parameter β; e.g. if the sequence w_t moves slowly (the stepsize is small), we will want large β because older iterates are still very relevant.

In adaptive methods, the underlying function $G(w)$ we want to estimate is $\mathbb{E}[gg^T]$ (or its diagonal $\mathbb{E}[g^2]$), and every stochastic gradient g gives us an unbiased estimate gg^T (resp. g^2) of $G(w)$. With this application in mind, we formalize our results in terms of matrix estimation. By combining standard matrix concentration inequalities (e.g. from Tropp (2011)) with bounds on how fast the sequence moves, we arrive at the following result, proved in Appendix D.7:

Theorem 7.4.1. *Assume $\|w_t - w_{t+1}\| \le \eta M$. The function $G : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ is matrix-valued and L-Lipschitz. The matrix sequence $\{Y_t : t = 1, 2, \ldots\}$ is adapted*

to a filtration $\mathcal{F}_t$ *and satisfies* $\mathbb{E}[Y_t|\mathcal{F}_{t-1}] = G(w_t)$ *for all* $t \geq 1$. *For shorthand we write* $G_t := G(w_t)$. *Additionally, we assume for each* t *that* $\|Y_t - G_t\| \leq R$ *and* $\|\mathbb{E}[(Y_t - G_t)^2|\mathcal{F}_{t-1}]\| \leq \sigma_{\max}^2$. *Set* $p_t \propto \beta^{T-t}$ *with* $\sum_{t=1}^T p_t = 1$ *and assume* $T > 4/(1-\beta)$. *Then with probability* $1-\delta$, *the estimation error* $\Phi = \left\|\sum_{t=1}^T p_t Y_t - G_T\right\|$ *is bounded by*

$$\Phi \leq O(\sigma_{\max}\sqrt{1-\beta}\sqrt{\log(d/\delta)} + ML\eta/(1-\beta)).$$

This is optimized by $\beta = 1 - C\eta^{2/3}$, *for which the bound is* $O((\eta M \sigma_{\max}^2(\log(d/\delta))L)^{1/3})$ *as long as* $T > C'\eta^{-2/3}$.

As long as T is sufficiently large, we can get a high quality estimate of $G_t = \mathbb{E}[g_t g_t^T]$. For this, it suffices to start off the underlying optimization algorithm with $W = O(\eta^{-2/3})$ burn-in iterations where our estimate is updated but the algorithm is not started. This burn-in period will not affect asymptotic runtime as long as $W = O(\eta^{-2/3}) = O(T)$. In our non-convex convergence results we will require $T = O(\tau^{-4})$ and $\eta = O(\tau^2)$, so that $W = O(\tau^{-4/3})$ which is much smaller than T. In practice, one can get away with much shorter (or no) burn-in period.

If β is properly tuned, while running an adaptive method like RMSProp, we will get good estimates of $G = \mathbb{E}[gg^T]$ from samples gg^T. However, we actually require a good estimate of $A = \mathbb{E}[gg^T]^{-1/2}$ and variants. To treat estimation in a unified way, we introduce estimable matrix sequences:

Definition 7.4.1. A $(W, T, \eta, \Delta, \delta)$-*estimable matrix sequence* is a sequence of matrices $\{A(w_t)\}_{t=1}^{W+T}$ generated from $\{w_t\}_t$ with $\|w_t - w_{t-1}\| \leq \eta$ so that with probability $1-\delta$, after a burn-in of time W, we can achieve an estimate sequence $\{\hat{A}_t\}$ so that $\|\hat{A}_t - A_t\| \leq \Delta$ simultaneously for all times $t = W+1, \ldots, W+T$.

Applying Theorem 7.4.1 and union bounding over all time $t = W+1, \ldots, W+T$, we may state a concise result in terms of Definition 7.4.1:

Proposition 7.4.1. *Suppose* $G = \mathbb{E}[g_t g_t^T]$ *is* L*-Lipschitz as a function of* w. *When applied to a generator sequence* $\{w_t\}$ *with* $\|w_t - w_{t-1}\| \leq \eta M$ *and samples* $Y_t = g_t g_t^T$,

the matrix sequence $G_t = \mathbb{E}[g_t g_t^T]$ *is* $(W, T, \eta M, \Delta, \delta)$*-estimable with* $W = O(\eta^{-2/3})$, $T = \Omega(W)$, *and* $\Delta = O(\eta^{1/3}\sigma_{\max}^{2/3}(\log(2Td/\delta)^{1/3}M^{1/3}L^{1/3})$.

We are hence guaranteed a good estimate of G. What we actually want, though, is a good estimate of the preconditioner $A = (G + \varepsilon I)^{-1/2}$. In Appendix D.8 we show how to bound the quality of an estimate of A. One simple result is:

Proposition 7.4.2. *Suppose* $G = \mathbb{E}[gg^T]$ *is L-Lipschitz as a function of* w. *Further suppose a uniform bound* $\lambda_{\min}(G)I \preceq G(w)$ *for all* w, *with* $\lambda_{\min}(G) > 0$. *When applied to a generator sequence* $\{w_t\}$ *with* $\|w_t - w_{t-1}\| \leq \eta M$ *and samples* $Y_t = g_t g_t^T$, *the matrix sequence* $A_t = (G_t + \varepsilon I)^{-1/2}$ *is* $(W, T, \eta M, \Delta, \delta)$*-estimable with* $W = O(\eta^{-2/3})$, $T = \Omega(W)$, *and* $\Delta = O((\eta\sigma_{\max}^2 \log(2Td/\delta)ML)^{1/3}(\varepsilon + \lambda_{\min}(G))^{-3/2})$.

7.4.2 Convergence results

We saw in the last two sections that it is simple to reason about adaptive methods via IAM, and that it is possible to compute a good estimate of the preconditioner. But we still need to glue the two together in order to get a convergence proof for practical adaptive methods.

In this section we will give non-convex convergence results, first for IAM and then for practical realizations thereof. We start with first-order convergence as a warm-up, and then move on to second-order convergence. In each case we give a bound for IAM, study it, and then give the corresponding bound for practical adaptive methods.

Assumptions and notation

We want results for a wide variety of preconditioners A, e.g. $A = I$, the RMSProp preconditioner $A = (G + \varepsilon I)^{-1/2}$, and the diagonal version thereof, $A = (\mathrm{diag}(G) + \varepsilon I)^{-1/2}$. To facilitate this and the future extension of our approach to other preconditioners, we give guarantees that hold for general preconditioners A. Our bounds depend on A via the following properties:

Definition 7.4.2. We say $A(w)$ is a $(\Lambda_1, \Lambda_2, \Gamma, \nu, \lambda_-)$*-preconditioner* if, for all w, the following bounds hold. First, $\|A\nabla f\|^2 \leq \Lambda_1 \|A^{1/2}\nabla f\|^2$. Second, if $\tilde{f}(w)$ is the

quadratic approximation of f at some point w_0, we assume $\|A(\nabla f - \nabla \tilde{f})\| \leq \Lambda_2 \|\nabla f - \nabla \tilde{f}\|$. Third, $\Gamma \geq \mathbb{E}[\|Ag\|^2]$. Fourth, $\nu \leq \lambda_{\min}(A\,\mathbb{E}[gg^T]A^T)$. Finally, $\lambda_- \leq \lambda_{\min}(A)$.

Note that we could bound $\Lambda_1 = \Lambda_2 = \lambda_{\max}(A)$. but in practice Λ_1 and Λ_2 may be smaller, since they depend on the behavior of A only in specific directions. In particular, if the preconditioner A is well-aligned with the Hessian, as may be the case if the natural gradient approximation is valid, then Λ_1 would be very small. If f is exactly quadratic, Λ_2 can be taken as a constant. The constant Γ controls the magnitude of (rescaled) gradient noise, which affects stability at a local minimum. Finally, ν gives a lower bound on the amount of gradient noise in any direction; when ν is larger it is easier to escape saddle points[2]. For shorthand, a $(\cdot, \cdot, \Gamma, \cdot, \lambda_-)$-preconditioner needs to satisfy only the corresponding inequalities.

In Appendix D.3 we provide bounds on these constants for variants of the second moment preconditioner. We highlight the two most relevant cases, for SGD and RMSProp:

Proposition 7.4.3. *The preconditioner $A = I$ is a $(\Lambda_1, \Lambda_2, \Gamma, \nu, \lambda_-)$-preconditioner, with $\Lambda_1 = \Lambda_2 = 1$, $\Gamma \leq \mathbb{E}[\|g\|^2] \leq d \cdot \mathrm{tr}(G)$, $\nu \leq \lambda_{\min}(G)$, and $\lambda_- = 1$.*

Proposition 7.4.4. *The preconditioner $A = (G + \varepsilon I)^{-1/2}$ is a $(\Lambda_1, \Lambda_2, \Gamma, \nu, \lambda_-)$-preconditioner, with*

$$\Lambda_1 = \Lambda_2 = \frac{1}{(\lambda_{\min}(G) + \varepsilon)^{1/2}}, \quad \Gamma = \frac{d\lambda_{\max}(G)}{\varepsilon + \lambda_{\max}(G)},$$
$$\nu = \frac{\lambda_{\min}(G)}{\lambda_{\min}(G) + \varepsilon}, \quad \textit{and} \quad \lambda_- = (\lambda_{\max}(G) + \varepsilon)^{-1/2}.$$

First-order convergence

Proofs are given in Appendix D.6. For all first-order results, we assume that A is a $(\cdot, \cdot, \Gamma, \cdot, \lambda_-)$-preconditioner. The proof technique is essentially standard, with minor changes in order to accomodate general preconditioners. First, suppose we have exact oracle access to the preconditioner:

[2]In cases where $G = \mathbb{E}[gg^T]$ is rank deficient, e.g. in high-dimensional finite sum problems, lower bounds on $\lambda_{\min}(G)$ should be understood as lower bounds on $\mathbb{E}[(v^T g)^2]$ for escape directions v from saddle points, analogous to the "CNC condition" from (Daneshmand et al., 2018).

Theorem 7.4.2. *Run preconditioned SGD with preconditioner A and stepsize $\eta = \tau^2\lambda_-/(L\Gamma)$. For small enough τ, after $T = 2(f(w_0) - f^*)L\Gamma/(\tau^4\lambda_-^2)$ iterations,*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(w_t)\|^2\right] \leq \tau^2. \tag{7.5}$$

Now we consider an alternate version where instead of the preconditioner A_t, we precondition by an noisy version $\hat{A}_t$ that is close to A_t, i.e. $\|\hat{A}_t - A_t\| \leq \Delta$.

Theorem 7.4.3. *Suppose we have access to an inexact preconditioner $\hat{A}$, which satisfies $\|\hat{A} - A\| \leq \Delta$ for $\Delta < \lambda_-/2$. Run preconditioned SGD with preconditioner $\hat{A}$ and stepsize $\eta = \tau^2\lambda_-/(4\sqrt{2}L\Gamma)$. For small enough τ, after $T = 32(f(w_0) - f^*)L\Gamma/(\tau^4\lambda_-^2)$ iterations, we will have*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(w_t)\|^2\right] \leq \tau^2. \tag{7.6}$$

The results are the same up to constants. In other words, as long as we can achieve less than $\lambda_-/2$ error, we will converge at essentially the same rate as if we had the exact preconditioner. In light of this, for the second-order convergence results, we treat only the noisy version.

Theorem 7.4.3 gives a convergence bound assuming a good estimate of the preconditioner, and our estimation results guarantee a good estimate. By gluing together Theorem 7.4.3 with our estimation results for the RMSProp preconditioner, i.e. Proposition 7.4.2, we can give a convergence result for bona fide RMSProp:

Corollary 7.4.1. *Consider RMSProp with burn-in, as in Algorithm 6, where we estimate $A = (G + \varepsilon I)^{-1/2}$. Retain the same choice of $\eta = O(\tau^2)$ and $T = O(\tau^{-4})$ as in Theorem 7.4.3. For small enough τ, such a choice of η will yield $\Delta < \lambda_-/2$. Choose all other parameters e.g. β in accordance with Proposition 7.4.2. In particular, choose $W = \Theta(\eta^{-2/3}) = \Theta(\tau^{-4/3}) = O(T)$ for the burn-in parameter. Then with probability*

Algorithm 6 RMSProp with burn-in

Input: initial w_0, time T, stepsize η, burn-in length W
$\hat{G}_0 \leftarrow \textsc{BurnIn}(W, \beta)$ ▷ Appendix D.2
for $t = 0, \dots, T$ **do**
  $g_t \leftarrow$ stochastic gradient
  $\hat{G}_t \leftarrow \beta \hat{G}_{t-1} + (1-\beta) g_t g_t^T$
  $\hat{A}_t \leftarrow \hat{G}_t^{-1/2}$
  $w_{t+1} \leftarrow w_t - \eta \hat{A}_t g_t$
end for

$1-\delta$, *in overall time* $O(W+T) = O(\tau^{-4})$, *we achieve*

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(w_t)\|^2\right] \leq \tau^2. \tag{7.7}$$

Second-order convergence

Now we leverage the power of our high level approach to prove nonconvex second-order convergence for adaptive methods. Like the first-order results, we start by proving convergence bounds for a generic, possibly inexact preconditioner A. Our proof is based on that of Daneshmand et al. (2018) for SGD, and therefore we achieve the same $O(\tau^{-5})$ rate. It may be possible to improve our result using the technique of Fang et al. (2019), which is concurrent work to ours. However, our focus is on the preconditioner, and our study of it is wholly new. Accordingly, we study the convergence of Algorithm 7, which is the same as Algorithm 4 (generic preconditioned SGD) except that once in a while we take a large stepsize so we may escape saddle-points. The proof is given completely in Appendix D.5. At a high level, we show the algorithm makes progress when the gradient is large and when we are at a saddle point, and does not escape from local minima. Our analysis uses all the constants specified in Definition 7.4.2, e.g. the speed of escape from saddle points depends on ν, the lower bound on stochastic gradient noise.

Then, as before, we simply fuse our convergence guarantees with our estimation guarantees. The end result is, to our knowledge, the first nonconvex second-order convergence result for any adaptive method.

Algorithm 7 Preconditioned SGD with increasing stepsize

Input: initial w_0, time T, stepsizes η, r, threshold t_{thresh}, matrix error Δ
for $t = 0, \dots, T$ **do**
  $A_t \leftarrow A(w_t)$ ▷ preconditioner at w_t
  $\hat{A}_t \leftarrow$ any matrix with $\|\hat{A}_t - A_t\| \le \Delta$
  $g_t \leftarrow$ stochastic gradient at w_t
  if $t \bmod t_{\text{thresh}} = 0$ **then**
    $w_{t+1} \leftarrow w_t - r\hat{A}_t g_t$
  else
    $w_{t+1} \leftarrow w_t - \eta\hat{A}_t g_t$
  end if
end for

Definitions for second-order results. Assume further that the Hessian H is ρ-Lipschitz and the preconditioner $A(w)$ is α-Lipschitz. The dependence on these constants is made precise in the proof, in Appendix D.5. The usual stepsize is η, while r is the occasional large stepsize that happens every t_{thresh} iterations. We tolerate a small probability of failure δ. For all results, we assume A is a $(\Lambda_1, \Lambda_2, \Gamma, \nu, \lambda_-)$-preconditioner. For simplicity, we assume the noisy estimate $\hat{A}$ also satisfies the Λ_1 inequality. We will also assume a uniform bound on $\|Ag\| \le M = O(\sqrt{\Gamma})$.

The proofs rely on a few other quantities that we optimally determine as a function of the problem parameters: f_{thresh} is a threshold on the function value progress, and $g_{\text{thresh}} = f_{\text{thresh}}/t_{\text{thresh}}$ is the time-amortized average of f_{thresh}. We specify the precise values of all quantities in the proof.

Theorem 7.4.4. *Consider Algorithm 7 with inexact preconditioner $\hat{A}_t$ and exact preconditioner A_t satisfying the preceding requirements. Suppose that for all t, we have $\|\hat{A}_t - A_t\| = O(\tau^{1/2})$. Then for small τ, with probability $1 - \delta$, we reach an $(\tau, \sqrt{\rho\tau})$-stationary point in time*

$$T = \tilde{O}\left(\frac{\Lambda_1^4 \Lambda_2^4 \Gamma^4}{\lambda_-^{10} \nu^4} \cdot \frac{L^3}{\rho\delta^3} \cdot \tau^{-5}\right). \tag{7.8}$$

The big-O suppresses other constants given in the proof.

To prove a result for bona fide RMSProp, we need to combine Theorem 7.4.4 with an algorithm that maintains a good estimate of $G = \mathbb{E}[gg^T]$ (and consequently

$A = (G + \varepsilon I)^{-1/2}$). This is more delicate than the first-order case because now the stepsize varies. Whenever we take a large stepsize, the estimation algorithm will need to hallucinate S number of smaller steps in order to keep the estimate accurate. Our overall scheme is formalized in Appendix D.2, for which the following convergence result holds:

Corollary 7.4.2. *Consider the RMSProp version of Algorithm 7 that is described in Appendix D.2. Retain the same choice of $\eta = O(\tau^{5/2})$, $r = O(\tau)$, and $T = O(\tau^{-5})$ as in Theorem 7.4.4. For small enough τ, such a choice of η will yield $\Delta < \lambda_-/2$. Choose $W = \Theta(\eta^{-2/3}) = \Theta(\tau^{-5/3}) = O(T)$ for the burn-in parameter Choose $S = O(\tau^{-3/2})$, so that as far as the estimation scheme is concerned, the stepsize is bounded by $\max\{\eta, r/S\} = O(\tau^{5/2}) = O(\eta)$. Then as before, with probability $1 - \delta$, we can reach an $(\tau, \sqrt{\rho\tau})$-stationary point in total time*

$$W + T = \tilde{O}\left(\frac{\Lambda_1^4 \Lambda_2^4 \Gamma^4}{\lambda_-^{10} \nu^4} \cdot \frac{L^3}{\rho \delta^3} \cdot \tau^{-5}\right), \tag{7.9}$$

where $\Lambda_1, \Lambda_2, \Gamma, \nu, \lambda_-$ are the constants describing $A = (G + \varepsilon I)^{-1/2}$.

Again, as in the first order results, one could substitute in any other estimable preconditioner. In particular, in Appendix D.4 we discuss the more common diagonal version of RMSProp.

7.5 Discussion

Separating the estimation step from the preconditioning enables evaluation of different choices for the preconditioner.

7.5.1 How to set the regularization parameter ε

In the adaptive methods literature, it is still a mystery how to properly set the regularization parameter ε that ensures invertibility of $G + \varepsilon I$. When the optimality tolerance τ is small enough, estimating the preconditioner is not the bottleneck. Thus,

focusing only on the idealized case, one could just choose ε to minimize the bound. Our first-order results depend on ε only through the following term:

$$\frac{\Gamma}{\lambda_{\min}(A)} \leq \frac{d\lambda_{\min}(G)}{\varepsilon + \lambda_{\min}(G)} \cdot (\lambda_{\max}(G) + \varepsilon), \tag{7.10}$$

where we have used the preconditioner bounds from Proposition 7.4.4. This is minimized by taking $\varepsilon \to \infty$, which suggests using identity preconditioner, or SGD. In contrast, for second-order convergence, the bound is

$$\frac{\Lambda_1^4 \Lambda_2^4 \Gamma^4}{\lambda_-^{10} \nu^4} \leq d^4 \kappa(G)^4 (\lambda_{\max}(G) + \varepsilon), \tag{7.11}$$

which is instead minimized with $\varepsilon = 0$. So for the best second-order convergence rate, it is desireable to set ε as small as possible. Note that since our bounds hold only for small enough convergence tolerance τ, it is possible that the optimal ε should depend in some way on τ.

7.5.2 Comparison to SGD

Another important question we make progress towards is: when are adaptive methods better than SGD? Our second-order result depends on the preconditioner only through $\Lambda_1^4 \Lambda_2^4 \Gamma^4 / (\lambda_-^{10} \nu^4)$. Plugging in Proposition 7.4.3 for SGD, we may bound

$$\frac{\Lambda_1^4 \Lambda_2^4 \Gamma^4}{\lambda_-^{10} \nu^4} \leq \frac{\mathbb{E}[\|g\|^2]^4}{\lambda_{\min}(G)^4} \leq d^4 \kappa(G)^4, \tag{7.12}$$

while for full-matrix RMSProp, we have

$$\frac{\Lambda_1^4 \Lambda_2^4 \Gamma^4}{\lambda_-^{10} \nu^4} \leq d^4 \kappa(G)^4 (\lambda_{\max}(G) + \varepsilon). \tag{7.13}$$

Setting $\varepsilon = 0$ for simplicity, we conclude that full-matrix RMSProp converges faster if $\lambda_{\max}(G) \leq 1$.

Now suppose that for a given optimization problem, the preconditioner A is well-aligned with the Hessian so that $\Lambda_1 = O(1)$ (e.g. if the natural gradient approx-

imation holds) and that near saddle points the objective is essentially quadratic so that $\Lambda_2 = O(1)$. In this regime, the preconditioner dependence of idealized full matrix RMSProp is $d^4\lambda_{\max}(G)^5$, which yields a better result than SGD when $\lambda_{\max}(G) \leq \lambda_{\min}(G)^{-4}$. This will happen whenever $\lambda_{\min}(G)$ is relatively small. Thus, when there is not much noise in the escape direction, and the Hessian and $G^{-1/2}$ are not poorly aligned, RMSProp will converge faster overall. Similar phenomenon can be shown for the diagonal case when the approximation is good, per the results in Appendix D.3 and D.4.

7.5.3 Alternative preconditioners

Our analysis inspires the design of other preconditioners: e.g., if at each iteration we sample two independent stochastic gradients g_1 and g_2, we have unbiased sample access to $(g_1 - g_2)(g_1 - g_2)^T$, which in expectation yields the covariance $\Sigma = \mathrm{Cov}(g)$ instead of the second moment matrix of g. It immediately follows that we can prove second-order convergence results for an algorithm that constructs an exponential moving average estimate of Σ and preconditions by $\Sigma^{-1/2}$, as advocated by Ida et al. (2017).

7.5.4 Tuning the EMA parameter β

Another mystery of adaptive methods is how to set the exponential moving average (EMA) parameter β. In practice β is typically set to a constant, e.g. 0.99, while other parameters such as the stepsize η are tuned more carefully and may vary over time. While our estimation guarantee Theorem 7.4.1, suggests setting $\beta = 1 - O(\eta^{2/3})$, the specific formula depends on constants that may be unknown, e.g. Lipschitz constants and gradient norms. Instead, one could set $\beta = 1 - C\eta^{2/3}$, and search for a good choice of the hyperparameter C. For example, the common initial choice of $\eta = 0.001$ and $\beta = 0.99$ corresponds to $C = 1$.

7.6 Experiments

We experimentally test our claims about adaptive methods escaping saddle points, and our suggestion for setting β.

Escaping saddle points. First, we test our claim that when the gradient noise is ill-conditioned, adaptive methods escape saddle points faster than SGD, and often converge faster to (approximate) local minima. We construct a two dimensional[3] non-convex problem $f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$ where $f_i(w) = \frac{1}{2}w^T H w + b_i^T w + \|w\|_{10}^{10}$. Here, $H = \mathrm{diag}([1, -0.1])$, so f has a saddle point at the origin with objective value zero. The vectors b_i are chosen so that sampling b uniformly from $\{b_i\}_{i=1}^n$ yields $\mathbb{E}[b] = 0$ and $\mathrm{Cov}(b) = \mathrm{diag}([1, 0.01])$. Hence at the origin there is an escape direction but little gradient noise in that direction.

We initialize SGD and (diagonal) RMSProp (with $\beta = 1 - \eta^{2/3}$) at the saddle point and test several stepsizes η for each. Results for the first 10^4 iterations are shown in Figure 7-1. In order to escape the saddle point as fast as RMSProp, SGD requires a substantially larger stepsize, e.g. SGD needs $\eta = 0.01$ to escape as fast as RMSProp does with $\eta = 0.001$. But with such a large stepsize, SGD cannot converge to a small neighborhood of the local minimum, and instead bounces around due to gradient noise. Since RMSProp can escape with a small stepsize, it can converge to a much smaller neighborhood of the local minimum. Overall, for any fixed final convergence criterion, RMSProp escapes faster and converges faster overall.

Setting the EMA parameter β. Next, we test our recommendations regarding setting the EMA parameter β. We consider logistic regression on MNIST. We use (diagonal) RMSProp with batch size 100, decreasing stepsize $\eta_t = 0.001/\sqrt{t}$ and $\varepsilon = 10^{-8}$, and compare different schedules for β. Specifically we test $\beta \in \{0.7, 0.9, 0.97, 0.99\}$ (so that $1 - \beta$ is spaced roughly logarithmically) as well as our recommendation of $\beta_t = 1 - C\eta_t^{2/3}$ for $C \in \{0.1, 0.3, 1\}$. As shown in Figure 7-2, all

[3]The same phenomenon still holds in higher dimensions but the presentation is simpler with $d = 2$.

options for β have similar performance initially, but as η_t decreases, large β yields substantially better performance. In particular, our decreasing β schedule achieved the best performance, and moreover was insensitive to how C was set.

7.7 Further discussion and future work

In this chapter, we gave the first second-order guarantees for AGMs. To achieve these guarantees, we introduced a new, simpler way of reasoning about AGMs, based on separating the estimation of the preconditioner A from the effect of the preconditioner on optimization. Our convergence guarantees also led to insights about how to set the various parameters of RMSProp, in particular ε and β, as well as better understanding of when AGMs can beat SGD.

There are many fruitful directions for further research. Chief among these is extending our proof technique to provide guarantees for Adam. Adam is slightly more complicated to study, but we are optimistic due to the close relationship between Adam and RMSProp. Other related algorithms may also be amenable to similar study.

It is also likely our results can be improved. As mentioned earlier, it is likely to improve the dependence of our results on τ by adapting the technique developed by Fang et al. (2019). Our results could also perhaps be strengthened for particular problem instances. Specifically, our proof relies on a pessimistic view of how the preconditioner A interacts with the Hessian $\nabla^2 f$. For some problems, these two matrices may be "aligned" in a way that is favorable to optimization and leads to tighter bounds.

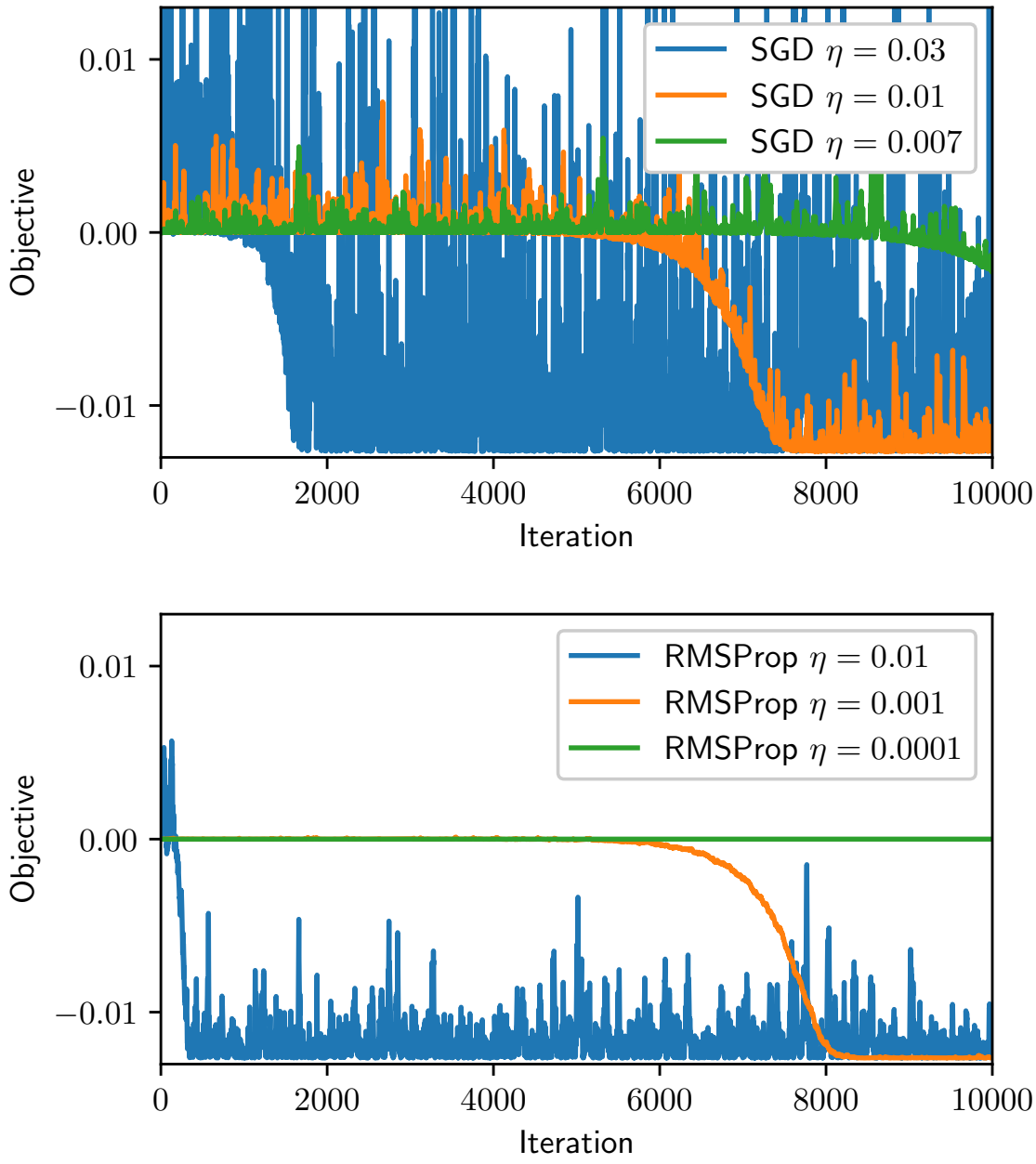


Figure 7-1: SGD (top) vs RMSProp (bottom) performance escaping a saddle point with poorly conditioned gradient noise. Compared to RMSProp, SGD requires a larger stepsize to escape as quickly, which negatively impacts convergence to the local minimum.

