Compression for Distributed Optimization and Timely Updates

A Thesis

Submitted for the Degree of

Doctor of Philosophy

in the **Faculty of Engineering**

by

Prathamesh Mayekar

under the Guidance of

Himanshu Tyagi

Electrical Communication Engineering
Indian Institute of Science
Bangalore – 560 012, INDIA

April 2022

TO

Aai, Baba, and Kshitij

Acknowledgments

This dissertation was made possible by a fellowship by the Ministry of Human Resource Development, Gov. of India during my first two academic years at IISc, and a fellowship by Wipro Ltd. during the subsequent three academic years at IISc. I would also like to acknowledge Robert Bosch Centre for Cyber-Physical Systems, IISc and SPCOM, 2018 travel grant for the travel support to the International Symposium on Information Theory (ISIT), 2018.

I consider myself fortunate to have Prof. Himanshu Tyagi as my advisor. He has been extremely generous with his time and ideas and often held my hand through most aspects of academic research, such as reading and writing papers, proving theorems, and preparing research presentations. For instance, before ISIT 2018, he sat through multiple practice talks and gave valuable feedback on each one of them. Moreover, he stayed with me during all conference submissions until we submitted the paper; this was often way past midnight. Without his support and guidance, perhaps this thesis would not have been possible.

I am thankful to my collaborators Prof. Jayadev Acharya, Prof. Clément Cannone, Prof. Parimal Parag, and Dr. Ananda Theetha Suresh for their guidance throughout our collaborations. I would also like to thank professors at IISc and IITB for their excellent teaching. For instance, at IISc, I thoroughly enjoyed the Information Theory courses by Prof. Himanshu Tyagi, the Optimization reading group led by Prof. Aditya Goplan's research group, the Probability Theory course by Prof. Srikanth Iyer, and the Foundations of Data Science course by Prof. Siddharth Barman. At IITB, I thoroughly enjoyed the Game Theory course by Prof. Ankur Kulkarni, the Integer Linear Programming course by Ashutosh Mahajan, and the Stochastic Processes course by Prof. Veeraruna Kavitha. The

courses at IITB motivated my decision to pursue a research career, and the courses at IISc became instrumental in my subsequent research work. I would also like to thank Prof. Nandyala Hemachandra and Prof. Veeraruna Kavitha at IEOR, IITB, who encouraged me to pursue a career in academia.

My interactions with fellow graduate students enriched my Ph.D. experience at IISc. I would like to thank Raghava for delving with me into the nuances of probability and information theory. I would like to thank Sanidhay for guiding me through the ups and downs of life at IISc in my early graduate years. I would like to thank Karthik for teaching me how to teach. I want to thank Saumya for sharing with me "tangocolors.sty", a customized tex color style file that I still use for all my presentations. Thanks are also due to my labmates Aditya, Mishfad, Raghava, Shubham, Siddharth, Sahasranand, Saumya, and Vaishali whose presence in the lab made my work a lot more enjoyable. I am already missing those coffee-break conversations.

My IEOR batchmates made my stay in Bangalore a pleasant one. The weekends spent at Anupam's place made the highs of graduate life even more enjoyable and the lows bearable.

I would like to thank my parents for their unconditional support, love, and encouragement throughout my graduate studies. Without their reassuring presence and many sacrifices, both my undergraduate and graduate education would not have been possible. I can never thank them enough! I would like to thank my younger brother Kshitij for patiently listening to all my ramblings and rants about graduate student life, guiding me through most aspects of life outside academia, and, in essence, always being the older, wiser one of the two of us.

Statement of Originality

I hereby declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of higher education.

I certify that to the best of my knowledge, the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Publications based on this Thesis

Journal Publications:

1. **P Mayekar** and H Tyagi. "RATQ: A Universal Fixed-Length Quantizer for Stochastic Optimization", IEEE Transactions on Information Theory 67(5) (pp. 3130 - 3154).

2. **P Mayekar**, P Parag, and H Tyagi. "Optimal source codes for timely updates", IEEE Transactions on Information Theory 66(6) (pp. 3714-3731).

Conference Proceedings:

1. J. Acharya, C. Canonne, **P. Mayekar**[1], and H. Tyagi. "Information-constrained optimization: Can adaptive processing of gradients help?", Advances in Neural Information Processing Systems, 2021.

2. **P Mayekar**, A. T. Suresh, and H. Tyagi. "Wyner-Ziv Estimators: Efficient Distributed Mean Estimation with Side Information", International Conference on Artificial Intelligence and Statistics (pp. 3502-3510). PMLR, 2021.

3. **P Mayekar** and H Tyagi. "Limits on gradient compression for stochastic optimization". IEEE International Symposium on Information Theory (pp. 2658-2663). IEEE, 2020.

[1] Alphabetical Order used for author list.

4. **P Mayekar** and H Tyagi. "RATQ: A universal fixed-length quantizer for stochastic optimization". International Conference on Artificial Intelligence and Statistics (pp. 1399-1409). PMLR, 2020.

5. **P Mayekar**, P Parag, and H Tyagi. "Optimal lossless source codes for timely updates". 2018 IEEE International Symposium on Information Theory (ISIT). IEEE, 2018. (Recipient of the Jack Keil Wolf Student Paper Award.)

Abstract

The goal of this thesis is to study the compression problems arising in distributed computing systematically.

In the first part of the thesis, we study gradient compression for distributed first-order optimization. We begin by establishing information theoretic lower bounds on optimization accuracy when only finite precision gradients are used. Also, we develop fast quantizers for gradient compression, which, when used with standard first-order optimization algorithms, match the aforementioned lower bounds.

In the second part of the thesis, we study distributed mean estimation, an important primitive for distributed optimization algorithms. We develop efficient estimators which improve over state of the art by efficiently using the side-information present at the center. We also revisit the Gaussian rate-distortion problem and develop efficient quantizers for this problem in both the side-information and the no side-information setting.

Finally, we study the problem of entropic compression of the symbols transmitted by the edge devices to the center, which commonly arise in cyber-physical systems. Our goal is to design entropic compression schemes that allow the information to be transmitted in a 'timely' manner, which, in turn, enables the center to have access to the latest information for computation. We shed light on the structure of the optimal entropic compression scheme and, using this structure, we develop efficient algorithms to compute this optimal compression scheme.

Contents

Notation

1. Sets
   (a) $\mathbb{R}^d$ is the set of d dimensional vectors, where each coordinate can take any value on the real line.
   (b) $\mathbb{Z}$ is the set of integers.
   (c) $\mathbb{N}$ is the set of positive integers.
   (d) $[n] := \{1, \ldots, n\}$ is the set of number from 1 to n.
   (e) $|\mathcal{X}|$ is the cardinality of a discrete set $\mathcal{X}$.
   (f) $\{e_1, \ldots, e_d\}$ is the Euclidean basis of $\mathbb{R}^d$, where e_i is a d-dimensional vectors with i coordinate equal to 1 and rest of the coordinates equal to 0.
2. Random Variables and Events
   (a) pmf is Probability mass function.
   (b) pdf is Probability density function.
   (c) *iid* is Independent and identically distributed.
   (d) $P(A)$ is Probability of event A.
   (e) $\mathbb{E}[Z]$ is Expectation of the random variable Z.
3. Norms
   (a) $\|x\|_p := \sum_{i\in[d]}(|x(i)|^p)^{1/p}$ is the ℓ_p-norm of $x \in \mathbb{R}^d$.
   (b) $\|X\|_p = \mathbb{E}\left[|X|^p\right]^{1/p}$ is the L_p norm of a random variable X.

(c) Throughout the paper, q denotes the Hölder conjugate of p (that is, $\frac{1}{p} + \frac{1}{q} = 1$).

4. Logarithms

(a) The logarithm to the base 2 is denoted by $\log a$ and the logarithm to the base e is denoted by $\ln a$. All the information theoretic measures considered in this paper – such as Entropy, Rényi divergence, Kullback-Leibler divergence, and Mutual Information – are defined with logarithm to the base 2.

(b) The iterated logarithms $\log^*(a)$ and $\ln^*(a)$ are defined as the number of times log and ln must be iteratively applied to a before the result is at most 1.

5. Maximum and minimum

(a) We write $a \vee b$ and $a \wedge b$ for $\max\{a, b\}$ and $\min\{a, b\}$, respectively.

Chapter 1

Overview

The recent years have witnessed a monumental rise in the data available for machine learning applications. For instance, ImageNet ([21]), a publicly available image database, has over fourteen million images, all available to train machine learning models. Closely mimicking the data rise is the aspiration to build more capable and accurate machine learning models. This, in turn, has led to the ever-increasing computing power needed to train sophisticated deep learning models. For instance, the ResNet ([41]) architectures trained on the ImageNet database can have roughly 20-60 million trainable parameters. One approach to ensure fast training of such sophisticated models is to employ distributed optimization methods, where at each iteration, workers *quantize* and share their updates, stochastic gradients, with other workers or the center. However, while this distributed optimization approach has enjoyed popularity recently, the precise effect quantization has on convergence rates is not entirely understood.

A related problem is that of federated learning ([49]). Federated learning is a machine learning paradigm where models are built from decentralized data residing on mobile devices while preserving the privacy of the data. A few of the devices share their stochastic gradient updates with the center in a typical iteration of a federated learning algorithm. In low bandwidth scenarios, efficiently quantizing these updates becomes crucial. Thus, understanding the tradeoff between the precision to which gradients are quantized and the convergence rate becomes crucial in designing efficient federated learning algorithms.

Finally, the problem of timely dissemination of information has become increasingly important in modern cyber-physical systems. For instance, consider the problem of control of the network of autonomous vehicles. In such a setting timely update of the vehicle state becomes exceptionally crucial. In such problems, designing compression specifically tailored to the application of timeliness is crucial.

Keeping in mind the applications listed above, the thesis considers the following three problems:

1. Communication-Constrained First-Order Optimization,
2. Efficient Quantization for Federated Learning Primitives,
3. Source Coding Schemes for Timeliness.

The first two parts of the thesis are dedicated to studying the distributed optimization scenarios listed above. In the first part of the thesis, we build a theory for a distributed optimization setting where the stochastic gradients are quantized to a given precision. The quantization algorithms we develop in this part improve over the state-of-the-art algorithms in many settings. In the second part of the thesis, we revisit primitives often used Federated Learning: 1) Distributed Mean Estimation 2) Gaussian Quantization. We build communication-efficient primitives for both these problems. In the final part of the thesis, we design entropic compression schemes to ensure timely update of information.

1.1 Communication-Constrained First-Order Optimization

In the first part of the thesis, we study a refinement of the classic query complexity model of Nemirovsky and Yudin ([71]). In our refinement, the gradient estimates supplied by the first-order oracle are not directly available to a first-order optimization algorithm but must pass through a channel, and only the output of the channel is available to the optimization algorithm. While we introduced this refinement to study the effect of communication constraints on convergence rate, the channel can also be used to model various other

information constraints such as local differential privacy constraints and computational constraints.

In Chapter 2, we derive lower bounds on the optimization error of any first-order optimization algorithm where it only has access to compressed gradients. In Theorem 2.4.4 and 2.4.5, we show that for the optimization of convex and ℓ_p lipschitz family, any optimization algorithm using gradients compressed to r bits would lead to the following blow-up over the classic convergence rate: for $p \in [1, 2)$, we see a blow-up of $\sqrt{\frac{d}{\min\{d, r\}}}$; for $p \in [2, \infty]$, we see a blow-up of $\left(\sqrt{\frac{d}{d \wedge 2^r}}\right) \vee \left(\sqrt{\frac{d^{2/p}}{d \wedge r}}\right)$, where d is the ambient dimension. Our lower bounds also extend to the class of strongly convex and ℓ_2 lipschitz function, which were missing in the literature of information-constrained optimization. For example, in Theorem 2.4.6, we show that for the optimization of strongly convex and ℓ_2 lipschitz function, any optimization algorithm using gradients compressed to r bits would lead to a blow-up of at least $\frac{d}{\min\{d,r\}}$ over the classic convergence rate.

In Chapter 2, we also derive lower bounds for local differential privacy constraints and computational constraints in Theorems 2.4.1, 2.4.2, and Theorems 2.4.3 and 2.4.7 and 2.4.8, respectively. In fact, our lower bounds allows us to establish optimality of the popular random coordinate descent algorithm for convex and ℓ_2 lipschitz family, when there is a computational constraint of computing just one coordinate.

Finally, all the lower-bounds derived in Chapter 2 allow for adaptive processing of gradients, while the previous literature on information-constrained optimization restricts to non-adaptive protocols. That is, the channel used to process the gradients at a given iteration can be chosen as a function of the information received at the previous iteration. However, as we see in the compressing schemes derived in the next chapters and privacy protocols employed in the literature, the non-adaptive processing of gradients is sufficient to match the lower bounds.

Our proof of lower bounds refines the recipe of [5] and reduces the problem of lower bounding optimization error to that of upper bounding a mutual information term. The mutual information term then can be bounded by taking recourse to recent strong data processing inequalities in [3]. The key observation in our proof is that we only need to

bound a coordinate-wise average mutual information term compared to the larger total mutual information considered in [5]. This allows our recipe to be applicable in settings where the recipe in [5] may not be suitable.

In Chapter 3, we focus on developing optimal quantizers to match lower bounds derived in Chapter 2 for convex and ℓ_2 lipschitz functions and strongly convex and ℓ_2 lipschitz functions. Since we assume that the gradient estimates' have their Euclidean distance almost surely bounded, this problem essentially reduces to developing efficient quantizers for input vector Y such that

$$Y \leq B^2 \quad a.s..$$

Our main contribution in this chapter is a quantizer RATQ used to quantize such input vectors Y. In Corollary 3.4.3, we show that employing RATQ to compress the gradients along with the optimization algorithm projected stochastic gradient descent (PSGD) requires precision of $d \log\log\log\log^* d$ bits to attain the convergence rate of the classic, unrestricted setting for convex, or strongly convex, and ℓ_2 lipschitz function family. This factor differs by only a minuscule $\log\log\log\log^* d$ from the lower bounds of $\Omega(d)$ on the precision necessary to attain the convergence of classic setting. Moreover, in Corollary 3.4.5, we show that employing a subsampled version of RATQ along with PSGD leads to almost optimal convergence rate, thus matching the lower bounds established in Chapter 2 for both convex and strongly convex functions, which are also ℓ_2 lipschitz.

In fact, our quantizer RATQ is part of a general family of quantizers called adaptive quantizers. An adaptive quantizer uses multiple dynamic ranges, $\{[-M_i, M_i] : i \in [h]\}$, to quantize the input. Once a dynamic range is chosen, the input is quantized uniformly within it using k uniform level. In designing RATQ, we stumble upon the following formula for the mean square error of the adaptive quantizer:

$$O\left(\frac{\sum_{i\in[h]} M_i^2.p(M_{i-1})}{(k-1)^2}\right),$$

where $p(M)$ is the probability of the input vector exceeding the value M. This formula guides the design of all of our subsequent adaptive quantizers.

Before adaptively quantizing an input, RATQ first preprocesses the input by randomly rotating it. Random rotation allows the input data to be "evenly" distributed across all the coordinates and gives us a handle over the data distribution. This classic idea of Random rotation is also crucial to many of our subsequent quantizers.

Then in Chapter 3, we relax the almost sure assumption on gradient estimates noise and study a general noise model for noisy gradient estimate where the expected Euclidean norm square of the estimates' output is bounded, termed the *mean square noise model*. We show the theoretical limitations imposed by quantizers for such noise models that do not quantize the norm carefully. We do this by deriving a lower bound on optimization error in Theorems 3.5.3 and 3.5.4, when a popular class of quantizers that quantize the gradient norm uniformly is employed. Our lower bound relies on a novel heavy-tailed construction which may be of independent interest. We then present a fixed-length, *gain-shape* variant of our quantizer RATQ, termed A-RATQ. In A-RATQ, the norm (*the gain*) of the input is quantized by employing another adaptive quantizer and the input normalized by the norm (*the shape*) is quantized by employing RATQ. In Corollaries 3.5.7 and 3.5.9, we show that A-RATQ along with PSGD almost matches the best possible performance for this mean square noise model. Finally, we present a variable-length update to A-RATQ. This variable-length version further improves the performance of A-RATQ but only satisfies the precision constraint in expectation.

In Chapter 4, we develop quantizers to match the lower bounds for communication-constrained optimization of convex and ℓ_p lipschitz family. Our results in this Chapter are primarily restricted to *the high-precision regime*. That is, we characterize the minimum precision needed by optimal quantization and optimization algorithms so that optimization with compressed gradient achieves the convergence of the classic, unrestricted setting. In Theorem 4.4.1, we show that for optimization of convex and ℓ_p lipschitz family with compressed gradients, if the gradients are compressed to a precision of $d^{2/p} \vee \log d$, for $p \in [2, \infty]$, and d, for $p \in [1, 2)$, we can attain the convergence rate of the classic, unrestricted setting. The necessity of this precision follows from the lower bound on optimization error derived in Chapter 2; for sufficiency, we construct new quantizers. For

$p \in [2, \infty]$, we propose a Quantizer SimQ$^+$ which along with PSGD exactly matches the lower bounds for $p = 2$ and $p = \infty$, and is only a logarithmic factor away from the lower bound for $p \in (2, \infty)$. Interestingly, for $p = \infty$, compressing the gradients to only $\log d$ bits using SimQ$^+$ and then employing PSGD with compressed gradients is sufficient to achieve the convergence of the classic case. Thus, this improves upon the precision required by RATQ and PSGD, $d \log \log \log \log^* d$, in the high-precision regime. For $p \in [1, 2)$, we propose another variant of RATQ, which, combined with appropriate mirror descent algorithms, is almost optimal.

In its simplest form, SimQ$^+$ represents the input in terms of the corner points of the ℓ_1 ball containing it. To achieve further compression, SimQ$^+$ uses a "type" based compression technique. We will use this type-based compression idea in some of our later schemes, too.

1.2 Efficient Quantization for Federated Learning Primitives

In Chapter 5, we study the primitive of distributed mean estimation. This primitive is a crucial subroutine in distributed learning scenarios when the server uses the average of updates from multiple clients. [88] considered a version of this problem where n clients communicate a quantized version of their update to the server, where the total precision of the quantized version can be at the most r bits. The center uses the quantized versions from all the clients to estimate the sample mean of the data. A lower bound of $\frac{d}{nr}$ was established on the mean square error (MSE) between the actual sample mean and the estimated sample mean. [88] also proposed a quantization procedure that matches this lower bound up to $\log \log d$ factor. In Theorem 5.4.1, we derive the best known upper bound on MSE, which is tight up to a $\log \ln^* d$ factor from the lower bound. Our scheme uses the quantizer RATQ from Chapter 3.

Then, motivated by the fact that in many federated learning scenarios, the server also has access to some side-information, we propose and study distributed mean estimation where the server also has access to side-information. We study this problem in two different

settings: 1) the distance between the update and the side-information is known to the clients and the server; 2) the distance between the update and the side-information is unknown to all, the universal setting. For the first setting, we propose a quantizer RMQ and show in Theorem 5.5.4 that it results in an overall MSE of roughly $\frac{d\Delta^2}{nr}$, where Δ^2 is the distance between the client's update and the respective side-information at the server. Thus, we can break the lower bound of no side-information setting using side-information, as long as $\Delta \leq 1$. Our quantizer RMQ first preprocesses the update using Random rotation like RATQ and then uses a modulo quantizer for each coordinate. Coming to the unknown setting, we propose a quantizer RDAQ and show in Theorem 5.6.4 that results in an overall MSE of roughly $\frac{d\Delta}{nr}$. Thus, we can again break the lower bound of the no side-information case with accurate side-information. Our quantizer RDAQ first preprocesses the updates by randomly rotating them and then uses the idea of correlated sampling to provide MSE bounds dependent on the distance without the knowledge of the distance.

In Chapter 6, we the revisit the Gaussian rate-distortion problem and show that the quantization schemes from earlier Chapters are almost optimal while being efficient for this problem. In Theorem 6.3.1, we show that a subroutine of RATQ attains a rate very-close to the Gaussian rate-distortion function while being computationally feasible relative to the optimal coding scheme. In Theorem 6.4.1, we show that the simple modulo quantizer achieves a rate close to the rate-distortion function for the version of Gaussian rate-distortion problem with side–information.

1.3 Source Coding for Timeliness

In the final part of the thesis, we study a source coding problem to facilitate the timely dissemination of information. This study focuses on communication systems where the time to transmit information is directly proportional to its code length, and the receiver needs to be apprised about only the latest information. Based on the *age of information* metric proposed in [50], we measure the performance of our schemes by the average age of information. For information received at time t which was generated at time $U(t) \leq t$, the

age of information at the receiver is $t - U(t)$. Our goal is to come up with coding schemes which minimize the average age

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} t - u(t).$$

In Theorem 7.3.2, we show that the average age equals

$$\mathbb{E}[L] + \frac{\mathbb{E}[L^2]}{2\mathbb{E}[L]} - \frac{1}{2}.$$

Our proof relies on the modification of the standard renewal reward theorem. We then show in Example 7.3.5 that standard prefix-free coding schemes such as Shannon codes can be suboptimal by as far as $O(\log|\mathcal{X}|)$ for these problems, where $|\mathcal{X}|$ is the cardinality of the information. Our main result is Theorem 7.5.1, where we show that the optimal source coding scheme for minimizing average age is Shannon coded corresponding to distribution, which is a tilting of the original distribution. Our proof relies on linearizing the average cost, which, in turn, relies on a variational formula for L_p norm of a random variable.

We then extend our recipe of linearizing the cost and identifying the structure of optimal coding schemes to design source coding schemes to minimize the average delay. In Theorem 7.7.4, we show that the optimal source coding scheme here, too, is a Shannon code for the tilting of the original distribution.

Part I

Communication-Constrained First-Order Optimization

Chapter 2

Lower Bounds for Information-Constrained Optimization

2.1 Synopsis

We revisit first-order optimization under local information constraints such as communication, local privacy, and computational constraints limiting access to a few coordinates of the gradient. In this setting, the optimization algorithm is not allowed to directly access the complete output of the gradient oracle, but only gets limited information about it subject to the local information constraints. We consider optimization for both convex and strongly convex functions and obtain tight or nearly tight lower bounds for the convergence rate under all three information constraints.

The results presented in this chapter are from [2].

2.2 Introduction

Distributed optimization has emerged as a central tool in federated learning for building statistical and machine learning models for distributed data. In addition, large scale

optimization is typically implemented in a distributed fashion over multiple machines or multiple cores within the same machine. These distributed implementations fit naturally in the oracle framework of first-order optimization (see [71]) where in each iteration a user or machine computes the gradient oracle output. Due to practical local constraints such as communication bandwidth, privacy concerns, or computational issues, the entire gradient cannot be made available to the optimization algorithm. Instead, the gradients must be passed through a mechanism which, respectively, ensures privacy of user data (local privacy constraints); or compresses them to a small number of bits (communication constraints); or only computes a few coordinates of the gradient (computational constraints). Motivated by these applications, in this chapter, we derive lower bounds on first-order optimization under such constraints. While our focus in the rest of the chapters would be to achieve these lower bounds for communication constraints.

When designing a first-order optimization algorithm under local information constraints, one not only needs to design the optimization algorithm itself, but also the algorithm for local processing of the gradient estimates. Many such algorithms have been proposed in recent years; see, for instance, [24], [1], [6], [33], [86], [37], and the references therein for privacy constraints; [83], [9], [88], [52], [28], [78], [58], [4], [17], [46], [82], [35], [38] and the references therein for communication constraints; [73, 80] for computational constraints. However, these algorithms primarily consider *nonadaptive* procedures for gradient processing (with the exception of [28]): that is, the scheme used to process the gradients at any iteration cannot depend on the information gleaned from previous iterations. In this chapter, we derive lower bounds for optimization under a much larger class of *adaptive* gradient processing protocols. As a result, we answer the following open question in this part of the thesis.

Can adaptively processing gradients improve convergence in information-constrained optimization?

For optimization of both convex and strongly convex function families and under the three different local constraints mentioned above – local privacy, communication, and computational – we answer this question in the negative.

That is, adaptive processing of gradients has no clear advantage over non-adaptive processing for convex or strongly convex optimization under information-constraints.

2.2.1 Main contributions

We model the information constraints using a family of channels $\mathcal{W}$; see Section 2.3.3 for a description of the channel families corresponding to our constraints of interest. We consider first-order optimization where the output of the gradient oracle must be passed through a channel W selected from $\mathcal{W}$. Specifically, the gradient is sent as input to this channel W, and the algorithm receives the output of the channel. In each iteration of the algorithm, the channel to be used in that iteration can be selected adaptively based on previously received channel outputs by the algorithm; or channels to be used throughout can be fixed upfront, nonadaptively. The detailed problem setup is given in Section 2.3.1. We obtain general lower bounds for optimization of convex and strongly convex functions using $\mathcal{W}$, when adaptivity is allowed. These bounds are then applied to the specific constraints of interest to obtain our main results.

In terms of overall contribution of this part of this thesis, we show that adaptive gradient processing does not help for some of the most typical first-order optimization problems under information constraints. Namely, we prove that for most regimes of local privacy, communication, or computational constraints, adaptive gradient processing has nearly the same convergence rate as nonadaptive gradient processing for both convex and strongly convex function families. As a consequence, this shows that the nondaptive LDP algorithms from [24] and nonadaptive compression protocols we develop in Chapters 3 and 4 are (nearly) optimal for private and communication-constrained optimization, respectively, even if adaptive gradient processing is allowed. In another direction, under computational constraints, where we are allowed to compute only one gradient coordinate, we show that standard Random Coordinate Descent (*cf.* [15, Section 6.4]), which employs uniform (nonadaptive) sampling of gradient coordinates, is optimal for both the convex and strongly convex function families. This proves that adaptive sampling of gradient coordinates does not improve over nonadaptive sampling strategies.

2.2.2 Remarks on techniques

Without information constraints, [5] provides a general recipe for proving oracle complexity lower bounds for convex optimization. Specifically, it reduces optimization problems with a first-order oracle to a mean estimation problem whose probability of error is lower bounded using Fano's method (*cf.* [95]). While our work, too, relies on a reduction to mean estimation, we deviate from the prior approach, using Assouad's method instead to prove lower bounds for various function families. This different approach, in turn, enables us to derive lower bounds for adaptive processing of gradients. We then combine our Assouad's type reduction with upper bounds on mutual information derived in [3], which crucially hold for adaptive protocols.

We note that the prior work in information-constrained optimization – primarily, locally private optimization – concerned itself with the family of convex functions, with no lower bounds known for the more restricted family of *strongly convex* functions, even for nonadaptive gradient processing protocols. The key obstacle is the fact that during the reduction from optimization to mean estimation, the known hard instance for the strongly convex family, even when analyzed for nonadaptive protocols, leads to an estimation problem using adaptive protocols; and thus the lack of known lower bounds for adaptive information-constrained estimation prevented this approach from succeeding. In more detail, this hard instance has gradients that can depend on the query point which in turn can be chosen based on previously observed channel outputs, an issue which does not arise in the case of the convex family where the lower bounds are derived using affine functions for which the gradients do not depend on the query point. We manage to circumvent this issue by relying on our different Assouad-type reduction.

2.2.3 Prior work

The framework we consider can be viewed as an extension of the classical query complexity model in [71]. We refer the reader to textbooks and monographs [15, 70, 73] for a review of the basic setup. In the information-constrained setting, motivated by privacy concerns, [24] consider the problem where the gradient estimates must pass through a locally differentially

private (LDP) channel. However, in their setting the LDP channels for all time steps are selected at the start of optimization algorithm – in other words, the channel selection strategy is nonadaptive. In contrast, we allow for *adaptive* channel selection strategies (as well as other information constraints); as a result, the lower bounds established in these papers do not apply to our setting, and are more restrictive than our bounds. The results of Duchi and Rogers [26] for Bernoulli product distributions could be combined with our construction to obtain tight lower bounds for optimization in $p \in [1, 2]$ under LDP constraints, but would not extend to the entire range of p. The work of Braverman, Garg, Ma, Nguyen, and Woodruff [14] on communication constraints, also for $p \in [1, 2]$, is relevant as well; however, their bounds on mutual information cannot be applied directly, as their setting (Gaussian distributions) would not satisfy our almost sure gradient oracle assumption. [28] provide adaptive quantization schemes for convex and ℓ_2 Lipschitz function family. While the worst-case convergence guarantees for the quantizers in [28] are similar to those in [9], it shows some practical improvements over the state-of-the-art for some specific problem instances. This suggests that while adaptive quantization may not help in the worst case for non-smooth convex and strongly convex optimization, it may be useful for a smaller subclass of convex optimization problems.

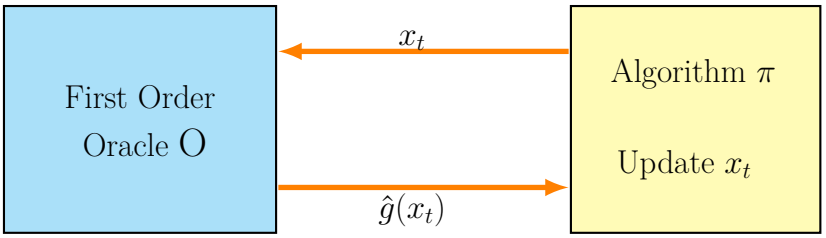
Organization

The rest of the chapter is organized as follows. After formally introducing in Section 2.3 the setting, the function classes considered (convex and strongly convex), and the information constraints we are concerned with, we state and discuss our lower bounds in Section 2.4. Proofs of these lower bounds are given in Section 2.5.

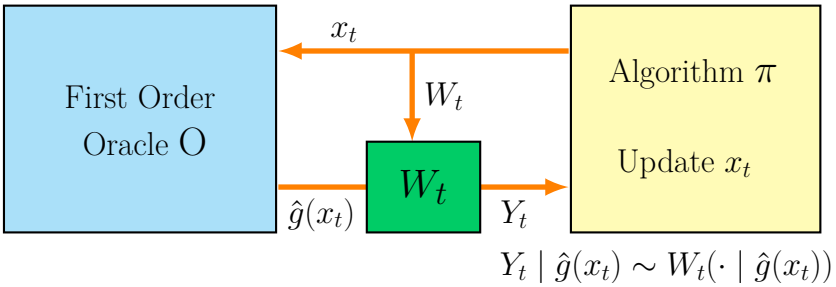
2.3 Setup and preliminaries

2.3.1 Optimization under information constraints

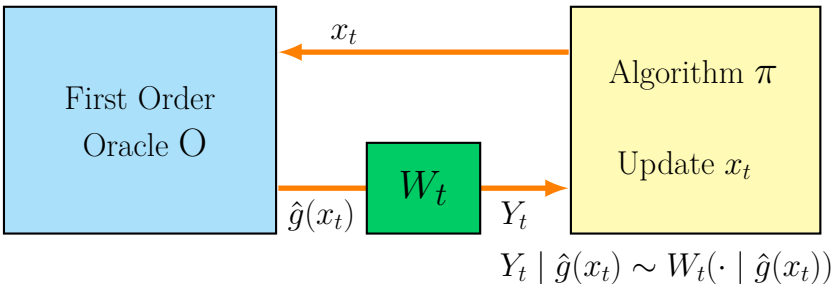
We consider the problem of minimizing an unknown convex function $f\colon \mathcal{X} \to \mathbb{R}$ over its domain $\mathcal{X}$ using *oracle access* to noisy subgradients of the function. That is, the algorithm



(a) Classical first-order optimization



(b) Information-constrained optimization with adaptive gradient processing.



(c) Information-constrained optimization with nonadaptive gradient processing.

is not directly given access to the function but can get subgradients of the function at different points of its choice. This class of optimization algorithms includes various descent algorithms, which often provide optimal convergence rate among all the algorithms in this class (*cf.* [71]).

In our setup, gradient estimates supplied by the oracle must pass through a channel W,[1] chosen by the algorithm from a fixed set of channels $\mathcal{W}$, and the optimization algorithm π only has access to the output of this channel. The *channel family* $\mathcal{W}$ represents information constraints imposed in our distributed setting. In detail, the framework is as follows:

1. At iteration t, the first-order optimization algorithm π makes a query for point x_t to the oracle O.

2. Upon receiving the point x_t, the oracle outputs $\hat{g}(x_t)$, where $\mathbb{E}\left[\hat{g}(x_t) \mid x_t\right] \in \partial f(x_t)$ and $\partial f(x_t)$ is the subgradient set of function f at x_t.

3. The subgradient estimate $\hat{g}(x_t)$ is passed through a channel $W_t \in \mathcal{W}$ and the output Y_t is observed by the first-order optimization algorithm. The algorithm then uses all the messages $\{Y_i\}_{i\in[t]}$ to further update x_t to x_{t+1}.

Let Π_T be the set of all first-order optimization algorithms that are allowed T queries to the oracle O and after the tth query gets back the output Y_t with distribution $W_t(\cdot \mid \hat{g}(x_t))$.

Our goal is to select gradient processing channels W_ts and an optimization algorithm π to guarantee a small worst-case optimization error. Two classes of *channel selection strategies* are of interest: *adaptive* and *nonadaptive*.

Definition 2.3.1. Under *adaptive gradient processing*, the channel W_t selected at time t may depend on the previous outputs of channels $\{W_i\}_{i\in[t-1]}$. Specifically, denoting by Y_t the output of the channel used at time t, which takes values in the output alphabet $\mathcal{Y}_t$, the *adaptive channel selection strategy* $S := (S_1, \ldots, S_T)$ over T iterations consists of

[1] A channel W with input alphabet $\mathcal{X}$ and output alphabet $\mathcal{Y}$, denoted $W\colon \mathcal{X} \to \mathcal{Y}$, represents the conditional distribution of the output of a randomized function given its input. In particular, $W(\cdot \mid x)$ is the conditional distribution of the channel given that the input is $x \in \mathcal{X}$.

mappings $S_t \colon \mathcal{Y}^{t-1} \to \mathcal{W}$ that take $Y_1, \dots Y_{t-1}$ as input and output a channel $W_t \in \mathcal{W}$ as output. We write $\mathcal{S}_{\mathcal{W},T}$ for the collection of all such channel selection strategies.

Definition 2.3.2. Under *nonadaptive gradient processing* all the channels $\{W_t\}_{t\in[T]}$ through which the gradient estimates must pass are decided at the start of the optimization algorithm. In other words, conditioned on the shared randomness, the channel W_t is selected independently of all the gradient observations received by the optimization algorithm until step t. Denote the class of all nonadaptive strategies by $\mathcal{S}^{\mathrm{NA}}_{\mathcal{W},T}$.

Figures 2.1a, 2.1b, and 2.1c, describe the classical optimization framework, information-constrained optimization under adaptive gradient processing, and information-constrained optimization under nonadaptive gradient processing, respectively.

We measure the performance of an optimization protocol π and a channel selection strategy S for a given function f and oracle O using the metric $\mathcal{E}(f, O, \pi, S)$ defined as

$$\mathcal{E}(f, O, \pi, S) = \mathbb{E}\left[f(x_T) - \min_{x\in\mathcal{X}} f(x)\right], \tag{2.1}$$

where the expectation is over the randomness in x_T.

For various function and oracle classes, denoted by $\mathcal{O}$, the channel constraint family $\mathcal{W}$, and the number of iterations T, we will characterize the *adaptive minmax optimization error*

$$\mathcal{E}^*(\mathcal{X}, \mathcal{O}, T, \mathcal{W}) = \inf_{\pi\in\Pi_T} \inf_{S\in\mathcal{S}_{\mathcal{W},T}} \sup_{(f,O)\in\mathcal{O}} \mathcal{E}(f, O, \pi, S)\,, \tag{2.2}$$

and the corresponding *nonadaptive minmax optimization error*

$$\mathcal{E}^{\mathrm{NA}*}(\mathcal{X}, \mathcal{O}, T, \mathcal{W}) = \inf_{\pi\in\Pi_T} \inf_{S\in\mathcal{S}^{\mathrm{NA}}_{\mathcal{W},T}} \sup_{(f,O)\in\mathcal{O}} \mathcal{E}(f, O, \pi, S)\,. \tag{2.3}$$

Since the adaptive channel selection strategies include the nonadaptive ones, we have $\mathcal{E}^{\mathrm{NA}*}(\mathcal{X}, \mathcal{O}, T, \mathcal{W}) \geq \mathcal{E}^*(\mathcal{X}, \mathcal{O}, T, \mathcal{W})$.

2.3.2 Function classes

We now define the function classes and the corresponding oracles that we consider.

Convex and ℓ_p Lipschitz function family. Our first set of function families are parameterized by a number $p \in [1, \infty]$. Throughout, we restrict ourselves to convex functions over a domain $\mathcal{X}$, i.e., functions f satisfying

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y), \quad \forall x, y \in \mathcal{X}, \quad \forall \lambda \in [0,1]. \tag{2.4}$$

Further, for a family parameterized by p, we assume that the subgradient estimates returned by the first-order oracle for a function f satisfy the following two assumptions:

$$\mathbb{E}\left[\hat{g}(x) \mid x\right] \in \partial f(x), \quad \text{(Unbiased estimates)} \tag{2.5}$$

$$\Pr\Big(\|\hat{g}(x)\|_q^2 \leq B^2 \mid x\Big) = 1, \quad \text{(Bounded estimates)} \tag{2.6}$$

where $\partial f(x)$ is the set of subgradient for f at x and $q := p/(p-1)$ is, as mentioned earlier, the Hölder conjugate of p.

Definition 2.3.3 (Convex and ℓ_p Lipschitz function family $\mathcal{O}_{\mathsf{c},p}$)**.** We denote by $\mathcal{O}_{\mathsf{c},p}$ the set of all pairs of functions and oracles satisfying Assumptions (2.4), (2.5), and (2.6).

We note that (2.5) is standard in stochastic optimization literature (*cf.* [71], [70], [15], [5]). To prove convergence guarantees on first-order optimization in the classic setup (without any information constraints on the oracle), it is enough to assume $\mathbb{E}\left[\|\hat{g}(x)\|_q^2\right] \leq B^2$. We make a slightly stronger assumption in this case since the more relaxed assumption leads to technical difficulties in finding unbiased quantizers for gradients.

Note that by (2.5) and (2.6) for every $x \in \mathcal{X}$ there exists a vector $g \in \partial f(x)$ such that $\|g\|_q \leq B$. Further, since f is convex, $f(x) - f(y) \leq g^T(x-y)$ for every $g \in \partial f(x)$, whereby $|f(x) - f(y)| \leq B\|x-y\|_p$. Namely, f is B-Lipschitz continuous in the ℓ_p norm.[2]

Before proceeding, we recall the optimal convergence results under no information constraints. No information constraints can be viewed as passing the subgradients estimates through the identity channel.

[2]The same could be said under the weaker assumption $\mathbb{E}\left[\|\hat{g}(x)\|_q^2\right] \leq B^2$.

Definition 2.3.4. We denote by $I : \mathbb{R}^d \to \mathbb{R}^d$ the identity channel, where the output always equals the input. Let $\mathcal{I}$ denote the singleton set consisting only of I.

Theorem 2.3.5. *Let $\mathbb{X}_p(D) := \{\mathcal{X} \subseteq \mathbb{R}^d : \max_{x,y\in\mathcal{X}} \|x - y\|_p \leq D\}$. There exist absolute constants c_0 and c_1 where $c_1 \geq c_0 > 0$ such that the following hold:*

1. *for*[3] $2 > p \geq 1$,

$$\frac{c_1 DB\sqrt{\log d}}{\sqrt{T}} \geq \sup_{\mathcal{X}\in\mathbb{X}_p(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathsf{c},p}, T, \mathcal{I}) \geq \frac{c_0 DB}{\sqrt{T}}.$$

2. *For $p \geq 2$,*

$$\frac{c_1 d^{1/2-1/p} DB}{\sqrt{T}} \geq \sup_{\mathcal{X}\in\mathbb{X}_p(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathsf{c},p}, T, \mathcal{I}) \geq \frac{c_0 d^{1/2-1/p} DB}{\sqrt{T}};$$

The lower bounds and the upper bounds can be found, for instance, in [5, Theorem 1] and [5, Appendix C].

Remark 1. An optimal achievable scheme for $p \in [1, 2)$ is the stochastic mirror descent with the mirror maps $\|x\|_{p'}^2/(p' - 1)$, where p' is chosen appropriately for a given p. When Hölder conjugate q of p is $o(\log d)$, we choose p' to be p. When q is $\Omega(\log d)$, we choose $p' = \frac{2\log d}{2\log d - 1}$. Further, these algorithms require only that the expected squared ℓ_q norm of the gradient estimates are bounded.

Remark 2. An optimal achievable scheme for p greater than 2 is simply projected subgradient descent(PSGD). To see this, note that PSGD gives a guarantee of $D'B'/\sqrt{T}$ (*cf.* [70]), where D' is the ℓ_2 diameter and B' is the bound on $\mathbb{E}\left[\|\hat{g}\|_2^2\right]$. Using the monotonicity of ℓ_q norms in q, for $q \geq 2$ we have $\mathbb{E}\left[\|\hat{g}\|_2^2\right] \leq \mathbb{E}\left[\|\hat{g}\|_q^2\right] \leq B^2$. Also, the ℓ_2 diameter of a unit ℓ_p ball is $d^{1/2-1/p}$. It follows that PSGD attains the upper bounds in Theorem 2.3.5.

Strongly convex and ℓ_2 Lipschitz function family. We now consider a special subset of the convex and ℓ_2 Lipschitz family described above, where the functions are strongly

[3] *For certain range of p closer to 2 the $\sqrt{\log d}$ factor can be removed; for simplicity, we state the slightly weaker result.*

convex. Recall that for $\gamma > 0$, a function f is *γ-strongly convex on $\mathcal{X}$* if the following function h is convex:

$$h(x) = f(x) - \frac{\gamma}{2}\|x\|_2^2, \quad \forall x \in \mathcal{X}. \tag{2.7}$$

Definition 2.3.6 (Strongly convex and ℓ_2 Lipschitz function family $\mathcal{O}_{\mathsf{sc}}$.)**.** We denote by $\mathcal{O}_{\mathsf{sc}}$ the set of all pairs of functions and oracles satisfying (2.4), (2.5), (2.7), and (2.6) for $q = 2$.

The strong convexity parameter γ is related to the parameter B, the upper bound on the ℓ_2 norm of the gradient estimate. We state a relation between them when the domain $\mathcal{X}$ contains an ℓ_∞ ball of radius D centered at the origin; this property will be used when we derive lower bounds.

Lemma 2.3.7. *For any $\mathcal{X} \supseteq \{x : \|x\|_\infty \le D\}$, we have $\frac{B}{\gamma} \ge \frac{Dd^{1/2}}{4}$.*

Theorem 2.3.8. *Let $\mathbb{X}_2(D) := \{\mathcal{X} \subseteq \mathbb{R}^d : \max_{x,y\in\mathcal{X}} \|x - y\|_2 \le D\}$. There exist absolute constants c_0 and c_1 where $c_1 \ge c_0 > 0$ such that the following hold:*

$$\frac{c_1 B^2}{\gamma T} \ge \sup_{\mathcal{X}\in\mathbb{X}_2(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathsf{sc}}, T, \mathcal{I}) \ge \frac{c_0 B^2}{\gamma T}$$

The lower bounds and the upper bounds can be found, for instance, in [5, Theorem 1] and [70].

Remark 3. The optimal achievable scheme for strongly convex functions is the stochastic gradient descent algorithm.

2.3.3 Information constraints

We describe three specific constraints of interest to us: local privacy, communication, and computation. The first two are well-studied; the third is new and arises in procedures such as random coordinate descent.

Local differential privacy. To model local privacy, we define the ε-locally differentially private (LDP) channel family $\mathcal{W}_{\mathtt{priv},\varepsilon}$.

Definition 2.3.9. A channel $W\colon \mathbb{R}^d \to \mathbb{R}^d$ is *ε-locally differentially private* (ε-LDP) if for all $x, x' \in \mathbb{R}^d$,

$$\frac{W(Y \in S \mid X = x)}{W(Y \in S \mid X = x')} \leq e^{\varepsilon}$$

for all Borel measurable subsets S of $\mathbb{R}^d$. We denote by $\mathcal{W}_{\mathtt{priv},\varepsilon}$ the set of all ε-LDP channels.

When operating under local privacy constraints, the oracle's subgradient estimates are passed through an ε-LDP channel, and only the output is available to the optimization algorithm. Thus, the resulting process which handles the data of individual users, accessed in each oracle query, is overall differentially private, a notion of privacy extensively studied and widely used in practice.

Communication constraints. To model communication constraints, we define the $\mathcal{W}_{\mathtt{com},r}$, the r-bit communication-constrained channel family, as follows.

Definition 2.3.10. A channel $W\colon \mathbb{R}^d \to \{0,1\}^r$ constitutes an *r-bit communication-constrained* channel. We denote by $\mathcal{W}_{\mathtt{com},r}$ the set of all r-bit communication-constrained channels.

Computational constraints. For high-dimensional optimization, altogether computing the subgradient estimates can be computationally expensive. Often in such cases, one resorts to computing only a few coordinates of the gradient estimates and using only them for optimization ([73, 80]). This motivates the oblivious sampling channel family $\mathcal{W}_{\mathtt{obl}}$, where the optimization algorithm gets to see only one randomly chosen coordinate of the gradient estimate.

Definition 2.3.11. An *oblivious sampling* channel W is a channel $W\colon \mathbb{R}^d \to \mathbb{R}^d$ specified by a probability vector $(p_i)_{i\in[d]}$, *i.e.*, a vector p such that $p_i \geq 0$ for all i and $\sum_{i\in[d]} p_i = 1$. For an input $g \in \mathbb{R}^d$, the output distribution of W is given by $W(g(i)e_i \mid g) = p_i, \forall i \in [d]$. We denote by $\mathcal{W}_{\mathtt{obl}}$ the set of all oblivious sampling channels.

Therefore, at most one coordinate of the oracle's the gradient estimate can be used by the optimization algorithm. Further, this coordinate is sampled obliviously to the input gradient estimate itself.

Remark 4. We note that the special case of $p_i = \frac{1}{d}\ \forall\, i \in [d]$ corresponds to sampling employed by standard *Random Coordinate Descent* (RCD) (*cf*. [15, Section 6.4]), where at each time step only one uniformly random coordinate of the gradient is used by the gradient descent algorithm.

2.4 Main results: lower bounds for information-constrained optimization

For $p \in [1, \infty]$ and $D > 0$, let $\mathbb{X}_p(D) := \{\mathcal{X} \subseteq \mathbb{R}^d : \max_{x,y\in\mathcal{X}} \|x - y\|_p \leq D\}$ be the collection of subsets of $\mathbb{R}^d$ whose ℓ_p diameter is at most D. In stating our results, we will fix throughout the parameter $B > 0$, the almost sure bound on the gradient magnitude defined in (2.6), as well as the strong convexity parameter $\gamma > 0$ defined in (2.7) (which, implicitly, is required to satisfy Lemma 2.3.7). Throughout this section, our lower bounds on minmax optimization error focus on tracking the convergence rate for large T, a standard regime of interest for the stochastic optimization setting.

2.4.1 Lower bounds for locally private optimization under adaptive gradient processing

Throughout, we consider $\varepsilon \in [0, 1]$, namely the high-privacy regime.

Convex function family. For the convex function family, we prove the following lower bounds.

Theorem 2.4.1. *Let $p \in [1, 2]$, $\varepsilon \in [0, 1]$, and $D > 0$. There exist absolute constants*

$c_0, c_1 > 0$ such that, for $T \geq c_0 \frac{d}{\varepsilon^2}$,

$$\sup_{\mathcal{X} \in \mathbb{X}_p(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{c},p}, T, \mathcal{W}_{\mathtt{priv},\varepsilon}) \geq \frac{c_1 DB}{\sqrt{T}} \cdot \sqrt{\frac{d}{\varepsilon^2}}.$$

(Moreover, one can take $c_0 := \frac{1}{2e(e-1)^2}$ and $c_1 := \frac{1}{36(e-1)\sqrt{2e}}$.)

See Section 2.5.5 for the proof.

Theorem 2.4.2. *Let $p \in (2, \infty], \varepsilon \in [0, 1]$, and $D > 0$. There exist absolute constants $c_0, c_1 > 0$ such that, for $T \geq c_0 \frac{d^2}{\varepsilon^2}$,*

$$\sup_{\mathcal{X} \in \mathbb{X}_p(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{c},p}, T, \mathcal{W}_{\mathtt{priv},\varepsilon}) \geq \frac{c_1 DBd^{1/2-1/p}}{\sqrt{T}} \cdot \sqrt{\frac{d}{\varepsilon^2}}.$$

(Moreover, one can take c_0 and c_1 as in Theorem 2.4.1.)

See Section 2.5.6 for the proof.

Remark 5 (Tightness of bounds for convex functions and LDP constraints). [24, Theorem 4 and 5] provide nonadaptive LDP algorithms which show that Theorem 2.4.1 is tight up to logarithmic factors for $p = 1$ and Theorem 2.4.2 is tight up to constant factors for all $p \in (2, \infty]$ (to the best of our knowledge, no non-trivial upper bound is known for $p \in (1, 2)$.). Therefore, adaptive processing of gradients under LDP cannot significantly improve the convergence rate for convex function families.

Interestingly, for $p = 1$, [24] also provide a slightly stronger lower bound of $\frac{c_0 DB}{\sqrt{T}} \cdot \sqrt{\frac{d \log d}{\varepsilon^2}}$ for nonadaptive protocols, which matches the performance of their nonadaptive protocols up to constant factors. This points to a minor gap in our understanding of adaptive protocols: Can we establish a stronger lower bound for adaptive protocols to match the performance of the nonadaptive algorithm of [24], or does there exist a better adaptive protocol?

From Theorem 2.3.5, the standard optimization error for ℓ_p, $p \in [1, \infty]$, convex family blows up by a factor of $\sqrt{d/\varepsilon^2}$ when the gradient estimates are passed through an ε-LDP channel.

Strongly convex family. We prove the following result for strongly convex functions.

Theorem 2.4.3. *Let $\varepsilon \in [0,1]$, and $D > 0$. There exist absolute constants $c_0, c_1 > 0$ such that, for $T \geq c_0 \cdot \frac{B^2}{\gamma^2 D^2} \cdot \frac{d}{\varepsilon^2}$,*

$$\sup_{\mathcal{X} \in \mathbb{X}_2(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{sc}}, T, \mathcal{W}_{\mathtt{priv},\varepsilon}) \geq \frac{c_1 B^2}{\gamma T} \cdot \frac{d}{\varepsilon^2}.$$

See Section 2.5.7 for the proof.

Remark 6 (Tightness of bounds for strongly convex functions and LDP constraints). One can use stochastic gradient descent with the nonadaptive protocol from [24, Appendix C.2] to obtain a nonadaptive protocol with convergence rate matching the lower bound in Theorem 2.4.3 up to constant factors, establishing that adaptivity does not help for strongly convex functions.

From Theorem 2.3.8, the standard optimization error for strongly convex functions blows up by a factor of $\frac{d}{\varepsilon^2}$ when the gradient estimates are passed through an ε-LDP channel.

2.4.2 Lower bounds on communication-constrained optimization

Convex function family. For convex functions, we prove the following lower bounds.

Theorem 2.4.4. *Let $p \in [1,2]$, and $D > 0$. There exists an absolute constant $c_0 > 0$ such that, for $r \in \mathbb{N}$, and $T \geq \frac{d}{6r}$,*

$$\sup_{\mathcal{X} \in \mathbb{X}_p(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{c},p}, T, \mathcal{W}_{\mathtt{com},r}) \geq \frac{c_0 DB}{\sqrt{T}} \cdot \sqrt{\frac{d}{d \wedge r}}.$$

(Moreover, one can take $c_0 := \frac{1}{12\sqrt{58}}$.)

See Section 2.5.7 for the proof.

Theorem 2.4.5. *Let $p \in (2, \infty]$, and $D > 0$. There exists an absolute constant $c_0 > 0$*

such that, for $r \in \mathbb{N}$, *and* $T \geq \frac{1}{4} \cdot \frac{d^2}{2^r \wedge d}$, *we have*

$$\sup_{\mathcal{X} \in \mathbb{X}_p(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{c},p}, T, \mathcal{W}_{\mathtt{com},r}) \geq \left(\frac{c_0 DB d^{1/2-1/p}}{\sqrt{T}} \cdot \sqrt{\frac{d}{d \wedge 2^r}} \right) \vee \left(\frac{c_0 DB}{\sqrt{T}} \cdot \sqrt{\frac{d}{d \wedge r}} \right)$$

(Moreover, one can take $c_0 := \frac{1}{12\sqrt{58}}$.*)*

See Section 2.5.6 for the proof.

In Chapters 3 and 4, we will derive upper bounds for most regimes of p and r. Specifically, restricting $r \geq r^*(T, p)$ our bounds are tight for all regimes of p. Moreover, for $p = 1$ and $[2, \infty]$, our bounds are nearly tight for all r.

From Theorem 2.3.5, the standard optimization errors for ℓ_1 and ℓ_p, $p \in (2, \infty]$, convex family blow up by a factor of $\sqrt{\frac{d}{d \wedge r}}$ and $\sqrt{\frac{d}{d \wedge 2^r}} \vee \sqrt{\frac{d^{2/p}}{d \wedge r}}$, respectively, when the gradient estimates are compressed to r bits.

Strongly convex family. We prove the following result for strongly convex functions.

Theorem 2.4.6. *Let* $D > 0$. *There exist absolute constants* $c_0, c_1 > 0$ *such that, for* $r \in \mathbb{N}$ *and* $T \geq c_0 \cdot \frac{B^2}{\gamma^2 D^2} \cdot \frac{d}{r}$,

$$\sup_{\mathcal{X} \in \mathbb{X}_2(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{sc}}, T, \mathcal{W}_{\mathtt{com},r}) \geq \frac{c_1 B^2}{\gamma T} \cdot \frac{d}{d \wedge r}\,.$$

See Section 2.5.7 for the proof.

In Chapters 3, we will derive upper bounds which are tight upto a factor of $\log \log^* d$ for all r. From Remark 3, the standard optimization error for strongly convex functions blows up by a factor of $\frac{d}{r}$ when the gradient estimates are compressed to r bits.

2.4.3 Lower bounds on computationally-constrained optimization

We restrict to the case of Euclidean geometry ($p = 2$) for the oblivious sampling channel family $\mathcal{W}_{\mathtt{obl}}$. Our motivation for introducing this class was to study the optimality of standard RCD, which is proposed to work in the Euclidean setting alone. Furthermore,

if we consider a slightly larger family of channels where the sampling probabilities can depend on the input itself, the resulting family will be similar to the 1-bit communication family, which we have addressed in Section 2.4.2.

Convex family. For convex functions, we establish the following lower bound, for $p = 2$.

Theorem 2.4.7. *Let $D > 0$. There exists an absolute constant $c_0 > 0$ such that, for $T \geq \frac{d}{4}$, we have*

$$\sup_{\mathcal{X} \in \mathbb{X}_2(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{c},2}, T, \mathcal{W}_{\mathtt{obl}}) \geq \frac{c_0 \sqrt{d} DB}{\sqrt{T}}.$$

(Moreover, one can take $c_0 := \frac{1}{72}$.)

See Section 2.5.5 for a proof.

The standard Random Coordinate Descent (RCD) (see for instance [15, Theorem 6.6]), which employs uniform sampling, matches this lower bound up to constant factors. The optimality of standard RCD motivates further the folklore approach of uniformly sampling coordinates for random coordinate descent unless there is an obvious structure to exploit (as in [72]). This establishes that adaptive sampling strategies do not improve over nonadaptive sampling strategies for the family $\mathcal{W}_{\mathtt{obl}}$. Also from Theorem 2.3.5, the standard optimization error for ℓ_2 convex family blows up by a factor of $\sqrt{d}$ when the gradient coordinates are sampled obliviously.

Strongly convex family. For strongly convex functions, we obtain the following lower bound, for $p = 2$.

Theorem 2.4.8. *Let $D > 0$. There exist absolute constants $c_0, c_1 > 0$ such that, for $T \geq c_0 \cdot d \frac{B^2}{\gamma^2 D^2}$, we have*

$$\sup_{\mathcal{X} \in \mathbb{X}_2(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{sc}}, T, \mathcal{W}_{\mathtt{obl}}) \geq \frac{c_1 dB^2}{\gamma T}.$$

See Section 2.5.7 for the proof.

Once again, the standard RCD algorithm matches this lower bound, which shows that adaptive sampling strategies do not improve over nonadaptive sampling strategies for

	LDP constraints	Communication -constraints	Computational -constraints
Convex and ℓ_p Lipschitz function family ($p \in [1,2]$)	$\frac{c_1 DB}{\sqrt{T}} \cdot \sqrt{\frac{d}{\varepsilon^2}}$ (Theorem 2.4.1)	$\frac{c_1 DB}{\sqrt{T}} \cdot \sqrt{\frac{d}{d \wedge r}}$ (Theorem 2.4.4)	(Only for $p = 2$) $\frac{c_1 DB}{\sqrt{T}} \cdot \sqrt{d}$ (Theorem 2.4.7)
Convex and ℓ_p Lipschitz function family ($p \in (2, \infty]$)	$\frac{c_1 DBd^{1/2-1/p}}{\sqrt{T}} \cdot \sqrt{\frac{d}{\varepsilon^2}}$ (Theorem 2.4.2)	$\left(\frac{c_0 DBd^{1/2-1/p}}{\sqrt{T}} \cdot \sqrt{\frac{d}{d \wedge 2^r}}\right) \vee \left(\frac{c_0 DB}{\sqrt{T}} \cdot \sqrt{\frac{d}{d \wedge r}}\right)$ (Theorem 2.4.5)	N.A.
Strongly convex and ℓ_2 Lipschitz function family	$\frac{c_1 B^2}{\gamma T} \cdot \frac{d}{\varepsilon^2}$ (Theorem 2.4.3)	$\frac{c_1 B^2}{\gamma T} \cdot \frac{d}{r}$ (Theorem 2.4.6)	$\frac{c_1 dB^2}{\gamma T}$ (Theorem 2.4.8)

Table 2.2: Summary of all our lower bounds on gap-to-optimality for information-constrained optimization.

strongly convex optimization. Further, from Theorem 2.3.8, the standard optimization error for strongly convex family blows up by a factor of d when the gradient coordinates are sampled obliviously.

A summary of all our lower bounds is provided in Table 2.2.

2.5 Proofs of lower bounds

2.5.1 Outline of the proof for our lower bounds

The proofs of our lower bounds for adaptive protocols follow the same general template, summarized below.

Step 1. Relating optimality gap to average information: We consider a family of functions $\mathcal{G} = \{g_v : v \in \{-1, 1\}^d\}$ satisfying suitable conditions and associate with it a

"discrepancy metric" $\psi(\mathcal{G})$ that allows us to relate the optimality gap of any algorithm to an average mutual information quantity. Specifically, for V distributed uniformly over $\{-1,1\}^d$, we show that the output $\hat{x}$ of any optimization algorithm satisfies

$$\mathbb{E}\left[g_V(\hat{x}) - \min_{x\in\mathcal{X}} g_V(x)\right] \geq \frac{d\psi(\mathcal{G})}{6}\left[1 - \sqrt{\frac{2}{d}\sum_{i=1}^{d} I(V(i) \wedge Y^T)}\right],$$

where Y_t is the channel output for the gradient in the tth iteration and $Y^T := (Y_1, \ldots, Y_T)$.

Heuristically, we have related the gap to optimality to the difficulty of inferring V by observing Y^T. We note that the bound above is similar to that of [5], but instead of mutual information $I\Big(V \wedge Y^T\Big)$ we get the average mutual information per coordinate. This latter quantity is amenable to analysis for adaptive protocols.

Step 2. Average information bounds: To bound the average mutual information per coordinate, $\frac{1}{d}\sum_{i=1}^{d} I\Big(V(i) \wedge Y^T\Big)$, we take recourse to the recently proposed bounds from [3]. These bounds hold for Y^T which is the output of adaptively selected channels from a fixed channel family $\mathcal{W}$, with i.i.d. input $X^T = (X_1, \ldots, X_T)$ generated from a family of distributions $\{\mathbf{p}_v, v \in \{-1,1\}^d\}$. We view the output of oracle as inputs X^T and derive the required bound.

While results in [3] provided bounds for $\mathcal{W}_{\texttt{priv},\varepsilon}$ and $\mathcal{W}_{\texttt{comm},r}$, we extend the approach to handle $\mathcal{W}_{\texttt{obl}}$. Specifically, under a smoothness and symmetry condition on $\{\mathbf{p}_v, v \in \{-1,1\}^d\}$, which has a parameter γ associated with it, we show the following:

For $|\mathcal{X}| < \infty$ and $\mathcal{X}_i := \{x(i) : x \in \mathcal{X}\}$, $i \in [d]$, we have

$$\sum_{i=1}^{d} I\Big(V(i) \wedge Y^T\Big) \leq \frac{C}{2} \cdot T\gamma^2,$$

where the constant C depends only on $\{\mathbf{p}_v, v \in \{-1,+1\}^d\}$ and, denoting by $v^{\oplus i} \in \{-1,1\}^d$ the vector with the sign of the ith coordinate of v flipped, is given by

$$C = (\max_{i\in[d]} |\mathcal{X}_i| - 1) \cdot \max_{x\in\mathcal{X}} \max_{v\in\{-1,+1\}^d} \max_{i\in[d]} \frac{\mathbf{p}_{v^{\oplus i}}(X(i) = x(i))}{\mathbf{p}_v(X(i) = x(i))}.$$

Step 3. Use appropriate difficult instances On the one hand, to prove lower

bounds for the convex family we will use the class of functions $\mathcal{G}_{\mathtt{c}} = \{g_v(x) \colon v \in \{-1,1\}^d\}$ defined on the domain $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_\infty \le b\}$ comprising functions g_v given below:

$$g_v(x) = a \cdot \sum_{i=1}^{d} |x(i) - v(i) \cdot b|, \quad \forall x \in \mathcal{X}, v \in \{-1,1\}^d.$$

On the other hand, to prove lower bounds for the strongly convex family, we will use the class of functions $\mathcal{G}_{\mathtt{sc}} = \{g_v(x) \colon v \in \{-1,1\}^d\}$ on $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_\infty \le b\}$ given by

$$g_v(x) = a \sum_{i=1}^{d} \left(\frac{1 + 2\delta v(i)}{2} f_i^+(x) + \frac{1 - 2\delta v(i)}{2} f_i^-(x) \right), \quad \forall x \in \mathcal{X}, v \in \{-1,1\}^d,$$

where f_i^+ and f_i^-, for $i \in [d]$, are given by

$$f_i^+(x) = \theta b |x(i) + b| + \frac{1-\theta}{4}(x(i)+b)^2, \qquad f_i^-(x) = \theta b |x(i) - b| + \frac{1-\theta}{4}(x(i)-b)^2.$$

Step 4. Carefully combine everything: We obtain our desired bounds by applying Steps 1 and 2 to difficult instances from Step 3. Since the difficult instance for convex family consists of linear functions, the gradient does not depend on x. Thus, we can design oracles which give i.i.d. output with distribution independent of the query point x_t, whereby the bound in Step 2 can be applied. Interestingly, we construct different oracles for $p < 2$ and $p \ge 2$.

However, the situation is different for the strongly convex family. The gradients now depend on the query point x_t, whereby it is unclear if we can comply with the requirements in Step 2. Interestingly, for communication and local privacy constraints, we construct oracles that allow us to view messages Y^T as the output of adaptively selected channels applied to independent samples from a common distribution $\mathbf{p}_v$. While it is unclear if the same can be done for computational constraints as well, we use an alternative approach and exhibit an oracle for which we can find an intermediate message vector $Z_1, \ldots, Z_T$ such that (i) V and Y^T are conditionally independent given Z^T and (ii) the message Z^T satisfies the requirements of Step 2.

2.5.2 Relating optimality gap to average information

In this section, we prove a general lower bound for the expected gap to optimality by considering a parameterized family of functions and oracles which is contained in our oracle family of interest. We present a bound that relates the expected gap to optimality to the average mutual information between the channel output and different coordinates of the unknown parameter. This step is the key difference between our approach and that of [5], which used Fano's method instead of our bound below. We remark that the bounds resulting from Fano's method are typically not amenable to analysis for adaptive protocols.

In more detail, our result can be used to prove bounds for the average optimization error over any class of functions which satisfies the two conditions below.

Assumptions 2.5.1. Let $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{V} = \{-1, 1\}^d$. Let $\mathcal{G} = \{g_v : v \in \mathcal{V}\}$ where $g_v : \mathcal{X} \to \mathbb{R}$ are real-valued functions from $\mathcal{X}$ such that

1. the g_vs are *coordinate-wise decomposable*, *i.e.*, there exist functions $g_{i,b} \colon \mathbb{R} \to \mathbb{R}$, $i \in [d]$, $b \in \{-1, 1\}$, such that

$$g_v(x) = \sum_{i=1}^{d} g_{i,v(i)}(x(i)).$$

2. the minimum of g_v is also a *coordinate-wise minimum*, *i.e.*, if we denote by x_v^* the minimum of g_v over $\mathcal{X}$, then, for all $i \in [d]$, we have

$$x_v^*(i) = \operatorname*{argmin}_{y \in \mathcal{X}_i} g_{i,v(i)}(y),$$

where $\mathcal{X}_i = \{x(i) : x \in \mathcal{X}\}$.

For $\mathcal{G}$ satisfying Assumptions 2.5.1 and for $i \in [d]$, we now define the following discrepancy metric:

$$\psi_i(\mathcal{G}) := \min_{y \in \mathcal{X}_i} \left(g_{i,1}(y) + g_{i,-1}(y) - \left(\min_{y' \in \mathcal{X}_i} g_{i,1}(y') + \min_{y' \in \mathcal{X}_i} g_{i,-1}(y') \right) \right) \tag{2.8}$$

$$\psi(\mathcal{G}) := \min_{i\in[d]} \psi_i(\mathcal{G}). \tag{2.9}$$

This is a "coordinate-wise counterpart" of the metric used in [5]. The next lemma follows readily from this definition.

Lemma 2.5.2. *Fix $i \in [d]$. For every $y \in \mathcal{X}_i$, there can be at most one $b \in \{-1, 1\}$ such that*

$$g_{i,b}(y) - \min_{y'\in\mathcal{X}_i} g_{i,b}(y') \le \frac{\psi_i(\mathcal{G})}{3}.$$

Proof. Let $b \in \{-1, 1\}$. By definition of $\psi_i(\mathcal{G})$, for all $y \in \mathcal{X}_i$ we have

$$\left(g_{i,b}(y) - \min_{y'\in\mathcal{X}_i} g_{i,b}(y')\right) + \left(g_{i,-b}(y) - \min_{y'\in\mathcal{X}_i} g_{i,-b}(y')\right) \ge \psi_i(\mathcal{G}).$$

For y such that $g_{i,b}(y) - \min_{y'\in\mathcal{X}_i} g_{i,b}(y') \le \frac{\psi_i(\mathcal{G})}{3}$, we now must have that

$$g_{i,-b}(y) - \min_{y'\in\mathcal{X}_i} g_{i,-b}(y') \ge \frac{2\psi_i(\mathcal{G})}{3}.$$ □

We will use this observation to bound the expected gap to optimality for any algorithm π optimizing an unknown function in $\mathcal{G}$ that has access to only the corresponding first-order oracle.

Lemma 2.5.3. *Suppose $\mathcal{G} = \{g_v : v \in \{-1, 1\}^d\}$ satisfies Assumption 2.5.1. Let π be any optimization algorithm that adaptively selects the channels $\{W_j\}_{j\in[T]}$. For a random variable V distributed uniformly over $\{-1, 1\}^d$, the output $\hat{x}$ of π when it is applied to a function from $\mathcal{G}$ and any associated (stochastic subgradient) oracle satisfies*

$$\mathbb{E}\left[g_V(\hat{x}) - g_V(x_V^*)\right] \ge \frac{d\psi(\mathcal{G})}{6}\left[1 - \sqrt{\frac{1}{d}\sum_{i=1}^{d} 2I(V(i) \wedge Y^T)}\right],$$

where $\psi(\mathcal{G}) = \min_{j\in[d]} \psi_j(\mathcal{G})$, Y_t is the channel output for the gradient at time step t and $Y^T := (Y_1, \ldots, Y_T)$.

Proof. Our proof is based on relating the gap to optimality to the error in estimation of V upon observing Y^T. Suppose the algorithm π along with channels $\{W_j\}_{j\in[T]}$ outputs

the point $\hat{x}$ after T iterations. By linearity of expectation, the decomposability of g_v, and Markov's inequality, we have

$$\begin{aligned}\mathbb{E}\left[g_V(\hat{x}) - g_V(x_V^*)\right] &= \sum_{i=1}^{d} \mathbb{E}\left[g_{i,V(i)}(\hat{x}(i)) - g_{i,V(i)}(x_V^*(i))\right] \\ &\geq \sum_{i=1}^{d} \frac{\psi_i(\mathcal{G})}{3} \Pr\left(g_{i,V(i)}(\hat{x}(i)) - g_{i,V(i)}(x_V^*(i)) \geq \frac{\psi_i(\mathcal{G})}{3}\right) \\ &\geq \frac{\psi(\mathcal{G})}{3} \sum_{i=1}^{d} \Pr\left(g_{i,V(i)}(\hat{x}(i)) - g_{i,V(i)}(x_V^*(i)) \geq \frac{\psi_i(\mathcal{G})}{3}\right). \end{aligned} \tag{2.10}$$

We proceed to bound each summand separately.

Fix any $i \in [d]$ and consider the following estimate for $V(i)$: Given $\hat{x}$, we output a $\hat{V}(i) \in \{-1, 1\}$ satisfying

$$g_{i,\hat{V}(i)}(\hat{x}(i)) - \min_{y' \in \mathcal{X}_i} g_{i,\hat{V}(i)}(y') < \frac{\psi_i(G)}{3};$$

if no such $\hat{V}(i)$ exists, we generate $\hat{V}(i)$ uniformly from $\{-1, 1\}$. Then, as a consequence of Lemma 2.5.2, we get

$$\Pr\left(\hat{V}(i) \neq v(i)\right) \leq \Pr\left(g_{i,v(i)}(\hat{x}(i)) - g_{i,v(i)}(x_v^*(i)) \geq \frac{\psi_i(\mathcal{G})}{3}\right). \tag{2.11}$$

Next, denote by $\mathbf{p}^{Y^T}$ the distribution of Y^T and by $\mathbf{p}_{+i}^{Y^T}$ and $\mathbf{p}_{-i}^{Y^T}$, respectively, the distributions of Y^T given $V(i) = +1$ and $V(i) = -1$. It is easy to verify that

$$\mathbf{p}^{Y^T} = \frac{1}{2}(\mathbf{p}_{+i}^{Y^T} + \mathbf{p}_{-i}^{Y^T}), \quad \forall\, i \in [d].$$

Noting that $V(i)$ is uniform and the estimate $\hat{V}(i)$ is formed as a function of Y^T, we get

$$\Pr\left(\hat{V}(i) \neq v(i)\right) \geq \frac{1}{2} - \frac{1}{2}\mathrm{d}_{\mathrm{TV}}\left(\mathbf{p}_{+i}^{Y^T}, \mathbf{p}_{-i}^{Y^T}\right). \tag{2.12}$$

From this, combining (2.11) and (2.12) and plugging the result into (2.10), we have

$$\begin{aligned}
\mathbb{E}\left[g_v(\hat{x}) - g_v(x_v^*)\right] &\geq \frac{\psi(\mathcal{G})}{6}\sum_{i=1}^{d}\left[1 - \mathrm{d}_{\mathrm{TV}}\left(\mathbf{p}_{+i}^{Y^T}, \mathbf{p}_{-i}^{Y^T}\right)\right] \\
&\geq \frac{\psi(\mathcal{G})}{6}\sum_{i=1}^{d}\left[1 - \mathrm{d}_{\mathrm{TV}}\left(\mathbf{p}_{+i}^{Y^T}, \mathbf{p}^{Y^T}\right) - \mathrm{d}_{\mathrm{TV}}\left(\mathbf{p}_{-i}^{Y^T}, \mathbf{p}^{Y^T}\right)\right] \\
&\geq \frac{\psi(\mathcal{G})}{6}\sum_{i=1}^{d}\left[1 - \sqrt{\frac{1}{2}\mathrm{D}\left(\mathbf{p}_{+i}^{Y^T}\|\mathbf{p}^{Y^T}\right)} - \sqrt{\frac{1}{2}\mathrm{D}\left(\mathbf{p}_{-i}^{Y^T}\|\mathbf{p}^{Y^T}\right)|}\right] \\
&\geq \frac{d\psi(\mathcal{G})}{6}\left[1 - \sqrt{\frac{1}{d}\sum_{i=1}^{d}\mathrm{D}\left(\mathbf{p}_{+i}^{Y^T}\|\mathbf{p}^{Y^T}\right) + \mathrm{D}\left(\mathbf{p}_{-i}^{Y^T}\|\mathbf{p}^{Y^T}\right)}\right] \\
&= \frac{d\psi(\mathcal{G})}{6}\left[1 - \sqrt{\frac{2}{d}\sum_{i=1}^{d} I(V(i) \wedge Y^T)}\right],
\end{aligned}$$

where the second inequality follows from the triangle inequality, the third is Pinsker's inequality, and the fourth is Jensen's inequality. □

2.5.3 Average information bounds

The next step in our proof is to bound the average mutual information that emerged in Section 2.5.2. A general recipe for bounding this average mutual information has been given recently in [3], which we recall below.

Let $\{\mathbf{p}_v, v \in \{-1,1\}^d\}$ be a family of distributions over some domain $\mathcal{X}$ and $\mathcal{W}$ be a fixed channel family. For $v \in \{-1,1\}^d$ and $i \in [d]$, denote by $v^{\oplus i}$ the element of $\{-1,1\}^d$ obtained by flipping the ith coordinate of v. For a fixed v, we obtain T independent samples $X_1, \ldots, X_T$ from $\mathbf{p}_v$. Let $Y_1, \ldots, Y_T$ be the output of channels selected from the channel family $\mathcal{W}$ by an adaptive channel selection strategy (see Section 2.3.1) when input to the channel at time t is X_t, $1 \leq t \leq T$.[4]

For V distributed uniformly on $\{-1,1\}^d$, we are interested in bounding $(1/d)\sum_{i=1}^{d} I\left(V(i) \wedge Y^T\right)$. In [3], different bounds were given for this quantity under different assumptions. We state these assumptions below.

[4]The bound in [3] allows even shared randomness U in its definition of interactive protocols. We have omitted U in the description for simplicity.

Assumptions 2.5.4. For every $v \in \{-1,1\}^d$ and $i \in [d]$, there exists $\phi_{v,i} \colon \mathcal{X} \to \mathbb{R}$ such that $\mathbb{E}_{\mathbf{p}_v}\left[\phi_{v,i}^2\right] = 1$, $\mathbb{E}_{\mathbf{p}_v}\left[\phi_{v,i}\phi_{v,j}\right] = \mathbb{1}_{\{i=j\}}$ holds for all $i, j \in [d]$, and

$$\frac{d\mathbf{p}_{v^{\oplus i}}}{d\mathbf{p}_v} = 1 + \gamma\phi_{v,i},$$

where $\gamma \in \mathbb{R}$ is a fixed constant independent of v, i.

Assumptions 2.5.5. There exists some $\kappa_{\mathcal{W}} \geq 1$ such that

$$\max_{v\in\{-1,1\}^d} \max_{y\in\mathcal{Y}} \sup_{W\in\mathcal{W}} \frac{\mathbb{E}_{\mathbf{p}_{v^{\oplus i}}}\left[W(y \mid X)\right]}{\mathbb{E}_{\mathbf{p}_v}\left[W(y \mid X)\right]} \leq \kappa_{\mathcal{W}}.$$

Assumptions 2.5.6. There exists some $\sigma \geq 0$ such that, for all $v \in \{-1,1\}^d$, the vector $\phi_v(X) := (\phi_{v,i}(X))_{i\in[d]} \in \mathbb{R}^d$ is σ^2-subgaussian for $X \sim \mathbf{p}_v$.[5] Further, for any fixed z, the random variables $\phi_{v,i}(X)$ are independent across $i \in [d]$.

We then have the following bound local privacy constraints.

Theorem 2.5.7 ([3, Corollary 6])**.** *Consider $\{\mathbf{p}_v, v \in \{-1,1\}^d\}$ satisfying Assumption 2.5.4 and the channel family $\mathcal{W} = \mathcal{W}_{\mathtt{priv},\varepsilon}$. Let V be distributed uniformly over $\{-1,1\}^d$ and Y^T be the output of channels selected by the optimization algorithm as above. Then, we have*

$$\sum_{i=1}^{d} I\left(V(i) \wedge Y^T\right) \leq T \cdot \frac{\gamma^2}{2} \cdot e^{\varepsilon}(e^{\varepsilon} - 1)^2.$$

For the case of communication constraints, we have the analogous statement below:

Theorem 2.5.8 ([3, Corollary 6])**.** *Consider $\{\mathbf{p}_v, v \in \{-1,1\}^d\}$ satisfying Assumptions 2.5.4 and 2.5.5 and the channel family $\mathcal{W} = \mathcal{W}_{\mathtt{com},r}$. Let V be distributed uniformly over $\{-1,1\}^d$ and Y^T be the output of channels selected by the optimization algorithm as above. Then, we have*

$$\sum_{i=1}^{d} I\left(V(i) \wedge Y^T\right) \leq \frac{1}{2}\kappa_{\mathcal{W}_{\mathtt{com},r}} \cdot T\gamma^2(2^r \wedge d).$$

[5] Recall that a random variable Y is σ^2-subgaussian if $\mathbb{E}[Y] = 0$ and $\mathbb{E}\left[e^{\lambda Y}\right] \leq e^{\sigma^2\lambda^2/2}$ for all $\lambda \in \mathbb{R}$; and that a vector-valued random variable Y is σ^2-subgaussian if its projection $\langle Y, u\rangle$ is σ^2-subgaussian for every unit vector u.

Moreover, if Assumption 2.5.6 holds as well, we have

$$\sum_{i=1}^{d} I\Big(V(i) \wedge Y^T\Big) \le (\ln 2)\kappa_{\mathcal{W}_{\mathtt{com},r}}\, \sigma^2 \cdot T\gamma^2 r.$$

Finally, we derive a bound for the oblivious sampling channel family.

Theorem 2.5.9. *Consider $\{\mathbf{p}_v, v \in \{-1,1\}^d\}$ satisfying Assumption 2.5.4 and the channel family $\mathcal{W} = \mathcal{W}_{\mathtt{obl}}$. Let V be distributed uniformly over $\{-1,1\}^d$ and Y^T be the output of channels selected by the optimization algorithm as above. Further, assume that $|\mathcal{X}| < \infty$. Then, we have*

$$\sum_{i=1}^{d} I\Big(V(i) \wedge Y^T\Big) \le \frac{C}{2} \cdot T\gamma^2,$$

where the constant C depends only on $\{\mathbf{p}_v, v \in \{-1,+1\}^d\}$ and, denoting $\mathcal{X}_i := \{x(i) : x \in \mathcal{X}\}$, is given by

$$C = (\max_{i\in[d]} |\mathcal{X}_i| - 1) \cdot \max_{x\in\mathcal{X}} \max_{v\in\{-1,+1\}^d} \max_{i\in[d]} \frac{\mathbf{p}_{v^{\oplus i}}(X(i) = x(i))}{\mathbf{p}_v(X(i) = x(i))}.$$

Proof. We recall another result from [3, Theorem 5]: Under Assumptions 2.5.4 and 2.5.5, we have[6]

$$\sum_{i=1}^{d} I\Big(V(i) \wedge Y^T\Big) \le \frac{1}{2}\kappa_{\mathcal{W}_{\mathtt{obl}}} \cdot T\gamma^2 \max_{v\in\{-1,1\}^d} \max_{W\in\mathcal{W}_{\mathtt{obl}}} \sum_{y\in\mathcal{Y}} \frac{\mathrm{Var}_{\mathbf{p}_v}[W(y \mid X)]}{\mathbb{E}_{\mathbf{p}_v}[W(y \mid X)]}.$$

We now evaluate various parameters involved in this bound. Let W be a oblivious sampling channel specified by the probability vector $(p_i)_{i\in[d]}$. Note that a channel $W \in \mathcal{W}_{\mathtt{obl}}$ can be equivalently viewed as having output alphabet $\mathcal{Y} = \{(i,z) \colon z \in \mathcal{X}_i, i \in [d]\}$. Recall that for an input x, the channel output is $x(i)$ with probability p_i, $i \in [d]$, *i.e.*, for $y = (i,z)$, $W(y \mid x) = p_i \mathbb{1}_{\{x(i)=z\}}$. Thus, we have

$$\sum_{y\in\mathcal{Y}} \frac{\mathrm{Var}_{\mathbf{p}_v}[W(y \mid X)]}{\mathbb{E}_{\mathbf{p}_v}[W(y \mid X)]} = \sum_{i=1}^{d} \sum_{z\in\mathcal{X}_i} \frac{p_i^2 \Pr(X(i) = z) - p_i^2 \Pr(X(i) = z)^2}{p_i \Pr(X(i) = z)}$$

[6] This is the general bound underlying Theorem 2.5.7.

$$= \sum_{i=1}^{d} p_i(|\mathcal{X}_i| - 1)$$
$$\leq \max_{i\in[d]} |\mathcal{X}_i| - 1.$$

Furthermore, proceeding similarly, we get that Assumption 2.5.5 holds as well with

$$\kappa_{\mathcal{W}_{\mathsf{obl}}} = \max_{x\in\mathcal{X}} \max_{v\in\{-1,+1\}^d} \max_{i\in[d]} \frac{\mathbf{p}_{v^{\oplus i}}(X(i) = x(i))}{\mathbf{p}_v(X(i) = x(i))}.$$

The proof is completed by combining the bounds above. □

2.5.4 The difficult instances for our lower bounds

With our general tools ready, we now describe the precise constructions of function families we use to get our lower bounds. We first provide the details of a family $\mathcal{G}_{\mathsf{c}}(a, b)$ of convex functions, before turning to $\mathcal{G}_{\mathsf{sc}}(a, b, \delta, \theta)$, our family of hard instances for the strongly convex setting. In both cases, our families of hard instances are parameterized (by a, b and a, b, δ, θ, respectively), and setting those parameters carefully will enable us to prove our various results.

Difficult functions for the convex family. To prove lower bounds for the convex family, we will use the class of functions $\mathcal{G}_{\mathsf{c}}(a, b)$ below, parameterized by $a, b > 0$ and defined on the domain $\mathcal{X}$ as follows:

$$\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_\infty \leq b\},$$
$$g_v(x) = a \cdot \sum_{i=1}^{d} |x(i) - v(i) \cdot b|, \quad \forall x \in \mathcal{X}, v \in \{-1, 1\}^d, \text{ and}$$
$$\mathcal{G}_{\mathsf{c}} = \{g_v(x) \colon v \in \{-1, 1\}^d\}. \tag{2.13}$$

Observe that the class $\mathcal{G}_{\mathsf{c}}$ satisfies the conditions in Assumption 2.5.1 with $g_{i,1}(x) = a|x(i) - b|$ and $g_{i,-1}(x) = a|x(i) + b|$ and $\mathcal{X}_i = [-b, b]$ for all $i \in [d]$. Further, we can bound the discreprency metric for this class as follows.

Lemma 2.5.10. *For the class of functions $\mathcal{G}_{\mathsf{c}}$ defined in (2.13), we have $\psi(\mathcal{G}_{\mathsf{c}}) \geq 2ab$.*

Proof. Note that $\min_{x\in[-b,b]} g_{i,1}(x) = \min_{x\in[-b,b]} g_{i,-1}(x) = 0$. Therefore, for all $i \in [d]$,

$$\psi_i(\mathcal{G}_{\mathsf{c}}) = \min_{x\in[-b,b]} \left(a|x(i) - b| + a|x(i) + b|\right) \geq 2ab,$$

where the inequality follows from the triangle inequality. □

Difficult functions for the strongly convex family. To prove lower bounds for the strongly convex family, we will use the class of functions $\mathcal{G}_{\mathsf{sc}}(a, b, \delta, \theta)$, parameterized by $a, b > 0$, $\delta > 0$, and $\theta \in [0, 1]$, and defined on the domain $\mathcal{X}$ as follows:

$$\begin{aligned}
&\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_\infty \leq b\},\\
&g_v(x) = a\sum_{i=1}^{d}\left(\frac{1 + 2\delta v(i)}{2} f_i^+(x) + \frac{1 - 2\delta v(i)}{2} f_i^-(x)\right), \quad \forall x \in \mathcal{X}, v \in \{-1, 1\}^d, \text{ and}\\
&\mathcal{G}_{\mathsf{sc}} = \{g_v(x) \colon v \in \{-1, 1\}^d\},
\end{aligned} \tag{2.14}$$

where f_i^+ and f_i^-, for $i \in [d]$, are given by

$$f_i^+(x) = \theta b|x(i) + b| + \frac{1 - \theta}{4}(x(i) + b)^2, \tag{2.15}$$

$$f_i^-(x) = \theta b|x(i) - b| + \frac{1 - \theta}{4}(x(i) - b)^2, \tag{2.16}$$

for all $x \in \mathcal{X}$. We can check that, for every $v \in \{-1, 1\}^d$, the function g_v is then γ-strongly convex for $\gamma := a \cdot \frac{1-\theta}{4}$. Moreover, we have the following bound for the discrepancy metric.

Lemma 2.5.11. *For the class of functions $\mathcal{G}_{\mathsf{sc}}$ defined in (2.14), if $\frac{1-\theta}{1+\theta} \geq 2\delta$ then $\psi(\mathcal{G}_{\mathsf{sc}}) \geq \frac{2ab^2\delta^2}{1-\theta}$.*

Proof. This follows from similar calculations as in [5, Appendix A]; we provide the proof here for completeness. Fixing any $v \in \{-1, 1\}^d$, we first note that by definition of $\mathcal{G}_{\mathsf{sc}}$, the function g_v can be indeed be decomposed as $g_v(x) = \sum_{i=1}^d g_{i,v(i)}(x_i)$ for $x \in \mathcal{X}$ (i.e.,

$\|x\|_\infty \le b$), where, for $i \in [d]$, $\nu \in \{-1,1\}$ and $y \in \mathcal{X}_i := [-b,b]$,

$$\begin{aligned} g_{i,\nu}(y) &= a\left(\frac{1+2\delta\nu}{2}\left(\theta b|y+b| + \frac{1-\theta}{4}(y+b)^2\right) + \frac{1-2\delta\nu}{2}\left(\theta b|y-b| + \frac{1-\theta}{4}(y-b)^2\right)\right) \\ &= a\left(\frac{1-\theta}{4}y^2 + \frac{1+3\theta}{4}b^2 + \delta\nu(1+\theta)by\right) \end{aligned}$$

where the second line relies on the fact that $|y+b| = y+b$ and $|y-b| = b-y$ for $|y| \le b$. One can easily see, e.g., by differentiation, that $g_{i,\nu}$ is minimized at $y^* := -2\delta\nu\frac{1+\theta}{1-\theta}b$ which does satisfy $|y^*| \le b$ given our assumption $\frac{1-\theta}{1+\theta} \ge 2\delta$. It follows that $\min_{y\in\mathcal{X}_i} g_{i,1}(y) = \min_{y\in\mathcal{X}_i} g_{i,-1}(y) = ab^2\left(\frac{1+3\theta}{4} - \delta^2\frac{(1+\theta)^2}{1-\theta}\right)$. Similarly, we have, for $y \in \mathcal{X}_i$,

$$g_{i,1}(y) + g_{i,-1}(y) = a\left(\frac{1-\theta}{2}y^2 + \frac{1+3\theta}{2}b^2\right)$$

which is minimized at $y^* = 0$, where it takes value $ab^2\frac{1+3\theta}{2}$. Putting it together,

$$\psi_i(\mathcal{G}_{\mathsf{sc}}) = \min_{y\in\mathcal{X}_i}(g_{i,1}(y) + g_{i,-1}(y)) - (\min_{y\in\mathcal{X}_i} g_{i,1}(y) + \min_{y\in\mathcal{X}_i} g_{i,-1}(y)) = 2ab^2\delta^2\frac{(1+\theta)^2}{1-\theta}\,.$$

Finally, $\psi(\mathcal{G}_{\mathsf{sc}}) = \min_{i\in[d]}\psi_i(\mathcal{G}_{\mathsf{sc}}) = 2ab^2\delta^2\frac{(1+\theta)^2}{1-\theta} \ge \frac{2ab^2\delta^2}{1-\theta}$, as claimed. □

2.5.5 Convex Lipschitz functions for $p \in [1,2]$: Proof of Theorems 2.4.1, 2.4.4, and 2.4.7

We first prove Theorems 2.4.1 and 2.4.4, our lower bounds on optimization of convex functions for $p \in [1,2]$ under privacy and communication constraints, respectively. We consider the class of functions $\mathcal{G}_{\mathsf{c}}$ defined in (2.13) with parameters $a := 2B\delta/d^{1/q}$ and $b := D/(2d^{1/p})$. That is, $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_\infty \le D/(2d^{1/p})\}$ and

$$g_v(x) := \frac{2B\delta}{d^{1/q}}\sum_{i=1}^{d}\left|x(i) - \frac{v(i)D}{2d^{1/p}}\right| \qquad x \in \mathcal{X}, v \in \{-1,1\}^d. \tag{2.17}$$

Note that the gradient of g_v is equal to $-2B\delta v/d^{1/q}$ at every $x \in \mathcal{X}$.

For each g_v, consider the corresponding gradient oracle O_v which outputs independent

values for each coordinate, with the ith coordinate taking values $-B/d^{1/q}$ and $B/d^{1/q}$ with probabilities $(1+2\delta v(i))/2$ and $(1-2\delta v(i))/2$, respectively, for some parameter $\delta > 0$ to be suitably chosen later.

Clearly, $\mathcal{X} \in \mathbb{X}_p(D)$ and all the functions g_v and the corresponding oracles O_v belong to the convex function family $\mathcal{O}_{\mathtt{c},p}$. We begin by noting that for V distributed uniformly over $\{-1,1\}^d$, we have

$$\sup_{\mathcal{X}\in\mathbb{X}_p(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{c},p}, T, \mathcal{W}_{\mathtt{priv},\varepsilon}) \geq \mathbb{E}\left[g_V(x_T) - g_V(x_V^*)\right],$$

where the expectation is over v as well as the randomness in x_T.

From Lemma 2.5.3 and 2.5.10, we have

$$\mathbb{E}\left[g_V(x_T) - g_V(x_V^*)\right] \geq \frac{d\cdot ab}{3}\cdot\left[1 - \sqrt{\frac{2}{d}\sum_{i=1}^{d} I(V(i)\wedge Y^T)}\right], \tag{2.18}$$

where $Y^T = (Y_1, ..., Y_T)$ are the channel outputs for the gradient estimates supplied by the oracle for the T queries.

Next, we apply the average information bound from Section 2.5.3. To do so, observe that by the definition of our oracle, the oracle output at each time step is an independent draw from the product distribution $\mathbf{p}_v$ on $\Omega := \left\{-\frac{B}{d^{1/q}}, \frac{B}{d^{1/q}}\right\}^d$ (in particular, $\mathbf{p}_v$ is the same at each time step, as it does not depend on the query x_t at time step t to the oracle). We treat the output of the independent outputs of the oracle as i.i.d. samples $X_1, ..., X_T$ in Section 2.5.3 and the corresponding channel outputs as Y^T. We can check that, for every $i \in [d]$, we have

$$\frac{\mathbf{p}_{v^{\oplus i}}(x)}{\mathbf{p}_v(x)} = \frac{1+2\delta v(i)\,\mathrm{sign}(x(i))}{1-2\delta v(i)\,\mathrm{sign}(x(i))} \tag{2.19}$$

for all $x \in \Omega$, and that Assumption 2.5.4 is satisfied with

$$\gamma := \frac{4\delta}{\sqrt{1-4\delta^2}}, \qquad \phi_{i,v}(x) := \frac{v(i)\,\mathrm{sign}(x(i)) + 2\delta}{\sqrt{1-4\delta^2}}. \tag{2.20}$$

Furthermore, noting that Assumption 2.5.5 always holds with

$$\kappa_{\mathcal{W}} = \max_{v\in\{-1,1\}^d} \max_{x\in\Omega} \max_{i\in[d]} \frac{\mathbf{p}_{v^{\oplus i}}(x)}{\mathbf{p}_v(x)},$$

it is satisfied with $\kappa_{\mathcal{W}} = 2$ (regardless of $\mathcal{W}$), as long as $\delta \le 1/6$, since the right-side above is bounded by 2 for such a δ. Finally, Assumption 2.5.6, is also satisfied as $(\phi_{i,v}(X))_{i\in[d]}$ for $X \sim \mathbf{p}_v$ is σ^2-subgaussian for $\sigma^2 := \frac{1}{1-4\delta^2}$.

Completing the proof of Theorem 2.4.1 (LDP constraints). From Theorem 2.5.7 and the bounds derived above, we have

$$\sum_{i=1}^{d} I\Big(V(i) \wedge Y^T\Big) \le T \cdot \frac{8\delta^2}{1-4\delta^2} \cdot e^{\varepsilon}(e^{\varepsilon}-1)^2,$$

and therefore,

$$\sum_{i=1}^{d} I\Big(V(i) \wedge Y^T\Big) \le c \cdot T\delta^2\varepsilon^2,$$

where $c := 9e(e-1)^2$ (recalling that $\varepsilon \in (0,1]$ and $\delta \le 1/6$). Substituting this bound on the average mutual information in (2.18) along with the values of a and b, we have

$$\mathbb{E}\left[g_V(x_T) - g_V(x_V^*)\right] \ge \frac{DB\delta}{3} \cdot \left[1 - \sqrt{\frac{2cT\delta^2\varepsilon^2}{d}}\right].$$

Upon setting $\delta := \sqrt{\frac{d}{8cT\varepsilon^2}}$, we get

$$\mathbb{E}\left[g_V(x_T) - g_V(x_V^*)\right] \ge \frac{1}{12\sqrt{2c}} \cdot \frac{DB}{\sqrt{T}} \cdot \sqrt{\frac{d}{\varepsilon^2}},$$

where we require $T \ge \frac{9}{2c} \cdot \frac{d}{\varepsilon^2}$ in order to enforce $\delta \le 1/6$. □

Completing the proof of Theorem 2.4.4 (Communication constraints). From Theorem 2.5.8 and γ, σ, and $\kappa_{\mathcal{W}}$ set as discussed above, we have

$$\sum_{i=1}^{d} I\Big(V(i) \wedge Y^T\Big) \le \frac{32(\ln 2)}{(1-4\delta^2)^2} \cdot T\delta^2 r,$$

whereby, using $\delta \leq 1/6$,

$$\sum_{i=1}^{d} I\Big(V(i) \wedge Y^T\Big) \leq 29T\delta^2 r.$$

Substituting this bound on mutual information in (2.18) along with the values of a and b, we have

$$\mathbb{E}\left[g_V(x_T) - g_V(x_V^*)\right] \geq \frac{DB\delta}{3} \cdot \left[1 - \frac{1}{\sqrt{d}} \cdot \sqrt{58T\delta^2 r}\right].$$

Setting $\delta := \sqrt{\frac{d}{232rT}}$, we finally get

$$\mathbb{E}\left[g_V(x_T) - g_V(x_V^*)\right] \geq \frac{1}{12\sqrt{58}} \cdot \frac{DB}{\sqrt{T}} \cdot \sqrt{\frac{d}{r}},$$

where we require $T \geq \frac{9}{58} \cdot \frac{d}{r}$ in order to enforce $\delta \leq 1/6$. □

Remark 7. Finally, we remark that the lower bound as in Theorem 2.4.4 also holds when the communication constraint of r bits is satisfied in expectation and not in the strict worst-case sense. The modification to the proof above is minimal. The only change is that the mutual information term is now bounded by using a strong data processing inequality from [25]. This bound holds for variable-length quantizers that satisfy the communication constraints in expectation, as opposed to the current bound from [3], which only holds for fixed-length quantizers. Specifically, from [25, Proposition 2], we have that

$$\sum_{i=1}^{d} I\Big(V(i) \wedge Y^T\Big) \leq cT\delta^2 r,$$

when the communication channel restricts the length of the outputs Y_t to r bits in expectation. A caveat here is that the strong data processing inequality from [25] holds only for nonadaptive channel selection strategies. Thus we have the same lower bound on optimization error of family $\mathcal{O}_{\mathsf{c},p}$, $p \in [1, 2]$, as in Theorem 2.4.4 even when the communication constraints are satisfied in expectation as long as the gradient processing is done in a nonadaptive manner.

Completing the proof of Theorem 2.4.7 (Computational constraints). Note that the sets $\mathcal{X}_i$s in Theorem 2.5.9 have $|\mathcal{X}_i| = 2$ for our oracle. Further,

$$\frac{\mathbf{p}_{v^{\oplus i}}(X(i) = x(i))}{\mathbf{p}_v(X(i) = x(i))} = \frac{\mathbf{p}_{v^{\oplus i}}(x)}{\mathbf{p}_v(x)} = \frac{1 + 2\delta v(i)\,\mathrm{sign}(x(i))}{1 - 2\delta v(i)\,\mathrm{sign}(x(i))} \leq 2,$$

when $\delta \leq 1/6$. Thus, the constant C in Theorem 2.5.9 is less than 2, whereby

$$\sum_{i=1}^{d} I\Big(V(i) \wedge Y^T\Big) \leq \frac{16\delta^2}{1 - 4\delta^2} \cdot T,$$

whereby, using $\delta \leq 1/6$,

$$\sum_{i=1}^{d} I\Big(V(i) \wedge Y^T\Big) \leq 18T\delta^2.$$

Substituting this bound on mutual information in (2.18) along with the values of a and b, we have

$$\mathbb{E}\left[g_V(x_T) - g_V(x_V^*)\right] \geq \frac{DB\delta}{3} \cdot \left[1 - \frac{1}{\sqrt{d}} \cdot \sqrt{36T\delta^2}\right].$$

Setting $\delta := \sqrt{\frac{d}{144T}}$, we finally get

$$\mathbb{E}\left[g_V(x_T) - g_V(x_V^*)\right] \geq \frac{1}{72} \cdot \frac{DB\sqrt{d}}{\sqrt{T}},$$

where we require $T \geq \frac{d}{4}$ in order to enforce $\delta \leq 1/6$.

2.5.6 Convex Lipschitz functions for $p \in (2, \infty]$: Proof of Theorems 2.4.2 and 2.4.5

Next, we establish Theorems 2.4.2 and 2.4.5, the analogous lower bounds on optimization of convex functions when $p \in [2, \infty)$. We again consider the class of functions $\mathcal{G}_{\mathsf{c}}$ defined in (2.13), this time with parameters $a := 2B\delta/d$ and $b := D/(2d^{1/p})$ That is, here $\mathcal{X} = \{x : \|x\|_\infty \leq D/(2d^{1/p})\}$ and

$$g_v(x) := \frac{2B\delta}{d} \sum_{i=1}^{d} \left|x(i) - \frac{v(i)D}{2d^{1/p}}\right|. \quad \forall x \in \mathcal{X}, v \in \{-1, 1\}^d.$$

It follows that the gradient of g_v is equal to $-2B\delta v/d$ at every $x \in \mathcal{X}$.

For each g_v, consider then the gradient oracle O_v which outputs 0 in all but a randomly chosen coordinate; if that coordinate is i, it takes values $-B$ and B with probabilities $\frac{1+2\delta v(i))}{2d}$ and $\frac{1-2\delta v(i)}{2d}$, respectively, for some parameter $\delta \in (0, 1/6]$ to be suitably chosen later. Thus, the oracle is no longer a product distribution.

Clearly, $\mathcal{X} \in \mathbb{X}_p(D)$ and all the functions g_v and the corresponding oracles O_v belong to the convex function family $\mathcal{O}_{\mathsf{c},p}$. Proceeding as in Section 2.5.5, we get for a uniformly distributed V that

$$\mathbb{E}\left[g_V(x_T) - g_V(x_V^*)\right] \geq \frac{DB\delta}{3d^{1/p}} \cdot \left[1 - \sqrt{\frac{1}{d}\sum_{i=1}^{d} 2I(V(i) \wedge Y^T)}\right]. \tag{2.21}$$

Further, proceeding as in the previous section to bound the average information, we note that the oracle outputs independent samples from the distribution $\mathbf{p}_v$ on $\Omega := \{-B, 0, B\}^d$ at each time. It can be checked easily that, for every $i \in [d]$, the expression of the ratio $\frac{\mathbf{p}_{v\oplus i}}{p_v}$ given in (2.19) still holds (as only the denominators of the Bernoulli parameters have changed, and they cancel out in the ratio), and that Assumption 2.5.4 is satisfied with the following γ, $\phi_{i,v}$s:

$$\gamma := \frac{1}{\sqrt{d}} \cdot \frac{4\delta}{\sqrt{1-4\delta^2}}, \qquad \phi_{i,v}(x) := \sqrt{d} \cdot \frac{v(i)\,\mathrm{sign}(x(i)) + 2\delta}{\sqrt{1-4\delta^2}}. \tag{2.22}$$

Observe the difference with the expressions from the previous section (specifically, (2.20)), as the orthonormality assumption now crucially introduces a factor $1/\sqrt{d}$ in the value of γ. Finally, because we will enforce $\delta \leq 1/6$ we also can take $\kappa_{\mathcal{W}_{\mathtt{com},r}} = 2$ for the communication constraints, as before. We remark that $\phi_{i,v}(X)$ is no longer subgaussian.

Completing the proof of Theorem 2.4.2 (LDP constraints). From Theorem 2.5.7 and the value of γ above, we get, analogously to the previous section,

$$\sum_{i=1}^{d} I\Big(V(i) \wedge Y^T\Big) \leq c \cdot \frac{T\delta^2\varepsilon^2}{d},$$

where $c := 9e(e-1)^2$ (recalling that $\varepsilon \in (0,1]$ and $\delta \leq 1/6$). Substituting this bound on mutual information in (2.21), we obtain

$$\mathbb{E}\left[g_V(x_T) - g_V(x_V^*)\right] \geq \frac{DB\delta}{3d^{1/p}} \left[1 - \sqrt{\frac{2cT\delta^2\varepsilon^2}{d^2}}\right].$$

Optimizing over δ, we set $\delta := \sqrt{\frac{d^2}{8cT\varepsilon^2}}$ and get

$$\mathbb{E}\left[g_V(x_T) - g_V(x_V^*)\right] \geq \frac{1}{12\sqrt{2c}} \cdot \frac{DBd^{1/2-1/p}}{\sqrt{T}} \cdot \sqrt{\frac{d}{\varepsilon^2}},$$

where we require $T \geq \frac{9}{2c} \cdot \frac{d^2}{\varepsilon^2}$ in order to guarantee $\delta \leq 1/6$. This concludes the proof. □

Completing the proof of Theorem 2.4.5 (Communication constraints). We prove the two parts of the lower bounds separately, starting with the first. From Theorem 2.5.8 and the setting of γ and $\kappa_{\mathcal{W}}$ as above, we have

$$\sum_{i=1}^{d} I\left(V(i) \wedge Y^T\right) \leq \frac{16}{1-4\delta^2} \cdot T\delta^2 \frac{2^r \wedge d}{d},$$

whereby, using $\delta \leq 1/6$,

$$\sum_{i=1}^{d} I\left(V(i) \wedge Y^T\right) \leq 18T\delta^2 \frac{2^r \wedge d}{d}.$$

Substituting this bound on mutual information in (2.21), we have

$$\mathbb{E}\left[g_V(x_T) - g_V(x_V^*)\right] \geq \frac{DB\delta}{3d^{1/p}} \cdot \left[1 - \frac{1}{\sqrt{d}} \cdot \sqrt{36T\delta^2 \frac{2^r \wedge d}{d}}\right].$$

Setting $\delta := \sqrt{\frac{d^2}{144(2^r \wedge d)T}}$, we finally get

$$\mathbb{E}\left[g_V(x_T) - g_V(x_V^*)\right] \geq \frac{1}{72} \cdot \frac{DBd^{1/2-1/p}}{\sqrt{T}} \cdot \sqrt{\frac{d}{2^r \wedge d}},$$

where we require $T \geq \frac{1}{4} \cdot \frac{d^2}{2^r \wedge d}$ in order to guarantee $\delta \leq 1/6$.

The second bound follows by noting that the lower bound in Theorem 2.4.4 is still valid. Finally, since $\frac{d^2}{2^r \wedge d} \geq \frac{d}{r}$ for all $1 \leq r \leq d$, both bounds apply whenever $T = \Omega\Big(\frac{d^2}{2^r \wedge d}\Big)$, as claimed. □

2.5.7 Strongly convex functions: Proof of Theorem 2.4.3, 2.4.6, and 2.4.8

Next, we establish our lower bounds on strongly convex optimization. We consider the class of functions $\mathcal{G}_{\mathtt{sc}}$ defined in (2.14) with parameters $a := B/(\sqrt{d}b)$ and $b := D/(2\sqrt{d})$. That is, $\mathcal{X} = \{x : \|x\|_\infty \leq D/(2\sqrt{d})\}$, and, for every $x \in \mathcal{X}$ and $v \in \{-1, 1\}^d$,

$$g_v(x) := \frac{B}{b \cdot \sqrt{d}} \sum_{i=1}^{d} \frac{1 + 2\delta v(i)}{2} f_i^+(x) + \frac{1 - 2\delta v(i)}{2} f_i^-(x),$$

and

$$f_i^+(x) = \theta b|x(i) + b| + \frac{1-\theta}{4}(x(i) + b)^2 \text{ and } f_i^-(x) = \theta b|x(i) - b| + \frac{1-\theta}{4}(x(i) - b)^2.$$

Moreover, in order to ensure that the every g_v is γ-strongly convex, we choose $\theta := 1 - \frac{4\gamma}{a}$ (so that $a\frac{1-\theta}{4} = \gamma$). It remains to specify δ, which we will choose such that $0 < \delta \leq \frac{1}{2} \cdot \frac{1-\theta}{1+\theta}$ in the course of the proof.

For each g_v, consider the gradient oracle O_v which on query x outputs independent values for each coordinate, with the ith coordinate taking values $\frac{B}{b\sqrt{d}} \cdot \frac{\partial f_i^+(x)}{\partial x_i}$ and $\frac{B}{b\sqrt{d}} \cdot \frac{\partial f_i^-(x)}{\partial x_i}$ with probabilities $\frac{1+2\delta v(i))}{2}$ and $\frac{1-2\delta v(i)}{2}$, respectively.

Note that we have $\left|\frac{\partial f_i^+(x)}{\partial x_i}\right|, \left|\frac{\partial f_i^-(x)}{\partial x_i}\right| \leq b$ for all x and i, and therefore the gradient estimate $\hat{g}(x)$ supplied by the oracle O_v at x satisfies $\|\hat{g}(x)\|_2^2 \leq B^2$ with probability one, for every query $x \in \mathcal{X}$. Further, it is clear that $\mathcal{X} \in \mathbb{X}_2(D)$ and all the functions g_v and the corresponding oracles O_v belong to the strongly convex function family $\mathcal{O}_{\mathtt{sc}}$.

Using our assumption that $\delta \leq \frac{1}{2} \cdot \frac{1-\theta}{1+\theta}$, we obtain by Lemma 2.5.11

$$\psi(\mathcal{G}_{\mathtt{sc}}) \geq \frac{2ab^2\delta^2}{1-\theta} = \frac{2a^2b^2\delta^2}{4\gamma} = \frac{B^2\delta^2}{2d\gamma}, \tag{2.23}$$

where we first plug in $a(1-\theta) = 4\gamma$ and then substitute for a and b.

Completing the proof of Theorem 2.4.6 (Communication constraints). By proceeding as in Section 2.5.5, from Lemma 2.5.3 and using the inequality (2.23) above, we have

$$\sup_{\mathcal{X}\in\mathbb{X}_2(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{sc}}, T, \mathcal{W}_{\mathtt{com},r}) \geq \frac{B^2\delta^2}{12\gamma}\left[1 - \sqrt{\frac{2}{d}\sum_{i=1}^{d} I(V(i) \wedge Y^T)}\right]. \tag{2.24}$$

It remains to bound $\sum_{i=1}^{d} I\Big(V(i) \wedge Y^T\Big)$ to complete the proof. Note that unlike the proof in Section 2.5.5, the gradient estimates have different distributions for different x. However, for a point x we can still express the gradient estimate $\hat{z}(x)$ of $g_v(x)$ given by O_v as follows: abbreviating $f_i'^{+}(x) := \frac{\partial f_i^+(x)}{\partial x_i}$ and $f_i'^{-}(x) := \frac{\partial f_i^-(x)}{\partial x_i}$, we have

$$\hat{z}(x)(i) = aZ_i f_i'^{+}(x) + a(1-Z_i) f_i'^{-}(x), \tag{2.25}$$

where $Z_i \sim \mathrm{Ber}(1/2 + \delta v(i))$ and the Z_i's are mutually independent. Thus, for a fixed x, $\hat{z}(x)$ can be viewed as a function of $\{Z_i\}_{i\in[d]}$. Furthermore, for a channel $W \in \mathcal{W}_{\mathtt{com},r}$ consider the channel W'_x which first passes the Bernoulli vector $\{Z_i\}_{i\in[d]}$ through the function $\hat{z}(x)(i)$ and the resulting output is passed through the channel W. This composed channel W_x belongs to $\mathcal{W}_{\mathtt{com},r}$, too.

Therefore, we can treat the independent copies of $Z \sim \mathbf{p}_v$ revealed by the oracle as i.i.d. random variables $X_1, ..., X_n$ in Section 2.5.3. Further, note that at time t, the query is for a point x_t which is a random function of Y^{t-1}, and so, Y^T can be viewed as the channel outputs with adaptively selected channels from $\mathcal{W}_{\mathtt{com},r}$. Thus, we can apply the bounds in Theorem 2.5.8.

Doing so, analogously to the computations in Section 2.5.5,[7] we get

$$\sum_{i\in[d]} I(v(i) \wedge \{Y_i\}_{i\in[T]}) \leq \sum_{i=1}^{d} I\Big(V(i) \wedge Y^T\Big) \leq c\delta^2 rT,$$

[7] As we have, in both cases, unknown Bernoulli product distribution over $\{-1,1\}^d$ with bias vector $\frac{1}{2} + \delta v$.

for an appropriate constant c, which in view of (2.24) leads to

$$\sup_{\mathcal{X}\in\mathbb{X}_2(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{sc}}, T, \mathcal{W}_{\mathtt{com},r}) \geq \frac{B^2\delta^2}{12\gamma}\cdot\left[1 - \frac{1}{\sqrt{d}}\cdot\sqrt{2cT\delta^2 r}\right] = \frac{1}{192c}\cdot\frac{B^2}{\gamma T}\cdot\frac{d}{r}$$

the last equality by setting $\delta := \sqrt{\frac{d}{8cTr}}$. Finally, observe that this choice of δ indeed satisfies $\delta < \frac{1}{2}\cdot\frac{1-\theta}{1+\theta}$, as long as $T \geq 2c\cdot\frac{B^2}{D^2}\cdot\frac{d}{\gamma^2 r}$. This completes the proof. □

Completing the proof of Theorem 2.4.3 (Privacy constraints). Proceeding as in the proof of Theorem 2.4.6 above, we have the analogue of (2.24),

$$\sup_{\mathcal{X}\in\mathbb{X}_2(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{sc}}, T, \mathcal{W}_{\mathtt{priv},\varepsilon}) \geq \frac{B^2\delta^2}{12\gamma}\left[1 - \sqrt{\frac{2}{d}\sum_{i=1}^{d} I(V(i) \wedge Y^T)}\right].$$

As stated in the proof of Theorem 2.4.6, the privatization of the gradient $\hat{z}(x)$ can be viewed as first preprocessing $\{Z_i\}_{i\in[d]}$ and the passing the preprocessed output through the LDP channel. Such a composed channel also belongs to $\mathcal{W}_{\mathtt{priv},p}$. Thus, we can apply the bound in Theorem 2.5.7 and proceed as in the proof of Theorem 2.4.1 to obtain

$$\sum_{i=1}^{d} I\left(V(i) \wedge Y^T\right) \leq cT\delta^2\varepsilon^2$$

where $c > 0$ is an absolute constant. Choosing $\delta := \sqrt{\frac{d}{8cT\varepsilon^2}}$, which makes 2δ less than $\frac{1-\theta}{1+\theta}$ for $T \geq 2c\cdot\frac{B^2}{D^2}\cdot\frac{d}{\gamma^2\varepsilon^2}$, for some universal positive constant c, then yields

$$\sup_{\mathcal{X}\in\mathbb{X}_2(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{sc}}, T, \mathcal{W}_{\mathtt{priv},\varepsilon}) \geq c_0\cdot\frac{B^2}{\gamma T}\cdot\frac{d}{\varepsilon^2}$$

for some absolute constant $c_0 > 0$, concluding the proof. □

Completing the proof of Theorem 2.4.8 (Computational constraints). As before, we can get

$$\sup_{\mathcal{X}\in\mathbb{X}_2(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{sc}}, T, \mathcal{W}_{\mathtt{obl}}) \geq \frac{B^2\delta^2}{12\gamma}\left[1 - \sqrt{\frac{2}{d}\sum_{i=1}^{d} I(V(i) \wedge Y^T)}\right].$$

Recall that we can express the subgradient estimate as in (2.25). Note that for an oblivious sampling channel W_t used at time t, specified by a probability vector $(p_j)_{j\in[d]}$, the output is given by

$$Y_i = (aZ_{J_t} f'^{+}_{J_t}(x) + a(1 - Z_{J_t}) f'^{-}_{J_t}(x))e_{J_t},$$

where $J_t = j$ with probability p_j. To proceed, we observe that the Markov relation V—$\{Z_{J_t}, J_t\}_{t\in[T]}$—$Y^T$ holds. Indeed, we can confirm this by noting that $\{Z_{J_t}\}_{t\in[T]}$ are generated i.i.d. from $\mathbf{p}_V$ and, for each $t \in [T]$, Y_t is a function of (Y^{t-1}, Z_{J_t}, J_t) and a local randomness U available only to the optimization algorithm which is independent jointly of V and $\{Z_{J_t}, J_t\}_{t\in[T]}$. It follows that Y^T itself is a function of U and $\{Z_{J_t}, J_t\}_{t\in[T]}$, which gives

$$I\Big(V \wedge Y^T \mid \{Z_{J_t}, J_t\}_{t\in[T]}\Big) \le I\Big(V \wedge U \mid \{Z_{J_t}, J_t\}_{t\in[T]}\Big) = 0. \tag{2.26}$$

From the previous observation, we also get that the Markov relation $V(i)$—$\{Z_{J_t}, J_t\}_{t\in[T]}$—$Y^T$ holds for every $i \in [d]$. Thus, by the data processing inequality for mutual information, we have

$$\sum_{i=1}^{d} I\Big(V(i) \wedge Y^T\Big) \le \sum_{i=1}^{d} I\Big(V(i) \wedge \{Z_{J_t}, J_t\}_{t\in[T]}\Big).$$

Now since vector $(Z_j)_{j\in[d]}$ is a Bernoulli vector, the mutual information on the right-side can be bounded by the same computation as in the proof of Theorem 2.4.7 using Theorem 2.5.9. This follows by observing that for all $t \in [T]$, (Z_{J_t}, J_t) is a function of $Z_{J_t}e_{J_t}$, which in turn can be seen as a output of the oblivious sampling channel for an input vector $(Z_j)_{j\in[d]}$. Therefore, we have

$$\sum_{i=1}^{d} I\Big(V(i) \wedge Y^T\Big) \le cT\delta^2$$

for an appropriate constant c and $\delta \leq \frac{1}{6}$, which in view of (2.24) leads to

$$\sup_{\mathcal{X} \in \mathbb{X}_2(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{sc}}, T, \mathcal{W}_{\mathtt{com},r}) \geq \frac{B^2\delta^2}{12\gamma}\left[1 - \frac{1}{\sqrt{d}} \cdot \sqrt{2cT\delta^2}\right] = \frac{1}{c_0} \cdot \frac{dB^2}{\gamma T},$$

where the last identity is obtained by setting $\delta := c_1\sqrt{\frac{d}{T}}$, where c_0 and c_1 are universal positive constants. Finally, observe that this choice of δ indeed satisfies $\delta < \frac{1}{2} \cdot \frac{1-\theta}{1+\theta}$, as long as $T \geq c_2 \cdot \frac{B^2}{D^2} \cdot \frac{d^2}{\gamma}$, for some universal positive constant c_2. This completes the proof. □

2.6 Concluding Remarks

In this chapter, we derived lower bounds on the optimization error incurred by any first-order algorithm when the stochastic gradients used by the optimization algorithm need to be further processed to satisfy information constraints. We also saw that the gradient processing schemes proposed in the literature and appropriate first-order algorithms almost match our derived lower bounds in the case of privacy and computational constraints. Therefore, in the rest of the first part, we will develop gradient compression algorithms to match the lower bounds for communication-constrained optimization.

Chapter 3

Communication-Constrained Optimization over Euclidean Space

3.1 Synopsis

For communication-constrained optimization over the Euclidean Space, where the subgradient estimate's norm is almost surely bounded, we present Rotated Adaptive Tetra-iterated Quantizer (RATQ), a fixed-length quantizer for subgradient estimates. RATQ is easy to implement and involves only a Hadamard transform computation and adaptive uniform quantization with appropriately chosen dynamic ranges. We show that RATQ along with PSGD achieves the lower bound for communication-constrained optimization over Euclidean Space.

We further extend our results for communication-Constrained optimization over Euclidean space when the subgradient estimates are mean square bounded. In this setting, we use a gain-shape subgradient quantizer which separately quantizes the Euclidean norm and uses RATQ to quantize the normalized unit norm vector. We establish lower bounds for performance of any optimization procedure and shape quantizer, when used with a uniform gain quantizer. Finally, we propose an adaptive quantizer for gain which when used with RATQ for shape quantizer outperforms uniform gain quantization and is, in fact, close to optimal.

The results presented in this chapter are from [68] and [69].

3.2 Introduction

In this chapter, we develop new algorithms to match the lower bounds for communication-constrained optimization over Euclidean Space. More precisely, we study communication-constrained optimization for convex and ℓ_2 lipschitz function family as well as strongly convex and ℓ_2 lipschitz function family. We consider two *oracle models*: the first where the subgradient estimate's Euclidean norm is *almost surely bounded* and the second where it is *mean square bounded*. While our lower bounds in Chapter 2 were derived for the simpler, almost surely bounded oracles, where the subgradient estimates have their noise almost surely bounded. In this setting, we also develop algorithms for the more general mean square bounded oracle. Our main contributions include new quantizers for the two oracle models and theoretical insights into the limitations imposed by heavy-tailed gradient distributions admitted under the mean square bounded oracles. A more specific description of our results and their relation to prior work is provided below.

3.2.1 Main contributions

We start with almost surely bounded oracles and consider communication-constrained optimization for convex and ℓ_2 lipschitz function family. Our precision-dependent lower bound in Theorem 2.4.4 shows that no optimization protocol using a first order oracle and gradient updates of precision $r < d$ bits can have optimization error smaller than roughly $\sqrt{d}/\sqrt{rT}$. In particular, we need precision exceeding $\Omega(d)$ bits to get the classic convergence rate of $1/\sqrt{T}$ for convex functions. As our main contribution, we propose a new fixed-length quantizer we term *Rotated Adaptive Tetra-iterated Quantizer* (RATQ) that along with projected subgradient descent (PSGD) is merely a factor of $O(\log\log\log\log^* d)$ far from this minimum precision required to attain the $O(1/\sqrt{T})$ convergence rate. In a different setting, when the precision is fixed upfront to r, we modify RATQ by roughly quantizing and sending only a subset of coordinates of the rotated vector. We show

that this modified version of RATQ is only a factor $O(\log \log^* d)$ far from the optimal convergence rate. For almost sure bounded oracles, all our results for convex and ℓ_2 lipschitz family are then extended to strongly convex and ℓ_2 lipschitz family.

For mean square bounded oracles, we state our results for convex and ℓ_2 lipschitz family. However, most of our results in this setting can be extended to strongly convex family *mutatis mutandis*. In this setting, most of the prior work makes an additional assumption that the gradient norm can be expressed using only a finite number of bits without accruing any quantization error. One of our main contributions for mean square bounded oracles is to analyse the quantization error without any such additional assumptions. For such oracles, we establish an information theoretic lower bound in Section 3.5.1 which shows (using a heavy-tailed oracle) that the precision used for gain[1] quantizer must exceed $\log T$ when the gain is quantized uniformly for T iterations and we seek $O(1/\sqrt{T})$ optimization accuracy. Thus, if 32 bits are used to describe the gain using a uniform quantizer, they will suffice for roughly a billion iterations. Interestingly, we present a new, adaptive gain quantizer which can attain the same performance using only $\log \log T$ bits for quantizing gain. In particular, using our scheme, only 5 bits assigned for describing gain will suffice for a billion iterations; these many bits will work for less than 100 iterations using uniform gain quantizers. In a different setting, when the precision is fixed upfront we propose a quantizer which along with PSGD achieves the almost optimal convergence rate.

3.2.2 Remarks on techniques

In this work we use adaptive quantizers with multiple dynamic-ranges $\{[-M_i, M_i] : i \in [h]\}$, with possibly a different dynamic range chosen for each coordinate. Once a dynamic-range $[-M_i, M_i]$ is chosen for a coordinate, the coordinate is quantized uniformly within this dynamic-range using k levels. Using a different dynamic-range for each coordinate allows us to reduce error per coordinate, but costs us in communication since we need to communicate which M_i is used for each coordinate. In devising our scheme, we need to

[1] In the vector quantization literature, the norm of the vector to be quantized is called the *gain* and vector normalized by this norm is called the *shape*.

carefully balance this tradeoff. We do this by taking recourse to the following observation: when the same dynamic range is chosen for all coordinates, the mean square error per coordinate roughly grows as

$$O\left(\frac{\sum_{i\in[h]} M_i^2 \cdot p(M_{i-1})}{(k-1)^2}\right), \tag{3.1}$$

where $p(M)$ is the probability of the ℓ_∞ norm of the input vector exceeding M and k denotes the number of levels of the uniform quantizer. This observation allows us to relate the mean square error to the tail-probabilities of the ℓ_∞ norm of the input vector. In particular, we exploit it to decide on the subvectors which we quantize using the same dynamic range.

As an aside, we believe that this approach and equation (3.1), in particular, will yield very efficient rate-distortion codes for various sources with different tail probabilities, answering questions of fundamental interest and having many applications. We point out an application to the classic Gaussian rate-distortion problem in Chapter 6.

We use another classic trick (see [34]): we transform the input vector before we apply our adaptive quantizer. In particular, we use a randomized transform that expresses the input vector over a random basis. The specific choice of our random transform is determined by our assumption for the gradients, namely that their ℓ_2 norms are almost surely bounded by B.

Drawing from these ideas, we propose the quantizer RATQ for quantizing random vectors with ℓ_2 norm almost surely bounded by B.

We remark that using an adaptively chosen dynamic-range can alternatively be implemented by transforming the input using a monotone function. This, too, is a classic technique in quantization known as *companding* (*cf.* [34]). Companding is known as a popular alternative to entropic coding for fixed-length codes. However, to the best of our knowledge, this work is the first to combine it with other techniques and rigorously analyze it for the ℓ_2 norm bounded vector quantization problem. Perhaps it is a bit surprising that this combination of classic technique was not analysed for constructing an efficient covering of the unit Euclidean ball, the problem underlying our quantization problem.

Moving to oracles with mean square bounded ℓ_2 norms, we take recourse to gain-shape quantizers and quantize the (normalized) shape vector using RATQ. However, unlike prior work, we rigorously treat gain quantization. Our proposed quantizer for gain is once again an adaptive quantizer.

Our lower bounds derived in Chapter 2 use almost surely bounded oracles as the difficult oracle. However, this only allows us to obtain lower bounds for the almost surely bounded setting. For the mean square bounded setting, we need a new construction with "heavy tails". In particular, our proposed heavy-tailed construction shows a bottleneck for uniform gain quantizers which can be circumvented by our proposed quantizer, thereby establishing a strict improvement over uniform gain quantizers.

3.2.3 Prior work

Our work is motivated by the results in [9, 88], and we elaborate on the connection. Specifically, [9] considers a problem very similar to ours. The paper [88] considers the related problem of distributed mean estimation – we elaborate on the distributed mean estimation results in Chapter 5 – but the quantizer and its analysis is directly applicable to distributed optimization. The two papers present different quantizers that encode each input using a variable number of bits. Both these quantizers require the optimal expected precision to achieve the $1/\sqrt{T}$ convergence rate for almost surely bounded oracles. However, their worst-case (fixed-length) performance maybe suboptimal. Our proposed quantizer RATQ requires a precision only slightly more than the optimal precision to achieve the $1/\sqrt{T}$ convergence rate for almost surely bounded oracles, while still being fixed-length. Moreover, in the slightly different setting of operating for any precision constraint r less than the dimension, we significantly improve upon the current state-of-the-art.

In fact, the problem of designing fixed-length quantizers for almost surely bounded oracles is closely related to designing small-size covering for the Euclidean unit ball. There has been a longstanding interest in this problem in the vector quantization and information theory literature (*cf.* [19, 27, 34, 47, 55, 92]).

In a slightly different direction, a seminal, but perhaps not so widely known, result

of [100] provides a very simple universal quantizer for random vectors with independent and identically distributed (*iid*) coordinates, with each coordinate almost surely bounded. In this scheme, we first quantize each coordinate uniformly, separately using a "scalar-quantizer," and then apply a universal entropic compression scheme to the quantized vector. We note that the variable-length schemes proposed in [9,88] are very similar, albeit with a specific choice of the entropic compression scheme.

All these schemes (the ones in [9, 88, 100]) are variable-length schemes, while it is desirable to get a fixed-length scheme for the ease of both protocol and hardware implementation. We remark that indeed [88] presents an interesting randomly-rotate and quantize fixed-length scheme, but it still requires communicating $O(\log \log d)$ times more than the optimal fixed-length quantizer for the unit Euclidean ball given in [92]. To the best of our knowledge, prior to our work, the quantizer in [88] is the best known efficient fixed-length quantizer for the unit Euclidean ball.

In fact, a randomized orthogonal transform scheme similar to that in [88] appeared almost concurrently in [39] as well, where an analysis for Gaussian source is presented. However, a rate-distortion analysis has not been done in [39]. Remarkably, an early instance of the "rotated dithering" scheme for distributing energy equally appears in the image compression literature in [75], albeit without formal error or performance analysis. Another interesting scheme was proposed in [8] where nonuniform quantization (using *companding*) was combined with dithering. Our adaptive choice of dynamic range for uniform quantizers is similar, in essence, to companding. But our scheme differs from the one in [8] in several ways: First, [8] uses the knowledge of input distribution to design their companding function, whereas we only need knowledge of the tail behaviour of the input distribution in our setting; second, we apply a random rotation to our input leading to a universal quantizer, which is not needed in [8]; and finally, the specific structure of our quantizer with adaptive dynamic ranges makes it amenable to mean square error analysis for a large variety of sources.