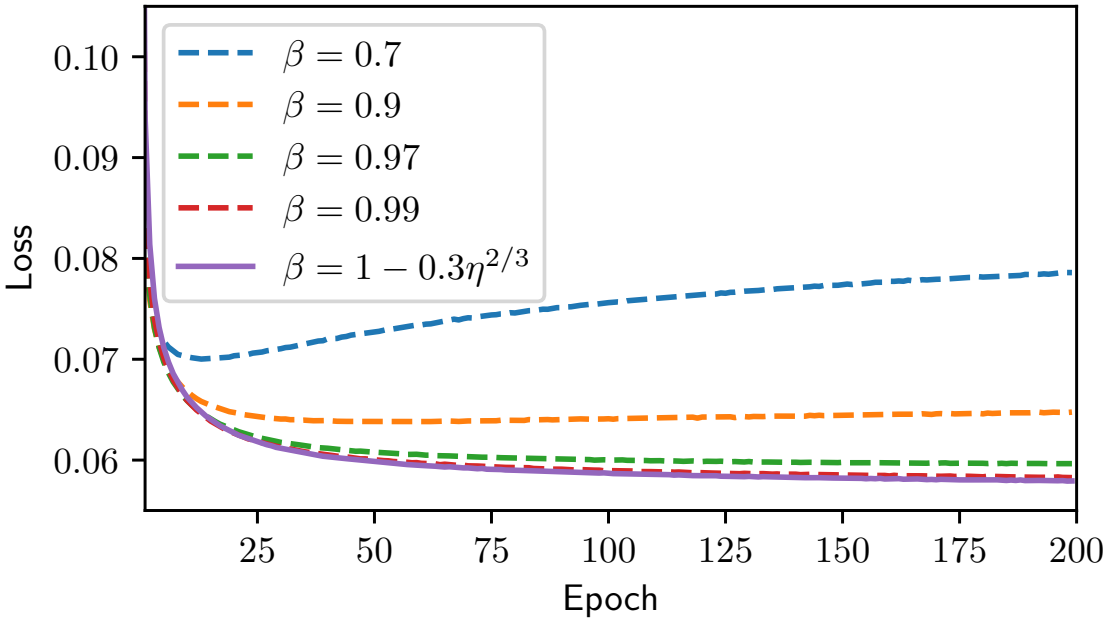


Figure 7-2: Performance on MNIST logistic regression of RMSProp with different choices of β and decreasing stepsize.

Part IV

Conclusion

Chapter 8

Conclusion

8.1 High-level summary

As depicted in Figure 1-1, in this thesis we studied the interplay of data perturbations, learning and generalization, and non-convex optimization. In Part I we studied how data perturbations, particularly DRO, shed light on generalization. We developed a new type of DRO problem and developed deep connections to kernel methods. In Part II, we developed algorithms for non-convex perturbation-robust optimization problems. We focused on submodular objectives, and were able to give algorithms with solution quality guarantees for some perturbation-robust problems. And, in Part III, we showed that perturbations in the form of subsampling error, typically thought of as a nuisance, can be reshaped in order to yield better non-convex optimization performance.

Though the mechanical tools employed in each part vary widely, the goals and ideas are strongly linked. We wish to understand how algorithms and models are affected by perturbations, then leverage this understanding to improve their performance. We seek new algorithms that, given limited data, learn perturbation-robust models and make robust decisions that succeed in many environments.

8.2 Future directions

We have already discussed, at the end of each chapter, directions for future work that are most relevant to that chapter. Here we instead speculate on broader themes and more ambitious future directions.

8.2.1 Perturbations and generalization

While the connection between perturbations and (statistical and otherwise) generalization is already rich, there are many ways to further strengthen this connection.

For DRO specifically, there are many exciting directions. For existing DRO problems, it is likely possible to develop tighter generalization bounds with fewer assumptions. There is also much room for further employment of DRO as a tool to enable algorithmic fairness, causal inference, transfer learning, etc.

Other kinds of perturbations also present exciting opportunities. For example, new types of data augmentation will continue to improve real-world performance of learned models. Other notions of robustness will also expand the set of inferences we can make from data. For example, in the line of work building off of Chernozhukov et al. (2018), a kind of robustness or insensitivity called Neyman orthogonality enables estimation of average treatment effect (a causal quantity) in new settings using only observational data.

8.2.2 Perturbation-aware optimization

Perturbation-aware optimization such as DRO is difficult for two reasons: the objective may be non-convex, and incorporating the perturbations is often challenging. Most of the work done so far has focused on convex objectives, together with one of a few well-behaved types of perturbations or uncertainty sets. Our work in Part II broke out of this mold by considering non-convex objectives.

Even for standard uncertainty sets, there is much work to be done in building out tools for perturbation-aware non-convex optimization. There are many submodular DRO problems we have not yet explored that may prove tractable. Beyond submod-

ularity, other specific subclasses of non-convex objective functions may also admit DRO problems with performance guarantees.

An orthogonal direction of future work is to consider broader kinds of perturbations, and develop algorithms to make optimzation tractable in those settings. For example, while Meinshausen (2018) captures certain causal inference problems in the language of DRO, the resulting DRO formulations are typically too challenging to be useful at the present. We are optimistic that many DRO problems, currently of theoretical interest only, will eventually admit tractable optimization.

8.2.3 Perturbations for optimization

We studied how to adjust a base algorithm (SGD) to make it better use subsampling error. Beyond directly extending our results, there are other potential avenues for better using noise in optimization. For example, there is substantial work that changes the subsampling distribution via e.g. changing the sampling frequencies of each data point, or even sampling non-iid. Finally, it remains to be understood how these perturbations due to subsampling interact with intentional perturbations added by techniques such as DRO.

Part V

Bibliography and Appendix

Bibliography

Last.fm dataset - 360k users. URL http://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/lastfm-360K.html.

Yahoo! Webscope dataset ydata-ysm-advertiser-bids-v1_0. URL http://research.yahoo.com/Academic_Relations.

Marek Adamczyk, Maxim Sviridenko, and Justin Ward. Submodular Stochastic Probing on Matroids. *Mathematics of Operations Research*, 41(3):1022–1038, April 2016.

Lada A. Adamic and Natalie Glance. The political blogosphere and the 2004 u.s. election: Divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery*, LinkKDD '05, pages 36–43, New York, NY, USA, 2005. ACM. ISBN 1-59593-215-1.

Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding approximate local minima faster than gradient descent. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 1195–1199, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4528-6.

Naman Agarwal, Brian Bullins, Xinyi Chen, Elad Hazan, Karan Singh, Cyril Zhang, and Yi Zhang. Efficient full-matrix adaptive regularization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 102–110, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Shipra Agrawal, Yichuan Ding, Amin Saberi, and Yinyu Ye. Correlation robust stochastic optimization. In *SODA*, 2010.

Zeyuan Allen-Zhu and Yuanzhi Li. Neon2: Finding local minima via first-order oracles. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 3720–3730. Curran Associates, Inc., 2018.

Noga Alon, Iftah Gamzu, and Moshe Tennenholtz. Optimizing Budget Allocation Among Channels and Influencers. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 381–388, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1229-5.

Nima Anari, Nika Haghtalab, Seffi Naor, Sebastian Pokutta, Mohit Singh, and Alfredo Torrico. Structured robust submodular maximization: Offline and online algorithms. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 3128–3137. PMLR, 16–18 Apr 2019.

Alper Atamtürk and Vishnu Narayanan. Polymatroids and mean-risk minimization in discrete optimization. *Operations Research Letters*, 36(5):618–622, September 2008.

Brian Axelrod, Yang P. Liu, and Aaron Sidford. *Near-optimal Approximate Discrete and Continuous Submodular Function Minimization*, pages 837–853. 2020.

Francis Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trendső in Machine Learning*, 6(2-3):145–373, 2013.

Francis Bach. Submodular Functions: From Discrete to Continous Domains. *Mathematical Programming*, 175(1-2):419–459, 2019.

Eric Balkanski, Aviad Rubinstein, and Yaron Singer. The power of optimization from samples. In *Advances In Neural Information Processing Systems*, pages 4017–4025, 2016.

Eric Balkanski, Aviad Rubinstein, and Yaron Singer. The limitations of optimization from samples. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1016–1027. ACM, 2017.

Lukas Balles and Philipp Hennig. Dissecting Adam: The sign, magnitude and variance of stochastic gradients. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 404–413, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Stephen R Becker, Emmanuel J Candès, and Michael C Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical programming computation*, 3(3):165–218, 2011.

Aharon Ben-Tal and Arkadi Nemirovski. Robust solutions of Linear Programming problems contaminated with uncertain data. *Mathematical Programming*, 88(3): 411–424, September 2000.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

Aharon Ben-Tal, Dick den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.

D. Bertsimas, D. Brown, and C. Caramanis. Theory and Applications of Robust Optimization. *SIAM Review*, 53(3):464–501, January 2011.

Dimitris Bertsimas and Melvyn Sim. Robust Discrete Optimization and Network Flows. *Mathematical programming*, 98(1):49–71, 2003.

Dimitris Bertsimas, Vishal Gupta, and Nathan Kallus. Data-driven robust optimization. *Mathematical Programming*, 167(2):235–292, Feb 2018.

Michael J. Best and Nilotpal Chakravarti. Active set algorithms for isotonic regression; A unifying framework. *Mathematical Programming*, 47(1-3):425–439, 1990.

Andrew An Bian, Baharan Mirzasoleiman, Joachim M. Buhmann, and Andreas Krause. Guaranteed non-convex optimization: Submodular maximization over continuous domains. In *AISTATS*, 2017.

Yatao An Bian. Provable non-convex optimization and algorithm validation via submodularity, 2019.

Alberto Bietti and Julien Mairal. Group invariance, stability to deformations, and complexity of deep convolutional representations. *The Journal of Machine Learning Research*, 20(1):876–924, 2019.

Alberto Bietti, Grégoire Mialon, Dexiong Chen, and Julien Mairal. A kernel perspective for regularizing deep neural networks. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019.

Garrett Birkhoff. Rings of sets. *Duke Mathematical Journal*, 3(3):443–454, 1937.

Mikoaj Bikowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018.

Jose Blanchet, Yang Kang, Fan Zhang, and Karthyek Murthy. Data-driven optimal transport cost selection for distributionally robust optimization. *arXiv preprint arXiv:1705.07152*, 2017.

Jose Blanchet, Karthyek Murthy, and Fan Zhang. Optimal transport based distributionally robust optimization: Structural properties and iterative schemes. *arXiv preprint arXiv:1810.02403*, 2018.

Jose Blanchet, Yang Kang, and Karthyek Murthy. Robust wasserstein profile inference and applications to machine learning. *Journal of Applied Probability*, 56(3):830–857, 2019.

Ilija Bogunovic, Slobodan Mitrović, Jonathan Scarlett, and Volkan Cevher. Robust submodular maximization: A non-uniform partitioning approach. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages

508–516, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. Maximizing Social Influence in Nearly Optimal Time. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 946–957, Philadelphia, PA, USA, 2014. Society for Industrial and Applied Mathematics. ISBN 978-1-61197-338-9.

Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

Stephen Boyd, Seung-Jean Kim, Lieven Vandenberghe, and Arash Hassibi. A tutorial on geometric programming. *Optimization and engineering*, 8(1):67–127, 2007.

Niv Buchbinder, Moran Feldman, Joseph SeffiNaor, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SIAM Journal on Computing*, 44(5):1384–1402, 2015.

Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

Y. Carmon, J. Duchi, O. Hinder, and A. Sidford. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, 2018.

Deeparnab Chakrabarty, Prateek Jain, and Pravesh Kothari. Provable submodular minimization using wolfe's algorithm. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 802–809. Curran Associates, Inc., 2014.

Deeparnab Chakrabarty, Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. Subquadratic Submodular Function Minimization. In *STOC*, 2017.

V. Chandrasekaran and P. Shah. Relative Entropy Relaxations for Signomial Optimization. *SIAM Journal on Optimization*, 26(2):1147–1173, January 2016.

C. Chekuri, J. Vondrak, and R. Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 575–584, Oct 2010.

Robert S Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. Robust optimization for non-convex objectives. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4708–4717. Curran Associates, Inc., 2017.

Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM, 2009.

Wei Chen, Chi Wang, and Yajun Wang. Scalable Influence Maximization for Prevalent Viral Marketing in Large-scale Social Networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 1029–1038, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0055-1.

Wei Chen, Tian Lin, Zihan Tan, Mingfei Zhao, and Xuren Zhou. Robust influence maximization. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 795–804. ACM, 2016.

Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. In *International Conference on Learning Representations*, 2019.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 2018.

Mung Chiang. Geometric Programming for Communication Systems. *Commun. Inf. Theory*, 2(1/2):1–154, July 2005.

Kacper Chwialkowski, Heiko Strathmann, and Arthur Gretton. A kernel test of goodness of fit. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2606–2615, New York, New York, USA, 20–22 Jun 2016. PMLR.

Hadi Daneshmand, Jonas Kohler, Aurelien Lucchi, and Thomas Hofmann. Escaping saddles with stochastic gradients. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1155–1164, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Abhimanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 1057–1064, New York, NY, USA, June 2011. ACM. ISBN 978-1-4503-0619-5.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger,

editors, *Advances in Neural Information Processing Systems 27*, pages 1646–1654. Curran Associates, Inc., 2014.

Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3): 595–612, 2010.

Amol Deshpande, Lisa Hellerstein, and Devorah Kletenik. Approximation Algorithms for Stochastic Submodular Set Cover with Applications to Boolean Function Evaluation and Min-Knapsack. *ACM Trans. Algorithms*, 12(3):42:1–42:28, April 2016.

Josip Djolonga and Andreas Krause. From MAP to marginals: Variational inference in Bayesian submodular models. In *Advances in Neural Information Processing Systems*, pages 244–252, 2014.

Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66. ACM, 2001.

Nan Du, Le Song, Manuel Gomez Rodriguez, and Hongyuan Zha. Scalable influence estimation in continuous-time diffusion networks. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3147–3155. Curran Associates, Inc., 2013.

Nan Du, Yingyu Liang, Maria-Florina Balcan, and Le Song. Influence function learning in information diffusion networks. In *ICML*, pages 2016–2024, 2014.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12: 2121–2159, July 2011.

John Duchi, Peter Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *arXiv preprint arXiv:1610.03425*, 2016.

John C Duchi, Peter L Bartlett, and Martin J Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012.

Joseph C Dunn and S Harshbarger. Conditional gradient algorithms with open loop step size rules. *Journal of Mathematical Analysis and Applications*, 62(2):432–444, 1978.

Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, pages 258–267, Arlington, Virginia, United States, 2015. AUAI Press. ISBN 978-0-9966431-0-8.

J. Ecker. Geometric Programming: Methods, Computations and Applications. *SIAM Review*, 22(3):338–362, July 1980.

Jack Edmonds. Submodular functions, matroids, and certain polyhedra. In Richard K Guy, editor, *Calgary International Conference on Combinatorial Structures and Their Applications*. Gordon and Breach, 1970.

Alina Ene and Huy L. Nguyen. A Reduction for Optimizing Lattice Submodular Functions with Diminishing Returns. *arXiv:1606.08362 [cs]*, June 2016.

Cong Fang, Zhouchen Lin, and Tong Zhang. Sharp analysis for nonconvex sgd escaping from saddle points. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 1192–1234, Phoenix, USA, 25–28 Jun 2019. PMLR.

Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011.

Moran Feldman, Joseph (Seffi) Naor, and Roy Schwartz. A unified continuous greedy algorithm for submodular maximization. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2011.

Nicolas Fournier and Arnaud Guillin. On the rate of convergence in Wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3): 707–738, Aug 2015.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, March 1956.

Satoru Fujishige. *Submodular Functions and Optimization*, volume 58. Elsevier, 2005.

Rui Gao and Anton J Kleywegt. Distributionally robust stochastic optimization with Wasserstein distance. *arXiv preprint arXiv:1604.02199*, 2016.

Rui Gao, Xi Chen, and Anton J Kleywegt. Wasserstein distributional robustness and regularization in statistical learning. *arXiv preprint arXiv:1712.06050*, 2017.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points — online stochastic gradient for tensor decomposition. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 797–842, Paris, France, 03–06 Jul 2015. PMLR.

Gauthier Gidel, Tony Jebara, and Simon Lacoste-Julien. Frank-Wolfe Algorithms for Saddle Point Problems. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 362–371, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

Gagan Goel, Chinmay Karande, Pushkar Tripathi, and Lei Wang. Approximability of combinatorial problems with multi-agent submodular cost functions. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 755–764. IEEE, 2009.

Michel Goemans and Jan Vondrák. Stochastic Covering and Adaptivity. In *LATIN 2006: Theoretical Informatics*, pages 532–543. Springer Berlin Heidelberg, March 2006.

Joel Goh and Melvyn Sim. Distributionally robust optimization and its tractable approximations. *Operations Research*, 58(4-part-1):902–917, 2010.

Daniel Golovin and Andreas Krause. Adaptive Submodularity: Theory and Applications in Active Learning and Stochastic Optimization. *Journal of Artificial Intelligence*, 42:427–486, 2011.

M Gomez Rodriguez, B Schölkopf, Langford J Pineau, et al. Influence maximization in continuous time diffusion networks. In *29th International Conference on Machine Learning (ICML 2012)*, pages 1–8. International Machine Learning Society, 2012.

Manuel Gomez Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 1019–1028, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0055-1.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Jun-ya Gotoh, Michael Jong Kim, and Andrew E.B. Lim. Robust empirical optimization is almost the same as mean-variance optimization. *Operations Research Letters*, 46(4):448 – 452, 2018.

Corinna Gottschalk and Britta Peis. Submodular function maximization on the bounded integer lattice. In *Approximation and Online Algorithms: 13th International Workshop (WAOA)*, 2015.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, March 2012.

Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

Marwa El Halabi and Stefanie Jegelka. Minimizing approximately submodular functions. *arXiv preprint arXiv:1905.12145*, 2019.

Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1929–1938, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Hamed Hassani, Mahdi Soltanolkotabi, and Amin Karbasi. Gradient Methods for Submodular Maximization. In *Advances in Neural Information Processing Systems 30*, pages 5843–5853, 2017.

Avinatan Hassidim and Yaron Singer. Submodular optimization under noise. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1069–1122, Amsterdam, Netherlands, 07–10 Jul 2017. PMLR.

Daisuke Hatano, Takuro Fukunaga, Takanori Maehara, and Ken-ichi Kawarabayashi. Lagrangian Decomposition Algorithm for Allocating Marketing Channels. In *AAAI*, pages 1144–1150, 2015.

Xinran He and David Kempe. Robust influence maximization. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 885–894. ACM, 2016.

Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does distributionally robust supervised learning give robust classifiers? In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2029–2037, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Yasutoshi Ida, Yasuhiro Fujiwara, and Sotetsu Iwamura. Adaptive learning rate via covariance matrix based preconditioning for deep neural networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1923–1929, 2017.

Shinji Ito. Submodular function minimization with noisy evaluation oracle. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 12103–12113. Curran Associates, Inc., 2019.

Satoru Iwata and Kiyohito Nagano. Submodular function minimization under covering constraints. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 671–680. IEEE, 2009.

Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *J. ACM*, 48(4):761–777, July 2001.

Robert A Jacobs. Increased rates of convergence through learning rate adaptation. *Neural networks*, 1(4):295–307, 1988.

Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 427–435, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

Stefanie Jegelka, Francis Bach, and Suvrit Sra. Reflection methods for user-friendly submodular optimization. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1313–1321. Curran Associates, Inc., 2013.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1724–1732, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Wittawat Jitkrittum, Wenkai Xu, Zoltan Szabo, Kenji Fukumizu, and Arthur Gretton. A linear-time kernel goodness-of-fit test. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 262–271. Curran Associates, Inc., 2017.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 315–323. Curran Associates, Inc., 2013.

Dimitris Kalimeris, Gal Kaplun, and Yaron Singer. Robust influence maximization for hyperparametric models. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3192–3200, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Amin Karbasi, Hamed Hassani, Aryan Mokhtari, and Zebang Shen. Stochastic continuous greedy++: When upper and lower bounds match. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 13066–13076. Curran Associates, Inc., 2019.

Mohammad Karimi, Mario Lucic, Hamed Hassani, and Andreas Krause. Stochastic Submodular Maximization: The Case of Coverage Functions. In *Advances in Neural Information Processing Systems 30*, pages 6856–6866, 2017.

Ehsan Kazemi, Morteza Zadimoghaddam, and Amin Karbasi. Scalable deletion-robust submodular maximization: Data summarization with privacy and fairness

constraints. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2544–2553, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the Spread of Influence Through a Social Network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 137–146, New York, NY, USA, 2003. ACM. ISBN 978-1-58113-737-8.

David Kempe, Jon M Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11(4):105–147, 2015.

Vladimir R. Khachaturov, Roman V. Khachaturov, and Ruben V. Khachaturov. Supermodular Programming on Finite Lattices. *Computational Mathematics and Mathematical Physics*, 52(6):855–878, 2012.

Sunyoung Kim and Masakazu Kojima. Exact Solutions of Some Nonconvex Quadratic Optimization Problems via SDP and SOCP Relaxations. *Computational Optimization and Applications*, 26(2):143–154, Nov 2003.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Johannes Kirschner, Ilija Bogunovic, Stefanie Jegelka, and Andreas Krause. Distributionally robust bayesian optimization. *arXiv preprint arXiv:2002.09038*, 2020.

Vladimir Kolmogorov and Akiyoshi Shioura. New algorithms for convex cost tension problem with application to computer vision. *Discrete Optimization*, 6:378–393, 2009.

Andreas Krause, H Brendan McMahan, Carlos Guestrin, and Anupam Gupta. Robust submodular observation selection. *Journal of Machine Learning Research*, 9(Dec): 2761–2801, 2008a.

Andreas Krause, H Brendan McMahan, Carlos Guestrin, and Anupam Gupta. Robust submodular observation selection. *Journal of Machine Learning Research*, 9(Dec): 2761–2801, 2008b.

Andreas Krause, Alex Roper, and Daniel Golovin. Randomized sensing in adversarial environments. In *IJCAI*, 2011.

Alex Kulesza and Ben Taskar. *Determinantal Point Processes for Machine Learning*. Now Publishers Inc., Hanover, MA, USA, 2012. ISBN 1601986289, 9781601986283.

Simon Lacoste-Julien. Convergence Rate of Frank-Wolfe for Non-Convex Objectives. *arXiv:1607.00345 [cs, math, stat]*, July 2016.

Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing Systems*, pages 496–504, 2015.

Henry Lam. Robust Sensitivity Analysis for Stochastic Systems. *Mathematics of Operations Research*, 41(4):1248–1275, 2016.

Guanghui Lan. The complexity of large-scale convex programming under a linear optimization oracle. *arXiv preprint arXiv:1309.5550*, 2013.

Jason D. Lee, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1246–1257, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 1049–1065. IEEE, 2015.

Jing Lei. Convergence and concentration of empirical measures under wasserstein distance in unbounded functional spaces. *Bernoulli*, 26(1):767–798, 02 2020.

Kfir Y Levy. The power of normalization: Faster evasion of saddle points. *arXiv preprint arXiv:1611.04831*, 2016.

Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1718–1727, Lille, France, 07–09 Jul 2015. PMLR.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 510–520, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9.

Qiang Liu, Jason Lee, and Michael Jordan. A kernelized Stein discrepancy for goodness-of-fit tests. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 276–284, New York, New York, USA, 20–22 Jun 2016. PMLR.

L. Lovász. *Submodular functions and convexity*, pages 235–257. Springer Berlin Heidelberg, Berlin, Heidelberg, 1983. ISBN 978-3-642-68874-4.

Meghna Lowalekar, Pradeep Varakantham, and Akshat Kumar. Robust Influence Maximization: (Extended Abstract). In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, AAMAS '16, pages 1395–1396, Richland, SC, 2016. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-4239-1.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Takanori Maehara. Risk averse submodular utility maximization. *Operations Research Letters*, 43(5):526–529, September 2015.

Takanori Maehara, Akihiro Yabe, and Ken ichi Kawarabayashi. Budget allocation problem with multiple advertisers: A game theoretic view. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 428–437, Lille, France, 07–09 Jul 2015. PMLR.

Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. In *Conference on Learning Theory*, 2009.

H. Brendan McMahan and Matthew J. Streeter. Adaptive bound optimization for online convex optimization. In *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, pages 244–256, 2010.

N. Meinshausen. Causality from a distributional robustness point of view. In *2018 IEEE Data Science Workshop (DSW)*, pages 6–10, June 2018.

Michel Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In J. Stoer, editor, *Optimization Techniques*, pages 234–243, Berlin, Heidelberg, 1978. Springer Berlin Heidelberg. ISBN 978-3-540-35890-9.

Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, Amin Karbasi, Jan Vondrák, and Andreas Krause. Lazier than lazy greedy. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization. *Journal of Machine Learning Research*, 17(238):1–44, 2016.

Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the Wasserstein metric: performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166, Sep 2018.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Conditional gradient method for stochastic submodular maximization: Closing the gap. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1886–1895, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018a. PMLR.

Aryan Mokhtari, Asuman Ozdaglar, and Ali Jadbabaie. Escaping saddle points in constrained optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 3633–3643. Curran Associates, Inc., 2018b.

MOSEK ApS. *MOSEK MATLAB Toolbox 8.0.0.57*, 2015.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trendső in Machine Learning*, 10(1-2):1–141, 2017.

Mahesh Chandra Mukkamala and Matthias Hein. Variants of RMSProp and Adagrad with logarithmic regret bounds. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2545–2553, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Kazuo Murota. *Discrete convex analysis*. SIAM, 2003.

Kazuo Murota and Akiyoshi Shioura. Exact bounds for steepest descent algorithms of l-convex function minimization. *Operations Research Letters*, 42:361–366, 2014.

Kiyohito Nagano, Yoshinobu Kawahara, and Kazuyuki Aihara. Size-Constrained Submodular Minimization through Minimum Norm Base. In *ICML*, pages 977–984, 2011.

Hongseok Namkoong and John C. Duchi. Variance-based Regularization with Convex Objectives. In *Advances in Neural Information Processing Systems 30*, pages 2975–2984, 2017.

Harikrishna Narasimhan, David C Parkes, and Yaron Singer. Learnability of influence in networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3186–3194. Curran Associates, Inc., 2015.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions I. *Mathematical Programming*, 14(1):265–294, 1978.

Yurii Nesterov. *Introductory lectures on convex optimization : a basic course*. Kluwer Academic Publishers, Boston, 2004. ISBN 1402075537.

Yurii Nesterov and Boris T Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Praneeth Netrapalli and Sujay Sanghavi. Learning the graph of epidemic cascades. In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '12, pages 211–222, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1097-0.

Evdokia Nikolova. Approximation algorithms for reliable stochastic combinatorial optimization. In Maria Serna, Ronen Shaltiel, Klaus Jansen, and José Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 338–351, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15369-3.

Naoto Ohsaka and Yuichi Yoshida. Portfolio optimization for influence spread. In *Proceedings of the 26th International Conference on World Wide Web*, WWW ' 17, page 977–985, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee. ISBN 9781450349130.

Yonatan Oren, Shiori Sagawa, Tatsunori Hashimoto, and Percy Liang. Distributionally robust language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4227–4237, Hong Kong, China, November 2019. Association for Computational Linguistics.

James B. Orlin, Andreas Schulz, and Rajan Udwani. Robust monotone submodular function maximization. In *Conference on Integer Programming and Combinatorial Optimization (IPCO)*, 2016.

Luis D. Pascual and Adi Ben-Israel. Constrained maximization of posynomials by geometric programming. *Journal of Optimization Theory and Applications*, 5(2): 73–80, March 1970.

Boris T. Polyak. *Introduction to Optimization*. Number 04; QA402. 5, P6. 1987.

Sashank Reddi, Manzil Zaheer, Suvrit Sra, Barnabas Poczos, Francis Bach, Ruslan Salakhutdinov, and Alex Smola. A generic approach for escaping saddle points. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1233–1242, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018a. PMLR.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of Adam and beyond. In *International Conference on Learning Representations*, 2018b.

R Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

R Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of banking & finance*, 26(7):1443–1471, 2002.

Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020.

Herbert Scarf. A min-max solution of an inventory problem. *Studies in the mathematical theory of inventory and production*, 1958.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346 – 355, 2000.

Soroosh Shafieezadeh Abadeh, Peyman Mohajerin Mohajerin Esfahani, and Daniel Kuhn. Distributionally robust logistic regression. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1576–1584. Curran Associates, Inc., 2015.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(Feb): 567–599, 2013.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Shashank Singh and Barnabás Póczos. Minimax distribution estimation in wasserstein distance. *arXiv preprint arXiv:1802.08855*, 2018.

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying some distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.

Tasuku Soma and Yuichi Yoshida. A Generalization of Submodular Cover via the Diminishing Return Property on the Integer Lattice. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 847–855. Curran Associates, Inc., 2015.

Tasuku Soma, Naonori Kakimura, Kazuhiro Inaba, and Ken ichi Kawarabayashi. Optimal budget allocation: Theoretical guarantee and efficient algorithm. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 351–359, Bejing, China, 22–24 Jun 2014. PMLR.

Matthew Staib and Stefanie Jegelka. Distributionally robust deep learning as a generalization of adversarial training. In *NIPS Machine Learning and Computer Security Workshop*, 2017a.

Matthew Staib and Stefanie Jegelka. Robust budget allocation via continuous submodular functions. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3230–3240, International Convention Centre, Sydney, Australia, 06–11 Aug 2017b. PMLR.

Matthew Staib and Stefanie Jegelka. Wasserstein k-means++ for cloud regime histogram clustering. In *Proceedings of the Seventh International Workshop on Climate Informatics: CI 2017*, 2017c.

Matthew Staib and Stefanie Jegelka. Distributionally robust optimization and generalization in kernel methods. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 9131–9141. Curran Associates, Inc., 2019a.

Matthew Staib and Stefanie Jegelka. Robust budget allocation via continuous submodular functions. *Applied Mathematics & Optimization*, Mar 2019b.

Matthew Staib, Sebastian Claici, Justin M Solomon, and Stefanie Jegelka. Parallel streaming Wasserstein barycenters. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2644–2655. Curran Associates, Inc., 2017.

Matthew Staib, Sashank Reddi, Satyen Kale, Sanjiv Kumar, and Suvrit Sra. Escaping saddle points with adaptive gradient methods. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5956–5965, Long Beach, California, USA, 09–15 Jun 2019a. PMLR.

Matthew Staib, Bryan Wilder, and Stefanie Jegelka. Distributionally robust submodular maximization. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 506–516. PMLR, 16–18 Apr 2019b.

Serban Stan, Morteza Zadimoghaddam, Andreas Krause, and Amin Karbasi. Probabilistic submodular maximization in sub-linear time. In Doina Precup and

Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3241–3250, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Dougal J Sutherland, Hsiao-Yu Tung, Heiko Strathmann, Soumyajit De, Aaditya Ramdas, Alex Smola, and Arthur Gretton. Generative models and model criticism via optimized maximum mean discrepancy. In *International Conference on Learning Representations*, 2017.

Zoya Svitkina and Lisa Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM Journal on Computing*, 40(6):1715–1737, 2011.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

Donald M Topkis. Minimizing a submodular function on a lattice. *Operations research*, 26(2):305–321, 1978.

Joel Tropp. Freedman's inequality for matrix martingales. *Electronic Communications in Probability*, 16:262–270, 2011.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.

Cédric Villani. *Optimal Transport: Old and New (Grundlehren der mathematischen Wissenschaften)*. Springer, 2008. ISBN 9788793102132.

John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

Jan Vondrak. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC ' 08, page 67–74, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605580470.

Kevin Wainwright and Alpha Chiang. *Fundamental Methods of Mathematical Economics*. McGraw-Hill Education, 2004. ISBN 0070109109.

Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019.

Rachel Ward, Xiaoxia Wu, and Leon Bottou. AdaGrad stepsizes: Sharp convergence over nonconvex landscapes. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6677–6686, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Jonathan Weed, Francis Bach, et al. Sharp asymptotic and finite-sample rates of convergence of empirical measures in wasserstein distance. *Bernoulli*, 25(4A):2620–2648, 2019.

Bryan Wilder. Equilibrium computation and robust optimization in zero sum games with submodular structure. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018a.

Bryan Wilder. Risk-Sensitive Submodular Optimization. In *AAAI Conference on Artificial Intelligence*, 2018b.

Philip Wolfe. Finding the nearest point in a polytope. *Mathematical Programming*, 11(1):128–149, Dec 1976.

Huan Xu, Constantine Caramanis, and Shie Mannor. Robustness and regularization of support vector machines. *Journal of Machine Learning Research*, 10(Jul):1485–1510, 2009.

Yi Xu, Jing Rong, and Tianbao Yang. First-order stochastic algorithms for escaping from saddle points in almost linear time. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 5531–5541. Curran Associates, Inc., 2018.

Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9793–9803. Curran Associates, Inc., 2018.

Mingrui Zhang, Zebang Shen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. One sample stochastic frank-wolfe. *arXiv preprint arXiv:1910.04322*, 2019.

Dongruo Zhou, Yiqi Tang, Ziyan Yang, Yuan Cao, and Quanquan Gu. On the convergence of adaptive gradient methods for nonconvex optimization. *arXiv preprint arXiv:1808.05671*, 2018.

Fangyu Zou, Li Shen, Zequn Jie, Weizhong Zhang, and Wei Liu. A sufficient condition for convergences of adam and rmsprop. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Appendix A

DRO, MMD, kernels, and generalization

A.1 Proofs of main structural results

Proof of Theorem 3.3.1. We will use weak duality to derive a candidate solution, and then use that solution to show strong duality. First, note that

$$\sup_{\mu_{\mathbb{Q}} \in \mathcal{H}: \|\mu_{\mathbb{Q}} - \mu_{\mathbb{P}}\|_{\mathcal{H}} \leq \varepsilon} \langle f, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} = \sup_{\mu_{\mathbb{Q}} \in \mathcal{H}} \inf_{\lambda \geq 0} \left\{ \langle f, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} - \lambda(\|\mu_{\mathbb{Q}} - \mu_{\mathbb{P}}\|_{\mathcal{H}}^2 - \varepsilon^2) \right\} \tag{A.1}$$

$$\leq \inf_{\lambda \geq 0} \sup_{\mu_{\mathbb{Q}} \in \mathcal{H}} \left\{ \langle f, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} - \lambda(\|\mu_{\mathbb{Q}} - \mu_{\mathbb{P}}\|_{\mathcal{H}}^2 - \varepsilon^2) \right\} \tag{A.2}$$

$$= \inf_{\lambda \geq 0} \left\{ \lambda \varepsilon^2 + \sup_{\mu_{\mathbb{Q}} \in \mathcal{H}} \left\{ \langle f, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} - \lambda \|\mu_{\mathbb{Q}} - \mu_{\mathbb{P}}\|_{\mathcal{H}}^2 \right\} \right\}. \tag{A.3}$$

We first focus on the innermost objective, which may be rewritten:

$$\langle f, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} - \lambda \|\mu_{\mathbb{Q}} - \mu_{\mathbb{P}}\|_{\mathcal{H}}^2 = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}} + \langle f, \mu_{\mathbb{Q}} - \mu_{\mathbb{P}} \rangle_{\mathcal{H}} - \lambda \|\mu_{\mathbb{Q}} - \mu_{\mathbb{P}}\|_{\mathcal{H}}^2 \tag{A.4}$$

$$= \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}} - \lambda \left[\|\mu_{\mathbb{Q}} - \mu_{\mathbb{P}}\|_{\mathcal{H}}^2 - 2 \left\langle \frac{1}{2\lambda} f, \mu_{\mathbb{Q}} - \mu_{\mathbb{P}} \right\rangle_{\mathcal{H}} \right] \tag{A.5}$$

$$= \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}} - \lambda \left[\left\| \mu_{\mathbb{Q}} - \mu_{\mathbb{P}} - \frac{1}{2\lambda} f \right\|_{\mathcal{H}}^2 + \left\| \frac{1}{2\lambda} f \right\|_{\mathcal{H}}^2 \right], \tag{A.6}$$

where the final inequality is by completing the square. Only one term depends on $\mu_{\mathbb{Q}}$, namely $-\lambda\|\mu_{\mathbb{Q}} - \mu_{\mathbb{P}} - \frac{1}{2\lambda}f\|_{\mathcal{H}}^2$; since norms are nonnegative, this term can never exceed zero, and zero is achieved by $\mu_{\mathbb{Q}}^* = \mu_{\mathbb{P}} + \frac{1}{2\lambda}f \in \mathcal{H}$, yielding inner objective value

$$\langle f, \mu_{\mathbb{P}}\rangle_{\mathcal{H}} - \lambda \left\| \frac{1}{2\lambda} f \right\|_{\mathcal{H}}^2 = \langle f, \mu_{\mathbb{P}}\rangle_{\mathcal{H}} - \frac{1}{4\lambda}\|f\|_{\mathcal{H}}^2. \tag{A.7}$$

Plugging this in for the inner problem, and then solving for the optimal dual variable λ^*, we derive the upper bound:

$$\sup_{\mu_{\mathbb{Q}} \in \mathcal{H}: \|\mu_{\mathbb{Q}} - \mu_{\mathbb{P}}\|_{\mathcal{H}} \leq \varepsilon} \langle f, \mu_{\mathbb{Q}}\rangle_{\mathcal{H}} \leq \inf_{\lambda \geq 0} \left\{ \lambda\varepsilon^2 + \langle f, \mu_{\mathbb{P}}\rangle_{\mathcal{H}} + \frac{1}{4\lambda}\|f\|_{\mathcal{H}}^2 \right\} \tag{A.8}$$

$$= \langle f, \mu_{\mathbb{P}}\rangle_{\mathcal{H}} + \varepsilon\|f\|_{\mathcal{H}}. \tag{A.9}$$

The optimal dual variable $\lambda^* = \frac{1}{2\varepsilon}\|f\|_{\mathcal{H}}$ is that which balances the two terms. Plugging this in, we find that $\mu_{\mathbb{Q}}^* = \mu_{\mathbb{P}} + \frac{\varepsilon}{\|f\|_{\mathcal{H}}} f$.

In order to prove equality, it remains to show strong duality holds. We will achieve this by lower bounding the original objective. Specifically, the supremum over all $\mu_{\mathbb{Q}}$ can be lower bounded by plugging in our particular $\mu_{\mathbb{Q}}^*$:

$$\sup_{\mu_{\mathbb{Q}} \in \mathcal{H}: \|\mu_{\mathbb{Q}} - \mu_{\mathbb{P}}\|_{\mathcal{H}} \leq \varepsilon} \langle f, \mu_{\mathbb{Q}}\rangle_{\mathcal{H}} = \sup_{\mu_{\mathbb{Q}} \in \mathcal{H}} \inf_{\lambda \geq 0} \left\{ \langle f, \mu_{\mathbb{Q}}\rangle_{\mathcal{H}} - \lambda(\|\mu_{\mathbb{Q}} - \mu_{\mathbb{P}}\|_{\mathcal{H}}^2 - \varepsilon^2) \right\} \tag{A.10}$$

$$\geq \inf_{\lambda \geq 0} \left\{ \langle f, \mu_{\mathbb{Q}}^*\rangle_{\mathcal{H}} - \lambda(\|\mu_{\mathbb{Q}}^* - \mu_{\mathbb{P}}\|_{\mathcal{H}}^2 - \varepsilon^2) \right\} \tag{A.11}$$

$$= \inf_{\lambda \geq 0} \left\{ \left\langle f, \mu_{\mathbb{P}} + \frac{\varepsilon}{\|f\|_{\mathcal{H}}} f \right\rangle_{\mathcal{H}} - \lambda \left(\left\| \frac{\varepsilon}{\|f\|_{\mathcal{H}}} f \right\|_{\mathcal{H}}^2 - \varepsilon^2 \right) \right\} \tag{A.12}$$

$$= \inf_{\lambda \geq 0} \left\{ \left\langle f, \mu_{\mathbb{P}} + \frac{\varepsilon}{\|f\|_{\mathcal{H}}} f \right\rangle_{\mathcal{H}} - \lambda \left(\varepsilon^2 - \varepsilon^2 \right) \right\} \tag{A.13}$$

$$= \left\langle f, \mu_{\mathbb{P}} + \frac{\varepsilon}{\|f\|_{\mathcal{H}}} f \right\rangle_{\mathcal{H}} = \langle f, \mu_{\mathbb{P}}\rangle_{\mathcal{H}} + \varepsilon\|f\|_{\mathcal{H}}. \tag{A.14}$$

Since the same bound appears on both sides, we have equality. □

A.2 Gaussian kernel bounds

We first reproduce Proposition 3.4.1 for convenience:

Proposition A.2.1. *Let $f, g \in \mathcal{H}_\sigma$ have the expansions $f = \sum_i a_i k_\sigma(x_i, \cdot)$ and $g = \sum_j b_j k_\sigma(x_j, \cdot)$. For shorthand denote by $z_i = \phi_{\sqrt{2}\sigma}(x_i)$ the (possibly infinite) feature expansion of x_i in $\mathcal{H}_{\sqrt{2}\sigma}$. Then*

$$\|fg\|^2_{\sigma/\sqrt{2}} = \mathrm{tr}(A^2B^2), \quad \|f\|^2_\sigma = \mathrm{tr}(A^2), \quad \textit{and} \quad \|g\|^2_\sigma = \mathrm{tr}(B^2),$$

where $A = \sum_i a_i z_i z_i^T$ and $B = \sum_j a_j z_j z_j^T$.

In order to prove Proposition 3.4.1, we will need a utility lemma that helps translate between $\mathcal{H}_\sigma$ and $\mathcal{H}_{\sigma/\sqrt{2}}$:

Lemma A.2.1. *Let $\langle \cdot, \cdot \rangle_{\sigma/\sqrt{2}}$ be the inner product in the RKHS $\mathcal{H}_{\sigma/\sqrt{2}}$. Let $\langle \cdot, \cdot \rangle_{\sigma'}$ refer to the inner product in $H_{\sigma'}$. Then,*

$$\langle k_\sigma(x, \cdot)k_\sigma(y, \cdot), k_\sigma(a, \cdot)k_\sigma(b, \cdot) \rangle_{\sigma/\sqrt{2}} \tag{A.15}$$

can be simplified as

$$k_{\sigma\sqrt{2}}(x, a)k_{\sigma\sqrt{2}}(x, b)k_{\sigma\sqrt{2}}(y, a)k_{\sigma\sqrt{2}}(y, b). \tag{A.16}$$

In order to make the proof cleaner, we first derive a couple of identities involving norms and sums.

Lemma A.2.2. *Let x, y, z be vectors in an inner product space with norm $\|\cdot\|$. Then the following identity holds:*

$$\|x - z\|^2 + \|y - z\|^2 = \frac{1}{2}\|x - y\|^2 + 2\left\|z - \frac{x + y}{2}\right\|^2. \tag{A.17}$$

Proof. The parallelogram law states that for u, v in an an inner product space with

norm $\|\cdot\|$, it holds that

$$2\|u\|^2 + 2\|u\|^2 = \|u - v\|^2 + \|u + v\|^2. \tag{A.18}$$

Set $u = x - z$ and $v = y - z$, and note that $u - v = x - y$ and $u + v = 2z - (x + y)$. Then, for the norm $\|\cdot\|$, it follows that:

$$2\|x - z\|^2 + 2\|y - z\|^2 = \|x - y\|^2 + \|2z - (x + y)\|^2. \tag{A.19}$$

The result follows by dividing by two. □

Lemma A.2.3. *Let x, y, a, b be arbitrary vectors in an inner product space with inner product $\langle \cdot, \cdot \rangle$ and norm $\|\cdot\|$. Define S and T by:*

$$S := \|x - y\|^2 + \|a - b\|^2 + \|(x + y) - (a + b)\|^2$$
$$T := \|x - a\|^2 + \|x - b\|^2 + \|y - a\|^2 + \|y - b\|^2.$$

Then $S = T$.

Proof. Start by expanding the third term of S:

$$\|x - y\|^2 + \|a - b\|^2 + \|(x + y) - (a + b)\|^2 \tag{A.20}$$
$$= \|x - y\|^2 + \|a - b\|^2 + \|(x - a) + (y - b)\|^2 \tag{A.21}$$
$$= \|x - y\|^2 + \|a - b\|^2 + 2\langle x - a, y - b\rangle + \|x - a\|^2 + \|y - b\|^2. \tag{A.22}$$

The first three terms of equation (A.22) can be expanded as

$$\begin{aligned}
&\|x-y\|^2 + \|a-b\|^2 + 2\langle x-a, y-b\rangle && \text{(A.23)}\\
&= \|x\|^2 + \|y\|^2 - 2\langle x, y\rangle + \|a\|^2 + \|b\|^2 && \text{(A.24)}\\
&\quad - 2\langle a, b\rangle + 2\langle x-a, y-b\rangle && \text{(A.25)}\\
&= \|x\|^2 + \|y\|^2 - 2\langle x, y\rangle + \|a\|^2 + \|b\|^2 && \text{(A.26)}\\
&\quad - 2\langle a, b\rangle + 2\langle x, y\rangle - 2\langle x, b\rangle - 2\langle a, y\rangle + 2\langle a, b\rangle && \text{(A.27)}\\
&= \|x\|^2 + \|y\|^2 + \|a\|^2 + \|b\|^2 - 2\langle x, b\rangle - 2\langle a, y\rangle && \text{(A.28)}\\
&= \|x-b\|^2 + \|y-a\|^2. && \text{(A.29)}
\end{aligned}$$

Replacing the first three terms in equation (A.22) by $\|x-b\|^2 + \|y-a\|^2$ yields T, i.e. $S = T$. □

We are now equipped to prove Lemma A.2.1:

Proof of Lemma A.2.1. First, write

$$\begin{aligned}
k_\sigma(x,z)k_\sigma(y,z) &= \exp\left(-\frac{1}{2\sigma^2}\left(\|x-z\|^2 + \|y-z\|^2\right)\right) && \text{(A.30)}\\
&= \exp\left(-\frac{1}{2\sigma^2}\left(\frac{1}{2}\|x-y\|^2 + 2\left\|z - \frac{x+y}{2}\right\|^2\right)\right) && \text{(A.31)}\\
&= \exp\left(-\frac{1}{4\sigma^2}\|x-y\|^2\right)\exp\left(-\frac{1}{\sigma^2}\left\|z-\frac{x+y}{2}\right\|^2\right) && \text{(A.32)}\\
&= k_{\sigma\sqrt{2}}(x,y)k_{\sigma/\sqrt{2}}\left(z, \frac{x+y}{2}\right), && \text{(A.33)}
\end{aligned}$$

where in the second line we used Lemma A.2.2. Note that the first term does not

depend on z. Now, applying this identity to Equation (A.15), we find:

$$\langle k_\sigma(x,\cdot)k_\sigma(y,\cdot), k_\sigma(a,\cdot)k_\sigma(b,\cdot)\rangle_{\sigma/\sqrt{2}} \tag{A.34}$$

$$= k_{\sigma\sqrt{2}}(x,y)k_{\sigma\sqrt{2}}(a,b)\left\langle k_{\sigma/\sqrt{2}}\left(\frac{x+y}{2},\cdot\right), k_{\sigma/\sqrt{2}}\left(\frac{a+b}{2},\cdot\right)\right\rangle_{\sigma/\sqrt{2}} \tag{A.35}$$

$$= k_{\sigma\sqrt{2}}(x,y)k_{\sigma\sqrt{2}}(a,b)k_{\sigma/\sqrt{2}}\left(\frac{x+y}{2},\frac{a+b}{2}\right) \tag{A.36}$$

$$= k_{\sigma\sqrt{2}}(x,y)k_{\sigma\sqrt{2}}(a,b)k_{\sigma\sqrt{2}}\left(x+y,a+b\right). \tag{A.37}$$

To simplify this expression, notice that it takes the form $\exp(-S/(4\sigma^2))$, where

$$S = \|x-y\|^2 + \|a-b\|^2 + \|(x+y)-(a+b)\|^2. \tag{A.38}$$

By Lemma A.2.3, S is equal to

$$S = \|x-a\|^2 + \|x-b\|^2 + \|y-a\|^2 + \|y-b\|^2, \tag{A.39}$$

which means equation (A.37) can be rewritten as

$$\begin{aligned}\exp\left(-\frac{S}{4\sigma^2}\right) &= \exp\left(-\frac{\|x-a\|^2}{4\sigma^2}\right)\exp\left(-\frac{\|x-b\|^2}{4\sigma^2}\right)\exp\left(-\frac{\|y-a\|^2}{4\sigma^2}\right)\exp\left(-\frac{\|y-b\|^2}{4\sigma^2}\right)\\ &= k_{\sigma\sqrt{2}}(x,a)k_{\sigma\sqrt{2}}(x,b)k_{\sigma\sqrt{2}}(y,a)k_{\sigma\sqrt{2}}(y,b).\end{aligned}$$

□

With Lemma A.2.1 available, it is possible to prove Proposition 3.4.1:

Proof of Proposition 3.4.1. Define the vectors z_i as described, so that $z_i^T z_j = k_{\sqrt{2}\sigma}(x_i,x_j)$. For convenience, also write $K_{ij} = k_{\sqrt{2}\sigma}(x_i,x_j)$, and observe that $K_{ij}^2 = k_\sigma(x_i,x_j)$. It follows that

$$\|f\|_\sigma^2 = \sum_i\sum_j a_i a_j k_\sigma(x_i,x_j) = \sum_i\sum_j a_i a_j K_{ij}^2 = \sum_i\sum_j a_i a_j z_i^T z_j z_j^T z_i \tag{A.40}$$

Rearranging the inner terms, we find

$$\|f\|_\sigma^2 = \sum_i a_i z_i^T \left(\sum_j a_j z_j z_j^T \right) z_i = \sum_i a_i z_i^T A z_i = \operatorname{tr}\left(\sum_i a_i z_i z_i^T A \right) = \operatorname{tr}(A^2), \tag{A.41}$$

where we have used the definition of A, the fact that the trace of a scalar is simply that scalar, and the cyclic property of the trace. The proof that $\|g\|_\sigma^2 = \operatorname{tr}(B^2)$ is identical, so we omit it.

The derivation of the trace form of $\|fg\|_{\sigma/\sqrt{2}}^2$ is more complicated. Expanding out fg, we see that

$$(fg)(x) = \sum_{i,j} a_i b_j k_\sigma(x_i, x) k_\sigma(x_j, x). \tag{A.42}$$

Therefore the norm $\|fg\|_{\sigma/\sqrt{2}}^2$, which is simply $\langle fg, fg \rangle_{\sigma/\sqrt{2}}$, is equal to:

$$\langle fg, fg \rangle_{\sigma/\sqrt{2}} = \left\langle \sum_{i,j} a_i b_j k_\sigma(x_i, x) k_\sigma(x_j, x), \sum_{i',j'} a_{i'} b_{j'} k_\sigma(x_{i'}, x) k_\sigma(x_{j'}, x) \right\rangle_{\sigma/\sqrt{2}} \tag{A.43}$$

$$= \sum_{i,j,i',j'} a_i a_{i'} b_j b_{j'} \left\langle k_\sigma(x_i, x) k_\sigma(x_j, x), k_\sigma(x_{i'}, x) k_\sigma(x_{j'}, x) \right\rangle_{\sigma/\sqrt{2}} \tag{A.44}$$

$$= \sum_{i,j,i',j'} a_i a_{i'} b_j b_{j'} k_{\sigma\sqrt{2}}(x_i, x_{i'}) k_{\sigma\sqrt{2}}(x_i, x_{j'}) k_{\sigma\sqrt{2}}(x_j, x_{i'}) k_{\sigma\sqrt{2}}(x_j, x_{j'}) \tag{A.45}$$

$$= \sum_{i,j,i',j'} a_i a_{i'} b_j b_{j'} K_{ii'} K_{ij'} K_{ji'} K_{jj'}, \tag{A.46}$$

where in the second to last step we have used Lemma A.2.1. Before continuing, observe the identity

$$\sum_\ell a_\ell K_{i\ell} K_{j\ell} = \sum_\ell a_\ell z_i^T z_\ell z_\ell^T z_j = z_i^T \left(\sum_\ell a_\ell z_\ell z_\ell^T \right) z_j = z_i^T A z_j \tag{A.47}$$

Similarly, $\sum_\ell b_\ell K_{i\ell} K_{j\ell} = z_i^T B z_j$. Leveraging these identities, we continue:

$$\sum_{i,j,i',j'} a_i a_{i'} b_j b_{j'} K_{ii'} K_{ij'} K_{ji'} K_{jj'} = \sum_{i,i',j} a_i a_{i'} b_j K_{ii'} K_{ji'} \sum_{j'} b_{j'} K_{ij'} K_{jj'} \tag{A.48}$$

$$= \sum_{i,i',j} a_i a_{i'} b_j K_{ii'} K_{ji'} (z_i^T B z_j) \tag{A.49}$$

$$= \sum_{i,j} a_i b_j \left(\sum_{i'} a_{i'} K_{ii'} K_{ji'} \right) (z_i^T B z_j) \tag{A.50}$$

$$= \sum_{i,j} a_i b_j (z_j^T A z_i)(z_i^T B z_j). \tag{A.51}$$

At this point we leverage the cyclic property of the trace, so the above expression equals:

$$\operatorname{tr}\left(\sum_{i,j} a_i b_j A z_i z_i^T B z_j z_j^T \right) = \operatorname{tr}\left(A \left(\sum_i a_i z_i z_i^T \right) B \left(\sum_j b_j z_j z_j^T \right) \right) = \operatorname{tr}(A^2 B^2). \quad \square$$

A.2.1 Trace inequality

Proof of Lemma 3.4.1. Consider the trace inner product $\langle X, Y \rangle = \operatorname{tr}(X^T Y) = \operatorname{tr}(XY)$, where the final equality is because X is symmetric. By the Cauchy-Schwarz inequality, we have $\operatorname{tr}(XY) \leq \sqrt{\operatorname{tr}(X^2)\operatorname{tr}(Y^2)}$. Let $\{\lambda_i\}_{i=1}^n$ be the eigenvalues of X. Then,

$$\operatorname{tr}(X^2) = \sum_{i=1}^n \lambda_i^2 \leq \sum_{i=1}^n \lambda_i^2 + 2 \sum_{i=1}^n \sum_{j=i+1}^n \lambda_i \lambda_j = \left(\sum_{i=1}^n \lambda_i \right)^2 = \operatorname{tr}(X)^2, \tag{A.52}$$

where the inequality holds because λ_i are all nonnegative. The same holds for any positive semidefinite matrix, in particular, Y. Combining these two inequalities, we have

$$\operatorname{tr}(XY) \leq \sqrt{\operatorname{tr}(X^2)\operatorname{tr}(Y^2)} \leq \sqrt{\operatorname{tr}(X)^2 \operatorname{tr}(Y)^2} = \operatorname{tr}(X)\operatorname{tr}(Y). \tag{A.53}$$

$\square$

A.2.2 Extensions of Proposition 3.4.1

There are many useful corollaries and extensions of Proposition 3.4.1. Here, we give a result that makes it tractable to actually compute $\|fg\|_{\sigma/\sqrt{2}}$:

Corollary A.2.1. *Suppose* $f = \sum_{i=1}^n a_i k_\sigma(x_i, \cdot)$ *and* $g = \sum_{i=1}^n b_i k_\sigma(x_i, \cdot)$ *have the same finite expansion, but with potentially different coefficients. Form the kernel matrix* K *with* $K_{ij} = k_{\sqrt{2}\sigma}(x_i, x_j)$*, where we have replaced the bandwidth* σ *with* $\sqrt{2}\sigma$*. Write* $D_a = \mathrm{diag}(a)$ *and similarly for* D_b*. Then,*

$$\|fg\|_{\sigma/\sqrt{2}}^2 = \mathrm{tr}((D_a K)^2 (D_b K)^2). \tag{A.54}$$

Proof. Pick vectors z_i so that $z_i^T z_j = K_{ij}$, and let Z be the matrix with i-th column z_i. Note that $A = \sum_{i=1}^n a_i z_i z_i^T = Z D_a Z^T$, and similarly for B. Then we may write

$$\begin{aligned}
\|fg\|_{\sigma/\sqrt{2}}^2 &\overset{(a)}{=} \mathrm{tr}(A^2 B^2) && \text{(A.55)}\\
&= \mathrm{tr}((Z D_a Z^T)(Z D_a Z^T)(Z D_b Z^T)(Z D_b Z^T)) && \text{(A.56)}\\
&\overset{(b)}{=} \mathrm{tr}(D_a Z^T Z D_a Z^T Z D_b Z^T Z D_b Z^T Z) && \text{(A.57)}\\
&\overset{(c)}{=} \mathrm{tr}(D_a K D_a K D_b K D_b K) && \text{(A.58)}\\
&= \mathrm{tr}((D_a K)^2 (D_b K)^2), && \text{(A.59)}
\end{aligned}$$

where (a) is by Proposition 3.4.1, (b) is by the cyclic property of the trace, and (c) follows since $Z^T Z = K$ by definition of z_i. □

A.3 Proofs for Section 3.5

Proof of Lemma 3.5.1. For notational convenience, we just write ℓ instead of $\vec{\ell}$. First, notice that problem (3.22), once the $w_i \geq 0$ constraint is dropped, can be written

$$\begin{aligned}
\sup_w \quad & \ell^T w \\
\text{s.t.} \quad & \left(w - \tfrac{1}{n}\mathbf{1}\right)^T K \left(w - \tfrac{1}{n}\mathbf{1}\right) \leq \varepsilon^2 \\
& \mathbf{1}^T w = 1
\end{aligned} \tag{A.60}$$

Write $v = w - \frac{1}{n}\mathbf{1}$. Then the value of problem (A.60) is equal to

$$\begin{array}{lll} & \sup_v & \ell^T v \\ \frac{1}{n}\mathbf{1}^T\ell + & \text{s.t.} & v^T K v \leq \varepsilon^2 \\ & & \mathbf{1}^T v = 0 \end{array} \tag{A.61}$$

and we can focus on this slightly simpler problem. This problem can be in turn rewritten as:

$$\sup_v \inf_{\eta \geq 0, \lambda} \left\{ \ell^T v - \eta(v^T K v - \varepsilon^2) - \lambda \mathbf{1}^T v \right\}. \tag{A.62}$$

Slater's condition holds since $v = 0$ is strictly feasible. Therefore strong duality holds, so the optimal value is equal to:

$$\inf_{\eta \geq 0, \lambda} \left\{ \eta\varepsilon^2 + \sup_v \left\{ \ell^T v - \eta v^T K v - \lambda \mathbf{1}^T v \right\} \right\} \tag{A.63}$$

$$= \inf_{\eta \geq 0, \lambda} \left\{ \eta\varepsilon^2 + \sup_v \left\{ -\eta v^T K v + (\ell - \lambda\mathbf{1})^T v \right\} \right\}. \tag{A.64}$$

The inner problem is a concave quadratic maximization problem. In general, if A is symmetric, $-x^T A x + b^T x$ is maximized when $x = \frac{1}{2}A^{-1}b$, and the resulting objective value is $\frac{1}{4}b^T A^{-1} b$. Applying this to the problem at hand, we find that the optimal v^* satisfies:

$$v^* = \frac{1}{2\eta} K^{-1}(\ell - \lambda\mathbf{1}), \tag{A.65}$$

and the corresponding objective value of the inner problem is

$$\frac{1}{4\eta}(\ell - \lambda\mathbf{1})^T K^{-1}(\ell - \lambda\mathbf{1}). \tag{A.66}$$

The overall problem is therefore

$$\inf_{\eta \geq 0, \lambda} \left\{ \eta\varepsilon^2 + \frac{1}{4\eta}(\ell - \lambda\mathbf{1})^T K^{-1}(\ell - \lambda\mathbf{1}) \right\}. \tag{A.67}$$

The objective is a convex quadratic in λ, and it is simple to check that $\lambda^* = (\mathbf{1}^T K^{-1}\ell)/(\mathbf{1}^T K^{-1}\mathbf{1})$. Then, both remaining terms are positive, so it is optimal to balance them. This leads to

$$\eta^*\varepsilon^2 = \frac{1}{4\eta^*}(\ell - \lambda^*\mathbf{1})^T K^{-1}(\ell - \lambda^*\mathbf{1}) \tag{A.68}$$

$$\implies \frac{1}{2\eta^*} = \frac{\varepsilon}{\sqrt{(\ell - \lambda^*\mathbf{1})^T K^{-1}(\ell - \lambda^*\mathbf{1})}}, \tag{A.69}$$

and the overall optimal value is

$$2 \cdot \frac{1}{4\eta^*}(\ell - \lambda^*\mathbf{1})^T K^{-1}(\ell - \lambda^*\mathbf{1}) \tag{A.70}$$

$$= \varepsilon\sqrt{(\ell - \lambda^*\mathbf{1})^T K^{-1}(\ell - \lambda^*\mathbf{1})}. \tag{A.71}$$

The term inside the square root is equal to

$$(\ell - \lambda^*\mathbf{1})^T K^{-1}(\ell - \lambda^*\mathbf{1}) = \ell^T K^{-1}\ell - 2\lambda^*\mathbf{1}^T K^{-1}\ell + (\lambda^*)^2\mathbf{1}^T K^{-1}\mathbf{1} \tag{A.72}$$

$$= \ell^T K^{-1}\ell - \frac{(\mathbf{1}^T K^{-1}\ell)^2}{\mathbf{1}^T K^{-1}\mathbf{1}}, \tag{A.73}$$

from which we can simply compute the overall objective of the original problem. □

Proof of Lemma 3.5.2. One can prove via the matrix inversion lemma that

$$K^{-1} = (aI + b\mathbf{1}\mathbf{1}^T)^{-1} = a^{-1}\left[I - \frac{b}{a+bn}\mathbf{1}\mathbf{1}^T\right]. \tag{A.74}$$

As a consequence,

$$a\ell^T K^{-1}\ell = \|\ell\|^2 - \frac{b}{a+bn}(\mathbf{1}^T\ell)^2 \tag{A.75}$$

$$a\ell^T K^{-1}\mathbf{1} = \mathbf{1}^T\ell - \frac{b}{a+bn}(\mathbf{1}^T\ell)(\mathbf{1}^T\mathbf{1}) = \frac{a}{a+bn}\cdot\mathbf{1}^T\ell \tag{A.76}$$

$$a\mathbf{1}^T K^{-1}\mathbf{1} = \mathbf{1}^T\mathbf{1} - \frac{b}{a+bn}(\mathbf{1}^T\mathbf{1})^2 = \frac{a}{a+bn}\cdot n. \tag{A.77}$$

It follows that

$$\frac{(a\ell^T K^{-1}\mathbf{1})^2}{a\mathbf{1}^T K^{-1}\mathbf{1}} = a \cdot \frac{\left(\frac{a}{a+bn}\mathbf{1}^T\ell\right)^2}{\frac{a}{a+bn}\cdot n} = a \cdot \frac{(\mathbf{1}^T\ell)^2}{n} \cdot \frac{a}{a+bn} = a \cdot (\mathbf{1}^T\ell)^2 \cdot \left(\frac{1}{n} - \frac{b}{a+bn}\right) \tag{A.78}$$

and therefore

$$a \cdot \left[\ell^T K^{-1}\ell - \frac{(\ell^T K^{-1}\mathbf{1})^2}{\mathbf{1}^T K^{-1}\mathbf{1}}\right] = a \cdot \left[\|\ell\|^2 - \frac{b}{a+bn}(\mathbf{1}^T\ell)^2 - (\mathbf{1}^T\ell)^2 \cdot \left(\frac{1}{n} - \frac{b}{a+bn}\right)\right] \tag{A.79}$$

$$= a \cdot \left[\|\ell\|^2 - \frac{(\mathbf{1}^T\ell)^2}{n}\right] = a \cdot \mathrm{Var}_{\hat{\mathbb{P}}_n}(\ell), \tag{A.80}$$

from which the conclusion follows. □

Appendix B

Distributionally robust submodular maximization

B.1 Tail Bound

We use the following one-sided Bernstein's inequality:

Lemma B.1.1 (Wainwright (2019), Chapter 2)**.** *Let $X_1, \ldots, X_n$ be iid realizations of a random variable X which satisfies $X \leq B$ almost surely. Then,*

$$\Pr\left[\frac{1}{n}\sum_{i=1}^{n} X_i - \mathbb{E}[X] \geq \varepsilon\right] \leq \exp\left(-\frac{n\varepsilon^2}{\mathrm{Var}\,(X) + \frac{B\varepsilon}{3}}\right).$$

We apply Lemma B.1.1 with $X_i = f_i(S)$. If we set the probability on the right hand side to be at most δ, then a simple calculation shows that it suffices to have $n = \frac{\mathrm{Var}(X)}{\varepsilon^2}\log\frac{1}{\delta} + \frac{B\varepsilon}{3}\log\frac{1}{\delta}$. Hence, for a given value of n, we can guarantee error of at most

$$\varepsilon = \sqrt{2\log\left(\frac{1}{\delta}\right)\frac{\mathrm{Var}(X)}{n}} + \frac{2}{3}\log\left(\frac{1}{\delta}\right)\frac{B}{n}.$$

Therefore, we can take $C_1 = \sqrt{2\log\frac{1}{\delta}}$ and $C_2 = \frac{2B}{3}\log\frac{1}{\delta}$. For many submodular maximization problems, B can be bounded by the ground set size $|V|$. For instance,

this bound holds for influence maximization, though tighter bounds may be available for specific graphs and distributions.