Another scheme, similar, in spirit, to [88], appears earlier in [62]. In this scheme, the input vector is preprocessed using a redundant system of vectors (the resulting

representation is called Kashin's representation) instead of random rotation as in [88]. In theory, if the underlying system of vectors satisfies certain desirable properties, then preprocessing the vector in this manner and then uniformly quantizing each coordinate in the representation will lead to an orderwise optimal fixed-length quantizer for the unit ball. Unfortunately, [62] provides only a randomized constructions[2] for the system of vectors that satisfy the aforementioned properties with high probability. Thus, the scheme in [62] is not explicit. Further, as a side remark, we note that the preprocessing step in [62] requires $O(d^2 \log d)$ real operations, much worse than the $O(d \log d)$ real operations required by RATQ.

Another recent independent work [33] presents a different scheme where a different random transform is used instead of random rotation. The optimal scheme in [33] is similar to the one in [62] and in essence to classical information-theoretic schemes (*cf.* [92], [55]). Specifically, [33] provides a randomized algorithm that outputs the orderwise optimal quantizer for the unit ball without an explicit construction. Moreover, the time complexity of the overall quantization procedure in [33] is much worse than RATQ.

Returning to the literature on quantizers for first order stochastic optimization, prior works including [9] remain vague about the analysis for mean square bounded oracles. Most of the works use gain-shape quantizers that separately quantize the Euclidean norm (*gain*) and the normalized vector (*shape*). But they operate under an engineering assumption: "the standard 32 bit precision suffices for describing the gain." One of our goals in this work is to carefully examine this heuristic. For instance, can we use a simple uniform quantizer for gain with 32 bits, or even say 8 bits?

Independent of our work, a non-uniform quantizer similar to the one we use for gain-quantization with geometrically increasing dynamic-ranges appears in [78]. However, there are some key differences between the two quantizers. First, note that we use this quantizer for gain-quantization, while [78] uses it to quantize the shape. Second, in the case of our quantizer the geometrically growing dynamic ranges are further quantized uniformly

[2]Note that randomly producing the optimal quantizer with high probability is different from constructing the optimal random quantizer, as we do. The former only gives high probability guarantees for bounds on loss, but need not yield a bound for expected loss.

whereas [78] chooses to use geometrically growing points as final quantization points. Third, the analysis of mean square quantization error in this work and [78] differ significantly. In particular, our mean square analysis follows from the general principle stated in (3.1), whereas [78] builds upon the analysis of QSGD in [9]. Finally, the setting considered in [78] is similar to that of [9] where quantization is followed by entropic compression. In particular, the fixed-length performance may be suboptimal for almost surely bounded oracles and mean square bounded oracles are not handled.

Organization

We formalize our problem in the next section and describe our results for almost surely and mean square bounded oracles in Sections 3.4 and 3.5, respectively, along with some of the shorter proofs. The more elaborate proofs are provided in Section 3.6, with additional details relegated to Section 3.7.

3.3 Setup and preliminaries

3.3.1 Setup

In the next two chapters, we develop efficient subgradient compression algorithms for the setting of communication-constrained first-order optimization described in the previous Chapter. Our domain $\mathcal{X}$ throughout this chapter has Euclidean diameter less than D. That is,

$$\mathcal{X} \in \mathbb{X}_2(D) = \{\mathcal{X}' : \sup_{x,y\in\mathcal{X}'} \|x - y\|_2 \leq D.\} \tag{3.2}$$

For the domain of optimization $\mathcal{X}$, we develop subgradient compression schemes for function and oracle families given by $\mathcal{O}_{\mathsf{c},2}$ and $\mathcal{O}_{\mathsf{sc}}$, which are defined in Definitions 2.3.3 and 2.3.6, respectively.

We remark that the typical assumption made on the optimization literature on the oracle noise is that it is *mean square bounded*. That is, for a query point $x \in \mathcal{X}$, the oracle

random estimates of the subgradient $\hat{g}(x)$ which for all $x \in \mathcal{X}$ satisfy

$$\mathbb{E}\left[\|\hat{g}(x)\|_2^2 | x\right] \leq B^2, \tag{3.3}$$

where $\partial f(x)$ denotes the set of subgradients of f at x. In this chapter, we also want to develop schemes for mean square bounded oracles. Towards that end, we define generalization of class $\mathcal{O}_{\mathtt{c},2}^{\mathtt{m}}$ below.

Definition 3.3.1 (Convex and ℓ_2 Lipschitz function family for a mean square bounded oracle $\mathcal{O}_{\mathtt{c},2}^{\mathtt{m}}$)**.** We denote by $\mathcal{O}_{\mathtt{c},2}^{\mathtt{m}}$ the set of all pairs of functions and oracles satisfying Assumptions (2.4), (2.5), and (3.3).

Thus the assumption (2.6) is replaced by (3.3) for mean square bounded families. Clearly,

$$\mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{c},2}^{\mathtt{m}}, T, \mathcal{W}_{\mathtt{com},r)} \geq \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{c},2}, T, \mathcal{W}_{\mathtt{com},r}).$$

Therefore, the lower bounds derived in Chapter 2 derived for almost surely bounded oracles still hold for mean square bounded oracles. Although, we still derive another lower for mean square bounded oracle to point out the limitations of specific quantization procedures in Section 3.5.

3.3.2 Structure of our protocols

It will be instructive to recall the definition of the quantizer, since the the outputs of the oracle are passed through a quantizer. An *r-bit quantizer* consists of randomized mappings[3] $(Q^{\mathtt{e}}, Q^{\mathtt{d}})$ with the encoder mapping $Q^{\mathtt{e}} : \mathbb{R}^d \to \{0,1\}^r$ and the decoder mapping $Q^{\mathtt{d}} : \{0,1\}^r \to \mathbb{R}^d$. The overall quantizer is given by the composition mapping $Q = Q^{\mathtt{d}} \circ Q^{\mathtt{e}}$.

In both this Chapter and the next Chapter, we restrict to *memoryless* quantization schemes where the same quantizer will be applied to each new gradient vector, without using any information from the previous updates. Specifically, at each instant t and

[3]We can use public randomness U for randomizing.

for any precision r, the quantizers in $\mathcal{W}_r$ do not use any information from the previous time instants to quantize the subgradient outputted by O at t. Our primary motivation for restriction to memoryless quantization schemes is ease of implementation and their application to other problems, as we see in Chapter 5 and 6.

Thus our channel selection strategy S is nonadaptive (recall definition 2.3.2) and is simply denoted by the quantizer Q we choose to use. Thus the optimization error for a function f and oracle O when employing a first order optimization π and quantizer Q is given by

$$\mathcal{E}(f, O, \pi, Q) = \mathbb{E}\left[f(x_T)\right] - \mathbb{E}\left[f(x^*\right].$$

3.3.3 Quantizer performance for finite precision optimization

Our overall optimization protocol throughout is the *projected SGD* (PSGD) (see [15]). In fact, we establish lower bound showing roughly the optimality of PSGD with our quantizers.

In PSGD the standard SGD updates are projected back to the domain using the projection map $\Gamma_{\mathcal{X}}$ given by $\Gamma_{\mathcal{X}}(y) := \min_{x\in\mathcal{X}} \|x - y\|_2$. We use the *quantized PSGD* algorithm described in Algorithm 3.1.

Require: $x_0 \in \mathcal{X}, \eta \in \mathbb{R}^+$, T and access to composed oracle QO

1: **for** $t = 0$ to $T - 1$ **do**

$$x_{t+1} = \Gamma_{\mathcal{X}}\left(x_t - \eta_t Q(\hat{g}(x_t))\right)$$

2: **Output:** $\frac{1}{T} \cdot \sum_{t=1}^{T} x_t$

Algorithm 3.1: Quantized PSGD with quantizer Q

The quantized output $Q(\hat{g}(x_t))$, too, constitutes a noisy oracle, but it can be biased for mean square bounded oracles. Though biased first-order oracles were considered in [45], the effect of quantizer-bias has not been studied in the past. The performance of a quantizer Q, when it is used with PSGD for mean square bounded oracles, is controlled by the

worst-case L_2 norm $\alpha_2^{\mathtt{m}}(Q)$ of its output and the worst-case bias $\beta_2^{\mathtt{m}}(Q)$ defined as[4]

$$\begin{aligned}\alpha_2^{\mathtt{m}}(Q) &:= \sup_{Y\in\mathbb{R}^d:\mathbb{E}\left[\|Y\|_2^2\right]\le B^2} \sqrt{\mathbb{E}\left[\|Q(Y)\|_2^2\right]},\\ \beta_2^{\mathtt{m}}(Q) &:= \sup_{Y\in\mathbb{R}^d:\mathbb{E}\left[\|Y\|_2^2\right]\le B^2} \|\mathbb{E}\left[Y-Q(Y)\right]\|_2. \end{aligned} \tag{3.4}$$

The corresponding quantities for almost surely bounded oracles are

$$\begin{aligned}\alpha_2(Q) &:= \sup_{Y\in\mathbb{R}^d:\|Y\|_2\le B \text{ a.s.}} \sqrt{\mathbb{E}\left[\|Q(Y)\|_2^2\right]},\\ \beta_2(Q) &:= \sup_{Y\in\mathbb{R}^d:\|Y\|_2\le B \text{ a.s.}} \|\mathbb{E}\left[Y-Q(Y)\right]\|_2. \end{aligned} \tag{3.5}$$

Using a slight modification of the standard proof of convergence for PSGD, we get the following result for convex functions.

Theorem 3.3.2. *Let the domain $\mathcal{X}$ satisfy 3.2. For any quantizer Q, the output x_T of optimization protocol π given in Algorithm 3.1 satisfies*

$$\begin{aligned}\sup_{(f,O)\in\mathcal{O}_{\mathtt{c},2}} \mathcal{E}(f,O,\pi,Q) &\le D\left(\frac{\alpha_2(Q)}{\sqrt{T}}+\beta_2(Q)\right),\\ \sup_{(f,O)\in\mathcal{O}^{\mathtt{m}}_{\mathtt{c},2}} \mathcal{E}(f,O,\pi,Q) &\le D\left(\frac{\alpha_2^{\mathtt{m}}(Q)}{\sqrt{T}}+\beta_2^{\mathtt{m}}(Q)\right),\end{aligned}$$

when the parameter $\eta_t=\eta$, for all t, is set to $D/(\alpha_2(Q)\sqrt{T})$ and $D/(\alpha_2^{\mathtt{m}}(Q)\sqrt{T})$, respectively.

See Section 3.6.1 for the proof.

Remark 8 (Knowledge of time horizon in setting the learning rate.). Note that the choice of learning rate η_t in 3.3.2 requires the knowledge of the time horizon T. In fact, all the convergence results in this thesis require setting η_t based on the time horizon. One could employ the doubling trick –see, for instance, [70, Pg. 129] – to remove this restriction. However, this would add a multiplicative $\sqrt{\log T}$ factor to the convergence rate.

[4]We omit the dependence on B and d from our notation.

We have also have the following counterpart of the previous result for strongly convex functions.

Theorem 3.3.3. *Let the domain $\mathcal{X}$ satisfy 3.2. For any quantizer Q, the output x_T of optimization protocol π given in Algorithm 3.1 satisfies*

$$\sup_{(f,O)\in\mathcal{O}_{\mathtt{sc}}} \mathcal{E}(f,O,\pi,Q) \leq D\left(\frac{\alpha_2(Q)^2}{D\gamma T} + \beta_2(Q)\right).$$

when the parameter η_t is set to $2/\gamma(t+1)$.

See Section 3.6.2 for the proof.

Remark 9 (Choice of learning rate). For the class of convex functions, we fix the parameter η_t of Algorithm 3.1 to a constant value η, for all t. η is set to $D/(\alpha_2(Q)\sqrt{T})$ and $D/(\alpha_2^{\mathtt{m}}(Q)\sqrt{T})$ for all the results in Section 3.4 and Section 3.5, respectively. For the class of strongly convex functions, we fix the parameter $\eta_t = 2/\gamma(t+1)$ all the results in Section 3.4.

3.4 Main results for almost surely bounded oracles

Our main results will be organized along two regimes: the high-precision and the low-precision regime. For the high-precision regime, we seek to attain the optimal, classic convergence rate of $1/\sqrt{T}$, for convex functions, and $1/T$, for strongly convex functions, using the minimum precision possible. For the low-precision regime, we seek to attain the fastest convergence rate possible for a given, fixed precision r.

From our lower bounds on minmax optimization error of families $\mathcal{O}_{\mathtt{c},2}$ and $\mathcal{O}_{\mathtt{sc}}$ under communication constraints, which are derived in Theorem 2.4.4 and 2.4.6, we have the following corollaries. Our corollaries show that there is no hope of getting the desired convergence rate of $1/\sqrt{T}$ for convex function and ℓ_2 lipschitz function families ($\mathcal{O}_{\mathtt{c},2}$, $\mathcal{O}^{\mathtt{m}}_{\mathtt{c},2}$) and $1/T$ for strongly convex function families $\mathcal{O}_{\mathtt{sc}}$ by using a precision of less than d.

Corollary 3.4.1. *Let* $\mathbb{X}_2(D) = \{\mathcal{X}' : \sup_{x,y\in\mathcal{X}'} \|x-y\|_2 \le D\}$. *Then, the precision* r *must be at least* $\Omega(d)$, *for either one of the following to hold:*

$$\sup_{\mathcal{X}\in\mathbb{X}_2(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{c},2}, T, \mathcal{W}_{\mathtt{com},r}) \le \frac{DB}{\sqrt{T}}, \quad \sup_{\mathcal{X}\in\mathbb{X}_2(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}^{\mathtt{m}}_{\mathtt{c},2}, T, \mathcal{W}_{\mathtt{com},r}) \le \frac{DB}{\sqrt{T}},$$
$$\textit{and} \sup_{\mathcal{X}\in\mathbb{X}_2(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{sc}}, T, \mathcal{W}_{\mathtt{com},r}) \le \frac{B^2}{\gamma T}.$$

Thus, from Corollary 3.4.1, our quantization schemes in high-precision regime will use a precision of atleast d bits.

3.4.1 RATQ: Our quantizer for the ℓ_2 ball

We propose *Rotated Adaptive Tetra-iterated Quantizer* (RATQ) to quantize any random vector Y with $\|Y\|_2^2 \le B^2$, which is what we need for almost surely bounded oracles. RATQ first rotates the input vector, then divides the coordinates of the rotated vectors into smaller groups, and finally quantizes each subgroup-vector using a *Coordinate-wise Uniform Quantizer* (CUQ). However, the dynamic-range used for each subvector is chosen adaptively from a set of tetra-iterated levels. We call this adaptive quantizer *Adaptive Tetra-iterated Uniform Quantizer* (ATUQ), and it is the main workhorse of our construction. The encoder and decoder for RATQ are given in Algorithm 3.2 and Algorithm 3.3, respectively. The details of all the components involved are described below.

Require: Input $Y \in \mathbb{R}^d$, rotation matrix R

1: Compute $\tilde{Y} = RY$

2: **for** $i \in [d/s]$ **do**

$$\tilde{Y}_i^T = [\tilde{Y}((i-1)s+1), \cdots \tilde{Y}(\min\{is, d\})]^T$$

3: **Output:** $Q^{\mathtt{e}}_{\mathtt{at},R}(Y) = \{Q^{\mathtt{e}}_{\mathtt{at}}(\tilde{Y}_1) \cdots Q^{\mathtt{e}}_{\mathtt{at}}(\tilde{Y}_{\lceil d/s \rceil})\}$

Algorithm 3.2: Encoder $Q^{\mathtt{e}}_{\mathtt{at},R}(Y)$ for RATQ

Rotation and division into subvectors. RATQ first rotates the input vector by multiplying it with a random Hadamard matrix. Specifically, denoting by H the $d \times d$

Require: Input $\{Z_i, j_i\}$ for $i \in [\lceil d/s \rceil]$, rotation matrix R

1: $\hat{Y}^T = [Q^{\mathtt{d}}_{\mathtt{at}}(Z_1, j_1), \cdots, Q^{\mathtt{d}}_{\mathtt{at}}(Z_{\lceil d/s \rceil}, j_{\lceil d/s \rceil})]^T$

2: **Output:** $Q^{\mathtt{d}}_{\mathtt{at},R}(\{Z_i, j_i\}_{i=1}^{\lceil d/s \rceil}) = R^{-1}\hat{Y}$

Algorithm 3.3: Decoder $Q^{\mathtt{d}}_{\mathtt{at},R}(Z, j)$ for RATQ

Walsh-Hadamard Matrix (see [44])[5], define

$$R := \frac{1}{\sqrt{d}} \cdot HD, \tag{3.6}$$

where D is a diagonal matrix with each diagonal entry generated uniformly from $\{-1, +1\}$. The input vector y is multiplied by R in the rotation step. The matrix D can be generated using shared randomness between the encoder and decoder.

Next, the rotated vector of dimension d is partitioned into $\lceil d/s \rceil$ smaller subvectors. The i^{th} subvector comprises the coordinates $\{(i-1)s+1, \cdots, \min\{is, d\}\}$, for all $i \in [d/s]$. Note that the dimension of all the sub vectors except the last one is s, with the last one having a dimension of $d - s\lfloor d/s \rfloor$.

Remark 10. As an aside, we remark that preprocessing the data by such a random transform R was used by [7] for Fast Johnson Lindestrauss transform.

We now describe the advantage of random rotation. The advantage of subvector division will be clear once we describe the rest of the scheme.

Remark 11 (Advantage of random rotation). While by almost sure assumption the input vector to the quantizer is inside the Euclidean ball of radius B, to set the dynamic range[6], we need upper bounds for each coordinate of the vector. After random rotation, each coordinate of the input vector is a centered subgaussian random variable with a variance of $O(B^2/d)$, as opposed to a variance factor of $O(B^2)$, which is all that can be said for the original input vector.

[5] We assume that d is a power of 2.

[6] We mean the interval $[-M, M]$.

Coordinate-wise Uniform Quantizer (CUQ). RATQ uses CUQ as a subroutine; we describe the latter for d dimensional inputs, but it will only be applied to subvectors of lower dimension in RATQ. CUQ has a dynamic range $[-M, M]$ associated with it, and it uniformly quantizes each coordinate of the input to k-levels as long as the component is within the dynamic-range $[-M, M]$. Specifically, it partitions the interval $[-M, M]$ into parts $I_\ell := (B_{M,k}(\ell), B_{M,k}(\ell+1)]$, $\ell \in \{0, \ldots, k-1\}$, where $B_{M,k}(\ell)$ are given by

$$B_{M,k}(\ell) := -M + \ell \cdot \frac{2M}{k-1}, \quad \forall\, \ell \in \{0, \ldots, k-1\}.$$

Then, for a coordinate $y \in (B_{M,k}(\ell), B_{M,k}(\ell+1)]$, CUQ randomly outputs either $B_{M,k}(\ell)$ or $B_{M,k}(\ell+1)$ with probabilities such that the output value equals the input y in expectation. Note that each output coordinate of the CUQ encoder takes $k+1$ values – k of these symbols correspond to the k uniform levels and the additional symbol corresponds to the overflow symbol $\emptyset$. Thus we need a total precision of $d\lceil \log(k+1) \rceil$ bits to represent the output of the CUQ encoder. The encoder and decoders used in CUQ are given in Algorithms 3.4 and 3.5, respectively. In the decoder, we have set $B_{M,k}(\emptyset)$ to 0.

Require: Parameter $M \in \mathbb{R}^+$ and input $Y \in \mathbb{R}^d$

for $i \in [d]$ **do**

  if $|Y(i)| > M$ **then**

    $Z(i) = \emptyset$

  else

    for $\ell \in \{0, \ldots, k-1\}$ **do**

      if $Y(i) \in (B_{M,k}(\ell), B_{M,k+1}(\ell+1)]$ **then**

$$Z(i) = \begin{cases} \ell+1, & w.p.\ \frac{Y(i) - B_{M,k}(\ell)}{B_{M,k}(\ell+1) - B_{M,k}(\ell)} \\ \ell, & w.p.\ \frac{B_{M,k}(\ell+1) - Y(i)}{B_{M,k}(\ell+1) - B_{M,k}(\ell)} \end{cases}$$

Output: $Q_{\mathsf{u}}^{\mathsf{e}}(Y; M) = Z$

Algorithm 3.4: Encoder $Q_{\mathsf{u}}^{\mathsf{e}}(Y; M)$ of CUQ

Require: Parameter $M \in \mathbb{R}^+$, input $Z \in \{0, \ldots, k-1, \emptyset\}^d$

1: Set $\hat{Y}(i) = B_{M,k}(Z(i))$, for all $i \in [d]$

2: **Output:** $Q_{\mathtt{u}}^{\mathtt{d}}(Z; M) = \hat{Y}$

Algorithm 3.5: Decoder $Q_{\mathtt{u}}^{\mathtt{d}}(Z; M)$ of CUQ

Adaptive Tetra-iterated Uniform Quantizer (ATUQ). The quantizer ATUQ is CUQ with its dynamic-range chosen in an adaptive manner. In order to a quantize a particular input vector, it first chooses a dynamic range from $[-M_i, M_i]$, $1 \leq i \leq h$. To describe these M_is, we first define the i^{th} tetra-iteration for e, denoted by e^{*i}, recursively as follows:

$$e^{*0} := 1, \quad e^{*1} := e, \quad e^{*i} := e^{e^{*(i-1)}}, \quad i \in \mathbb{N}.$$

Also, for any non negative number b, we define $\ln^* b := \inf\{i \in \mathbb{N} : e^{*i} \geq b\}$. With this notation, the values M_is are defined in terms of m and m_0 as follows:

$$M_i^2 = m \cdot e^{*i} + m_0, \quad \forall\, i \in \{0, \ldots, h-1\},$$

where the parameters m and m_0 will be set later. ATUQ finds the smallest level M_i which bounds the infinity norm of the input vector; if no such M_i exists, it simply uses M_{h-1}. It then uses CUQ with dynamic range $[-M_i, M_i]$ to quantize the input vector. In RATQ, we apply ATUQ to each subvector. The decoder of ATUQ is simply the decoder of CUQ using the dynamic range outputted by the ATUQ encoder.

Note that in order to represent the output of ATUQ for d dimensional inputs, we need a precision of at the most $\lceil \log h \rceil + d \lceil \log(k+1) \rceil$ bits: $\lceil \log h \rceil$ bits to represent the dynamic range and at the most $d \lceil \log(k+1) \rceil$ bits to represent the output of CUQ. The encoder and decoder for ATUQ are given in Algorithms 3.6 and 3.7, respectively.

When ATUQ is applied to each subvector in RATQ, each of the $\lceil d/s \rceil$ subvectors are represented using less than $\lceil \log h \rceil + s \lceil \log(k+1) \rceil$ bits. Thus, the overall precision for

Require: Input $Y \in \mathbb{R}^d$

1: **if** $\|Y\|_\infty > M_{h-1}$ **then**

Set $M^* = M_{h-1}$

2: **else**

Set $j^* = \min\{j : \|Y\|_\infty \leq M_j\}$, $M^* = M_{j^*}$

3: Set $Z = Q_{\mathtt{u}}^{\mathtt{e}}(Y; M^*)$

4: **Output:** $Q_{\mathtt{at}}^{\mathtt{e}}(Y) = \{Z, j^*\}$

Algorithm 3.6: Encoder $Q_{\mathtt{at}}^{\mathtt{e}}(Y)$ for ATUQ

Require: Input $\{Z, j\}$ with $Z \in \{0, \ldots, k-1, \emptyset\}^d$ and $j \in \{0, \ldots h-1\}$

1: **Output:** $Q_{\mathtt{at}}^{\mathtt{d}}(Z, j) = Q_{\mathtt{u}}^{\mathtt{d}}(Z; M_j)$

Algorithm 3.7: Decoder $Q_{\mathtt{at}}^{\mathtt{d}}(Z, j)$ for ATUQ

RATQ is less than[7]

$$\lceil d/s \rceil \cdot \lceil \log h \rceil + d \lceil \log(k+1) \rceil$$

bits. The decoder of RATQ is simply formed by collecting the output of the ATUQ decoders for all the subvectors to form a d-dimensional vector, and rotating it back using the matrix R^{-1} (the inverse of the rotation matrix used at the encoder).

Remark 12 (Advantage of division into subvectors). The overall precision of RATQ allows us to understand the advantage of clubbing multiple coordinates into subvectors. Since we use the same dynamic range for all coordinates of a subvector, we save on coordinate-wise communication of the dynamic range.

Remark 13 (Mean square error of ATUQ). The per coordinate mean square error between the input to ATUQ and its output roughly grows as

$$O\left(\frac{\sum_{i\in[h]} M_i^2 \cdot p(M_{i-1})}{(k-1)^2}\right), \tag{3.7}$$

[7] log denotes the logarithm to the base 2, ln denotes logarithm to the base e.

where $p(M)$ is the probability of the ℓ_∞ norm of the input vector exceeding M and k denotes the number of levels of the uniform quantizer. This observation allows us to relate the mean square error to the tail-probabilities of the ℓ_∞ norm of the input vector. In particular, we exploit it to decide on the dimension s of subvectors as well as the growth rate of M_is.

Remark 14 (Growth rate of Tetration). A key distinguishing feature of RATQ is choosing the set of M_is to grow as a tetration, roughly as $M_{i+1} = e^{M_i}$. The large growth rate of a tetration allows us to cover the complete range of each coordinate using only a small number of dynamic ranges, which leads to an unbiased quantizer and reduces the communication. Also, after random rotation, each coordinate of the vector is a centered subgaussian random variable with a variance-parameter of $O(B^2/d)$ (see Remark 11), which, despite the large growth rate of a tetration, ensures that the per coordinate mean square error between the quantized output and the input is almost a constant, as can be seen from (3.7).

Choice of parameters. Throughout the remainder of this section, we set our parameters m, m_0, and h as follows

$$m = \frac{3B^2}{d}, \quad m_0 = \frac{2B^2}{d} \cdot \ln s, \quad \log h = \lceil \log(1 + \ln^*(d/3)) \rceil \,. \tag{3.8}$$

In particular, this results in $M_{h-1} \geq B$ whereby, for an input Y with $\|Y\|_2^2 \leq B^2$, RATQ outputs an unbiased estimate of Y.

We close with a remark on the computational complexity of RATQ.

Remark 15 (Computational complexity of RATQ). Since R is a Hadamard matrix, the matrix multiplication at the encoder and the decoder requires $O(d \log d)$ real operations[8]. Further, it takes $O(\log h)$ real operations to find the dynamic-range for each subvector, whereby the overall complexity for finding dynamic-ranges for $\lceil d/s \rceil$ subvectors is $O(\lceil d/s \rceil \log h)$ real operations; we can represent each of these dynamic-ranges as a $\log h$-bit

[8]Each addition, subtraction, multiplication, or division operation on the real field will be referred to as a real operation

binary string in another $O(\lceil d/s \rceil \log h)$ real operations, too. Note that the encoding complexity of CUQ for an input subvector of dimension s is s real operations, and thus, the overall complexity of $\lceil d/s \rceil$ CUQ operations is $O(d)$ real operations. Finally, we need $O(d \log k)$ real operations to represent the quantized values of d coordinates to k levels $\log k$-bit binary strings. Putting it all together, the overall complexity of the encoding procedure is $O(d \log d + d \log h/s + d \log k)$. By similar arguments, the complexity of real operations at the decoder would also be $O(d \log d + d \log h/s + d \log k)$.

Throughout the chapter, our choice of parameters s, h, and k for RATQ, which is roughly the optimal choice of these parameters for quantizing the ℓ_2 ball, would result in quantities $d \log h/s$ and $d \log k$ to be much lesser than $d \log d$. Thus, for parameters as chosen in this chapter or other reasonable choices of s, h, k, the encoding and decoding complexity of RATQ is $O(d \log d)$. Note that the random rotation based quantizer from [88] also has encoding and decoing complexity of $O(d \log d)$.

3.4.2 RATQ in the high-precision regime

The following result shows that RATQ is unbiased for almost surely bounded inputs and provides a bound for its worst-case second order moment; this constitutes a key technical tool for characterizing the performance of RATQ.

Theorem 3.4.2 (Performance of RATQ)**.** *Let $Q_{\mathtt{at},R}$ be the quantizer RATQ with M_js set by (3.8). Then, for all $s, k \in \mathbb{N}$,*

$$\alpha_2(Q_{\mathtt{at},R}) \leq B\sqrt{\frac{9 + 3\ln s}{(k-1)^2} + 1}, \quad \beta_2(Q_{\mathtt{at},R}) = 0. \tag{3.9}$$

The proof is deferred to Section 3.6.3.

Thus, α_2 is lower when s is small, but the overall precision needed grows since the number of subvectors increases. The following choice of parameters yields almost optimal performance:

$$s = \log h, \quad \log(k+1) = \left\lceil \log(2 + \sqrt{9 + 3\ln s}) \right\rceil. \tag{3.10}$$

For these choices, we obtain the following.

Corollary 3.4.3. *The overall precision r used by the quantizer $Q = Q_{\mathtt{at},R}$ with parameters set as in (3.8), (3.10) satisfies*

$$r \leq d(1+\Delta_1)+\Delta_2,$$

where $\Delta_1 = \left\lceil \log\left(2+\sqrt{9+3\ln\Delta_2}\right)\right\rceil$ *and* $\Delta_2 = \lceil \log(1+\ln^*(d/3))\rceil$.
Furthermore, the optimization protocol π given in Algorithm 3.1 satisfies

$$\sup_{(f,O)\in\mathcal{O}_{\mathtt{c},2}} \mathcal{E}(f,O,\pi,Q) \leq \frac{\sqrt{2}DB}{\sqrt{T}} \text{ and}$$

$$\sup_{(f,O)\in\mathcal{O}_{\mathtt{sc}}} \mathcal{E}(f,O,\pi,Q) \leq \frac{2B^2}{\gamma T}.$$

Proof. By the description RATQ, it encodes the subgradients using a fixed-length code of at the most $\lceil d/s\rceil\cdot\lceil\log h\rceil + d\,\lceil\log(k+1)\rceil$ bits. Upon substituting s, $\log h$, and $\log(k+1)$ as in (3.10) and (3.8), we obtain that the total precision is bounded above by $d(1+\Delta_1)+\Delta_2$.

For the second statement of the corollary, we have

$$\begin{aligned}\sup_{(f,O)\in\mathcal{O}_{\mathtt{c},2}} \mathcal{E}(f,O,\pi,Q) &\leq D\left(\frac{\alpha_2(Q_{\mathtt{at},R})}{\sqrt{T}}+\beta_2(Q_{\mathtt{at},R})\right)\\ &\leq \frac{DB}{\sqrt{T}}\cdot\sqrt{\frac{9+3\ln s}{(k-1)^2}+1}\\ &\leq \frac{\sqrt{2}DB}{\sqrt{T}},\end{aligned}$$

where the first inequality follows by Theorem 3.3.2, the second inequality follows by upper bounding $\alpha_2(Q_{\mathtt{at},R})$ and $\beta_2(Q_{\mathtt{at},R})$ using Theorem 3.4.2, and the third follows by substituting the parameters in the corollary statement. The upper bound for the strongly convex family follows in precisely the same manner. In particular, by combining Theorem 3.3.3 with Theorem 3.4.2, we have

$$\sup_{(f,O)\in\mathcal{O}_{\mathtt{sc}}} \mathcal{E}(f,O,\pi,Q) \leq \frac{2B^2}{\gamma T}.$$

□

Remark 16. The precision requirement in Corollary 3.4.3 matches the d-bit lower bound of Corollary 3.4.1 upto a multiplicative factor of $O\left(\log\log\log\ln^*(d/3)\right)$.

3.4.3 RATQ in the low-precision regime

We present a general method for reducing precision to much below d. This scheme is applicable when the output of the quantizer's encoder is a d length vector, where each coordinate is a separate fixed-length code. We simply reduce the length of the output message vector from the quantizer's encoder by sub-sampling a subset of coordinates using shared randomness. The decoder obtains the values of these coordinates using the decoder for the original quantizer and sets the rest of the coordinate-values to zero. This subsampling layer, which we call the *Random Coordinate Sampler* (RCS), can be added to RATQ after applying random rotation. In particular, *RATQ* we need the parameter s of these quantizers to be set to 1. This requirement of setting $s = 1$ ensures that the subsampled coordinates of the rotated vector can be decoded separately. This is a randomized scheme and requires the encoder and the decoder to share a random set $S \subset [d]$ distributed uniformly over all subsets of $[d]$ of cardinality μd.

The encoder $Q_S^{\mathtt{e}}$ of RCS simply outputs the vector

$$Q_S^{\mathtt{e}}(Y) := \{Y(i), i \in S\},$$

and the decoder $Q_S^{\mathtt{d}}(\tilde{Y})$, when applied to a vector $\tilde{Y} \in \mathbb{R}^{\mu d}$, outputs

$$Q_S^{\mathtt{d}}(\tilde{Y}) := \mu^{-1} \sum_{i \in S} \tilde{Y}(i) e_i,$$

where e_i denotes the ith element of standard basis for $\mathbb{R}^d$.

We can compose RCS with RATQ with parameter $s = 1$ by setting the encoder to $Q_S^{\mathtt{e}} \circ Q^{\mathtt{e}}$, and setting the decoder to $Q^{\mathtt{d}} \circ Q_S^{\mathtt{d}}$. Here we follow the convention that all 0-coordinates outputted by $Q_S^{\mathtt{d}}$ are decoded as 0 by $Q^{\mathtt{d}}$. Note that since we need to retain

RATQ encoder output for only μd coordinates, the overall precision of the quantizer is reduced by a factor of μ. We analyze the performance of this combined quantizer in the following theorem.

Theorem 3.4.4. *Let $Q_{\mathtt{at},R}$ be RATQ with $s = 1$ and $\tilde{Q}$ be the combination of RCS and $Q_{\mathtt{at},R}$ as described above. Then,*

$$\mathbb{E}\left[\tilde{Q}(Y)|Y\right] = \mathbb{E}\left[Q_{\mathtt{at},R}(RY)|Y\right] \quad \textit{and} \quad \mathbb{E}\left[\|\tilde{Q}(Y)\|_2^2|Y\right] = \frac{1}{\mu}\mathbb{E}\left[\|Q_{\mathtt{at},R}(RY)\|_2^2|Y\right],$$

which further leads to

$$\alpha_2(\tilde{Q}) \leq \frac{\alpha_2(Q_{\mathtt{at},R})}{\sqrt{\mu}} \quad \textit{and} \quad \beta_2(\tilde{Q}) = \beta_2(Q_{\mathtt{at},R}).$$

Proof. By the description of $Q_{\mathtt{at},R}$, we have

$$\tilde{Q}(Y) = \frac{1}{\mu}R^{-1}\sum_{i\in S} Q_{\mathtt{at},I}(RY)(i)e_i,$$

where $Q_{\mathtt{at},I}$ is the output vector formed by combining the d quantized values outputted by ATUQ ($Q_{\mathtt{at}}$) when input is the rotated vector. Namely,

$$Q_{\mathtt{at},I}(RY) = [Q_{\mathtt{at}}(RY(1)), \cdots, Q_{\mathtt{at}}(RY(d))]^T.$$

For the mean of $\tilde{Q}(Y)$, it holds that

$$\begin{aligned}
\mathbb{E}\left[\tilde{Q}(Y)|Y\right] &= \mathbb{E}\left[R^{-1}\sum_{i\in d} Q_{\mathtt{at},I}(RY)(i)e_i\frac{1}{\mu}\mathbb{1}_{i\in S}|Y\right] \\
&= \sum_{i\in d}\mathbb{E}\left[R^{-1}Q_{\mathtt{at},I}(RY)(i)e_i|Y\right]\cdot\frac{1}{\mu}\mathbb{E}\left[\mathbb{1}_{i\in S}|Y\right] \\
&= \sum_{i\in d}\mathbb{E}\left[R^{-1}Q_{\mathtt{at},I}(RY)(i)e_i|Y\right] \\
&= \mathbb{E}\left[R^{-1}\sum_{i\in d} Q_{\mathtt{at},I}(RY)(i)e_i|Y\right]
\end{aligned}$$

$$= \mathbb{E}\left[R^{-1}Q_{\mathtt{at},I}(RY)|Y\right]$$
$$= \mathbb{E}\left[Q_{\mathtt{at},R}(RY)|Y\right], \tag{3.11}$$

where the second identity follows from the fact that randomness used to generate a set S is independent of the randomness used in the quantizer and the randomness of Y; the third identity holds since $P(i \in S) = \mu$.

Next, moving to the computation of the second moment of the output of $\tilde{Q}$, we have

$$\begin{aligned}
\mathbb{E}\left[\|\tilde{Q}(Y)\|_2^2|Y\right] &= \mathbb{E}\left[\|\frac{1}{\mu}R^{-1}\sum_{i\in S}Q_{\mathtt{at},I}(RY)(i)e_i\|_2^2|Y\right] \\
&= \frac{1}{\mu^2}\mathbb{E}\left[\|\sum_{i\in S}Q_{\mathtt{at},I}(RY)(i)e_i\|_2^2|Y\right] \\
&= \frac{1}{\mu^2}\sum_{i\in[d]}\mathbb{E}\left[Q_{\mathtt{at},I}(RY)(i)^2|Y\right]\mathbb{E}\left[\mathbb{1}_{i\in S}|Y\right] \\
&= \frac{1}{\mu}\mathbb{E}\left[\|Q_{\mathtt{at}}(RY)\|_2^2|Y\right] \\
&= \frac{1}{\mu}\mathbb{E}\left[\|Q_{\mathtt{at},R}(RY)\|_2^2|Y\right],
\end{aligned} \tag{3.12}$$

where the second identity follows from the fact that R is a unitary matrix and the remaining steps follow simply by the description of the quantizers used. It follows that

$$\alpha(\tilde{Q}) = \frac{1}{\sqrt{\mu}}\alpha(Q_{\mathtt{at},R}), \quad \beta(\tilde{Q}) = \beta(Q_{\mathtt{at},R}).$$

□

We now set the parameter k to be a constant and sample roughly r coordinates. Specifically, we set

$$\begin{aligned}
&s = 1, \quad \log(k+1) = 3, \\
&\mu d = \min\{d, \lfloor r/(3 + \lceil \log(1 + \ln^*(d/3))\rceil)\rfloor\}.
\end{aligned} \tag{3.13}$$

For these choices, we have the following corollary.

Corollary 3.4.5. *For $r \geq 3 + \lceil \log(1 + \ln^*(d/3)) \rceil$, let Q be the composition of RCS and RATQ with parameters set as in (3.8), (3.13). Then, the optimization protocol π in Algorithm 3.1 satisfies*

$$\sup_{(f,O)\in\mathcal{O}_{\mathtt{c},2}} \mathcal{E}(f, O, \pi, Q) \leq \frac{\sqrt{2}DB}{\sqrt{\mu T}},$$

$$\sup_{(f,O)\in\mathcal{O}_{\mathtt{sc}}} \mathcal{E}(f, O, \pi, Q) \leq \frac{2B^2}{\mu\gamma T}.$$

Proof. When Q is a composition of RCS and RATQ, from Theorem 5.5.3 $\alpha_2^{\mathtt{m}}(Q) \leq \frac{1}{\sqrt{\mu}}\alpha(Q_{\mathtt{at},R})$, $\beta_2^{\mathtt{m}}(Q) \leq \beta(Q_{\mathtt{at},R})$, which by Theorem 3.3.2 yields

$$\begin{aligned}
\sup_{(f,O)\in\mathcal{O}_{\mathtt{c},2}} \mathcal{E}(f, O, \pi, Q) &\leq D\left(\frac{\alpha_2(Q_{\mathtt{at},R})}{\sqrt{\mu T}} + \beta_2(Q_{\mathtt{at},R})\right) \\
&\leq \frac{DB}{\sqrt{\mu T}} \cdot \sqrt{\frac{9}{(k-1)^2} + 1} \\
&\leq \frac{\sqrt{2}DB}{\sqrt{T}} \cdot \frac{\sqrt{d}}{\sqrt{\min\left\{d, \lfloor r/(3 + \log\ln^*(d/3))\rfloor\right\}}},
\end{aligned}$$

where the second inequality follows from Theorem 3.4.2 with $s = 1$, and the final inequality is obtained upon substituting the parameters as in the statement of the result. Similarly, for strongly convex function family, the result follows by combining Theorem 2.4.6 and 5.5.3.

□

Remark 17. Note that the convergence rate slows down by a μ specified in (3.13), which matches the lower bounds in Theorem 2.4.4 (for $p = 2$) and 2.4.6 upto a multiplicative factor of $O(\log\ln^*(d/3))$

3.5 Main results for mean square bounded oracles

In this section, we present results only for the convex family. We do this to avoid repetition since we can derive upper bounds for strongly convex family precisely in the same manner as convex family, as seen from the previous section.

With the mean square bounded assumption, we now need to quantize random vectors Y such that $\mathbb{E}\left[\|Y\|_2^2\right] \leq B^2$. We take recourse to the standard *gain-shape* quantization paradigm in vector quantization (*cf.*[34]).

Definition 3.5.1 (Gain-shape quantizer)**.** A Quantizer Q is defined to be a gain-shape quantizer if it has the following form

$$Q(Y) = Q_g(\|Y\|_2) \cdot Q_s(Y/\|Y\|_2),$$

where Q_g is any $\mathbb{R} \to \mathbb{R}$ quantizer and Q_s is any $\mathbb{R}^d \to \mathbb{R}^d$ quantizer.

Specifically, we separately quantize the *gain* $\|Y\|_2$ and the *shape*[9] $Y/\|Y\|_2$ of Y, and form the estimate of Y by simply multiplying the estimates for the gain and the shape. Note that we already have a good shape quantizer: RATQ. We only need to modify the parameters in (3.8) to make it work for the unit sphere; we set

$$m = \frac{3}{d}, \quad m_0 = \frac{2}{d} \cdot \ln s, \quad \log h = \lceil \log(1 + \ln^*(d/3)) \rceil \,. \tag{3.14}$$

We now proceed to derive the worst-case α and β for a general gain-shape. In order to make clear the dependence on B and d, we refine our notations for $\{\alpha_2^{\mathtt{m}}(Q), \beta_2^{\mathtt{m}}(Q)\}$ and $\{\alpha_2(Q), \beta_2(Q)\}$, defined in (3.4) and (3.5), respectively, to $\{\alpha_2^{\mathtt{m}}(Q; B, d), \beta_2^{\mathtt{m}}(Q; B, d)\}$ and $\{\alpha_2(Q; B, d), \beta_2(Q; B, d)\}$.

Theorem 3.5.2. *Let $Q(Y) = Q_1(\|Y\|_2) \cdot Q_2(Y/\|Y\|_2)$, where Q_1 is any gain quantizer and Q_2 is any shape quantizer. Also, suppose $Q_1(\|Y\|_2)$ and $Q_2(Y/\|Y\|_2)$ are conditionally independent given Y. Then,*

$$\alpha_2^{\mathtt{m}}(Q; B, d) \leq \alpha_2^{\mathtt{m}}(Q_1; B, 1) \cdot \alpha_2(Q_2; 1, d).$$

Furthermore, suppose that Q_2 satisfies

$$\mathbb{E}\left[Q_2(y_s)\right] = y_s, \quad \forall y_s \quad s.t. \quad \|y_s\|_2^2 \leq 1.$$

[9]For the event $\|Y\|_2 = 0$, we follow the convention that $Y/\|Y\|_2 = e_1$.

Then, we have[10]

$$\beta_2^{\mathfrak{m}}(Q;B,d) \leq \sup_{Y\in\mathbb{R}^d:\mathbb{E}\left[\|Y\|_2^2\right]\leq B^2} \mathbb{E}\left[\left|\mathbb{E}\left[Q_1(\|Y\|_2) - \|Y\|_2 \mid Y\right]\right|\right].$$

Proof. Denote by Y_s the shape of the vector Y given by

$$Y_s := \frac{Y}{\|Y\|_2}.$$

The worst-case second moment: Towards evaluating $\alpha(Q;B,d)$, we have

$$\begin{aligned}
\mathbb{E}\left[\|Q(Y)\|_2^2\right] &= \mathbb{E}\left[Q_1(\|Y\|_2)^2\|Q_2(Y_s)\|_2^2\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[Q_1(\|Y\|_2)^2\|Q_2(Y_s)\|_2^2|Y\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[Q_1(\|Y\|_2)^2|Y\right]\mathbb{E}\left[\|Q_2(Y_s)\|_2^2|Y\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[Q_1(\|Y\|_2)^2|Y\right]\mathbb{E}\left[\|Q_2(Y_s)\|_2^2|Y_s\right]\right],
\end{aligned}$$

where the third identity follows by conditional independence of $Q_1(\|Y\|_2)^2$ and $\|Q_2(Y_s)\|_2^2$ given Y and the fourth follows from the law of iterated expectations.

Consider the random variable $\mathbb{E}\left[\|Q_2(Y_s)\|_2^2|Y_s\right]$. We claim that this is less than $\alpha_0(Q_2;1,d)$ almost surely. Towards this end, note that

$$\mathbb{E}\left[\|Q_2(Y_s)\|_2^2|Y_s = y\right] = \mathbb{E}\left[\|Q_2(y)\|_2^2\right],$$

since the randomness used in implementation of Q_2 is independent of the input random variable Y. Moreover, for any y with $\|y\|_2^2 \leq 1$, we have from the definition of $\alpha_2(Q_2;1,d)$ that $\mathbb{E}\left[\|Q_2(y)\|_2^2\right] \leq \alpha_2(Q_2;1,d)^2$. Therefore, for any Y with $\mathbb{E}\left[\|Y\|_2^2\right] \leq B^2$, we have

$$\begin{aligned}
\mathbb{E}\left[\|Q(Y)\|_2^2\right] &= \mathbb{E}\left[\mathbb{E}\left[Q_1(\|Y\|_2)^2|Y\right]\mathbb{E}\left[\|Q_2(Y_s)\|_2^2|Y_s\right]\right] \\
&\leq \mathbb{E}\left[\mathbb{E}\left[Q_1(\|Y\|_2)^2|Y\right]\right]\cdot\alpha_2(Q_2;1,d)^2
\end{aligned}$$

[10] *The quantity on the right-side of this bound exceeds the bias $\beta_2^{\mathfrak{m}}(Q_1;B,1)$. Nonetheless, in all our bounds for bias, this is the quantity we have been handling.*

$$= \mathbb{E}\left[Q_1(\|Y\|_2)^2\right] \cdot \alpha_2(Q_2; 1, d)^2$$
$$\leq \alpha_2^{\mathtt{m}}(Q_1; B, 1)^2 \cdot \alpha_2(Q_2; 1, d)^2. \tag{3.15}$$

Taking the supremum of the left-side over all random vectors Y with $\mathbb{E}\left[\|Y\|_2^2\right] \leq B^2$ gives us the desired bound for $\alpha_2^{\mathtt{m}}(Q; B, d)$.

The worst-case bias: Towards evaluating $\beta_2^{\mathtt{m}}(Q; B, d)$, we note from our hypothesis that $\mathbb{E}\left[Q_2(Y_s)|Y\right] = \mathbb{E}\left[Q_2(Y_s)|Y_s\right] = Y_s$, which further yields

$$\begin{aligned}
\mathbb{E}\left[Q(Y) - Y\right] &= \mathbb{E}\left[\mathbb{E}\left[Q_1(\|Y\|_2)Q_2(Y_s) - Y|Y\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[Q_1(\|Y\|_2)|Y\right]\mathbb{E}\left[Q_2(Y_s)|Y\right] - Y\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[Q_1(\|Y\|_2)|Y\right]Y_s - \|Y\|_2 Y_s\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[Q_1(\|Y\|_2) - \|Y\|_2|Y\right]Y_s\right], \qquad (3.16)
\end{aligned}$$

where the second identity uses conditional independence of $Q_1(\|Y\|_2)$ and $Q_2(Y_s)$. By using the conditional Jensen's inequality, we get

$$\begin{aligned}
\|\mathbb{E}\left[Q(Y) - Y\right]\|_2 &= \|\mathbb{E}\left[\mathbb{E}\left[Q_1(\|Y\|_2) - \|Y\|_2|Y\right]Y_s\right]\|_2 \\
&\leq \mathbb{E}\left[\|\mathbb{E}\left[Q_1(\|Y\|_2) - \|Y\|_2|Y\right]Y_s\|_2\right] \\
&= \mathbb{E}\left[\left|\mathbb{E}\left[Q_1(\|Y\|_2) - \|Y\|_2|Y\right]\right|\right].
\end{aligned}$$

□

We remark that quantizers proposed in most of the prior work can be cast in this gain-shape framework. Most works simply state that gain is a single parameter which can be quantized using a fixed number of bits; for instance, a single double precision number is prescribed for storing the gain. However, the quantizer is not specified. We carefully analyze this problem and establish lower bounds when a uniform quantizer with a fixed dynamic range is used for quantizing the gain. Further, we present our own quantizer which significantly outperforms uniform gain quantization.

3.5.1 Limitation of uniform gain quantization

We establish lower bounds for a general class of gain-shape quantizers $Q(y) = Q_g(\|y\|_2)Q_s(y/\|y\|_2)$ of precision r that satisfy the following *structural assumptions*:

1. **(Independent gain-shape quantization)** For any given $y \in \mathbb{R}^d$, the output of the gain and the shape quantizers are independent.

2. **(Bounded dynamic-range)** For any $y \in \mathbb{R}^d$, there exists a $M > 0$ such that whenever $\|y\|_2 > M$, $Q(y)$ has a fixed distribution $P_\emptyset$.

3. **(Uniformity)** There exists $m \in [M/2^r, M]$ such that for every t in $[0, m]$,

   (a) $\texttt{supp}(Q_g(t)) \subseteq \{0, m\}$;

   (b) If $P(Q_g(t) = m) > 0$, then

$$\frac{P(Q_g(t_2) = m)}{P(Q_g(t_1) = m)} \leq \frac{t_2}{t_1}, \quad \forall\, 0 \leq t_1 \leq t_2 \leq m.$$

The first two assumptions are perhaps clear and hold for a large class of quantizers. The third one is the true limitation and is satisfied by different forms of uniform gain quantizers. For instance, for the one-dimensional version of CUQ with dynamic range $[0, M]$, which is an unbiased, uniform gain quantizer with k_g levels, it holds with $m = M/(k_g - 1)$ (corresponding to the innermost level $[0, M/(k_g - 1)]$). It can also be shown to include a deterministic uniform quantizer that rounds-off at the mid-point. The third condition, in essence, captures the unbiasedness requirement that the probability of declaring higher level is proportional to the value. Note that (t_2/t_1) on the right-side can be replaced with any constant multiple of (t_2/t_1). For easy reference, we will refer to these assumptions as Structural Assumptions 1-3.

Below we present lower bounds for performance of any optimization protocol using a gain-shape quantizer that satisfies the assumptions above. We present separate results for high-precision and low-precision regimes, but both are obtained using a general construction that exploits the admissibility of heavy-tail distributions for mean square bounded oracles. This construction is new and may be of independent interest.

Theorem 3.5.3. *Consider a gain-shape quantizer Q satisfying Structural Assumptions 1-3. Suppose that for $\mathcal{X} = \{x : \|x\|_2 \leq D/2\}$ we can find an optimization protocol π which, using at most T iterations, achieves $\sup_{f,O\in\mathcal{O}} \mathcal{E}(f, O, \pi, Q) \leq \frac{3DB}{\sqrt{T}}$. Then, we can find a universal constant c such that the overall precision r of the quantizer must satisfy*

$$r \geq c(d + \log T).$$

Theorem 3.5.4. *Consider a gain-shape quantizer Q satisfying Structural Assumptions 1-3. Suppose that the number of bits r_g used by the gain quantizer are fixed independently of T. Then, for $\mathcal{X} = \{x : \|x\|_2 \leq D/2\}$, there exists $(f, O) \in \mathcal{O}$ such that for any optimization protocol π using at most T iterations, we must have*

$$\mathcal{E}(f, O, \pi, Q) \geq \frac{c(r_g)DB}{T^{1/3}},$$

where $c(r_g)$ is a constant depending only on the number of bits used by the gain quantizer (but not on T).

The proofs of Theorems 3.5.3 and 3.5.4 are technical and long; we defer them to Section 3.6.6.

Remark 18. Thus, from Theorem 3.5.3, for any optimization algorithm to achieve the error of $O(1/\sqrt{T})$ after T iterations, when used a quantizer satisfying the Structural Assumptions 1-3, the precision of the quantizer at every iteration must scale at least as roughly $\log T$. Conversely, from Theorem 3.5.4, if we use a quantizer with precision fixed independently of T, then any optimization algorithm used with this quantizer must have error at least $\Omega(1/T^{1/3})$.

3.5.2 A-RATQ in the high-precision regime

Instead of quantizing the gain uniformly, we propose to use an adaptive quantizer termed *Adaptive Geometric Uniform Quantizer* (AGUQ) for gain. AGUQ operates similar to the one-dimensional ATUQ, except the possible dynamic-ranges $M_{g,0}, \ldots, M_{g,h}$ grow

geometrically (and not using tetra-iterations of ATUQ) as follows:

$$M_{g,j}^2 = B^2 \cdot a_g^j, \quad 0 \le j \le h_g - 1. \tag{3.17}$$

Specifically, for a given gain $G \ge 0$, AGUQ first identifies the smallest j such that $G \le M_{g,j}$ and then represents G using the one-dimensional version of CUQ with a dynamic range $[0, M_{g,j}]$ and k_g uniform levels

$$B_{M_{g,j},k}(\ell) := \ell \cdot \frac{M_{g,j}}{k_g - 1}, \quad \forall\, \ell \in \{0, \dots, k-1\}.$$

As in ATUQ, if $G > M_{h_g-1}$, the overflow $\emptyset$ symbol is used and the decoder simply outputs 0. The overall procedure is the similar to Algorithms (3.6) and (3.7) for $s = 1, h = h_g$, and $M_j = M_{g,j}$, $0 \le j \le h_g - 1$; the only changes is that now we restrict to nonnegative interval $[0, M_{g,j}]$ for the one-dimensional version of CUQ with uniform levels k_g.

The following result characterizes the performance of one-dimensional quantizer AGUQ; it is the only component missing in the analysis of A-RATQ.

Lemma 3.5.5. *Let $Q_{\mathtt{a}}$ be the quantizer AGUQ described above, with $h_g \ge 2$. Then,*

$$\alpha_2^{\mathtt{m}}(Q_{\mathtt{a}}; B, 1) \le B\sqrt{\frac{1}{4(k_g-1)^2} + \frac{a_g(h_g-1)}{4(k_g-1)^2} + 1},$$
$$\beta_2^{\mathtt{m}}(Q_{\mathtt{a}}; B, 1) \le \sup_{Y \ge 0 \text{ a.s. } :\mathbb{E}[Y^2] \le B^2} \mathbb{E}\left[\left|\mathbb{E}\left[Q_a(Y) - Y | Y\right]\right|\right] \le \frac{B^2}{M_{g,h_g-1}}.$$

Proof of this result, too, is deferred to Section 3.6.4. Note that we have derived a bound for a quantity that is slightly larger than the bias of $Q_{\mathtt{a}}$, since we want to use this result along with Theorem 3.5.2.

Thus, our overall quantizer termed the *adaptive-RATQ* (A-RATQ) is given by

$$Q(Y) := Q_a(\|Y\|_2) \cdot Q_{\mathtt{at},R}(Y/\|Y\|_2),$$

where Q_a denotes the one dimensional AGUQ and $Q_{\mathtt{at},R}$ denotes the d-dimensional RATQ. Note that we use independent randomness for $Q_a(\|Y\|_2)$ and $Q_{\mathtt{at},R}(Y/\|Y\|_2)$, rendering

them conditionally independent given Y.

The parameters s, k for RATQ and a_g, k_g for AGUQ are yet to be set. We first present a result which holds for all choices of these parameters.

Theorem 3.5.6 (Performance of A-RATQ)**.** *For Q set to A-RATQ with parameters set as in (3.14), (3.17), we have*

$$\alpha_2^{\mathtt{m}}(Q; B, d) \leq B\sqrt{\frac{1}{4(k_g-1)^2} + \frac{a_g(h_g-1)}{4(k_g-1)^2} + 1} \cdot \sqrt{\frac{9+3\ln s}{(k-1)^2} + 1},$$
$$\beta_2^{\mathtt{m}}(Q; B, d) \leq \frac{B^2}{M_{g,h_g-1}}.$$

Proof.

The worst-case second moment of A-RATQ: By Theorem 3.5.2 we have

$$\begin{aligned}\alpha_2^{\mathtt{m}}(Q; B, d) &\leq \alpha_2^{\mathtt{m}}(Q_{\mathtt{a}}; B, 1) \cdot \alpha_2(Q_{\mathtt{at},R}; 1, d) \\ &\leq \alpha_2^{\mathtt{m}}(Q_{\mathtt{a}}; B, 1) \cdot \sqrt{\frac{9+3\ln s}{(k-1)^2} + 1} \\ &\leq B\sqrt{\frac{1}{4(k_g-1)^2} + \frac{a_g(h_g-1)}{4(k_g-1)^2} + 1} \cdot \sqrt{\frac{9+3\ln s}{(k-1)^2} + 1},\end{aligned}$$

where the second inequality used Theorem 3.4.2 with $B = 1$, and the third follows by Lemma 3.5.5.

The worst-case bias of A-RATQ: With parameters of RATQ set as in (3.14), we have that

$$\mathbb{E}\left[Q_{\mathtt{at},R}(y)\right] = y, \quad \forall y \quad s.t \quad \|y\|_2^2 \leq 1.$$

Therefore, by Theorem 3.5.2 it follows that

$$\beta_2^{\mathtt{m}}(Q; B, d) \leq \sup_{Y:\mathbb{E}\left[\|Y\|_2^2\right] \leq B^2} \mathbb{E}\left[\left|\mathbb{E}\left[Q_{\mathtt{a}}(\|Y\|_2) - \|Y\|_2 | Y\right]\right|\right] \leq \frac{B^2}{M_{g,h_g-1}},$$

where the second inequality follows from Lemma 3.5.5. □

Note that RATQ yields an unbiased estimator; the bias in A-RATQ arises from AGUQ since the gain is not bounded. Further, AGUQ uses a precision of $\lceil \log h_g \rceil + \lceil \log(k_g + 1) \rceil$ bits, and therefore, the overall precision of A-RATQ is $\lceil \log h_g \rceil + \lceil \log(k_g + 1) \rceil + \lceil d/s \rceil \lceil \log h \rceil + d \lceil \log(k + 1) \rceil$ bits.

In the high-precision regime, we set

$$\begin{aligned} a_g = 2, \quad \log h_g &= \left\lceil \log(1 + \frac{1}{2} \log T) \right\rceil, \\ \log(k_g + 1) &= \left\lceil \log \left(2 + \frac{1}{2} \sqrt{\log T + 1} \right) \right\rceil. \end{aligned} \tag{3.18}$$

Corollary 3.5.7. *Denote by Q the quantizer A-RATQ with parameters set as in (3.14), (3.10), and (3.18). Then, the overall precision r used by Q is less than*

$$d(1 + \Delta_1) + \Delta_2 + \left\lceil \log \left(2 + \sqrt{\log T + 1} \right) \right\rceil,$$

where $\Delta_1 = \left\lceil \log \left(2 + \sqrt{9 + 3 \ln \Delta_2} \right) \right\rceil$ and $\Delta_2 = \lceil \log(1 + \ln^*(d/3)) \rceil$, the same as Corollary 3.4.3. Furthermore, the optimization protocol π given in algorithm 3.1 satisfies $\sup_{(f,O) \in \mathcal{O}} \mathcal{E}(f, O, \pi, Q) \leq 3DB/\sqrt{T}$.

Proof. The proof is similar to the proof of Corollary 3.4.3. The first statement follows by simply upper bounding the precision of the fixed-length code for A-RATQ with parameters as in the statement. The second statement follows by bounding $\sup_{(f,O) \in \mathcal{O}} \mathcal{E}(f, O, \pi, Q)$ using Theorem 3.3.2, using the upper bounds for α and β given in Theorem 3.5.6, and finally substituting the parameters. □

Remark 19. The precision used in Corollary 3.5.7 matches the lower bound in Corollary 3.4.1 upto an additive factor of $\log \log T$ (ignoring the mild factor of $\log \log \log \ln^*(d/3)$), which is much lower than the $\log T$ lower bound we established for uniform gain quantizers. Hence, the precision requirement of A-RATQ in the high-precision regime is considerably smaller than the precision requirement of uniform gain quantizers established in Theorem 3.5.3 for $\log T \gg d(1 + \Delta_1)$, while remaining roughly the same for $\log T = O(d(1 + \Delta_1))$.

3.5.3 A-RATQ in the low-precision regime

In order to operate with a fixed precision r, we combine A-RATQ with RCS. We simply combine RCS with RATQ as in Section 3.4.3 to limit the precision and use AGUQ as the gain quantizer. Note that we use independent randomness in our gain quantizer $Q_a(\|Y\|_2)$ and our shape quantizer $\tilde{Q}(Y/\|Y\|_2)$, rendering them conditionally independent given Y. We have the following theorem characterizing α and β for this quantizer.

Theorem 3.5.8. *Let $Q(Y) = Q_a(\|Y\|) \cdot \tilde{Q}(Y/\|Y\|_2)$, where $\tilde{Q}$ is the composition of RCS and RATQ described in Theorem 5.5.3 with parameters m, m_0, and h of RATQ as in (3.14) and Q_a is AGUQ. Then,*

$$\alpha_2^{\mathtt{m}}(Q; B, d) \le B\sqrt{\frac{1}{4(k_g-1)^2} + \frac{a_g(h_g-1)}{4(k_g-1)^2} + 1} \cdot \frac{1}{\sqrt{\mu}}\sqrt{\frac{9+3\ln s}{(k-1)^2}+1},$$
$$\beta_2^{\mathtt{m}}(Q; B, d) \le \frac{B^2}{M_{g,h_g-1}}.$$

Proof.

The worst-case second moment: Starting by applying Theorem 3.5.2, we have

$$\begin{aligned}\alpha_2^{\mathtt{m}}(Q; B, d) &\le \alpha_2^{\mathtt{m}}(Q_{\mathtt{a}}; B, 1) \cdot \alpha_2(\tilde{Q}; 1, d) \\ &\le \alpha_2^{\mathtt{m}}(Q_{\mathtt{a}}; B, 1) \cdot \frac{1}{\sqrt{\mu}}\alpha_2(Q_{\mathtt{at},R}; 1, d) \\ &\le B\sqrt{\frac{1}{4(k_g-1)^2} + \frac{a_g(h_g-1)}{4(k_g-1)^2} + 1} \cdot \frac{1}{\sqrt{\mu}}\sqrt{\frac{9+3\ln s}{(k-1)^2}+1},\end{aligned}$$

where the second inequality follows by Theorem 5.5.3 and the third follows by Theorem 3.4.2 and Lemma 3.5.5.

The worst-case bias: With parameters of RATQ set as in (3.14), we have that

$$\mathbb{E}\left[\tilde{Q}(y)\right] = y, \quad \forall y \quad s.t \quad \|y\|_2^2 \le 1.$$

Therefore, by Theorem 3.5.2 we get

$$\begin{aligned}\beta_2^{\mathtt{m}}(Q;B,d) &\le \sup_{Y:\mathbb{E}\left[\|Y\|_2^2\right]\le B^2} \mathbb{E}\left[\left|\mathbb{E}\left[Q_{\mathtt{a}}(\|Y\|_2)-\|Y\|_2|Y\right]\right|\right] \\ &\le \frac{B^2}{M_{g,h_g-1}},\end{aligned}$$

where the second inequality follows from Lemma 3.5.5.