B.2 Equivalence of Variance Regularization and Distributionally Robust Optimization

Lemma B.2.1. *Suppose that $f(\{i\}) \leq b$ for all f in the support of $\mathbb{P}$ and all $i \in V$. Then, for each such f, its multilinear extension F is b-Lipschitz in the ℓ_1 norm.*

Proof. Consider any two points $x, x' \in [0,1]^{|V|}$ and any function f in the support of $\mathbb{P}$. Without loss of generality, let $f(x') \geq f(x)$. Let $[x]^+ = \max(x, 0)$ elementwise, $\vee$ denote elementwise minimum, and e_i be the i-th standard basis vector. Then we bound $F(x')$:

$$\begin{aligned}
F(x') &\overset{(a)}{\leq} F(x' \vee x) \\
&= F(x + [x' - x]^+) \\
&\overset{(b)}{\leq} F(x) + F([x' - x]^+) \\
&\overset{(c)}{\leq} F(x) + \sum_{i=1}^{|V|} F([x' - x]_i^+ e_i) \\
&\overset{(d)}{=} F(x) + \sum_{i=1}^{|V|} f(\{i\})[x' - x]_i^+ \\
&\overset{(e)}{\leq} F(x) + b \sum_{i=1}^{|V|} [x' - x]_i^+ \\
&\leq F(x) + b\|x' - x\|_1.
\end{aligned}$$

Here, (a) follows from monotonicity, while (b) and (c) follow because submodular functions are subadditive, i.e., $F(x+y) \leq F(x)+F(y)$. Then, (d) follows by definition of the multilinear extension, and (e) by assumption. Rearranging yields $|F(x') - F(x)| \leq b\|x - x'\|_1$ as desired. □

We will use the following concentration result for the sample variance of a random

variable:

Lemma B.2.2 (Namkoong and Duchi (2017), Section A.1)**.** *Let Z be a random variable bounded in $[0, B]$ and $z_1, \ldots, z_n$ be iid realizations of Z with $n \geq 64$. Let σ^2 denote $\mathrm{Var}(Z)$ and s_n^2 denote the sample variance. Then $s_n^2 \geq \frac{1}{4}\sigma^2$ with probability at least $1 - \exp\left(-\frac{n\sigma^2}{36B^2}\right)$.*

This allows us to get a uniform result for the variance expansion of the distributionally robust objective:

Corollary B.2.1. *Let $\mathcal{X}$ be the polytope $\{x \in [0,1]^{|V|} : \sum_{i=1}^{|V|} x_i = k\}$ corresponding to the k-uniform matroid. With probability at least $1 - \delta$, for all $x \in \mathcal{X}$ such that*

$$\mathrm{Var}_{\mathbb{P}}(F(x)) \geq \frac{\max\left\{\sqrt{\frac{32}{7}\rho B^2}, \sqrt{36B^2\left(\log\left(\frac{1}{\delta}\right) + |V|\log\left(1 + 24k\right)\right)}\right\}}{\sqrt{n}},$$

the variance expansion holds with equality.

Proof. Let $\mathcal{X}_{\geq\alpha} = \{x : \mathrm{Var}_{\mathbb{P}}(F(x)) \geq \alpha\}$ be the set of points x with variance at least α. Let $\mathcal{Y}$ be a minimal ℓ_1-cover of $\mathcal{X}_{\geq\alpha}$ with fineness $\frac{\varepsilon}{b}$, for a parameter ε to be fixed later. Since the ℓ_1-diameter of $\mathcal{X}$ is $2k$ (by definition), we know that $|\mathcal{Y}| \leq \left(1 + \frac{2kb}{\varepsilon}\right)^{|V|}$. Let $s_n^2(x)$ be the sample variance of $F_1(x), \ldots, F_n(x)$ and $\sigma^2(x) = \mathrm{Var}_{\mathbb{P}}(F(x))$. Via Lemma B.2.2 and union bound over all elements in $\mathcal{Y}$, we have

$$\Pr\left[s_n^2(x) \geq \frac{1}{4}\sigma^2(x)\ \forall x \in \mathcal{Y}\right] \geq 1 - |\mathcal{Y}|\exp\left(-\frac{n\alpha^2}{36B^2}\right).$$

Denote by $\mathcal{E}$ the above event, i.e. the event that $s_n^2(x) \geq \frac{1}{4}\sigma^2(x)$. Conditioning on $\mathcal{E}$, we will proceed to extend the sample variance lower bound to the entirety of $\mathcal{X}_{\geq\alpha}$.

Consider any $x \in \mathcal{X}_{\geq\alpha}$, and let $x' \in \arg\min_{x' \in \mathcal{Y}} \|x - x'\|_1$ be the closest point to x in the cover $\mathcal{Y}$. By definition of $\mathcal{Y}$, $\|x - x\|_1 \leq \frac{\varepsilon}{b}$, and so by Lemma B.2.1, which guarantees Lipschitzness of each F_i, we have $|F_i(x) - F_i(x')| \leq \varepsilon$ for all i. Accordingly, it can be shown that $|s_n(x) - s_n(x')| \leq \varepsilon$ and $|\sigma(x) - \sigma(x')| \leq \varepsilon$. It follows that:

$$s_n(x) \geq s_n(x') - \varepsilon \geq \frac{1}{2}\sigma(x') - \varepsilon \geq \frac{1}{2}\sigma(x) - \frac{3}{2}\varepsilon.$$

Now, conditioned on the event $\mathcal{E}$, we can set $\varepsilon = \frac{\alpha}{24}$ in order to ensure that $s_n(x) \geq \frac{7}{16}\alpha$.

All that is left is to determine the appropriate setting for α. We would like the exact variance expansion to hold on all elements of $\mathcal{X}_{\geq\alpha}$ with probability at least $1-\delta$. To have sufficiently high population variance, we must take $\alpha \geq \sqrt{\frac{16}{7} \cdot \frac{2\rho B^2}{n}}$. In order for the concentration bound to hold, a simple calculation shows that

$$\alpha \geq \sqrt{\frac{36B^2\left(\log\left(\frac{1}{\delta}\right) + |V|\log\left(1+24k\right)\right)}{n}}$$

suffices. Taking the max of the two, we require

$$\alpha \geq \frac{\max\left\{\sqrt{\frac{32}{7}\rho B^2}, \sqrt{36B^2\left(\log\left(\frac{1}{\delta}\right) + |V|\log\left(1+24k\right)\right)}\right\}}{\sqrt{n}}. \qquad \square$$

B.3 Exact Linear Oracle

Proof of Lemma 5.3.1. Since $z_1 = \cdots = z_k$ are less than the other elements of z, if it is feasible, it is optimal to place all the mass of p on the first k coordinates. In particular, the assignment $p_i = 1/k$ for $i = 1, \ldots, k$ accomplishes this while minimizing the χ^2 cost. The χ^2 cost incurred by this assignment can be computed as

$$\frac{1}{2}\sum_{i=1}^{k}\left(\frac{n}{k}-1\right)^2 + \frac{1}{2}\sum_{i=k+1}^{n}(0-1)^2 = \frac{1}{2}\left[k\cdot\left(\frac{n-k}{k}\right)^2 + (n-k)\right] \tag{B.1}$$

$$= \frac{1}{2}\cdot(n-k)\cdot\left[\frac{n-k}{k}+1\right] \tag{B.2}$$

$$= n(n-k)/(2k). \tag{B.3}$$

Hence if $\rho \geq n(n-k)/(2k)$, we can place all the mass of p on the first k coordinates; otherwise, the χ^2 constraint must be tight. $\square$

Proof of Lemma 5.3.2. At optimality, we must have $\mathbf{1}^T p^\star = 1$:

$$1 = \sum_{i=1}^{n} p_i^\star = \sum_{i=1}^{m} p_i^\star = \sum_{i=1}^{m} \left(1 - \frac{(z_i + \theta^\star)}{\lambda^\star n}\right) \cdot \frac{1}{n}. \tag{B.4}$$

Simplifying,

$$n = \sum_{i=1}^{m} \left(1 - \frac{(z_i + \theta^\star)}{\lambda^\star n}\right) = m - \frac{1}{\lambda^\star n} \sum_{i=1}^{m} (z_i + \theta^\star) \tag{B.5}$$

$$= m - \frac{m\overline{z}_m}{\lambda^\star n} - \frac{\theta^\star m}{\lambda^\star n}. \tag{B.6}$$

Multiplying through by $\lambda^\star n$ and solving for $\theta^\star$, we have

$$\lambda^\star n^2 = \lambda^\star mn - m\overline{z}_m - \theta^\star m \implies \theta^\star = \left(1 - \frac{n}{m}\right) \lambda^\star n - \overline{z}_m. \qquad \square$$

Proof of Lemma 5.3.3. By equation (5.16),

$$\langle z, p^\star \rangle = \frac{1}{n} \sum_{i=1}^{m} \left(1 - \frac{(z_i + \theta^\star)}{\lambda^\star n}\right) z_i \tag{B.7}$$

$$= \frac{1}{n} \sum_{i=1}^{m} z_i - \frac{1}{n} \sum_{i=1}^{m} \frac{(z_i + \theta^\star) z_i}{\lambda^\star n} \tag{B.8}$$

$$= \frac{m}{n} \overline{z}_m - \frac{1}{\lambda^\star n^2} \sum_{i=1}^{m} (z_i^2 + \theta^\star z_i) \tag{B.9}$$

$$= \frac{m}{n} \overline{z}_m - \frac{1}{\lambda^\star n^2} (b_m + \theta^\star m \overline{z}_m). \tag{B.10}$$

Plugging in the expression for $\theta^\star$ derived in Lemma 5.3.2, we compute

$$\theta^\star m \overline{z}_m = \left(\left(1 - \frac{n}{m}\right) \lambda^\star n - \overline{z}_m\right) m \overline{z}_m \tag{B.11}$$

$$= (m - n) \lambda^\star n \overline{z}_m - m(\overline{z}_m)^2, \tag{B.12}$$

so that we may further compute

$$-\frac{1}{\lambda^\star n^2}(b_m + \theta^\star m\overline{z}_m) = -\frac{b_m}{\lambda^\star n^2} - \frac{1}{\lambda^\star n^2}\left((m-n)\lambda^\star n\overline{z}_m - m(\overline{z}_m)^2\right) \tag{B.13}$$

$$= -\frac{(b_m - m(\overline{z}_m)^2)}{\lambda^\star n^2} + \frac{(n-m)}{n}\overline{z}_m \tag{B.14}$$

$$= -\frac{ms_m^2}{\lambda^\star n^2} + \frac{(n-m)}{n}\overline{z}_m. \tag{B.15}$$

When we finally plug this into equation (B.10), the $\frac{m}{n}\overline{z}_m$ and $\frac{n-m}{n}\overline{z}_m$ terms combine to form $\overline{z}_m$, leaving:

$$\langle z, p^\star\rangle = \overline{z}_m - \frac{ms_m^2}{\lambda^\star n^2}. \qquad \square$$

Proof of Lemma 5.3.4. First we check the χ^2 (quadratic) constraint; since the constraint is active, we have:

$$\rho \geq \frac{1}{2}\|np^\star - \mathbf{1}\|_2^2 \tag{B.16}$$

$$= \frac{1}{2}\sum_{i=1}^{n}(np_i^\star - 1)^2 \tag{B.17}$$

$$= \frac{1}{2}\sum_{i=1}^{m}\left(\left(1 - \frac{(z_i + \theta^\star)}{\lambda^\star n}\right) - 1\right)^2 + \frac{1}{2}\sum_{i=m+1}^{n}(-1)^2 \tag{B.18}$$

$$= \frac{1}{2}\cdot\frac{1}{(\lambda^\star)^2 n^2}\sum_{i=1}^{m}(z_i + \theta^\star)^2 + \frac{1}{2}(n-m) \tag{B.19}$$

where the third line is due to equation (5.16). We expand the sum of squares:

$$\sum_{i=1}^{m}(z_i + \theta^\star)^2 = \sum_{i=1}^{m}(z_i^2 + 2z_i\theta^\star + (\theta^\star)^2) \tag{B.20}$$

$$= \sum_{i=1}^{m} z_i^2 + 2\theta^\star\sum_{i=1}^{m} z_i + \sum_{i=1}^{m}(\theta^\star)^2 \tag{B.21}$$

$$= b_m + 2\theta^\star m\overline{z}_m + (\theta^\star)^2 m. \tag{B.22}$$

We expand the second and third terms, plugging in our expression for θ, and find

that

$$(\theta^\star)^2 m = \left[\left(1 - \frac{n}{m}\right)\lambda^\star n - \overline{z}_m\right]^2 \cdot m \tag{B.23}$$

$$= \left[\left(1 - \frac{n}{m}\right)^2 (\lambda^\star)^2 n^2 - 2\left(1 - \frac{n}{m}\right)\lambda^\star n \cdot \overline{z}_m + (\overline{z}_m)^2\right] \cdot m \tag{B.24}$$

$$= \left(1 - \frac{n}{m}\right)^2 (\lambda^\star)^2 n^2 m - 2\left(1 - \frac{n}{m}\right)\lambda^\star n m \overline{z}_m + m(\overline{z}_m)^2 \tag{B.25}$$

and also

$$2\theta^\star m \overline{z}_m = 2m\overline{z}_m \theta^\star \tag{B.26}$$

$$= 2m\overline{z}_m \left[\left(1 - \frac{n}{m}\right)\lambda^\star n - \overline{z}_m\right] \tag{B.27}$$

$$= 2\left(1 - \frac{n}{m}\right)\lambda^\star n m \overline{z}_m - 2m(\overline{z}_m)^2. \tag{B.28}$$

Note that the first term matches the second term of equation (B.25), so that they cancel when we add $2\theta^\star m\overline{z}_m$ and $(\theta^\star)^2 m$. Using this fact, we now expand equation (B.22):

$$b_m + 2\theta^\star m\overline{z}_m + (\theta^\star)^2 m = b_m - 2m(\overline{z}_m)^2 + \left(1 - \frac{n}{m}\right)^2 (\lambda^\star)^2 n^2 m + m(\overline{z}_m)^2 \tag{B.29}$$

$$= b_m - m(\overline{z}_m)^2 + \left(1 - \frac{n}{m}\right)^2 (\lambda^\star)^2 n^2 m \tag{B.30}$$

$$= m s_m^2 + \left(1 - \frac{n}{m}\right)^2 (\lambda^\star)^2 n^2 m. \tag{B.31}$$

Finally, plugging this back into equation (B.19) yields:

$$\rho \geq \frac{1}{2} \cdot \frac{1}{(\lambda^\star)^2 n^2} \cdot \left[m s_m^2 + \left(1 - \frac{n}{m}\right)^2 (\lambda^\star)^2 n^2 m \right] + \frac{1}{2} \cdot (n - m) \tag{B.32}$$

$$\Leftrightarrow 2\rho \geq \frac{m s_m^2}{(\lambda^\star)^2 n^2} + \left(1 - \frac{n}{m}\right)^2 m + (n - m) \tag{B.33}$$

$$\Leftrightarrow 2\rho \geq \frac{m s_m^2}{(\lambda^\star)^2 n^2} + \left(1 - \frac{2n}{m} + \frac{n^2}{m^2}\right) m + (n - m) \tag{B.34}$$

$$\Leftrightarrow 2\rho \geq \frac{m s_m^2}{(\lambda^\star)^2 n^2} + m - 2n + \frac{n^2}{m} + (n - m) \tag{B.35}$$

$$\Leftrightarrow 2\rho \geq \frac{m s_m^2}{(\lambda^\star)^2 n^2} - n + \frac{n^2}{m} \tag{B.36}$$

$$\Leftrightarrow \frac{2\rho m}{n^2} \geq \frac{m^2 s_m^2}{(\lambda^\star)^2 n^4} - \frac{m}{n} + 1 \tag{B.37}$$

$$\Leftrightarrow \frac{m^2 s_m^2}{(\lambda^\star)^2 n^4} \leq \alpha(m, n, \rho). \tag{B.38}$$

If $\alpha(m, n, \rho) \leq 0$, there is no feasible choice of $\lambda^\star$ for this m, so m cannot be correct. Otherwise, we can divide and solve for $\lambda^\star$:

$$\lambda^\star \geq \sqrt{\frac{m^2 s_m^2}{n^4 \alpha(m, n, \rho)}} = \frac{1}{n} \sqrt{\frac{m s_m^2}{2\rho + n - n^2/m}}, \tag{B.39}$$

or equivalently

$$\lambda^\star n^2 \geq \sqrt{\frac{m^2 s_m^2}{\alpha(m, n, \rho)}}. \tag{B.40}$$

Now we check the other remaining constraint on $\lambda^\star$, that the constraint $p_i \geq 0$ for $i = 1, \ldots, m$ must hold. In particular, we must have $p_m \geq 0$:

$$0 \leq p_m = \frac{1}{n} \cdot \left(1 - \frac{z_m + \theta^\star}{\lambda^\star n}\right) \tag{B.41}$$

$$\Leftrightarrow z_m + \theta^\star \leq \lambda^\star n \tag{B.42}$$

$$\Leftrightarrow z_m + \left(1 - \frac{n}{m}\right) \lambda^\star n - \overline{z}_m \leq \lambda^\star n \tag{B.43}$$

$$\Leftrightarrow z_m - \overline{z}_m \leq \frac{\lambda^\star n^2}{m} \tag{B.44}$$

$$\Leftrightarrow m(z_m - \overline{z}_m) \leq \lambda^\star n^2. \tag{B.45}$$

Hence $\lambda^\star$ must satisfy

$$\lambda^\star n^2 \geq \max\left\{\sqrt{\frac{m^2 s_m^2}{\alpha(m,n,\rho)}},\ m(z_m - \overline{z}_m)\right\}. \tag{B.46}$$

Since we seek minimal $\lambda^\star$, we select $\lambda^\star$ which makes this constraint tight. □

B.3.1 Unique solutions

Here we provide results for understanding when there is a unique solution to Problem (5.8). Recall that our solution to Problem (5.8) first checks whether the optimal solutions have tight χ^2 constraint. By choosing ρ small enough, this can be guaranteed uniformly:

Lemma B.3.1. *Suppose $\{z_i\}$ attain at least ℓ distinct values. If $\rho \leq (\ell-1)/2$ then all optimal solutions to Problem (5.8) have tight χ^2 constraint.*

Proof. Assume $z_1 \leq \cdots \leq z_n$. If $\{z_i\}$ attain at least ℓ distinct values, then the maximum number k so that $z_1 = \cdots = z_k$ can be bounded by $n - \ell + 1$. Recall from earlier in section B.3 that the constraint is tight if $\rho \leq n(n-k)/(2k)$, and note that this bound is monotone decreasing in k. Hence, we can guarantee the constraint is tight as long as

$$\rho \leq \frac{n(n-(n-\ell+1))}{2(n-\ell+1)} = \frac{n(\ell-1)}{2(n-\ell+1)}. \tag{B.47}$$

Since $n - \ell + 1 \leq n$, the previous inequality is implied by

$$\rho \leq \frac{(n-\ell+1)(\ell-1)}{2(n-\ell+1)} = \frac{\ell-1}{2}. \tag{B.48}$$

□

Now, assuming the χ^2 constraint is tight, we can characterize the set of optimal solutions:

Lemma B.3.2. *Suppose the optimal solutions for Problem (5.8) all have tight χ^2 constraint. Then there is a unique optimal solution p^* with minimum cardinality among all optimal solutions.*

Proof. This is a consequence of our characterization of the optimal dual variable λ as a function of the sparsity m. For each choice of m, we solved earlier for the unique dual variable λ_m which determines a unique solution p. Hence, even if there are multiple values of m that are feasible and that yield optimal objective value, there is still a unique minimal m_{opt}, which in turn yields a unique optimal solution. □

B.3.2 Lipschitz gradient

Lemma B.3.3. *Define $h(z) = \min_{p \in \mathcal{P}_{\rho,n}} \langle z, p \rangle$. Then on the subset of z's satisfying the high sample variance condition $s_n^2 \geq (2\rho B^2)/n^2$, $h(z)$ has Lipschitz gradient with constant $L \leq \frac{2\sqrt{2\rho}}{n^{3/2}} + \frac{2}{Bn^{1/2}}$.*

Proof. In this regime, there is a unique worst-case $p \in \mathcal{P}_{\rho,n}$, and it is the gradient of $h(z)$. In the high sample variance regime, we have $m = n$, i.e. each $p_i > 0$ and:

$$p_i = \left(1 - \frac{z_i + \theta}{\lambda n}\right) \cdot \frac{1}{n} \text{ for all } i = 1, \ldots, n. \tag{B.49}$$

In particular, $\theta = (1 - n/n)\lambda n - \overline{z}_n = -\overline{z}_n$, and $\lambda = \frac{1}{n^2}\sqrt{n^2 s_n^2/(2\rho/n)}$. Simplifying, we have

$$p_i = \left(1 - \frac{z_i - \overline{z}_n}{\lambda n}\right) \cdot \frac{1}{n} \tag{B.50}$$

$$= \left(1 - \frac{z_i - \overline{z}_n}{\frac{1}{n}\sqrt{n^2 s_n^2/(2\rho/n)}}\right) \cdot \frac{1}{n} \tag{B.51}$$

$$= \left(1 - \frac{z_i - \overline{z}_n}{\sqrt{n s_n^2/(2\rho)}}\right) \cdot \frac{1}{n}. \tag{B.52}$$

We will bound the Lipschitz constant of p as a function of z by computing the Hessian which has entries $H_{ij} = \frac{\partial p_i}{\partial z_j}$ and bounding its largest eigenvalue. For the element H_{ij}

we have two cases. If $i = j$, then

$$H_{ii} = -\frac{\sqrt{2\rho}}{n^{3/2}} \cdot \frac{\partial}{\partial z_i} \left(\frac{z_i - \overline{z}_n}{\sqrt{s_n^2}} \right) \tag{B.53}$$

$$= -\frac{\sqrt{2\rho}}{n^{3/2}} \cdot \left(\frac{\sqrt{s_n^2}(1 - \frac{1}{n}) - (z_i - \overline{z}_n) \cdot \frac{2}{n} \cdot (z_i - \overline{z}_n)}{s_n^2} \right). \tag{B.54}$$

If $i \neq j$, then

$$H_{ij} = -\frac{\sqrt{2\rho}}{n^{3/2}} \cdot \frac{\partial}{\partial z_j} \left(\frac{z_i - \overline{z}_n}{\sqrt{s_n^2}} \right) \tag{B.55}$$

$$= -\frac{\sqrt{2\rho}}{n^{3/2}} \cdot \left(\frac{-\frac{1}{n} \cdot \sqrt{s_n^2} - (z_i - \overline{z}_n) \cdot \frac{2}{n} \cdot (z_j - \overline{z}_n)}{s_n^2} \right). \tag{B.56}$$

Define $\tilde{H}$ so that $\frac{\sqrt{2\rho}}{n^{3/2} s_n^2} \tilde{H} = H$, i.e.

$$\tilde{H}_{ij} = \begin{cases} \sqrt{s_n^2}(\frac{1}{n} - 1) + (z_i - \overline{z}_n) \cdot \frac{2}{n} \cdot (z_i - \overline{z}_n) & i = j \\ \frac{1}{n} \cdot \sqrt{s_n^2} + (z_i - \overline{z}_n) \cdot \frac{2}{n} \cdot (z_j - \overline{z}_n) & i \neq j. \end{cases} \tag{B.57}$$

It is easy to see that $\tilde{H}$ is given by

$$\tilde{H} = -\text{diag}(\sqrt{s_n^2}\mathbf{1}) + \frac{\sqrt{s_n^2}}{n}\mathbf{1}\mathbf{1}^T + \frac{2}{n}(z - \overline{z}_n\mathbf{1})(z - \overline{z}_n\mathbf{1})^T. \tag{B.58}$$

By the triangle inequality, the operator norm of $\tilde{H}$ can thus be bounded by

$$\|\tilde{H}\| \leq \|\text{diag}(\sqrt{s_n^2}\mathbf{1})\| + \frac{\sqrt{s_n^2}}{n}\|\mathbf{1}\mathbf{1}^T\| + \frac{2}{n}\|(z - \overline{z}_n\mathbf{1})(z - \overline{z}_n\mathbf{1})^T\| \tag{B.59}$$

$$= \sqrt{s_n^2} + \frac{\sqrt{s_n^2}}{n}\|\mathbf{1}\|_2^2 + \frac{2}{n}\|z - \overline{z}_n\mathbf{1}\|_2^2 \tag{B.60}$$

$$= 2\sqrt{s_n^2} + \frac{2}{n}\sum_{i=1}^{n}(z_i - \overline{z}_n)^2 \tag{B.61}$$

$$= 2\sqrt{s_n^2} + 2s_n^2. \tag{B.62}$$

It follows that the Lipschitz constant of the gradient of $h(z)$ can be bounded by

$$\|H\| = \frac{\sqrt{2\rho}}{n^{3/2}s_n^2}\|\tilde{H}\| \tag{B.63}$$

$$\leq \frac{\sqrt{2\rho}}{n^{3/2}s_n^2}\left(2\sqrt{s_n^2} + 2s_n^2\right) \tag{B.64}$$

$$= \frac{2\sqrt{2\rho}}{n^{3/2}} \cdot \left(1 + \frac{1}{\sqrt{s_n^2}}\right). \tag{B.65}$$

Since we are in the high variance regime $s_n^2 \geq (2\rho B^2)/n$, it follows that $1/\sqrt{s_n^2} \leq \sqrt{n}/(B\sqrt{2\rho})$ and therefore

$$\|H\| \leq \frac{2\sqrt{2\rho}}{n^{3/2}} \cdot \left(1 + \frac{\sqrt{n}}{B\sqrt{2\rho}}\right) \tag{B.66}$$

$$= \frac{2\sqrt{2\rho}}{n^{3/2}} + \frac{2}{Bn}. \tag{B.67}$$

□

B.4 Convergence analysis for MFW

Here we establish the convergence rate of the MFW algorithm specifically for the DRO problem. For completeness, we reproduce the MFW convergence guarantee here:

Lemma B.4.1 (adapted from Mokhtari et al. (2018a)). *Let F be an up-concave function with L-Lipschitz gradient. MFW run for T iterations returns a point $x^{(T)}$ satisfying*

$$\mathbb{E}[F(x^{(T)})] \geq \left(1 - \frac{1}{e}\right) OPT - \frac{2DQ^{1/2}}{T^{1/3}} - \frac{LD^2}{2T}$$

where $D = \max_{x\in\mathcal{X}}\|x\|$, $Q = \max\{9^{2/3}\|\nabla F(x_0) - d_0\|^2, 16\sigma^2 + 3L^2D^2\}$, and σ is the variance of the stochastic gradients.

The main work is to establish Lipschitz continuity of ∇G, the gradient of the DRO objective. In fact, Mokhtari et al. (2018a) get a better bound by controlling

changes in ∇G specifically along the updates used by MFW. We bound this same quantity as follows:

Lemma B.4.2. *When the high sample variance condition is satisfied, for any two points $x^{(t)}$ and $x^{(t+1)}$ produced by MFW, ∇G satisfies $\|\nabla G(x^{(t)}) - \nabla G(x^{(t+1)})\| \leq \left(b\sqrt{n|V|}L + b\sqrt{k}\right)\|x^{(t)} - x^{(t+1)}\|$.*

Proof. We write $\vec{F}(x) = (F_1(x), ..., F_n(x))$, and are interested in the composition $G = h(\vec{F}(x))$ (recall that h is defined in Lemma 5.4.2 as the value of the inner minimization problem for a given set of values). Let $D\vec{F}(x)$ be the matrix derivative of $\vec{F}$. That is, $\left[D\vec{F}(x)\right]_{ij} = \frac{\partial}{\partial x_j}F_i(x)$. The chain rule yields

$$\nabla h(\vec{F}(x)) = \left(\nabla h(\vec{F}(x))\right) D\vec{F}(x).$$

Consider two points $x, y \in \mathcal{X}$. To apply the argument of Mokhtari et al. (2018a), we would like a bound on the change in ∇h along the MFW update from x in the direction of y. Let $x' = x + \frac{1}{T}y$ be the updated point. We have

$$\begin{aligned}
\|\nabla h(\vec{F}(x)) - \nabla h(\vec{F}(x'))\| &= \left\|\left(\nabla h(\vec{F}(x))\right) D\vec{F}(x) - \left(\nabla h(\vec{F}(x'))\right) D\vec{F}(x')\right\| \\
&= \left\|\left(\nabla h(\vec{F}(x))\right) D\vec{F}(x) - \left(\nabla h(\vec{F}(x))\right) D\vec{F}(x') \right. \\
&\qquad \left. + \left(\nabla h(\vec{F}(x))\right) D\vec{F}(x') - \left(\nabla h(\vec{F}(x'))\right) D\vec{F}(x')\right\| \\
&\leq \left\|\left(\nabla h(\vec{F}(x))\right) D\vec{F}(x) - \left(\nabla h(\vec{F}(x))\right) D\vec{F}(x')\right\| \\
&\qquad + \left\|\left(\nabla h(\vec{F}(x))\right) D\vec{F}(x') - \left(\nabla h(\vec{F}(x'))\right) D\vec{F}(x')\right\| \\
&= \left\|\left(\nabla h(\vec{F}(x))\right) \left(D\vec{F}(x) - D\vec{F}(x')\right)\right\| \\
&\qquad + \left\|\left(\nabla h(\vec{F}(x)) - \nabla h(\vec{F}(x'))\right) D\vec{F}(x')\right\|.
\end{aligned}$$

Starting out with the first term, we note that $\nabla h(\vec{F}(x))$ is a probability vector

(the optimal p for the DRO problem). Hence, we have

$$\left\| \left(\nabla h(\vec{F}(x)) \right) \left(D\vec{F}(x) - D\vec{F}(x') \right) \right\| \leq \max_{i=1,\ldots,n} \left\| D\vec{F}(x)_i - D\vec{F}(x')_i \right\|$$
$$= \max_{i=1,\ldots,n} \|\nabla F_i(x) - \nabla F_i(x')\|$$

And from Lemma 3 of Mokhtari et al. (2018a), we have that when x' is an updated point of the MFW algorithm starting at x,

$$\|\nabla F_i(x) - \nabla F_i(x')\| \leq b\sqrt{k}\|x - x'\| \quad \forall i = 1, \ldots, n.$$

We now turn to the second term. Note that the jth component of this vector is just the dot product

$$\left(\nabla h(\vec{F}(x)) - \nabla h(\vec{F}(x')) \right) \cdot D\vec{F}(x)_{\cdot,j}$$

where $D\vec{F}(x)_{\cdot,j}$ collects the partial derivative of each F_i with respect to x_j. Via the Cauchy-Schwartz inequality, we have

$$\left(\nabla h(\vec{F}(x)) - \nabla h(\vec{F}(x')) \right) \cdot D\vec{F}(x)_{\cdot,j} \leq \left\| \left(\nabla h(\vec{F}(x)) - \nabla h(\vec{F}(x')) \right) \right\| \left\| D\vec{F}(x)_{\cdot,j} \right\|$$

Lemma 5.4.2 shows that $\left\| \left(\nabla h(\vec{F}(x)) - \nabla h(\vec{F}(x')) \right) \right\| \leq L\|x - x'\|$. In order to bound the second norm, we claim that for all i, j, $\nabla_j F_i(x) \leq b$. To show this, note that we can use the definition of the multilinear extension to write

$$\nabla_j F_i(x) = \mathbb{E}_{S \sim x}[f_i(S|\{j\} \in S)] - \mathbb{E}_{S \sim x}[f_i(S|\{j\} \notin S)]$$

where $S \sim x$ denotes that S is drawn from the product distribution with marginals x. Now it is simple to show using submodularity of f_i that

$$\mathbb{E}_{S \sim x}[f_i(S|\{j\} \in S)] - \mathbb{E}_{S \sim x}[f_i(S|\{j\} \notin S)] \leq f_i(\{j\}) - f_i(\emptyset) \leq b.$$

Accordingly, we have that

$$\left\| D\vec{F}(x)_{\cdot,j} \right\| \leq b\|\mathbf{1}\| = b\sqrt{n}.$$

This gives us a component-wise bound on each element of the vector

$$\left(\nabla h(\vec{F}(x)) - \nabla h(\vec{F}(x'))\right) D\vec{F}(x').$$

Putting it all together, we have

$$\begin{aligned}\left\| \left(\nabla h(\vec{F}(x)) - \nabla h(\vec{F}(x'))\right) D\vec{F}(x') \right\| &\leq b\sqrt{n}L\|x - x'\| \cdot \|\mathbf{1}\| \\ &\leq b\sqrt{n|V|} \cdot L \cdot \|x - x'\|,\end{aligned}$$

and summing the two terms yields the final Lipschitz constant $b\sqrt{n|V|}L + b\sqrt{k}$. □

Now the final convergence rate for MFW stated in Theorem 5.4.1 follows from plugging the above Lipschitz bound into Lemma B.4.1. We also remark that the above argument trivially goes through for an arbitrary (not necessarily submodular) functions:

Lemma B.4.3. *Suppose that each function $f : \mathbb{R}^{|V|} \to \mathbb{R}$ in the support of P has bounded norm gradients $\max_{i=1,\ldots,|V|} |\nabla_i f| \leq b$ which are also L_f-Lipschitz. Then under the high variance condition, the corresponding DRO objective G has L_G-Lipschitz gradient with $L_G \leq L_f + b\sqrt{n|V|}L$, where L is as defined in Lemma 5.4.2.*

B.5 Rounding to a distribution over subsets

The output of MFW is a fractional vector $x \in \mathcal{X}$. Lemma 5.4.1 guarantees this x can be converted into a distribution $\mathcal{D}$ over feasible subsets, and moreover, that the attainable solution value from doing so is within a $(1 - 1/e)$ factor of the optimal value for the DRO problem. This result is essentially standard (see Wilder (2018a) for a more detailed presentation), but we sketch the process here for completeness.

There are two steps. First, we argue that x can be converted into a distribution over subsets with equivalent value for the DRO problem. Second, we argue that the *optimal* x (product distribution) has value within $(1 - 1/e)$ of the optimal arbitrary distribution over subsets.

For the first step, our starting point is the swap rounding algorithm of Chekuri et al. (2010). Swap rounding is a randomized rounding algorithm which takes a vector x and returns a feasible subset S. For any single submodular function and its multilinear extension F, swap rounding guarantees $\mathbb{E}[f(S)] \geq F(x)$. In our setting, such guarantees cannot be obtained for a single S since we want to simultaneously match the value of x with respect to n submodular functions $f_1, \ldots, f_n$. However, swap rounding obeys a desirable concentration property which allows us to form a distribution $\mathcal{D}$ by running swap rounding independently several times and returning the empirical distribution over the outputs. Provided that we take sufficiently many samples, $\mathcal{D}$ is guaranteed to satisfy $\mathbb{E}_{S\sim\mathcal{D}}[f_i(S)] \geq F_i(x) - \varepsilon$ for all $i = 1, \ldots, n$ with high probability. Specifically, Wilder (2018a) show that it suffices to draw $O\left(\frac{\log \frac{n}{\delta}}{\varepsilon^3}\right)$ sets via swap rounding in order for this guarantee to hold with probability $1 - \delta$.

The other piece of Lemma 5.4.1 relates the optimal value for Problem (5.20) (optimizing over product distributions) to the optimal value for the complete DRO problem (optimizing over arbitrary distributions). These values are easily shown to be within $(1 - 1/e)$ of each other by applying the correlation gap result of Agrawal et al. (2010). For any product distribution p over subsets, let $\text{marg}(p)$ denote the set of (potentially correlated) distributions with the same marginals as p. This result shows that for any submodular function f,

$$\max_{p\text{: a product distribution}} \max_{q\in\text{marg}(p)} \frac{\mathbb{E}_{S\sim q}[f(S)]}{\mathbb{E}_{S\sim p}[f(S)]} \leq \frac{e}{e-1}$$

and now Lemma 5.4.1 follows by applying the correlation gap bound to each of the f_i.

Appendix C

Robust Budget Allocation

C.1 Worst-Case Approximation Ratio versus True Worst-Case

Consider the function $f(x;\theta)$ defined on $\{0,1\} \times \{0,1\}$, with values given by:

$$f(x;0) = \begin{cases} 1 & x = 0 \\ 0.6 & x = 1, \end{cases} \qquad f(x;1) = \begin{cases} 1 & x = 0 \\ 2 & x = 1. \end{cases} \tag{C.1}$$

We wish to choose x to maximize $f(x;\theta)$ robustly with respect to adversarial choices of θ. If θ were fixed, we could directly choose x^*_θ to maximize $f(x;\theta)$. In particular, $x^*_0 = 0$ and $x^*_1 = 1$. Of course, we want to deal with worst-case θ. One option is to maximize the worst-case approximation ratio:

$$\max_x \min_\theta \frac{f(x;\theta)}{f(x^*_\theta;\theta)}. \tag{C.2}$$

One can verify that the best x according to this criterion is $x = 1$, with worst-case approximation ratio 0.6 and worst-case function value 0.6. In this chapter, we optimize the worst-case of the actual function value:

$$\max_x \min_\theta f(x;\theta). \tag{C.3}$$

This criterion will select $x = 0$, which has a worse worst-case approximation ratio of 0.5, but actually guarantees a function value of 1, significantly better than the 0.6 achieved by the other formulation of robustness.

C.2 DR-submodularity and $L^\natural$-convexity

A function is $L^\natural$-convex if it satisfies a discrete version of midpoint convexity, i.e. for all x, y it holds that

$$f(x) + f(y) \geq f\left(\left\lceil \frac{x+y}{2} \right\rceil\right) + f\left(\left\lfloor \frac{x+y}{2} \right\rfloor\right), \tag{C.4}$$

where the floor $\lfloor \cdot \rfloor$ and ceiling $\lceil \cdot \rceil$ functions are interpreted elementwise.

Remark C.2.1. An $L^\natural$-convex function need not be DR-submodular, and vice-versa. Hence algorithms for optimizing one type may not apply for the other.

Proof. Consider $f_1(x_1, x_2) = -x_1^2 - 2x_1x_2$ and $f_2(x_1, x_2) = x_1^2 + x_2^2$, both defined on $\{0, 1, 2\} \times \{0, 1, 2\}$. The function f_1 is DR-submodular but violates discrete midpoint convexity for the pair of points $(0, 0)$ and $(2, 2)$, while f_2 is $L^\natural$-convex but does not have diminishing returns in either dimension. □

Intuitively-speaking, $L^\natural$-convex functions look like discretizations of convex functions. The continuous objective function $\mathcal{I}(x, y)$ we consider need not be convex, hence its discretization need not be $L^\natural$-convex, and we cannot use those tools. However, in some regimes (namely if each $y(s) \in \{0\} \cup [1, \infty)$), it happens that $\mathcal{I}(x, y)$ is DR-submodular in x.

C.3 Constrained Continuous Submodular Function Minimization

C.3.1 Solving the Optimization Problem

Here, we describe how to solve the convex problem (6.15) to which we reduced the original constrained submodular minimization problem. Bach (2019), at the beginning of Section 5.2, states that this surrogate problem can be optimized via the Frank-Wolfe method and its variants. However, Bach (2019) only elaborates on the simpler version of Problem (6.15) without the extra functions a_{ix_i}. Here we detail how Frank-Wolfe algorithms can be used to solve the more general parametric regularized problem. Our aim is to spell out very clearly the applicability of Frank-Wolfe to this problem, for the ease of practitioners.

Bach (2019) notes that, by duality, Problem (6.15) is equivalent to:

$$\begin{aligned}
&\min_{\rho \in \prod_{i=1}^n \mathbb{R}_{\downarrow}^{k_i-1}} h_{\downarrow}(\rho) - H(0) + \sum_{i=1}^{n} \sum_{x_i=1}^{k_i-1} a_{ix_i}[\rho_i(x_i)] \\
= &\min_{\rho \in \prod_{i=1}^n \mathbb{R}_{\downarrow}^{k_i-1}} \max_{w \in B(H)} \langle \rho, w \rangle + \sum_{i=1}^{n} \sum_{x_i=1}^{k_i-1} a_{ix_i}[\rho_i(x_i)] \\
= &\max_{w \in B(H)} \left\{ \min_{\rho \in \prod_{i=1}^n \mathbb{R}_{\downarrow}^{k_i-1}} \langle \rho, w \rangle + \sum_{i=1}^{n} \sum_{x_i=1}^{k_i-1} a_{ix_i}[\rho_i(x_i)] \right\} \\
:= &\max_{w \in B(H)} f(w).
\end{aligned}$$

Here, the base polytope $B(H)$ happens to be the convex hull of all vectors w which could be output by the greedy algorithm in (Bach, 2019).

It is the dual problem, where we maximize over w, which is amenable to Frank-Wolfe. For Frank-Wolfe methods, we need two oracles: an oracle which, given w, returns $\nabla f(w)$; and an oracle which, given $\nabla f(w)$, produces a point s which solves the linear optimization problem $\max_{s \in B(H)} \langle s, \nabla f(w) \rangle$.

Per Bach (2019), an optimizer of the linear problem can be computed directly from the greedy algorithm. For the gradient oracle, recall that we can find a subgradient

of $g(x) = \min_y h(x, y)$ at the point x_0 by finding $y(x_0)$ which is optimal for the inner problem, and then computing $\nabla_x h(x, y(x_0))$. Moreover, if such $y(x_0)$ is the unique optimizer, then the resulting vector is indeed the *gradient* of $g(x)$ at x_0. Hence, in our case, it suffices to first find $\rho(w)$ which solves the inner problem, and then $\nabla f(w)$ is simply $\rho(w)$ because the inner function is linear in w. Since each function a_{ix_i} is strictly convex, the minimizer $\rho(w)$ is unique, confirming that we indeed get a gradient of f, and that f is differentiable.

Of course, we still need to compute the minimizer $\rho(w)$. For a given w, we want to solve

$$\min_{\rho \in \prod_{i=1}^n \mathbb{R}_{\downarrow}^{k_i - 1}} \langle \rho, w \rangle + \sum_{i=1}^{n} \sum_{x_i = 1}^{k_i - 1} a_{ix_i}[\rho_i(x_i)]$$

There are no constraints coupling the vectors ρ_i, and the objective is similarly separable, so we can independently solve n problems of the form

$$\min_{\rho \in \mathbb{R}_{\downarrow}^{k-1}} \langle \rho, w \rangle + \sum_{j=1}^{k-1} a_j(\rho_j).$$

Recall that each function $a_{iy_i}(t)$ takes the form $\frac{1}{2} t^2 r_{iy_i}$ for some $r_{iy_i} > 0$. Let $D = \operatorname{diag}(r)$, the $(k-1) \times (k-1)$ matrix with diagonal entries r_j. Our problem can then be written as

$$\begin{aligned} \min_{\rho \in \mathbb{R}_{\downarrow}^{k-1}} \langle \rho, w \rangle + \frac{1}{2} \sum_{j=1}^{k-1} r_j \rho_j^2 &= \min_{\rho \in \mathbb{R}_{\downarrow}^{k-1}} \langle \rho, w \rangle + \frac{1}{2} \langle D\rho,\ \rho \rangle \\ &= \min_{\rho \in \mathbb{R}_{\downarrow}^{k-1}} \langle D^{1/2}\rho,\ D^{-1/2} w \rangle + \frac{1}{2} \langle D^{1/2}\rho,\ D^{1/2}\rho \rangle. \end{aligned}$$

Completing the square, the above problem is equivalent to

$$\begin{aligned} \min_{\rho \in \mathbb{R}_{\downarrow}^{k-1}} \| D^{1/2}\rho + D^{-1/2} w \|_2^2 &= \min_{\rho \in \mathbb{R}_{\downarrow}^{k-1}} \sum_{j=1}^{k-1} (r_j^{1/2} \rho_j + r_j^{-1/2} w_j)^2 \\ &= \min_{\rho \in \mathbb{R}_{\downarrow}^{k-1}} \sum_{j=1}^{k-1} r_j (\rho_j + r_j^{-1} w_j)^2. \end{aligned}$$

This last expression is precisely the problem which is called weighted isotonic regression: we are fitting ρ to $\text{diag}(r^{-1})w$, with weights r, subject to a monotonicity constraint. Weighted isotonic regression is solved efficiently via the Pool Adjacent Violators algorithm of Best and Chakravarti (1990).

C.3.2 Runtime

Frank-Wolfe returns an τ-suboptimal solution in $O(\tau^{-1}D^2L)$ iterations, where D is the diameter of the feasible region, and L is the Lipschitz constant for the gradient of the objective (Jaggi, 2013). Our optimization problem is $\max_{w \in B(H)} f(w)$ as defined in the previous section. Each $w \in B(H)$ has $O(n\delta^{-1})$ coordinates of the form $H^\delta(x + e_i) - H^\delta(x)$. Since H^δ is an expected influence in the range $[0, T]$, we can bound the magnitude of each coordinate of w by T and hence D^2 by $O(n\delta^{-1}T^2)$. If α is the minimum derivative of the functions R_i, then the smallest coefficient of the functions $a_{ix_i}(t)$ is bounded below by $\alpha\delta$. Hence the objective is the conjugate of an $\alpha\delta$-strongly convex function, and therefore has $\alpha^{-1}\delta^{-1}$-Lipschitz gradient. Combining these, we arrive at the $O(\tau^{-1}n\delta^{-2}\alpha^{-1}T^2)$ iteration bound. The most expensive step in each iteration is computing the subgradient, which requires sorting the $O(n\delta^{-1})$ elements of ρ in time $O(n\delta^{-1}\log n\delta^{-1})$. Hence the total runtime of Frank-Wolfe is $O(\tau^{-1}n^2\delta^{-3}\alpha^{-1}T^2\log n\delta^{-1})$.

As specified in the main text, relating an approximate solution of (6.15) to a solution of (6.12) is nontrivial. Assume ρ^* has distinct elements separated by η, and chose τ to be less than $\eta^2\alpha\delta/8$. If ρ is τ-suboptimal, then by $\alpha\delta$-strong convexity we must have $\|\rho - \rho^*\|_2 < \eta/2$, and therefore $\|\rho - \rho^*\|_\infty < \eta/2$. Since the smallest consecutive gap between elements of ρ^* is η, this implies that ρ and ρ^* have the same ordering, and therefore admit the same solution x after thresholding. Accounting for this choice in τ, we have an exact solution to (6.12) in total runtime of $O(\eta^{-2}n^2\delta^{-4}\alpha^{-2}T^2\log n\delta^{-1})$.

Appendix D

Escaping saddle points with Adaptive Gradient Methods and perturbations

D.1 More Insights from Idealized Adaptive Methods (IAM)

Suppose for now that we have oracle access to $G_t = \mathbb{E}[g_t g_t^T]$. Why should preconditioning by $A = \mathbb{E}[gg^T]^{-1/2}$ help optimization? The original Adam paper (Kingma and Ba, 2015) argues that Adam is an approximation to natural gradient descent, since if the objective f is a log-likelihood, $\mathbb{E}[gg^T]$ approximates the Fisher information matrix F, which captures curvature information in the space of distributions. This connection is tenuous at best, since the approximation $F \approx \mathbb{E}[gg^T]$ is only valid near optimality. Moreover, the exponent is wrong: Adam preconditions by $\mathbb{E}[gg^T]^{-1/2}$, but natural gradient should precondition by $\mathbb{E}[gg^T]^{-1}$. But using the exponent -1 is reported in the literature as unstable, even for Adagrad: "*without the square root operation, the algorithm performs much worse*" (Ruder, 2016). So the exponent is changed to $-1/2$ instead of -1.

Both of the above issues with the natural gradient interpretation are also pointed

out by Balles and Hennig (2018), who argue that the primary function of adaptive methods is to equalize the stochastic gradient noise in each direction. But it is still *not* clear precisely why or how equalized noise should help optimization.

By assuming oracle access to $\mathbb{E}[gg^T]$, we can immediately argue that the exponent cannot be more aggressive than $-1/2$. Suppose we run preconditioned SGD with the preconditioner G_t^{-1} (instead of $G_t^{-1/2}$ as in RMSProp), and apply this to a noiseless problem; that is, g_t always equals the full gradient $\nabla_t = \nabla f(x_t)$. The preconditioner is then

$$A_t = (\mathbb{E}[g_t g_t^T] + \varepsilon I)^{-1} = (\nabla_t \nabla_t^T + \varepsilon I)^{-1}. \tag{D.1}$$

Taking $\varepsilon \to 0$, the idealized RMSProp update approaches

$$w_{t+1} \leftarrow w_t - \eta \frac{\nabla_t}{\|\nabla_t\|^2}. \tag{D.2}$$

First, the actual descent direction is not changed, and curvature is totally absent. Second, the resulting algorithm is unstable unless η decreases rapidly: as x_t approaches a stationary point, the magnitude of the step $\nabla_t / \|\nabla_t\|^2$ grows arbitrarily large, making it impossible to converge without rapidly decreasing the stepsize.

By contrast, using the standard $-1/2$ exponent and taking $\varepsilon \to 0$ in the noiseless case yields normalized gradient descent:

$$w_{t+1} \leftarrow w_t - \eta \frac{\nabla_t}{\|\nabla_t\|}. \tag{D.3}$$

In neither case do adaptive methods actually change the direction of descent (e.g. via curvature information); only the stepsize is changed.

D.2 Algorithm Details

Per our estimation results in Section 7.4.1, we must alter RMSProp to ensure it achieves an accurate estimate of the preconditioner. Namely, before updating the parameter w_t, we need to burn-in the estimate for several iterations so the initial

Algorithm 8 BurnIn

function BURNIN(burn-in length W, β)
  for $t = 0, \ldots, W$ **do**
    $g_t \leftarrow$ stochastic gradient
    $\hat{G}_t \leftarrow \beta \hat{G}_{t-1} + (1-\beta) g_t g_t^T$
  end for
  return $\hat{G}_t$
end function

Algorithm 9 Hallucinate

function HALLUCINATE(hallucination length S, β, $\hat{G}$, w_{start}, w_{end})
  for $s = 0, \ldots, S$ **do**
    $g_s \leftarrow$ stochastic gradient at $w_{\text{start}} + \frac{s}{S}(w_{\text{end}} - w_{\text{start}})$
    $\hat{G} \leftarrow \beta \hat{G} + (1-\beta) g_s g_s^T$
  end for
  return $\hat{G}$
end function

estimate $\hat{G}_0$ is accurate. This subroutine is given in Algorithm 8.

Later, when we prove second-order convergence, we need to modify RMSProp to occassionally take a large step. However, this complicates estimation: per Theorem 7.4.1, estimation quality deteriorates as the step size increases. Naively applying Theorem 7.4.1 to the large stepsize yields an estimate of G that is not accurate enough. To get around this, every time RMSProp takes a large step, we will hallucinate a number of smaller steps to feed into the estimation procedure. This is formalized in Algorithm 9. Overall, the variant of RMSProp we study is formalized in Algorithm 10.

D.3 Curvature and noise constants for different preconditioners

Our analysis for general preconditioners depends on constants $\Lambda_1, \Lambda_2, \Gamma, \nu$, as well as $\lambda_- = \lambda_{\min}(A)$ that measure various properties of the preconditioner A. For convenience, we reproduce the definition:

Definition D.3.1. We say $A(w)$ is a $(\Lambda_1, \Lambda_2, \Gamma, \nu, \lambda_-)$-*preconditioner* if, for all x in the

Algorithm 10 Full-matrix RMSProp with increasing stepsize

Input: initial w_0, time T, stepsizes η, r, threshold t_{thresh}, time S, burn-in length W, momentum β

$\hat{G}_0 \leftarrow \textsc{BurnIn}(W, \beta)$ ▷ Algorithm 8

for $t = 0, \dots, T$ **do**

  $g_t \leftarrow$ stochastic gradient at w_t

  $\hat{G}_t \leftarrow \beta \hat{G}_{t-1} + (1-\beta) g_t g_t^T$

  $A_t \leftarrow \hat{G}_t^{-1/2}$

  if $t \bmod t_{\text{thresh}} = 0$ **then**

    $w_{t+1} \leftarrow w_t - r A_t g_t$

    $\hat{G}_t \leftarrow \textsc{Hallucinate}(S, \beta, \hat{G}_t, w_t, w_{t+1})$ ▷ Algorithm 9

  else

    $w_{t+1} \leftarrow w_t - \eta A_t g_t$

  end if

end for

domain, the following bounds hold. First, $\|A\nabla f\|^2 \le \Lambda_1 \|A^{1/2}\nabla f\|^2$. Second, if $\tilde{f}(w)$ is the quadratic approximation of f at some point w_0, we assume $\|A(\nabla f - \nabla \tilde{f})\| \le \Lambda_2 \|\nabla f - \nabla \tilde{f}\|$. Third, $\Gamma \ge \mathbb{E}[\|Ag\|^2]$. Fourth, $\nu \le \lambda_{\min}(A\,\mathbb{E}[gg^T]A^T)$. Finally, $\lambda_- \le \lambda_{\min}(A)$.

As before, we write $G = \mathbb{E}[gg^T]$ throughout.

D.3.1 Constants for identity preconditioner

In the simplest case, $A = I$ and we merely run SGD. We reproduce Proposition 7.4.3:

Proposition D.3.1. *The preconditioner $A = I$ is a $(\Lambda_1, \Lambda_2, \Gamma, \nu, \lambda_-)$-preconditioner, with $\Lambda_1 = \Lambda_2 = 1$, $\Gamma = \mathbb{E}[\|g\|^2]$, $\nu \le \lambda_{\min}(G)$, and $\lambda_- = 1$.*

The overall second-order complexity depends on

$$\frac{\Lambda_1 \Lambda_2 \Gamma}{\nu} = \frac{\mathbb{E}[\|g\|^2]}{\lambda_{\min}(G)}, \tag{D.4}$$

as well as $\lambda_- = \lambda_{\min}(A) = 1$.

Proof of Proposition D.3.1. Clearly, $\Lambda_1 = \Lambda_2 = \lambda_- = 1$. Then,

$$\mathbb{E}[\|Ag\|^2] = \mathbb{E}[\|g\|^2] =: \Gamma. \tag{D.5}$$

Finally,

$$\lambda_{\min}(AGA^T) = \lambda_{\min}(G) =: \nu. \tag{D.6}$$

□

D.3.2 Constants for full matrix IAM

Write $G = \mathbb{E}[gg^T]$, and define the preconditioner A by $A = (G+\varepsilon I)^{-1/2}$. We reproduce Proposition 7.4.4:

Proposition D.3.2. *The preconditioner $A = (G + \varepsilon I)^{-1/2}$ is a $(\Lambda_1, \Lambda_2, \Gamma, \nu, \lambda_-)$-preconditioner, with*

$$\Lambda_1 = \Lambda_2 = (\lambda_{\min}(G) + \varepsilon)^{-1/2},\ \Gamma = \frac{d\lambda_{\max}(G)}{\varepsilon + \lambda_{\max}(G)},\ \nu = \frac{\lambda_{\min}(G)}{\lambda_{\min}(G) + \varepsilon}, \tag{D.7}$$

and $\lambda_- = (\lambda_{\max}(G) + \varepsilon)^{-1/2}$.

Overall, the complexity depends on $\Lambda_1\Lambda_2\Gamma/\nu$:

$$\frac{\Lambda_1\Lambda_2\Gamma}{\nu} = \frac{1}{\sqrt{\lambda_{\min}(G) + \varepsilon}} \cdot \frac{1}{\sqrt{\lambda_{\min}(G) + \varepsilon}} \cdot \frac{d\lambda_{\max}(G)}{\varepsilon + \lambda_{\max}(G)} \cdot \frac{\lambda_{\min}(G) + \varepsilon}{\lambda_{\min}(G)} \tag{D.8}$$

$$= \frac{d\lambda_{\max}(G)}{(\varepsilon + \lambda_{\max}(G))\lambda_{\min}(G)}. \tag{D.9}$$

Therefore

$$\frac{\Lambda_1^4\Lambda_2^4\Gamma^4}{\lambda_-^{10}\nu^4} \le \left(\frac{d\lambda_{\max}(G)}{(\varepsilon + \lambda_{\max}(G))\lambda_{\min}(G)}\right)^4 (\lambda_{\max}(G) + \varepsilon)^5 \tag{D.10}$$

$$= d^4\kappa(G)^4(\lambda_{\max}(G) + \varepsilon) \tag{D.11}$$

Note that when $\varepsilon = 0$ and we do not regularize the preconditioner, the complexity bound is

$$\frac{\Lambda_1^4\Lambda_2^4\Gamma^4}{\lambda_-^{10}\nu^4} = d^4\kappa(G)^4\lambda_{\max}(G). \tag{D.12}$$

If we make the optimistic but often reasonable assumptions that $\Lambda_1 = O(1)$ (if A is aligned well with the Hessian) and $\Lambda_2 = O(1)$ (the function f is essentially quadratic at saddle points) then all dependence on $\lambda_{\min}(G)$ vanishes, and the bound is

$$\frac{\Gamma^4}{\lambda_-^{10}\nu^4} = d^4\lambda_{\max}(G)^5. \tag{D.13}$$

Proof of Proposition 7.4.4. We can bound both Λ_1 and Λ_2 by

$$\Lambda_1, \Lambda_2 \leq \lambda_{\max}(A) = \lambda_{\min}(G + \varepsilon I)^{-1/2} = (\lambda_{\min}(G) + \varepsilon)^{-1/2}. \tag{D.14}$$

For Γ, we need to bound $\mathbb{E}[\|Ag\|^2] = \operatorname{tr}(A^2G)$. Expanding, we may write

$$A^2G = (G + \varepsilon I)^{-1}G. \tag{D.15}$$

The mapping $t \mapsto t/(t + \varepsilon)$ is increasing, so by using the bound $\lambda_{\max}(G)I \succeq G$, we may bound

$$A^2G \preceq \frac{\lambda_{\max}(G)}{\varepsilon + \lambda_{\max}(G)} I. \tag{D.16}$$

It follows that we can bound the trace of A^2G by

$$\Gamma = d \cdot \frac{\lambda_{\max}(G)}{\varepsilon + \lambda_{\max}(G)}. \tag{D.17}$$

Next, ν is a bound on the least eigenvalue of

$$AGA^T = (G + \varepsilon I)^{-1/2}G(G + \varepsilon I)^{-1/2} = (G + \varepsilon I)^{-1}G. \tag{D.18}$$

Since $t \mapsto t/(t + \varepsilon)$ is increasing, it is minimized when t is small. Therefore

$$\lambda_{\min}(AGA^T) \geq \frac{\lambda_{\min}(G)}{\lambda_{\min}(G) + \varepsilon} =: \nu. \tag{D.19}$$

□

D.3.3 Constants for diagonal IAM

Define the preconditioner A by $A = \operatorname{diag}(\mathbb{E}[g^2] + \varepsilon)^{-1/2}$.