□

We divide the total precision r into r_g and r_s bits: r_g to quantize the gain, r_s to quantize the subsampled shape vector. We set

$$\begin{aligned}&s,k, \text{ and } \mu d \text{ as in (3.13), with } r_s \text{ replacing } r,\\ &\log h_g = \log(k_g+1) = \frac{r_g}{2},\ a_g = (\mu T)^{\frac{1}{h_g+1}}\end{aligned} \tag{3.19}$$

That is, our shape quantizer simply quantizes μd randomly chosen coordinates of the rotated vector using ATUQ with r_s bits, and the remaining bits are used by the gain quantizer AGUQ. The result below shows the performance of this quantizer.

Corollary 3.5.9. *For any r with gain quantizer being assigned $r_g \ge 4$ bits and shape quantizer being assigned $r_s \ge 3+\lceil\log(1+\ln^*(d/3))\rceil$, let Q be the combination of RCS and A-RATQ with parameters set as in (3.14), (3.17), (3.19). Then for $\mu T \ge 1$, the optimization protocol π in Algorithm 3.1 can obtain*

$$\sup_{(f,O)\in\mathcal{O}} \mathcal{E}(f,O,\pi,Q) \le O\left(DB\left(\frac{d}{T\min\{d,\frac{r_s}{\log\ln^*(d/3)}\}}\right)^{\frac{1}{2}\cdot\frac{2^{r_g/2}-1}{2^{r_g/2}+1}}\right).$$

Proof. By using Theorem 3.3.2 to upper bound $\sup_{(f,O)\in\mathcal{O}^{\mathtt{m}}_{\mathtt{c},2}} \mathcal{E}(f,O,\pi,Q)$ and then Theorem 3.5.8 to upper-bound α and β, we get

$$\sup_{(f,O)\in\mathcal{O}^{\mathtt{m}}_{\mathtt{c},2}} \mathcal{E}(f,O,\pi,Q) \le D\left(\frac{1}{\sqrt{\mu T}}\sqrt{\frac{B^2}{4(k_g-1)^2}+\frac{a_g(h_g-1)B^2}{4(k_g-1)^2}+B^2}\sqrt{\frac{9+3\ln s}{(k-1)^2}+1}+\frac{B^2}{M_{g,h-1}}\right).$$

By substituting the parameters as in the statement and using the fact that $\mu T \geq 1$ completes the proof. □

Remark 20. Our fixed precision quantizer in Corollary 3.5.9 establishes that using only a constant number of bits for gain-quantization, we get very close to the lower bound in Theorem 2.4.4. For instance, given access to a large enough precision r, if we set r_g to be 16 bits, we get

$$\sup_{(f,O)\in\mathcal{O}} \mathcal{E}(f, O, \pi, Q) \leq O\left(DB\left(\frac{d}{T\min\{d, \frac{r-16}{\log\ln^*(d/3)}\}}\right)^{\frac{1}{2}\cdot\frac{255}{257}}\right).$$

Here, the ratio of $d/(\min\{d, \frac{r-16}{\log\ln^*(d/3)}\})$ is very close to the optimal ratio of $d/(\min\{d, r\})$, and the exponent $255/(2\cdot 257)$ is close to the optimal exponent $1/2$.

Remark 21. We remark that A-RATQ satisfies Assumptions (1) and (2) in Section 3.5.1 but not (3), and breaches the lower bound for uniform gain quantizers established in Section 3.5.1.

3.5.4 A variable-length quantizer

So far in this thesis, we have restricted our quantizers to be fixed-length. We now present a variable-length quantizer for mean square bounded oracles which improves over the convergence rate of Corollary 3.5.9. The quantizer we present is a gain-shape quantizer that uses RATQ as the shape quantizer but uses a variable-length version of the gain quantizer called AGUQ$^+$, an update on AGUQ presented in the previous section.

AGUQ$^+$ differs from AGUQ in two crucial aspects: 1) The number of uniform levels of the uniform quantizer for different dynamic ranges is different. Denote by $k_{g,j}$ the number of uniform levels corresponding to the range $[0, M_{g,j}]$. We choose $k_{g,j}$ to grow geometrically. 2) AGUQ$^+$ employs entropic compression further to reduce the expected code-length of the quantized representation. Besides the differences, the quantization in AGUQ$^+$ is the same as that of AGUQ.

Specifically, AGUQ$^+$ is once again an adaptive quantizer like AGUQ with dynamic

ranges growing in a goemetric manner, precisely in the same way as in (3.17). For a given gain $G \geq 0$, AGUQ$^+$ first identifies the smallest j such that $G \leq M_{g,j}$ and then represents G using the one-dimensional version of CUQ with a dynamic range $[0, M_{g,j}]$ and $k_{g,j}$ uniform levels

$$B_{M_{g,j}}(\ell) := \ell \cdot \frac{M_{g,j}}{k_{g,j} - 1}, \quad \forall\, \ell \in \{0, \ldots, k_{g,j} - 1\}.$$

As in AGUQ, if $G > M_{g,h_g-1}$, the overflow $\emptyset$ symbol is used and the decoder simply outputs 0.

Note that since we are quantizing unbounded random variables, it is difficult to avoid bias. Nevertheless, we will make the effect of the bias negligible. Specifically, for the application of communication-constrained optimization of convex functions, it will be desirable to have a bias of at the most $1/\sqrt{T}$. To achieve this, we set

$$a_g = 2, \quad h_g = 1 + \frac{1}{2}\log T, \quad k_{g,j} + 1 = 2^{j+1}. \tag{3.20}$$

We now describe the variable length coding procedure. The variable-length bit string itself is a concatenation of two separate bit strings. The first string represents the non-uniform level j in $\{0, \cdots, h_g - 1\}$ using the first h_g symbols of the Huffman code for geometric distribution with parameter $1/2$. The second string uses a fixed-length code of $k_{g,j}$ bits. We will show that the total number of bits used for both the strings would be $O(1)$ in expectation.

Remark 22 (Entropic compression in *AGUQ*$^+$). *AGUQ*$^+$ improves over vanilla *AGUQ* by exploiting that the probability of a larger dynamic range being chosen decays exponentially with the dynamic range level j. For concreteness, since $\mathbb{E}[Y^2] \leq B^2$, we have by Markov's inequality $P(|Y| > M_{g,j-1}) \leq B^2/M_{g,j-1}^2 = a_g^{-j} = 2^{-j+1}$. This probability bound allows us to use a code similar to that of Huffman code for geometric distribution and only have a constant code-length.

The following result characterizes the performance of one-dimensional quantizer AGUQ$^+$.

Lemma 3.5.10. *Let $Q_{\mathtt{a}^+}$ be the quantizer AGUQ$^+$ described above with parameters set as in (3.17) and (3.20). Then, for Y such that $\mathbb{E}[Y^2] \leq B^2$, $Q_{\mathtt{a}^+}(Y)$ can be represented using at the most $O(1)$ bits of precision in expectation,*

$$\alpha_2^{\mathtt{m}}(Q_{\mathtt{a}^+}; B, 1) \leq O(1), \text{ and}$$
$$\beta_2^{\mathtt{m}}(Q_{\mathtt{a}^+}; B, 1) \leq \sup_{Y \geq 0 \text{ a.s. } : \mathbb{E}[Y^2] \leq B^2} \mathbb{E}\left[\left|\mathbb{E}\left[Q_a(Y) - Y | Y\right]\right|\right] \leq O\left(\frac{B}{\sqrt{T}}\right).$$

The proof is deferred to Section 3.6.5

Remark 23. Thus employing a gain-shape quantizer where the gain is quantized by AGUQ$^+$ and the shape is quantized by the subsampled version of RATQ along with PSGD improves over the convergence guarantees of Corollary 3.5.9, and essentially has the same order of convergence guarantees as that in Corollary 3.4.5. That is, this particular gain-shape quantizer along with PSGD achieves roughly the convergence rate of $O\left(\frac{DB}{\sqrt{T}} \cdot \frac{d}{r}\right)$, which is the same as that in lower-bound for communication constrained optimization of convex and ℓ_2 lipschitz functions, $\mathcal{O}_{\mathtt{c},2}^m$, as stated in Theorem 2.4.4. From Remark 7, the lower bound in Theorem 2.4.4 holds for variable length quantizers, too, provided the gradient processing is done in a nonadaptive manner. Thus employing a gain-shape quantizer where the gain is quantized by AGUQ$^+$ and the shape is quantized by the subsampled version of RATQ along with PSGD is optimal for communication-constrained optimization of $\mathcal{O}_{\mathtt{c},2}^m$ when nonadaptive, variable-length quantizers are allowed.

3.6 Main proofs

3.6.1 Proof of Theorem 3.3.2

We proceed as in the standard proof of convergence (see, for instance, [15]): Denoting by $\Gamma_{\mathcal{X}}(x)$ the projection of x on the set $\mathcal{X}$ (in the Euclidean norm), the error at time t can be bounded as

$$\|x_t - x^*\|_2^2 = \|\Gamma_{\mathcal{X}}\Big(x_{t-1} - \eta Q(\hat{g}(x_{t-1}))\Big) - x^*\|_2^2$$

$$
\begin{aligned}
&\leq \|\Big(x_{t-1} - \eta Q(\hat{g}(x_{t-1}))\Big) - x^*\|_2^2 \\
&= \|x_{t-1} - x^*\|_2^2 + \|\eta Q(\hat{g}(x_{t-1}))\|_2^2 - 2\eta(x_{t-1} - x^*)^T Q(\hat{g}(x_{t-1})) \\
&= \|x_{t-1} - x^*\|_2^2 + \|\eta Q(\hat{g}(x_{t-1}))\|_2^2 - 2\eta(x_{t-1} - x^*)^T \Big(Q(\hat{g}(x_{t-1})) - \hat{g}(x_{t-1})\Big) \\
&\quad - 2\eta(x_{t-1} - x^*)^T \hat{g}(x_{t-1}),
\end{aligned}
$$

where the first inequality is a well known property of the projection operator Γ (see, for instance, Lemma 3.1, [15]). By rearranging the terms, we have

$$
\begin{aligned}
2\eta(x_{t-1} - x^*)^T \hat{g}(x_{t-1}) \leq \|x_{t-1} - x^*\|_2^2 - \|x_t - x^*\|_2^2 + \|\eta Q(\hat{g}(x_{t-1}))\|_2^2 \qquad (3.21) \\
- 2\eta(x_{t-1} - x^*)^T \left(Q(\hat{g}(x_{t-1})) - \hat{g}(x_{t-1})\right). \qquad (3.22)
\end{aligned}
$$

Also, since $\mathbb{E}\left[\hat{g}(x_{t-1})|x_{t-1}\right]$ is a subgradient at x_{t-1} for the convex function f, upon taking expectation over the randomness in the subgradient estimates as well as the quantizer output we get

$$
\mathbb{E}\left[f(x_{t-1}) - f(x^*)\right] \leq \mathbb{E}\left[(x_{t-1} - x^*)^T \mathbb{E}\left[\hat{g}(x_{t-1})|x_{t-1}\right]\right], \qquad (3.23)
$$

which with the previous bound yields

$$
\begin{aligned}
2\eta\mathbb{E}\left[f(x_{t-1}) - f(x^*)\right] \leq \mathbb{E}\left[\|x_{t-1} - x^*\|_2^2\right] - \mathbb{E}\left[\|x_t - x^*\|_2^2\right] + \eta^2\mathbb{E}\left[\|Q(\hat{g}(x_{t-1}))\|_2^2\right] \\
- 2\eta\mathbb{E}\left[(x_{t-1} - x^*)^T \Big(Q(\hat{g}(x_{t-1})) - \hat{g}(x_{t-1})\Big)\right].
\end{aligned}
$$

Next, by the Cauchy-Schwarz inequality and the assumption in (1), the third term on the right-side above can be bounded further to obtain

$$
\begin{aligned}
2\eta\mathbb{E}\left[f(x_{t-1}) - f(x^*)\right] \leq \mathbb{E}\left[\|x_{t-1} - x^*\|_2^2\right] - \mathbb{E}\left[\|x_t - x^*\|_2^2\right] + \eta^2\mathbb{E}\left[\|Q(\hat{g}(x_{t-1}))\|_2^2\right] \\
+ 2\eta \cdot D \cdot \mathbb{E}\left[\|\mathbb{E}\left[Q(\hat{g}(x_{t-1})) - \hat{g}(x_{t-1})|x_{t-1}\right]\|_2\right].
\end{aligned}
$$

Finally, we note that, by the definition of α and β, for L_2-bounded oracles we have

$$\mathbb{E}\left[\|Q(\hat{g}(x_{t-1}))\|_2^2\right] \leq \alpha_2^{\mathtt{m}}(Q)^2,$$
$$\|\mathbb{E}\left[Q(\hat{g}(x_{t-1})) - \hat{g}(x_{t-1})|x_{t-1}\right]\|_2 \leq \beta_2^{\mathtt{m}}(Q),$$

which gives

$$2\eta\mathbb{E}\left[f(x_{t-1}) - f(x^*)\right] \leq \mathbb{E}\left[\|x_{t-1} - x^*\|_2^2\right] - \mathbb{E}\left[\|x_t - x^*\|_2^2\right] + \eta^2\alpha_2^{\mathtt{m}}(Q)^2 + 2\eta D\beta_2^{\mathtt{m}}(Q).$$

Therefore, by summing from $t = 2$ to $T + 1$, dividing by T, and using assumption that the domain $\mathcal{X}$ has diameter at the most D, we have

$$2\eta\mathbb{E}\left[f(\bar{x}_T) - f(x^*)\right] \leq \frac{D^2}{T} + \eta^2\alpha_2^{\mathtt{m}}(Q)^2 + 2\eta D\beta_2^{\mathtt{m}}(Q).$$

The first statement of Theorem 3.3.2 follows upon dividing by η and setting the value of η as in the statement. The second statement holds in a similar manner by replacing α and β with α_2 and β_2, respectively. □

3.6.2 Proof of Theorem 3.3.3

Note that since the gradient descent step remains the same, (3.21) still holds. Except now, instead of (3.23), we have a stronger inequality

$$\mathbb{E}\left[f(x_{t-1}) - f(x^*)\right] \leq \mathbb{E}\left[(x_{t-1} - x^*)^T\mathbb{E}\left[\hat{g}(x_{t-1})|x_{t-1}\right]\right] - \frac{\gamma}{2}\|x_{t-1} - x^*\|_2^2, \tag{3.24}$$

Combining (3.21) and (3.24), and then using the definition of α, β as in the previous proof, we get

$$2\eta_{t-1}\mathbb{E}\left[f(x_{t-1}) - f(x^*)\right] \leq \mathbb{E}\left[\|x_{t-1} - x^*\|_2^2\right] - \mathbb{E}\left[\|x_t - x^*\|_2^2\right] + \eta_{t-1}^2\alpha_2(Q)^2 + 2\eta_{t-1}D\beta_2(Q)$$
$$- \frac{\gamma}{2}\|x_{t-1} - x^*\|_2^2.$$

Then multiplying by $(t-1)/2\eta_{t-1}$, we get

$$(t-1)\mathbb{E}\left[f(x_{t-1})-f(x^*)\right] \leq \frac{(t-1)\eta_{t-1}}{2}\alpha_2(Q)^2 + \left(\frac{t-1}{2\eta_{t-1}} - \frac{(t-1)\gamma}{2}\right)\mathbb{E}\left[\|x_{t-1}-x^*\|_2^2\right] - \frac{t-1}{2\eta_{t-1}}\mathbb{E}\left[\|x_t-x^*\|_2^2\right] + (t-1)D\beta_2(Q).$$

Substituting for η_{t-1} as in the theorem statement, summing the resultant inequality from $t=1$ to T, and then applying Jensen's inequality completes the proof.

3.6.3 Proof of Theorem 3.4.2

Step 1: Analysis of CUQ. We first prove a result for CUQ (with a dynamic range of $[-M, M]$) which will bound the expected value of

$$\sum_{i\in[d]}\left(Q_{\mathtt{u}}(Y)(i)-Y(i)\right)^2\mathbb{1}_{\{|Y(i)|\leq M\}},$$

namely the mean square error when there is no overflow. This will be useful in the analysis of RATQ, too.

Lemma 3.6.1. *For an $\mathbb{R}^d$-valued random variable Y and $Q_{\mathtt{u}}$ denoting the quantizer CUQ with parameters M (with dynamic range $[-M, M]$) and k, let $Q_{\mathtt{u}}(Y)$ be the quantized value of Y. Then,*

$$\mathbb{E}\left[\sum_{i\in[d]}\left(Q_{\mathtt{u}}(Y)(i)-Y(i)\right)^2\mathbb{1}_{\{|Y(i)|\leq M\}} \mid Y\right] \leq \frac{dM^2}{(k-1)^2}\left(\frac{1}{d}\sum_{j\in[d]}\mathbb{1}_{\{|Y(j)|\leq M\}}\right).$$

The proof is relatively straightforward with the calculations similar to [88, Theorem 2]; it is deferred to Section 3.7.1.

Also, the quantizer AGUQ in Section 3.5.2 uses the one-dimensional CUQ with dynamic range $[0, M]$ as a subroutine. The uniform levels for this variant of CUQ are given by

$$B_{M,k}(\ell) = \ell\cdot\frac{M}{k-1}, \forall \ell\in[k-1].$$

We have the following lemma for this variant of CUQ.

Lemma 3.6.2. *For an $\mathbb{R}$-valued random variable Y which is almost surely nonnegative and the quantizer $Q_{\mathtt{u}}$ with dynamic range $[0, M]$ and parameter k, let $Q_{\mathtt{u}}(Y)$ denote the quantized value of Y. Then,*

$$\mathbb{E}\left[\left(Q_{\mathtt{u}}(Y) - Y\right)^2 \mathbb{1}_{\{|Y| \le M\}} \mid Y\right] \le \frac{M^2}{4(k-1)^2}\left(\mathbb{1}_{\{|Y| \le M\}}\right).$$

The proof is very similar to the proof of Lemma 3.6.1 and is deferred to Section 3.7.1.

Step 2: Mean square error for adaptive quantizers. The quantizers RATQ and A-RATQ use ATUQ as subroutine; in addition, A-RATQ uses AGUQ for gain quantization. Thus, in order to analyze RATQ and A-RATQ, we need to analyze ATUQ and AGUQ first.

In this step we provide a general bound on the mean square error of adaptive quantizers. We capture the performances of ATUQ and AGUQ in two separate results below.

Lemma 3.6.3. *For an $\mathbb{R}^d$-valued random variable Y and Q denoting the quantizer ATUQ with dynamic-range parameters M_js, we have*

$$\mathbb{E}\left[\sum_{i\in[d]} \left(Q(Y)(i) - Y(i)\right)^2 \mathbb{1}_{\{|Y(i)| \le M_{h-1}\}}\right] \le \frac{d}{(k-1)^2}\left(m + m_0 + \sum_{j=1}^{h-1} M_j^2 P\left(\|Y\|_\infty > M_{j-1}\right)\right).$$

Proof. Consider the events A_js corresponding to different levels used by the adaptive quantizer of the norm, defined as follows:

$$\begin{aligned} A_0 &:= \{\|Y\|_\infty \le m\}, \\ A_j &:= \{M_{j-1} < \|Y\|_\infty \le M_j\}, \quad \forall j \in [h-2], \\ A_{h-1} &:= \{M_{h-2} < \|Y\|_\infty\}. \end{aligned}$$

By construction, $\sum_{j=0}^{h-1} \mathbb{1}_{A_j} = 1$ a.s.. Therefore, we have

$$\mathbb{E}\left[\sum_{i\in[d]}\Big(Q(Y)(i)-Y(i)\Big)^2\mathbb{1}_{\{|Y(i)|\le M_{h-1}\}}\right] = \mathbb{E}\left[\|Q(Y)-Y\|_2^2\mathbb{1}_{A_0}\right] + \sum_{j=1}^{h-2}\mathbb{E}\left[\|Q(Y)-Y\|_2^2\mathbb{1}_{A_j}\right] + \mathbb{E}\left[\sum_{i\in[d]}\Big(Q_{\mathtt{u}}(Y)(i)-Y(i)\Big)^2\mathbb{1}_{\{|Y(i)|\le M_{h-1}\}}\mathbb{1}_{A_{h-1}}\right].$$

Note that $\mathbb{1}_{A_0}$ implies that we are using a k-level uniform quantization with a dynamic range of $[-m, m]$. Therefore, this term can be bounded by Lemma 3.6.1 as follows:

$$\mathbb{E}\left[\|Q(Y)-Y\|_2^2\mathbb{1}_{A_0}\right] \le \frac{dm}{(k-1)^2}.$$

Under the event A_j with $j \in [h-1]$, we use a k-level uniform quantization with a dynamic range of $[-M_j, M_j]$. Therefore, by Lemma 3.6.1, we have

$$\begin{aligned}\mathbb{E}\left[\|Q(Y)-Y\|_2^2\mathbb{1}_{A_j}\right] &\le \frac{dM_j^2}{(k-1)^2}\mathbb{E}\left[\mathbb{1}_{A_j}\right]\\ &\le \frac{dM_j^2}{(k-1)^2}P\left(\|Y\|_\infty > M_{j-1}\right).\end{aligned}$$

□

Note that the proof above does note use specific form of M_j's and therefore applies as it is for the one-dimensional AGUQ gain quantizer used in A-RATQ; the only change is the fact that instead of using Lemma 3.6.1 for uniform quantization we use Lemma 3.6.2. This leads to the following lemma, which will be useful later in the analysis of A-RATQ.

Lemma 3.6.4. *For an $\mathbb{R}$-valued random variable Y which is almost surely nonnegative and Q denoting the quantizer AGUQ with dynamic-range parameters $M_{g,j}$s, we have*

$$\mathbb{E}\left[(Q(Y)-Y)^2\mathbb{1}_{\{|Y|\le M_{g,h-1}\}}\right] \le \frac{1}{4(k-1)^2}\left(B^2 + \sum_{j=1}^{h-1} M_{g,j}^2 P\left(|Y| > M_{g,j-1}\right)\right).$$

The proof is similar to that of Lemma 3.6.3 and is omitted.

Step 3: Mean square error of ATUQ for a subgaussian input vector. In our analysis, we need to evaluate the performance of ATUQ for *subgaussian* input vectors.

Definition 3.6.5 (*cf.* [13])**.** A centered random variable X is said to be subgaussian with variance factor v if for all λ in $\mathbb{R}$, we have

$$\ln \mathbb{E}\left[e^{\lambda X}\right] \leq \frac{\lambda^2 v}{2}.$$

The following well-known fact (*cf.* [13, Chapter 2]) will be used throughout.

Lemma 3.6.6. *For a centered subgaussian random variable X with variance factor v the*

$$P(|X| > x) \leq 2e^{-x^2/2v}, \quad \forall\, x > 0,$$
$$\mathbb{E}\left[X^2\right] \leq 4v, \quad \mathbb{E}\left[X^4\right] \leq 32v^2.$$

Next, consider the quantizer $Q_{\mathtt{at},I}$ which is similar to RATQ but skips the rotation step. Specifically, $Q_{\mathtt{at},I}$ is obtained by replacing the random matrix R in the encoder and decoder of RATQ (given in Algorithms 3.2 and 3.3, respectively) by the identity matrix I. Symbolically, the quantizer $Q_{\mathtt{at},I}$ can be described as follows for the d-dimensional input vector Y

$$Q_{\mathtt{at},I}(Y) = [Q_{\mathtt{at}}(Y_1)^T, \cdots, Q_{\mathtt{at}}(Y_{\lceil d/s \rceil})^T], \tag{3.25}$$

where $Q_{\mathtt{at}}$ is the quantizer ATUQ and Y_i is the i^{th} subvector of Y. Recall that the i^{th} subvector Y_i comprises the coordinates $\{(i-1)s+1, \cdots, \min\{is, d\}\}$, for all $i \in [d/s]$. Also, recall that the dimension of all the sub vectors except the last one is s, with the last one having dimension $d - s\lfloor d/s \rfloor$.

Notice that like RATQ, $Q_{at,I}$ has parameters k, h, s, m, and m_0 which need to be set. We set the parameters m and m_0 to be $3v$ and $2v \ln s$, respectively, and prove a general lemma in terms of the other parameters of $Q_{at,I}$ for a subgaussian input vector.

Lemma 3.6.7. *Consider $Y = [Y(1), \ldots, Y(d))]^T$, where for all i in $[d]$, $Y(i)$ is a centered subgaussian random variable with variance factor v. Let Q denote the quantizer $Q_{\mathtt{at},I}$ with*

parameters m and m_0 set to $3v$ and $2v \ln s$, respectively. Then, for every $s, k, h \in \mathbb{N}$, we have

$$\frac{1}{d} \cdot \mathbb{E}\left[\sum_{i\in[d]} (Y(i) - Q(Y)(i))^2 \mathbb{1}_{\{|Y(i)|\le M_{h-1}\}}\right] \le v \cdot \frac{9 + 3\ln s}{(k-1)^2}.$$

Proof. Since

$$\mathbb{E}\left[\sum_{i\in[d]} (Y(i) - Q(Y)(i))^2 \mathbb{1}_{\{|Y(i)|\le M_{h-1}\}}\right] = \sum_{i=1}^{\lceil \frac{d}{s} \rceil} \sum_{j=(i-1)s+1}^{\min\{is,d\}} \mathbb{E}\left[\left(Q_{\mathtt{at}}(Y)(j) - Y(j)\right)^2 \mathbb{1}_{\{|Y(j)|\le M_{h-1}\}}\right],$$

by using Lemma 3.6.3 for each of the $\lceil d/s \rceil$ subvectors, we get

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i\in[d]} (Y(i) - Q(Y)(i))^2 \mathbb{1}_{\{|Y(i)|\le M_{h-1}\}}\right] \\
&\le \frac{s}{(k-1)^2} \sum_{i\in 1}^{\lfloor \frac{d}{s} \rfloor} \left(m + m_0 + \sum_{j\in[h-1]} M_j^2 P\left(\|Y_{i,s}\|_\infty > M_{j-1}\right)\right) \\
&\quad + \frac{(d - s\lfloor \frac{d}{s} \rfloor)}{(k-1)^2} \left(m + m_0 + \sum_{j\in[h-1]} M_j^2 P\left(\|Y_{\lceil d/s \rceil, s}\|_\infty > M_{j-1}\right)\right).
\end{aligned}$$

For all $i \in \lfloor d/s \rfloor$, it follows from the union bound that

$$P\left(\|Y_{1,s}\|_\infty > M_{j-1}\right) \le 2se^{\frac{-M_{j-1}^2}{2v}}.$$

Also, since $d - s\lfloor d/s \rfloor \le s$, we have

$$P\left(\|Y_{\lceil d/s \rceil, s}\|_\infty > M_{j-1}\right) \le 2se^{\frac{-M_{j-1}^2}{2v}}.$$

Using these tail bounds in the previous inequality, we get

$$\mathbb{E}\left[\sum_{i\in[d]} (Y(i) - Q(Y)(i))^2 \mathbb{1}_{\{|Y(i)|\le M_{h-1}\}}\right] \le \frac{d}{(k-1)^2} \left(m + m_0 + 2s \sum_{j\in[h-1]} M_j^2 e^{\frac{-M_{j-1}^2}{2v}}\right).$$

Setting $m = 3v$ and $m_0 = 2v$, the summation on the right-side is bounded further as

$$\begin{aligned}
&2s\left(\frac{3v}{s}\sum_{j=1}^{h-1}(e^{*j})\cdot e^{-1.5e^{*(j-1)}}\right) + 2s\left(\frac{2v}{s}\sum_{j=1}^{h-1} e^{-1.5e^{*(j-1)}}\right) \\
&= 6v\sum_{j=1}^{h-1} e^{-0.5e^{*(j-1)}} + 4v\ln s\sum_{j=1}^{h-1} e^{-1.5e^{*(j-1)}} \\
&\leq 6v\sum_{j=1}^{\infty} e^{-0.5e^{*(j-1)}} + 4v\ln s\sum_{j=1}^{h-1} e^{-1.5e^{*(j-1)}} \\
&\leq 6v + v\ln s,
\end{aligned}$$

where we use a bound of 1 for $\sum_{j=1}^{\infty} e^{-0.5e^{*(j-1)}}$, whose validity can be seen as follows[11]

$$\begin{aligned}
\sum_{j=1}^{\infty} e^{-0.5e^{*(j-1)}} &= e^{-0.5} + e^{-0.5e} + e^{-0.5e^e} + \sum_{j=3}^{\infty} e^{-0.5e^{*(j)}} \\
&\leq e^{-0.5} + e^{-0.5e} + e^{-0.5e^e} + \sum_{j=3}^{\infty} e^{-0.5je^e} \\
&\leq e^{-0.5} + e^{-0.5e} + e^{-0.5e^e} + \frac{1}{e^{e^e}-1} \\
&\leq 1,
\end{aligned}$$

and $1/4$ for $\sum_{j=1}^{h-1} e^{-1.5e^{*(j-1)}}$, whose validity can be seen as follows

$$\begin{aligned}
\sum_{j=1}^{\infty} e^{-1.5e^{*(j-1)}} &= e^{-1.5} + e^{-1.5e} + e^{-1.5e^e} + \sum_{j=3}^{\infty} e^{-1.5e^{*(j)}} \\
&\leq e^{-1.5} + e^{-1.5e} + e^{-1.5e^e} + \sum_{j=3}^{\infty} e^{-1.5je^e} \\
&\leq e^{-1.5} + e^{-1.5e} + e^{-1.5e^e} + \frac{1}{e^{3e^e} - 1/e^{1.5e^e}} \\
&\leq 0.2401.
\end{aligned}$$

[11] In fact, these bounds motivate the use of tetration as our choice for M_js.

Therefore, we obtain

$$\frac{1}{d} \cdot \mathbb{E}\left[\sum_{i\in[d]} (Y(i) - Q(Y)(i))^2 \mathbb{1}_{|Y(i)|\leq M_{h-1}}\right] \leq v \cdot \frac{9 + 3\ln s}{(k-1)^2}.$$

□

We remark that calculations present in this lemma are at the heart of the analysis of RATQ. Also, this lemma will be useful for other applications discussed in Chapters 5 and 6.

Step 4: Completing the proof. Recall that the random matrix R defined in (3.6) is used at the encoder of RATQ to randomly rotate the input vector. We observe that the rotated vector has subgaussian entries.

Lemma 3.6.8. *For an $\mathbb{R}^d$-valued random variable Y such that $\|Y\|_2^2 \leq B^2$ a.s., all coordinates of the rotated vector RY are centered subgaussian random variables with a variance factor of B^2/d, whereby*

$$P(|RY(j)| \geq M) \leq 2e^{-dM^2/2B^2}, \quad \forall\, j \in [d],$$

where $RY(j)$ is the j^{th} coordinate of the rotated vector.

The proof uses similar calculations as [7] and [88]; it is deferred to Section 3.7.2.

Intuitively, the Lemma 3.6.8 highlights the fact that overall energy $\|Y\|_2^2$ in the input vector Y is divided equally among all the coordinates after random rotation.

The worst-case second moment of RATQ. Note that by the description of RATQ which will be denoted by $Q_{\mathtt{at},R}(RY)$, we have that

$$Q_{\mathtt{at},R}(Y) = R^{-1} Q_{\mathtt{at},I}(RY),$$

where $Q_{\mathtt{at},I}$ is as defined in (3.25). Thus,

$$Q_{\mathtt{at},I}(RY) = [Q_{\mathtt{at}}(RY_{1,s})^T, \cdots, Q_{\mathtt{at}}(RY_{\lceil d/s \rceil,s})^T]^T, \tag{3.26}$$

where the subvector $RY_{i,s}$ is given by

$$RY_{i,s} = [RY((i-1)s+1), \cdots, RY(\min\{is, d\})]^T.$$

To compute $\alpha(Q_{\mathtt{at},R}(Y))$, we will first compute the second moment for the output of RATQ. Specifically, using the fact R is a unitary transform, we obtain

$$\begin{aligned}
\mathbb{E}\left[\|Q_{\mathtt{at},R}(Y)\|_2^2\right] &= \mathbb{E}\left[\|R^{-1}Q_{\mathtt{at},I}(RY)\|_2^2\right] \\
&= \mathbb{E}\left[\|Q_{\mathtt{at},I}(RY)\|_2^2\right] \\
&= \sum_{j\in[d]} \mathbb{E}\left[(Q_{\mathtt{at},I}(RY)(j))^2\right] \\
&= \sum_{i=1}^{\lceil \frac{d}{s} \rceil} \sum_{j=(i-1)s+1}^{\min\{is,d\}} \mathbb{E}\left[(Q_{\mathtt{at},I}(RY)(j))^2\right].
\end{aligned}$$

We now observe that for our choice of m and h for RATQ given by (3.8), we have

$$M_{h-1}^2 \geq m(e^{*\log_e^*(d/3)}) = (3B^2/d).(d/3) = B^2.$$

Using this observation and noting that R is a unitary matrix, we have that

$$\mathbb{1}_{\{\|RY\|_2 \leq M_{h-1}\}} = 1 \text{ a.s..}$$

Also, noting that $|RY(j)| \leq \|RY\|_2 = \|Y\|_2 = B$ a.s., for all $j \in [d]$, we get

$$\mathbb{1}_{\{|RY(j)| \leq M_{h-1}\}} = 1 \text{ a.s.}, \forall j \in [d]. \tag{3.27}$$

Proceeding with these observations, we get

$$\begin{aligned}\mathbb{E}\left[\|Q_{\mathtt{at},R}(Y)\|_2^2\right] &\le \sum_{i=1}^{\lceil \frac{d}{s} \rceil} \sum_{j=(i-1)s+1}^{\min\{is,d\}} \mathbb{E}\left[(Q_{\mathtt{at},I}(RY)(j))^2 \mathbb{1}_{\{|RY(j)|\le M_{h-1}\}}\right] \\ &= \sum_{i=1}^{\lceil \frac{d}{s} \rceil} \sum_{j=(i-1)s+1}^{\min\{is,d\}} \mathbb{E}\left[(Q_{\mathtt{at},I}(RY)(j) - RY(j) + RY(j))^2 \mathbb{1}_{\{|RY(j)|\le M_{h-1}\}}\right] \\ &\le \sum_{i=1}^{\lceil \frac{d}{s} \rceil} \sum_{j=(i-1)s+1}^{\min\{is,d\}} \mathbb{E}\left[\left((Q_{\mathtt{at},I}(RY)(j) - RY(j))^2 + RY(j)^2\right) \mathbb{1}_{\{|RY(j)|\le M_{h-1}\}}\right],\end{aligned}$$

where the previous inequality uses the fact that, under the event $\{|RY(j)| \le M_{h-1}\}$, $Q_{\mathtt{at},I}(RY)(j)$ is an unbiased estimate of $RY(j)$. Namely,

$$\mathbb{E}\left[Q_{\mathtt{at},I}(RY)(j)\mathbb{1}_{\{|RY(j)|\le M_{h-1}\}} \mid R, Y\right] = \mathbb{E}\left[RY(j)\mathbb{1}_{\{|RY(j)|\le M_{h-1}\}} \mid R, Y\right].$$

Therefore, noting that R is a unitary matrix, we have

$$\mathbb{E}\left[\|Q_{\mathtt{at},R}(Y)\|_2^2\right] \le \mathbb{E}\left[\sum_{j\in[d]} (RY(i) - Q_{\mathtt{at},I}(RY)(j))^2 \mathbb{1}_{\{|RY(j)|\le M_{h-1}\}}\right] + \mathbb{E}\left[\|Y\|_2^2\right].$$

To bound the first term on the right-side we have the following lemma.

Lemma 3.6.9. *For an $\mathbb{R}^d$-valued random variable Y such that $\|Y\|_2^2 \le B^2$ a.s.. Then, for m and m_0 set to be $3B^2/d$ and $(2B^2/d)\ln s$, respectively, we have that*

$$\mathbb{E}\left[\sum_{j\in[d]} (RY(i) - Q_{\mathtt{at},I}(RY)(j))^2 \mathbb{1}_{\{|RY(j)|\le M_{h-1}\}}\right] \le B^2 \cdot \frac{9 + 3\ln s}{(k-1)^2}.$$

Proof. By Lemma 3.6.8 we have that all coordinates $RY(j)$ are centered subgaussian random variable with variance factor B^2/d. Thus, the parameters m and m_0 of RATQ set as in (3.8), and equal $3v$ and $2v\ln s$, respectively, where v is the variance factor of each subgaussian coordinate. The result follows by invoking Lemma 3.6.7. □

Therefore, for any Y such that $\|Y\|_2^2 \leq B^2$, we have

$$\mathbb{E}\left[\|Q_{\mathtt{at},R}(Y)\|_2^2\right] \leq B^2 \cdot \frac{9 + 3\ln s}{(k-1)^2} + B^2,$$

whereby

$$\alpha_2(Q_{\mathtt{at},R}) \leq B\sqrt{\frac{9 + 3\ln s}{(k-1)^2} + 1}.$$

The worst-case bias of RATQ. By (3.27) we have that the input always remains in the dynamic-range of the quantizer, resulting in unbiased quantized values. In other words, $\beta_2(Q_{\mathtt{at},\mathtt{R}}) = 0$.

3.6.4 Proof of Lemma 3.5.5

We first note AGUQ is used to quantize a scalar Y. It follows from the description of the quantizer that

$$\mathbb{1}_{\{|Y| \leq M_{g,h_g-1}\}} \mathbb{E}\left[Q_{\mathtt{a}}(Y)|Y\right] = \mathbb{1}_{\{|Y| \leq M_{g,h_g-1}\}} Y, \tag{3.28}$$

and that[12]

$$\mathbb{1}_{\{|Y| > M_{g,h_g-1}\}} Q_{\mathtt{a}}(Y) = 0. \tag{3.29}$$

The worst-case second moment of AGUQ. Towards evaluating $\alpha(Q_{\mathtt{a}})$ for AGUQ, for any $Y \in \mathbb{R}$ we have

$$\begin{aligned}
\mathbb{E}\left[Q_{\mathtt{a}}(Y)^2\right] &= \mathbb{E}\left[Q_{\mathtt{a}}(Y)^2 \mathbb{1}_{\{|Y| \leq M_{g,h_g-1}\}}\right] + \mathbb{E}\left[Q_{\mathtt{a}}(Y)^2 \mathbb{1}_{\{|Y| > M_{g,h-1}\}}\right] \\
&= \mathbb{E}\left[(Q_{\mathtt{a}}(Y) - Y + Y)^2 \mathbb{1}_{\{|Y| \leq M_{g,h_g-1}\}}\right] + \mathbb{E}\left[Q_{\mathtt{a}}(Y)^2 \mathbb{1}_{\{|Y| > M_{g,h_g-1}\}}\right] \\
&= \mathbb{E}\left[(Q_{\mathtt{a}}(Y) - Y)^2 \mathbb{1}_{\{|Y| \leq M_{g,h_g-1}\}}\right] + \mathbb{E}\left[Y^2 \mathbb{1}_{\{|Y| \leq M_{g,h_g-1}\}}\right],
\end{aligned}$$

[12] Once again, this follows from our convention that the outflow symbol is evaluated to 0.

where the last identity uses (3.29), and the fact that $\mathbb{E}\left[(Q_{\mathtt{a}}(Y)-Y)Y\mathbb{1}_{\{|Y|\le M_{g,h-1}\}}|Y\right]=0$, which follows from (3.28). From Lemma 3.6.4 it follows that

$$\mathbb{E}\left[(Q_{\mathtt{a}}(Y)-Y)^2\mathbb{1}_{\{|Y|\le M_{g,h-1}\}}\right]\le\frac{1}{4(k_g-1)^2}\left(B^2+\sum_{j=1}^{h-1}M_j^2P\left(|Y|>M_{g,j-1}\right)\right).$$

By Markov's inequality we get that for any random variable Y with $\mathbb{E}[Y^2]\le B^2$, we have $P(|Y|>M_{g,j-1})\le B^2/M_{g,j-1}^2$, which further leads to

$$\begin{aligned}\mathbb{E}\left[(Q_{\mathtt{a}}(Y)-Y)_2^2\mathbb{1}_{\{|Y|\le M_{g,h-1}\}}\right]&\le\frac{B^2}{4(k_g-1)^2}+\sum_{j=1}^{h_g-1}\frac{M_{g,j}^2}{4(k_g-1)^2}\frac{B^2}{M_{g,j-1}^2}\\&=\frac{B^2}{4(k_g-1)^2}+\frac{a_g(h_g-1)B^2}{4(k_g-1)^2}.\end{aligned}$$

Therefore, we have

$$\mathbb{E}\left[Q_{\mathtt{a}}(Y)^2\right]\le\frac{B^2}{4(k_g-1)^2}+\frac{a_g(h_g-1)B^2}{4(k_g-1)^2}+\mathbb{E}\left[Y^2\mathbb{1}_{\{|Y|\le M_{g,h-1}\}}\right].$$

The result follows upon taking the supremum of the left-side over all random variables Y with $\mathbb{E}[Y^2]\le B^2$.

The worst-case bias of AGUQ. Towards evaluating $\beta(Q_{\mathtt{a}})$, we note first using Jensen's inequality that

$$\left|\mathbb{E}\left[Q_{\mathtt{a}}(Y)-Y\right]\right|\le\mathbb{E}\left[\left|\mathbb{E}\left[Q_{\mathtt{a}}(Y)-Y|Y\right]\right|\right].$$

Then, for Y with $\mathbb{E}[Y^2]\le B^2$, using (3.28) and Markov's inequality, we get

$$\begin{aligned}\mathbb{E}\left[\left|\mathbb{E}\left[Q_{\mathtt{a}}(Y)-Y|Y\right]\right|\right]&=\mathbb{E}\left[|Y|\mathbb{1}_{\{|Y|\ge M_{g,h-1}\}}\right]\\&\le\sqrt{\mathbb{E}\left[Y^2\right]P(|Y|\ge M_{g,h-1})}\\&\le\frac{B^2}{M_{g,h-1}}.\end{aligned}\tag{3.30}$$

Therefore, for any Y with $\mathbb{E}[Y^2] \leq B^2$, we have

$$\left|\mathbb{E}\left[Q_{\mathtt{a}}(Y) - Y\right]\right| \leq \sup_{Y \geq 0 \text{a.s.}: \mathbb{E}[Y^2] \leq B^2} \mathbb{E}\left[\left|\mathbb{E}\left[Q_{\mathtt{a}}(Y) - Y | Y\right]\right|\right] \leq \frac{B^2}{M_{g,h-1}}.$$

The result follows upon taking the supremum of left-side over all random variables Y with $\mathbb{E}[Y^2] \leq B^2$. □

3.6.5 Proof of Lemma 3.5.10

$O(1)$ expected precision. Recall that the variable-length bit string is a concatenation of two bit strings: The first bit string represents the dynamic range $M_{g,j}$, $j \in \{0, \ldots h-1\}$; the second-bit string represents the uniform level within that dynamic range.

The first string uses the first h symbols of the Huffman codes corresponding to the geometric distribution with parameter $1/2$. Its code length can be bounded as follows. By Markov's inequality, we have $P(|Y| > M_{g,j-1}) \leq B^2/M_{g,j-1}^2 = a_g^{-j} = 2^{-j+1}$. For a symbol $j \in \{0, \cdots, h-1\}$ representing the chosen dynamic range, let $\ell(j)$ denote the length of the codeword of that symbol. Therefore, the expected codelength $\mathbb{E}[L]$ can be upper bounded as follows.

$$\begin{aligned}\mathbb{E}[L] &\leq \sum_{j=0}^{h-1} P(|Y| > M_{g,j-1})\ell(j) \\ &\leq 2\sum_{j=0}^{h-1} 2^{-j}\ell(j).\end{aligned}$$

Since we will assign code-lengths $\ell(j)$ as that assigned to the first h symbols of the Huffman code corresponding to the geometric distribtution with parameter $1/2$, we have the following.

$$\begin{aligned}\mathbb{E}[L] &\leq 2\sum_{j=0}^{h-1} 2^{-j}\ell(j) \\ &= 4\sum_{j=0}^{h-1} 2^{-j-1}\ell(j) \\ &\leq 4(H(Z)+1),\end{aligned}$$

where Z is the geometric distribution with parameter $1/2$. Above, the final inequality can be seen by the fact that expected code-length for Huffman codes for a particular pmf is upper bounded by one plus entropy of that pmf. This bounds the code-length of the first bit string by a constant.

Coming to the bounding the code-length of the second bit string, the expected code length of the second string is upper bounded by

$$\sum_{j=0}^{h-1} P(|Y| > M_{g,j-1}) \log(k_{g,j}+1) \leq 2 \sum_{j=0}^{h-1} 2^{-j}(j+1) \leq 12.$$

Worst-case second moment of AGUQ$^+$. The only change from the worst-case second moment upper bound calculations in the proof of Lemma 3.5.5 is that the number of uniform levels for different dynamic ranges is different. The rest of proof remains precisely the same, and is skipped.

Bias of AGUQ$^+$. The proof is identical to one in Lemma 3.5.5, and is skipped. □

3.6.6 Proof of Theorems 3.5.3 and 3.5.4

Before we proceed with our lower bounds, we will set up some notation. We consider quantizers of the form

$$Q(Y) = Q_g(\|Y\|_2) Q_s(Y/\|Y\|_2).$$

Let $W(\cdot|y)$, $W_g(\cdot|y)$, and $W_s(\cdot|y)$, respectively, denote the distribution of the output of quantizers $Q(y)$, $Q_g(y)$, and $Q_s(y)$. We prove a general lower bound for a quantizer satisfying Structural Assumptions 1-3 in Section 3.5.1 in terms of the precision r; Theorems 3.5.3 and 3.5.4 are obtained as corollaries of this general lower bound.

Theorem 3.6.10. *Suppose that $\mathcal{X}$ contains the set $\{x \in \mathbb{R}^d : \|x\|_2 \leq D/2\}$. Consider a gain-shape quantizer Q of precision r satisfying the Assumptions 1-3 in Section 3.5.1. Then, there exists an oracle $(f, O) \in \mathcal{O}$ such that for any optimization protocol π using T*

iterations, we have

$$\mathcal{E}(f, O, \pi, Q) \geq \frac{DB}{2\sqrt{2}} \min\left\{\frac{1}{2^r}, \frac{1}{4 \cdot 2^{r/3} T^{1/3}}, \frac{1}{2(2T)^{1/3}}\right\}.$$

Proof. Consider the function $f_\alpha : \mathbb{R}^d \to \mathbb{R}$, $\alpha \in \{-1, 1\}$ given by

$$f_\alpha(x) := \delta \frac{B}{\sqrt{2}} |x(1) - \alpha D/2|, \quad \alpha \in \{-1, 1\}.$$

Note that the functions f_1 and f_{-1} are convex and depend only on the first coordinate of x. Further, for $x \in \mathcal{X}$, the subgradient of f_α is $-\delta\alpha B e_1/\sqrt{2}$, since $sign(x(1) - \alpha D/2) = -sign(\alpha)$, where e_1 is the vector $[1, 0, 0, \ldots, 0]^T$. We consider oracles O_α, $\alpha \in \{-1, 1\}$, that produce noisy subgradient updates with distribution

$$P_\alpha\left(\frac{B}{\sqrt{2}} e_1\right) = \frac{1 - \delta^2}{2}, \quad P_\alpha\left(\frac{-B}{\sqrt{2}} e_1\right) = \frac{1 - \delta^2}{2}, \quad P_\alpha\Big(-\frac{\alpha B}{\sqrt{2}\delta} e_1\Big) = \delta^2.$$

It is easy to check that the oracle outputs satisfy (2.5) and (3.3) described in Section 3.3.1. That is, the output of O_α is an unbiased estimate of the subgradient of f_α, and the expected Euclidean norm square of the oracle output is bounded by B^2.

We now take recourse to the standard reduction of optimization to hypothesis testing: To estimate the optimal value of f_1 and f_{-1} to an accuracy δ, the optimization protocol must determine if the oracle outputs are generated by P_1 or P_{-1}. However in order to distinguish between P_1 or P_{-1}, the optimization protocol only has access to the quantized oracle outputs. Specifically, the protocol sees the samples from $Q(Y)$ at every time step, where Y has distribution either P_1 or P_{-1}.

Denoting by $P_\alpha W$ the distribution of the output $Q(Y)$ when the input Y is generated from P_α, we have from the standard reduction (see, for instance, [23, Theorem 5.2.4]) that

$$\max_{\alpha \in \{-1,1\}} \mathcal{E}(f, O, \pi, Q) \geq \frac{DB}{2\sqrt{2}} \delta \left(1 - \sqrt{\frac{T}{2} \chi^2(P_1 W, P_{-1} W)}\right),$$

where $\chi^2(P, Q) = \sum_x (P(x) - Q(x))^2 / Q(x)$ denotes the chi-squared divergence between P

and Q.

Note that Assumption 2 on the structure of the quantizer implies that when $M < B/\delta\sqrt{2}$, the distributions P_1W and $P_{-1}W$ are the same. It follows that for every $\delta < \min\{\sqrt{B^2/2M^2}, 1\}$, the left-side of the previous inequality exceeds $(DB/2\sqrt{2})\delta$, whereby

$$\max_{\alpha\in\{-1,1\}} \mathcal{E}(f, O, \pi, Q) \geq \frac{DB}{2\sqrt{2}} \min\left\{\frac{B}{\sqrt{2}M}, 1\right\}. \tag{3.31}$$

Next, we consider the following modification of the previous construction in the case when $B/\sqrt{2} < m$:

$$P_\alpha\left(\frac{B}{\sqrt{2}}e_1\right) = \frac{1-\delta^{1+y}}{2}, \quad P_\alpha\left(\frac{-B}{\sqrt{2}}e_1\right) = \frac{1-\delta^{1+y}}{2}, \quad P_\alpha\Big(-\frac{\alpha B}{\sqrt{2}\delta^y}e_1\Big) = \delta^{1+y}.$$

for $y \in [0, 1]$. Once again, the oracle outputs satisfy (2.5) and (3.3) described in Section 3.3.1. In this case, the vector $Y \sim P_\alpha$ has entries with ℓ_2 norm at the most $B/(\sqrt{2}\delta^y)$. We set y such that this value is less than m and $\chi^2(P_1W, P_{-1}W)$ is minimized. Note that if $B/(\delta^y\sqrt{2d}) < m$, then $\texttt{supp}(Q_g(\|a\|)) \subseteq \{0, m\}$ for all the a's in the support of P_1 or P_{-1}.

For all $z \neq 0$, $z \in \texttt{supp}(Q(a))$, when a is in the support of P_1 or P_{-1}, we have

$$W\Big(z \mid a\Big) = W_g\Big(m \mid \|a\|_2\Big)W_s\Big(\frac{z}{m} \mid \frac{a}{\|a\|_2}\Big).$$

Therefore,

$$P_1W(z) - P_{-1}W(z) = \delta^{1+y}W_g\Big(m \mid \frac{B}{\sqrt{2}\delta^y}\Big)\left(W_s\Big(\frac{z}{m} \mid -e_1\Big) - W_s\Big(\frac{z}{m} \mid e_1\Big)\right)$$

$$P_{-1}W(z) \geq \frac{1-\delta^{1+y}}{2}W_g\Big(m \mid \frac{B}{\sqrt{2}}\Big)W_s\Big(\frac{z}{m} \mid e_1\Big) + \frac{1-\delta^{1+y}}{2}W_g\Big(m \mid \frac{B}{\sqrt{2}}\Big)W_s\Big(\frac{z}{m} \mid -e_1\Big).$$

Using the preceding two inequalities

$$\frac{(P_1W(z) - P_{-1}W(z))^2}{P_{-1}W(z)} \leq \frac{\delta^{2+2y}W_g\Big(m \mid \frac{B}{\sqrt{2}\delta^y}\Big)^2\left(W_s\Big(\frac{z}{m} \mid e_1\Big) - W_s\Big(\frac{z}{m} \mid -e_1\Big)\right)^2}{\frac{1-\delta^{1+y}}{2}W_g\Big(m \mid \frac{B}{\sqrt{2}}\Big)W_s\Big(\frac{z}{m} \mid e_1\Big) + \frac{1-\delta^{1+y}}{2}W_g\Big(m \mid \frac{B}{\sqrt{2}}\Big)W_s\Big(\frac{z}{m} \mid -e_1\Big)}$$

$$\leq \frac{2\delta^{2+2y}}{1-\delta^{1+y}} \cdot \frac{W_g\Big(m \mid \frac{B}{\sqrt{2}\delta^y}\Big)^2 \Big(W_s\Big(\frac{z}{m} \mid e_1\Big) + W_s\Big(\frac{z}{m} \mid -e_1\Big)\Big)}{W_g\Big(m \mid \frac{B}{\sqrt{2}}\Big)}$$
$$\leq \frac{2\delta^{2+y}}{1-\delta^{1+y}} \cdot W_g\Big(m \mid \frac{B}{\sqrt{2}\delta^y}\Big) \Big(W_s\Big(\frac{z}{m} \mid e_1\Big) + W_s\Big(\frac{z}{m} \mid -e_1\Big)\Big)$$
$$\leq \frac{2\delta^{2+y}}{1-\delta^{1+y}} \cdot \Big(W_s\Big(\frac{z}{m} \mid e_1\Big) + W_s\Big(\frac{z}{m} \mid -e_1\Big)\Big),$$

where the third inequality uses Assumption 3b for the quantizer in Section 3.5.1, i.e., it uses

$$\frac{W_g\Big(m \mid \frac{B}{\sqrt{2}\delta^y}\Big)}{W_g\Big(m \mid \frac{B}{\sqrt{2}}\Big)} \leq \delta^{-y}.$$

For $z = 0$, $z \in \mathtt{supp}(Q(a))$, when a is in the support of P_1 or P_{-1}, we have

$$W\Big(0 \mid a\Big) = W_g\Big(0 \mid \|a\|_2\Big) + W_g\Big(m \mid \|a\|_2\Big) W_s\Big(0 \mid a/\|a\|_2\Big).$$

Therefore, by similar calculations for $z \neq 0$, we have

$$\frac{(P_1W(0) - P_{-1}W(0))^2}{P_{-1}W(0)} \leq \frac{\delta^{2+2y} W_g\Big(m \mid \frac{B}{\sqrt{2}\delta^y}\Big)^2 \Big(W_s\Big(0 \mid e_1\Big) + W_s\Big(0 \mid -e_1\Big)\Big)^2}{(\frac{1-\delta^{1+y}}{2}) W_g\Big(m \mid \frac{B}{\sqrt{2}}\Big) W_s\Big(0 \mid e_1\Big) + (\frac{1-\delta^{1+y}}{2}) W_g\Big(m \mid \frac{B}{\sqrt{2}}\Big) W_s\Big(0 \mid -e_1\Big)}$$
$$\leq \frac{2\delta^{2+y}}{1-\delta^{1+y}} \Big(W_s\Big(0 \mid e_1\Big) + W_s\Big(0 \mid -e_1\Big)\Big).$$

In conclusion,

$$\chi^2(P_1W, P_{-1}W) \leq \frac{4\delta^{2+y}}{1-\delta^{1+y}}.$$

Now, if $\delta < 1/2$, we have

$$\chi^2(P_1W, P_{-1}W) \leq 8\delta^{2+y}.$$

Upon setting $\delta = (16T)^{-1/(2+y)}$, which satisfies $\delta < 1/2$ for all T, we get

$$\max_{\alpha \in \{-1,1\}} \mathcal{E}(f, O, \pi, Q) \geq \frac{DB}{2\sqrt{2}} \delta (1 - \sqrt{4T\delta^{2+y}}) = \frac{DB}{4\sqrt{2}} \Big(\frac{1}{16T}\Big)^{\frac{1}{2+y}}. \tag{3.32}$$

But we can only set δ to this value if

$$\frac{B}{\sqrt{2}} \cdot (16T)^{\frac{y}{2+y}} < m. \tag{3.33}$$

Thus, for each y such that (3.33) holds, we get (3.32). Taking the the supremum of RHS in (3.32) over all $y \in [0,1]$ such that (3.33) holds, we obtain whenever $B/\sqrt{2} \leq m$,

$$\max_{\alpha \in \{-1,1\}} \mathcal{E}(f, O, \pi, Q) \geq \frac{DB}{2\sqrt{2}} \cdot \min\left\{\frac{1}{8}\sqrt{\frac{m\sqrt{2}}{BT}}, \frac{1}{2(2T)^{1/3}}\right\},$$

where we use the following lemma proved in Section 3.7.3.

Lemma 3.6.11. *For $a, c > 0$, and $b > 1$.*

$$\sup_{y \in [0,1]: a(b)^{y/(2+y)} < c.} a\left(\frac{1}{b}\right)^{\frac{1}{2+y}} = \min\left\{\sqrt{\frac{ca}{b}}, \frac{a}{b^{\frac{1}{3}}}\right\}$$

Upon combining this bound with (3.31), we obtain

$$\sup_{(f,O)\in\mathcal{O}} \varepsilon(f_\alpha, \pi^{QO}) \geq \frac{DB}{2\sqrt{2}} \max\left\{\min\left\{\frac{cm}{M}, 1\right\}, \min\left\{\frac{1}{8}\sqrt{\frac{1}{cT}}, \frac{1}{2(2T)^{1/3}}\right\} \mathbb{1}_{\{c<1\}}\right\},$$

where $c = B/(m\sqrt{2})$. By making cases $1 \leq c$, $\frac{1}{8(2T)^{1/3}} \leq c < 1$, and $c < \frac{1}{8(2T)^{1/3}}$, and using the fact that for $a, b \geq 0$, $\max\{a, b\} \geq a^{1/3}b^{2/3}$ in the second case, we get

$$\sup_{(f,O)\in\mathcal{O}} \varepsilon(f_\alpha, \pi^{QO}) \geq \frac{DB}{2\sqrt{2}} \min\left\{1, \frac{1}{(M/m)}, \frac{1}{4(M/m)^{1/3}T^{1/3}}, \frac{1}{2(2T)^{1/3}}\right\}.$$

By Assumption 3 in Section 3.5.1, we know that $\frac{M}{m} \leq 2^r$. Therefore,

$$\sup_{(f,O)\in\mathcal{O}} \varepsilon(f_\alpha, \pi^{QO}) \geq \frac{DB}{2\sqrt{2}} \min\left\{\frac{1}{2^r}, \frac{1}{4(2)^{r/3}T^{1/3}}, \frac{1}{2(2T)^{1/3}}\right\}.$$

□

Theorem 3.5.4 follows as an immediate corollary; Theorem 3.5.3, too, is obtained by

noting that

$$\sup_{(f,O)\in\mathcal{O}} \varepsilon(f_\alpha, \pi^{QO}) < \frac{3DB}{\sqrt{T}}$$

holds only if $\sqrt{T} < 2^r$.

3.7 Remaining proofs for the main results

3.7.1 Analysis of CUQ: Proof of Lemmas 3.6.1 and 3.6.2

Proof of Lemma 3.6.1: Denoting by $\mathcal{B}_{j,\ell}$ the event $\Big\{Y(j) \in [B_{M,k}(\ell), B_{M,k}(\ell+1))\Big\}$, we get

$$\begin{aligned}
&\mathbb{E}\left[\sum_{j\in[d]} \Big(Q_{\mathtt{u}}(Y)(j) - Y(j)\Big)^2 \mathbb{1}_{\{|Y(j)|\le M\}} \mid Y\right] \\
&= \sum_{j\in[d]} \sum_{\ell=0}^{k-1} \mathbb{E}\left[\Big(Q_{\mathtt{u}}(Y)(j) - Y(j)\Big)^2 \mathbb{1}_{\mathcal{B}_{j,\ell}} \mid Y\right] \mathbb{1}_{\{|Y(j)|\le M\}}.
\end{aligned}$$

For the term inside the summation on the right-side, we obtain

$$\begin{aligned}
&\mathbb{E}\left[\Big(Q_{\mathtt{u}}(Y)(j) - Y(j)\Big)^2 \mathbb{1}_{\mathcal{B}_{j,\ell}} \mid Y\right] \\
&= \left((B_{M,k}(\ell+1) - Y(j))^2 \frac{Y(j) - B_{M,k}(\ell)}{B_{M,k}(\ell+1) - B_{M,k}(\ell)}\right) \mathbb{1}_{\mathcal{B}_{j,\ell}} \\
&\qquad + \left((B_{M,k}(\ell) - Y(j))^2 \frac{B_{M,k}(\ell+1) - Y(j)}{B_{M,k}(\ell+1) - B_{M,k}(\ell)}\right) \mathbb{1}_{\mathcal{B}_{j,\ell}} \\
&= (B_{M,k}(\ell+1) - Y(j))(Y(j) - B_{M,k}(\ell))\, \mathbb{1}_{\mathcal{B}_{j,\ell}} \\
&\le \frac{1}{4} (B_{M,k}(\ell+1) - B_{M,k}(\ell))^2 \\
&= \frac{M^2}{(k-1)^2},
\end{aligned} \tag{3.34}$$

where the inequality uses the GM-AM inequality and the final identity is simply by the definition of $B_{M,k}(\ell)$. Upon combining the bounds above, we obtain

$$\mathbb{E}\left[\sum_{j\in[d]} \Big(Q_{\mathtt{u}}(Y)(j) - Y(j)\Big)^2 \mathbb{1}_{\{|Y(j)|\le M\}} \mid Y\right] \le \frac{dM^2}{(k-1)^2} \cdot \frac{1}{d} \sum_{j\in[d]} \mathbb{1}_{\{|Y(j)|\le M\}}.$$

□

Proof of Lemma 3.6.2 The proof is the same as the proof of Lemma 3.6.1, except that we need to set $d = 1$ and replace the identity used in(3.34) with

$$B_{M,k}(\ell + 1) - B_{M,k}(\ell) = \frac{M}{k-1}.$$

3.7.2 Proof of Lemma 3.6.8

For the rotation matrix $R = (1/\sqrt{d})HD$, each entry of $RY(j)$ of the rotated matrix has the same distribution as $(1/\sqrt{d})V^T Y$, where $V = [V(1), ..., V(d)]^T$ has independent Rademacher entries. We will use this observation to bound the moment generating function of $RY(i)$ conditioned on Y. Towards that end, we have

$$\begin{aligned}
\mathbb{E}\left[e^{\lambda RY(i)} \mid Y\right] &= \prod_{i=1}^{d} \mathbb{E}\left[e^{\lambda V(i)Y(i)/\sqrt{d}} \mid Y\right] \\
&= \prod_{i=1}^{d} \frac{e^{\lambda Y(i)/\sqrt{d}} + e^{-\lambda Y(i)/\sqrt{d}}}{2} \\
&\leq \prod_{i=1}^{d} e^{\lambda^2 Y(i)^2/2d} \\
&= e^{\lambda^2 \|Y\|_2^2/2d},
\end{aligned}$$

where the first identity follows from independence of $V(i)$s and the first inequality follows by the fact that $(e^x + e^{-x})/2$ is less than $e^{x^2/2}$, which in turn can be seen from the Taylor series expansion of these terms. Thus, we have proved the following:

$$\mathbb{E}\left[e^{\lambda RY(i)} \mid Y\right] \leq e^{\lambda^2 \|Y\|_2^2/2d}, \quad \forall \lambda \in \mathbb{R}, \forall i \in [d]. \tag{3.35}$$

Note that $\|Y\|_2^2$ can be further bounded by B^2, which along with (3.35) leads to

$$\mathbb{E}\left[e^{\lambda RY(i)}\right] \leq e^{\lambda^2 B^2/2d} \quad \forall \lambda \in \mathbb{R}, \forall i \in [d].$$

Using this inequality and the observation that $\mathbb{E}[RY(i)] = 0$, we note that $RY(i)$ is a centered subgaussian with a variance parameter B^2/d. The second statement of the lemma trivially follows from Lemma 3.6.6. □.