Proposition D.3.3. *The preconditioner $A = \operatorname{diag}(\mathbb{E}[g^2]+\varepsilon)^{-1/2}$ is a $(\Lambda_1, \Lambda_2, \Gamma, \nu, \lambda_-)$-preconditioner, with*

$$\Lambda_1 = \Lambda_2 = \left(\varepsilon + \min_{i\in[d]} \mathbb{E}[g_i^2]\right)^{-1/2}, \tag{D.20}$$

$$\Gamma = \frac{d \max_{i\in[d]} \mathbb{E}[g_i^2]}{\varepsilon + \max_{i\in[d]} \mathbb{E}[g_i^2]}, \tag{D.21}$$

$$\nu = \frac{\lambda_{\min}(G \operatorname{diag}(G)^{-1}) \cdot \min_{i\in[d]} \mathbb{E}[g_i^2]}{\varepsilon + \min_{i\in[d]} \mathbb{E}[g_i^2]}, \text{ and} \tag{D.22}$$

$$\lambda_- = (\varepsilon + \max_{i\in[d]} \mathbb{E}[g_i]^2)^{-1/2}. \tag{D.23}$$

Overall,

$$\frac{\Lambda_1\Lambda_2\Gamma}{\nu} = \frac{\varepsilon + \min_{i\in[d]} \mathbb{E}[g_i^2]}{(\varepsilon + \min_{i\in[d]} \mathbb{E}[g_i^2]) \cdot \lambda_{\min}(G \operatorname{diag}(G)^{-1}) \min_{i\in[d]} \mathbb{E}[g_i^2]} \cdot \frac{d \cdot \max_{i\in[d]} \mathbb{E}[g_i^2]}{\varepsilon + \max_{i\in[d]} \mathbb{E}[g_i^2]} \tag{D.24}$$

$$= \frac{1}{\lambda_{\min}(G \operatorname{diag}(G)^{-1}) \min_{i\in[d]} \mathbb{E}[g_i^2]} \cdot \frac{d \cdot \max_{i\in[d]} \mathbb{E}[g_i^2]}{\varepsilon + \max_{i\in[d]} \mathbb{E}[g_i^2]} \tag{D.25}$$

$$\tag{D.26}$$

so the overall second-order dependence is

$$\frac{\Lambda_1^4\Lambda_2^4\Gamma^4}{\lambda_-^{10}\nu^4} = \frac{(\varepsilon + \max_{i\in[d]} \mathbb{E}[g_i^2])^5}{\lambda_{\min}(G \operatorname{diag}(G)^{-1})^4 (\min_{i\in[d]} \mathbb{E}[g_i^2])^4} \cdot \frac{d^4 \cdot (\max_{i\in[d]} \mathbb{E}[g_i^2])^4}{(\varepsilon + \max_{i\in[d])} \mathbb{E}[g_i^2])^4} \tag{D.27}$$

$$= \frac{(\varepsilon + \max_{i\in[d]} \mathbb{E}[g_i^2]) \cdot d^4 \cdot (\max_{i\in[d]} \mathbb{E}[g_i^2])^4}{\lambda_{\min}(G \operatorname{diag}(G)^{-1})^4 (\min_{i\in[d]} \mathbb{E}[g_i^2])^4}. \tag{D.28}$$

If we set $\varepsilon = 0$ and do not regularize the preconditioner, the complexity bound is

$$\frac{\Lambda_1^4\Lambda_2^4\Gamma^4}{\lambda_-^{10}\nu^4} = \frac{d^4 \cdot (\max_{i\in[d]} \mathbb{E}[g_i^2])^5}{\lambda_{\min}(G \operatorname{diag}(G)^{-1})^4 (\min_{i\in[d]} \mathbb{E}[g_i^2])^4}. \tag{D.29}$$

Proof of Proposition D.3.3. As before, we can bound both Λ_1 and Λ_2 by

$$\Lambda_1, \Lambda_2 \leq \lambda_{\max}(A) = \left(\varepsilon + \min_{i \in [d]} \mathbb{E}[g_i^2]\right)^{-1/2}. \tag{D.30}$$

For Γ, using the same manipulations as before, we want to bound

$$\begin{aligned}
\mathbb{E}[\|Ag\|^2] &= \operatorname{tr}(\operatorname{diag}(\mathbb{E}[g^2]) \operatorname{diag}(\varepsilon + \mathbb{E}[g^2])^{-1}) && \text{(D.31)}\\
&= \operatorname{tr}\left(\operatorname{diag}\left(\frac{\mathbb{E}[g^2]}{\varepsilon + \mathbb{E}[g^2]}\right)\right) && \text{(D.32)}\\
&\leq d \cdot \frac{\max_{i \in [d]} \mathbb{E}[g_i^2]}{\varepsilon + \max_{i \in [d]} \mathbb{E}[g_i^2]}. && \text{(D.33)}
\end{aligned}$$

Again, bounding ν is difficult, as we would need to bound the least eigenvalue of

$$\begin{aligned}
A\,\mathbb{E}[gg^T]A &= \mathbb{E}[gg^T] \operatorname{diag}(\varepsilon + \mathbb{E}[g^2])^{-1} && \text{(D.34)}\\
&= G(\varepsilon + \operatorname{diag}(G))^{-1} && \text{(D.35)}\\
&= G(\operatorname{diag}(G)^{-1} - \operatorname{diag}(G)^{-1}(\varepsilon^{-1} I + \operatorname{diag}(G)^{-1})^{-1} \operatorname{diag}(G)^{-1}) && \text{(D.36)}\\
&= G \operatorname{diag}(G)^{-1}(I - (\varepsilon^{-1} I + \operatorname{diag}(G)^{-1})^{-1} \operatorname{diag}(G)^{-1}). && \text{(D.37)}
\end{aligned}$$

The first two terms are ν if we had not added ε to A. The remaining terms can be bounded as before by

$$I - (\varepsilon^{-1} I + \operatorname{diag}(G)^{-1})^{-1} \operatorname{diag}(G)^{-1} \succeq \frac{\min_{i \in [d]} \mathbb{E}[g_i^2]}{\varepsilon + \min_{i \in [d]} \mathbb{E}[g_i^2]} \cdot I \tag{D.38}$$

so that overall we can take

$$\nu = \lambda_{\min}(G \operatorname{diag}(G)^{-1}) \cdot \frac{\min_{i \in [d]} \mathbb{E}[g_i^2]}{\varepsilon + \min_{i \in [d]} \mathbb{E}[g_i^2]} \leq \lambda_{\min}(G(\varepsilon + \operatorname{diag}(G))^{-1}). \tag{D.39}$$

Finally,

$$\lambda_- = \lambda_{\min}(A) = \frac{1}{(\max_{i \in [d]} \mathbb{E}[g_i^2] + \varepsilon)^{1/2}}. \tag{D.40}$$

□

Algorithm 11 Diagonal RMSProp with burn-in

Input: initial w_0, time T, stepsize η, burn-in length W
$\hat{v}_0 \leftarrow \text{diag}(\textsc{BurnIn}(W, \beta))$ ▷ Appendix D.2
for $t = 0, \ldots, T$ **do**
  $g_t \leftarrow$ stochastic gradient
  $\hat{v}_t \leftarrow \beta \hat{v}_{t-1} + (1-\beta) g_t^2$
  $\hat{A}_t \leftarrow \text{diag}(\hat{v}_t)^{-1/2}$
  $w_{t+1} \leftarrow w_t - \eta \hat{A}_t g_t$
end for

Algorithm 12 Diagonal RMSProp with increasing stepsize

Input: initial w_0, time T, stepsizes η, r, threshold t_{thresh}, time S, burn-in length W, momentum β
$\hat{v}_0 \leftarrow \text{diag}(\textsc{BurnIn}(W, \beta))$ ▷ Algorithm 8
for $t = 0, \ldots, T$ **do**
  $g_t \leftarrow$ stochastic gradient at w_t
  $\hat{v}_t \leftarrow \beta \hat{v}_{t-1} + (1-\beta) g_t^2$
  $A_t \leftarrow \text{diag}(\hat{v}_t)^{-1/2}$
  if $t \bmod t_{\text{thresh}} = 0$ **then**
    $w_{t+1} \leftarrow w_t - r A_t g_t$
    $\hat{v}_t \leftarrow \text{diag}(\textsc{Hallucinate}(S, \beta, \text{diag}(\hat{v}_t), w_t, w_{t+1}))$ ▷ Algorithm 9
  else
    $w_{t+1} \leftarrow w_t - \eta A_t g_t$
  end if
end for

D.4 Convergence results for the diagonal case

In this section we give convergence results for the diagonal approximation $A = \text{diag}(\mathbb{E}[g^2] + \varepsilon)^{-1/2}$.

There are three interacting components to the results. First, using estimates $Y_t = \text{diag}(g_t^2)$, Theorem 7.4.1 says we can accurately estimate $\mathbb{E}[Y_t] = \text{diag}(\mathbb{E}[g_t^2])$ via an exponential moving average, under reasonable assumptions. Second, the curvature and noise constants for this case are already given in Appendix D.3, specifically Proposition D.3.3. Finally, we plug these results in, together with Theorems 7.4.3 and 7.4.4, to get convergence bounds for the common diagonal version of RMSProp:

Corollary D.4.1. *Consider diagonal RMSProp with burn-in, as in Algorithm 11, where we estimate* $A = (\mathbb{E}[g^2] + \varepsilon)^{-1/2}$. *Retain the same choice of* $\eta = O(\tau^2)$ *and*

$T = O(\tau^{-4})$ as in Theorem 7.4.3. For small enough τ, such a choice of η will yield $\Delta < \lambda_-/2$. Choose all other parameters e.g. β in accordance with Proposition 7.4.2. In particular, choose $W = \Theta(\eta^{-2/3}) = \Theta(\tau^{-4/3}) = O(T)$ for the burn-in parameter. Then with probability $1-\delta$, in overall time $O(W+T) = O(\tau^{-4})$, we achieve

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(w_t)\|^2\right] \le \tau^2. \tag{D.41}$$

Corollary D.4.2. *Consider the diagonal RMSProp version of Algorithm 7 that is formalized in Algorithm 12. Retain the same choice of $\eta = O(\tau^{5/2})$, $r = O(\tau)$, and $T = O(\tau^{-5})$ as in Theorem 7.4.4. For small enough τ, such a choice of η will yield $\Delta < \lambda_-/2$. Choose $W = \Theta(\eta^{-2/3}) = \Theta(\tau^{-5/3}) = O(T)$ for the burn-in parameter Choose $S = O(\tau^{-3/2})$, so that as far as the estimation scheme is concerned, the stepsize is bounded by $\max\{\eta, r/S\} = O(\tau^{5/2}) = O(\eta)$. Then with probability $1-\delta$, we can reach an $(\tau, \sqrt{\rho\tau})$-stationary point in total time*

$$W + T = \tilde{O}\left(\frac{\Lambda_1^4\Lambda_2^4\Gamma^4}{\lambda_-^{10}\nu^4} \cdot \frac{L^3}{\rho\delta^3} \cdot \tau^{-5}\right), \tag{D.42}$$

where $\Lambda_1, \Lambda_2, \Gamma, \nu, \lambda_-$ are the constants describing $A = \mathrm{diag}(\mathbb{E}[g^2] + \varepsilon)^{-1/2}$.

Note that in Algorithms 11 and 12, it is simple to implement efficient diagonal versions of the BurnIn (Algorithm 8) and Hallucinate (Algorithm 9) subroutines.

D.5 Main Proof

Here we will study the convergence of Algorithm 7. This is the same as Algorithm 4 except that once in a while we take a large stepsize so we may escape saddlepoints.

In order to unify our results, we prove second order convergence for general preconditioners $A(w)$. The convergence rate will depend on various properties of $A(w)$, and $A = \mathbb{E}[gg^T]^{-1/2}$ will turn out to be particularly well-behaved.

D.5.1 Definitions

Let ρ be the Lipschitz constant of the Hessian H, and let α be the Lipschitz constant of the preconditioner matrix $A(w)$ as a function of the current iterate w. The usual stepsize is η, while r is the occasional large stepsize that happens every t_{thresh} iterations. δ is a small probability of failure, d is the dimension. Since it will recur often, we define $\kappa = (1 + \eta\gamma)$, where γ is the magnitude of the most negative eigenvalue of $A^{1/2}HA^{1/2}$. By the following lemma, we will be able to lower bound γ by $\lambda_{\min}(A)|\lambda_{\min}(H)| \geq \lambda_-\sqrt{\rho\tau}$:

Lemma D.5.1. *Suppose A and H are symmetric matrices, with $A \succ 0$ and $\lambda_{\min}(H) < 0$. Then there is a negative eigenvalue of $A^{1/2}HA^{1/2}$ with magnitude at least $\lambda_{\min}(A)|\lambda_{\min}(H)|$.*

Proof. Let v be the minimum eigenvector of H, so that $v^THv = -\lambda_{\min}(H)\|v\|^2 = -\lambda_{\min}(H)$. Define the unit vector $u = A^{-1/2}v/\|A^{-1/2}v\|$. Then,

$$u^TA^{1/2}HA^{1/2}u = \frac{1}{\|A^{-1/2}v\|^2}v^THv = -\frac{\lambda_{\min}(H)}{\|A^{-1/2}v\|^2}. \tag{D.43}$$

The vector u is not necessarily an eigenvector of $A^{1/2}HA^{1/2}$, but the above expression guarantees that $A^{1/2}HA^{1/2}$ has a negative eigenvalue with magnitude at least

$$\frac{\lambda_{\min}(H)}{\|A^{-1/2}v\|^2} \geq \frac{\lambda_{\min}(H)}{\lambda_{\max}(A^{-1})\|v\|^2} = \lambda_{\min}(H)\lambda_{\min}(A). \tag{D.44}$$

□

Throughout, we will assume that A is a $(\Lambda_1, \Lambda_2, \Gamma, \nu, \lambda_-)$-preconditioner, that $\hat{A}$ also satisfies the Λ_1 inequality, and that $\|\hat{A} - A\| \leq \Delta$.

Differing from Daneshmand et al. (2018), we will assume a uniform bound on $\|Ag\| \leq M$. In general this bound need not depend on either the spectrum of A or any uniform bound on g. For example, if g were Gaussian, Ag would be a Gaussian with zero mean and identity covariance, so we would expect $\|Ag\| = O(\sqrt{d})$ with high probability. In general M should have the same scale as $\sqrt{\Gamma}$, and the statement of Theorem 7.4.4 reflects this.

The proofs rely on a few other quantities that we will optimally determine as a function of the problem parameters: f_{thresh} is a threshold on the function value progress, and $g_{\text{thresh}} = f_{\text{thresh}}/t_{\text{thresh}}$ is the time-amortized average of f_{thresh}.

D.5.2 High level picture

For shorthand we write $A_t := A(w_t)$. Since we want to converge to a second order stationary point, our overall goal is to study the event

$$\mathcal{E}_t := \{\|\nabla f(w_t)\| \geq \tau \text{ or } \lambda_{\min}(\nabla^2 f(w_t)) \leq -\sqrt{\rho}\tau^{1/2}\} \tag{D.45}$$

$$= \{\|\nabla f(w_t)\| \geq \tau \text{ or } (\|\nabla f(w_t)\| \leq \tau \text{ and } \lambda_{\min}(\nabla^2 f(w_t)) \leq -\sqrt{\rho}\tau^{1/2})\}. \tag{D.46}$$

(where t is obvious from context, we will omit it. In words, $\mathcal{E}_t$ is the event that we are not at a second order stationary point. The main theorem results from bounding the progress we make when $\mathcal{E}_t$ does not yet hold, while also ensuring we do not leave once we hit a second order stationary point:

Lemma D.5.2. *Suppose that both*

$$\mathbb{E}[f(w_{t+1}) - f(w_t)|\mathcal{E}_t] \leq -g_{\text{thresh}} \tag{D.47}$$

$$\textit{and } \mathbb{E}[f(w_{t+1}) - f(w_t)|\mathcal{E}_t^c] \leq \delta g_{\text{thresh}}/2. \tag{D.48}$$

Set $T = 2(f(w_0) - \min_w f(w))/(\delta g_{\text{thresh}})$. *We return* w_t *uniformly randomly from* $w_1, \ldots, w_T$. *Then, with probability at least* $1 - \delta$, *we will have chosen a time* t *where* $\mathcal{E}_t$ *did not occur.*

Proof. Let P_t be the probability that $\mathcal{E}_t$ occurs. Then,

$$\mathbb{E}[f(w_{t+1}) - f(w_t)] = \mathbb{E}[f(w_{t+1}) - f(w_t)|\mathcal{E}_t]P_t + \mathbb{E}[f(w_{t+1}) - f(w_t)|\mathcal{E}_t^c](1 - P_t) \tag{D.49}$$

$$\leq -g_{\text{thresh}}P_t + \delta g_{\text{thresh}}/2 \cdot (1 - P_t) \tag{D.50}$$

$$\leq \delta g_{\text{thresh}}/2 - (1 + \delta/2)g_{\text{thresh}}P_t \tag{D.51}$$

$$\leq \delta g_{\text{thresh}}/2 - g_{\text{thresh}}P_t. \tag{D.52}$$

Summing over all T iterations, we have:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[f(w_{t+1}) - f(w_t)] \leq g_{\text{thresh}} \cdot \frac{1}{T}\sum_{t=1}^{T}(\delta/2 - P_t) \tag{D.53}$$

$$\implies \frac{1}{T}\sum_{t=1}^{T}P_t \leq \delta/2 + \frac{f(w_0) - f^*}{Tg_{\text{thresh}}} \leq \delta \tag{D.54}$$

$$\implies \frac{1}{T}\sum_{t=1}^{T}(1 - P_t) \geq 1 - \delta. \tag{D.55}$$

□

Theorem D.5.1. *Write $\gamma = \lambda_-\sqrt{\rho}\tau^{1/2}$. Let K be a universal constant. The parameter ω will be set later and depends only logarithmically on the other parameters. Set*

$$r = \gamma^2 \cdot \frac{\delta\nu K}{54\Lambda_1\Lambda_2\Gamma L\rho M}$$

$$\eta = \gamma^5 \cdot \frac{\delta^2\nu^2K^2}{324M^2L^2\Lambda_1^2\Lambda_2^2\Gamma^2\rho^2\omega}$$

$$f_{\text{thresh}} = \gamma^4 \cdot \frac{\delta\nu^2K^2}{54 \cdot 12\Lambda_1^2\Lambda_2^2\Gamma L\rho^2M^2}.$$

Let $t_{\text{thresh}} = \omega/(\eta\gamma)$, $\Delta = O(\tau^{1/2})$, and set $g_{\text{thresh}} = f_{\text{thresh}}/t_{\text{thresh}}$. Then we have both

$$\mathbb{E}[f(w_{t+1}) - f(w_t)|\mathcal{E}_t] \leq -g_{\text{thresh}} \tag{D.56}$$

$$\textit{and } \mathbb{E}[f(w_{t+1}) - f(w_t)|\mathcal{E}_t^c] \leq \delta g_{\text{thresh}}/2. \tag{D.57}$$

Corollary D.5.1. *In the above setting, with probability* $1-\delta$*, we reach an* $(\tau, \sqrt{\rho}\tau^{1/2})$*-stationary point in time*

$$\tilde{O}\left(\frac{M^4L^3}{\rho\delta^3}\cdot\frac{\Lambda_1^4\Lambda_2^4\Gamma^2}{\lambda_-^{10}\nu^4}\cdot\tau^{-5}\right). \tag{D.58}$$

Proof. Simply observe $T = C(f_0 - f^*)/(\delta g_{\text{thresh}})$ and plug in g_{thresh}. □

D.5.3 Amortized increase due to large stepsize iterations

Before we start casework on whether $\mathcal{E}_t$ holds We want to bound the amortized effect on the objective of increasing the stepsize every t_{thresh} iterations. By Corollary D.5.2,

$$\mathbb{E}[f(w_{t+1})] - f(w_t) \le \frac{9L\Gamma r^2}{8}. \tag{D.59}$$

Note that for our particular setting of r and f_{thresh}, we have

$$\frac{9L\Gamma}{8}\cdot r^2 = \frac{9L\Gamma}{8}\cdot\gamma^4\cdot\frac{\delta^2\nu^2K^2}{54^2\Lambda_1^2\Lambda_2^2\Gamma^2L^2\rho^2M^2} \tag{D.60}$$

$$= \frac{9\delta}{8}\cdot\frac{12}{54}\cdot\gamma^4\cdot\frac{\delta\nu^2K^2}{54\cdot 12\Lambda_1^2\Lambda_2^2\Gamma L\rho^2M^2} \tag{D.61}$$

$$= \frac{\delta f_{\text{thresh}}}{4}, \tag{D.62}$$

so also

$$\mathbb{E}[f(w_{t+1})] - f(w_t) \le \frac{\delta f_{\text{thresh}}}{4}. \tag{D.63}$$

Therefore on average

$$\frac{\mathbb{E}[f(w_{t+1})] - f(w_t)}{t_{\text{thresh}}} \le \delta g_{\text{thresh}}/4. \tag{D.64}$$

D.5.4 Bound on possible increase when $\mathcal{E}_t^c$ occurs

For the main result we need to bound

$$\mathbb{E}[f(w_{t+1}) - f(w_t)|\mathcal{E}_t^c] \leq \delta g_{\text{thresh}}/4. \tag{D.65}$$

Note that

$$\mathcal{E}_t^c = \{\|\nabla f(w_t)\| \geq \tau \text{ or } \lambda_{\min}(\nabla^2 f(w_t)) \leq -\sqrt{\rho}\tau^{1/2}\}^c \tag{D.66}$$

$$= \{\|\nabla f(w_t)\| < \tau \text{ and } \lambda_{\min}(\nabla^2 f(w_t)) > -\sqrt{\rho}\tau^{1/2}\}. \tag{D.67}$$

Hence it suffices to bound the function increase conditioned on $\|\nabla f(w_t)\| \leq \tau$. By Corollary D.5.2 we have

$$\mathbb{E}[f(w_{t+1})] - f(w_t) \leq \frac{9L\Gamma\eta^2}{8}. \tag{D.68}$$

We want this to not exceed $\delta g_{\text{thresh}}/4$:

$$\frac{9L\Gamma\eta^2}{8} \overset{?}{\leq} \frac{\delta}{4} g_{\text{thresh}} \tag{D.69}$$

$$\Leftrightarrow \frac{9L\Gamma\eta^2}{8} \overset{?}{\leq} \frac{\delta}{4} f_{\text{thresh}} \cdot \frac{\eta\gamma}{\omega} \tag{D.70}$$

$$\Leftrightarrow \frac{9L\Gamma\eta}{2} \overset{?}{\leq} \delta f_{\text{thresh}} \cdot \frac{\gamma}{\omega} \tag{D.71}$$

$$\Leftrightarrow \frac{9L\Gamma}{2} \cdot \gamma^5 \cdot \frac{\delta^2\nu^2K^2}{324M^2L^2\Lambda_1^2\Lambda_2^2\Gamma^2\rho^2\omega} \overset{?}{\leq} \delta \cdot \frac{\gamma}{\omega} \cdot \gamma^4 \cdot \frac{\delta\nu^2K^2}{54 \cdot 12\Lambda_1^2\Lambda_2^2\Gamma L\rho^2M^2}. \tag{D.72}$$

Cancelling like terms, we find that the inequality is equivalent to $\omega \geq 9/4$, which we can easily enforce later. Therefore we may indeed write that

$$\mathbb{E}[f(w_{t+1})] - f(w_t) \leq \frac{\delta g_{\text{thresh}}}{4}. \tag{D.73}$$

D.5.5 Bound on decrease (progress) when $\mathcal{E}_t$ occurs

We need to bound

$$\mathbb{E}[f(w_{t+1}) - f(w_t)|\mathcal{E}_t] \leq -g_{\text{thresh}}. \tag{D.74}$$

By definition,

$$\mathcal{E}_t = \{\|\nabla f(w_t)\| \geq \tau\} \cup \{\lambda_{\min}(\nabla^2 f(w_t)) \leq -\sqrt{\rho}\tau^{1/2} \text{ and } \|\nabla f(w_t)\| \leq \tau\}. \tag{D.75}$$

In words, we split $\mathcal{E}_t$ into two cases: either the gradient is large, or we are near a saddlepoint but there is an escape direction.

Large gradient regime

If the norm of the gradient is large enough, i.e.

$$\|\nabla f(w_t)\|^2 \geq \tau^2 \tag{D.76}$$

then by Corollary D.5.4,

$$\mathbb{E}[f(w_{t+1})] - f(w_t) \leq -\frac{\eta\tau^2\lambda_-}{4} \leq -g_{\text{thresh}} \tag{D.77}$$

as long as $\eta \leq \frac{4\lambda_-\tau^2}{9L\Gamma}$ and $g_{\text{thresh}} \leq \frac{\eta\tau^2\lambda_-}{4}$. For our choice of $\eta = O(\tau^{5/2})$ and $g_{\text{thresh}} = \tilde{O}(\tau^5)$, each of these will hold for small enough τ.

Sharp negative curvature regime

We start at a point w_0 around which we base our Hessian approximation:

$$g(w) = f(w_0) + (w - w_0)^T \nabla f(w_0) + \frac{1}{2}(w - w_0)^T H(w - w_0) \tag{D.78}$$

where we write $H = \nabla^2 f(w_0)$. We will also write $A = \mathbb{E}[g_0 g_0^T]^{1/2}$ as the preconditioner at w_0.

Lemma D.5.3. *For every twice differentiable ρ-Hessian Lipschitz function f we have*

$$\|\nabla f(w) - \nabla g(w)\| \le \frac{\rho}{2}\|w - w_0\|^2. \tag{D.79}$$

Proof. From Lemma 1.2.4 in (Nesterov, 2004), we have

$$\|\nabla f(w) - \nabla f(w_0) - H(w - w_0)\| \le \frac{\rho}{2}\|w - w_0\|^2. \tag{D.80}$$

Now simply observe that $\nabla g(w) = \nabla f(w_0) + H(w - w_0)$. □

Lemma D.5.4. *Suppose that $\|\nabla f(w_0)\| \le \tau$. Also suppose the Hessian at w_0 has a strong escape direction, i.e. $\lambda_{\min}(\nabla^2 f(w_0)) \le -\sqrt{\rho}\tau^{1/2}$, and define $\gamma = \lambda_-\sqrt{\rho}\tau^{1/2}$ so that $\sqrt{\rho}\tau^{1/2} = \lambda_-^{-1}\gamma$. Then there exists $k < t_{\text{thresh}}$ so that*

$$\mathbb{E}[f(w_k)] - f(w_0) \le -f_{\text{thresh}} \tag{D.81}$$

Proof. Suppose not, i.e. suppose that for all $t < t_{\text{thresh}}$ it holds that

$$\mathbb{E}[f(w_t)] - f(w_0) \ge -f_{\text{thresh}}. \tag{D.82}$$

Under this assumption we will prove bounds which will imply that the assumption cannot hold. In particular, we will give a lower bound on $\mathbb{E}[\|w_t - w_0\|^2]$ that conflicts with Lemma D.5.11.

Define the following terms, each of which is selected to satisfy a certain recursion:

Term	Recursion identity
$u_t = (I - \eta AH)^t(w_1 - w_0)$.	$u_t = (I - \eta AH)u_{t-1}$
$\delta_t = \sum_{i=1}^t (I - \eta AH)^{t-i} A(-\nabla f(w_i) + \nabla g(w_i))$	$\delta_t = A(-\nabla f(w_t) + \nabla g(w_t)) + (I - \eta AH)\delta_{t-1}$
$d_t = -\sum_{i=1}^t (I - \eta AH)^{t-i} A\nabla f(w_0)$	$d_t = -A\nabla f(w_0) + (I - \eta AH)d_{t-1}$
$\zeta_t = \sum_{i=1}^t (I - \eta AH)^{t-i}\xi_i$	$\zeta_t = \xi_t + (I - \eta AH)\zeta_{t-1}$
$\chi_t = \sum_{i=1}^t (I - \eta AH)^{t-i}(A - A_i)\nabla f(w_i)$	$\chi_t = (A - A_t)\nabla f(w_t) + (I - \eta AH)\chi_{t-1}$
$\iota_t = \sum_{i=1}^t (I - \eta AH)^{t-i}(A_i - \hat{A}_i)\nabla f(w_i)$	$\iota_t = (A_t - \hat{A}_t)\nabla f(w_t) + (I - \eta AH)\iota_{t-1}$.

By convention we take $\delta_0 = d_0 = \zeta_0 = \chi_0 = \iota_0 = 0$, and for convenience we will write $\pi_t = \delta_t + d_t + \zeta_t + \chi_t + \iota_t$. These terms are chosen so that each is a kind of error term in a stale Taylor expansion of f. The error terms cancel so that the following identity holds:

$$\pi_t = (I - \eta AH)\pi_{t-1} + AH(w_t - w_0) - \hat{A}_t \nabla f(w_t) + \xi_t. \tag{D.83}$$

We will inductively prove the identity

$$w_{t+1} - w_0 = u_t + \eta \pi_t \tag{D.84}$$

$$= u_t + \eta(\delta_t + d_t + \zeta_t + \chi_t + \iota_t) \tag{D.85}$$

for $t \geq 0$. The base case is simple because $u_0 = w_1 - w_0$ and the other terms are all zero. For the inductive step, note

$$\begin{aligned}(w_{t+1} - w_0) - (w_t - w_0) &= w_{t+1} - w_t \\ &= -\eta \hat{A}_t \nabla f(w_t) + \eta \xi_t \\ &= \eta[\pi_t - (I - \eta AH)\pi_{t-1} - AH(w_t - w_0)]\end{aligned}$$

where the last equality follows from equation (D.83). Rearranging, we have

$$w_{t+1} - w_0 = \eta[\pi_t - (I - \eta AH)\pi_{t-1}] + (I - \eta AH)(w_t - w_0) \tag{D.86}$$

$$= \eta[\pi_t - (I - \eta AH)\pi_{t-1}] + (I - \eta AH)(u_{t-1} + \eta \pi_{t-1}) \tag{D.87}$$

by induction. Since $u_t = (I - \eta AH)u_{t-1}$, we can further simplify:

$$w_{t+1} - w_0 = \eta[\pi_t - (I - \eta AH)\pi_{t-1}] + (I - \eta AH)(u_{t-1} + \eta \pi_{t-1}) \tag{D.88}$$

$$= \eta[\pi_t - (I - \eta AH)\pi_{t-1}] + u_t + \eta(I - \eta AH)\pi_{t-1}) \tag{D.89}$$

$$= u_t + \eta \pi_t \tag{D.90}$$

as desired.