3.7.3 Proof of Lemma 3.6.11

For any $y \in [0,1]$ such that $ab^{y/(2+y)} < c$, we have $ab^{y/(2+y)} < \min\left\{c, ab^{1/3}\right\}$. By multiplying by a/b on both sides and taking square root, we get

$$\frac{a}{b^{\frac{1}{2+y}}} < \min\left\{\sqrt{\frac{ca}{b}}, \frac{a}{b^{1/3}}\right\},$$

which gives

$$\sup_{y\in[0,1]:a(b)^{y/(2+y)}<c.} \frac{a}{b^{\frac{1}{2+y}}} \leq \min\left\{\sqrt{\frac{ca}{b}}, \frac{a}{b^{1/3}}\right\}.$$

Making cases $ab^{1/3} \geq c$ and $ab^{1/3} < c$, we note that the supremum on the left-side equals the right-side in both the cases. □

3.8 Concluding Remarks

In this chapter, we developed quantizers for communication-constrained optimization over Euclidean spaces. The problem here essentially reduces to minimizing the worst-case ℓ_2 norm, $\alpha_2(Q)$ or $\alpha_2^{\mathrm{m}}(Q)$, of the quantized gradient under the constraint that worst-case ℓ_2 bias between the quantized gradient and input gradient, $\beta_2(Q)$ or $\beta_2^{\mathrm{m}}(Q)$, is small and the output of the quantizer can be represented in r bits. In fact, in the next chapter, we will see that for designing gradient quantizers for communication-constrained optimization over ℓ_p spaces, we need to solve a similar problem where the ℓ_q norms are considered instead, where q is the Hölder conjugate of p. Since the only knowledge we have of the input gradient to be quantized is either an almost sure or mean square-bound on the ℓ_2 norm of the gradient, we first preprocess the gradient to have some handle over the distribution of the gradient coordinates. We do this by randomly rotating the gradient input since it divides the overall gradient value equally across coordinates. We believe that

without any more information on gradient distribution, this is a reasonable preprocessing. We will resort to this idea of random rotation once again in Chapter 5. Another idea we will pick up from the quantizer design in this chapter is that of adaptive quantization. As we saw in our achievability proofs, adaptive gradient quantization became crucial to come up with tight upper bounds. We will see in Chapter 6 that this idea can also be used for classic information theory problems such as the Gaussian rate-distortion problem.

Chapter 4

Communication-Constrained Optimization over ℓ_p Spaces

4.1 Synopsis

In this chapter, we study communication-constrained optimization for ℓ_p lipschitz and convex function family. For this class of functions, we characterize the minimum precision to which the oracle output must be quantized to retain the unrestricted convergence rates? We characterize this precision for every $p \geq 1$ by accessing the information theoretic lower bounds derived in Chapter 2 and by providing quantizers that (almost) achieve these lower bounds. Our quantizers are new and easy to implement. In particular, our results are exact for $p = 2$ and $p = \infty$, showing the minimum precision needed in these settings are $\Theta(d)$ and $\Theta(\log d)$, respectively. The latter result is surprising since recovering the gradient vector will require $\Omega(d)$ bits.

The results presented in this Chapter are from [67].

4.2 Introduction

In this chapter, we develop new algorithms to match the lower-bounds for communication-constrained optimization over ℓ_p spaces. Specifically, we study communication-constrained

optimization for convex and ℓ_p lipschitz function family. We study this problem in the *high-precision regime* introduced in Chapter 3. That is, we ask what is the minimum precision to which the subgradient estimates' supplied by the oracle must be quantized so that we can attain the convergence of the classic, unrestricted setting. We derive a lower bound on this precision using the lower bounds on optimization error derived in Chapter 2. As our main contribution, we propose simple, efficient subgradient quantization algorithms which along with appropriate mirror decent algorithms match these lower bounds.

4.2.1 Main Contributions

We show that for $p \in [1, 2]$ and $p \geq 2$, respectively, roughly d and $d^{2/p} \log(d^{1-2/p} + 1)$ bits are necessary and sufficient for retaining the standard convergence rates. These bounds are tight upto an $O(\log d)$ factor, in general, but are exact for $p = 2$ and $p = \infty$. Prior work has only considered the problem for the Euclidean case, and not for general ℓ_p geometry. Note that in the previous Chapter, we show that RATQ along with PSGD requires a precision of $O(d \log\log\log \ln^* d)$ per iteration to attain the classic convergence rate. In this chapter we get rid of the nagging $\log\log\log \ln^* d$ factor and establish tight bounds.

We use different quantizers for $p \geq 2$ and $p \in [1, 2]$. In the $p \geq 2$ range, we use a quantizer we call SimQ$^+$. SimQ$^+$, in turn, uses multiple repetitions of another quantizer we call SimQ which expresses a vector as a convex combination of corner points of an ℓ_1 ball. It is SimQ that yields an $O(\log d)$ bit quantizer for optimization over ℓ_∞. Also, SimQ$^+$ yields the exact upper bound in the ℓ_2 case. In the $[1, 2]$ range, we divide the vector into two parts with small and large coordinates. We use a uniform quantizer for the first part and RATQ of Chapter 3 for the second part.

The main observation in our analysis for upper bound is that the role of quantizer in optimization is not to express the gradient with small error. It suffices to have an unbiased estimate with appropriately bounded norms.

4.2.2 Prior Work

A detailed literature review on the quantizer used in communication-constrained optimization is presented in Chapter 3. Most of the literature in this area looks at the Euclidean setting. In the general setting of Information-constrained optimization, convex and ℓ_p Lipschitz family was considered in [29], in a statistical query setup, and [24], in a local differential privacy setup.

Finally, we remark that while our quantizers are related to the ones used in prior works, our main contribution is to show that our specific design choices yield optimal precision. For instance, the quantizers in [33] express the input as a convex combination of a set of points, similar to SimQ. One of the quantizers in [33] uses a similar set of points as that of SimQ with a different scaling. However, the quantizers in [33] are designed keeping in mind other objectives, and they fall short of attaining the optimal precision guarantees of SimQ and SimQ$^+$. SimQ is also closely related Maurey's empirical method (see [76], or [90] for a recent reference), however, the use in gradient quantization is new.

Orgainization

In the next section, we describe the setup and the structure of the schemes we will be employing. In Section 4.4, we describe the main result of the paper – a characterization of the minimum precision required to attain classic convergence rate for all p. In Section 4.5 and 4.5.1, we describe our quantizers used to achieve our upper bounds. Finally, we close with comments on achieving tight upper bounds for the lower bounds in Theorems 2.4.4 and 2.4.4 for all p and for generalizing our results to mean square bounded oracles in Section 4.7.

4.3 Setup and preliminaries

4.3.1 Setup

We consider optimization domains $\mathcal{X}_p$ such that ℓ_p diameter is less than D. That is,

$$\mathcal{X}_p \in \mathbb{X}_p(D) := \{\mathcal{X}' : \sup_{x,y\in\mathcal{X}'} \|x-y\|_p \le D.\} \tag{4.1}$$

For the domain of optimization $\mathcal{X}_p$, we develop subgradient compression schemes for function and oracle families given by $\mathcal{O}_{\mathsf{c},p}$, which are defined in Definition 2.3.3.

We want to study communication-constrained optimization for $\mathcal{O}_{\mathsf{c},p}$ in the high precision regime. Thus, the fundamental quantity of interest in this work is the minimum precision to achieve the optimization accuracy of the classic case, denoted by $r^*(T,p)$. Symbolically,

$$r^*(T,p) := \inf\{r \in \mathbb{N} : \sup_{\mathcal{X}\in\mathbb{X}_p(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathsf{c},p}, T, \mathcal{W}_{\mathtt{com},r}) \le \mathcal{U}(T,p)\}, \tag{4.2}$$

where

$$\begin{aligned} \mathcal{U}(T,p) &:= \frac{4c_1 d^{1/2-1/p} DB}{\sqrt{T}}, \quad \forall p \in (2,\infty], \\ \mathcal{U}(T,p) &:= \frac{4c_1\sqrt{\log d} DB}{\sqrt{T}}, \quad \forall p \in [1,2), \end{aligned} \tag{4.3}$$

and $\sup_{\mathcal{X}\in\mathbb{X}_p(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathsf{c},p}, T, \mathcal{W}_{\mathtt{com},r})$ is as defined in Chapter 2. Recall that $\mathcal{U}(T,p)$ denotes the classic convergence rate for the family $\mathcal{O}_{\mathsf{c},p}$ given in Theorem 2.3.5, where the oracle output is available as it is to the optimization algorithm, without any quantization.

4.3.2 Quantizer performance for finite precision optimization

As described in Section 3.3.2, we restrict to memoryless quantization schemes, where the same quantizer will be used for each new subgradient vector, without any information about the previous updates. Also recall from Section 3.3.2, our nonadaptive channel selection strategy is simply denote by quantizer Q. Further, the optimization error for a function f and oracle O when employing a first order optimization π and quantizer Q is

given by

$$\mathcal{E}(f, O, \pi, Q) = \mathbb{E}\left[f(x_T)\right] - \mathbb{E}\left[f(x^*)\right].$$

We now define $\alpha_p(Q)$, which generalizes the definition of $\alpha_2(Q)$ in Chapter 3, to charactize the performance of a quantizer Q for optimization of convex and ℓ_p lipschitz funciton class. Since we restrict to unbiased quantizers, we don't need to define β.

$$\alpha_p(Q) := \sup_{Y \in \mathbb{R}^d : \|Y\|_q^2 \leq B^2 \text{ a.s.}} \sqrt{\mathbb{E}\left[\|Q(Y)\|_2^2\right]}, \quad p \in (2, \infty],$$

$$\alpha_p(Q) := \sup_{Y \in \mathbb{R}^d : \|Y\|_q^2 \leq B^2 \text{ a.s.}} \sqrt{\mathbb{E}\left[\|Q(Y)\|_q^2\right]}, \quad p \in [1, 2].$$

Note that for all $p \geq 1$, the composed oracle QO satisfies assumption (2.5). We employ the stochastic mirror descent (SMD) algorithm with mirror maps given by Remarks 1 and 2 to use the output from the composed oracle. The algorithm's description is given in Algorithm 4.1. Recall that for a mirror map Φ, the Bregman divergence associated with Φ is defined as

$$D_\Phi(x, y) \colon = \Phi(x) - \Phi(y) - \langle \nabla \Phi(y), x - y \rangle.$$

Require: $x_0 \in \mathcal{X}, \eta \in \mathbb{R}^+$, T and access to composed oracle QO

1: **for** $t = 0$ to $T - 1$ **do**

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} (\eta_t \langle x, Q(\hat{g}(x_t)) \rangle) + D_{\Phi_a}(x, x_t))$$

2: **Output:** $\frac{1}{T} \cdot \sum_{t=1}^{T} x_t$

Algorithm 4.1: Quantized SMD with quantizer Q

Moreover, in view of Remarks 1 and 2 , we have the following convergence guarantees for first-order stochastic optimization using gradients quantized by Q.

Theorem 4.3.1. *Consider a quantizer Q for the gradients. Then the algorithm 4.1 with mirror maps as in Remarks 1 and 2 and an unbiased quantizer Q performs as follows.*

1. *For* $2 \geq p \geq 1$,
$$\frac{c_1\sqrt{\log d}D\alpha_p(Q)}{\sqrt{T}} \geq \sup_{(f,O)\in\mathcal{O}_{c,p}} \mathcal{E}(f,O,\pi,Q);$$

2. *For* $p > 2$,
$$\frac{c_1 d^{1/2-1/p}D\alpha_p(Q)}{\sqrt{T}} \geq \sup_{(f,O)\in\mathcal{O}_{c,p}} \mathcal{E}(f,O,\pi,Q).$$

Proof. The proof straightaway follows from Theorem 2.3.5 and Remarks 1 and 2. For completeness, we provide the details below.

First statement simply follows by noting that the bounds in Theorem 3.3.2 hold when instead of $\|\hat{g}(x)\|_q \leq B$, we have $\mathbb{E}\left[\|\hat{g}(x)\|_q^2\right] \leq B^2$ and then using definition of α_p. The second statement simply follows by noting that the bounds in Theorem 3.3.2 are obtained by employing PSGD. Thus it suffices to only have a bound on $\mathbb{E}\left[\|\hat{g}(x)\|_2^2\right]$ and then using the definition of α_p.

□

An interesting insight offered by the result above, which is perhaps simple in hindsight, is that even when dealing with ℓ_p oracles for $p > 2$, we only need to be concerned about the expected ℓ_2 norm of the quantizers output. This follows from the fact that PSGD is the optimal optimization algorithm for $p > 2$ and it's convergence rate is only concerned with the ℓ_2 norm of the quantizers output. It is this insight that leads to the realization that SimQ$^+$ is optimal for these settings.

In the rest of the Chapter, we design unbiased, fixed length quantizers which have $\alpha_p(\cdot)$ of the same order as B. Then, using Theorem 4.3.1 the quantized updates give the same convergence guarantees as that of the classical case, which leads to upper bounds for $r^*(T,p)$. Further, we using Theorems 2.4.4 and 2.4.5, we derive lower bounds for $r^*(T,p)$ to prove optimality of our quantizers.

4.4 Main Result: Characterization of $r^*(T,p)$

The main result of this Chapter is the almost complete characterization of $r^*(T,p)$. We divide the result into cases $p \in [1,2]$ and $p \geq 2$; as mentioned earlier, we use different

quantizers for these two cases.

Theorem 4.4.1. *For stochastic optimization using T accesses to a first-order oracle, the following bounds for $r^*(T,p)$ hold.*

1. *For $p > 2$, we have*

$$d^{2/p}\log\left(2e\cdot d^{1-2/p}+2e\right) \geq r^*(T,p) \geq \left(\frac{c_0}{4c_1}\cdot d^{1/p}\right)^2 \vee 2\log\left(\frac{c_0}{4c_1}\cdot d^{1/2}\right).$$

2. *For $2 \geq p \geq 1$, we have*

$$d\left(\left\lceil \log(2\sqrt{2}{\Delta_1}^{1/q}+2)\right\rceil + 3\right) + \Delta_2 \geq \ r^*(T,p) \geq \left(\frac{c_0}{4c_1\sqrt{\log d}}\right)^2 \cdot d,$$

where $\Delta_1 = \left\lceil \log\left(2+\sqrt{18+6\ln\Delta_2}\cdot d^{1/2-1/q}\right)\right\rceil$ *and* $\Delta_2 = \lceil \log(1+\ln^*(d/3))\rceil$.

Note that for $p > 2$ the upper bounds and lower bounds for $r^*(T,p)$ are off by nominal factor of $\log(d^{1-2/p}+1)$. Also, for $p \in [1,2]$ the bounds are roughly off by $O(\log d \cdot \log(\log d^{1/2-1/q})^{1/q})$ (ignoring the $\log^* d$ terms).

We present the quantizers achieving these upper bounds, and the proof of the upper bounds, in the next two sections. For $p > 2$, we use a quantizer SimQ and its extension SimQ$^+$, presented in Section 4.5. For $p \in [1,2]$, we use a combination of uniform quantization and the quantizer RATQ from previous chapter, presented in Section 4.6. The lower bounds on $r^*(T,p)$ follow immediately by the lower bounds in Theorem 2.4.5 and 2.4.4.

We highlight the most interesting features of the result above in separate remarks below.

Remark 24 ($r^*(T,p)$ is independent of T). Theorem 4.4.1 shows that $r^*(T,p)$ is a function only of p and d, and is independent of T. The number of queries T is a proxy for the desired optimization accuracy. Therefore, the fact that $r^*(T,p)$ is independent of such a parameter is interesting. We note, however, that for oracle models with milder assumptions, such as mean square bounded oracles, this may not hold. In fact, the results of previous chapter suggest that for mean square bounded oracles $r^*(T,2)$ is dependent on T.

Remark 25 (Optimality for $p = \infty$). Our bounds match for $p = \infty$, namely our quantizer SimQ offers optimal convergence rate with gradient updates at the least precision. A surprising observation is that this precision is merely $O(\log d)$, much smaller than $O(d)$ bits needed to recover the gradient vector under any reasonable loss function.

Remark 26 (Optimality for $p = 2$). The high-precision regime for $p = 2$ was already considered in the previous chapter. Both [9] and [88] give variable-length quantization schemes to exactly achieve the lower bound on $r^*(T, p)$, but the worst-case precision can be order-wise greater than d. The quanitzer RATQ from the previous Chapter was within a small factor of $O(\log \log \log \ln^* d)$ of this lower bound. In this chapter, we remove this nagging factor using a different fixed-length quantizer SimQ$^+$.

Remark 27 (Fixed precision). The quantizer RATQ remains optimal upto a factor of $O(\sqrt{\log \ln^* d})$ for the more general problem of characterizing $\sup_{\mathcal{X} \in \mathbb{X}_p(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{c},p}, T, \mathcal{W}_{\mathtt{com},r})$ for any precision r less than d bits. In this setting of small precision, the performance of SimQ$^+$ is much worse.

4.5 Our quantizers for $p > 2$

We present our quantizer SimQ and its extension SimQ$^+$. The former is seen to be optimal for $p = \infty$ while the latter for $p = 2$.

4.5.1 An optimal quantizer for $p = \infty$

Simplex Quantizer (SimQ) Our first quantizer SimQ is described in Algorithms 4.2 and 4.3. For $p = \infty$, our quantizer's input vector Y is an unbiased estimate of the subgradient of the function at the point queried and satisfies $\|Y\|_1 \leq B$. SimQ takes such a Y as an input and produces an output vector which, too, satisfies both these properties. The main idea behind SimQ is the fact that any point inside the unit ℓ_1 ball can be represented as a convex combination of at the most $2d$ points: $\{e_i, -e_i : i \in [d]\}$. With this observation, we can create an unbiased estimate of the input vector using only these

Require: Input $Y \in \mathbb{R}^d$, Parameter B

$$i^* = \begin{cases} i & w.p. \quad |Y(i)|/B \\ 0 & w.p. \quad 1 - \|Y\|_1/B \end{cases}$$

if $i^* \in [d]$ **then**

$$j^* = sign(Y(i^*))$$

else

$$j^* = 1$$

Output: $Q^{\mathtt{e}}_{\mathtt{SimQ}}(Y; B) = i^* \cdot j^*$

Figure 4.2: Encoder $Q^{\mathtt{e}}_{\mathtt{SimQ}}(Y; B)$ for SimQ

Require: Input $i' \in \{-d, -(d-1), \cdots 0, \cdots, d\}$

if $i' \neq 0$ **then**

$$Z = Bsign(i')e_{|i'|}$$

else

$$Z = 0$$

Output: $Q^{\mathtt{d}}_{\mathtt{SimQ}}(i'; B) = Z$

Algorithm 4.3: Decoder $Q^{\mathtt{d}}_{\mathtt{SimQ}}(i'; B)$ for SimQ

$2d$ corner points along with the zero vector. Since all of these $2d+1$ points have a ℓ_2 norm of at the most B, the output vector, too, has a ℓ_2 norm of at the most B.

Theorem 4.5.1. *Let Q be the quantizer SimQ described in Algorithms 4.2, 4.3. Then, for Y such that $\|Y\|_1 \leq B$ a.s., $Q(Y)$ can be represented in $\log(2d+1)$ bits, $\mathbb{E}[Q(Y)|Y] = Y$, and $\alpha_\infty(Q) \leq B$.*

Proof. Since $i^* \in [d]$ and $j^* \in \{-1, 1\}$, we can represent the output of the encoder of SimQ using $\log(2d+1)$ bits. Next, denoting the quantizer SimQ by Q, note that

$$\mathbb{E}[Q(Y)|Y] = \sum_{i=1}^{d} B \cdot sign(Y(i)) \cdot e_i \cdot \frac{|Y(i)|}{B} = Y,$$

namely SimQ is unbiased. To complete the proof, note that $\|Q(Y)\|_2^2 \leq B^2$ *a.s.*. □

Theorem 4.5.1 along with Theorem 4.3.1 establishes Theorem 4.4.1 for $p = \infty$.

4.5.2 Our Quantizer for $p \in [2, \infty)$

For this case, we need to quantize inputs that are bounded in ℓ_q norm with $q \in (1, 2]$ so that the quantized output is unbiased and has small expected ℓ_2 norm square; we will use SimQ$^+$ to do this.

SimQ$^+$ The quantizer SimQ$^+$ outputs the average of k independent repetitions of the SimQ quantizer for a given input vector. The input vectors Y satisfy $\|Y\|_1 \leq Bd^{1/p}$. Therefore, we use SimQ with parameter $Bd^{1/p}$ instead of B. The repetitions help reduce the error to compensate for the extra loss factor. Specifically, the output of SimQ$^+$ denoted by $Q(Y)$ is given by

$$Q(Y) = \frac{1}{k} \cdot \sum_{i=1}^{k} Q^i_{\texttt{SimQ}}(Y; Bd^{1/p}), \tag{4.4}$$

where $Q^i_{\texttt{SimQ}}$ are independent iterations of SimQ.

The next component of SimQ$^+$ is how the encoder of SimQ$^+$ expresses the output of these k copies of SimQ to attain compression. If represented naively, this will require $O(d^{2/p} \log d)$. But we can do much better since we only need the average value of these entries. For that, we can simply report the *type* of this vector – the frequency of each index in the k length sequence. The signs of the input coordinates for the non-zero entries can be sent separately.

Note that there are $d + 1$ indices overall, as SimQ can pick any index from $\{0, \dots d\}$. Therefore, the total number of types is $\binom{d+k}{k}$, which can at the most be $(\frac{ed+ek}{k})^k$ bits. Hence, the precision needed to represent the type is at the most $k \log e + k \log(\frac{d}{k} + 1)$.

The type of the input can be used to determine a set $\mathcal{I}_0$ of non-zero indices that appear at least once. There are at most k such entries. Therefore, we can use a binary vector of length k to store the signs for these entries. We use this representation in SimQ$^+$, with the indices in $\mathcal{I}_0$ represented in the vector in increasing order.

Theorem 4.5.2. *For a $p \in [2, \infty)$, let Q be the quantizer SimQ$^+$ described in (4.4). Then, for Y such that $\|Y\|_q \le B$ a.s., $Q(Y)$ can be represented in $k \log e + k \log(\frac{d}{k} + 1) + k$ bits, $\mathbb{E}[Q(Y)|Y] = Y$, and $\alpha_p(Q) \le \sqrt{\frac{B^2 d^{2/p}}{k} + B^2}$.*

Proof. We already saw how to represent the output of the k copies of SimQ using $k \log e + k \log(\frac{d}{k} + 1) + k$ bits. For bounding $\alpha_p(Q)$, note from (4.4) that SimQ$^+$ is an unbiased quantizer since SimQ is unbiased. Further, denoting by $Q_i(Y)$ the output $Q^i_{\texttt{SimQ}}(Y; Bd^{1/p})$, we get

$$\begin{aligned}
\mathbb{E}\left[\|Q(Y)\|_2^2\right] &= \mathbb{E}\left[\|Q(Y) - Y\|_2^2\right] + \mathbb{E}\left[\|Y\|_2^2\right] \\
&= \frac{1}{k^2} \sum_{i=1}^{k} \mathbb{E}\left[\mathbb{E}\left[\|Q_i(Y) - Y\|_2^2 | Y\right]\right] + \mathbb{E}\left[\|Y\|_2^2\right] \\
&= \frac{\mathbb{E}\left[\|Q_1(Y) - Y\|_2^2\right]}{k} + \mathbb{E}\left[\|Y\|_2^2\right] \\
&\le \frac{d^{2/p} B^2}{k} + B^2,
\end{aligned}$$

where the first identity uses the fact that $Q(Y)$ is an unbiased estimate of Y; the second uses the fact that $Q_i(Y) - Y$ are zero-mean, independent random variables when conditioned on Y; the third uses the fact that $Q_i(Y) - Y$ are identically distributed; and the final inequality is by the performance of SimQ. □

The proof of upper bound for $p \in [2, \infty)$ in Theorem 4.4.1 is completed by setting $k = d^{2/p}$ and using Theorems 4.5.2 and 4.3.1.

4.6 Our Quantizers for $p \in [1, 2]$

For p in $[1, 2]$, the oracle yields unbiased subgradient estimates Y such that $\|Y\|_q \le B$ almost surely. Our goal is to quantize such Ys in an unbiased manner and ensure that $\mathbb{E}\left[\|Q(Y)\|_q^2\right]$ is $O(B^2)$. It can be seen that a simple unbiased uniform quantizer will achieve this using $d(\log d^{1/q} + 1)$. However, our goal here is to get a result that is stronger than this baseline performance. To that end, we split the input vector Y in two parts Y_1 and Y_2 with the first part having ℓ_∞ norm less than c and the second part having less than

d/Δ_1 nonzero coordinates. We use an "ℓ_∞ ball quantizer" (a uniform quantizer) for Y_1 and an "ℓ_2 ball quantizer" for Y_2.

Specifically, set $c := \dfrac{B\Delta_1^{1/q}}{d^{1/q}}$, where Δ_1 is that in Theorem 4.4.1. Then, define

$$Y_1 := \sum_{i=1}^{d} Y(i)\mathbb{1}_{\{|Y(i)|\le c\}}e_i,\ \ Y_2 := \sum_{i=1}^{d} Y(i)\mathbb{1}_{\{|Y(i)|>c\}}e_i. \tag{4.5}$$

Clearly, $\|Y_1\|_\infty \le c$. Further, since $\|Y\|_q \le B$, the number of nonzero coordinates in Y_2 can be at the most $B^q/c^q = d/\Delta_1$. For quantizing Y_1, we use the coordinate-wise uniform quantizer (CUQ) described in Section 3.4.1. In order to quantize Y_1 in (4.5), we set the parameters of CUQ to

$$M = c, \quad \log(k+1) = \left\lceil \log(2\sqrt{2}\Delta_1^{1/q} + 2) \right\rceil. \tag{4.6}$$

Lemma 4.6.1. *Let Q_{u} be the quantizer CUQ with parameters M and k set as in (4.6). Then, for Y such that $\|Y\|_q \le B$ a.s. and Y_1 as that in (4.5), $Q_{\mathrm{u}}(Y_1)$ can be represented in $d\left\lceil \log(2\sqrt{2}\Delta_1^{1/q} + 2) \right\rceil$ bits, $\mathbb{E}\left[Q_{\mathrm{u}}(Y_1) \mid |Y|\right] = Y_1$, and $\mathbb{E}\left[\|Q_{\mathrm{u}}(Y_1)\|_q^2\right] \le 3B^2$.*

Proof. CUQ requires a precision of $d\log(k+1)$, which coincides with the statement above for our choice of k. To see unbiasedness, note that CUQ is an unbiased quantizer as long as all the coordinates of the input do not exceed M. Since we have set $M = c$ and $\|Y_1\|_\infty = c$, this property holds. Finally, to show that $\mathbb{E}\left[\|Q_{\mathrm{u}}(Y_1)\|_q^2\right] \le 3B^2$, note that $\mathbb{E}\left[\|Q_{\mathrm{u}}(Y_1)\|_q^2\right] \le 2\mathbb{E}\left[\|Q_{\mathrm{u}}(Y_1) - Y_1\|_q^2\right] + 2\mathbb{E}\left[\|Y_1\|_q^2\right]$. Also, $\mathbb{E}\left[\|Q_{\mathrm{u}}(Y_1) - Y_1\|_q^2\right] \le B^2$, where we use the fact that for M set as in (4.6) we have that $|Q_{\mathrm{u}}(Y_1)(i) - Y_1(i)| \le \frac{2M}{(k-1)}$ a.s., $\forall i \in [d]$, by the description of CUQ. □

In order to quantize Y_2, we indicate the coordinates with non-zero entries. This takes less than d bits. Then, we quantize the restriction Y_2' of Y_2 to these nonzero entries. Recall that the dimension of Y_2' is less than $d' := d/\Delta_1$. Also, the ℓ_2 norm of Y_2' is less than $\|Y\|_2 \le \|Y\|_q d^{1/2-1/q} \le Bd^{1/2-1/q} =: B'$.

We need a quantizer Q such that $\mathbb{E}\left[\|Q(Y_2')\|_q^2\right]$ is $O(B^2)$. As seen in the proof of Lemma 4.6.1, one way to do this is to ensure $\mathbb{E}\left[\|Q(Y_2') - Y'\|_q^2\right]$ is $O(B^2)$, which, in turn,

can be ensured if $\mathbb{E}\left[\|Q(Y_2') - Y_2'\|_2^2\right]$ is $O(B^2)$. To achieve this, we can use an unbiased quantizer for the unit ℓ_2 ball in $\mathbb{R}^d$, which can quantize it to an MSE of $O(1/d^{1-2/q})$ using $O(d\log(d^{1/2-1/q})$ bits. We note that SimQ$^+$, while optimal for the stochastic optimization use-case, does not yield the required scaling of bits in MSE. A natural candidate quantizer is RATQ, which is, in fact, close to information theoretically optimal. We set the parameters of RATQ in terms of B' and d'. We set

$$\begin{aligned} &m = \frac{3B'^2}{d'}, \quad m_0 = \frac{2B'^2}{d'} \cdot \ln s, \quad \log h = \left\lceil \log(1 + \ln^*(d'/3)) \right\rceil, \\ &s = \log h, \quad \log(k+1) = \Delta_1. \end{aligned} \tag{4.7}$$

Lemma 4.6.2. *Let $Q_{\mathtt{at},R}$ be the quantizer RATQ with parameters set as (4.7). Then, for Y such that $\|Y\|_q \le B$ a.s. and Y_2' the restriction of Y_2 in (4.5), $Q_{\mathtt{at},R}(Y_2')$ can be represented in $2d + \Delta_2$ bits, $\mathbb{E}\left[Q_{\mathtt{at},R}(Y_2') \mid |Y] = Y_2'\right.$, and $\mathbb{E}\left[\|Q_{\mathtt{at},R}(Y_2')\|_q^2\right] \le 3B^2$.*

Proof. First, we note that the output of RATQ can be represented in $\lceil d'/s \rceil (\log h) + d\log(k+1)$ bits, which, in this case, is less than

$$\frac{d}{\Delta_1 \log h} \cdot (\log h) + \log h + \left(\frac{d}{\Delta_1}\log(k+1)\right) \le 2d + \Delta_2.$$

For unbiasedness, note that for our choice of m, m_0, h, RATQ is always an unbiased quantizer of the input. Finally, for showing $\mathbb{E}\left[\|Q_{\mathtt{at},R}(Y_2')\|_q^2\right] \le 3B^2$, we note that

$$\begin{aligned} \mathbb{E}\left[\|Q_{\mathtt{at},R}(Y_2')\|_q^2\right] &\le 2\mathbb{E}\left[\|Q_{\mathtt{at},R}(Y_2') - Y_2'\|_q^2\right] + 2\mathbb{E}\left[\|Y_2'\|_q^2\right] \\ &\le 2\mathbb{E}\left[\|Q_{\mathtt{at},R}(Y_2') - Y_2'\|_q^2\right] + 2B^2 \\ &\le 2\mathbb{E}\left[\|Q_{\mathtt{at},R}(Y_2') - Y_2'\|_2^2\right] + 2B^2. \end{aligned}$$

The proof will be complete upon showing that $\mathbb{E}\left[\|Q_{\mathtt{at},R}(Y_2') - Y_2'\|_2^2\right] \le B^2/2$, towards which we apply Lemma 3.6.9 to get

$$\mathbb{E}\left[\|Q_{\mathtt{at},R}(Y_2') - Y_2'\|_2^2\right] \le B^2 d^{1-2/q} \cdot \frac{9 + 3\ln s}{(k-1)^2},$$

and substituting our choice of k. □

The overall quantizer Q of input vector Y is the sum of quantized outputs of Y_1 and Y_2. By Lemmas 4.6.1 and 4.6.2, the quantized output of Y can be represented in $d\left(\left\lceil \log(2\sqrt{2}{\Delta_1}^{1/q}+2)\right\rceil+3\right)+\Delta_2$ bits[1]. Furthermore, $\alpha_p(Q)\le\sqrt{12}B$. These facts along with Theorem 4.3.1 prove the upper bound in Theorem 4.4.1 for $p\in[1,2]$.

4.7 Characterization of general tradeoff and mean square bounded oracles

We close with the remark that an almost complete characterization of optimization error $\mathcal{E}^*(\mathcal{X},\mathcal{O}_{\mathsf{c},p},T,\mathcal{W}_{\mathtt{com},r})$ for any r,p (namely, the low-precision regime) can be obtained using our quantizers and the ideas developed in this Chapter. We describe in detail algorithms to achieve these bounds below.

4.7.1 Upper Bounds on $\mathcal{E}^*(\mathcal{X},\mathcal{O}_{\mathsf{c},1},T,\mathcal{W}_{\mathtt{com},r})$ for $p\in(2,\infty]$

For upper bounds when $p\in[2,\infty)$, note that the parameter k of SimQ$^+$ gives us a nice lever to operate under any precision constraint $r\ge\log d$. It turns out that such a quantizer along with PSGD leads to upper bounds which are off by at the most $\sqrt{\log d}$ factor from the lower bounds in Theorem 2.4.5.

4.7.2 Upper Bounds on $\mathcal{E}^*(\mathcal{X},\mathcal{O}_{\mathsf{c},1},T,\mathcal{W}_{\mathtt{com},r})$

We now describe a new scheme to match the lower bound for $p=1$. Our scheme divides the entire horizon of T iterations into Tr/d different phases. For any phase $t\in[Tr/d]$, the same point x_t in the domain is queried d/r times. For each of the d/r queries in a phase, we use r-bit quantizers to quantize different coordinates of the subgradient output. At a high level, we want to use these r bits to send 1 bit each for r different coordinates, sending 1 bit for each coordinate across the phases. However, there is one technical difficulty. We

[1]This accounts for the communication needed to send the nonzero indices of Y_2, too.

have not assumed that making queries for the same point gives identically distributed random variables. We circumvent this difficulty using random permutations to create unbiased estimates for the subgradients.

Specifically, for a permutation $\sigma\colon [d] \to [d]$ chosen uniformly at random using public randomness, we select the coordinates $\sigma(1 + (i-1)\cdot r)$ to $\sigma(i \cdot r)$ of the subgradient estimate $\hat{g}_i$ supplied by the oracle for the ith query in the tth phase (i.e., ith time we query the point x_t) and quantize all of these coordinates using an 1-bit unbiased quantizer for the interval $[-B, B]$. Note that such a quantizer can be formed since $\|\hat{g}_i\|_\infty \le B$.

Using this procedure, the quantized gradient for every query in each phase can be stored in r bits. Furthermore, using all the d/r quantized estimates received in a phase, we can create an estimate of the subgradient by simply adding all the estimates. Denote by $\bar{Q}_t$ our subgradient estimate in the tth phase. Then,

$$\bar{Q}_t = \sum_{i=1}^{d} Q_{\pi(i)}(\hat{g}_i) e_{\sigma(i)},$$

where $\hat{g}_i$ is the subgradient estimate returned by the oracle when we query x_t for the ith time and Q_i is a 1-bit unbiased estimator of the ith coordinate of gradient estimate given below: For all vectors g, such that $\|g\|_\infty \le B$, we have

$$Q_i(g) = \begin{cases} B & \text{w.p.} \quad \frac{g(i)+B}{2B} \\ -B & \text{w.p.} \quad \frac{B-g(i)}{2B} \end{cases}.$$

Then, we use $\bar{Q}_t$ to update x_t to x_{t+1} using stochastic mirror descent with mirror map

$$\phi_a(x) \colon = \frac{\|x\|_a^2}{a-1},$$

where $a = \frac{2\log d}{2\log d - 1}$.

Theorem 4.7.1. *For $r \in \mathbb{N}$, we have*

$$\sup_{\mathcal{X} \in \mathbb{X}_1(D)} \mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{c},1}, T, \mathcal{W}_{\mathtt{com},r}) \le \frac{c_0 DB\sqrt{\log d}}{\sqrt{T}} \cdot \sqrt{\frac{d}{d \wedge r}}$$

1: **for** $t \in [Tr/d]$ **do**

2: **for** $i \in [d/r]$ **do**

3: At Center: Query the oracle for x_t

4: At Oracle: Output the r-bit vector of 1-bit unbiased estimates of the r coordinates $\{1 + (i-1) \cdot r, \ldots, i \cdot r\}$ of $\hat{g}_i(x_t)$ given by

$$\bar{Q}_t = \sum_{j=1+(i-1)\cdot r}^{i \cdot r} Q_{\pi(j)}(\hat{g}_i(x_t)) e_{\sigma(j)}$$

5: At Center: $x_{t+1} = \arg\min_{x \in \mathcal{X}}(\eta_t \langle x, \bar{Q}_t \rangle) + D_{\Phi_a}(x, x_t))$

6: **Output:** $\frac{\sum_{i=1}^{T} x_t}{T}$

Figure 4.4: π^* Almost optimal Scheme for Communication constrained optimization for convex and ℓ_1 lipschitz family

for every $D > 0$.

Proof. Note that our first order optimization algorithm π^* uses Tr/d iterations. Moreover, the subgradient estimates $\bar{Q}_t$ are unbiased and have their infinity norm bounded by B. Namely, we have obtained an unbiased subgradient oracle which produces estimates with infinity norm bounded by B. Thus, using the standard analysis of mirror descent using noisy subgradient oracle for optimization over an ℓ_1 ball with mirror map $\phi_a(x) \colon = \frac{\|x\|_a^2}{a-1}$ (see Theorem 4.3.1), the proof is complete. □

4.7.3 Upper Bounds on $\mathcal{E}^*(\mathcal{X}, \mathcal{O}_{\mathtt{c},p}, T, \mathcal{W}_{\mathtt{com},r})$ for $p \in (1, 2)$.

For $p \in (1, 2)$, the scheme described in Algorithm 4.4 can still be used. However, the upper bounds would be off by a factor of $d^{1/q}$. This factor increases with increase in p and and as we get closer to 2, employing optimal quantizers for the Euclidean setting is a better option. For instance, employing RATQ along with the appropriate mirror descent algorithm would lead to optimization error off by $d^{1/2-q}$ from the lower bound. Thus,

inteprolationg between the schemes for $p = 1$ and $p = 2$, we match the lower bound in Theorem 2.4.4 upto a factor of $\min d^{1/q}, d^{1/2-q}$ for any precision r and $p \in (1, 2)$.

Finally, we believe that removing these remaining factors can lead to new quantizers, and is of research interest.

4.7.4 Mean square bounded oracles

For mean square bounded oracles mentioned in Chapter 3, the bias in the quantized oracle output is nearly inevitable. In our previous chapter, we proposed appropriate *gain-shape* quantizers for quantizing the oracle output in the Euclidean setup, which resulted in lesser bias over standard quantizers. This idea is valid for the general ℓ_p setup; in particular, we can use a gain quantizer to quantize the ℓ_q norm of the oracle output and a shape quantizer to quantize the oracle output vector normalized by the ℓ_q norm, the shape of the oracle output vector. Note that the shape vector has has ℓ_q norm 1, which allows us to use the quantizers developed in this chapter to quantize the shape. The gain is a scalar random variable which has its second moment bounded by B^2. To quantize such a random variable, we can use AGUQ from previous chapter. Clearly, the lower bounds for almost surely bounded oracles remain valid for mean square bounded oracles as well. Additionally, we can also derive lower bounds for a specific class of quantizers, such as those derived in previous chapter, which help in capturing the reduction in the convergence rate due to mean square bounded noise.

Part II

Efficient Quantization for Federated Learning Primitives

Chapter 5

Communication-Efficient Distributed Mean Estimation

5.1 Synopsis

Communication efficient distributed mean estimation is an important primitive that arises in many distributed learning and optimization scenarios such as federated learning. Without any probabilistic assumptions on the underlying data, we study the problem of distributed mean estimation in two different settings: 1) where the server does not have access to side information and 2) where the server has access to side-information. In the first setting, we use RATQ proposed in Chapter 3 and improve over the state of the art.

In the second setting, we propose *Wyner-Ziv estimators*, which are communication and computationally efficient and near-optimal when an upper bound for the distance between the side information and the data is known. In a different direction, when there is no knowledge assumed about the distance between side information and the data, we present an alternative Wyner-Ziv estimator that uses correlated sampling. This latter setting offers *universal recovery guarantees*, and perhaps will be of interest in practice when the number of users is large and keeping track of the distances between the data and the side information may not be possible.

The results presented in this Chapter are from [69] and [66].

5.2 Introduction

Consider the problem of distributed mean estimation for n vectors $\{x_i\}_{i=1}^n$ in $\mathbb{R}^d$, where x_i is available to client i. Each client communicates to a server using a few bits to enable the server to compute the empirical mean

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{5.1}$$

This estimation problem has become a crucial primitive for distributed optimization scenarios such as federated learning, where the data is distributed across multiple clients. One of the main bottlenecks in such distributed scenarios is the significant communication cost incurred due to client communication at each iteration of the distributed algorithm. This has spurred a recent line of work which seeks to design quantizers to express x_is using a low precision and, yet, enable the server to compute a high accuracy estimate of $\bar{x}$ (see [88], [52], [17], [46], and the references therein).

Most of the recent works on distributed mean estimation focus on the setting where the server must estimate the sample mean based on the client vectors, and nothing else. However, in practice, the server may also have access to some side information. For example, consider the task of training a machine learning model based on remote client data as well as some publicly accessible data. At each iteration, the server communicates its global model to the client, based on which the clients compute their updates (the gradient estimates based on their local data), compress them, and then send them to the server. The server may choose to compute its own update using the publicly available dataset to complement the updates from the client. In a related setting, the server can use the previously received gradients as side information for the next gradients expected from the clients. Similarly, distributed mean estimation with side information can be used for variance reduction in other problems such as power iteration or parallel SGD (*cf.* [20]).

Motivated by these observations, for the distributed mean estimation problem described at the start of the section, we study both the settings of distributed mean estimation:

1. The *no side information* setting, where the server does not have access to any side

information.

2. The *side information* setting, where the server has access to some side information $\{y_i\}_{i=1}^n$ in $\mathbb{R}^d$, in addition to the communication from clients. Here, y_i can be viewed as server's initial estimate (guess) of x_i. We emphasize that the side information y_i is available only to the sever and can, therefore, be used for estimating the mean at the server, but is not available to the clients while quantizing the updates $\{x_i\}_{i=1}^n$.

We close this section with the remark that distributed mean estimation in the no side information setting can be viewed as a special case of distributed mean estimation in the side information setting, where the side-information $\{y_i\}_{i=1}^n$ is set to 0. We will, therefore, describe our model for the side-information setting.

5.2.1 The model

Consider the input $\mathbf{x} := (x_1, \ldots, x_n)$ and the side information $\mathbf{y} := (y_1, \ldots, y_n)$. The clients use a communication protocol to send r bits each about their observed vector to the server. For the ease of implementation, we restrict to non-interactive protocols. Specifically, we allow *simultaneous message passing* (SMP) protocols $\pi = (\pi_1, ..., \pi_n)$ where the communication $C_i = \pi_i(x_i, U) \in \{0,1\}^r$ of client i, $i \in [n]$, can only depend on its local observation x_i and public randomness U. Note that the clients are not aware of side information $\mathbf{y}$, which is available only to the server. In effect, the message C_i is obtained by *quantizing* x_i using an appropriately chosen randomized quantizer. Denoting the overall communication by $C^n := (C_1, C_2, ..., C_n)$, the server uses the transcript (C^n, U) of the protocol and the side information $\mathbf{y}$ to form the estimate of the sample mean[1] $\hat{\bar{x}} = \hat{\bar{x}}(C^n, U, \mathbf{y})$; see Figure 5.1 for a depiction of our setting. We call such a π an *r-bit SMP protocol* with input $(\mathbf{x}, \mathbf{y})$ and output $\hat{\bar{x}}$.

We measure the performance of protocol π for inputs $\mathbf{x}$ and $\mathbf{y}$ and output $\hat{\bar{x}}$ using

[1]While side information y_i is associated with client i, we do not enforce this association in our general formulation at this point.

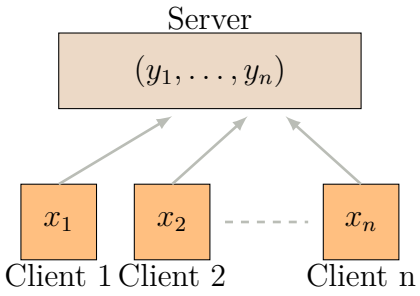


Figure 5.1: Problem setting of mean estimation with side information

mean squared error (MSE) given by

$$\mathcal{E}(\pi, \mathbf{x}, \mathbf{y}) := \mathbb{E}\left[\|\hat{\bar{x}} - \bar{x}\|_2^2\right],$$

where the expectation is over the public randomness U and $\bar{x}$ is given in (5.1). We study the MSE of protocols for $\mathbf{x}$ and $\mathbf{y}$ such that the Euclidean distance between x_i and y_i is at most Δ_i, i.e.,

$$\|x_i - y_i\|_2 \leq \Delta_i, \quad \forall\, i \in [n]. \tag{5.2}$$

Denoting $\mathbf{\Delta} := (\Delta_1, \ldots, \Delta_n)$, we are interested in the performance of our protocols for the following settings:

1. **The *no side information* setting**, where $\Delta_i = 1$ and $y_i = 0$, for all $i \in [n]$. That is, the server does not have access to side information and the input vectors lie in the unit Euclidean ball.

2. **The *side information* setting**, where the server has access to some side-information. We study two different cases for this setting, which are described as follow:

   (a) **The *known* $\mathbf{\Delta}$ setting**, where Δ_i is known to client i and the server;

   (b) **The *unknown* $\mathbf{\Delta}$ setting**, where Δ_is are unknown to everyone.

In all these settings, we seek to find efficient r-bit quantizers for x_i that will allow accurate sample mean estimation. We now point out the difference between the two

different settings of distributed mean estimation in presence of side information. In the known $\mathbf{\Delta}$ setting, the quantizers of different clients can be chosen using the knowledge of $\mathbf{\Delta}$; in the unknown $\mathbf{\Delta}$ setting, they must be fixed irrespective of $\mathbf{\Delta}$.

In another direction, we distinguish the *small-precision* setting of $r \leq d$ from the *large-precision*[2] setting of $r > d$. The former is perhaps of more relevance for federated learning and high-dimensional distributed optimization, while the latter has received a lot of attention in the information theory literature on rate-distortion theory.

As a benchmark, we recall the result for distributed mean estimation with no side-information from [88]. [88] showed that the minmax MSE in the no side-information setting is

$$\Omega\left(\frac{d}{nr}\right). \tag{5.3}$$

Further, [88] derive an upper bound which matches the lower bound upto a factor of $\log\log d$.

5.2.2 Our contributions

In the no side-information setting, we improve over the upper bound of [88] and match the lower bound upto a miniscule $\log\log^* d$ factor by using RATQ from Chapter 3.

In the side-information setting, drawing on ideas from distributed quantization problems in information theory (*cf.* [91]), specifically the Wyner-Ziv problem, we present *Wyner-Ziv estimators*. In the known $\mathbf{\Delta}$ setting, for a fixed $\mathbf{\Delta}$, and the small-precision setting of $r \leq d$, we propose an *r-bit SMP protocol* $\pi^*_{\mathtt{k}}$ which satisfies

$$\mathcal{E}(\pi^*_{\mathtt{k}}, \mathbf{x}, \mathbf{y}) = O\left(\sum_{i=1}^{n} \frac{\Delta_i^2}{n} \cdot \frac{d \log\log n}{nr}\right),$$

for all $\mathbf{x}$ and $\mathbf{y}$ satisfying (5.2). Thus, in the case where all x_is lie in the Euclidean ball of radius 1, we improve upon the optimal estimator for distributed mean estimation in

[2]The definition of large-precision setting is different from the definition of high-precision setting described in the first part of the thesis. Observe that the large precision setting here simply refers to the setting of $r > d$, whereas in the high-precision setting from earlier chapters we looked to characterize the minimum precision required to attain the classical convergence rate.

the no side information setting (5.3) in the regime $\sum_{i=1}^{n} \frac{\Delta_i^2 \log\log n}{n} \leq 1$. Our estimator is motivated by the classic Wyner-Ziv problem, and hence, we refer to it as the *Wyner-Ziv estimator*. The details of the algorithm are given in Section 5.5.3.

Our protocol uses the same (randomized) r-bit quantizer for each client's data and simply uses the sample mean of the quantized vectors as the estimate for $\bar{x}$. Furthermore, the common quantizer used by the clients is efficient and has nearly linear time-complexity of $O(d \log d)$. As was the case for RATQ, our proposed quantizer first applies a random rotation to the input vectors x_i at client i and the side information vector y_i at the server. This ensures that the Δ_i upper bound on the ℓ_2 distance of x_i and y_i is converted to roughly a $\Delta_i/\sqrt{d}$ upper bound on the ℓ_∞ distance between x_i and y_i. This then enables us to use efficient one-dimensional quantizers for each coordinate of the x_i, which can now operate with the knowledge that the server knows a y_i with each coordinate within roughly $\Delta_i/\sqrt{d}$ of x_i's coordinates.

Moreover, we show that this protocol $\pi^*_{\mathtt{k}}$ has optimal (worst-case) MSE up to an $O(\log\log n)$ factor. That is, we show that for any other r-bit SMP protocol π for $r \leq d$, we can find $\mathbf{x}$ and $\mathbf{y}$ satisfying (5.2) such that

$$\mathcal{E}(\pi, \mathbf{x}, \mathbf{y}) = \Omega\left(\min_{i\in\{1,\ldots,n\}} \Delta_i^2 \cdot \frac{d}{nr}\right).$$

In the unknown $\boldsymbol{\Delta}$ setting, we propose a protocol $\pi^*_{\mathtt{u}}$ which adapts to the unknown distance Δ_i between x_i and y_i and, remarkably, provides MSE guarantees dependent on $\boldsymbol{\Delta}$. Specifically, for the small-precision setting of $r \leq d$, the protocol satisfies

$$\mathcal{E}(\pi^*_u, \mathbf{x}, \mathbf{y}) = O\left(\sum_{i=1}^{n} \frac{\Delta_i}{n} \cdot \frac{d \ln^* d}{nr}\right),$$

for all $\mathbf{x}$ and $\mathbf{y}$ in the unit Euclidean ball $\mathcal{B} := \{x \in \mathbb{R}^d : \|x\|_2 \leq 1\}$ and satisfying (5.2). Thus, we improve upon the optimal estimator for the no side information counterpart (5.3) in the regime $\sum_{i=1}^{n} \frac{\Delta_i \ln^* d}{n} \leq 1$. Once again, the quantizer employed by the protocol is efficient and has nearly linear time-complexity of $O(d \log d)$. At the heart of our proposed quantizer is the technique of correlated sampling from [43] which enables to derive a $\boldsymbol{\Delta}$

dependent MSE bound.

Furthermore, both our quantizers can be extended to the large-precision regime of $r > d$. The quantizer for the known $\mathbf{\Delta}$ setting directly extends by using r/d bits per dimension. The MSE of the SMP protocol using this quantizer for all the clients is only a factor of $\log n + r/d$ from the lower bound derived in [20] for the large-precision regime. The quantizer for the unknown $\mathbf{\Delta}$ setting can be extended by sending the "type" of the communication vector, following an idea proposed in Chapter 4 for SimQ$^+$. The MSE of the SMP protocol using this quantizer for all the clients falls as $2^{-r/d \ln^* d}$ as opposed to d/r that can be obtained using naive extensions of our quantizer.

5.2.3 Prior work

The version of the distributed mean estimation problem with no side information at the server has been extensively studied. For any protocol in this setting operating with a precision constraint of $r \leq d$ bits per client, using a strong data processing inequality from [25], [88] shows a lower bound on MSE of $\Omega\left(\frac{d}{nr}\right)$, when all x_is lie in the Euclidean ball of radius one. [88] propose a rotation based uniform quantization scheme which matches this lower bound up to a factor of $\log \log d$ for any precision constraint r.

The known $\mathbf{\Delta}$ setting described above was first considered in [20]. The scheme of [20] relies on lattice quantizers with information theoretically optimal covering radius. Explicit lattices to be used and computationally efficient decoding is not provided.

In contrast, we provide explicit computationally efficient protocols for both small- and large-precision settings. Also, we establish lower bounds showing the optimality of our quantizer upto a multiplicative factor of $\log \log n$ in the small-precision regime of $r \leq d$. In comparison, the scheme of [20] is off by a factor of $\frac{d}{r}$ from this lower bound. Thus, when $r \ll d$, our scheme performs significantly better than that in [20]. We remark that the unknown $\mathbf{\Delta}$ setting, which is perhaps more important in certain applications where estimating the distance of side information of each client is infeasible, has not been considered before.

Organization

We will review some preliminaries in the next section. Our results for the small-precision regime in known $\mathbf{\Delta}$ setting are provided in Section 5.5 and in the unknown $\mathbf{\Delta}$ setting are provided in Section 5.6. In Section 5.7, we extend our results to the large-precision regime. Finally, we close with all the proofs in Section 5.8.

5.3 Preliminaries and the structure of our protocols

While our lower bound for the known $\mathbf{\Delta}$ setting holds for an arbitrary SMP protocol, all the protocols we propose in this chapter, for the no side information setting, as well as the known $\mathbf{\Delta}$ and the unknown $\mathbf{\Delta}$ settings in the side information case, have a common structure. We use r-bit quantizers to form estimates of x_is at the server and then compute the sample mean of the estimates of x_is. To describe our protocols and facilitate our analysis, we begin by concretely defining the distributed quantizers needed for this problem. Further, we present a simple result relating the performance of the resulting protocol to the parameters of the quantizer.

An r-bit quantizer Q for input vectors in $\mathcal{X} \subset \mathbb{R}^d$ and side information $\mathcal{Y} \subset \mathbb{R}^d$ consists of randomized mappings[3] $(Q^{\mathsf{e}}, Q^{\mathsf{d}})$ with the encoder mapping $Q^{\mathsf{e}} : \mathcal{X} \to \{0,1\}^r$ used by the client to quantize and the decoder mapping $Q^{\mathsf{d}} : \{0,1\}^r \times \mathcal{Y} \to \mathcal{X}$ used by the server to aggregate quantized vectors. The overall quantizer Q is given by the composition mapping $Q(x,y) = Q^{\mathsf{d}}((Q^{\mathsf{e}}(x), y)$.

In our protocols, for input $\mathbf{x}$ and side information $\mathbf{y}$, client i uses the encoder Q_i^{e} for the r-bit quantizer Q_i to send $Q_i^{\mathsf{e}}(x_i)$. The server uses $Q_i^{\mathsf{e}}(x_i)$ and y_i to form the estimate $\hat{x}_i = Q_i(x_i, y_i)$ of x_i. We assume that the randomness used in quantizers Q_i for different i is independent, whereby $\hat{x}_i$ are independent of each other for different i. Then server finally forms the estimate of the sample mean as

$$\hat{\bar{x}} := \frac{1}{n} \sum_{i=1}^{n} \hat{x}_i. \tag{5.4}$$

[3] We can use public randomness U for randomizing.

For any quantizer Q, the following two quantities will determine its performance when used in our distributed mean estimation protocol:

$$\alpha(Q;\Delta) := \sup_{x\in\mathcal{X},y\in\mathcal{Y}:\|x-y\|_2\le\Delta} \mathbb{E}\left[\|Q(x,y)-x\|_2^2\right],$$
$$\beta(Q;\Delta) := \sup_{x\in\mathcal{X},y\in\mathcal{Y}:\|x-y\|_2\le\Delta} \|\mathbb{E}\left[Q(x,y)-x\right]\|_2^2,$$

where[4] the expectation is over the randomization of the quantizer. Note that $\alpha(Q;\Delta)$ can be interpreted as the worst-case MSE and $\beta(Q,\Delta)$ the worst-case bias over $x\in\mathcal{X}$ and $y\in\mathcal{Y}$ such that $\|x-y\|_2\le\Delta$.

The result below will be very handy for our analysis.

Lemma 5.3.1. *For $\mathbf{x}\in\mathcal{X}^n$ and $\mathbf{y}\in\mathcal{Y}^n$ satisfying (5.2) and r-bit quantizers Q_i, $i\in[n]$, using independent randomness for different $i\in[n]$, the estimate $\hat{\bar{x}}$ in (5.4) and the sample mean $\bar{x}$ in (5.1) satisfy*

$$\mathbb{E}\left[\|\hat{\bar{x}}-\bar{x}\|_2^2\right] \le \sum_{i=1}^n \frac{\alpha(Q_i;\Delta_i)}{n^2} + \sum_{i=1}^n \frac{\beta(Q_i;\Delta_i)}{n}.$$

5.4 Distributed mean estimation with no side information

As stated previously, in the setting of distributed mean estimation with no side information we have $\Delta_i = 1$ and $y_i = 0$, $\forall\ i\ \in[n]$. We will therefore state our results under these assumptions for this case. Our protocol π_n^* uses subsampled RATQ as the quantizer for each client, which is described in Section 3.4.3, with parameters of the Quantizer set as in (3.13) and (3.14).

Theorem 5.4.1. *For $n\ge 2$, fixed $\mathbf{\Delta} = (\Delta_1,\dots,\Delta_n)$, $d\ge r\ge 2\left(3+\lceil\log(1+\ln^*(d/3))\rceil\right)$, and $\mathbf{y}$, where $\Delta_i = 1$ and $y_i = 0$, $\forall\ i\ \in[n]$, the protocol π_n^* with parameters set as in (3.13)*

[4]The α above differs from the α_2 defined in the first part of the thesis; the former characterizes the worst-case MSE, while the latter describes the worst-case L_2 norm. However, β is similar to β_2 defined in the first part of the thesis, as both characterize the worst-case bias of the quantizer.

and (3.14) *is an* r*-bit protcol which satisfies*

$$\mathcal{E}(\pi_n^*, \mathbf{x}, \mathbf{y}) \leq (6 + 2\lceil \log(1 + \ln^*(d/3)) \rceil) \left(\sum_{i \in [n]} \frac{1}{n} \cdot \frac{d}{nr} \right).$$

The proof is a direct extension of the analysis of RATQ presented in Chapter 3 and is deferred to Section 5.8.2.

This matches the lower bound of $\Omega\left(\frac{d}{nr}\right)$, derived in [88, Theorem 5], upto a tight $\log \log^*(d)$. To the best of our knowledge, for $r \ll d$, this is the tightest known upper bound for distributed mean estimation with no-side information. Moreover, the protocol π_n^* can be efficiently implemented as the encoding and decoding complexity of RATQ is $d \log d$.

5.5 Distributed mean estimation with known Δ

In this section, we present our Wyner-Ziv estimator for the known $\boldsymbol{\Delta}$ setting. As described in Section 5.3, we use the the same (randomized) quantizer across all the clients and form the estimate of sample mean as in (5.4). We only need to define the common quantizer used by all the clients, which we do in Section 5.5.3. In Sections 5.5.1 and 5.5.2, we provide the basic building blocks of our final quantizer. Further, in Section 5.5.4, we derive a lower bound for the worst-case MSE that establishes the near-optimality of our protocol. Throughout we restrict to the small-precision setting of $r \leq d$.

5.5.1 Modulo Quantizer (MQ)

The first subroutine used by our larger quantizer is the *Modulo Quantizer* (MQ). MQ is a one dimensional distributed quantizer that can be applied to the input $x \in \mathbb{R}$ with side information $y \in \mathbb{R}$. We give an input parameter Δ' to MQ where $|x - y| \leq \Delta'$. In addition to Δ', MQ also has the resolution parameter k and the lattice parameter ε as inputs.

For an appropriate ε to be specified later, we consider the lattice $\mathbb{Z}_\varepsilon = \{\varepsilon z : z \in \mathbb{Z}\}$. For a given input x, the encoder $Q_{\mathtt{M}}^{\mathtt{e}}$ finds the closest points in $\mathbb{Z}_\varepsilon$ larger and smaller than

x. Then, one of these points is sampled randomly to get an unbiased estimate of x. The sampled point will be of the form $\tilde{z}\varepsilon$, where $\tilde{z}$ is in $\mathbb{Z}$. We note that the chosen point $\tilde{z}$ satisfies

$$\begin{aligned} \varepsilon\mathbb{E}\left[\tilde{z}\right] &= x \text{ and} \\ |x - \varepsilon\tilde{z}| &< \varepsilon, \quad \text{almost surely.} \end{aligned} \tag{5.5}$$

The encoder sends $w = \tilde{z} \mod k$ to the decoder, which requires $\log k$ bits.

Upon receiving this w, the decoder $Q^{\mathtt{d}}$ looks at the set $\mathbb{Z}_{w,\varepsilon} = \{(zk + w) \cdot \varepsilon : z \in \mathbb{Z}\}$ and decodes the point closest to y, which we denote by $Q_{\mathtt{M}}(x, y)$. Note that declaring y will already give a MSE of less than Δ. A useful property of this decoder is that its output is always within a bounded distance from y; namely, since in Step 1 of Alg. 5.3 we look for the closest point to y in the lattice $Z_{w,\varepsilon} := \{(zk + w) \cdot \varepsilon : z \in \mathbb{Z}\}$, the output $Q_{\mathtt{M}}(x, y)$ satisfies

$$|Q_{\mathtt{M}}(x, y) - y| \leq k\varepsilon, \quad \text{almost surely.} \tag{5.6}$$

We summarize MQ in Alg. 5.2 and 5.3.

Require: Input $x \in \mathbb{R}$, Parameters k, Δ', and ε

1: Compute $z_u = \lceil x/\varepsilon \rceil$, $z_l = \lfloor x/\varepsilon \rfloor$

2: Generate $\tilde{z} = \begin{cases} z_u, & w.p.\ x/\varepsilon - z_l \\ z_l, & w.p.\ z_u - x/\varepsilon \end{cases}$

3: **Output:** $Q^{\mathtt{e}}_{\mathtt{M}}(x) = \tilde{z} \mod k$

Algorithm 5.2: Encoder $Q^{\mathtt{e}}_{\mathtt{M}}(x)$ of MQ

The result below provides performance guarantees for $Q_{\mathtt{M}}$. The key observation is that the output $Q_{\mathtt{M}}(x, y)$ of the quantizer equals $\tilde{z}\varepsilon$ with $\tilde{z}$ found at the encoder, if ε is set appropriately.

Lemma 5.5.1. *Consider the Modulo Quantizer $Q_{\mathtt{M}}$ described in Alg. 5.2 and 5.3 with*

Require: Input $w \in \{0, \dots, k-1\}$, $y \in \mathbb{R}$

1: Compute $\hat{z} = \arg\min\{|(zk + w) \cdot \varepsilon - y| : z \in \mathbb{Z}\}$

2: **Output:** $Q_{\mathtt{M}}^{\mathtt{d}}(w, y) = (\hat{z}k + w)\varepsilon$

Algorithm 5.3: Decoder $Q_{\mathtt{M}}^{\mathtt{d}}(w, y)$ of MQ

parameter ε set to satisfy

$$k\varepsilon \geq 2(\varepsilon + \Delta'). \tag{5.7}$$

Then, for every x, y in $\mathbb{R}$ such that $|x - y| \leq \Delta'$, the output $Q_{\mathtt{M}}(x, y)$ of MQ satisfies

$$\mathbb{E}\left[Q_{\mathtt{M}}(x, y)\right] = x \quad \textit{and}$$
$$|Q_{\mathtt{M}}(x, y) - x| \leq \varepsilon, \quad \textit{almost surely.}$$

In particular, we can set $\varepsilon = 2\Delta'/(k-2)$, to get $|Q_{\mathtt{M}}(x, y) - x| \leq 2\Delta'/(k-2)$. Furthermore, the output of $Q_{\mathtt{M}}$ can be described in $\log k$ bits.

We close with a remark that the modulo operation used in our scheme is the simplest and easily implementable version of classic coset codes obtained using nested lattices used in distributed quantization (*cf.* [30, 60, 96]) and was used in [20] as well.