To proceed with our saddle point escape argument, we must bound all the terms $u_t, \delta_t, d_t, \zeta_t, \chi_t, \iota_t$ to show that $w_t - w_0$ grows fast enough.

Lemma D.5.5. *Under the above conditions, we have*

$$\mathbb{E}[\|\chi_t\|] \le \alpha\tau\sqrt{\eta^3 L\Gamma\Lambda_1}\cdot\kappa^t\cdot\left(\frac{4}{(\eta\gamma)^2}+\frac{6f_{\text{thresh}}}{\eta^3\gamma L\Gamma}+\frac{2}{\eta\gamma}\cdot\sqrt{\frac{2r^2}{\eta^3 L\Lambda_1}}\right). \tag{D.91}$$

Proof. We assume $A(x)$ is α Lipschitz, so that $\|A_i - A\| \le \alpha\|w_i - w_0\|$. Then,

$$\mathbb{E}[\|\chi_t\|] = \mathbb{E}\left[\left\|\sum_{i=1}^{t}(I-\eta AH)^{t-i}(A-A_i)\nabla f(x_i)\right\|\right] \tag{D.92}$$

$$\le \sum_{i=1}^{t}(1+\eta\gamma)^{t-i}\,\mathbb{E}\left[\|(A-A_i)\nabla f(x_i)\|\right] \tag{D.93}$$

$$\le \sum_{i=1}^{t}(1+\eta\gamma)^{t-i}\,\mathbb{E}\left[\|A-A_i\|\|\nabla f(x_i)\|\right] \tag{D.94}$$

$$\le \tau\sum_{i=1}^{t}(1+\eta\gamma)^{t-i}\,\mathbb{E}\left[\|A-A_i\|\right] \tag{D.95}$$

$$\le \alpha\tau\sum_{i=1}^{t}(1+\eta\gamma)^{t-i}\,\mathbb{E}\left[\|w_i-w_0\|\right] \tag{D.96}$$

$$\le \alpha\tau\sum_{i=1}^{t}(1+\eta\gamma)^{t-i}\sqrt{\mathbb{E}\left[\|w_i-w_0\|^2\right]} \tag{D.97}$$

$$\le \alpha\tau\sum_{i=1}^{t}(1+\eta\gamma)^{t-i}\sqrt{6\eta f_{\text{thresh}}\Lambda_1 i+\eta^3 L\Gamma\Lambda_1 i^2+2\Gamma r^2} \tag{D.98}$$

where for the last identity we have applied Lemma D.5.11. By Lemma D.5.12, we may further bound this by

$$\mathbb{E}[\|\chi_t\|] \le \alpha\tau\sqrt{\eta^3 L\Gamma\Lambda_1}\sum_{i=1}^{t}(1+\eta\gamma)^{t-i}\left(2i+\frac{3\eta f_{\text{thresh}}\Lambda_1}{\eta^3 L\Gamma\Lambda_1}+\sqrt{\frac{2\Gamma r^2}{\eta^3 L\Gamma\Lambda_1}}\right) \tag{D.99}$$

$$= \alpha\tau\sqrt{\eta^3 L\Gamma\Lambda_1}\sum_{i=1}^{t}(1+\eta\gamma)^{t-i}\left(2i+\frac{3f_{\text{thresh}}}{\eta^2 L\Gamma}+\sqrt{\frac{2r^2}{\eta^3 L\Lambda_1}}\right). \tag{D.100}$$

Applying Lemma D.5.14 with $\beta = \eta\gamma$ yields:

$$\mathbb{E}[\|\chi\|] \leq \alpha\tau\sqrt{\eta^3 L\Gamma\Lambda_1} \cdot \kappa^t \cdot \left(\frac{4}{(\eta\gamma)^2} + \frac{2}{\eta\gamma} \cdot \frac{3f_{\text{thresh}}}{\eta^2 L\Gamma} + \frac{2}{\eta\gamma} \cdot \sqrt{\frac{2r^2}{\eta^3 L\Lambda_1}}\right) \tag{D.101}$$

$$= \alpha\tau\sqrt{\eta^3 L\Gamma\Lambda_1} \cdot \kappa^t \cdot \left(\frac{4}{(\eta\gamma)^2} + \frac{6f_{\text{thresh}}}{\eta^3\gamma L\Gamma} + \frac{2}{\eta\gamma} \cdot \sqrt{\frac{2r^2}{\eta^3 L\Lambda_1}}\right). \tag{D.102}$$

□

Lemma D.5.6. *Under the above conditions, we have*

$$\mathbb{E}[\|\delta_t\|] \leq \Lambda_2\rho\kappa^t \left[\frac{2\Gamma r^2}{\eta\gamma} + \frac{6\eta f_{\text{thresh}}\Lambda_1}{(\eta\gamma)^2} + \frac{3\eta^3 L\Gamma\Lambda_1}{(\eta\gamma)^3}\right]. \tag{D.103}$$

Proof. We write

$$\mathbb{E}[\|\delta_t\|] = \mathbb{E}\left[\left\|\sum_{i=1}^{t}(I - \eta AH)^{t-i}A(\nabla f(w_i) - \nabla g(w_i))\right\|\right] \tag{D.104}$$

$$\leq \sum_{i=1}^{t}(1+\eta\gamma)^{t-i}\,\mathbb{E}\left[\|A(\nabla f(w_i) - \nabla g(w_i))\|\right] \tag{D.105}$$

$$\leq \Lambda_2 \sum_{i=1}^{t}\kappa^{t-i}\,\mathbb{E}\left[\|\nabla f(w_i) - \nabla g(w_i)\|\right] \tag{D.106}$$

$$\leq \Lambda_2(\rho/2)\sum_{i=1}^{t}\kappa^{t-i}\,\mathbb{E}\left[\|w_i - w_0\|^2\right] \tag{D.107}$$

$$\leq \Lambda_2(\rho/2)\sum_{i=1}^{t}\kappa^{t-i}\left(6\eta f_{\text{thresh}}\Lambda_1 i + \eta^3 L\Gamma\Lambda_1 i^2 + 2\Gamma r^2\right), \tag{D.108}$$

where again, the last inequality comes from Lemma D.5.11. Applying Lemma D.5.14 with $\beta = \eta\gamma$ yields:

$$\mathbb{E}[\|\delta_t\|] \leq \frac{\Lambda_2\rho\kappa^t}{2}\left[(6\eta f_{\text{thresh}}\Lambda_1)\cdot\frac{2}{\eta^2\gamma^2} + \eta^3 L\Gamma\Lambda_1 \cdot \frac{6}{\eta^3\gamma^3} + 2\Gamma r^2 \cdot \frac{2}{\eta\gamma}\right] \tag{D.109}$$

$$= \Lambda_2\rho\kappa^t\left[\frac{2\Gamma r^2}{\eta\gamma} + \frac{6\eta f_{\text{thresh}}\Lambda_1}{(\eta\gamma)^2} + \frac{3\eta^3 L\Gamma\Lambda_1}{(\eta\gamma)^3}\right]. \tag{D.110}$$

□

Lemma D.5.7. *Under the above conditions,*

$$\mathbb{E}\|\iota_t\|] \leq 2\tau(\eta\gamma)^{-1}\Delta\kappa^t. \tag{D.111}$$

Proof. Write

$$\mathbb{E}\|\iota_t\|] = \mathbb{E}\left[\left\|\sum_{i=1}^{t}(I-\eta AH)^{t-i}(A_i-\hat{A}_i)\nabla f(w_i)\right\|\right] \tag{D.112}$$

$$\leq \sum_{i=1}^{t}(1+\eta\gamma)^{t-i}\,\mathbb{E}\left[\left\|(A_i-\hat{A}_i)\nabla f(w_i)\right\|\right] \tag{D.113}$$

$$\leq \tau\sum_{i=1}^{t}(1+\eta\gamma)^{t-i}\,\mathbb{E}\left[\left\|A_i-\hat{A}_i\right\|\right] \tag{D.114}$$

$$\leq 2\tau(\eta\gamma)^{-1}\kappa^t \max_i \mathbb{E}\left[\left\|A_i-\hat{A}_i\right\|\right] \tag{D.115}$$

$$\leq 2\tau(\eta\gamma)^{-1}\Delta\kappa^t. \tag{D.116}$$

□

Lemma D.5.8. *Under the above conditions,* $\mathbb{E}[u_t^T]d_t \geq 0$.

Proof. We have

$$\mathbb{E}[u_t] = (I-\eta AH)^t\,\mathbb{E}[w_1-w_0] = -r(I-\eta AH)^t A\nabla f(w_0). \tag{D.117}$$

For small enough η, we have $\|\eta AH\| \leq 1$ and hence:

$$\mathbb{E}[u_t^T]d_t = r\left[(I-\eta AH)^t A\nabla f(w_0)\right]^T\sum_{i=1}^{t}(I-\eta AH)^{t-i}A\nabla f(w_0) \tag{D.118}$$

$$= r\sum_{i=1}^{t}(A\nabla f(w_0))^T(I-\eta AH)^{2t-i}(A\nabla f(w_0)) \geq 0. \tag{D.119}$$

□

Lemma D.5.9. *Under the above conditions, we get an exponentially growing lower*

bound on the expected squared norm of u_t:

$$\mathbb{E}[\|u_t\|^2] \geq (1+\eta\gamma)^{2t} r^2 \nu = \kappa^{2t} r^2 \nu. \tag{D.120}$$

Proof. For unit vectors v, we may write

$$\mathbb{E}[\|u_t\|^2] \geq \mathbb{E}[(v^T u_t)^2]. \tag{D.121}$$

In particular, by definition of u_t,

$$\mathbb{E}[\|u_t\|^2] \geq \mathbb{E}[(v^T (I - \eta AH)^t (w_1 - w_0))^2]. \tag{D.122}$$

We wish to choose a unit vector v so that this is as large as possible. If AH were symmetric, we could choose v to be an eigenvector, but the product of symmetric matrices is not in general symmetric. However, because A and H are both symmetric, and A is positive definite, it follows that $A^{1/2}$ exists and that $A^{1/2} H A^{1/2}$ is symmetric. Hence for orthonormal U and diagonal Λ, we have

$$A^{1/2} H A^{1/2} = U \Lambda U^T \tag{D.123}$$

$$\implies A^{-1/2} A H A^{1/2} = U \Lambda U^T \tag{D.124}$$

$$\implies AH = A^{1/2} U \Lambda (A^{1/2} U)^{-1}. \tag{D.125}$$

The diagonal matrix Λ contains the eigenvalues of $A^{1/2} H A^{1/2}$. Without loss of generality, Λ_{11} corresponds to a negative eigenvalue with absolute value γ. Therefore

$$(I - \eta AH)^t = (A^{1/2} U (I - \eta\Lambda)(A^{1/2} U)^{-1})^t \tag{D.126}$$

$$= A^{1/2} U (I - \eta\Lambda)^t (A^{1/2} U)^{-1}. \tag{D.127}$$

Since we can choose v to be any unit vector we want, we will set it equal to $C(U^T A^{1/2})^{-1} e_1$ so that $U^T A^{1/2} v = C e_1$. Here e_1 is the first standard basis vector and C is a scalar constant chosen to make v a unit vector. Taking transposes, we have $v^T A^{1/2} U = C e_1^T$.

Now,

$$v^T(I - \eta AH)^t = v^T A^{1/2} U (I - \eta\Lambda)^t (A^{1/2}U)^{-1} \tag{D.128}$$

$$= Ce_1^T (I - \eta\Lambda)^t (A^{1/2}U)^{-1} \tag{D.129}$$

$$= C(1 + \eta\Lambda_{11})^t e_1^T (A^{1/2}U)^{-1} \tag{D.130}$$

$$= (1 + \eta\gamma)^t \cdot Ce_1^T (A^{1/2}U)^{-1}. \tag{D.131}$$

Substituting in the definition of v, this is equal to:

$$v^T(I - \eta AH)^t = (1 + \eta\gamma)^t \cdot v^T (A^{1/2}U)(A^{1/2}U)^{-1} \tag{D.132}$$

$$= (1 + \eta\gamma)^t v^T. \tag{D.133}$$

This equality holds for any v of the form specified above; in particular, choose C so that v is unit. Then, we may finally bound

$$\mathbb{E}[\|u_t\|^2] \geq \mathbb{E}[(v^T(I - \eta AH)^t(w_1 - w_0))^2] \tag{D.134}$$

$$\geq (1 + \eta\gamma)^{2t}\, \mathbb{E}[(v^T(w_1 - w_0))^2] \tag{D.135}$$

$$= (1 + \eta\gamma)^{2t} r^2\, \mathbb{E}[(v^T A g_0)^2] \tag{D.136}$$

$$= (1 + \eta\gamma)^{2t} r^2 v^T\, \mathbb{E}[A g_0 g_0^T A^T] v \tag{D.137}$$

$$= (1 + \eta\gamma)^{2t} r^2 v^T A\, \mathbb{E}[g_0 g_0^T] A^T v \tag{D.138}$$

$$\geq (1 + \eta\gamma)^{2t} r^2 \lambda_{\min}(A\, \mathbb{E}[g_0 g_0^T] A^T) \tag{D.139}$$

$$\geq (1 + \eta\gamma)^{2t} r^2 \nu, \tag{D.140}$$

where the last two lines follow by the fact that $\|v\| = 1$ and by definition of ν. $\square$

Lemma D.5.10. *Under the above conditions we have a deterministic bound on $\|u_t\|$:*

$$\|u_t\| \leq \kappa^t r M \tag{D.141}$$

Proof. We write

$$\|u_t\| = \|(I - \eta AH)^t (w_1 - w_0)\| \tag{D.142}$$

$$\leq \|I - \eta AH\|^t \cdot \|w_1 - w_0\| \tag{D.143}$$

$$\leq (1 + \eta\gamma)^t \cdot r\|Ag_0\| \tag{D.144}$$

$$\leq (1 + \eta\gamma)^t \cdot rM. \tag{D.145}$$

□

Putting all these results together, we can give a lower bound on the distance between iterates:

$$\begin{aligned}
\mathbb{E}[\|w_{t+1} - w_0\|^2] &= \mathbb{E}\left[\|u_t + \eta(\delta_t + d_t + \zeta_t + \chi_t + \iota_t)\|^2\right] \\
&= \mathbb{E}[\|u_t\|^2] + 2\eta\,\mathbb{E}\left[u_t^T(\delta_t + d_t + \zeta_t + \chi_t + \iota_t)\right] \\
&\quad + \eta^2\,\mathbb{E}\left[\|\delta_t + d_t + \zeta_t + \chi_t + \iota_t\|^2\right] \\
&\geq \mathbb{E}[\|u_t\|^2] + 2\eta\,\mathbb{E}\left[u_t^T(\delta_t + d_t + \zeta_t + \chi_t + \iota_t)\right] \\
&= \mathbb{E}[\|u_t\|^2] + 2\eta\,\mathbb{E}\left[u_t^T(\delta_t + d_t + \chi_t + \iota_t)\right] \\
&= \mathbb{E}[\|u_t\|^2] + 2\eta\,\mathbb{E}[u_t^T\delta_t] + 2\eta\,\mathbb{E}[u_t^T d_t] + 2\eta\,\mathbb{E}[u_t^T\chi_t] \\
&= \mathbb{E}[\|u_t\|^2] + 2\eta\,\mathbb{E}[u_t^T\delta_t] + 2\eta\,\mathbb{E}[u_t]^T d_t + 2\eta\,\mathbb{E}[u_t^T\chi_t] + 2\eta\,\mathbb{E}[u_t^T\iota_t] \\
&\geq \mathbb{E}[\|u_t\|^2] + 2\eta\,\mathbb{E}[u_t^T\delta_t] + 2\eta\,\mathbb{E}[u_t^T\chi_t] + 2\eta\,\mathbb{E}[u_t^T\iota_t] \\
&\geq \mathbb{E}[\|u_t\|^2] - 2\eta\|u_t\|\,\mathbb{E}[\|\delta_t\|] - 2\eta\|u_t\|\,\mathbb{E}[\|\chi_t\|] - 2\eta\|u_t\|\,\mathbb{E}[\|\iota_t\|] \\
&\geq \kappa^{2t} r^2\nu - 2\eta\kappa^t rM\,\mathbb{E}[\|\delta_t\| + \|\chi_t\| + \|\iota_t\|].
\end{aligned}$$

Substituting in the bounds for $\mathbb{E}[\|\delta_t\|]$, $\mathbb{E}[\|\chi_t\|]$, and $\mathbb{E}[\|\iota_t\|]$,we finally have the lower bound:

$$\left(r\nu - 2\eta M\left[\Lambda_2\rho\left[\frac{2\Gamma r^2}{\eta\gamma} + \frac{6\eta f_{\text{thresh}}\Lambda_1}{(\eta\gamma)^2} + \frac{3\eta^3 L\Gamma\Lambda_1}{(\eta\gamma)^3}\right]\right.\right. \tag{D.146}$$

$$\left.\left. + \alpha\tau\sqrt{\eta^3 L\Gamma\Lambda_1}\left(\frac{4}{(\eta\gamma)^2} + \frac{6f_{\text{thresh}}}{\eta^3\gamma L\Gamma} + \frac{2}{\eta\gamma}\cdot\sqrt{\frac{2r^2}{\eta^3 L\Lambda_1}}\right) + 2\tau(\eta\gamma)^{-1}\Delta\right]\right) r\kappa^{2t}. \tag{D.147}$$

As long as the sum in the parentheses is positive, this term will grow exponentially and grant us the contradiction we seek. We want to bound each of the seven terms in brackets by $r\nu/8$, so that the overall bound is $r^2\kappa^{2t}\nu/8$. For simplicity, we will write $K = 1/8$ as a universal constant. Then, we want to choose parameters so the following inequalities all hold.

We start with the last term (from ι_t) because it is the most simple. Since $\gamma = \Theta(\tau^{1/2})$, we require that

$$2\eta M \cdot 2\tau(\eta\gamma)^{-1}\Delta \le r\nu K \tag{D.148}$$

$$\Leftrightarrow 4M\tau\gamma^{-1}\Delta \le r\nu K \tag{D.149}$$

$$\Leftrightarrow \tau \cdot \tau^{-1/2}\Delta \le O(r) \tag{D.150}$$

$$\Leftrightarrow \Delta \le O(\tau^{-1/2}r). \tag{D.151}$$

Since we will eventually set $r = O(\tau)$, this constraint is simply $\Delta \le O(\tau^{1/2})$.

Next we move onto the first three terms, which correspond to δ_t:

$$2\eta M\Lambda_2\rho \cdot \frac{2\Gamma r^2}{\eta\gamma} \le r\nu K \Leftrightarrow r \le \frac{\gamma\nu K}{4\Lambda_2\Gamma\rho M} \tag{D.152}$$

$$2\eta M\Lambda_2\rho \cdot \frac{6\eta f_{\text{thresh}}\Lambda_1}{\eta^2\gamma^2} \le r\nu K \Leftrightarrow f_{\text{thresh}} \le \frac{\gamma^2 r\nu K}{12\Lambda_1\Lambda_2\rho M} \tag{D.153}$$

$$2\eta M\Lambda_2\rho \cdot \frac{3\eta^3 L\Gamma\Lambda_1}{\eta^3\gamma^3} \le r\nu K \Leftrightarrow \eta \le \frac{\gamma^3 r\nu K}{6ML\Lambda_1\Lambda_2\Gamma\rho}. \tag{D.154}$$

The first constraint is satisfied for small enough τ because we chose $r = O(\tau) \le O(\tau^{1/2})$. The second term is equivalent to

$$f_{\text{thresh}} \overset{?}{\le} \frac{\gamma^2\nu K}{12\Lambda_1\Lambda_2\rho M} \cdot r \tag{D.155}$$

$$\Leftrightarrow \gamma^4 \cdot \frac{\delta\nu^2K^2}{54 \cdot 12\Lambda_1^2\Lambda_2^2\Gamma L\rho^2M^2} \overset{?}{\le} \frac{\gamma^2\nu K}{12\Lambda_1\Lambda_2\rho M} \cdot \gamma^2 \cdot \frac{\delta\nu K}{54\Lambda_1\Lambda_2\Gamma L\rho M} \tag{D.156}$$

$$\Leftrightarrow \frac{\delta\nu^2K^2}{54 \cdot 12\Lambda_1^2\Lambda_2^2\Gamma L\rho^2M^2} \overset{?}{\le} \frac{\delta\nu^2K^2}{54 \cdot 12\Lambda_1^2\Lambda_2^2\Gamma L\rho^2M^2} \tag{D.157}$$

which trivially always holds since the two expressions are equal.

Finally, we address the three terms corresponding to χ_t. For small enough τ, it

will turn out that none of the resulting constraints are tight, i.e. they are all weaker than some other constraint we already require. First,

$$2\eta M\alpha\tau\sqrt{\eta^3 L\Gamma\Lambda_1}\cdot\frac{4}{\eta^2\gamma^2}\leq r\nu K \tag{D.158}$$

$$\Leftarrow \eta^{1/2}\tau\leq O(r\gamma^2) \tag{D.159}$$

$$\Leftrightarrow \eta\leq O(r^2\gamma^4\tau^{-2})=O(\tau^2). \tag{D.160}$$

Next,

$$2\eta M\alpha\tau\sqrt{\eta^3 L\Gamma\Lambda_1}\cdot\frac{6f_{\text{thresh}}}{\eta^3\gamma L\Gamma}\leq r\nu K \tag{D.161}$$

$$\Leftarrow \eta\tau\eta^{3/2}\frac{f_{\text{thresh}}}{\eta^3\gamma}\leq O(r) \tag{D.162}$$

$$\Leftrightarrow f_{\text{thresh}}\leq O(\eta^{1/2}r\gamma\tau^{-1})=O(\tau^{7/4}). \tag{D.163}$$

Finally,

$$2\eta M\alpha\tau\sqrt{\eta^3 L\Gamma\Lambda_1}\cdot\frac{2}{\eta\gamma}\cdot\sqrt{\frac{2r^2}{\eta^3 L\Lambda_1}}\leq r\nu K \tag{D.164}$$

$$\Leftarrow \tau\sqrt{\eta^3}\cdot\frac{1}{\gamma}\cdot\frac{r}{\sqrt{\eta^3}}\leq O(r) \tag{D.165}$$

$$\Leftrightarrow \tau\gamma^{-1}r\leq O(r) \tag{D.166}$$

$$\Leftrightarrow \tau\leq O(\gamma)=O(\tau^{1/2}). \tag{D.167}$$

Hence, for small enough τ, for the above parameter settings, we have

$$\mathbb{E}[\|w_{t+1}-w_0\|^2]\geq r^2\kappa^{2t}\nu K. \tag{D.168}$$

We now have a lower bound and an upper bound that when combined yield $(1+\eta\gamma)^{2t}\leq C$, where

$$C=\left[(6\eta f_{\text{thresh}}\Lambda_1)\,t+\eta^3 L\Gamma\Lambda_1 t^2+2\Gamma r^2\right]\cdot\frac{1}{r^2\nu K}. \tag{D.169}$$

We can choose ω that is only logarithmic in all parameters, i.e. $\omega = O(\log(\frac{\Lambda_1\Lambda_2\Gamma L\eta f_{\text{thresh}}}{\nu r}))$, so that setting $t \geq t_{\text{thresh}} = \omega/(\eta\gamma)$ yields $(1+\eta\gamma)^{2t} \geq C$. This contradicts the upper bound, as desired.

□

Lemma D.5.11. *Assume that Equation (D.82) holds. Assume also that* $\eta \leq \frac{f_{\text{thresh}}\Lambda_1}{\Gamma}$. *Then,*

$$\mathbb{E}[\|w_t - w_0\|^2] \leq 6\eta f_{\text{thresh}}\Lambda_1 t + \eta^3 L\Gamma\Lambda_1 t^2 + 2\Gamma r^2. \tag{D.170}$$

Proof. By Lemma D.5.16,

$$-f_{\text{thresh}} \leq \mathbb{E}[f(w_t)] - f(w_0) \tag{D.171}$$

$$= \mathbb{E}\left[\sum_{i=0}^{t-1} f(w_{i+1}) - f(w_i)\right] \tag{D.172}$$

$$\leq -\eta \sum_{i=0}^{t-1} \mathbb{E}[\|\hat{A}_i^{1/2}\nabla f(w_i)\|^2] + \frac{\eta^2 L\Gamma(t-1)}{2} + \frac{r^2 L\Gamma}{2}. \tag{D.173}$$

Remember, we are making the simplifying assumption that Λ_1 serves as a bound in the same way for $\hat{A}$ as it does for A. This is trivially true if $\Delta = 0$. Applying the definition of Λ_1 yields:

$$-f_{\text{thresh}} \leq -\eta\Lambda_1^{-1} \sum_{i=0}^{t-1} \mathbb{E}[\|\hat{A}_i\nabla f(w_i)\|^2] + \frac{\eta^2 L\Gamma t}{2} + \frac{r^2 L\Gamma}{2}. \tag{D.174}$$

By rearranging, we can get a bound on the gradient norms:

$$\sum_{i=0}^{t-1} \mathbb{E}[\|\hat{A}_i\nabla f(w_i)\|^2] \leq \frac{\Lambda_1}{\eta}\left(\frac{\eta^2 L\Gamma t}{2} + \frac{r^2 L\Gamma}{2} + f_{\text{thresh}}\right) \tag{D.175}$$

$$= \frac{\eta L\Gamma\Lambda_1 t}{2} + \frac{r^2 L\Gamma\Lambda_1}{2\eta} + \frac{f_{\text{thresh}}\Lambda_1}{\eta}. \tag{D.176}$$

Before we proceed, note that we already have

$$\frac{\delta f_{\text{thresh}}}{4} \geq \frac{9L\Gamma r^2}{8} \implies \frac{f_{\text{thresh}}\Lambda_1}{\eta} \geq \frac{9}{2\delta}\frac{r^2 L\Gamma\Lambda_1}{\eta} \geq \frac{r^2 L\Gamma\Lambda_1}{2\eta}. \tag{D.177}$$

Hence we can further bound equation (D.176) by

$$\sum_{i=0}^{t-1} \mathbb{E}[\|\hat{A}_i \nabla f(w_i)\|^2] \leq \frac{\eta L\Gamma\Lambda_1 t}{2} + \frac{2f_{\text{thresh}}\Lambda_1}{\eta}. \tag{D.178}$$

Now we will work toward bounding the norm of the difference $w_t - w_0$. We will first bound the difference $w_t - w_1$, then the difference $w_1 - w_0$.

$$\mathbb{E}[\|w_t - w_1\|^2] \leq \mathbb{E}\left[\left\|\sum_{i=1}^{t-1} w_{i+1} - w_i\right\|^2\right] \tag{D.179}$$

$$\leq \eta^2\, \mathbb{E}\left[\left\|\sum_{i=1}^{t-1}(\xi_i - \hat{A}_i \nabla f(w_i))\right\|^2\right], \tag{D.180}$$

where $\xi_i = \hat{A}_i(\nabla f(w_i) - g_i)$ is the zero mean effective noise that arises from rescaling the stochastic gradient noise. We may write

$$\mathbb{E}\left[\left\|\sum_{i=1}^{t-1}(\xi_i - \hat{A}_i \nabla f(w_i))\right\|^2\right] = \mathbb{E}\left[\left\|\sum_{i=1}^{t-1}\xi_i - \sum_{i=1}^{t-1}\hat{A}_i \nabla f(w_i)\right\|^2\right] \tag{D.181}$$

$$= \mathbb{E}\left[\left\|\sum_{i=1}^{t-1}\hat{A}_i \nabla f(w_i)\right\|^2 + \left\|\sum_{i=1}^{t-1}\xi_i\right\|^2 - 2\sum_{i=1}^{t-1}\sum_{j=1}^{t-1}\langle \xi_i, \hat{A}_j \nabla f(w_j)\rangle\right] \tag{D.182}$$

$$= \mathbb{E}\left[\left\|\sum_{i=1}^{t-1}\hat{A}_i \nabla f(w_i)\right\|^2\right] + \mathbb{E}\left[\left\|\sum_{i=1}^{t-1}\xi_i\right\|^2\right] \tag{D.183}$$

because ξ_i are zero mean. Since $\mathbb{E}[\xi_i^T \xi_j] = 0$ for $i \neq j$, the expression can be simplified

as:

$$\mathbb{E}\left[\left\|\sum_{i=1}^{t-1}(\xi_i - \hat{A}_i\nabla f(w_i))\right\|^2\right] = \mathbb{E}\left[\left\|\sum_{i=1}^{t-1}\hat{A}_i\nabla f(w_i)\right\|^2\right] + \sum_{i=1}^{t-1}\mathbb{E}\left[\|\xi_i\|^2\right] \tag{D.184}$$

$$\leq \mathbb{E}\left[\left\|\sum_{i=1}^{t-1}\hat{A}_i\nabla f(w_i)\right\|^2\right] + \sum_{i=1}^{t-1}\mathbb{E}\left[\|\xi_i\|^2\right] \tag{D.185}$$

$$\leq \mathbb{E}\left[\left(\sum_{i=1}^{t-1}\left\|\hat{A}_i\nabla f(w_i)\right\|\right)^2\right] + \sum_{i=1}^{t-1}\mathbb{E}\left[\|\xi_i\|^2\right] \tag{D.186}$$

$$\leq (t-1)\sum_{i=1}^{t-1}\mathbb{E}\left[\left\|\hat{A}_i\nabla f(w_i)\right\|^2\right] + \sum_{i=1}^{t-1}\mathbb{E}\left[\|\xi_i\|^2\right]. \tag{D.187}$$

Note

$$\mathbb{E}[\|\xi_i\|^2] \leq \mathbb{E}[\|\hat{A}_i\nabla f(w_i)\|^2] + \mathbb{E}[\|\hat{A}_i g_i\|^2] \tag{D.188}$$

$$\leq \mathbb{E}[\|\hat{A}_i\nabla f(w_i)\|^2] + \frac{9}{4}\Gamma \tag{D.189}$$

where we have used Lemma D.5.15. We can then bound

$$\mathbb{E}\left[\left\|\sum_{i=1}^{t-1}(\xi_i - \hat{A}_i\nabla f(w_i))\right\|^2\right] \leq (t-1+1)\sum_{i=1}^{t-1}\mathbb{E}\left[\left\|\hat{A}_i\nabla f(w_i)\right\|^2\right] + \frac{9t\Gamma}{4}. \tag{D.190}$$