5.5.2 Rotated Modulo Quantizer (RMQ)

We now describe *Rotated Modulo Quantizer (RMQ)*. RMQ and the subsequent quantizers in this section will be used to quantize input vector x in $\mathbb{R}^d$ with side information y in $\mathbb{R}^d$, where $\|x - y\|_2 \leq \Delta$. RMQ first preprocesses the input x and side information y by randomly rotating them and then simply applies MQ for each coordinate. For rotation, we multiply both x and y with a random matrix R, given in (3.6), which is sampled using shared randomness between the encoder and decoder. We formally describe the quantizer in Alg. 5.4 and 5.5.

Remark 28. We remark that the vector $R\,(x - y)$ has zero mean subgaussian coordinates

with a variance factor of Δ^2/d. From Lemma 3.6.6, this implies that for all coordinates i in $[d]$, we have

$$P\left(\left|R\left(x-y\right)\left(i\right)\right| \geq \Delta'\right) \leq 2e^{-\frac{\Delta'^2 d}{2\Delta^2}}.$$

This observation allows us to use $\Delta' \approx \Delta/\sqrt{d}$ for MQ applied to each coordinate.

Require: Input $x \in \mathbb{R}^d$, Parameters k and Δ'

1: Sample R as in (3.6) using public randomness

2: $x' = Rx$

3: **Output:** $Q^{\mathtt{e}}_{\mathtt{M},R}(x) = [Q^{\mathtt{e}}_{\mathtt{M}}(x'(1)), \ldots, Q^{\mathtt{e}}_{\mathtt{M}}(x'(d)]^T$ using parameters k, ε, and Δ' for $Q^{\mathtt{e}}_{\mathtt{M}}$ of Alg. 5.2

Algorithm 5.4: Encoder $Q^{\mathtt{e}}_{\mathtt{M},R}(x)$ of RMQ

Require: Input $w \in \{0, \ldots, k-1\}^d$, $y \in \mathbb{R}^d$,
Parameters k and Δ'

1: Get R from public randomness.

2: $y' = Ry$

3: **Output:** $Q^{\mathtt{d}}_{\mathtt{M},R}(w, y) = R^{-1} \sum_{i \in [d]} Q^{\mathtt{d}}_{\mathtt{M}}(w(i), y'(i))e_i$
using parameters k, ε, and Δ' for $Q^{\mathtt{d}}_{\mathtt{M}}$ of Alg. 5.3,

Algorithm 5.5: Decoder $Q^{\mathtt{d}}_{\mathtt{M},R}(w, y)$ of RMQ

Lemma 5.5.2. *Fix $\Delta \geq 0$. Let $Q_{\mathtt{M},R}$ be RMQ described in Alg. 5.4 and 5.5. Then, for*[5] *$k \geq 4$, $\delta \in (0, \Delta)$, $\Delta' = \sqrt{6(\Delta^2/d)\ln(\Delta/\delta)}$ and the parameter ε of MQ set to $\varepsilon = 2\Delta'/(k-2)$, we have for $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$ that*

$$\alpha(Q_{\mathtt{M},R}; \Delta) \leq \frac{24\,\Delta^2}{(k-2)^2} \ln \frac{\Delta}{\delta} + 154\,\delta^2 \quad \textit{and}$$

$$\beta(Q_{\mathtt{M},R}; \Delta) \leq 154\,\delta^2.$$

[5] *In the proof, we provide a general bound which holds for all k.*

Furthermore, the output of quantizer $Q_{\mathtt{M},R}$ can be described in $d \log k$ bits.

Remark 29. The choice of Δ' in the first statement of the Lemma 5.5.2 is based on Remark 28. We note that δ is a parameter to control the bias incurred by our quantizer. By setting $\Delta' = \Delta$ we can get an unbiased quantizer, but it only recovers the performance obtained by simply using MQ for each coordinate, an algorithm considered in [20] as well.

5.5.3 Subsampled RMQ: A Wyner-Ziv quantizer for $\mathbb{R}^d$

Our final quantizer is a modification of RMQ of previous section where we make the precision less than r bits by randomly sampling a subset of coordinates. Specifically, note that $Q^{\mathtt{e}}_{\mathtt{M},R}(x)$ sends d binary strings of $\log k$ bits each. We reduce the resolution by sending only a random subset S of these strings. This subset is sampled using shared randomness and is available to the decoder, too. Note that $Q^{\mathtt{d}}_{\mathtt{M},R}$ applies $Q^{\mathtt{d}}_{\mathtt{M}}$ to these strings separately; now, we use $Q^{\mathtt{d}}_{\mathtt{M}}$ to decode the entries in S alone. We describe the overall quantizer in Alg. 5.6 and 5.7.

Require: Input $x \in \mathbb{R}$, Parameters k, Δ', and μ

1: Sample $S \subset [d]$ u.a.r. from all subsets of $[d]$ of cardinality μd and sample R as in (3.6) using public randomness

2: **Output:** $Q^{\mathtt{e}}_{\mathtt{WZ}}(x) = \{Q^{\mathtt{e}}_{\mathtt{M}}(Rx(i)) : i \in S\}$ using parameters k, ε, and Δ' for $Q^{\mathtt{e}}_{\mathtt{M}}$ of Alg. 5.2

Algorithm 5.6: Encoder $Q^{\mathtt{e}}_{\mathtt{WZ}}(x)$ of subsampled RMQ

Remark 30. We remark that, typically, when implementing random sampling, we set the unsampled components to 0, as was the case in Chapter 3. However, to get Δ dependent bounds on MSE, we set the unsampled coordinates to the corresponding coordinate of side information and center our estimate appropriately to only have small bias.

The result below relates the performance of our final quantizer $Q_{\mathtt{WZ}}$ to that of $Q_{\mathtt{M},R}$, which was already analysed in the previous section.

Require: Input $w \in \{0, \dots, k-1\}^{\mu d}$, $y \in \mathbb{R}$

1: Get S and R from public randomness

2: Compute $\tilde{x} = (Q_{\mathtt{M}}^{\mathtt{d}}(w(i), Ry(i)), i \in S)$ using parameters k, ε, and Δ' for $Q_{\mathtt{M}}^{\mathtt{d}}$ of Alg. 5.3

3: $\hat{x}_R = \frac{1}{\mu} \sum_{i \in S} (\tilde{x}(i) - Ry(i))\, e_i + Ry$

4: **Output:** $Q_{\mathtt{WZ}}^{\mathtt{d}}(w, y) = R^{-1} \hat{x}_R$

Algorithm 5.7: Decoder $Q_{\mathtt{WZ}}^{\mathtt{d}}(w, y)$ of subsampled RMQ

Lemma 5.5.3. *Fix $\Delta > 0$. Let $Q_{\mathtt{WZ}}$ and $Q_{\mathtt{M},R}$ be the quantizers described in Alg. 5.6 and 5.7 and Alg. 5.4 and 5.5, respectively. Then, for $\mu d \in [d]$, we have for $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$ that*

$$\alpha(Q_{\mathtt{WZ}}; \Delta) \leq \frac{\alpha(Q_{\mathtt{M},R}; \Delta)}{\mu} + \frac{\Delta^2}{\mu} \quad \text{and}$$
$$\beta(Q_{\mathtt{WZ}}; \Delta) = \beta(Q_{\mathtt{M},R}; \Delta).$$

Furthermore, the output of quantizer $Q_{\mathtt{WZ}}$ can be described in $\mu d \log k$ bits.

We are now equipped to prove our first main result. Our protocol $\pi_{\mathtt{k}}^*$ uses $Q_{\mathtt{WZ}}$ for each client as described in Section 5.3 and forms the estimate $\hat{\bar{x}}$ as in (5.4). We set the parameters needed for $Q_{\mathtt{WZ}}$ in Alg. 5.6 and 5.7 as follows: For client i, we set the parameters of MQ as

$$\delta = \frac{\Delta_i}{\sqrt{n}}, \quad \log k = \left\lceil \log(2 + \sqrt{12 \ln n}) \right\rceil, \quad \Delta' = \sqrt{6(\Delta_i^2/d) \ln(\Delta_i/\delta)}, \quad \varepsilon = 2\Delta'/(k-2), \tag{5.8}$$

and set the parameter μ as

$$\mu d = \left\lfloor \frac{r}{\log k} \right\rfloor. \tag{5.9}$$

We characterize the resulting error performance in the next result.

Theorem 5.5.4. *For a $n \geq 2$, a fixed $\mathbf{\Delta} = (\Delta_1, ..., \Delta_n)$, and $d \geq r \geq 2\left\lceil \log(2 + \sqrt{12 \ln n}) \right\rceil$, the protocol π_k^* with parameters as set in (6.2) and (5.9) is an r-bit protocol which satisfies*

$$\mathcal{E}(\pi_k^*, \mathbf{x}, \mathbf{y}) \leq (79 \lceil \log(2 + \sqrt{12 \ln n}) \rceil + 26) \left(\sum_{i=1}^{n} \frac{\Delta_i^2}{n} \cdot \frac{d}{nr} \right),$$

for all $\mathbf{x}, \mathbf{y}$ satisfying (5.2).

Proof. Denoting by Q_i the quantizer $Q_{\texttt{WZ}}$ with parameters set for user i, by Lemmas 5.3.1 and 5.5.3, we get

$$\begin{aligned}
\mathbb{E}\left[\|\hat{\bar{x}} - \bar{x}\|_2^2 \right] &\leq \sum_{i=1}^{n} \frac{\alpha(Q_i; \Delta_i)}{n^2} + \sum_{i=1}^{n} \frac{\beta(Q_i; \Delta_i)}{n} \\
&\leq \frac{1}{\mu n^2} \sum_{i=1}^{n} (\alpha(Q_{\texttt{M},R,i}; \Delta_i) + \Delta_i^2) + \sum_{i=1}^{n} \frac{\beta(Q_{\texttt{M},R,i}; \Delta_i)}{n},
\end{aligned}$$

where $Q_{\texttt{M},R,i}$ denotes RMQ with parameters set for user i. Further, since $k \geq 4$ holds when $n \geq 2$ for our choice of parameters, by using Lemma 5.5.2 and substituting $\delta^2 = \Delta_i^2/n$, we get

$$\begin{aligned}
\alpha(Q_{\texttt{M},R,i}; \Delta_i) &\leq \frac{12\Delta_i^2 \ln n}{(k-2)^2} + \frac{154\Delta_i^2}{n}, \\
\beta(Q_{\texttt{M},R,i}; \Delta_i) &\leq \frac{154\Delta_i^2}{n},
\end{aligned}$$

which with the previous bound gives

$$\begin{aligned}
\mathbb{E}\left[\|\hat{\bar{x}} - \bar{x}\|_2^2 \right] &\leq \frac{1}{\mu d} \left(\frac{12 \ln n}{(k-2)^2} + \frac{154}{n} + 1 + 154\mu \right) \sum_{i=1}^{n} \frac{d\Delta_i^2}{n^2} \\
&\leq \frac{79\lceil \log(2 + \sqrt{12 \ln n}) \rceil + 26}{r} \sum_{i=1}^{n} \frac{d\Delta_i^2}{n^2},
\end{aligned}$$

where in the final bound we used our choice of k, the assumption that $n \geq 2$ (which implies that $d \geq r \geq 6$), and the fact that $\lceil r/\log k \rceil \geq r/2$ if $r \geq 2 \log k$. □

Remark 31. We note that by using MQ for each coordinate without rotating (or even with

rotation using R as above) and with $\Delta' = \Delta_i$ yields MSE less than

$$O\left(\sum_{i=1}^{n} \frac{\Delta_i^2}{n} \cdot \frac{d \log d}{nr}\right),$$

for $r \leq d$. Thus, our approach above allows us to remove the $\log d$ factor at the cost of a (milder for large d) $\log \log n$ factor.

Thus, as can be seen from the lower bound presented in Theorem 5.5.5 below, our Wyner-Ziv estimator π_k^* is nearly optimal. Finally, $Q_{\texttt{WZ}}$ can be efficiently implemented as both the encoding and decoding procedures have nearly-linear time complexity[6] of $O(d \log d)$.

5.5.4 Lower bound

We now prove a lower bound on the MSE incurred by any SMP protocol using r bits per client. The proof relies on the strong data processing inequality in [25] and is similar in structure to the lower bound for distributed mean estimation without side-information in [88].

Theorem 5.5.5. *Fix $\boldsymbol{\Delta} = (\Delta_1, \ldots, \Delta_n)$. There exists a universal constant $c < 1$ such that for any r-bit SMP protocol π, with $r \leq cd$, there exists input $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{2d}$ satisfying (5.2) and such that*

$$\mathcal{E}(\pi, \mathbf{x}, \mathbf{y}) \geq c \min_{i \in [d]} \Delta_i^2 \cdot \frac{d}{nr}.$$

5.6 Distributed mean estimation for unknown $\boldsymbol{\Delta}$

Finally, we present our Wyner-Ziv estimator for the unknown $\boldsymbol{\Delta}$ setting. We first, in Section 5.6.1, describe the idea of correlated sampling from [43], which will serve as an essential building block for all our quantizers in this section. We then build towards our

[6] The most expensive operation at both the encoder and decoder of this estimator is the Hadamard matrix multiplication operation, which requires $d \log d$ real operations.

final quantizer, described in 5.6.4, by first describing its simpler versions in Section 5.6.2 and 5.6.3. Once again, we restrict to the small-precision setting of $r \leq d$.

5.6.1 The correlated sampling idea

Suppose we have two numbers x and y lying in $[0, 1]$. A 1-bit unbiased estimator for x is the random variable $\mathbb{1}_{\{U \leq x\}}$, where U is a uniform random variable in $[0, 1]$. The variance of such an estimator is $x - x^2$. We consider a variant of this estimator given by:

$$\hat{X} = \mathbb{1}_{\{U \leq x\}} - \mathbb{1}_{\{U \leq y\}} + y, \tag{5.10}$$

where, like before, U is a uniform random variable in $[0, 1]$. Such an estimator still uses only 1-bit of information related to x. It is easy to check that this estimator unbiased estimator of x, namely $\mathbb{E}\left[\hat{X}\right] = x$. The variance of this estimator is given by

$$\texttt{Var}(\hat{X}) = \mathbb{E}\left[(\hat{X} - x)^2\right] = |x - y| - (x - y)^2,$$

which is lower than that of the former quantizer when x is close to y. We build-on this basic primitive to obtain a quantizer with MSE bounded above by a $\mathbf{\Delta}$-dependent expression, without requiring the knowledge of $\mathbf{\Delta}$.

5.6.2 Distance Adaptive Quantizer (DAQ)

DAQ and subsequent quantizers in this Section will be described for input x and side information y lying in $\mathbb{R}^d$. The first component of our quantizer, DAQ, which uses (5.10) and incorporates the correlated sampling idea discussed earlier. Both the encoder and the decoder of DAQ use the same d uniform random variables $\{U(i)\}_{i=1}^d$ between $[-1, 1]$, which are generated using public randomness. At the encoder, each coordinate of vector x is encoded to the bit $\mathbb{1}_{\{U(i) \leq x(i)\}}$. At the decoder, using the bits received from the encoder, side information y, and the public randomness $\{U(i)\}_{i=1}^d$, we first compute bits $\mathbb{1}_{\{U(i) \leq y(i)\}}$

for each $i \in [d]$. Then, the estimate of x is formed as follows:

$$Q_{\mathtt{D}}(x, y) = \sum_{i=1}^{d} \left(\mathbb{1}_{\{U(i) \le x(i)\}} - \mathbb{1}_{\{U(i) \le y(i)\}} \right) e_i + y.$$

We formally describe the quantizer in Alg. 5.8 and 5.9.

Require: Input $x \in \mathbb{R}^d$

1: Sample $U(i) \sim Unif[-1, 1], \forall i \in [d]$

2: $\tilde{x} = \sum_{i=1}^{d} \mathbb{1}_{\{U(i) \le x(i)\}} \cdot e_i$

3: **Output:** $Q_{\mathtt{D}}^{\mathtt{e}}(x) = \tilde{x}$, where $\tilde{x}$ is viewed as binary vector of length d

Algorithm 5.8: Encoder $Q_{\mathtt{D}}^{\mathtt{e}}(x)$ of DAQ

Require: Input $w \in \{0, 1\}^d$, $y \in \mathbb{R}^d$,

1: Get $U(i), \forall i \in [d]$, using public randomness

2: Set $\tilde{y} = \sum_{i=1}^{d} \mathbb{1}_{\{U(i) \le y(i)\}} \cdot e_i$

3: **Output:** $Q_{\mathtt{D}}^{\mathtt{d}}(w, y) = 2(w - \tilde{y}) + y$, where w is viewed as a vector in $\mathbb{R}^d$

Algorithm 5.9: Decoder $Q_{\mathtt{D}}^{\mathtt{d}}(w, y)$ of DAQ

The next result characterizes the performance for DAQ.

Lemma 5.6.1. *Let $Q_{\mathtt{D}}$ denote DAQ described in Algorithms 5.8 and 5.9. Then, for $\mathcal{X} = \mathcal{Y} = \mathcal{B}$ and every $\Delta > 0$, we have*

$$\alpha(Q_{\mathtt{D}}; \Delta) \le 2\Delta\sqrt{d} \quad \textit{and} \quad \beta(Q_{\mathtt{D}}; \Delta) = 0.$$

Furthermore, the output of quantizer $Q_{\mathtt{D}}$ can be described in d bits.

5.6.3 Rotated Distance Adaptive Quantizer (RDAQ)

Next, we proceed as for the known $\boldsymbol{\Delta}$ setting and add a preprocessing step of rotating x and y using random matrix R of (3.6), which is sampled using shared randomness. We

remark that here random rotation is used to exploit the subgaussianity of the rotated x and y, whereas in RMQ of previous section it was used to exploit the subgaussianity of $x - y$. That is, RMQ exploited the fact that each coordinate of the Rotated vector $R(x - y)$ is much smaller compared to each of the coordinate of $(x - y)$, whereas RDAQ exploits the fact that coordinates of both the rotated vectors Rx and Ry are much smaller relative to coordinates of x and y. After this rotation step, we proceed with a quantizer similar to DAQ, but we quantize each coordinate at multiple "scales." We describe this step in detail below.

Using multiple scales. In DAQ, we considered each coordinate of the input vector x to be anywhere between $[-1, 1]$ and used one uniform random variable for each coordinate. Now, we will use h independent uniform random variables for each coordinate, each corresponding to a different scale $[-M_j, M_j]$, $j \in \{0, 1, 2, \ldots, h-1\}$. For convenience, we abbreviate $[h]_0 := \{0, 1, 2, \ldots, h-1\}$.

Specifically, let $U(i, j)$ be distributed uniformly over $[-M_j, M_j]$, independently for different $i \in [d]$ and different $j \in [h]_0$. The values M_js correspond to different scales and are set, along with h, as follows: For all $j \in [h]_0$,

$$M_j^2 := \frac{6}{d} \cdot e^{*j}, \quad \log h := \lceil \log(1 + \ln^*(d/6)) \rceil \,, \tag{5.11}$$

where e^{*j} denotes the jth iteration of e given by $e^{*0} := 1, \quad e^{*1} := e, \quad e^{*j} := e^{e^{*(j-1)}}$. All the dh uniform random variables are generated using public randomness and are available to both the encoder and the decoder.

The intervals $[-M_j, M_j]$ are designed to minimize the MSE of our quantizer by tuning its "resolution" to the "scale" of the input, and while still ensuring unbiased estimates. Observe that this is the second time we are using the general idea of using multiple intervals for quantizing randomly rotated vectors, with the first time being RATQ in Chapter 3.

Multiscale DAQ. After rotation, we proceed as in DAQ, except that we use different scale M_j for different coordinates. Ideally, for the ith coordinate, we would like to

use $M_{z^*(i)}$, where $z^*(i)$ is the smallest index such that both $Rx(i)$ and $Ry(i)$ lie in $[-M_{z^*(i)}, M_{z^*(i)}]$. However, since y is not available to the encoder, we simply resort to sending the smallest value $z(i)$ which is the smallest index such that $Rx(i) \in [-M_{z(i)}, M_{z(i)}]$ and apply the encoder of DAQ h times to compress x at all scales, *i.e.*, we send h bits $(\mathbb{1}_{\{U(i,j)\leq Rx(i)\}}, j \in [h]_0)$.

Thus, the overall number of bits used by RDAQ's encoder is $d \cdot (h + \lceil \log h \rceil)$. At RDAQ's decoder, using $z(i)$, we compute the smallest index $z^*(i)$ containing both $Rx(i)$ and $Ry(i)$. In effect, the decoder emulates the decoder for DAQ applied to Ry, but for scale $M_{z^*(i)}$. The encoding and decoding algorithm of RDAQ are described in Alg. 5.10 and 5.11, respectively.

Require: Input $x \in \mathcal{B}$

1: Sample $U(i,j) \sim Unif[-M_j, M_j]$, $i \in [d], j \in [h]_0$, and sample R as in(3.6) using public randomness.

2: $x_R = Rx$

3: **for** $i \in [d]$ **do**

$$z(i) = \min\{j \in [h]_0 : |x_R(i)| \leq M_j\}$$

4: **for** $j \in [h]_0$ **do**

$$\tilde{x}_j = \sum_{i=1}^{d} \mathbb{1}_{\{U(i,j)\leq x_R(i)\}} e_i$$

5: **Output:** $Q^{\mathsf{e}}_{\mathsf{D},R}(x) = ([\tilde{x}_0, \ldots, \tilde{x}_{h-1}], z)$, where we view $\tilde{x}_j$s as binary vectors

Algorithm 5.10: Encoder $Q^{\mathsf{e}}_{\mathsf{D},R}(x)$ at for RDAQ

Then, the quantized output $Q_{\mathsf{D},R}$ corresponding to input vector x and side-information y is

$$Q_{\mathsf{D},R}(x,y) = R^{-1}\left[\sum_{i=1}^{d} 2M_{z^*(i)}\left(\mathbb{1}_{\{U(i,z^*(i))\leq Rx(i)\}} - \mathbb{1}_{\{U(i,z^*(i))\leq Ry(i)\}}\right) + Ry\right].$$

We remark that since rotated coordinates $Rx(i)$ and $Ry(i)$ have subgaussian tails, with very high probability $M_{z^*(i)}$ will be much less than 1, which helps in reducing the overall

Require: Input $(w, z) \in \{0,1\}^{d\times h} \times [h]_0^d$ and $y \in \mathcal{B}$

1: Get $U(i,j)$, $i \in [d]$, $j \in [h]_0$, and R using public randomness.

2: $y_R = Ry$

3: **for** $i \in [d]$ **do**

$$z'(i) = \min\{j \in \{[h]_0\} : |y_R(i)| \le M_j\}$$

$$z^*(i) = \max\{z(i), z'(i)\}$$

4: $w' = \sum_{i=1}^{d} 2M_{z^*(i)} \left(w(i, z^*(i)) - \mathbb{1}_{\{U(i,z^*(i))\le y_R\}}\right)$

5: $\hat{x}_R = w' + Ry$

6: **Output:** $Q^{\mathtt{d}}_{\mathtt{D},R}(w, y) = R^{-1}\hat{x}_R$.

Algorithm 5.11: Decoder $Q^{\mathtt{d}}_{\mathtt{D},R}(x)$ for RDAQ

MSE significantly. The performance of the algorithm is characterized below.

Lemma 5.6.2. *Let $Q_{\mathtt{D},R}$ be RDAQ described in Alg. 5.10 and 5.11. Then, for $\mathcal{X} = \mathcal{Y} = \mathcal{B}$ and every $\Delta > 0$, we have*

$$\alpha(Q_{\mathtt{D},R}; \Delta) \le 16\sqrt{3}\Delta \quad \textit{and} \quad \beta(Q_{\mathtt{D},R}; \Delta) = 0.$$

Furthermore, the output of quantizer Q can be described in $d(h + \log h)$ bits.

5.6.4 Subsampled RDAQ: A universal Wyner-Ziv quantizer for unit Euclidean ball

Finally, we bring down the precision of RDAQ to r, as before for the known $\boldsymbol{\Delta}$ setting, by retaining the output of RDAQ for only coordinates $i \in S$, where S is generated uniformly at random from all subsets of $[d]$ of cardinality μd using public randomness. Specifically, we execute Alg. 5.10 and 5.11 with S replacing $[d]$ and multiplying w' in Step 4 of Alg. 5.11 by normalization factor of $d/|S|$. The output of the resulting encoder is given by

$$Q^{\mathtt{e}}_{\mathtt{WZ},u}(x) = \{Q^{\mathtt{e}}_{\mathtt{D},R}(x)(i) : i \in S\}, \tag{5.12}$$

where $Q^{\mathtt{e}}_{\mathtt{D},R}(x)(i)$ represents the encoded bits $([\tilde{x}_0(i),\dots,\tilde{x}_{h-1}(i)], z(i))$ for the ith coordinate using RDAQ, and the output of the resulting decoder is given by

$$Q_{\mathtt{WZ},u}(x,y) = R^{-1}\left[\frac{1}{\mu}\sum_{i\in S} 2M_{z^*(i)}\left(\mathbb{1}_{\{U(i,z^*(i))\le Rx(i)\}} - \mathbb{1}_{\{U(i,z^*(i))\le Ry(i)\}}\right) + Ry\right]. \tag{5.13}$$

Lemma 5.6.3. *Let $Q_{\mathtt{WZ},u}$ be the quantizers described in (5.12) and (5.13) and $Q_{\mathtt{D},R}$ be RDAQ described in Alg. 5.10 and 5.11. Then, for $\mu d \in [d]$, $\mathcal{X} = \mathcal{Y} = \mathcal{B}$, and every $\Delta > 0$, we have*

$$\alpha(Q_{\mathtt{WZ},u};\Delta) \le \frac{\alpha(Q_{\mathtt{D},R};\Delta)}{\mu} \quad \text{and} \quad \beta(Q_{\mathtt{WZ},u};\Delta) = 0.$$

Furthermore, the output of quantizer $Q_{\mathtt{WZ},u}$ can be described in $\mu d(h+\log h)$ bits.

We are now equipped to prove our second main result. Our protocol $\pi^*_{\mathtt{u}}$ uses $Q_{\mathtt{WZ},u}$ for each client as described in Section 5.3 and forms the estimate $\hat{\bar{x}}$ as in (5.4). Unlike for the known $\boldsymbol{\Delta}$ setting, we now use the same parameters for $Q_{\mathtt{WZ},u}$ for all clients, given by

$$\mu d = \left\lfloor \frac{r}{h+\log h} \right\rfloor. \tag{5.14}$$

Theorem 5.6.4. *For $d \ge r \ge 2(h+\log h)$ and h given in (5.11), the r-bit protocol π^*_u with parameters as set in (5.14) satisfies*

$$\mathcal{E}(\pi^*_u, \mathbf{x}, \mathbf{y}) \le \left(128\sqrt{3}\,(1+\ln^*(d/6))\right)\left(\sum_{i\in[n]} \frac{\Delta_i}{n}\cdot\frac{d}{nr}\right),$$

for all $\mathbf{x},\mathbf{y}$ satisfying (5.2), for every $\boldsymbol{\Delta} = (\Delta_1,\dots,\Delta_n)$.

Proof. Denote by $\hat{\bar{x}}$ the output of the protocol. Then, by Lemmas 5.3.1 and Lemma 5.6.3, we get

$$\begin{aligned}\mathbb{E}\left[\|\hat{\bar{x}} - \bar{x}\|_2^2\right] &\le \frac{1}{n^2\mu}\sum_{i=1}^n \alpha(Q_{\mathtt{D},R};\Delta_i)\\ &\le \frac{16\sqrt{3}}{n^2\mu}\sum_{i=1}^n \Delta_i,\end{aligned}$$

where the previous inequality is by Lemma 5.6.2. The proof is completed by using $\mu \ge$

$\frac{r}{2d(h+\log h)} \geq \frac{r}{4dh}$, which follows from (5.14) and the assumption that $r \geq 2(h + \log h)$. □

The Wyner-Ziv estimator π_u^* is universal in $\boldsymbol{\Delta}$: it operates without the knowledge of the distance between the input and the side information and yet gets MSE depending on $\boldsymbol{\Delta}$. Moreover, it can be efficiently implemented as both the encoding and the decoding procedures have nearly linear time complexity of $O(d \log d)$.

5.7 The large-precision regime

5.7.1 RMQ in the large-precision regime.

For the known $\boldsymbol{\Delta}$ setting, our quantizer RMQ described in Alg. 4 and 5 remains valid even for $r > d$. We will assume $r = md$ for integer $m \geq 2$. For each client i, we set

$$\delta = \frac{\Delta_i}{n^{\frac{1}{2}}(2^{r/d} - 2)}, \quad \log k = \frac{r}{d}, \quad \Delta' = \sqrt{6(\Delta_i^2/d) \ln \Delta_i/\delta}, \quad \varepsilon = \frac{2\Delta'}{k-2}. \tag{5.15}$$

The performance of protocol π_k^* using RMQ with parameters set as in (5.15) for each client can be characterized as follows.

Theorem 5.7.1. *For a fixed $\boldsymbol{\Delta} = (\Delta_1, ..., \Delta_n)$ and $r = md$ for integer $m \geq 2$, the protocol π_k^* with parameters set as in (5.15) satisfies*

$$\mathcal{E}(\pi_k^*, \mathbf{x}, \mathbf{y}) = \left(12 \ln n + \frac{24r}{d} + 154/n + 166\right) \left(\sum_{i \in [n]} \frac{\Delta_i^2}{n} \cdot \frac{1}{n(2^{r/d} - 2)^2}\right),$$

for all $\mathbf{x}, \mathbf{y}$ satisfying (5.2).

Proof. Denoting by Q_i the quantizer $Q_{\mathtt{M},R}$ with parameters set for client i, by Lemmas 5.3.1 and 5.5.2, we get

$$\mathbb{E}\left[\|\hat{\bar{x}} - \bar{x}\|_2^2\right] \leq \sum_{i=1}^n \frac{\alpha(Q_i; \Delta_i)}{n^2} + \sum_{i=1}^n \frac{\beta(Q_i; \Delta_i)}{n}$$

Further, since $k \geq 4$ holds when $r \geq 2d$ for our choice of parameters, by using Lemma 5.5.2

and substituting $\delta^2 = \Delta_i^2/n(2^{r/d}-2)^2$, we get

$$\alpha(Q_i;\Delta_i) \le \frac{12\Delta_i^2 \ln(n(2^{r/d}-2)^2)}{(2^{r/d}-2)^2} + \frac{154\Delta_i^2}{n(2^{r/d}-2)^2},$$
$$\beta(Q_i;\Delta_i) \le \frac{154\Delta_i^2}{n(2^{r/d}-2)^2}.$$

which with the previous bound gives

$$\mathbb{E}\left[\|\hat{\bar{x}} - \bar{x}\|_2^2\right] \le \left(12\ln n + \frac{24r}{d} + \frac{154}{n} + 154\right)\sum_{i=1}^{n}\frac{\Delta_i^2}{n^2(2^{r/d}-2)^2},$$

where use the inequality $\ln x \le x$, $\forall x \ge 0$, to bound $\ln(2^{r/d}-2)^2/(2^{r/d}-2)^2$ by 1.

□

Remark 32. Similar to Remark 31, we note that using MQ for each coordinate without rotating (or even with rotation using R as above) with $\Delta' = \Delta_i$ yields MSE less than

$$O\left(\sum_{i=1}^{n}\frac{\Delta_i^2}{n}\cdot\frac{d}{n2^{2r/d}}\right),$$

for $r \ge d$. Thus our approach above allows us to remove the d factor at the cost of a (milder for large d) $\log n + r/d$ factor.

5.7.2 Boosted RDAQ: RDAQ in the large-precision regime.

Moving to the unknown $\boldsymbol{\Delta}$ setting, we describe an update to RDAQ described in Alg. 10 and 11 for the large-precision setting. For brevity, we denote by $m := r/d$ the number of bits per dimension. A straight-forward scheme to make use of the high precision is to independently implement the RDAQ quantizer approximately $\lfloor m/\ln^* d \rfloor$ times and use the average of the quantized estimates as the final estimate. We will see that the MSE incurred by such an estimator is $O(\Delta \ln^* d/m)$. We will show that this naive implementation can be significantly improved and an exponential decay in MSE with respect to m can be achieved.

We boost RDAQs performance as follows. Simply speaking, instead of sending the

bits produced by multiple instances of the encoder of RDAQ, we send the "type" of each sequence. A similar idea appeared in [67] for the case without any side information. At the encoding stage of RDAG given in Alg. 10 and 11, after random rotation and computing z in Steps 1 to 3 of Alg. 10, we repeat Step 4 N times with independent randomness each time and store only the total number of ones seen for each coordinate i and scale j. Specifically, let $U_t(i,j)$ be an independent uniform random variable in $[-M_j, M_j]$, for all $i \in [d], j \in [h]_0$, and $t \in [N]$, which are generated using public randomness between the encoder and the decoder. Using this randomness, we compute $\tilde{x}_{j,t} = \sum_{i=1}^{d} \mathbb{1}_{\{U_t(i,j) \le x_R(i)\}} e_i$ for all $j \in [h]_0$. Then, instead of storing $\tilde{x}_{j,t}$ for each j and t, we store the sum $\sum_{t=1}^{n} \tilde{x}_{j,t}$ for each $j \in [h]_0$. Since each coordinate of the sum can be stored in $\log N$ bits, the new encoder's output can be stored in $d(h \log N + \log h)$. Thus, we can implement this scheme by using $m = (h \log N + \log h)$ bits per dimension.

At the decoding stage, we rotate y and compute z^* in precisely the same manner as done in Steps 1 to 3 of the decoding Alg. 11 of RDAQ. Then, using the encoded input received, the side-information y, the same random variables $U_t(i,j)$ and random matrix R used by the encoder, the final estimate $Q(x)$ is

$$Q(x) = R^{-1} \left(\frac{1}{N} \cdot \sum_{i \in [d]} \sum_{t \in [N]} \left(B^t_{i,Rx} - B^t_{i,Ry} \right) e_i + Ry \right), \tag{5.16}$$

where $B^t_{i,v} = \mathbb{1}_{\{U_n(i,z^*(i)) \le v(i)\}}$ for v in $\mathbb{R}^d$.

The result below characterizes the performance of our quantizer Boosted RDAQ Q.

Lemma 5.7.2. *Let Q be Boosted RDAQ described above. Then, we have for $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$ and every $\Delta > 0$, we have*

$$\alpha_u(Q;\Delta) \le \frac{16\sqrt{3}\Delta}{N} \quad \textit{and} \quad \beta_u(Q;\Delta) = 0.$$

Furthermore, the output of the quantizer can be described in $d(h \log N + \log h)$ bits.

Thus, when we have a total precision budget of $r = dm$ bits using the Boosted RDAQ algorithm with number of repetitions $N = 2^{\lfloor (m - \log h)/h \rfloor}$, we get an exponential decay in

MSE with respect to m.

We consider the protocol π_u^* that uses the Q above for each client with M_j and h set as in (5.11), *i.e.*, with

$$N = 2^{\lfloor (m-\log h)/h \rfloor},\ M_j^2 = \frac{6e^{*j}}{d}, j \in [h]_0,\ \log h = \lceil \log(1 + \ln^*(d/6)) \rceil. \tag{5.17}$$

Therefore, by the previous lemma and Lemma 5.3.1, we get the following result.

Theorem 5.7.3. *For $r = dm$ with integer $m \geq h + \log h$, the protocol π_u^* with parameters as set in (5.17) satisfies*

$$\mathcal{E}(\pi_u^*, \mathbf{x}, \mathbf{y}) = \sum_{i \in [n]} \frac{\Delta_i}{n} \cdot \frac{64\sqrt{3}}{n 2^{r/(d(2+2\ln^*(d/6)))}},$$

for all $\mathbf{x}, \mathbf{y}$ satisfying (5.2), for every $\mathbf{\Delta} = (\Delta_1, ..., \Delta_n)$.

Proof. Denote by $\hat{\bar{x}}$ the output of the protocol. Then, by Lemmas 5.3.1 and Lemma 5.7.2, we get

$$\begin{aligned} \mathbb{E}\left[\|\hat{\bar{x}} - \bar{x}\|_2^2\right] &\leq \frac{1}{n^2} \sum_{i=1}^n \alpha(Q; \Delta_i) \\ &\leq \frac{16\sqrt{3}}{n^2 N} \sum_{i=1}^n \Delta_i, \end{aligned}$$

where the previous inequality is by Lemma 5.7.2. The proof is completed by using

$$N \geq \frac{2^{m/h}}{2^{1+(\log h)/h}} \geq \frac{2^{m/h}}{4} \geq \frac{2^{m/(2+2\ln^*(d/6))}}{4},$$

where the first inequality follows from using $\lfloor x \rfloor \geq x - 1$ for the floor function in the value of N in (5.17), the second follows from the fact that $\log x \leq x, \forall x \geq 0$, and the third follows from $\lceil x \rceil \leq x + 1$ for the ceil function in the value of h in (5.17). □

5.8 Proofs of results

5.8.1 Proof of Lemma 5.3.1

For the estimator $\hat{\bar{x}}$ in (5.4), with $\hat{x}_i = Q_i(x_i, y_i)$, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\frac{1}{n}\cdot\sum_{i\in[n]} Q_i(x_i,y_i) - \frac{1}{n}\cdot\sum_{i\in[n]} x_i\right\|_2^2\right] \\
&= \frac{1}{n^2}\cdot\sum_{i\in[n]}\mathbb{E}\left[\|Q_i(x_i,y_i)-x_i\|_2^2\right] + \frac{1}{n^2}\cdot\sum_{i\neq j}\mathbb{E}\left[\langle Q_i(x_i,y_i)-x_i, Q_j(x_j,y_j)-x_j\rangle\right] \\
&= \frac{1}{n^2}\cdot\sum_{i\in[n]}\mathbb{E}\left[\|Q_i(x_i,y_i)-x_i\|_2^2\right] + \frac{1}{n^2}\cdot\sum_{i\neq j}\langle\mathbb{E}\left[Q_i(x_i,y_i)\right]-x_i, \mathbb{E}\left[Q_j(x_j,y_j)\right]-x_j\rangle \\
&= \frac{1}{n^2}\cdot\sum_{i\in[n]}\mathbb{E}\left[\|Q_i(x_i,y_i)-x_i\|_2^2\right] + \left(\frac{1}{n}\cdot\sum_i\|\mathbb{E}\left[Q_i(x_i,y_i)\right]-x_i\|_2\right)^2 \\
&\qquad\qquad - \frac{1}{n^2}\cdot\sum_i\|\mathbb{E}\left[Q_i(x_i,y_i)\right]-x_i\|_2^2 \\
&\leq \frac{1}{n^2}\cdot\sum_{i\in[n]}\mathbb{E}\left[\|Q_i(x_i,y_i)-x_i\|_2^2\right] + \frac{(n-1)}{n^2}\cdot\sum_i\|\mathbb{E}\left[Q_i(x_i,y_i)\right]-x_i\|_2^2,
\end{aligned}$$

where the second identity uses the independence of $Q_i(x_i, y_i)$ for different i and the final step uses Jensen's inequality. The result follows by bound each term using the fact that $\mathbf{x}$ and $\mathbf{y}$ satisfy (2) and the definitions of $\alpha(Q_i, \Delta_i)$ and $\beta(Q_i, \Delta_i)$, for $i \in [n]$. □

5.8.2 Proof of Theorem 5.4.1

We will need the following Lemma to complete the proof of Theorem 5.4.1.

Lemma 5.8.1. *For any $r \geq \Omega(\log \ln^* d)$ and for any Y such that $\|Y_2\| \leq 1$, let Q be the composition of RCS with RATQ. Then, $Q(Y)$ can be represented in r bits, $\mathbb{E}\left[Q(Y) \mid Y\right] = Y$, and*

$$\mathbb{E}\left[\|Q(Y) - Y\|_2^2 \mid Y\right] \leq \frac{d(3 + \lceil\log(1 + \ln^* d/3)\rceil)}{r}.$$

Proof. By the description of RATQ we have that $Q_{\mathtt{at},R} = R^{-1}Q_{\mathtt{at},I}(RY)$, where $Q_{\mathtt{at},I}$ is

as defined in (3.25). Thus, using the fact that R is a unitary matrix

$$\mathbb{E}\left[\|Q_{\mathtt{at},R}(Y)-Y\|_2^2\right]=\mathbb{E}\left[\|Q_{\mathtt{at},I}(RY)-RY\|_2^2\right].$$

When the parameters are set as in (3.14), we get

$$RY(j)\leq M_{h-1}\text{ a.s., }\forall j\in[d],$$

whereby

$$\mathbb{E}\left[\|Q_{\mathtt{at},R}(Y)-Y\|_2^2\right]=\mathbb{E}\left[\sum_{j\in[d]}(Q_{\mathtt{at},I}(RY)(j)-RY(j))^2\mathbb{1}_{RY(j)\leq M_{h-1}}\right].$$

The proof is completed by noting that Y satisfies $\|Y\|_2\leq 1$ a.s., setting $m=3/d$ and $m_0=(2/d)\ln s$, and applying Lemma 3.6.9. □

Combining the Lemma above with Lemma 5.3.1 completes the proof of upper bound. □

5.8.3 Proof of Lemma 5.5.1

As mentioned in (5.5), the integer $\tilde{z}$ found in Alg. 5.2 satisfies $\mathbb{E}[\tilde{z}\varepsilon]=x$ and $|x-\tilde{z}\varepsilon|<\varepsilon$. Therefore, it suffices to show that the output of the quantizer satisfies $Q_{\mathtt{M}}(x,y)=\tilde{z}\varepsilon$.

To see that $Q_{\mathtt{M}}(x,y)=\tilde{z}\varepsilon$, denote the lattice used in decoding Alg. 5.3 as $\mathbb{Z}_{w,\varepsilon}:=\{(zk+w)\cdot\varepsilon : z\in\mathbb{Z}\}$. The decoding algorithm finds the point in $\mathbb{Z}_{w,\varepsilon}$ that is closest to y. Note that $w=\tilde{z} \mod k$, whereby $\tilde{z}\varepsilon$ is a point in this lattice. Further, for any other point $\lambda\neq\tilde{z}\varepsilon$ in the lattice, we must have

$$|\lambda-\tilde{z}\varepsilon|\geq k\varepsilon,$$

and so, by triangular inequality, that

$$|\lambda-y|\geq|\lambda-\tilde{z}\varepsilon|-|\tilde{z}\varepsilon-y|\geq k\varepsilon-|\tilde{z}\varepsilon-y|.$$

Thus, $\tilde{z}\varepsilon$ is closer to y than λ if

$$k\varepsilon > 2|\tilde{z}\varepsilon - y|. \tag{5.18}$$

Next, by using (5.5) once again, we have

$$|\tilde{z}\varepsilon - y| \leq |\tilde{z}\varepsilon - x| + |x - y| < \varepsilon + \Delta',$$

which by condition (5.7) in the lemma implies that (5.18) holds. It follows that $|\lambda - y| > |\tilde{z}\varepsilon - y|$ for every $\lambda \in \mathbb{Z}_{w,\varepsilon}$, which shows that $Q_{\mathtt{M}}(x, y) = \tilde{z}\varepsilon$ and completes the proof. □

5.8.4 Proof of Lemma 5.5.2

Recall from Remark 28 that for the random matrix R given in (3.6), for every vector $z \in \mathbb{R}^d$, the random variables $Rz(i)$, $i \in [d]$, are sub-Gaussian with variance parameter $\|z\|_2^2/d$. Furthermore, we need the following bound for "truncated moments" of sub-Gaussian random variables.

Lemma 5.8.2. *For a sub-Gaussian random Z with variance factor σ^2 and every $t \geq 0$, we have*

$$\mathbb{E}\left[Z^2 \mathbb{1}_{\{|Z|>t\}}\right] \leq 2(2\sigma^2 + t^2)e^{-t^2/2\sigma^2}.$$

Proof. Note that for any nonnegative random variable U, it can be verified that

$$\mathbb{E}\left[U \mathbb{1}_{\{U>x\}}\right] = xP(U > x) + \int_x^\infty P(U > u)\, du.$$

Upon substituting $U = Z^2$ and $x = t^2$, along with the fact that Z is sub-Gaussian with variance parameter σ^2, we get

$$\begin{aligned}
\mathbb{E}\left[Z^2 \mathbb{1}_{\{Z^2>t^2\}}\right] &= t^2 P(Z^2 > t^2) + \int_{t^2}^\infty P(Z^2 > u)\, du \\
&\leq 2t^2 e^{-t^2/2\sigma^2} + 2\int_{t^2}^\infty e^{-u/2\sigma^2}\, du \\
&\leq 2(t^2 + 2\sigma^2)e^{-t^2/2\sigma^2},
\end{aligned}$$

which completes the proof. □

We now handle the MSE $\alpha(Q)$ and bias $\beta(Q)$ separately below.

Bound for MSE $\alpha(Q)$: Denote by $Q_{\mathtt{M},R}(x,y)$ the final quantized value of the quantizer RMQ. For convenience, we abbreviate

$$\hat{x}_R := R\,Q_{\mathtt{M},R}(x,y).$$

Observe that $\hat{x}_R = \sum_{i\in[d]} Q_{\mathtt{M}}(Rx(i), Ry(i))e_i$, where $Q_{\mathtt{M}}$ is the MQ of Alg. 5.2 and 5.3 with parameters $k \geq$ and Δ' set as in the statement of the lemma. Since R is a unitary transform, we have

$$\begin{aligned}
\mathbb{E}\left[\|Q_{\mathtt{M},R}(x,y) - x\|_2^2\right] &= \mathbb{E}\left[\|\hat{x}_R - Rx\|_2^2\right] \\
&= \sum_{i=1}^{d} \mathbb{E}\left[(\hat{x}_R(i) - Rx(i))^2\right] \\
&= \sum_{i=1}^{d} \mathbb{E}\left[(\hat{x}_R(i) - Rx(i))^2 \mathbb{1}_{\{|R(x-y)(i)|\leq\Delta'\}}\right] \\
&\qquad + \sum_{i=1}^{d} \mathbb{E}\left[(\hat{x}_R(i) - Rx(i))^2 \mathbb{1}_{\{|R(x-y)(i)|\geq\Delta'\}}\right] \qquad (5.19)
\end{aligned}$$

We consider each error term on the right-side above separately. We can view the first term as the error corresponding to MQ, when the input lies in its "acceptance range." Specifically, under the event $\{|R\,(x-y)\,(i)| \leq \Delta'\}$, we get by Lemma 5.5.1 that

$$|\hat{x}_R(i) - Rx(i)| \leq \varepsilon = \frac{2\Delta'}{k-2}, \quad \text{almost surely,}$$

whereby

$$\sum_{i=1}^{d} \mathbb{E}\left[(\hat{x}_R(i) - Rx(i))^2 \mathbb{1}_{|R(x-y)(i)|\leq\Delta'}\right] \leq d\,\varepsilon^2. \qquad (5.20)$$

The second term on the right-side of (5.19) corresponds to the error due to "overflow" and is handled using concentration bounds for the rotated vectors. Specifically, we get

$$\begin{aligned}
&\sum_{i=1}^{d} \mathbb{E}\left[(\hat{x}_R(i) - Rx(i))^2 \mathbb{1}_{\{|R(x-y)(i)|\geq \Delta'\}}\right] \\
&\leq 2\sum_{i=1}^{d}\left[\mathbb{E}\left[(\hat{x}_R(i) - Ry(i))^2 \mathbb{1}_{\{|R(x-y)(i)|\geq \Delta'\}}\right] + \mathbb{E}\left[(Rx(i) - Ry(i))^2 \mathbb{1}_{\{|R(x-y)(i)|\geq \Delta'\}}\right]\right] \\
&\leq 2k^2\varepsilon^2 \sum_{i=1}^{d} P(|R\,(x-y)\,(i)| \geq \Delta') + 2\sum_{i=1}^{d}\mathbb{E}\left[(Rx(i) - Ry(i))^2 \mathbb{1}_{\{|R(x-y)(i)|\geq \Delta'\}}\right] \\
&\leq 4dk^2\varepsilon^2 e^{-d\Delta'^2/2\Delta^2} + 2\sum_{i=1}^{d}\mathbb{E}\left[(Rx(i) - Ry(i))^2 \mathbb{1}_{\{|R(x-y)(i)|\geq \Delta'\}}\right] \\
&\leq 4dk^2\varepsilon^2 e^{-d\Delta'^2/2\Delta^2} + 4(2\Delta^2 + d\Delta'^2) e^{-\frac{d\Delta'^2}{2\Delta^2}}, \qquad (5.21)
\end{aligned}$$

where the second inequality follows upon noting that from the description decoder of MQ in Alg. 5.3 that $|\hat{x}_R(i) - Ry(i)| \leq \varepsilon k$ almost surely for each $i \in [d]$; the third inequality uses the fact that $R(x-y)(i)$ is sub-Gaussian with variance parameter $\|x-y\|_2^2/d \leq \Delta^2/d$; and fourth inequality is by Lemma 5.8.2.

Upon combining (5.19), (5.20), and (5.21), and substituting $\varepsilon = 2\Delta'/(k-2)$ and $\Delta'^2 = 6(\Delta^2/d)\log \Delta/\delta$, we obtain

$$\begin{aligned}
\mathbb{E}\left[\|Q_{\mathtt{M},R}(x,y) - x\|_2^2\right] &\leq d\,\varepsilon^2 + 4dk^2\varepsilon^2 e^{-\frac{d\Delta'^2}{2\Delta^2}} + 4(2\Delta^2 + d\Delta'^2) e^{-\frac{d\Delta'^2}{2\Delta^2}} \qquad (5.22) \\
&= 24\,\frac{\Delta^2}{(k-2)^2}\ln\frac{\Delta}{\delta} + 96\delta^2\left(\frac{k}{k-2}\right)^2 \cdot \frac{\ln(\Delta/\delta)}{(\Delta/\delta)} + 8\delta^2 \cdot \frac{1+3\ln(\Delta/\delta)}{(\Delta/\delta)} \\
&\leq 24\,\frac{\Delta^2}{(k-2)^2}\ln\frac{\Delta}{\delta} + \left(\frac{96}{e}\left(\frac{k}{k-2}\right)^2 + \frac{24}{e^{2/3}}\right)\cdot\delta^2,
\end{aligned}$$

where we used $(1+3\ln u)/u \leq 3/e^{2/3}$ and $(\ln u)/u \leq 1/e$ for every $u > 0$. We conclude by noting that for $k \geq 4$,

$$\left(\frac{96}{e}\left(\frac{k}{k-2}\right)^2 + \frac{24}{e^{2/3}}\right) \leq 154.$$

Bias $\beta(Q)$: The calculation for the bias is similar to that we used to bound the second term on the right-side of (5.19). Using the notation $\hat{x}_R$ introduced above, we have

$$\begin{aligned}
&\|\mathbb{E}\left[Q_{\mathtt{M},R}\right] - x\|_2 \\
&= \|\mathbb{E}\left[R^{-1}\left(\hat{x}_R - Rx\right)\right]\|_2 \\
&= \|R\mathbb{E}\left[R^{-1}\left(\hat{x}_R - Rx\right)\right]\|_2 \\
&= \|\mathbb{E}\left[RR^{-1}\left(\hat{x}_R - Rx\right)\right]\|_2 \\
&= \|\mathbb{E}\left[\hat{x}_R - Rx\right]\|_2,
\end{aligned}$$

where the second identity holds since R is a unitary matrix.

Further, since $Q_{\mathtt{M}}(x, y)$ is an unbiased estimate of x when $|x-y| \leq \Delta'$ (see Lemma 5.5.1), by (5.20) and (5.21) we obtain

$$\begin{aligned}
\|\mathbb{E}\left[\hat{x}_R - Rx\right]\|_2^2 &\leq \sum_{i=1}^{d} \mathbb{E}\left[\left(\hat{x}_R(i) - Rx(i)\right)\mathbb{1}_{|R(x-y)_i| \geq \Delta'}\right]^2 \\
&\leq \sum_{i=1}^{d} \mathbb{E}\left[\left(\hat{x}_R(i) - Rx(i)\right)^2 \mathbb{1}_{|R(x-y)(i)| \geq \Delta'}\right] \\
&\leq 154\,\delta^2,
\end{aligned}$$

which completes the proof. □

5.8.5 Proof of Lemma 5.5.3

Mean Square Error $\alpha(Q_{S,R})$: From the description of Algorithms 5.6 and 5.7, we know that the quantized output of subsampled RMQ $Q_{\mathtt{WZ}}$ for an input x is

$$\begin{aligned}
&Q_{\mathtt{WZ}}(x) = R^{-1}\hat{x}_R, \text{ where} \\
&\hat{x}_R = \frac{1}{\mu}\sum_{i\in[d]} \left(Q_{\mathtt{M}}(Rx(i), Ry(i)) - Ry(i)\right)\mathbb{1}_{\{i\in S\}}\, e_i + Ry,
\end{aligned}$$

and $Q_{\mathtt{M}}(Rx(i), Ry(i))$ denotes the quantized output of the modulo quantizer for an input $Rx(i)$ and side-information $Ry(i)$. Use the shorthand $Q(Rx(i))$ for $Q_{\mathtt{M}}(Rx(i), Ry(i))$, we

have

$$\begin{aligned}
&\mathbb{E}\left[\|Q_{\mathtt{WZ}}(x)-x\|_2^2\right]\\
&=\sum_{i\in[d]}\mathbb{E}\left[\left(\frac{1}{\mu}\left(Q(Rx(i))-Ry(i)\right)\mathbb{1}_{\{i\in S\}}-\left(Rx(i)-Ry(i)\right)\right)^2\right]\\
&=\sum_{i\in[d]}\mathbb{E}\left[\frac{1}{\mu^2}Q(Rx(i))-Rx(i)^2\mathbb{1}_{\{i\in S\}}\right]\\
&\qquad\qquad+\sum_{i\in[d]}\mathbb{E}\left[\left(\frac{1}{\mu}\left(Rx(i)-Ry(i)\right)\mathbb{1}_{\{i\in S\}}-\left(Rx(i)-Ry(i)\right)\right)^2\right]\\
&=\sum_{i\in[d]}\frac{1}{\mu}\mathbb{E}\left[\left(Q(Rx(i))-Rx(i)\right)^2\right]+\sum_{i\in[d]}\mathbb{E}\left[\left(Rx(i)-Ry(i)\right)^2\right]\cdot\mathbb{E}\left[\left(\frac{1}{\mu}\mathbb{1}_{\{i\in S\}}-1\right)^2\right]\\
&=\sum_{i\in[d]}\frac{1}{\mu}\mathbb{E}\left[\left(Q(Rx(i))-Rx(i)\right)^2\right]+\sum_{i\in[d]}\mathbb{E}\left[\left(Rx(i)-Ry(i)\right)^2\right]\cdot\frac{1-\mu}{\mu}\\
&\leq\frac{\alpha(Q_{\mathtt{M},R})}{\mu}+\frac{\Delta^2}{\mu},
\end{aligned}$$

where we used the independence of S and R in the third identity and used the fact that R is unitary in the final step.

Bias $\beta(Q_{S,R})$: This follows upon noting that the conditional expectation (over S) of the output of subsampled RMQ given R is the vector $R^{-1}\sum_{i\in[d]}Q_{\mathtt{M}}(Rx(i),Ry(i))e_i$, which, in turn, is equivalent in distribution to the output of RMQ. □

5.8.6 Proof of Theorem 5.5.5

We denote $\Delta_{min}=\min_{i\in[d]}\Delta_i$ and set y_is to be 0. Let $x_1,...,x_n$ be an *iid* sequence with common distribution such that for all $j\in[d]$ we have

$$x_1(j)=\begin{cases}\frac{\Delta_{min}}{\sqrt{d}} & \text{w.p.}\frac{1+\alpha(j)\delta}{2}\\ -\frac{\Delta_{min}}{\sqrt{d}} & \text{w.p.}\frac{1-\alpha(j)\delta}{2},\end{cases}$$

where $\alpha\in\{-1,1\}^d$ is generated uniformly at random. We have the following Lemma for such x_is, which provides a lower bound for the MSE of any estimator of the mean of the

distribution of x_is.

Lemma 5.8.3. *For $x_1, ..., x_n$ generated as above and any estimator $\hat{\bar{x}}$ of the mean formed using only r-bit quantized version of x_is, we have*[7]

$$\mathbb{E}\left[\left\|\hat{\bar{x}} - \frac{\delta\Delta_{min}}{\sqrt{d}}\alpha\right\|_2^2\right] \geq c' \cdot \frac{d\Delta_{min}^2}{nr},$$

where $c' < 1$ is a universal constant.

Proof of Lemma 5.8.3 follows from either [25, Proposition 2] or [3, Theorem 11].

The proof of Theorem 5.5.5 is completed by using this claim. Specifically, using $2a^2 + 2b^2 \geq (a+b)^2$, we have

$$2\mathbb{E}\left[\|\hat{\bar{x}} - \bar{x}\|_2^2\right] + 2\mathbb{E}\left[\|\bar{x} - \frac{\delta\Delta_{min}}{\sqrt{d}}\alpha\|_2^2\right] \geq \mathbb{E}\left[\|\hat{\bar{x}} - \frac{\delta\Delta_{min}}{\sqrt{d}}\alpha\|_2^2\right],$$

which, along with the observation that

$$\mathbb{E}\left[\|\bar{x} - \frac{\delta\Delta_{min}}{\sqrt{d}}\alpha\|_2^2\right] \leq \frac{\Delta_{min}^2}{n},$$

gives

$$\begin{aligned}\mathbb{E}\left[\|\hat{\bar{x}} - \bar{x}\|_2^2\right] \geq& \frac{c'd\Delta_{min}^2}{2nr} - \frac{\Delta_{min}^2}{n} \\ \geq& \frac{c'\Delta_{min}^2 d}{4nr},\end{aligned}$$

when $(d/r) \geq 4/c'$. The proof is completed by setting $c = c'/4$. □

Remark 33. Since the lower bound in [3] holds for sequentially interactive protocols, if we allow interactive protocols for mean estimation where client i gets to see the messages transmitted by the clients j in $[i-1]$, and can design its quantizers based on these previous messages, even then the lower bound above will hold.

[7] *Note that the side information y_is are all set to 0.*

5.8.7 Proof of Lemma 5.6.1

We will prove a general result which will not only prove Lemma 5.6.1 but will also be useful in the proof of Lemma 5.6.2. Consider x and y in $\mathbb{R}^d$ such that each coordinate of both x and y lies in $[-M, M]$. Also, consider the following generalization of DAQ:

$$Q_{\mathtt{D}}(x, y) = \sum_{i=1}^{d} 2M \left(\mathbb{1}_{\{U(i) \leq x(i)\}} - \mathbb{1}_{\{U(i) \leq y(i)\}} \right) e_i + y,$$

where $\{U_i\}_{i \in [d]}$ are *iid* uniform random variables in $[-M, M]$. We will show that

$$\mathbb{E}\left[Q_{\mathtt{D}}(x, y)\right] = x \quad \text{and} \quad \mathbb{E}\left[\|Q_{\mathtt{D}}(x, y) - x\|_2^2\right] \leq 2M\|x - y\|_1, \tag{5.23}$$

which upon setting $M = 1$ proves Lemma 5.6.1.

Towards proving (5.23), note that from the estimate formed by $Q_{\mathtt{D}}$, it is easy to see that $\mathbb{E}\left[Q_{\mathtt{D}}(x, y)\right] = x$. The MSE can be bounded as follows:

$$\begin{aligned}
\mathbb{E}\left[\|Q_{\mathtt{D}}(x, y) - x\|_2^2\right] &= \sum_{i=1}^{d} \mathbb{E}\left[\left(2M\left(\mathbb{1}_{\{U_i \leq x(i)\}} - \mathbb{1}_{\{U_i \leq y(i)\}}\right) - (x(i) - y(i))\right)^2\right] \\
&= \sum_{i=1}^{d} 4M^2 \frac{|x(i) - y(i)|}{2M} - \|x - y\|_2^2 \\
&= 2M\|x - y\|_1 - \|x - y\|_2^2,
\end{aligned}$$

where we used the observations that $2M\left(\mathbb{1}_{\{U_i \leq x(i)\}} - \mathbb{1}_{\{U_i \leq y(i)\}}\right)$ is an unbiased estimate of $(x(i) - y(i))$ and that $\left(\mathbb{1}_{\{U_i \leq x(i)\}} - \mathbb{1}_{\{U_i \leq y(i)\}}\right)^2$ equals one if and only if exactly one of the indicators is one, which in turn happens with probability $\frac{|x(i)-y(i)|}{2M}$. □.

5.8.8 Proof of Lemma 5.6.2

Worst-case bias $\beta(Q_{\mathtt{D},R}\Delta)$: Since the final interval $[-M_{h-1}, M_{h-1}]$ contains $[-1, 1]$, we can see that $\mathbb{E}\left[Q_{\mathtt{D},R}(x, y)\right] = x$.

Worst-case MSE $\alpha(Q_{\mathtt{D},R};\Delta)$: We denote by B^x_{ij} and B^y_{ij} the bits

$$B^x_{ij} = \mathbb{1}_{\{U(i,j)\leq Rx(i)\}} \text{ and } B^y_{ij} = \mathbb{1}_{\{U(i,j)\leq Ry(i)\}}.$$

Then, the final quantized value of the quantizer RDAQ can be expressed as $Q_{\mathtt{D},R}(X) = R^{-1}\hat{x}_R$ where, with $z^*(i)$ denoting the smallest M_j such that the interval $[-M_j, M_j]$ contains $Rx(i)$ and $Ry(i)$ and $[h]_0 = \{0,\dots,h-1\}$,

$$\hat{x}_R := \sum_{i\in\{1,\dots,d\}} \left(\sum_{j\in[h]_0} 2M_j \cdot \left(B^x_{ij} - B^y_{ij}\right) + Ry(i) \right) \mathbb{1}_{\{z^*(i)=j\}} e_i.$$

Since R is a unitary transform, we get

$$\begin{aligned}
\mathbb{E}\left[\|Q_{\mathtt{D},R}(x) - x\|_2^2\right] &= \mathbb{E}\left[\|RQ_{\mathtt{D},R}(x) - Rx\|_2^2\right] \\
&= \mathbb{E}\left[\|\hat{x}_R - Rx\|_2^2\right] \\
&= \sum_{i\in[d]} \mathbb{E}\left[(\hat{x}_R(i) - Rx(i))^2\right] \\
&= \sum_{i\in[d]} \mathbb{E}\left[\left(\sum_{j\in[h]_0} (2M_j \cdot \left(B^x_{ij} - B^y_{ij}\right) + Ry(i) - Rx(i)) \mathbb{1}_{\{z^*(i)=j\}} \right)^2\right] \\
&= \sum_{i\in[d]} \sum_{j\in[h]_0} \mathbb{E}\left[\left(2M_j \left(B^x_{ij} - B^y_{ij}\right) + Ry(i) - Rx(i)\right)^2 \mathbb{1}_{\{z^*(i)=j\}},\right]
\end{aligned}$$

where the last identity uses $\mathbb{1}_{\{z^*(i)=j_1\}}\mathbb{1}_{\{z^*(i)=j_2\}} = 0$ for all $j_1 \neq j_2$, to cancel the cross-terms in the expansion of $(\hat{x}_R(i) - Rx(i))^2$. Conditioning on R and using the independence of $\mathbb{1}_{\{z^*(i)=j\}}$ from the randomness used in MQ, we get

$$\begin{aligned}
\mathbb{E}\left[\|Q_{\mathtt{D},R}(x) - x\|_2^2\right] &= \sum_{i\in[d]} \sum_{j\in[h]_0} \mathbb{E}\left[\mathbb{E}\left[\left(2M_j \left(B^x_{ij} - B^y_{ij}\right) + Ry(i) - Rx(i)\right)^2 \mid R\right] \mathbb{1}_{\{z^*(i)=j\}}\right] \\
&\leq \sum_{i\in[d]} \sum_{j\in[h]_0} \mathbb{E}\left[2M_j |Rx(i) - Ry(i)| \mathbb{1}_{\{z^*(i)=j\}}\right],
\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{i\in[d]} \mathbb{E}\left[2M_0|Rx(i)-Ry(i)|\mathbb{1}_{\{z^*(i)=0\}}\right] \\
&+ \sum_{i\in[d]}\sum_{j\in[h-1]} \mathbb{E}\left[2M_j|Rx(i)-Ry(i)|\mathbb{1}_{\{z^*(i)=j\}}\right], \\
&\leq \sum_{i\in[d]} \mathbb{E}\left[2M_0|Rx(i)-Ry(i)|\right] \\
&+ \sum_{i\in[d]}\sum_{j\in[h-1]} \mathbb{E}\left[2M_j|Rx(i)-Ry(i)|\mathbb{1}_{\{z^*(i)=j\}}\right], \qquad (5.24)
\end{aligned}
$$

where the first inequality follows from (5.23) in the proof of Lemma 5.6.1.

Next, noting that

$$
\mathbb{1}_{\{z^*(i)=j\}} \leq \mathbb{1}_{\{|RX(i)|\geq M_{j-1}\}} + \mathbb{1}_{\{|RY(i)|\geq M_{j-1}\}} \quad \text{almost surely,}
$$

an application of the Cauchy-Schwarz inequality yields

$$
\begin{aligned}
&\mathbb{E}\left[2M_j|Rx(i)-Ry(i)|\mathbb{1}_{\{z^*(i)=j\}}\right] \\
&\leq 2M_j\mathbb{E}\left[(Rx(i)-Ry(i))^2\right]^{1/2}\mathbb{E}\left[(\mathbb{1}_{\{|RX(i)|\geq M_{j-1}\}}+\mathbb{1}_{\{|RY(i)|\geq M_{j-1}\}})^2\right]^{1/2} \\
&\leq 2M_j\mathbb{E}\left[(Rx(i)-Ry(i))^2\right]^{1/2}(2P(|Rx(i)|\geq M_{j-1})+2P(|Ry(i)|\geq M_{j-1}))^{1/2} \\
&\leq 2M_j\mathbb{E}\left[(Rx(i)-Ry(i))^2\right]^{1/2}\left(8e^{\frac{-dM_{j-1}^2}{2}}\right)^{1/2}, \qquad (5.25)
\end{aligned}
$$

where the second ineqaulity uses $(a+b)^2 \leq 2a^2+2b^2$ and the third uses subgaussianity of $Rx(i)$ and $Ry(i)$.

Substituting the upper bound in (5.25) for the second term in the RHS of (5.24) and using $\mathbb{E}[X] \leq \mathbb{E}[X^2]^{1/2}$ for the first term, we get

$$
\begin{aligned}
\mathbb{E}\left[\|Q_{\mathtt{D},R}(x)-x\|_2^2\right] &\leq \sum_{i\in[d]}\mathbb{E}\left[|Rx(i)-Ry(i)|^2\right]^{1/2}\left(2M_0+\sum_{j\in[h-1]}2M_j\cdot\left(8e^{-\frac{dM_{j-1}^2}{2}}\right)^{1/2}\right) \\
&\leq \sqrt{d\cdot\mathbb{E}\left[\|Rx-Ry\|_2^2\right]}\left(2M_0+\sum_{j\in[h-1]}2M_j\cdot\left(8e^{-\frac{dM_{j-1}^2}{2}}\right)^{1/2}\right) \\
&= \sqrt{d\cdot\|x-y\|_2^2}\left(2M_0+\sum_{j\in[h-1]}2M_j\cdot\left(8e^{-\frac{dM_{j-1}^2}{2}}\right)^{1/2}\right)
\end{aligned}
$$

$$
\begin{aligned}
&= \sqrt{d \cdot \|x-y\|_2^2}\left(2\sqrt{\frac{6}{d}} + \sum_{j\in[h-1]} 2\sqrt{\frac{6e^{*j}}{d}} \cdot \left(8e^{-1.5e^{*(j-1)}}\right)\right) \\
&= 8\sqrt{3}\cdot\sqrt{\|x-y\|_2^2}\left(1 + \sum_{j\in[h-1]} e^{-0.5e^{*}(j-1)}\right) \\
&\leq 16\sqrt{3}\cdot\sqrt{\|x-y\|_2^2},
\end{aligned}
$$

where the second inequality uses the fact that $\sum_i \|a\|_1 \leq \sqrt{d}\|a\|_2$, the first and second indentities follow from the fact that R is unitary transform and substituting for M_is, the final inequality follows from the bound of 1 for $\sum_{j=1}^{\infty} e^{-0.5e^{*}(j-1)}$, which, in turn, can seen as follows

$$
\begin{aligned}
e^{-0.5e^{*}(j-1)} &= e^{-0.5} + e^{-0.5e} + e^{-0.5e^{e}} + \sum_{j=3}^{\infty} e^{-0.5e^{*}(j)} \\
&\leq e^{-0.5} + e^{-0.5e} + e^{-0.5e^{e}} + \sum_{j=3}^{\infty} e^{-0.5je^{e}} \\
&\leq e^{-0.5} + e^{-0.5e} + e^{-0.5e^{e}} + \frac{1}{e^{e^{e}}-1} \\
&\leq 1.
\end{aligned}
$$

□

5.8.9 Proof of Lemma 5.6.3

Worst-case bias $\beta(Q_{\mathtt{WZ},u};\Delta)$: It is straightforward to see that $\mathbb{E}\left[Q_{\mathtt{WZ},u}(x)\right] = x$.

Worst-case MSE $\alpha(Q_{\mathtt{WZ},u};\Delta)$: We denote by B_{ij}^x and B_{ij}^y the bits

$$
B_{ij}^x = \mathbb{1}_{\{U(i,j)\leq Rx(i)\}} \quad \text{and} \quad B_{ij}^y = \mathbb{1}_{\{U(i,j)\leq Ry(i)\}}.
$$

Then, the quantized output can be stated as follows: noting that $Q_{\mathtt{WZ},u}(x) = R^{-1}\hat{x}_R$ where, with $z^*(i)$ denoting the smallest M_j such that the interval $[-M_j, M_j]$ contains $Rx(i)$ and

$Ry(i)$,

$$\hat{x}_R := \left(\sum_{i\in\{1,\dots,d\}} \sum_{j\in\{0,\dots,h-1\}} 2M_j \cdot \left(B^x_{ij} - B^y_{ij}\right) \mathbb{1}_{\{z^*(i)=j\}} \mathbb{1}_{\{i\in S\}} \cdot e_i + Ry \right),$$

Since R is a unitary transform, the mean square error between $Q_{\mathtt{WZ},u}(x)$ and x can be bounded as in the proof of Lemma 5.6.2 as follows:

$$\begin{aligned}
\mathbb{E}\left[\|Q_{\mathtt{WZ},u}(x) - x\|_2^2\right] &= \mathbb{E}\left[\|\hat{x}_R - Rx\|_2^2\right] \\
&= \mathbb{E}\left[\|\hat{x}_R - Rx\|_2^2\right] \\
&= \sum_{i\in[d]} \mathbb{E}\left[\hat{x}_R(i) - Rx(i))^2\right] \\
&= \sum_{i\in[d]} \sum_{j\in[h]} \mathbb{E}\left[\left(2M_j\left(B^x_{ij} - B^y_{ij}\right)\mathbb{1}_{\{i\in S\}} + Ry(i) - Rx(i)\right)^2 \mathbb{1}_{\{z^*(i)=j\}}\right] \\
&= \sum_{i\in[d]} \sum_{j\in[h]} \mathbb{E}\left[\mathbb{E}\left[\left(2M_j\left(B^x_{ij} - B^y_{ij}\right)\mathbb{1}_{\{i\in S\}} + Ry(i) - Rx(i)\right)^2 \mid R\right] \mathbb{1}_{\{z^*(i)=j\}}\right] \\
&\leq \sum_{i\in[d]} \sum_{j\in[h]} \mathbb{E}\left[\frac{2M_j}{\mu} \cdot |Rx(i) - Ry(i)| \cdot \mathbb{1}_{\{z^*(i)=j\}}\right],
\end{aligned}$$

where the inequality follows from similar calculations in the proof of Lemma 5.6.1. The rest of the analysis proceeds as that in the proof of Lemma 5.6.2. □

5.8.10 Proof of Lemma 5.7.2

For $Q(x)$ as in (5.16), we have

$$Q(x) = \sum_{i=1}^{N} q_i/N,$$

where q_i for all $i \in \{1, \dots N\}$ is an unbiased estimate of x and equals in distribution the output of the RDAQ quantizer for an input x and side information y. Moreover, q_is are mutually independent conditioned on R. Therefore,

$$\begin{aligned}
\mathbb{E}\left[\|Q(x) - x\|_2^2\right] &= \mathbb{E}\left[\|\sum_{i=1}^{N} \frac{q_i}{N} - x\|_2^2\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\|\sum_{i=1}^{N} \frac{q_i}{N} - x\|_2^2 | R\right]\right]
\end{aligned}$$

$$
\begin{aligned}
&= \mathbb{E}\left[\sum_{i=1}^{N} \frac{1}{N^2}\mathbb{E}\left[\|q_i - x\|_2^2 | R\right]\right] \\
&\leq 16\sqrt{3}\,\frac{\Delta}{N},
\end{aligned}
$$

where the third identity follows from the conditional independence of q_is after conditioning on R and the fact that q_i is an unbiased estimate of x. The final inequality follows from the fact that q_i equals in distribution the output of the RDAQ quantizer and then using Lemma 5.6.2. □

5.9 Concluding Remarks

In this chapter, we saw that having access to side-information helps for the problem of communication-constrained distributed mean estimation. Using this side-information allows us to break the lower bounds for this problem in the no-side information setting. We suspect that identifying side-information sources and then using them will improve the performance in communication-constrained distributed learning scenarios.

Finally, our techniques could also be used to further exploit the correlation between client data, as was shown in [57]. [57] built upon our work and showed that our proposed quantizer RMQ could also be used to exploit the correlation between client data. Specifically, [57] showed that when client data is "close", the bound in Theorem 5.5.4 can be further improved. The key idea was to use the quantized data from clients as side-information to decode other clients' data.

Chapter 6

Revisiting Gaussian Rate-Distortion

6.1 Synopsis

We consider the problem of Gaussian rate-distortion in both the no side-information and side-information case. In the no side-information case, as a by-product of the quantizers designed in Chapter 3, we obtain an efficient quantizer for Gaussian vectors which attains a rate very close to the Gaussian rate-distortion function and is, in fact, universal for subgaussian input vectors. In the setting where the decoder has access to some side-information, popularly known as the Wyner-Ziv problem, we leverage the quantizers developed in Chapter 5 and obtain an efficient scheme in this setting. Once again, our scheme is universal for subgaussian vectors.

The results presented in this chapter are from [69] and [66].

6.2 Introduction

We revisit the classic Gaussian rate-distortion problem. In the classic Gaussian rate-distortion we seek to quantize a random Gaussian vector to within a specified mean squared error while using as few bits per dimension as possible (*cf.* [18, 32]). Typical fixed-length schemes for this problem draw on its duality with the channel coding problem and modify channel codes to obtain coverings; see, for instance, [64, 85, 93]. However, these

schemes may not be acceptable for two reasons: First, the resulting complexity is still too high for hardware implementation; and second, the resulting schemes are not universal and are tied to Gaussian distributions, specifically.

We also revisit the Gaussian Wyner-Ziv problem (*cf.* [74,91]). Similar to the problem described above, in the Gaussian Wyner-Ziv problem we seek to quantize a random Gaussian vector to within a specified mean squared error while using as few bits per dimension. Except, in this case, the decoder has access to a correlated Gaussian vector. Practical codes for this problem can be found in [53,59,61,77,96]. However, once again, these codes are computationally too expensive and the analysis is tied to the Gaussian distribution.

For the Gaussian rate-distortion problem, we evaluate the performance of a subroutine of RATQ, ATUQ, presented in Chapter 3. Similarly, for the Gaussian Wyner-Ziv problem, we use the quantizer developed in the known $\mathbf{\Delta}$ setting in Chapter 5. Our schemes in both settings have almost constant computational complexity per dimension and require a minuscule excess rate over the optimal asymptotic rate. Moreover, unlike the classical schemes for these problems, we do not require the distribution to be exactly Gaussian, and subgaussianity suffices.

Organization

In Section 6.3, we describe the Gaussian rate-distortion problem, our scheme for this problem which employs quantizers from Chapter 3, and its performance. In Section 6.4, we describe the Gaussian Wyner-Ziv problem, our scheme for this problem which employs quantizers from Chapter 5, and its performance.

6.3 The Gaussian rate-distortion problem

Consider a random vector $X = [X(1), \cdots, X(d)]^T$ with *iid* components $X(1), \cdots, X(d)$ generated from a zero-mean Gaussian distribution with variance σ^2. For a pair (R, D) of nonnegative numbers is an *achiveable* rate-distortion pair if we can find a quantizer Q_d

of precision dR and with mean square error $\mathbb{E}\left[\|X - Q_d(X)\|_2^2\right] \leq dD$. For $D > 0$, denote by $R(D)$ the infimum over all R such that (R, D) constitute an achievable rate-distortion pair for all d sufficiently large. A well-known result in information theory characterizes $R(D)$ as follows (*cf.* [18]):

$$R(D) = \begin{cases} \frac{1}{2}\log\frac{\sigma^2}{D} & \text{if } D \leq \sigma^2, \\ 0 & \text{if } D > \sigma^2. \end{cases}$$

The function $R(D)$ is called the *Gaussian rate-distortion function*.

Over the years, several constructions using error correcting codes and lattices have evolved that attain the rate-distortion function, asymptotically for large d. In this section, we show that a slight variant of ATUQ, too, attains a rate very close to the Gaussian rate-distortion function, when applied to Gaussian random vectors.

Specifically, consider the quantizer $Q_{\mathtt{at},I}$ described earlier in (3.25). Recall that $Q_{\mathtt{at},I}$ can be described by algorithm 3.2 and 3.3 with random matrix R replaced with I. That is, we divide the input vector in $\lceil d/s \rceil$ subvectors and employ ATUQ to quantize them. In fact, we will apply this quantizer not only to a Gaussian random vector, but any random vector with subgaussian components; the components need not even be independent. Thus, we show that our quantizer is almost optimal *universally* for all subgaussian random vectors.

We set the parameters m, m_0, h, s, and $\log(k+1)$ of $Q_{at,I}$ as follows:

$$\begin{aligned} & m = 3v, \quad m_0 = 2v\ln s, \\ & \log h = \left\lceil \log\left(1 + \ln^*\left(\frac{4\ln(8\sqrt{2}v/D)}{3}\right)\right)\right\rceil, \\ & s = \min\{\log h, d\}, \\ & \text{and } \log(k+1) = \left\lceil \log\left(2 + \sqrt{\frac{18v + 6v\ln s}{D}}\right)\right\rceil. \end{aligned} \tag{6.1}$$

Theorem 6.3.1. *Consider a random vector X taking values in $\mathbb{R}^d$ and with components X_i, $1 \leq i \leq d$ such that each X_i is a centered subgaussian random variables with a*

variance factor v. Let Q_d be the d-dimensional $Q_{at,I}$ with parameters as in (6.1). Then, for $d \geq \log h$ and $D < v/4$, Q_d gets the mean square error less than dD using rate R satisfying

$$R \leq \frac{1}{2} \log \frac{v}{D} + O\left(\log \log \log \log^* \log \left(\frac{v}{D}\right)\right).$$

Proof. We split the overall mean square error into two terms and derive upper bounds for each of them. Specifically, we have

$$\begin{aligned}\frac{1}{d} \cdot \mathbb{E}\left[\|X_d - Q_d(X_d)\|_2^2\right] = \frac{1}{d} \cdot \mathbb{E}\left[\sum_{i\in[d]} (X_d(i) - Q_d(X_d)(i))^2 \mathbb{1}_{\{|X_d(i)|\leq M_{h-1}\}}\right] \\ + \frac{1}{d} \cdot \mathbb{E}\left[\sum_{i\in[d]} (X_d(i) - Q_d(X_d)(i))^2 \mathbb{1}_{\{|X_d(i)|> M_{h-1}\}}\right].\end{aligned}$$

The second term on the right-side above can be bounded as follows:

$$\begin{aligned}\frac{1}{d} \cdot \mathbb{E}\left[\sum_{i\in[d]} (X_d(i) - Q_d(X_d)(i))^2 \mathbb{1}_{\{|X_d(i)|> M_{h-1}\}}\right] &= \frac{1}{d} \cdot \mathbb{E}\left[\sum_{i\in[d]} X_d(i)^2 \mathbb{1}_{\{|X_d(i)|> M_{h-1}\}}\right] \\ &\leq \mathbb{E}\left[X_d(1)^4\right]^{1/2} P\left(|X_d(1)| > M_{h-1}\right)^{1/2} \\ &\leq 4\sqrt{2} v e^{-\frac{M_{h-1}^2}{4v}},\end{aligned}$$

where the first inequality follows by the Cauchy-Schwarz inequality and the second follows by Lemma 3.6.6. Note that $M_{h-1}^2 \geq m e^{*(h-1)} \geq 3v e^{*\ln^*(4\ln(8\sqrt{2}v/D)/3)} = 4v \ln(8\sqrt{2}v/D)$, which with the previous bound leads to

$$\frac{1}{d} \cdot \mathbb{E}\left[\sum_{i\in[d]} (X_d(i) - Q_d(X_d)(i))^2 \mathbb{1}_{\{|X_d(i)|> M_{h-1}\}}\right] \leq \frac{D}{2}.$$

Furthermore, by Lemma 3.6.7 we have

$$\frac{1}{d} \cdot \mathbb{E}\left[\sum_{i\in[d]} (X_d(i) - Q_d(X_d)(i))^2 \mathbb{1}_{\{|X_d(i)|\leq M_{h-1}\}}\right] \leq \frac{9v + 3v\ln s}{(k-1)^2} \leq \frac{D}{2},$$

where the last equality holds since $k \geq 1 + \sqrt{\frac{18v + 6v\ln s}{D}}$.

It remains to bound the rate. Note that the overall resolution used for the entire vector

is

$$d\log(k+1)+\left\lceil\frac{d}{s}\right\rceil\log h\leq d\left\lceil\log\left(2+\sqrt{\frac{18v+6v\ln s}{D}}\right)\right\rceil+d+\log h$$

Therefore, for $d\geq\log h$ and $D<v/4$, the proof is completed by bounding the rate R as

$$\begin{aligned}R&\leq\log\left(2+\sqrt{\frac{v}{D}}\sqrt{18+6\ln\log h}\right)+3\\&\leq\frac{1}{2}\log\frac{v}{D}+\log\left(1+\sqrt{18+6\ln\left\lceil\log\left(1+\ln^*\left(\frac{4\ln(8\sqrt{2}v/D)}{3}\right)\right)\right\rceil}\right)+3\\&\leq\frac{1}{2}\log\frac{v}{D}+O\left(\log\log\log\log^*\log\frac{v}{D}\right).\end{aligned}$$

□

We remark that the additional term is a small constant for reasonable values of the parameters v and D. Note that our proposed quantizer just uses uniform quantizers with different dynamic ranges, and yet is almost universally rate optimal.

6.4 The Gaussian Wyner-Ziv problem

Consider the random vectors $X=[X(1),\cdots,X(d)]^T$ and $Y=[Y(1),\cdots,Y(d)]^T$, where the coordinates $\{X(i),Y(i)\}_{i=1}^d$ form an *iid.* sequence. Furthermore, for all $i\in[d]$, let

$$X(i)=Y(i)+Z(i),$$

where $Y(i)$ and $Z(i)$ are independent and zero-mean Gaussian random variables with variances σ_y^2 and σ_z^2, respectively. The encoder has access X, which it quantizes and sends to the decoder. The decoder, on the other hand, has access to Y (note that encoder does not have acess to Y) and can use it to decode X. A pair (R,D) of non-negative numbers is an achievable rate-distortion pair if we can find a quantizer Q_d of precision dR and with mean square error $\mathbb{E}\left[\|Q_d(X,Y)-X\|_2^2\right]\leq dD$. For $D\geq 0$, denote by $R_{\mathtt{WZ}}(D)$ the infimum over all R such that (R,D) constitute an achievable rate-distortion pair for all d

sufficiently large. From[1] [91], $R_{\texttt{WZ}}(D)$ can be characterized as follows:

$$R_{\texttt{WZ}}(D) = \begin{cases} \frac{1}{2}\log\frac{\sigma_z^2}{D} & \text{if} \quad D \le \sigma_z^2, \\ 0 & \text{if} \quad D > \sigma_z^2. \end{cases}$$

Several constructions that involve computational heavy methods such as error correcting codes and lattice encoding attain the rate-distortion function, asymptotically for large d. In this section, we show that modulo quantizer with parameters set appropriately attains a rate very close to the rate-distortion function $R_{\texttt{WZ}}(D)$. Moreover, we will show that this rate can be achieved for arbitrary Y and Z, as long as Z is a zero mean subgaussian random variable with variance factor σ_z^2. Our proposed quantizer $Q_d(X, Y)$ uses the modulo quantizer to quantize $X(i)$ with side information $Y(i)$ at the decoder and the parameter k, Δ' set as follows:

$$\begin{aligned} &\delta = \sqrt{D/308}, \quad \log k = \left\lceil \log\left(2 + (\sigma_z/\sqrt{D})4\sqrt{3\ln(2\sqrt{77}\sigma_z/\sqrt{D})}\right)\right\rceil \\ &\Delta' = \sqrt{6(\sigma_z^2)\ln(\sigma_z/\delta)}, \quad \varepsilon = 2\Delta'/(k-2), \end{aligned} \tag{6.2}$$

Theorem 6.4.1. *Consider random vectors X, Y in $\mathbb{R}^d$, where for all coordinates $i \in \{1, \ldots, d\}$, we have*

$$X(i) = Y(i) + Z(i),$$

and $Z(i)$ is a centered subgaussian random variable with variance factor of σ_z^2, independent of $Y(i)$. Let $Q_d(X, Y)$ be the quantizer described above. Then, for $D \le \frac{\sigma^2}{308}$, we have MSE less than dD using rate satisfying

$$R \le \frac{1}{2}\log\frac{\sigma_z^2}{D} + O\left(\log\log\frac{\sigma_z^2}{D}\right).$$

[1]The model considered in [91] and perhaps the more popular Wyner-Ziv model is $Y = X + Z$. Nevertheless, through MMSE rescaling this model can be converted to $X = Y' + Z'$ (see, for instance, [60]).

Proof. The proof of this Theorem is similar to that of Lemma 5.5.2. We denote by $Q(X(i), Y(i))$ the output of the modulo quantizer with side information $Y(i)$ and parameters k, Δ' set as in (6.2). Then, we have

$$\begin{aligned}\mathbb{E}\left[\|Q_d(X,Y)-X\|^2\right] &\leq \sum_{i=1}^{d}\mathbb{E}\left[(Q(X(i),Y(i))-X(i))^2\right] \\ &\leq \sum_{i=1}^{d}\mathbb{E}\left[(Q(X(i),Y(i))-X(i))^2\mathbb{1}_{\{|(X(i)-Y(i))|\leq\Delta'\}}\right] \\ &\qquad + \sum_{i=1}^{d}\mathbb{E}\left[(Q(X(i),Y(i))-X(i))^2\mathbb{1}_{\{|(X(i)-Y(i))|\geq\Delta'\}}\right].\end{aligned} \tag{6.3}$$

We bound the first term on the right-side in a similar manner as the bound in (5.20). Specifically, under the event $\{|X(i)-Y(i)|\leq\Delta'\}$, we get by Lemma 5.5.1 that

$$|Y(i)-X(i)|\leq\varepsilon=\frac{2\Delta'}{k-2},\quad \text{almost surely,}$$

whereby

$$\sum_{i=1}^{d}\mathbb{E}\left[(Y(i)-X(i))^2\mathbb{1}_{\{|X(i)-Y(i)\}|\leq\Delta'}\right]\leq d\,\varepsilon^2. \tag{6.4}$$

For the second term in the RHS note that $X(i)-Y(i)$ is subgaussian with variance factor σ_z^2. Therefore, by proceeding in a similar manner as the derivation of (5.21) we get

$$\begin{aligned}&\sum_{i=1}^{d}\mathbb{E}\left[(Q(X(i),Y(i))-X(i))^2\mathbb{1}_{\{|X(i)-Y(i)|\geq\Delta'\}}\right] \\ &\leq 2\sum_{i=1}^{d}\left[\mathbb{E}\left[(Q(X(i),Y(i))-Y(i))^2\mathbb{1}_{\{|X(i)-Y(i)|\geq\Delta'\}}\right]+\mathbb{E}\left[(Y(i)-X(i))^2\mathbb{1}_{\{|X(i)-Y(i)|\geq\Delta'\}}\right]\right] \\ &\leq 2k^2\varepsilon^2\sum_{i=1}^{d}P(|X(i)-Y(i)|\geq\Delta')+2\sum_{i=1}^{d}\mathbb{E}\left[(X(i)-Y(i))^2\mathbb{1}_{\{|X(i)-Y(i)|\geq\Delta'\}}\right] \\ &\leq 4dk^2\varepsilon^2e^{-d\Delta'^2/2\sigma_z^2}+2\sum_{i=1}^{d}\mathbb{E}\left[(X(i)-Y(i))^2\mathbb{1}_{\{|X(i)-Y(i)|\geq\Delta'\}}\right]\end{aligned}$$

$$\leq 4dk^2\varepsilon^2 e^{-\Delta'^2/2\sigma_z^2} + 4(2\sigma_z^2 + d\Delta'^2)e^{-\frac{\Delta'^2}{2\sigma_z^2}}, \tag{6.5}$$

where the second inequality follows upon noting from the description decoder of MQ in Alg. 5.3 that $|Q(X(i), Y(i)) - Y(i)| \leq \varepsilon k$ almost surely for each $i \in [d]$; the third inequality uses the fact that $X(i) - Y(i)$ is sub-Gaussian with variance parameter σ_z^2; and the fourth inequality is by Lemma 5.8.2.

Upon bounding the two terms on the right-side of (6.3) from above using (6.4), (6.5), we obtain

$$\mathbb{E}\left[\|Q_d(X,Y) - X\|^2\right] \leq d\varepsilon^2 + 4dk^2\varepsilon^2 e^{-\Delta'^2/2\sigma_z^2} + 4(2\sigma_z^2 + d\Delta'^2)e^{-\frac{\Delta'^2}{2\sigma_z^2}}.$$

Note that the RHS in the upper bound above is precisely the same as in (5.22) with σ_z^2 replacing Δ^2/d.Therefore proceeding in the same manner as in (5.22), we get

$$\mathbb{E}\left[\|Q_d(X,Y) - X\|^2\right] \leq 24\frac{\sigma_z^2}{(k-2)^2}\ln\frac{\sigma_z}{\delta} + 154\delta^2.$$

Substituting the value of k and δ completes the proof.

□

6.5 Concluding Remarks

The key difference between our proposed quantizers and those proposed in the literature is that we don't focus on precisely matching the rate-distortion function. This allows us to design computationally efficient quantizers, which still have a rate close to the rate-distortion function.

Part III

Source Coding Schemes for Timeliness

Chapter 7

Minimum Age Source Codes

7.1 Synopsis

A transmitter observing a sequence of independent and identically distributed random variables seeks to keep a receiver updated about its latest observations. The receiver need not be apprised about each symbol seen by the transmitter, but needs to output a symbol at each time instant t. If at time t the receiver outputs the symbol seen by the transmitter at time $U(t) \leq t$, the age of information at the receiver at time t is $t - U(t)$. We study the design of lossless source codes that enable transmission with minimum average age at the receiver. We show that the asymptotic minimum average age can be attained up to a constant gap by the Shannon codes for a tilted version of the original pmf generating the symbols, which can be computed easily by solving an optimization problem. Furthermore, we exhibit an example with alphabet $\mathcal{X}$ where Shannon codes for the original pmf incur an asymptotic average age of a factor $O(\sqrt{\log |\mathcal{X}|})$ more than that achieved by our codes. Underlying our prescription for optimal codes is a new variational formula for integer moments of random variables, which may be of independent interest. Also, we discuss possible extensions of our formulation to randomized schemes and to the erasure channel, and include a treatment of the related problem of source coding for minimum average queuing delay.

The results presented in this Chapter are from [65] and [**?**]

7.2 Introduction

Timeliness is emerging as an important requirement for communication in cyber-physical systems (CPS). Broadly, it refers to the requirement of having the latest information from the transmitter available at the receiver in a timely fashion. It is important to distinguish the requirement of timeliness from that of low delay transmission: The latter places a constraint on the delay in transmission of each message, while timeliness is concerned about how recent is the current information at the receiver. In particular, the instantaneous staleness at the receiver is low if a message is received with low delay. However, the instantaneous staleness increases linearly at the receiver until a subsequent message is decoded successfully. A heuristically appealing metric that can capture the notion of timeliness of information in a variety of applications, termed its *age*, was first used in [50] for a setting involving queuing and link layer delays and was analyzed systematically for a queuing model in the pioneering work [51]; see [10, 12, 42, 54, 87, 94] for a sampling of subsequent developments in problems related to minimum age scheduling. In this paper, we initiate a systematic study of the design of source codes with the goal of minimizing the age of the information at the receiver.

As a motivating application, consider remote sensor data monitoring where at each instant the sensor observes real-valued, time-series measurements. For concreteness, the reader may consider voltage and current data recording using intelligent electronic devices in a power distribution network. The sensor communicates to a center over a network to enable fault detection and fault analysis. On the one hand, the communication protocol and buffer constraints at the sensor limits the rate at which the sensor can send data packets to the center. On the other hand, it is not very important for the center to get all the packets from the sensor. Rather the center wants timely updates about the sensor observations. In fact, when operating with cheap hardware with limited front-end buffers, it is common to have observation values in the buffer overwritten as new recordings are made even before the previous one waiting in the buffer has been picked-up for processing. Our work focuses on data compression for such applications where there is no direct cost of skipping packets and the interest is only in timely updates.

7.2.1 Main Contributions

Specifically, we consider the problem of source coding where a transmitter receives symbols generated from a known distribution and seeks to communicate them to a receiver in a timely fashion.[1] To that end, it encodes a symbol x to $e(x)$ using a variable length prefix-free code e. The coded sequence is then transmitted over a noiseless communication channel that sends one bit per unit time. We restrict our treatment to a simple class of deterministic[2] update schemes, termed *memoryless update schemes*, where the transmitter does not have have a buffer to store the symbols it has seen previously and simply sends the next observed symbol once the channel is free.

Specifically, denoting the source alphabet by $\mathcal{X}$, the transmitter observes a symbol $X_t \in \mathcal{X}$ at each discrete time t. At time $t = 1$, the transmitter communicates the symbol $X_1 = x_1$ by encoding it to codeword $e(x_1)$ of length $\ell(x_1)$ bits. This transmission requires $\ell(x_1)$ channel uses and is received perfectly at the decoder at time $1 + \ell(x_1)$. Since the channel remains busy sending $e(x_1)$ for time instants 1 to $\ell(x_1)$, the transmitter cannot send any new symbols during this period. At time $t' = 1 + \ell(x_1)$, the transmitter observes the symbol $X_{t'} = x_{t'}$. Under a memoryless update scheme, the transmitter cannot store the symbols seen during the time interval $\{2, \ldots, \ell(x)\}$ and communicates codeword $e(x_{t'})$ over the next $\ell(x_{t'})$ channel uses, starting from the time instant $t' = 1 + \ell(x_1)$. The communication process continues repeatedly in this fashion.

We emphasize that under memoryless schemes, the source symbols generated and observed by the transmitter while the channel is busy sending a previous symbol are simply skipped. This skipping is only allowed when the channel is busy, and not at the will of the encoder when the channel is free (see Section 7.7 for discussion on randomized schemes that allow the transmitted to skip symbols even when channel is free). Furthermore, the encoder need not indicate to the decoder that a symbol has been skipped using a special symbol – the decoder can ascertain this from the received communication since the channel is noiseless and compression is done using prefix-free codes.

[1] This assumption of known distribution is realized in practice by building a model for sensor data offline, before initiating the live monitoring process.

[2] Our analysis of average age extends to randomized schemes as well; see Section 7.7.

On the receiver side, at each instance t the decoder outputs a time $U(t)$ and the symbol $X_{U(t)}$ seen by the transmitter at time $U(t)$. Thus, the *age of information* at the receiver at time t is given by $A(t) = t - U(t)$. We note that age of information measures timeliness at the receiver. When the transmitter skips source symbols, $U(t)$ remains unchanged at the receiver and the age $A(t)$ increases. Therefore, the age metric implicitly penalizes for skipping symbols.

We illustrate the setup in Figure 7.1. In this example, the symbol X_1 generated at time $t = 1$ is encoded to a two-bit codeword $e(X_1)$ and received at the decoder at time $t = 3$ after two channel uses. At time $t = 2$, the transmitter skips symbol X_2 since the channel was busy sending X_1 when it arrived. Further, the decoder retains $U(t) = 0$ since it has not received any symbol. At time $t = 3$, the decoder receives the codeword $e(X_1)$, updates $U(3) = 1$, and outputs the corresponding symbol X_1. Thus, the age of information at the receiver at time $t = 3$ is $A(3) = 2$. Since the channel becomes available at time $t = 3$, the transmitter encodes the symbol X_3 and transmits the one-bit codeword $e(X_3)$, which is received after a single channel-use at time $t = 4$. At time $t = 4$, the decoder outputs time $U(4) = 3$ with outputs the corresponding symbol X_3, and the age of information at the receiver is $A(4) = 1$. Once again, the channel becomes available at time $t = 4$ for the transmitter. It encodes the current symbol X_4 into the codeword $e(X_4)$ of length 3 bits and sends $e(X_4)$ over the channel; $e(X_4)$ is received at time $t = 7$. The decoder retains the output $U(t) = 3$ and $X_{U(t)} = X_3$ for times $t \in \{4, 5, 6\}$. At time $t = 7$, the decoder outputs time $U(7) = 4$ and the corresponding symbol X_4; the age of information at the receiver is $A(7) = 3$.

Our goal in this paper is to design prefix-free codes for which the average age of the memoryless scheme above is minimized; namely codes e that minimize

$$\bar{A}(e) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} A(t).$$

This formulation is apt for the timely update problem where the transmitter need not send each update and strives only to reduce the average age of the information at the receiver.

Using a simple extension of the renewal reward theorem, we derive a closed form

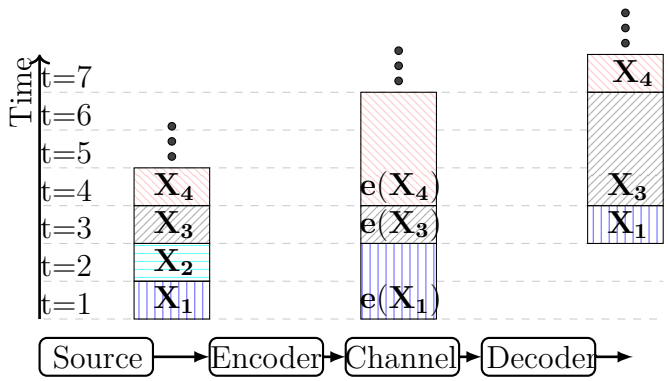


Figure 7.1: Illustration of a memoryless update scheme for the first 4 packets in the source-queue.

formula for the asymptotic average age attained by a prefix-free code. Interestingly, this formula is a rational function of the first and the second moment of the random codeword length. Our main technical contribution in this paper is a variational formula for the second moment of random variables that enables an algorithm for finding the code that attains the minimum asymptotic average age up to a constant gap. The variational formula is of independent interest and may be useful in other settings where such cost functions arise; we point-out one such setting in Section 7.7. In fact, our prescribed prefix-free code is a Shannon code[3] for a tilted version of the original pmf. See (7.10) below for the description of the tilted version; it can be computed by solving an optimization problem entailing entropy maximization.

The formula for average age that we derive yields an $O(\log|\mathcal{X}|)$ upper bound on the minimum average age, attained by a fixed length code. We show that the same upper bound of $O(\log|\mathcal{X}|)$ holds for the average age of a Shannon code for the original distribution as well. However, we exhibit an example where Shannon codes for the original distribution have $\Omega(\log|\mathcal{X}|)$ age, while our aforementioned proposed code yields an average age of $O(\sqrt{\log|\mathcal{X}|})$.

In addition to our basic formulation, we present a few extensions of our formulations and other use cases for our proposed variational formula. Specifically, while we restrict to

[3]A Shannon code for P is a prefix-free code that assigns lengths $\ell_S(x) = \lceil -\log P(x) \rceil$ to a symbol x (*cf.* [18]).

deterministic schemes for the most part, our analysis can be extended easily to analyze randomized schemes where the encoder can choose to skip an available transmission slot randomly. This idea of skipping transmission slots arises also in the recent work [87], albeit in a slightly different context. We exhibit an example where a particular randomized scheme outperforms every deterministic scheme. However, our analysis is limited and does not completely clarify the role of randomization; for instance, it remains unclear for which distributions can randomized schemes strictly outperform deterministic ones.

In another direction, we consider the case where the transmission channel is not error-free, but can erase each bit with a known probability. Furthermore, an ACK-NACK feedback indicating the success of transmission is available. Note that for the standard transmission problem, the simple repeat-until-succeed scheme is optimal in this setting. Our analysis can be used to design the optimal source code when we restrict our channel coding to this simple scheme. However, the optimality of the ensuing source-channel coding scheme remains unclear.

Finally, we study the related problem of source coding for ensuring minimum queuing delays. This problem, introduced in [48], is closely related to the minimum age formulation of this paper. Interestingly, our recipe for designing update codes with minimum average age can be extended to this setting as well. However, here, too, our results are somewhat unsatisfactory: Our approach only provides a solution to the real-relaxation of the underlying integer-valued optimization problem and naive rounding-off is far from optimal. Nonetheless, we have included these extensions in the current paper since they indicate the rich potential for our proposed techniques and provide new formulations for future research.

7.2.2 Prior Work

The problem of designing update codes with low average age is related to real-time source coding (*cf.* [63]) where we seek to transmit a stream of data under strict delay bounds. A related formulation has emerged in the control over communication network literature (*cf.* [89]) where an observation is quantized and sent to an estimator/controller to enable

control. Here, too, the requirement is that of communication under bounded delay.

An alternative formulation for minimum age source coding is considered in the recent work [98]. Unlike our formulation, skipping of symbols is prohibited in [98]. Instead, the authors consider fixed-to-variable length block codes and require that each coded symbol be transmitted over a constant rate, noiseless bit-pipe. In this setting, an exact expression for average age is not available, and the authors take recourse to an approximation for average age. This approximate average age is then optimized numerically over a set of prefix-free codes using the algorithm in [56]. The authors further reduce the computational complexity of this algorithm by using the algorithm in [11].

A recent paper [97] extends this problem to include random arrival times of source symbols and applies the algorithm from [56] for optimizing the cost function. Note that the cost function optimized in [56] is similar to the approximate average age of [97,98], but with one crucial difference: While the former is monotonic in both first and second moments of random lengths, the latter is not. In absence of this monotonicity, the optimality of the solution produced by algorithm in [56] is not guaranteed for the cost functions in [97,98]. In a related work [99], the same authors point-out that the average age can be further reduced by allowing the encoder to dynamically control the block-length of the fixed-to-variable length codes.

In contrast to [98], which is perhaps closest to our work, we derive an exact expression for average age and rigorously establish the structural properties of the optimal solution to the relaxed problem. In fact, our proposed minimum average age problem differs from all these prior formulations since we need not send the entire stream and are allowed to skip some symbols. In our applications of interest, such as that of real-time sensor data monitoring outlined earlier, the allowed communication rates are much lower than the rate at which data is generated. Thus, there is no hope of transmitting all the data at bounded delay, as mandated by the formulations available hitherto. Nonetheless, our setting is related closely to that in [98] and provides a complementary formulation for age optimal source coding. We note that our focus is on settings where the alphabet size of the streaming symbols is large. In such settings, the average age for any memoryless update

scheme would be much larger than a small constant. Therefore, it suffices to establish optimality up to small additive constants.

Some preliminaries

We recall the notions of Shannon lengths and Shannon codes, which will be used throughout. A source code is called *prefix-free* if no codeword is a prefix of another.

Definition 7.2.1 (Shannon lengths and Shannon codes for P)**.** For a pmf P on an alphabet $\mathcal{X}$, the real-values $\ell(x) = -\log P(x), x \in \mathcal{X}$, are called the *Shannon lengths* for the pmf P. A prefix-free source code for P with codeword lengths $\ell(x) = \lceil -\log P(x) \rceil\,,\ \forall x \in \mathcal{X}$, is called a *Shannon code*[4] for the pmf P.

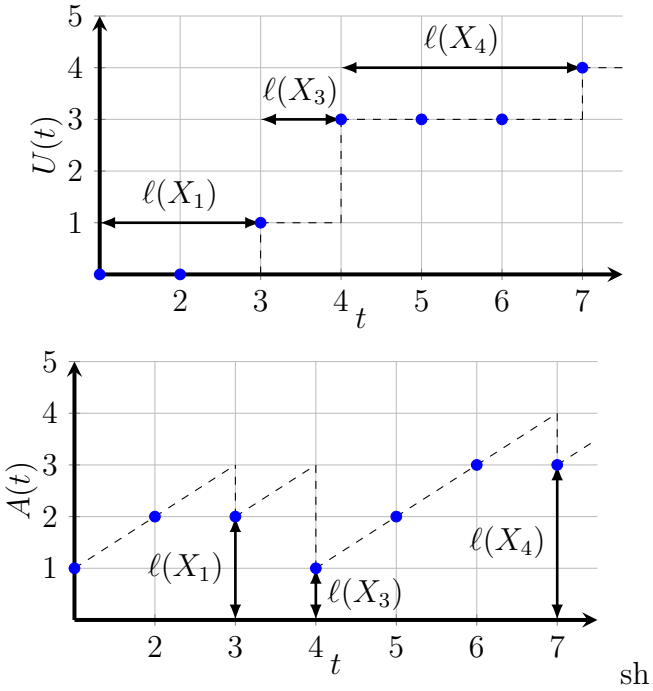
Organization

The next section contains a formal description of our setting and a formula for asymptotic average age of a code. Our main technical tool is presented in Section 7.4, and we apply it to the minimum average age code design problem in Section 7.5. Numerical evaluations of our proposed scheme for the family of Zipf distributions is presented in Section 7.6. Section 7.7 contains a discussion on extensions to randomized schemes and erasure channel, along with a treatment of source codes for minimum average waiting time. We provide all the proofs in the final section.

7.3 Average age for memoryless update schemes

Consider a discrete-time system in which at every time instant t, a transmitter observes a symbol X_t generated from a finite alphabet $\mathcal{X}$ with pmf P. We assume that the sequence $\{X_t\}_{t=1}^{\infty}$ is independent and identically distributed (iid). The transmitter has a noiseless communication channel at its disposal over which it can transmit one bit per unit time. A *memoryless update scheme* consists of a prefix-free code, represented by its encoder

[4]There can be different codes with codeword lengths required in our definition of a Shannon code. We simply refer to all of them as a Shannon code, since any of these can serve our purpose in this paper.



sh

Figure 7.2: A sample path of $U(t)$, $A(t)$ corresponding to Figure 7.1 starting with $U(1) = 0$.

$e : \mathcal{X} \to \{0,1\}^*$, and a decoder which at each time instant t declares a time index $U(t) \leq t$ and an estimate $\hat{X}_{U(t)}$ for the symbol $X_{U(t)}$ that was observed by the encoder at time $U(t)$. We focus on error-free schemes and require $\hat{X}_{U(t)}$ to equal $X_{U(t)}$ with probability 1.

In a memoryless update scheme, once the encoder starts communicating a symbol x, encoded as $e(x)$, it only picks up the next symbol once all the bits in $e(x)$ have been transmitted successfully to the receiver. The time index $U(t)$ is updated to a new value only upon receiving all the encoded bits for the current symbol. That is, if the transmission of a symbol is completed at time $t-1$, the encoder will start transmitting $e(X_t)$ in the next instant. Moreover, if the final bit of $e(X_t)$ is received at time t', $U(t')$ is updated to t. A typical sample path for $U(t)$ is given in Figure 7.2. The age $A(t)$ of the symbol available at the receiver at time t is given by

$$A(t) = t - U(t).$$

A more general treatment can allow errors in estimates of $X_{U(t)}$ as well as encoders with

memory, but we limit ourselves to the simple error-free and memoryless setting in this paper.

We are interested in designing prefix-free codes e that minimize the average age for the memoryless update scheme described above.

Definition 7.3.1. The *average age* for a prefix-free code e, denoted $\bar{A}(e)$, is given by

$$\bar{A}(e) = \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} (t - U(t)).$$

We remark that $\bar{A}(e)$ can be viewed as the average area under the curve of $A(t)$ (w.r.t. t). Note that $\bar{A}(e)$ is random variable, nevertheless we will prove that this random variable is a constant almost surely. For any symbol $x \in \mathcal{X}$, we denote the length of the codeword $e(x)$ by $\ell(x)$. Let $X \in \mathcal{X}$ be a random symbol with pmf P over the alphabet $\mathcal{X}$, then the length of the random codeword $e(X)$ is denoted by

$$L = \ell(X).$$

The result below uses a simple extension of the classical renewal reward theorem (*cf.* [81]) to provide a closed form expression for $\bar{A}(e)$ in terms of the first and the second moments of L.

Theorem 7.3.2. *Consider a random variable X with pmf P on $\mathcal{X}$. For a prefix-free code e, the average age $\bar{A}(e)$ is given by*

$$\bar{A}(e) = \mathbb{E}[L] + \frac{\mathbb{E}[L^2]}{2\mathbb{E}[L]} - \frac{1}{2} \quad a.s. \quad . \tag{7.1}$$

The proof is deferred to Section 7.8.1.

Denoting by $\bar{A}^*$ the minimum average age over all prefix-free codes e, as a corollary of the characterization above, we can obtain the following bounds for $\bar{A}^*$.

Corollary 7.3.3. *For any pmf P over $\mathcal{X}$, the optimal average age $\bar{A}^*$ is bounded as*

$$\frac{3}{2}H(P) - \frac{1}{2} \leq \bar{A}^* \leq \frac{3}{2}\log|\mathcal{X}| + 1.$$

The proof of lower bound simply uses Jensen's inequality $\mathbb{E}[L^2] \geq \mathbb{E}[L]^2$ and the fact that $\mathbb{E}[L] \geq H(P)$ for a prefix free code; the upper bound is obtained by using codewords of constant length $\lceil \log|\mathcal{X}| \rceil$.

Note that the lengths $\ell(x)$ are required to be nonnegative integers. However, for any set of real-valued lengths $\ell(x) \geq 0$, we can obtain integer-valued lengths by using the rounded-off values $\lceil \ell(x) \rceil$. Unlike the average length cost, the average age cost function identified in (7.1) is not an increasing function of the lengths. Nevertheless, by (7.1), the average age $\bar{A}(e)$ achieved when we use the rounded-off values can be bounded as follows: Denoting $\bar{L} := \lceil \ell(X) \rceil$, we have

$$\begin{aligned}
\mathbb{E}\left[\bar{L}\right] + \frac{\mathbb{E}\left[\bar{L}^2\right]}{2\mathbb{E}\left[\bar{L}\right]} - \frac{1}{2} &\leq \mathbb{E}[L+1] + \frac{\mathbb{E}[(L+1)^2]}{2\mathbb{E}[L]} - \frac{1}{2} \\
&\leq \mathbb{E}[L] + \frac{\mathbb{E}[L^2]}{2\mathbb{E}[L]} + \frac{2\mathbb{E}[L]}{2\mathbb{E}[L]} \\
&\quad + \frac{1}{2\mathbb{E}[L]} + \frac{1}{2} \\
&\leq \mathbb{E}[L] + \frac{\mathbb{E}[L^2]}{2\mathbb{E}[L]} + 2. \qquad (7.2)
\end{aligned}$$

Accordingly, in our treatment below we shall ignore the integer constraints and allow nonnegative real-valued length assignments.

Returning now to the bound of Corollary 7.3.3, the upper and lower bounds differ only by a constant 1.5 when P is uniform. In view of the foregoing discussion, Shannon codes for a uniform distribution attain the minimum average age up to a constant gap. The next result gives an upper bound on average age for Shannon codes for an arbitrary P on $\mathcal{X}$.

Lemma 7.3.4. *Given a pmf P on $\mathcal{X}$, a Shannon code e for P has average age $\overline{A}(e)$ at most $O(\log|\mathcal{X}|)$.*

Proof. Let $\ell(X)$ denote the lengths of Shannon code corresponding to P (see Definition

7.2.1). We establish the claim using the standard bound $H(P') \leq \log|\mathcal{X}|$ for an appropriately chosen pmf P' on $\mathcal{X}$. Specifically, for the tilting of P given by $P'(x) \propto \ell(x)P(x)$, we get

$$\begin{aligned}
\log|\mathcal{X}| &\geq \sum_{x\in\mathcal{X}} \frac{P(x)\ell(x)}{\mathbb{E}[\ell(X)]} \log \frac{\mathbb{E}[\ell(X)]}{P(x)\ell(x)} \\
&= \sum_{x\in\mathcal{X}} \frac{P(x)\ell(x)(-\log P(x))}{\mathbb{E}[\ell(X)]} \\
&\qquad - \sum_{x\in\mathcal{X}} \frac{P(x)\ell(x)}{\mathbb{E}[\ell(X)]} \log \frac{\ell(x)}{\mathbb{E}[\ell(X)]} \\
&\geq \sum_{x\in\mathcal{X}} \frac{P(x)\ell(x)(-\log P(x))}{\mathbb{E}[\ell(X)]} \\
&\qquad - \sum_{x\in\mathcal{X}:\ell(x)\geq\mathbb{E}[\ell(X)]} \frac{P(x)\ell(x)}{\mathbb{E}[\ell(X)]} \log \frac{\ell(x)}{\mathbb{E}[\ell(X)]}.
\end{aligned}$$

Using $-\log P(x) \geq \ell(x) - 1$ and $\ln x \leq \frac{x^2-1}{2x}$ for $x \geq 1$, we obtain

$$\begin{aligned}
\log|\mathcal{X}| &\geq \frac{\mathbb{E}[\ell^2(X)]}{\mathbb{E}[\ell(X)]} - 1 \\
&\qquad - \frac{1}{2\ln 2} \cdot \sum_{x\in\mathcal{X}:\ell(x)\geq\mathbb{E}[\ell(X)]} P(x) \left(\frac{\ell^2(x)}{\mathbb{E}[\ell(X)]^2} - 1 \right) \\
&\geq \frac{\mathbb{E}[\ell^2(X)]}{\mathbb{E}[\ell(X)]} - 1 \\
&\qquad - \frac{1}{2\ln 2} \cdot \sum_{x\in\mathcal{X}:\ell(x)\geq\mathbb{E}[\ell(X)]} P(x) \cdot \frac{\ell^2(x)}{\mathbb{E}[\ell(X)]^2} \\
&\geq \frac{\mathbb{E}[\ell^2(X)]}{\mathbb{E}[\ell(X)]} - 1 - \frac{1}{2\ln 2} \cdot \sum_{x\in\mathcal{X}} \frac{P(x)\ell^2(x)}{\mathbb{E}[\ell(X)]^2} \\
&\geq \frac{\mathbb{E}[\ell^2(X)]}{\mathbb{E}[\ell(X)]} - 1 - \frac{1}{2\ln 2} \cdot \sum_{x\in\mathcal{X}} \frac{P(x)\ell^2(x)}{\mathbb{E}[\ell(X)]} \\
&\geq \left(1 - \frac{1}{2\ln 2}\right) \cdot \frac{\mathbb{E}[\ell^2(X)]}{\mathbb{E}[\ell(X)]} - 1,
\end{aligned}$$

where the second-last inequality follows from the fact that $\mathbb{E}[\ell^2(X)] \geq \mathbb{E}[\ell(X)]$, which in turn follows from the fact that $\ell(X) \geq 1$. The proof is completed by rearranging the terms. □

It is of interest to examine if, in general, a Shannon code for P itself has average age close to $\bar{A}^*$, as was the case for the uniform distribution. In fact, it is not the case. Below we exhibit a pmf P where the average age of a Shannon code for P is $\Omega(\log|\mathcal{X}|)$, namely the previous bound is tight, and yet a Shannon code for another distribution (when evaluated for P) has an average age of only $O(\sqrt{\log|\mathcal{X}|})$.

Example 7.3.5. Consider $\mathcal{X} = \{0, ..., 2^n\}$ and a pmf P on $\mathcal{X}$ given by

$$P(x) = \begin{cases} 1 - \frac{1}{n}, & x = 0 \\ \frac{1}{n2^n}, & x \in \{1, \ldots, 2^n\}. \end{cases}$$

Using (7.1), the average age $\bar{A}(e_P)$ for a Shannon code for P can be seen to satisfy $\bar{A}(e_P) \approx (n + 2\log n)/2$. On the other hand, if we instead use a Shannon code for the pmf Q given by

$$Q(x) = \begin{cases} \frac{1}{2\sqrt{n}}, & x = 0 \\ \frac{1-2^{-\sqrt{n}}}{2^n}, & x \in \{1, \ldots, 2^n\}, \end{cases}$$

we get $\mathbb{E}[L] \approx \sqrt{n}$ and $EL^2 \approx 2n$, whereby $\bar{A}(e_Q) \approx 2\sqrt{n}$, just $O(\sqrt{\log|\mathcal{X}|})$. □

Thus, one needs to look beyond the standard Shannon codes for P to find codes with minimum average age. Interestingly, we show that Shannon codes for a tilted version of P attain the optimal asymptotic average age (up to the constant loss of at most 2.5 bits incurred by rounding-off lengths to integers). In particular, for the example above, our proposed optimal codes will have an average age of only $O(\sqrt{\log|\mathcal{X}|})$ in comparison to $\Omega(\log|\mathcal{X}|)$ of Shannon codes for P.

A key technical tool in design of our codes is a variational formula that will allow us to linearize the cost function in (7.1), thereby rendering Shannon codes for a tilted distribution optimal. We present this in the next section.

7.4 A variational formula for p-norm

The expression for average age identified in Theorem 7.3.2 involves the second moment of the random codeword length L. This is in contrast to the traditional variable length source coding problem where the goal is to minimize the average codeword length $\mathbb{E}[L]$. For this standard cost, Shannon codes which assign a codeword of length $\lceil -\log P(x) \rceil$ to the symbol x come within 1-bit of the optimal cost (see, for instance, [18]). A variant of this standard problem was studied in [16], where the goal was to minimize the log-moment generating function $\log \mathbb{E}[\exp(\lambda L)]$. A different approach for solving this problem is given in [40] where the *Gibbs variational principle* is used to linearize the nonlinear cost function $\log \mathbb{E}[\exp(\lambda L)]$. The next result provides the necessary variational formula to extend the aforementioned approach to another nonlinear function, namely $\|L\|_p := (\mathbb{E}[L^p])^{\frac{1}{p}}$ for $p > 1$.

We believe that our result is of independent interest, and present it in a general form that applies to general distributions (and not just the discrete random variables considered in this paper). To state the general result, we recall a basic notation from probability theory. For two probability measures P and Q on the same probability space such that Q is absolutely continuous with respect to P, denoted $Q \ll P$, denote by $\frac{dQ}{dP}$ the Radon-Nikodym derivative of Q with respect to P. Note that $\frac{dQ}{dP}$, too, is a random variable measurable with respect to the underlying sigma-algebra. A reader not familiar with these notions can see a standard textbook on probability theory for definitions. For the discrete case, $Q \ll P$ corresponds to the condition[5] $\texttt{supp}(Q) \subset \texttt{supp}(P)$ and $\frac{dQ}{dP}$ equals the ratio of the pmfs of the distributions Q and P.

Note that expectations are always taken with respect to the reference measure. In particular, the expectations without any subscript in Theorem 7.4.1 below and its proof denote the expectation with respect to P, which is the reference measure in this case. The expectation in Remark 34 denotes the expectation with respect to R.

Theorem 7.4.1. *For a real-valued random variable X with distribution P and $p \geq 1$ such*

[5] $\texttt{supp}(P)$ denotes the support of distribution P over an alphabet $\mathcal{X}$, *i.e.*, $\texttt{supp}(P) := \{x \in \mathcal{X} : P(x) > 0\}$. .

that $\|X\|_p < \infty$*, we have*

$$\|X\|_p = \max_{Q \ll P} \mathbb{E}\left[\left(\frac{dQ}{dP}\right)^{\frac{1}{p'}} |X|\right],$$

where $p' = p/(p-1)$ *is the Hölder conjugate of* p*.*

Proof. For $Q \ll P$ and $0 < \alpha \neq 1$, let $D_\alpha(P, Q)$ denote the Rényi divergence of order α between distributions Q and P (see [79]), defined by

$$D_\alpha(P, Q) := \frac{1}{\alpha - 1} \log \mathbb{E}\left[\left(\frac{dQ}{dP}\right)^{\alpha}\right].$$

It is well-known that $D_\alpha(P, Q) \geq 0$ with equality if and only if $P = Q$. Consider the probability measure $P_p \ll P$ defined by

$$\frac{dP_p}{dP} := \frac{1}{\|X\|_p^p} \cdot |X|^p.$$

Then, for $\alpha = 1/p'$,

$$\begin{aligned} 0 \leq D_\alpha(P_p, Q) &= \frac{1}{\alpha - 1} \log \mathbb{E}\left[\left(\frac{dQ}{dP}\right)^{\alpha} \left(\frac{dP_p}{dP}\right)^{1-\alpha}\right] \\ &= -p \log \mathbb{E}\left[\left(\frac{dQ}{dP}\right)^{\alpha} |X|\right] + p \log \|X\|_p, \end{aligned}$$

where the previous equality holds since $p(1 - \alpha) = 1$. Thus, for every $Q \ll P$,

$$\mathbb{E}\left[\left(\frac{dQ}{dP}\right)^{\alpha} |X|\right] \leq \|X\|_p,$$

with equality if and only if $P_p = Q$. □

Remark 34. The given definition of Rényi divergence restricts Theorem 7.4.1 to the case $P(X = 0) = 0$. To remove this restriction, the following general definition of Rényi

divergence with respect to a common measure can be used: For all $Q, P \ll R$, define

$$D_\alpha(P,Q) := \frac{1}{\alpha - 1} \log \mathbb{E}\left[\left(\frac{dQ}{dR}\right)^{\alpha}\left(\frac{dP}{dR}\right)^{1-\alpha}\right].$$

The proof then follows by using the positivity of $D_\alpha(P_p, Q)$, then by proceeding in the same manner as the previous proof.

Returning to the problem at hand, we apply the variational formula above to the L_2 norm of a discrete random variable. We highlight this special case separately below.

Corollary 7.4.2. *For a discrete random variable X with a pmf P such that $\|X\|_2 < \infty$, we have*

$$\|X\|_2 = \max_{\mathtt{supp}(Q) \subset \mathtt{supp}(P)} \sum_{x \in \mathcal{X}} \sqrt{Q(x)P(x)}x,$$

where $\mathtt{supp}(P)$ *denotes the support-set of the distribution P.*

7.5 Prefix-free codes with minimum average age

We now present a recipe for designing prefix-free codes with minimum average age. By Theorem 7.3.2, we seek prefix-free codes that minimize the cost

$$\mathbb{E}[L] + \frac{\mathbb{E}[L^2]}{2\mathbb{E}[L]}, \tag{7.3}$$

where $L = \ell(X)$ for X with pmf P. Recall that a prefix-free code with lengths $\{\ell(x) \in \mathbb{N}, x \in \mathcal{X}\}$ exists if and only if lengths satisfy Kraft's inequality (*cf.* [18]), *i.e.*, if and only if

$$\sum_{x \in \mathcal{X}} 2^{-\ell(x)} \leq 1. \tag{7.4}$$

Following the discussion leading to (7.2), we relax the integral constraints for $\ell(x)$ and search over all real-valued $\ell(x) \geq 0$ satisfying (7.4). Specifically, we solve the relaxed optimization problem

$$\min_{\ell \in \Lambda} \mathbb{E}[L] + \frac{\mathbb{E}[L^2]}{2\mathbb{E}[L]}, \tag{7.5}$$

where

$$\Lambda = \Big\{ \ell \in \mathbb{R}^{|\mathcal{X}|} : \sum_{x \in \mathcal{X}} 2^{-\ell(x)} \leq 1,\ \ell(x) \geq 0\ \forall x \in \mathcal{X} \Big\}.$$

As noticed in (7.2), this can incur a loss of only a constant. A key challenge in minimizing (7.3) is that it is nonlinear. We linearize this cost as follows:

1. Note first the identity below, which is obtained by maximizing the expression on the right-side:

$$\mathbb{E}\left[L\right] + \frac{\mathbb{E}\left[L^2\right]}{2\mathbb{E}\left[L\right]} = \max_{z \geq 0} \left(1 - \frac{z^2}{2}\right) \mathbb{E}\left[L\right] + z\|L\|_2. \tag{7.6}$$

2. Then, Corollary 7.4.2 yields

$$\|L\|_2 = \max_{Q \ll P} \sum_{x \in \mathcal{X}} \sqrt{Q(x)P(x)}\ell(x),$$

which further leads to

$$\begin{aligned} &\mathbb{E}\left[L\right] + \frac{\mathbb{E}\left[L^2\right]}{2\mathbb{E}\left[L\right]} \\ &= \max_{z \geq 0} \left(1 - \frac{z^2}{2}\right) \mathbb{E}\left[L\right] + z \max_{Q \ll P} \sum_{x \in \mathcal{X}} \sqrt{Q(x)P(x)}\ell(x) \\ &= \max_{z \geq 0} \max_{Q \ll P} \sum_{x \in \mathcal{X}} g_{z,Q,P}(x)\ell(x), \end{aligned}$$

where

$$g_{z,Q,P}(x) := \left(1 - \frac{z^2}{2}\right) P(x) + z\sqrt{Q(x)P(x)}. \tag{7.7}$$

As remarked earlier, as the source distribution P is discrete, the constraint $Q \ll P$ simplifies to $\texttt{supp}(Q) \subset \texttt{supp}(P)$. Thus, our goal is to identify the minimizer ℓ^* that achieves

$$\Delta^*(P) = \min_{\ell \in \Lambda} \max_{z \geq 0} \max_{Q \ll P} \sum_{x \in \mathcal{X}} g_{z,Q,P}(x)\ell(x). \tag{7.8}$$

The result below captures our main observation and facilitates the computation of optimal lengths attaining the minmax cost $\Delta^*(P)$.