Plugging in Equation (D.178) we get:

$$\mathbb{E}\left[\left\|\sum_{i=1}^{t-1}(\xi_i - \hat{A}_i\nabla f(w_i))\right\|^2\right] \leq t\left(\frac{\eta L\Gamma\Lambda_1 t}{2} + \frac{2f_{\text{thresh}}\Lambda_1}{\eta}\right) + t\Gamma. \tag{D.191}$$

Plugging this into Equation (D.180) yields:

$$\mathbb{E}[\|w_t - w_1\|^2] \leq t\eta^2\left(\frac{\eta L\Gamma\Lambda_1 t}{2} + \frac{2f_{\text{thresh}}\Lambda_1}{\eta}\right) + \eta^2\Gamma t \tag{D.192}$$

$$= \left(4\eta f_{\text{thresh}}\Lambda_1 + \eta^2\Gamma\right)t + \frac{\eta^3 L\Gamma\Lambda_1 t^2}{2}. \tag{D.193}$$

Then we may write

$$\mathbb{E}[\|w_t - w_0\|^2] \leq 2\,\mathbb{E}[\|w_t - w_1\|^2] + 2\,\mathbb{E}[\|w_1 - w_0\|^2] \tag{D.194}$$

$$\leq \left(4\eta f_{\text{thresh}}\Lambda_1 + 2\eta^2\Gamma\right) t + \eta^3 L\Gamma\Lambda_1 t^2 + 2\Gamma r^2. \tag{D.195}$$

We are almost done. By our additional assumption that $\eta \leq \frac{f_{\text{thresh}}\Lambda_1}{\Gamma}$ (which will wind up being true for small enough τ), it also follows that

$$2\eta^2\Gamma \leq 2\eta f_{\text{thresh}}\Lambda_1 \tag{D.196}$$

and therefore

$$\mathbb{E}[\|w_t - w_0\|^2] \leq 6\eta f_{\text{thresh}}\Lambda_1 t + \eta^3 L\Gamma\Lambda_1 t^2 + 2\Gamma r^2. \tag{D.197}$$

□

D.5.6 Auxiliary lemmas

Lemma D.5.12. *For $z, A, B, C \geq 0$,*

$$\sqrt{Az^2 + Bz + C} \leq \sqrt{A} \cdot \left(2z + \frac{B}{2A} + \sqrt{\frac{C}{A}}\right). \tag{D.198}$$

Proof. Note the following two facts:

$$Az^2 + Bz + C = A(z^2 + B/Az + C/A) = A[(z + B/(2A))^2 + C/A - B^2/(2A)^2] \tag{D.199}$$

and

$$Az^2 + Bz + C = A(z^2 + B/Az + C/A) = A[(z + \sqrt{C/A})^2 - 2\sqrt{C/A} + B/A]. \tag{D.200}$$

If $B^2 \geq 4AC$, then $C/A - B^2/(2A)^2 \leq 0$. Otherwise, $-2\sqrt{C/A} + B/A \leq 0$. Hence,

$$\sqrt{Az^2 + Bz + C} \leq \begin{cases} \sqrt{A} \cdot (z + B/(2A)) & \text{case 1} \\ \sqrt{A} \cdot (z + \sqrt{C/A}) & \text{case 2.} \end{cases} \tag{D.201}$$

$$\leq \sqrt{A} \cdot \left[(z + B/(2A)) + (z + \sqrt{C/A}) \right]. \tag{D.202}$$

□

Lemma D.5.13. *Let $0 < x < 1$. For $t \geq 2 \log C/x$, we have $(1+x)^t \geq C$.*

Proof. For $x < 1$ we have $\log(1+x) \leq x - x^2/2 \leq x/2$. Hence,

$$t \log(1+x) \geq tx/2 \tag{D.203}$$

$$\geq \log C, \tag{D.204}$$

and the lemma follows by exponentiating both sides. □

Series lemmas

Lemma D.5.14 (As in (Daneshmand et al., 2018))**.** *For $0 < \beta < 1$ the following inequalities hold:*

$$\sum_{i=1}^{t} (1+\beta)^{t-i} \leq 2\beta^{-1}(1+\beta)^t \tag{D.205}$$

$$\sum_{i=1}^{t} (1+\beta)^{t-i} i \leq 2\beta^{-2}(1+\beta)^t \tag{D.206}$$

$$\sum_{i=1}^{t} (1+\beta)^{t-i} i^2 \leq 6\beta^{-3}(1+\beta)^t. \tag{D.207}$$

D.5.7 Descent lemmas

First we need a quick lemma relating the constants of the true preconditioner to those of an approximate preconditioner:

Lemma D.5.15. *Let Γ be an upper bound on $\mathbb{E}[\|Ag\|^2]$. Let $\hat{A}$ be another matrix with $\|\hat{A} - A\| \le \Delta < \lambda_-/2$. Then, $\mathbb{E}[\|\hat{A}g\|^2] \le \frac{9}{4}\Gamma$.*

Proof. The proof is straightforward:

$$\mathbb{E}[\|\hat{A}g\|^2] \le \mathbb{E}[\|(A + \Delta I)g\|^2] \tag{D.208}$$

$$\le \mathbb{E}\left[\left\|\frac{3}{2}Ag\right\|^2\right] \tag{D.209}$$

$$= \frac{9}{4}\mathbb{E}[\|Ag\|^2] = \frac{9}{4}\Gamma \tag{D.210}$$

where the penultimate line follows by $\Delta < \lambda_-/2$ and $\Delta I \preceq \frac{1}{2}A_t$. □

Note that in the noiseless case $\Delta = 0$, all the below results still apply, and we only lose a constant factor compared to the typical descent lemma.

Lemma D.5.16. *Assume f has L-Lipschitz gradient. Suppose we perform the updates $w_{t+1} \leftarrow w_t - \eta\hat{A}_t g_t$, where g_t is a stochastic gradient, A_t is a $(\Lambda_1, \Lambda_2, \Gamma, \nu, \lambda_-)$-preconditioner, and $\|\hat{A}_t - A_t\| \le \Delta < \frac{\lambda_-}{2}$. Then,*

$$\mathbb{E}[f(w_{t+1})] \le f(w_t) - \frac{\eta\lambda_-}{2}\|\nabla f(w_t)\|^2 + \frac{9\eta^2 L\Gamma}{8} \tag{D.211}$$

Proof. We write

$$\mathbb{E}[f(w_{t+1})] \le f(w_t) + \langle \nabla f(w_t), \mathbb{E}[w_{t+1} - w_t]\rangle + \frac{L}{2}\mathbb{E}[\|w_{t+1} - w_t\|^2] \tag{D.212}$$

$$= f(w_t) - \eta\langle \nabla f(w_t), \hat{A}_t \nabla f(w_t)\rangle + \frac{\eta^2 L}{2}\mathbb{E}\left[\|\hat{A}_t g_t\|^2\right] \tag{D.213}$$

$$\le f(w_t) - \eta(\lambda_- - \Delta)\|\nabla f(w_t)\|^2 + \frac{9\eta^2 L\Gamma}{8} \tag{D.214}$$

$$\le f(w_t) - \frac{\eta\lambda_-}{2}\|\nabla f(w_t)\|^2 + \frac{9\eta^2 L\Gamma}{8} \tag{D.215}$$

where the third line follows by Lemma D.5.15. □

Corollary D.5.2. *Always*

$$\mathbb{E}[f(w_1)] - f(w_0) \le \frac{9\eta^2 L\Gamma}{8}. \tag{D.216}$$

Corollary D.5.3. *Suppose* $\eta \leq 4\lambda_-\|\nabla f(w_0)\|^2/(9L\Gamma)$. *Then,*

$$\mathbb{E}[f(w_1)] - f(w_0) \leq -\frac{\eta\lambda_-}{4}\|\nabla f(w_0)\|^2. \tag{D.217}$$

Corollary D.5.4. *Suppose* $\|\nabla f(w_0)\|^2 \geq \tau^2$. *Then if* $\eta \leq 4\lambda_-\tau^2/(9L\Gamma)$

$$\mathbb{E}[f(w_1)] - f(w_0) \leq -\frac{\eta\lambda_-}{4}\|\nabla f(w_0)\|^2 \leq -\frac{\eta\lambda_-}{4}\tau^2. \tag{D.218}$$

D.6 Convergence to First-Order Stationary Points

D.6.1 Generic Preconditioners: Proof of Theorem 7.4.2

Proof. Let g be the stochastic gradient at time t. We will precondition by $A_t = A(w_t)$. We write

$$\mathbb{E}[f(w_{t+1})] \leq f(w_t) + \langle \nabla f(w_t), \mathbb{E}[w_{t+1} - w_t]\rangle + \frac{L}{2}\,\mathbb{E}[\|w_{t+1} - w_t\|^2] \tag{D.219}$$

$$= f(w_t) - \eta\langle \nabla f(w_t), A_t \nabla f(w_t)\rangle + \frac{\eta^2 L}{2}\,\mathbb{E}\left[\|A_t g_t\|^2\right] \tag{D.220}$$

$$\leq f(w_t) - \eta\langle \nabla f(w_t), A_t \nabla f(w_t)\rangle + \frac{\eta^2 L\Gamma}{2} \tag{D.221}$$

$$\leq f(w_t) - \eta\lambda_{\min}(A_t)\|\nabla f(w_t)\|^2 + \frac{\eta^2 L\Gamma}{2} \tag{D.222}$$

$$\leq f(w_t) - \eta\lambda_-\|\nabla f(w_t)\|^2 + \frac{\eta^2 L\Gamma}{2}. \tag{D.223}$$

Summing and telescoping, we have

$$\mathbb{E}[f(w_T)] \leq \mathbb{E}[f(w_0)] - \eta\lambda_- \sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(w_t)\|^2\right] + \frac{\eta^2 LT\Gamma}{2}. \tag{D.224}$$

Now rearrange, and bound $f(w_T)$ by f^* to get:

$$\frac{1}{T}\cdot\lambda_-\cdot\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(w_t)\|^2\right] \leq \frac{f(w_0) - f^*}{T\eta} + \frac{\eta L\Gamma}{2}. \tag{D.225}$$

and therefore

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(w_t)\|^2\right] \leq \left(\frac{f(w_0)-f^*}{T\eta}+\frac{\eta L\Gamma}{2}\right)\cdot\frac{1}{\lambda_-}. \tag{D.226}$$

Optimally choosing $\eta = \sqrt{2(f(w_0)-f^*)/(TL\Gamma)}$ yields the overall bound

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(w_t)\|^2\right] \leq \sqrt{\frac{2(f(w_0)-f^*)L\Gamma}{T}}\cdot\frac{1}{\lambda_-}. \tag{D.227}$$

Rephrasing, in order to be guaranteed that the left hand term is bounded by τ^2, it suffices to choose T so that

$$\sqrt{\frac{2(f(w_0)-f^*)L\Gamma}{T}}\cdot\frac{1}{\lambda_-} \leq \tau^2 \tag{D.228}$$

$$\Leftrightarrow T \geq \frac{2(f(w_0)-f^*)L\Gamma}{\tau^4\lambda_-^2} \tag{D.229}$$

and

$$\eta = \sqrt{\frac{2(f(w_0)-f^*)}{TL\Gamma}} \tag{D.230}$$

$$\leq \sqrt{\frac{2(f(w_0)-f^*)}{L\Gamma}\cdot\frac{\tau^4\lambda_-^2}{2(f(w_0)-f^*)L\Gamma}} = \frac{\tau^2\lambda_-}{L\Gamma}. \tag{D.231}$$

□

D.6.2 Generic Preconditioners with Errors: Proof of Theorem 7.4.3

Proof. Let g be the stochastic gradient at time t. We will precondition by $\hat{A}_t$ which satisfies $\|\hat{A}_t - A_t\| \leq \Delta < \lambda_-/2$. We write

$$\mathbb{E}[f(w_{t+1})] \leq f(w_t) + \langle \nabla f(w_t), \mathbb{E}[w_{t+1} - w_t] \rangle + \frac{L}{2}\mathbb{E}[\|w_{t+1} - w_t\|^2] \tag{D.232}$$

$$= f(w_t) - \eta \langle \nabla f(w_t), \hat{A}_t \nabla f(w_t) \rangle + \frac{\eta^2 L}{2}\mathbb{E}\left[\|\hat{A}_t g_t\|^2\right] \tag{D.233}$$

$$\leq f(w_t) - \eta(\lambda_- - \Delta)\|\nabla f(w_t)\|^2 + \frac{\eta^2 L}{2}\mathbb{E}\left[\|(A_t + \Delta I)g_t\|^2\right] \tag{D.234}$$

$$\leq f(w_t) - \frac{\eta\lambda_-}{2}\|\nabla f(w_t)\|^2 + \frac{\eta^2 L}{2}\mathbb{E}\left[\left\|\frac{3}{2}A_t g_t\right\|^2\right] \tag{D.235}$$

$$= f(w_t) - \frac{\eta\lambda_-}{2}\|\nabla f(w_t)\|^2 + \frac{9\eta^2 L\Gamma}{8} \tag{D.236}$$

where the penultimate line follows by $\Delta < \lambda_-/2$ and $\Delta I \preceq \frac{1}{2}A_t$. Summing and telescoping, and further bounding $9/8 < 2$, we have

$$\mathbb{E}[f(w_T)] \leq \mathbb{E}[f(w_0)] - \frac{\eta\lambda_-}{2}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(w_t)\|^2\right] + 2\eta^2 L\Gamma. \tag{D.237}$$

Now rearrange, and bound $f(w_T)$ by f^* to get:

$$\frac{1}{T}\cdot\frac{\lambda_-}{2}\cdot\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(w_t)\|^2\right] \leq \frac{f(w_0) - f^*}{T\eta} + 2\eta L\Gamma \tag{D.238}$$

and therefore

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(w_t)\|^2\right] \leq \left(\frac{f(w_0) - f^*}{T\eta} + 2\eta L\Gamma\right)\frac{2}{\lambda_-}. \tag{D.239}$$

Optimally choosing $\eta = \sqrt{(f(w_0) - f^*)/(2TL\Gamma)}$ yields the overall bound

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(w_t)\|^2\right] \leq \sqrt{\frac{32(f(w_0) - f^*)L\Gamma}{T}}\cdot\frac{1}{\lambda_-}. \tag{D.240}$$

Rephrasing, in order to be guaranteed that the left hand term is bounded by τ^2, it suffices to choose T so that

$$\sqrt{\frac{32(f(w_0)-f^*)L\Gamma}{T}} \cdot \frac{1}{\lambda_-} \leq \tau^2 \tag{D.241}$$

$$\Leftrightarrow T \geq \frac{32(f(w_0)-f^*)L\Gamma}{\tau^4\lambda_-^2} \tag{D.242}$$

and

$$\eta = \sqrt{\frac{f(w_0)-f^*}{2TL\Gamma}} \leq \sqrt{\frac{(f(w_0)-f^*)\tau^4\lambda_-^2}{32(f(w_0)-f^*)L^2\Gamma^2}} = \frac{\tau^2\lambda_-}{4\sqrt{2}L\Gamma}. \tag{D.243}$$

□

D.7 Online Matrix Estimation

We first reproduce the Matrix Freedman inequality as presented by Tropp (2011):

Theorem D.7.1 (Matrix Freedman)**.** *Consider a matrix martingale* $\{Y_i : i = 0, 1, \dots\}$ *(adapted to the filtration* $\mathcal{F}_i$*) whose values are symmetric* $d \times d$ *matrices, and let* $\{Z_i : i = 1, 2, \dots\}$ *be the difference sequence, i.e.* $Z_i = Y_i - Y_{i-1}$*. For simplicity, let* $Y_0 = 0$*, so that* $Y_n = \sum_{i=1}^n Z_i$*. Assume that* $\|Z_i\| \leq R$ *almost surely for each* $i = 1, 2, \dots$*. Define* $W_i := \sum_{j=1}^i \mathbb{E}[Z_j^2|\mathcal{F}_{j-1}]$*. Then for all* $k \geq 0$*,*

$$\mathbb{P}\left(\|Y_n\| \geq k \text{ and } \|W_n\| \leq \sigma^2\right) \leq d \exp\left(\frac{-k^2/2}{\sigma^2 + Rk/3}\right).$$

Corollary D.7.1. *Let* $\{Z_i : i = 1, 2, \dots\}$ *be a martingale difference sequence (adapted to the filtration* $\mathcal{F}_i$*) whose values are symmetric* $d \times d$ *matrices. Assume* $\|Z_i\| \leq R$ *and* $\|\mathbb{E}[Z_i^2|\mathcal{F}_{i-1}]\| \leq \sigma_{\max}^2$ *for all* i*. Let* $p \in \Delta_n$ *in the simplex. Then for all* $k \leq 3\|p\|_2^2\sigma_{\max}^2/R$*,*

$$\mathbb{P}\left(\left\|\sum_{i=1}^n p_i Z_i\right\| \geq k\right) \leq d \exp\left(\frac{-k^2}{4\|p\|_2^2\sigma_{\max}^2}\right).$$

Proof. Observe that $Y_i := \sum_{j=1}^{i} p_j Z_j$ is a matrix martingale; we are trying to bound $\mathbb{P}(\|Y_n\| \geq k)$. Define the predictable quadratic variation process $W_i := \sum_{j=1}^{i} \mathbb{E}[(p_j Z_j)^2 | \mathcal{F}_{j-1}]$. By assumption, we may bound

$$\|W_n\| = \left\| \sum_{j=1}^{n} \mathbb{E}[p_j^2 Z_j^2 | \mathcal{F}_{j-1}] \right\| \leq \sum_{j=1}^{n} \left\| \mathbb{E}[p_j^2 Z_j^2 | \mathcal{F}_{j-1}] \right\| = \sum_{j=1}^{n} p_j^2 \left\| \mathbb{E}[Z_j^2 | \mathcal{F}_{j-1}] \right\| \quad \text{(D.244)}$$

$$\leq \sum_{j=1}^{n} p_j^2 \sigma_{\max}^2 = \sigma_{\max}^2 \|p\|_2^2. \quad \text{(D.245)}$$

In other words, we can deterministically bound $\|W_n\| \leq \sigma_{\max}^2 \|w\|_2^2$. Combining this bound with Theorem D.7.1, it follows that for any $k \geq 0$,

$$\mathbb{P}\left(\|Y_n\| \geq k\right) = \mathbb{P}\left(\|Y_n\| \geq k \text{ and } \|W_n\| \leq \sigma_{\max}^2 \|p\|_2^2\right) \quad \text{(D.246)}$$

$$\leq d \exp\left(\frac{-k^2/2}{\sigma_{\max}^2 \|p\|_2^2 + Rk/3}\right). \quad \text{(D.247)}$$

By assumption, $k \leq 3\|p\|_2^2 \sigma_{\max}^2 / R$, so $Rk/3 \leq \sigma_{\max}^2 \|p\|_2^2$, and we may further bound

$$d \exp\left(\frac{-k^2/2}{\sigma_{\max}^2 \|p\|_2^2 + Rk/3}\right) \leq d \exp\left(\frac{-k^2}{4\|p\|_2^2 \sigma_{\max}^2}\right).$$

□

Now we can apply the above matrix concentration results to prove Theorem 7.4.1:

Proof of Theorem 7.4.1. First we separately bound the bias and variance of the estimate $\sum_{t=1}^{T} p_t Y_t$, then use Corollary D.7.1. Since $\mathbb{E}[Y_t | \mathcal{F}_{t-1}] = G_t = G(w_t)$, the bias

of the estimate is:

$$\left\|\sum_{t=1}^{T} p_t G(w_t) - G(w_T)\right\| = \left\|\sum_{t=1}^{T} p_t (G(w_t) - G(w_T))\right\| \tag{D.248}$$

$$\leq \sum_{t=1}^{T} p_t \|G(w_t) - G(w_T)\| \tag{D.249}$$

$$\leq L \sum_{t=1}^{T} p_t \|w_t - w_T\| \tag{D.250}$$

$$\leq L \sum_{t=1}^{T} p_t \sum_{s=t+1}^{T} \|w_s - w_{s-1}\| \tag{D.251}$$

$$\leq \eta ML \sum_{t=1}^{T} p_t (T-t) \tag{D.252}$$

$$= \eta ML \cdot \frac{1}{\sum_{t=1}^{T} \beta^{T-t}} \cdot \sum_{t=1}^{T} \beta^{T-t}(T-t). \tag{D.253}$$

Note that by a well-known identity,

$$\sum_{t=1}^{T} \beta^{T-t}(T-t) = \sum_{s=0}^{T-1} s\beta^s \leq \sum_{s=0}^{\infty} s\beta^s = \frac{\beta}{(1-\beta)^2}. \tag{D.254}$$

Hence, the bias is bounded by

$$\eta ML \cdot \frac{1}{\sum_{t=1}^{T} \beta^{T-t}} \cdot \frac{\beta}{(1-\beta)^2} = \eta ML \cdot \frac{1-\beta}{1-\beta^T} \cdot \frac{\beta}{(1-\beta)^2} \tag{D.255}$$

$$= \eta ML \cdot \frac{1}{1-\beta^T} \cdot \frac{\beta}{1-\beta} \tag{D.256}$$

$$\leq ML \cdot \frac{\eta}{(1-\beta)(1-\beta^T)}. \tag{D.257}$$

Applying Corollary D.7.1 to the martingale difference sequence $Z_t = Y_t - G(w_t)$, we have that

$$\mathbb{P}\left(\left\|\sum_{t=1}^{T} p_t (Y_t - G(w_t))\right\| > k\right) \leq d \exp\left(\frac{-k^2}{4\|p\|_2^2 \sigma_{\max}^2}\right).$$

Now note that

$$\|w\|_2^2 = \sum_{t=1}^{T} p_t^2 = \frac{1}{(\sum_{t=1}^{T} \beta^{T-t})^2} \sum_{t=1}^{T} (\beta^2)^{T-t} \tag{D.258}$$

$$= \frac{(1-\beta)^2}{(1-\beta^T)^2} \sum_{t=1}^{T} (\beta^2)^{T-t} \tag{D.259}$$

$$= \frac{(1-\beta)^2}{(1-\beta^T)^2} \cdot \frac{1-\beta^{2T}}{1-\beta^2} \tag{D.260}$$

$$= \frac{1-\beta^{2T}}{(1-\beta^T)^2} \cdot \frac{(1-\beta)^2}{1-\beta^2} \tag{D.261}$$

$$= \frac{1+\beta^T}{1-\beta^T} \cdot \frac{1-\beta}{1+\beta} \tag{D.262}$$

$$\leq \frac{2(1-\beta)}{1-\beta^T}. \tag{D.263}$$

Setting the right hand side of the high probability bound to δ, we have concentration w.p. $1-\delta$ for k satisfying

$$\delta \geq d \exp\left(\frac{-k^2}{4\|p\|_2^2 \sigma_{\max}^2}\right). \tag{D.264}$$

Rearranging, we find

$$\log(d/\delta) \leq \frac{k^2}{4\|p\|_2^2 \sigma_{\max}^2} \tag{D.265}$$

$$\Leftrightarrow k \geq 2\sigma_{\max}\|p\|_2 \sqrt{\log(d/\delta)}. \tag{D.266}$$

Combining this with the triangle inequality,

$$\left\| \sum_{t=1}^{T} p_t y_t - G(w_T) \right\| \leq \left\| \sum_{t=1}^{T} p_t y_t - \sum_{t=1}^{T} p_t G(w_t) \right\| + \left\| \sum_{t=1}^{T} p_t (G(w_t) - G(w_T)) \right\| \tag{D.267}$$

$$\leq 2\sigma_{\max}\|w\|_2 \sqrt{\log(d/\delta)} + ML \cdot \frac{\eta}{(1-\beta)(1-\beta^T)} \tag{D.268}$$

$$\leq 2^{3/2}\sigma_{\max} \frac{\sqrt{1-\beta}}{\sqrt{1-\beta^T}} \sqrt{\log(d/\delta)} + ML \cdot \frac{\eta}{(1-\beta)(1-\beta^T)}. \tag{D.269}$$

with probability $1-\delta$. Since $1/\sqrt{1-\beta^T} \leq 1/(1-\beta^T)$, this can further be bounded by

$$\left(2^{3/2}\sigma_{\max}\sqrt{1-\beta}\sqrt{\log(d/\delta)} + ML \cdot \frac{\eta}{(1-\beta)}\right) \cdot \frac{1}{1-\beta^T}. \tag{D.270}$$

Write $\alpha = 1-\beta$. The inner part of the bound is optimized when

$$2^{3/2}\sigma_{\max}\sqrt{\alpha}\sqrt{\log(d/\delta)} = ML \cdot \frac{\eta}{\alpha} \tag{D.271}$$

$$\Leftrightarrow \alpha^{3/2} = \frac{ML\eta}{2^{3/2}\sigma_{\max}\sqrt{\log(d/\delta)}} \tag{D.272}$$

$$\Leftrightarrow \alpha = \frac{M^{2/3}L^{2/3}\eta^{2/3}}{2\sigma_{\max}^{2/3}(\log(d/\delta))^{1/3}} \tag{D.273}$$

for which the overall inner bound is

$$2 \cdot 2^{3/2}\sigma_{\max}\sqrt{\alpha}\sqrt{\log(d/\delta)} = 4\sigma_{\max}^{2/3}(\log(d/\delta))^{1/3}M^{1/3}L^{1/3}\eta^{1/3}. \tag{D.274}$$

If T is sufficiently large, the $1/(1-\beta^T)$ term will be less than 2. In particular,

$$T > \frac{2}{\log(1+\alpha)} \implies \frac{1}{1-(1-\alpha)^T} < 2. \tag{D.275}$$

Since $\log(1+\alpha) > \alpha/2$ for $\alpha < 1$, it suffices to have $T > 4/\alpha$. □

D.8 Converting Noise Estimates into Preconditioner Estimates

Lemma D.8.1. *Suppose $\|G - \hat{G}\| \leq \varepsilon$, i.e. $\hat{G}$ is a good estimate of G in operator norm. Assume ε is so small that $\varepsilon\|G^{-1}\| < 1/2$. Then,*

$$\|G^{-1} - \hat{G}^{-1}\| \leq \frac{\varepsilon}{2(\lambda_{\min}(G))^2}. \tag{D.276}$$

Proof. Observe

$$G^{-1}(\hat{G} - G)\hat{G}^{-1} = G^{-1} - \hat{G}^{-1}. \tag{D.277}$$

Therefore,

$$\begin{aligned} \delta = \|G^{-1} - \hat{G}^{-1}\| &= \|G^{-1}(\hat{G} - G)\hat{G}^{-1}\| && \text{(D.278)} \\ &\leq \varepsilon \|G^{-1}\| \|\hat{G}^{-1}\| && \text{(D.279)} \\ &\leq \varepsilon \|G^{-1}\| (\|G^{-1}\| + \delta). && \text{(D.280)} \end{aligned}$$

Grouping δ terms together, we find

$$\begin{aligned} (1 - \varepsilon \|G^{-1}\|)\delta &\leq \varepsilon \|G^{-1}\|^2 && \text{(D.281)} \\ \implies \delta &\leq \frac{\|G^{-1}\|^2}{1 - \varepsilon \|G^{-1}\|} \cdot \varepsilon. && \text{(D.282)} \end{aligned}$$

By assumption ε is small enough so that $\varepsilon \|G^{-1}\| < 1/2$, so overall we have

$$\delta \leq \frac{\|G^{-1}\|^2}{2} \cdot \varepsilon = \frac{1}{2(\lambda_{\min}(G))^2} \cdot \varepsilon. \tag{D.283}$$

□

Lemma D.8.2. *Suppose $\|G - \hat{G}\| \leq \varepsilon$, i.e. $\hat{G}$ is a good estimate of G in operator norm. Assume ε is so small that $\varepsilon < \frac{3}{4}\lambda_{\min}(G)$. Then,*

$$\|G^{1/2} - \hat{G}^{1/2}\| \leq \frac{\varepsilon}{(\lambda_{\min}(G))^{1/2}}. \tag{D.284}$$

Proof. We can equivalently write

$$G - \varepsilon I \preceq \hat{G} \preceq G + \varepsilon I. \tag{D.285}$$

By monotonicity of the matrix square root,

$$(G - \varepsilon I)^{1/2} \preceq \hat{G}^{1/2} \preceq (G + \varepsilon I)^{1/2} \tag{D.286}$$

and therefore

$$(G - \varepsilon I)^{1/2} - G^{1/2} \preceq \hat{G}^{1/2} - G^{1/2} \tag{D.287}$$

$$\preceq (G + \varepsilon I)^{1/2} - G^{1/2}. \tag{D.288}$$

At this point we can bound each side by applying Lemma D.8.3 to G and to $G - \varepsilon I$. The result is the bound

$$\frac{-\varepsilon}{2(\lambda_{\min}(G) - \varepsilon)^{1/2}} \preceq \hat{G}^{1/2} - G^{1/2} \preceq \frac{\varepsilon}{2(\lambda_{\min}(G))^{1/2}}.$$

The lower bound is looser, so the operator norm of the difference is bounded by

$$\frac{\varepsilon}{2(\lambda_{\min}(G) - \varepsilon)^{1/2}} < \frac{\varepsilon}{2(\frac{1}{4}\lambda_{\min}(G))^{1/2}} = \frac{\varepsilon}{(\lambda_{\min}(G))^{1/2}}.$$

□

Lemma D.8.3. *Let $A \succ 0$ and $\varepsilon > 0$. Then*

$$\|(A + \varepsilon)^{1/2} - A^{1/2}\| \leq \frac{\varepsilon}{2(\lambda_{\min}(A))^{1/2}}. \tag{D.289}$$

Proof. The bound reduces to plugging in the eigenvalues of A to a scalar function $f : \mathbb{R} \to \mathbb{R}$. Define $f(x) = (x + \varepsilon)^{1/2} - x^{1/2}$. Note that

$$f(x) = \frac{((x + \varepsilon)^{1/2} - x^{1/2})((x + \varepsilon)^{1/2} + x^{1/2})}{(x + \varepsilon)^{1/2} + x^{1/2}} \tag{D.290}$$

$$= \frac{(x + \varepsilon) - x}{(x + \varepsilon)^{1/2} + x^{1/2}} \tag{D.291}$$

$$= \frac{\varepsilon}{(x + \varepsilon)^{1/2} + x^{1/2}} \tag{D.292}$$

$$\leq \frac{\varepsilon}{2x^{1/2}}, \tag{D.293}$$

from which the result follows. □

Corollary D.8.1. *Suppose* $\|G - \hat{G}\| \leq \varepsilon$*, for small enough* ε*. Then,*

$$\|(G + \delta I)^{-1/2} - (\hat{G} + \delta I)^{-1/2}\| \leq \frac{\varepsilon}{2(\delta + \lambda_{\min}(G))^{3/2}}.$$

Proof. Simply apply Lemma D.8.1 and Lemma D.8.2 to $G + \delta I$. □