Theorem 7.5.1 (Structure of optimal codes)**.** *The optimal minmax cost* $\Delta^*(P)$ *in* (7.8) *satisfies*

$$\begin{aligned}\Delta^*(P) &= \max_{z\geq 0}\max_{Q\ll P}\min_{\ell\in\Lambda}\sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\ell(x)\\ &= \max_{\substack{z\geq 0, Q\ll P,\\ (z,Q)\in\mathcal{G}}}\sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\log\frac{\sum_{x'\in\mathcal{X}} g_{z,Q,P}(x')}{g_{z,Q,P}(x)},\end{aligned} \tag{7.9}$$

where

$$\mathcal{G} := \{z\geq 0, Q\in\mathbb{R}^{|\mathcal{X}|} : g_{z,Q,P}(x)\geq 0 \quad \forall x\in\mathcal{X}\}.$$

Furthermore, if (z^*, Q^*) *is the maximizer of the right-side of* (7.9)*, then the minmax cost* (7.8) *is achieved uniquely by the Shannon lengths*[6] *for the pmf* P^* *on* $\mathcal{X}$ *given by*

$$P^*(x) = \frac{g_{z^*,Q^*,P}(x)}{\sum_{x'\in\mathcal{X}} g_{z^*,Q^*,P}(x')}. \tag{7.10}$$

Thus, our prescription for design of source codes is simple: Use a Shannon code for P^* instead of P. To compute P^*, we need to solve the optimization problem in (7.9). Note that is unclear a priori that the minimum average age for the problem in (7.5) would correspond to Shannon lengths for some pmf since our cost function is not monotonic in expected length, whereby the optimal solution may not satisfy Kraft's inequality with equality. Nonetheless, we show that the Shannon lengths $-\log P^*(x)$ are optimal for the relaxed problem given by (7.5).

We note that our formal result above only provides a structural result for the optimal solution. But we believe that this structural result leads to a recipe to design practical algorithms for finding the optimal solution; we describe this recipe below. Specifically, note that the resulting optimization problem for finding P^* is one of entropy maximization

[6] *Recall that Shannon lengths for the pmf* P *on* $\mathcal{X}$ *are given by* $\ell(x) = -\log P(x)$*,* $x\in\mathcal{X}$*, and are not necessarily integers.*

for which several heuristic recipes are available. Furthermore, we note the following structural simplification for the optimal solution which shows that if $P(x) = P(y)$, then $P^*(x) = P^*(y)$ must hold as well; the proof is relegated to the Appendix. Thus, the dimension of the optimization problem (7.9) can be reduced from $|\mathcal{X}|+1$ to M_P+1, where M_P denotes the number of distinct elements in the probability multiset $\{P(x) : x \in \mathcal{X}\}$. Let $A_1 \cdots A_{M_P}$ denote the partition of $\mathcal{X}$ such that

$$P(x) = P(y) \quad \forall x, y \in A_i, \quad \forall i \in [M_P].$$

Lemma 7.5.2. *Suppose that Q^* is an optimal Q for (7.9). Then, Q^* must satisfy*

$$Q^*(x) = Q^*(y) \quad \forall x, y \in A_i, \quad \forall i \in [M_P]. \tag{7.11}$$

In proving Lemma 7.5.2, we use the fact that the cost function in (7.9) is concave in Q for each fixed z and is concave in z for each fixed Q (see Lemma 7.8.6). However, it may not be jointly concave in (z, Q). Nevertheless, we apply standard numerical packages to optimize it in the next section to quantify the performance of our proposed codes and compare it with Shannon codes for the original distribution P.

7.6 Numerical results for Zipf distribution

We program all our optimization problems in *AMPL* [31] and solve it using *SNOPT* [36] and *CONOPT* [22] solvers. Specifically, for the pmfs P we consider in this section, we solve the optimization problem given by (7.9) to find the corresponding optimal (z^*, Q^*). In order to check if we have indeed found the optimal (z^*, Q^*), we once again use Theorem 7.5.1. In particular, it follows from Theorem 7.5.1 that the necessary and sufficient condition for a particular (z, Q) to be the optimal solution is that the value of the maximization problem (7.9) at (z, Q) equals

$$\mathbb{E}\left[-\log P'(X)\right] + \frac{\mathbb{E}\left[(\log P'(X))^2\right]}{2\mathbb{E}\left[-\log P'(X)\right]},$$

where

$$P'(X) = \frac{g_{z,Q,P}(x)}{\sum_{x' \in \mathcal{X}} g_{z,Q,P}(x')};$$

in all our numerical evaluations, the solution found by the solver satisfies this condition, which establishes its optimality.

We now illustrate our recipe for construction of prefix-free codes that yield minimum average age for memoryless update schemes when P is a Zipf distribution. Specifically, we illustrate our qualitative results using the $\texttt{Zipf}(s, N)$ distribution with alphabet $\mathcal{X} = \{1, \cdots, N\}$ and given by

$$P(i) = \frac{i^{-s}}{\sum_{j=1}^{N} j^{-s}}, \quad 1 \leq i \leq N.$$

Heuristically, the average age formula (7.1) suggests that the differences between the performances of a code under average codeword length cost and the average age cost will be the most for "peaky distribution," namely for distributions with heavy elements. The parameter s of the Zipf distribution allows us to vary from a uniform distribution to a "peaky distribution," making this family apt for our numerical study. Indeed, our numerical results confirm that our proposed scheme outperforms a Shannon code for P when the parameter s is high; see Figure 7.3. When we round-off real lengths to integers, the gains are subsided but still exist. Further, when the parameter s is close to 0, Shannon codes for P are close to optimal. With increase in s, the gain of our proposed schemes over Shannon codes starts becoming more prominent. As an aside, Figure 7.3 also provides an illustration of the non-monotonic nature of the average age function with respect to code lengths.

The distribution P^* we use to construct our codes seems to be a flattened version of the original Zipf distribution; we illustrate the two distributions for $\texttt{Zipf}(1, 8)$ in Figure 7.4. As we see in Figure 7.4, P^* and P are very close in this case. Indeed, we illustrate in Figure 7.5 that the average length $\mathbb{E}[L]$ when Shannon lengths $-\log P(x)$ are used and when $-\log P^*(x)$ are used are very close[7]. In Figure 7.5, we note the dependence of

[7]The difference of these two average lengths (averaged w.r.t. P) is given by the Kullback-Leibler

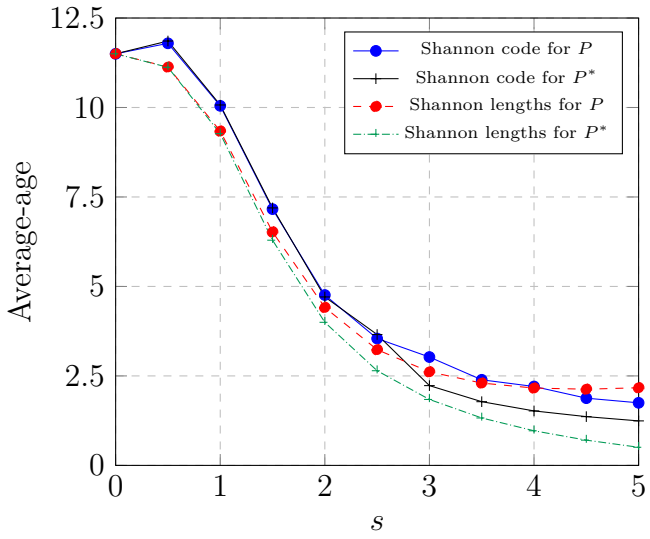


Figure 7.3: Comparison of proposed codes and Shannon codes for $\texttt{Zipf}(s, 256)$ with varying s. The average age is computed using real-valued lengths as well as lengths rounded-off to integer values.

average age on the entropy of the underlying distribution P. As expected, average age increases as $H(P)$ increases.

Thus, while Example 7.3.5 illustrated high gains of the proposed code over Shannon codes for P, for the specific case of Zipf distributions the gains may not be large. Characterizing this gain for any given distribution is a direction for future research.

7.7 Extensions

7.7.1 Randomization for Timely Updates

We have restricted our treatment to deterministic memoryless update schemes. A natural extension to randomized memoryless schemes would entail allowing the encoder to make a randomized decision to skip transmission of a symbol even when the channel is free (we can allocate a special symbol $\emptyset$ to signify no transmission to the receiver). Specifically, assume that we transmit the symbol $\emptyset$ using a codeword of length $\ell(\emptyset)$ when we choose

divergence $D(P\|P^*)$; see [18].

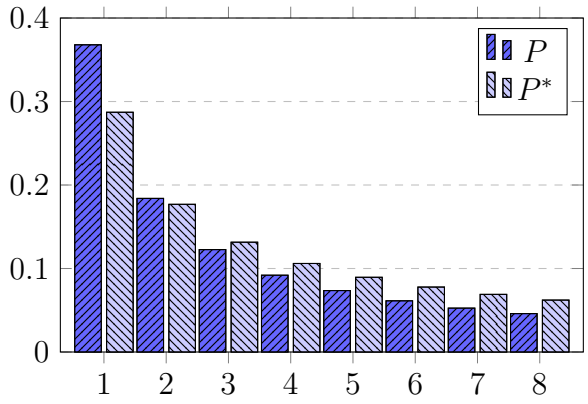


Figure 7.4: The pmf for P^* and P for $\texttt{Zipf}(1, 8)$.

not to transmit the observed symbol $x \in \mathcal{X}$. Denoting by $\theta(x)$ the probability with which the encoder will transmit the symbol x, the average age $\bar{A}(e, \theta)$ for the randomized scheme is given by

$$\bar{A}(e, \theta) = \frac{\mathbb{E}\left[L(\theta)\right]}{\mathbb{E}\left[\theta(X)\right]} + \frac{\mathbb{E}\left[L(\theta)^2\right]}{2\mathbb{E}\left[L(\theta)\right]} - \frac{1}{2}, \tag{7.12}$$

where the random variable $L(\theta)$ is defined as follows:

$$L(\theta) := \begin{cases} \ell(x), & w.p \quad P(x)\theta(x) \\ \ell(\emptyset), & w.p \quad 1 - \mathbb{E}\left[\theta(X)\right]. \end{cases} \tag{7.13}$$

Note that the expression in (7.12) is a slight generalization of Theorem 7.3.2 and is derived in Section 7.8.1.

Example 7.7.1. Consider $\mathcal{X} = \{1, ..., 64\}$ and the following pmf;

$$P(x) = \begin{cases} 1/4, & x \in \{1, \ldots, 3\}, \\ 1/244, & x \in \{4, \ldots, 64\}. \end{cases}$$

Since $H(P) = 3.483$, Corollary 7.3.3 yields that the average age of the deterministic memoryless update scheme is bounded below by 4.724. Next, consider a randomized update scheme with $\theta(x) = 1$ for $x \in \{1, 2, 3\}$ and 0 otherwise. For this choice, the

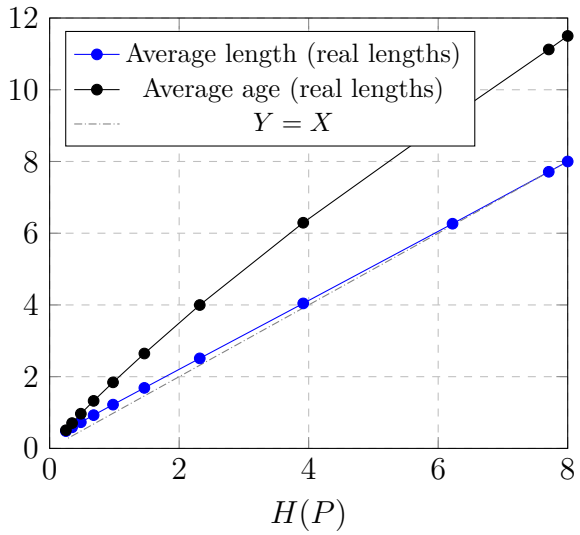


Figure 7.5: Average age and average length for our update codes as a function of $H(P)$ for $\texttt{Zipf}(s, 256)$ with s varying from 0 to 5 at step sizes of 0.5.

effective pmf P_θ is uniformly distributed over the symbols $\{1, 2, 3\} \cup \{\phi\}$. Thus, the optimal length assignment for this case assigns $\ell(x) = 2$ to all the symbols and the average age equals 3.17, which is less than the lower bound of 4.724 for the deterministic scheme.

The idea of skipping available transmission opportunities, i.e., not transmitting even when the channel is free, to minimize average age appears in the recent work [87] as well, albeit in a slightly different setting. Heuristically, the randomization scheme above operates as we expect – it ignores the rare symbols which will require longer codeword lengths. In practice, however, these rare symbols might be the ones we are interested in. But keep in mind that our prescribed solution only promises to minimize the average age and does not pay heed to any other consideration. Furthermore, for a given randomization vector θ, we can establish a result similar to Theorem 7.5.1. This will lead to the design of almost optimal source codes for a given randomization vector θ. However, the joint optimality over the class of randomized schemes and source coding schemes is still unclear.

In a more comprehensive treatment, one can study the design of update codes with other constraints imposed. We foresee the use of Corollary 7.4.2 in these more general settings as well. In another direction, we can consider the extension of our results to the

case when the transmission channel is an erasure channel with probability of erasure ϵ. If we assume the availability of perfect feedback, a natural model for the link or higher layer in a network, and restrict to simple repetition schemes where the transmitter keeps on transmitting the coded symbol until it is received, our formula for average age extends with (roughly) an additional multiplicative factor of $1/(1-\epsilon)$. Formally the average age over an erasure channel with ϵ probability of erasure; a source code e, along with a randomization vector θ and a repetition channel-coding scheme yields the following average age

$$\bar{A}_\epsilon(e,\theta) = \frac{1}{1-\epsilon} \cdot \bar{A}(e,\theta) + \frac{\epsilon}{2(1-\epsilon)}.$$

However, the optimality of repetition scheme is unclear, and the general problem constitutes a new formulation in joint-source channel coding which is of interest for future research.

7.7.2 Source Coding for Minimum Queuing Delay

Next, we point out a use case for Corollary 7.4.2 in a minimum queuing delay problem introduced in [48]. The setting is closely related to our minimum average age update formulation with two differences: First, the arrival process of source symbols is a Poisson process of rate λ; and second, the encoder is not allowed to skip source symbols. Instead, each symbol is encoded and scheduled for transmission in a first-come-first-serve (FCFS) queue. Our goal is to design a source code that minimizes the average queuing delay encountered by the source sequence. Formally, the symbols $\{X_n\}_{n=1}^{\infty}$ are generated iid from a finite alphabet $\mathcal{X}$, using a common pmf P. Every incoming symbol x is encoded as $e(x)$ using a prefix-free code specified by the encoder mapping $e : \mathcal{X} \to \{0,1\}^*$, and the bit string $e(x)$ is placed in a queue. The queue schedules bits for transmission using a FCFS policy. Each bit in the queue is transmitted over a noiseless communication channel. Denote by A_n the time of successful arrival of the nth symbol. Also, denote by D_n the time instant of successful reception of the nth symbol X_n. That is, D_n is the instant at which the last bit of $e(X_n)$ is received[8]. The delay for the nth symbol is given by $D_n - A_n$;

[8]Note both A_n and D_n may not be integer valued, unlike the age setup.

see Figure 7.6 for an illustration.

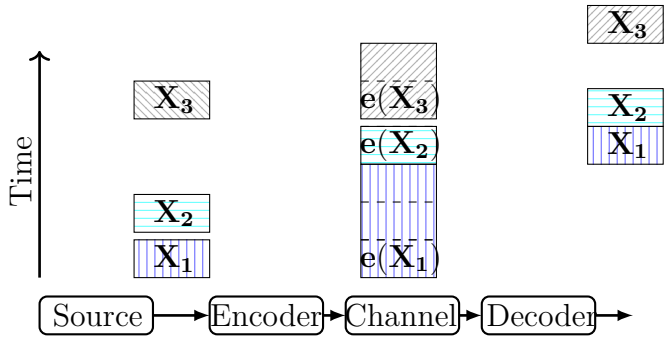


Figure 7.6: Figure describes a typical sample-path for transmission of encoded symbols over a FCFS queuing system. Symbol X_1 arrives at some time instant 1, it is encoded and transmitted over the channel. Recall that unlike the slotted setup of Figure 7.1, the setup here is that of continuous time with Poisson arrivals. It is decoded at time instant 4. Symbol X_2 arrives in between time instants 2 and 3, and is placed in the queue, as the channel is busy transmitting X_1. As soon as the channel becomes free at time instant 4, an encoded version of X_2 is transmitted over it. Symbol X_3 arrives when the channel is free and is transmitted immediately.

Thus, if $\ell(x)$ is the length of the encoded symbol $e(x)$ in bits, then the number of channel uses to transmit this symbol is $\ell(x)$, whereby the service time of the n^{th} arriving symbol is given by $S_n = \ell(X_n)$. Since $\{X_n\}_{n=1}^{\infty}$ is iid and the encoder mapping e is fixed, the sequence $(S_n)_{n\in\mathbb{N}}$, too, is iid with common mean $\mathbb{E}[L]$. Therefore, the resulting queue is an M/G/1 queuing system with Poisson arrivals of rate λ and iid service times $(S_n)_{n\in\mathbb{N}}$. Note that this queue will be stable only if $\lambda\mathbb{E}[S_n] = \lambda\mathbb{E}[L] < 1$.

We are interested in designing prefix-free codes e that minimize the average waiting time defined as follows:

Definition 7.7.2. The *average waiting time* $D(e)$ of a source code e is given by

$$D(e) := \limsup_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}[D_n - A_n],$$

where the expectation is over source symbol realizations $\{X_n\}_{n=1}^{\infty}$ and arrival instants

$\{A_n\}_{n\in\mathbb{N}}$.

We seek prefix-free codes e with the least possible average waiting time $D(e)$. In fact, a closed-form expression for $D(e)$ was obtained in [48]. For clarity of exposition, we denote the load for the queuing system above for a fixed λ by $\rho(L):=\lambda\mathbb{E}[L]$. Since $\rho(L)<1$ for the queue to be stable, the average codeword length $\mathbb{E}[L]$ must be strictly less than a threshold $L_{\mathtt{th}}:=\frac{\mathbb{E}[L]}{\rho(L)}=\frac{1}{\lambda}$ for the queue to be stable.

Theorem 7.7.3 ([48]). *Consider a random variable X with pmf P and a source code e which assigns a bit sequence of length $\ell(x)$ to $x\in\mathcal{X}$. Let L denote the random variable $\ell(X)$. Then, the average waiting time $D(e)$ for e is given by*

$$D(e)=\begin{cases}\frac{\mathbb{E}\left[L^2\right]}{2(L_{\mathtt{th}}-\mathbb{E}[L])}+\mathbb{E}[L], & \mathbb{E}[L]<L_{\mathtt{th}},\\ \infty, & \mathbb{E}[L]\geq L_{\mathtt{th}}.\end{cases} \tag{7.14}$$

Thus, the problem of designing source codes with minimum average waiting time reduces to that of designing a prefix-free code that minimizes the cost in (7.14). This problem was first considered in [48]. In fact, it was noted in [48, Chapter 1, Section 3] that codes which minimize the first moment are robust for (7.14). We will justify this empirical observation in Corollary 7.7.5. However, optimal codes can differ from Shannon codes for P. Indeed, an algorithm for finding the optimal length assignments $\ell(x)$, $x\in\mathcal{X}$, for a prefix-free code that minimizes $\bar{D}(e)$ was presented in [56] and the optimal code can be seen to outperform Shannon codes for P. While this algorithm has complexity that is polynomial in the alphabet size, it is computationally expensive for large alphabet sizes – the case of interest for our problem.

Interestingly, the cost function in (7.14) resembles closely the expression we obtained for asymptotic average age and our recipe used to design minimum average age codes can be applied to design minimum average delay codes as well. The underlying optimization problem can be solved numerically rather quickly, much faster than the optimization in [56]. However, as before, our procedure can only handle the real-relaxation of the underlying optimization problem, and unlike the previous case, naive rounding-off to integer lengths

yields a sub-optimal solution when $(1-\rho(L))$ is small. Nonetheless, the minimum average waiting time computed using our recipe serves as an easily computable lower bound for the optimal $D(e)$. In fact, we observe in our numerical simulations that the resulting lower bound is rather close to the optimal cost obtained using [56].

Now, we describe the modification of our recipe to design codes with $\mathbb{E}[L] < L_{\mathtt{th}}$ that minimize the cost

$$\|L\|_1 + \frac{\|L\|_2^2}{2(L_{\mathtt{th}} - \|L\|_1)}, \tag{7.15}$$

where $L = \ell(X)$ for X with pmf P. As before, we first obtain a variational form of (7.15) which entails a linear function of lengths. Specifically, we have the following steps.

1. First, we obtain a polynomial form from the rational function:

$$\frac{\|L\|_2^2}{2(L_{\mathtt{th}} - \|L\|_1)} = \max_{z \geq 0} z\|L\|_2 - \frac{z^2}{2}(L_{\mathtt{th}} - \|L\|_1).$$

2. Then, Corollary 7.4.2 yields that the cost in (7.15) equals

$$\max_{z \geq 0} \max_{Q \ll P} \sum_{x \in \mathcal{X}} g_{z,Q,P}(x)\ell(x) - \frac{z^2}{2} L_{\mathtt{th}}$$

   where the $g_{z,Q,P}(x)$ is defined as

$$g_{z,Q,P}(x) := \left(1 + \frac{z^2}{2}\right) P(x) + z\sqrt{Q(x)P(x)}.$$

Thus, our goal reduces to identifying the minimizer $\ell^* \in \Lambda$ that achieves

$$\Delta^*(P) = \min_{\substack{\ell \in \Lambda, \\ \mathbb{E}[L] < L_{\mathtt{th}}}} \max_{z \geq 0} \max_{Q \ll P} \sum_{x \in \mathcal{X}} g_{z,Q,P}(x)\ell(x) - \frac{z^2}{2} L_{\mathtt{th}}. \tag{7.16}$$

The result below is the counterpart of Theorem 7.5.1 for minimum delay source codes and is proved in Section 7.8.3.

Theorem 7.7.4. *Under the condition*

$$H(X) + \log(1 + 1/\sqrt{2}) < L_{\mathtt{th}}, \tag{7.17}$$

the optimal minmax cost $\Delta^*(P)$ *in* (7.16) *satisfies*

$$\begin{aligned}\Delta^*(P) &= \max_{z\geq 0} \max_{Q \ll P} \min_{\substack{\ell \in \Lambda, \\ \mathbb{E}[L] < L_{\mathtt{th}}}} \sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\ell(x) - \frac{z^2}{2} L_{\mathtt{th}} \\ &= \max_{z\geq 0} \max_{Q \ll P} \sum_{x\in\mathcal{X}} g_{z,Q,P}(x) \log \frac{\sum_{x'\in\mathcal{X}} g_{z,Q,P}(x')}{g_{z,Q,P}(x)} \\ &\qquad - \frac{z^2}{2} L_{\mathtt{th}}. \end{aligned} \tag{7.18}$$

Furthermore, if (z^*, Q^*) *is the maximizer of the right-side of* (7.18)*, then the minmax cost* (7.16) *is achieved uniquely by Shannon lengths for pmf* P^* *on* $\mathcal{X}$ *given by*

$$P^*(x) = \frac{g_{z\cdot Q^*,P}(x)}{\sum_{x'\in\mathcal{X}} g_{z^*,Q^*,P}(x')}.$$

We remark that (7.14) implies that $H(X) < L_{\mathtt{th}}$ is essential for the existence of a prefix free source coding scheme with finite average delay. Thus, the condition $H(X) + \log(1 + 1/\sqrt{2}) < L_{\mathtt{th}}$ is a mild one.

Thus, as before, the optimal codeword lengths for the relaxed problem (allowing real-valued lengths) correspond, once again, to Shannon lengths for a titled distribution P^*. As remarked earlier, the performance of the optimal source code is known to be not too far from the Shannon code for P. This observation can be justified by the following simple corollary of Theorem 7.7.4.

Corollary 7.7.5. *The KL-Divergence between* P*,* P^* *is bounded as*

$$D(P||P^*) \leq \log\left(1 + \frac{1}{\sqrt{2}}\right).$$

Proof. The proof follows from (7.37), which is in turn derived in the proof of theorem

7.7.4 in section 7.8.3. □

Thus, the average length for Shannon codes and our codes do not differ by more than $\log(1+1/\sqrt{2})$ (*cf.* [18]). Indeed, we note in Figures 7.7a, 7.7b via numerical simulations that the optimal cost in (7.18) is very close to the performance of optimal codes designed using [56]. This suggests that possibly there is an appropriate rounding-off procedure for real-valued lengths that can yield integer lengths with close to optimal performance; devising such a rounding-off procedure is an interesting research direction for the future. We close this section by noting that analogous versions of Lemma 7.5.2 and Lemma 7.8.6 in the Appendix can be obtained for optimization problem (7.18).

7.8 Proofs

7.8.1 Proof of Theorem 7.3.2

We establish the expression for average age given in (7.12) for the more general class of randomized schemes; Theorem 7.3.2 will follow upon setting $\theta(x) = 1$, for all $x \in \mathcal{X}$. Recall that the symbol $\emptyset$ is available only in the extended model in Section 7.7, and not in the original model discussed in rest of the paper. Note that the formula for average age given in Theorem 7.3.2 is similar in form to the expressions for average age derived in other settings; see [51] for an example.

We will first set up some notation. Let $S_0 := 0$ and

$$S_k := \inf\{t > S_{k-1} : U(t) > U(t-1)\},\ k \in \mathbb{N}.$$

Namely, S_k is the time at which the decoder updates its estimate for the symbol for the kth time. Recall that $U(t)$ is incremented only on successful reception at the receiver and is strictly increasing in t. For brevity, we introduce the notation $Y_k := S_k - S_{k-1}$ for the time between the $(k-1)$th and the kth information update at the decoder. Further, denote by $Z_k := S_k - U(S_k)$ the age at time S_k, which is simply the time taken for the

successful reception of the symbol[9] $x \in \mathcal{X}$ transmitted at time $U(S_k)$. Also, denote by R_k the sum of instantaneous age between S_{k-1} and S_k (the kth reward), namely

$$R_k := \sum_{t=S_{k-1}+1}^{S_k} (t - U(t)).$$

Heuristically, our proof can be understood as follows. We note that the asymptotic average age is roughly

$$\frac{\sum_{k=1}^{\infty} R_k}{\lim_{k\to\infty} S_k}.$$

It is easy to see that $\{Y_k\}_{k=1}^{\infty}$ is an iid sequence. Thus, if $\{R_k\}_{k=1}^{\infty}$, too, was an iid sequence, we would obtain the asymptotic average age to be $\mathbb{E}[R_1]/\mathbb{E}[Y_1]$ by the standard Renewal Reward Theorem [81]. Unfortunately, this is not the case. But it turns out that the dependence in sequence $\{R_k\}$ is only between consecutive terms. Therefore, we can obtain the same conclusion as above by dividing the sum $\sum_{k=1}^{\infty} R_k$ into the sum of odd terms and even terms, each of which is in turn a sum of iid random variables.

We will now proceed to prove that dependence in R_k is between consecutive terms. Since $U(t)$ remains $U(S_{k-1})$ for all $t < S_k$, we get for $k \geq 1$ that

$$\begin{aligned} R_k &= \frac{(S_k - S_{k-1} - 1)(S_k - S_{k-1})}{2} \\ &\qquad + (S_k - S_{k-1} - 1) \cdot (S_{k-1} - U(S_{k-1})) \\ &\qquad\qquad + S_k - U(S_k) \\ &= \frac{1}{2}Y_k^2 + Y_k\left(Z_{k-1} - \frac{1}{2}\right) + Z_k - Z_{k-1}, \end{aligned} \tag{7.19}$$

with Z_0 set to 0.

Note that since the source sequence $\{X_n\}$ is iid and the randomization θ is stationary, the sequences Y_k and Z_k are iid, too. Therefore, the $(R_{2n})_{n\in\mathbb{N}}$ and $(R_{2n+1})_{n\in\mathbb{N}}$ are both[10] iid sequences with $\mathbb{E}[R_{2n}] = \mathbb{E}[R_{2n+1}] = \mathbb{E}[R_2]$ for all n.

[9] This must be a symbol in $\mathcal{X}$ and not $\emptyset$ by the definition of S_k.
[10] The initial term R_1 has a different distribution since $Z_0 = 0$.

Using this observation, we can obtain the following expression for the average age:

$$\bar{A}(e,\theta) = \frac{\mathbb{E}[R_2]}{\mathbb{E}[Y_1]}. \tag{7.20}$$

Before we prove (7.20), which is the main ingredient of our proof, we evaluate the expression on the right-side.

For $\mathbb{E}[Y_1]$, note that Y_1 gets incremented by $\ell(\emptyset)$ each time $\emptyset$ is sent, and gets incremented finally by $\ell(x)$ once a symbol $x \in \mathcal{X}$ is sent. Thus, Y_1 takes the value $\ell(x) + r\ell(\emptyset)$ with probability $(1 - \mathbb{E}[\theta(X)])^r \theta(x) P(x)$. Denoting $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$, we get

$$\begin{aligned}
\mathbb{E}[Y_1] &= \sum_{x\in\mathcal{X}} \sum_{r\in\mathbb{N}_0} (\ell(x) + r\ell(\phi)) P(x)\theta(x)(1 - \mathbb{E}[\theta(X)])^r \\
&= \sum_{x\in\mathcal{X}} \sum_{r\in\mathbb{N}_0} \ell(x) P(x)\theta(x)(1 - \mathbb{E}[\theta(X)])^r \\
&\qquad + \sum_{x\in\mathcal{X}} \sum_{r\in\mathbb{N}_0} r\ell(\phi) P(x)\theta(x)(1 - \mathbb{E}[\theta(X)])^r \\
&= \frac{\sum_{x\in\mathcal{X}} \ell(x) P(x)\theta(x)}{\mathbb{E}[\theta(X)]} + \frac{\ell(\phi)(1 - \mathbb{E}[\theta(X)])}{\mathbb{E}[\theta(X)]} \\
&= \frac{\mathbb{E}[L(\theta)]}{\mathbb{E}[\theta(X)]}.
\end{aligned}$$

For $\mathbb{E}[R_2]$, it follows from (7.19) that

$$\mathbb{E}[R_2] = \frac{1}{2}\mathbb{E}\left[Y_2^2\right] + \mathbb{E}[Y_2 Z_1] - \frac{1}{2}\mathbb{E}[Y_2],$$

since $\mathbb{E}[Z_2] = \mathbb{E}[Z_1]$. Also, note that Z_1 only depends on the symbol $x \in \mathcal{X}$ received at time S_1 which in turn can depend only on the symbols X_n for $n \leq S_1 - 1$. On the other hand, $Y_2 = S_2 - S_1$ depends on symbols X_n for $n \geq S_1$ and the outputs of the independent coin tosses corresponding to randomization θ. Therefore, Z_1 is independent of Y_2, whereby

$$\mathbb{E}[R_2] = \frac{1}{2}\mathbb{E}\left[Y_2^2\right] + \mathbb{E}[Y_2]\left(\mathbb{E}[Z_1] - \frac{1}{2}\right).$$

Next, note that Z_1 takes the value $\ell(x)$, $x \in \mathcal{X}$, when the symbol received at S_1 is x. This

latter event happens with probability

$$\sum_{r=0}^{\infty}(1-\mathbb{E}\left[\theta(X)\right])^{r}\theta(x)P(x)=\frac{\theta(x)P(x)}{\mathbb{E}\left[\theta(X)\right]},$$

and so, by the definition of $L(\theta)$ in (7.13),

$$\begin{aligned}\mathbb{E}\left[Z_1\right]&=\frac{\sum_x \ell(x)\theta(x)P(x)}{\mathbb{E}\left[\theta(X)\right]}\\&=\frac{\mathbb{E}\left[L(\theta)\right]}{\mathbb{E}\left[\theta(X)\right]}-\frac{\ell(\emptyset)(1-\mathbb{E}\left[\theta(X)\right])}{\mathbb{E}\left[\theta(X)\right]}.\end{aligned}$$

Then by denoting $p_\emptyset=1-\mathbb{E}\left[\theta(X)\right]$, the second moment $\mathbb{E}\left[Y_1^2\right]$ can be computed by observing the following recursion:

$$\begin{aligned}&\mathbb{E}\left[Y_1^2\right]\\&=\sum_{x\in\mathcal{X}}\sum_{r\in\mathbb{N}_0}(\ell(x)+r\ell(\emptyset))^2P(x)\theta(x)p_\emptyset^r\\&=\sum_{x\in\mathcal{X}}\ell(x)^2P(x)\theta(x)\\&\qquad+p_\emptyset\sum_{x\in\mathcal{X}}\sum_{r\in\mathbb{N}}(\ell(x)+r\ell(\emptyset))^2P(x)\theta(x)p_\emptyset^{r-1}\\&=\sum_{x\in\mathcal{X}}\ell(x)^2P(x)\theta(x)\\&\qquad+p_\emptyset\sum_{x\in\mathcal{X}}\sum_{r\in\mathbb{N}}\Big(\ell(x)+(r-1)\ell(\emptyset)\Big)^2P(x)\theta(x)p_\emptyset^{r-1}\\&\qquad+2\ell(\emptyset)p_\emptyset\sum_{x\in\mathcal{X}}\sum_{r\in\mathbb{N}}\Big(\ell(x)+(r-1)\ell(\emptyset)\Big)P(x)\theta(x)p_\emptyset^{r-1}\\&\qquad+p_\emptyset\sum_{x\in\mathcal{X}}\sum_{r\in\mathbb{N}}\ell(\emptyset)^2P(x)\theta(x)p_\emptyset^{r-1}\\&=\sum_{x\in\mathcal{X}}\ell(x)^2P(x)\theta(x)\\&\qquad+p_\emptyset\mathbb{E}\left[Y_1^2\right]+2\ell(\emptyset)(1-\mathbb{E}\left[\theta(X)\right])\mathbb{E}\left[Y_1\right]+\ell(\emptyset)^2p_\emptyset,\end{aligned}$$

which upon rearrangement yields

$$\mathbb{E}\left[Y_1^2\right]=\frac{\mathbb{E}\left[L(\theta)^2\right]}{\mathbb{E}\left[\theta(X)\right]}+2\mathbb{E}\left[Y_1\right]\cdot\frac{\ell(\emptyset)p_\emptyset}{\mathbb{E}\left[\theta(X)\right]}.$$

Upon combining the relations derived above, we get

$$\frac{\mathbb{E}\left[R_2\right]}{\mathbb{E}\left[Y_1\right]} = \frac{\mathbb{E}\left[L(\theta)^2\right]}{2\mathbb{E}\left[L(\theta)\right]} + \frac{\mathbb{E}\left[L(\theta)\right]}{\mathbb{E}\left[\theta(X)\right]} - \frac{1}{2},$$

which with (7.20) completes the proof.

It remains to establish (7.20). The proof is a simple extension of the renewal reward theorem to our sequence of rewards R_n in which adjacent terms may be dependent. We include it here for completeness. Note that $(Y_n)_{n\in\mathbb{N}}$ is a sequence of non-negative iid random variables with mean $\mathbb{E}\left[Y_1\right]$, and $S_n = \sum_{k=1}^{n} Y_k$ for all $n \in \mathbb{N}$. The sequence $\{S_n\}$ serves as a sequence of renewal times and R_n denotes the reward accumulated in the nth renewal interval (though not in the standard iid sense). Define $N(t)$ to be the number of receptions up to time $t > 0$, *i.e.*,

$$N(t) = \sup\left\{n : S_n \leq t\right\},$$

and $R(t)$ to be the cumulative reward accumulated till time t, *i.e.*,

$$R(t) = \sum_{k=1}^{N(t)} R_k.$$

With this notation, we have

$$\frac{R(t)}{t} = \frac{\sum_{k=1}^{N(t)} R_k}{t} \tag{7.21}$$

$$= \frac{\sum_{k=1}^{N(t)} R_k}{N(t)} . \frac{N(t)}{t}. \tag{7.22}$$

Note that

$$\frac{\sum_{k=1}^{\lfloor \frac{N(t)}{2} \rfloor} \sum_{i\in\{0,1\}} R_{2k+i}}{N(t)} \leq \frac{\sum_{k=2}^{N(t)} R_k}{N(t)} \leq \frac{\sum_{k=1}^{\lceil \frac{N(t)}{2} \rceil} \sum_{i\in\{0,1\}} R_{2k+i}}{N(t)}.$$

We now analyze the two bounds in the previous set of inequalities. Since $\mathbb{E}[Y_1]$ is finite, we get (see [81] for a proof)

$$\lim_{t\to\infty} \frac{N(t)}{t} \to \frac{1}{\mathbb{E}[Y_1]} \quad a.s., \tag{7.23}$$

which also shows that $N(t) \to \infty \quad a.s.$ as $t \to \infty$. Therefore, for $i \in \{0,1\}$,

$$\frac{\sum_{k=1}^{\left\lceil \frac{N(t)}{2} \right\rceil} R_{2k+i}}{N(t)} = \frac{\sum_{k=1}^{\left\lceil \frac{N(t)}{2} \right\rceil} R_{2k+i}}{\left\lceil \frac{N(t)}{2} \right\rceil} \cdot \frac{\left\lceil \frac{N(t)}{2} \right\rceil}{N(t)}.$$

Since $(R_{2k+i})_{k\in\mathbb{N}}$ is iid and $N(t) \to \infty \quad a.s.$ as $t \to \infty$, strong law of large numbers yields

$$\lim_{t\to\infty} \frac{\sum_{k=1}^{\left\lceil \frac{N(t)}{2} \right\rceil} R_{2k+i}}{\left\lceil \frac{N(t)}{2} \right\rceil} = \mathbb{E}[R_2] \quad a.s. \quad \forall i \in \{0,1\},$$

which further gives

$$\lim_{t\to\infty} \frac{\sum_{k=1}^{\left\lceil \frac{N(t)}{2} \right\rceil} \sum_{i\in\{0,1\}} R_{2k+i}}{N(t)} = \mathbb{E}[R_2] \quad a.s..$$

Similarly,

$$\lim_{t\to\infty} \frac{\sum_{k=1}^{\left\lfloor \frac{N(t)}{2} \right\rfloor} \sum_{i\in\{0,1\}} R_{2k+i}}{N(t)} = \mathbb{E}[R_2] \quad a.s..$$

Combining the observations above, we get

$$\lim_{t\to\infty} \frac{\sum_{k=1}^{N(t)} R_k}{N(t)} = \mathbb{E}[R_2] \quad a.s.,$$

which together with (7.22) and (7.23) yields (7.20).

7.8.2 Proof of Theorem 7.5.1

Our proof is based on noticing that the minmax cost $\Delta^*(P)$ in (7.8) involves weighted average length with weights $g_{z,Q,P}(x)$. In fact, we will see below that there is no loss in restricting to nonnegative weights, whereby our cost has a form of average length with respect to a distribution that depends on (z, Q). The broad idea of the proof is to establish that a optimal code corresponding to the *least favorable* choice of (z, Q) is minmax optimal. However, the proof is technical since our cost function may not satisfy the assumptions in a standard saddle-point theorem.

A simpler form of the minmax cost $\Delta^*(P)$ from (7.6) is given by

$$\Delta^*(P) = \min_{\ell \in \Lambda} \max_{z \geq 0} f(\ell, z), \tag{7.24}$$

where

$$f(\ell, z) := -z^2 \frac{\mathbb{E}[L]}{2} + z\sqrt{\mathbb{E}[L^2]} + \mathbb{E}[L]. \tag{7.25}$$

We seek to apply the following version of Sion's minmax theorem to the function f.

Theorem 7.8.1 (Sion's Minmax Theorem [84])**.** *Let $\mathcal{X}$ be convex space and $\mathcal{Y}$ be a convex, compact space. Let h be a function on $\mathcal{X} \times \mathcal{Y}$ which is convex on $\mathcal{X}$ for every fixed y in $\mathcal{Y}$ and concave on $\mathcal{Y}$ for every fixed x in $\mathcal{X}$. Then,*

$$\inf_{x \in \mathcal{X}} \sup_{y \in \mathcal{Y}} h(x, y) = \sup_{y \in \mathcal{Y}} \inf_{x \in \mathcal{X}} h(x, y).$$

Indeed, the following lemma shows that our function f satisfies the convexity requirements of Sion's minmax theorem.

Lemma 7.8.2. *$f(\ell, z)$ is convex in ℓ for every fixed $z \geq 0$ and concave in z for a fixed $\ell \in \Lambda$.*

Proof. To show that $f(\ell, z)$ is a convex function of ℓ for every fixed $z \geq 0$, it suffices to show that $\sqrt{\mathbb{E}[L^2]}$ is convex in $L = \ell(X)$. To that end, let $L_1 = \ell_1(X)$ and $L_2 = \ell_2(X)$,

for some ℓ_1 and ℓ_2 in λ. For all $\lambda \in [0,1]$,

$$\sqrt{\mathbb{E}\left[(\lambda L_1 + (1-\lambda)L_2)^2\right]} \leq \lambda\sqrt{\mathbb{E}\left[L_1^2\right]} + (1-\lambda)\sqrt{\mathbb{E}\left[L_2^2\right]},$$

where the inequality is by Minkowski inequality for $\|L\|_2$.

The concavity in z can be seen easily by noticing that $\frac{\partial^2 f(\ell,z)}{\partial z^2} \leq 0$ for all ℓ in λ. □

However, our underlying domains of optimization are not compact. Our proof below circumvents this difficulty by showing that we may replace one of the domains by a compact set. For ease of reading, we divide the proof into 3 steps; we begin by summarize the flow at a high-level. The first step is to show that this minmax cost remains unchanged when the domain of z is restricted to a bounded interval $[0,K]$ for a sufficiently large K. This will allow us to interchange $\min_{l\in\Lambda}$ and $\max_{z\in[0,K]}$ in the second step by using Theorem 7.8.1 to obtain

$$\Delta^*(P) = \max_{z\in[0,K]} \min_{\ell\in\Lambda} f(\ell, z). \tag{7.26}$$

Furthermore, we then use Corollary 7.4.2 to linearize the cost. But this brings in the maximization over an additional parameter Q, which we again interchange with the minimum over ℓ using Sion's minmax theorem (Theorem 7.8.1). Note that the required convexity of the cost function is easy to see; we note it in the following lemma.

Lemma 7.8.3. *For every fixed $z \geq 0$, $\sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\ell(x)$ is convex in ℓ for a fixed $Q \ll P$ and concave in Q for a fixed $\ell \in \Lambda$.*

Proof. For every fixed $z \geq 0$, the cost function $\sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\ell(x)$ is linear, and thereby convex, in ℓ for a fixed Q . For concavity in Q, note that for a fixed $\ell \in \Lambda$, the function $\sqrt{Q(x)}$ is a concave function of $Q(x)$, for all x in $\mathcal{X}$. □

Thus, we obtain

$$\Delta^*(P) = \max_{z\in[0,K], Q\ll P} \min_{\ell\in\Lambda} \sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\ell(x).$$

In the final step, we will establish that the optimal code for linear cost with weights corresponding to the least favorable (z, Q) is minmax optimal. We now present each step in detail.

Step 1 We begin by noting that there is no loss in restricting to codes with[11] $\mathbb{E}[L] \leq \log|\mathcal{X}|$. Indeed, note that for $\mathbb{E}[L] > \log|\mathcal{X}|$ the average age is bounded as

$$\mathbb{E}[L] + \frac{\mathbb{E}[L^2]}{2\mathbb{E}[L]} \geq \frac{3}{2}\mathbb{E}[L] > \frac{3}{2}\log|\mathcal{X}|, \tag{7.27}$$

where we have used Jensen's inequality. On the other hand, a fixed-length code with $\ell(x) = \log|\mathcal{X}|$ attains

$$\mathbb{E}[L] + \frac{\mathbb{E}[L^2]}{2\mathbb{E}[L]} = \frac{3}{2}\log|\mathcal{X}|, \tag{7.28}$$

which gives the desired form

$$\begin{aligned}\Delta^*(P) &= \min_{\ell\in\Lambda, \mathbb{E}[L]\leq\log\mathcal{X}} \mathbb{E}[L] + \frac{\mathbb{E}[L^2]}{2\mathbb{E}[L]} \\ &= \min_{\ell\in\Lambda, \mathbb{E}[L]\leq\log\mathcal{X}} \max_{z\in\mathbb{R}} f(\ell, z),\end{aligned} \tag{7.29}$$

where $f(\ell, z)$ is defined in (7.25). Also, for a fixed ℓ in Λ the function $f(\ell, z)$ attains its maximum at $z^*(\ell)$ given by

$$z^*(\ell) := \frac{\sqrt{\mathbb{E}[L^2]}}{\mathbb{E}[L]}.$$

For $\mathbb{E}[L] \leq \log|\mathcal{X}|$, the maximizer $z^*(\ell)$ is bounded as[12]

$$\begin{aligned}z^*(\ell) &\leq \frac{\sqrt{\mathbb{E}[L^2]}}{H(X)} \\ &= \frac{\sqrt{\sum_x P(x)\ell(x)^2}}{H(X)}\end{aligned}$$

[11] For simplicity, we assume that $\log\mathcal{X}$ is an integer.
[12] We assume without loss of generality that $P(x) > 0$ for every $x \in \mathcal{X}$.

$$\leq \frac{\mathbb{E}[L]}{H(X)}\sqrt{\max_{x\in\mathcal{X}}\frac{1}{P(x)}}$$
$$\leq \frac{\log|\mathcal{X}|}{H(X)}\sqrt{\frac{1}{\min_{x\in\mathcal{X}}P(x)}},$$

where the first inequality uses $\mathbb{E}[L] \geq H(X)$, which holds for every prefix-free code, and the second holds since $\|a\|_2 \leq \|a\|_1$ for any sequence $a = (a_1, ..., a_n)$. Denoting

$$K := \frac{\log|\mathcal{X}|}{H(X)}\sqrt{\frac{1}{\min_{x\in\mathcal{X}}P(x)}},$$

(7.29) yields

$$\Delta^*(P) = \min_{\ell\in\Lambda,\mathbb{E}[L]\leq\log|\mathcal{X}|}\max_{z\in[0,K]} f(\ell, z).$$

Next, we show that the minmax cost above remains unchanged when we drop the constraint $\mathbb{E}[L] \leq \log|\mathcal{X}|$ in the outer minimum, which will complete the first step of the proof and establish (7.26). Indeed, since by (7.28) the minimum over $\ell \in \Lambda$ such that $\mathbb{E}[L] \leq \log|\mathcal{X}|$ is at most $(3/2)\log|\mathcal{X}|$, it suffices to show that

$$\min_{\ell\in\Lambda,\mathbb{E}[L]>\log|\mathcal{X}|}\max_{z\in[0,K]} f(\ell, z) > \frac{3}{2}\log|\mathcal{X}|. \tag{7.30}$$

We divide the proof of this fact into two cases. First consider the case when ℓ in Λ is such that $\mathbb{E}[L] > \log|\mathcal{X}|$ and $K \geq z^*(\ell)$. Then, $\max_{z\in[0,K]} f(\ell, z)$ equals $\max_{z\geq 0} f(\ell, z)$, which is bounded below by $(3/2)\log|\mathcal{X}|$ using (7.27) and the definition of $f(\ell, z)$. For the second case when $\mathbb{E}[L] > \log|\mathcal{X}|$ and $K < z^*(\ell)$, we have

$$\max_{z\in[0,K]} f(\ell, z) = -K^2\frac{\mathbb{E}[L]}{2} + K\sqrt{\mathbb{E}[L^2]} + \mathbb{E}[L]$$
$$> K^2\frac{\mathbb{E}[L]}{2} + \mathbb{E}[L]$$
$$> \frac{3}{2}\cdot\mathbb{E}[L]$$
$$> \frac{3}{2}\cdot\log|\mathcal{X}|,$$

where the first inequality uses $K < z^*(\ell) = \sqrt{\mathbb{E}[L^2]}/\mathbb{E}[L]$ and the second holds since $K \geq 1$ from its definition. Therefore, we have established (7.30), and so we have

$$\Delta^*(P) = \min_{\ell \in \Lambda, \mathbb{E}[L] \leq \log|\mathcal{X}|} \max_{z \in [0,K]} f(\ell, z) = \min_{\ell \in \Lambda} \max_{z \in [0,K]} f(\ell, z).$$

Step 2 By lemma 7.8.2 , $f(\ell, z)$ is convex in ℓ for every fixed $z \geq 0$ and concave in z for a fixed $\ell \in \Lambda$, z takes values in a convex compact set $[0, K]$, and the set $\{\ell : \ell \in \Lambda\}$ is convex, we get from Sion's minmax theorem (Theorem 7.8.1) that

$$\Delta^*(P) = \min_{\ell \in \Lambda} \max_{z \in [0,K]} f(\ell, z) = \max_{z \in [0,K]} \min_{\ell \in \Lambda} f(\ell, z).$$

Using Corollary 7.4.2, we have

$$\|L\|_2 = \max_{Q \ll P} \sum_{x \in \mathcal{X}} Q(x)^{\frac{1}{2}} P(x)^{\frac{1}{2}} \ell(x),$$

which by the definition of f in (7.25) further yields

$$f(\ell, z) = \max_{Q \ll P} \sum_{x \in \mathcal{X}} g_{z,Q,P}(x) \ell(x), \tag{7.31}$$

where

$$g_{z,Q,P}(x) = \left(1 - \frac{z^2}{2}\right) P(x) + z\sqrt{Q(x)P(x)}.$$

We have obtained

$$\Delta^*(P) = \max_{z \in [0,K]} \min_{\ell \in \Lambda} \max_{Q \ll P} \sum_{x \in \mathcal{X}} g_{z,Q,P}(x) \ell(x). \tag{7.32}$$

From Lemma 7.8.3, $\sum_{x \in \mathcal{X}} g_{z,Q,P}(x)\ell(x)$ is convex in ℓ , for all $Q \ll P$, and concave in Q, for a fixed $\ell \in \Lambda$. Furthermore, since the set $\{Q : Q \ll P\}$ is convex compact for a pmf P

on finite alphabet, using Sion's minmax theorem (Theorem 7.8.1) once again, we get

$$\Delta^*(P) = \max_{z\in[0,K]} \max_{Q\ll P} \min_{\ell\in\Lambda} \sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\ell(x), \tag{7.33}$$

which completes our second step.

Step 3 By (7.33), we get

$$\Delta^*(P) \leq \max_{z\geq 0} \max_{Q\ll P} \min_{\ell\in\Lambda} \sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\ell(x).$$

On the other hand, by (7.24) and (7.31) we have

$$\begin{aligned}\Delta^*(P) &= \min_{\ell\in\Lambda} \max_{z\geq 0} \max_{Q\ll P} \sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\ell(x) \\ &\geq \max_{z\geq 0} \max_{Q\ll P} \min_{\ell\in\Lambda} \sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\ell(x),\end{aligned}$$

whereby

$$\begin{aligned}\Delta^*(P) &= \min_{\ell\in\Lambda} \max_{z\geq 0} \max_{Q\ll P} \sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\ell(x) \\ &= \max_{z\geq 0} \max_{Q\ll P} \min_{\ell\in\Lambda} \sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\ell(x),\end{aligned} \tag{7.34}$$

which proves the first part of theorem 7.5.1.

Next, we claim that in the maxmin formula above, the maximum is attained by a (z, Q) for which $g_{z,Q,P}(x)$ is non-negative for every x. Indeed, if for some z, Q there exists an x' in $\mathcal{X}$ such that $g_{z,Q,P}(x')$ is negative, then the cost $\sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\ell(x)$ is minimized by any ℓ such that $\ell(x') = \infty$ and the minimum value is $-\infty$. Such z, Q clearly can't be the optimizer of the maxmin problem, since for $z = 0$, we have $g_{z,Q,P} \geq 0$, which in turn leads to $\min_{\ell\in\Lambda} \sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\ell(x) \geq 0$.

Finally, consider (z, Q) such that $g_{z,Q,P}(x) \geq 0$ for all $x \in \mathcal{X}$. For such a (z, Q), we

seek to identify the minimized ℓ below:

$$\begin{aligned}&\min_{\ell\in\Lambda}\sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\ell(x)\\&=\sum_{x'\in\mathcal{X}} g_{z,Q,P}(x')\min_{\ell\in\Lambda}\sum_{x\in\mathcal{X}}\frac{g_{z,Q,P}(x)}{\sum_{x'\in\mathcal{X}} g_{z,Q,P}(x')}\ell(x).\end{aligned}\tag{7.35}$$

Thus, our optimization problem reduces to the standard problem of designing minimum average length prefix-free codes for the pmf

$$P_{z,Q}(x)=\frac{g_{z,Q,P}(x)}{\sum_{x'\in\mathcal{X}} g_{z,Q,P}(x')}.$$

By Shannon's source coding theorem for variable length codes, the minimum is achieved by

$$\ell^*_{z,Q}(x):=\log\frac{\sum_{x'\in\mathcal{X}} g_{z,Q,P}(x)}{g_{z,Q,P}(x)}.$$

Furthermore, $\ell^*_{z,Q}$ is the unique minimizer in Λ.

Consider now a maximizer (z^*,Q^*) of the maxmin problem in (7.34), and let $\ell^o=\ell^*_{z^*,Q^*}$. Then, by Lemma 7.8.5 in the appendix,$(\ell^o,(z^*,Q^*))$ is a saddle-point for the minmax problem in (7.34). Moreover, ℓ^o is the unique minmax optimal solution.

7.8.3 Proof of Theorem 7.7.4

Denoting

$$f(\ell,z)=-z^2\frac{(L_{\mathtt{th}}-\mathbb{E}\left[L\right])}{2}+z\sqrt{\mathbb{E}\left[L^2\right]}+\mathbb{E}\left[L\right],\tag{7.36}$$

the optimal cost $\Delta^*(P)$ can be written as

$$\begin{aligned}\Delta^*(P)&=\inf_{\ell\in\Lambda,\mathbb{E}[L]<L_{\mathtt{th}}}\frac{\mathbb{E}\left[L^2\right]}{2(L_{\mathtt{th}}-\mathbb{E}\left[L\right])}+\mathbb{E}\left[L\right]\\&=\min_{\ell\in\Lambda,\mathbb{E}[L]<L_{\mathtt{th}}}\max_{z\geq 0} f(\ell,z).\end{aligned}$$

This form is similar to the one we had in Theorem 7.5.1. But the proof there does not extend to the case at hand. Specifically, note that for each ℓ, $f(\ell, z)$ attains its maximum value for $z^*(\ell) = \frac{\sqrt{\mathbb{E}[L^2]}}{(L_{\texttt{th}} - \mathbb{E}[L])}$ which, unlike the quantity that we obtained in the proof of Theorem 7.5.1, is unbounded over the set of $\ell \in \Lambda$ such that $\mathbb{E}[L] \leq L_{\texttt{th}}$. However, under the additional assumption $H(X) + \log(1 + 1/\sqrt{2}) < L_{\texttt{th}}$, we can provide a simpler alternative proof. We rely on the following lemma.

Lemma 7.8.4. *Consider a function $h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ such that the set $\mathcal{X}$ is compact convex, the set $\mathcal{Y}$ is convex, $h(x, y)$ is a convex function of x for every fixed y and a concave function of y for every fixed x. Suppose additionally that there exist a convex subset $\mathcal{X}_0$ of $\mathcal{X}$ and a compact convex subset $\mathcal{Y}_0$ of $\mathcal{Y}$ such that*

1. *for every for every $x \in \mathcal{X}_0$, an optimizer $y^*(x) \in \arg\max_{y \in \mathcal{Y}} h(x, y)$ belongs to $\mathcal{Y}_0$; and*

2. *for every $y \in \mathcal{Y}_0$, an optimizer $x^*(y) \in \arg\min_{x \in \mathcal{X}} h(x, y)$ belongs to $\mathcal{X}_0$.*

Then,

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} h(x, y) = \max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} h(x, y).$$

Proof. Note that since for x in $\mathcal{X}_0$, the y that maximizes $h(x, y)$ over $\mathcal{Y}$ is in $\mathcal{Y}_0$, we get

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} h(x, y) \leq \min_{x \in \mathcal{X}_0} \max_{y \in \mathcal{Y}} h(x, y) = \min_{x \in \mathcal{X}_0} \max_{y \in \mathcal{Y}_0} h(x, y).$$

Further, by Sion's minmax theorem (Theorem 7.8.1), the right-side equals $\max_{y \in \mathcal{Y}_0} \min_{x \in \mathcal{X}_0} h(x, y)$. But by our second assumption, the restriction $x \in \mathcal{X}_0$ can be dropped, and we have

$$\max_{y \in \mathcal{Y}_0} \min_{x \in \mathcal{X}_0} h(x, y) = \max_{y \in \mathcal{Y}_0} \min_{x \in \mathcal{X}} h(x, y) \leq \max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} h(x, y).$$

Thus, we have shown $\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} h(x, y) \leq \max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} h(x, y)$, which completes the proof since the inequality in the other direction holds as well. □

For our minmax cost, we will verify that both the conditions of the lemma above hold under the assumption $H(X) + \log(1 + 1/\sqrt{2}) < L_{\texttt{th}}$. Indeed, first note that for any

fixed $\ell \in \Lambda$ with $\mathbb{E}[L] \leq H(X) + \log(1 + 1/\sqrt{2})$, the maximizer z of $f(\ell, z)$ given by $\sqrt{\mathbb{E}[L^2]}/(L_{\mathtt{th}} - \mathbb{E}[L])$ satisfies

$$\begin{aligned}
&\frac{\sqrt{\mathbb{E}[L^2]}}{L_{\mathtt{th}} - \mathbb{E}[L]} \\
&\leq \sqrt{\frac{1}{\min_x P(x)}} \cdot \frac{\mathbb{E}[L]}{L_{\mathtt{th}} - \mathbb{E}[L]} \\
&\leq \sqrt{\frac{1}{\min_x P(x)}} \cdot \frac{H(X) + \log(1 + 1/\sqrt{2})}{L_{\mathtt{th}} - H(X) - \log(1 + 1/\sqrt{2})}.
\end{aligned}$$

Denote the right-side above by K and $L'_{\mathtt{th}} = H(X) + \log(1 + 1/\sqrt{2})$. Therefore, with the set $\{\ell \in \Lambda, \mathbb{E}[L] \leq L'_{\mathtt{th}}\}$ in the role of $\mathcal{X}_0$ in Lemma 7.8.4, the set $[0, K]$ can play the role of $\mathcal{Y}_0$.

To apply Lemma 7.8.4, we require two conditions to hold: first, that $f(l, z)$ is a convex function of ℓ for every fixed z and a concave function of z for every fixed ℓ, second, that for every $z \in [0, K]$, the minimizing ℓ satisfies $\mathbb{E}[L] \leq L'_{\mathtt{th}}$.The first easily follows from (7.36). The proof of this fact is exactly the same as Lemma 7.8.2. However, while the second condition can be shown to be true, the proof of this fact is almost the same as the proof of our theorem. For simplicity of presentation, we instead present an alternative proof of the theorem that uses a slight extension of the lemma above. Note that from our foregoing discussion and following the proof of the lemma, we already have obtained

$$\Delta^*(P) \leq \max_{z \in [0,K]} \min_{\ell \in \Lambda, \mathbb{E}[L] \leq L'_{\mathtt{th}}} f(\ell, z).$$

By using Corollary 7.4.2 and using Sion's minmax theorem once again, we get

$$\begin{aligned}
&\Delta^*(P) \\
&\leq \max_{z \in [0,K]} \max_{Q \ll P} \min_{\ell \in \Lambda, \mathbb{E}[L] \leq L'_{\mathtt{th}}} \sum_{x \in \mathcal{X}} g_{z,Q,P}(x)\ell(x) - \frac{z^2}{2} L_{\mathtt{th}},
\end{aligned}$$

where

$$g_{z,Q,P}(x) := \left(1 + \frac{z^2}{2}\right) P(x) + z\sqrt{Q(x)P(x)}.$$

In the preceding argument, we can use Sion's minmax theorem as the following two conditions hold. First, for every fixed $z \geq 0$, the function $\sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\ell(x) - \frac{z^2}{2}L_{\texttt{th}}$ is concave in Q for a fixed $\ell \in \Lambda$ and convex in ℓ for a fixed $Q \ll P$. Second, the sets $\{Q : Q \ll P\}$ and $\{\ell \in \Lambda : \mathbb{E}[L] \leq L'_{\texttt{th}}\}$ are compact and convex. Proof of the first is exactly the same as that of 7.8.3. Second is true as we have restricted to a finite alphabet $\mathcal{X}$. Thus, we can proceed as in the proof of the lemma, but we need to show now that for every $z \in [0, K]$ and $Q \ll P$, the optimal $\ell^*(z, Q)$ satisfies $\mathbb{E}[L^*] \leq L'_{\texttt{th}}$. Indeed, consider the following optimization problem for a fixed z, Q:

$$\begin{aligned}
&\min_{\ell\in\Lambda} \sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\ell(x) \\
&\qquad = \left(\sum_{x'\in\mathcal{X}} g_{z,Q,P}(x')\right) \min_{\ell\in\Lambda} \sum_{x\in\mathcal{X}} \frac{g_{z,Q,P}(x)}{\sum_{x'\in\mathcal{X}} g_{z,Q,P}(x')}\ell(x).
\end{aligned}$$

Since $\frac{g_{z,Q,P}(x)}{\sum_{x'\in\mathcal{X}} g_{z,Q,P}(x')}$ are nonnegative and add to 1, in the optimization problem above, we are minimizing the expected prefix free lengths for a finite alphabet for a particular distribution. Thus, by Shannon's Source Coding Theorem, the optimal $\ell^*_{z,Q}$ is given by

$$\ell^*_{z,Q}(x) := \log \frac{\sum_{x'\in\mathcal{X}} g_{z,Q,P}(x')}{g_{z,Q,P}(x)};$$

in fact, this optimizer is unique. But then for every x in $\mathcal{X}$,

$$\begin{aligned}
&\ell^*_{z,Q}(x) \\
&= \log \frac{\sum_{x'\in\mathcal{X}} g_{z,Q,P}(x')}{g_{z,Q,P(x)}} \\
&= \log \frac{\sum_{x'\in\mathcal{X}} \left(1 + \frac{z^2}{2}\right) P(x) + \sum_{x\in\mathcal{X}} z\sqrt{Q(x)P(x)}}{\left(1 + \frac{z^2}{2}\right) P(x) + z\sqrt{Q(x)P(x)}} \\
&\leq \log \frac{1}{P(x)}
\end{aligned}$$

$$
\begin{aligned}
&+ \log \left(\frac{\left(1+\frac{z^2}{2}\right)}{\left(1+\frac{z^2}{2}\right) + z\sqrt{\frac{Q(x)}{P(x)}}} + \frac{z}{\left(1+\frac{z^2}{2}\right) + z\sqrt{\frac{Q(x)}{P(x)}}} \right) \\
&\leq \log \frac{1}{P(x)} + \log \left(\frac{\left(1+\frac{z^2}{2}\right)}{\left(1+\frac{z^2}{2}\right)} + \frac{z}{\left(1+\frac{z^2}{2}\right)} \right) \\
&\leq \log \frac{1}{P(x)} + \log \left(1 + \frac{1}{\sqrt{2}} \right),
\end{aligned}
$$

where the first inequality is by the Cauchy-Schwarz inequality, the second inequality follows upon noting that $\frac{Q(x)}{P(x)}$ is nonnegative, and the last inequality follows from the fact that $z^2/2 + 1 \geq \sqrt{2}z$ (which holds with equality at $z = \sqrt{2}$). Thus as a consequence of this inequality the expected code length of such a code is upper bounded as follows,

$$
\mathbb{E}\left[L^*_{z,Q}\right] \leq H(x) + \log \left(1 + \frac{1}{\sqrt{2}} \right), \tag{7.37}
$$

which in the manner of Lemma 7.8.4 gives

$$
\Delta^*(P) = \max_{z \geq 0} \max_{Q \ll P} \min_{\ell \in \Lambda, \mathbb{E}[L] \leq L_{\mathtt{th}}} \sum_{x \in \mathcal{X}} g_{z,Q,P}(x)\ell(x) - \frac{z^2}{2} L_{\mathtt{th}}.
$$

Finally, it remains to establish that $\ell^*_{z^*,Q^*}$ is the unique minmax optimal solution. This can be shown in exactly the same manner as it was shown for Theorem 7.5.1 in the previous section; we skip the details. □

7.8.4 A saddle-point lemma

The following simple result is needed to establish the minmax optimality of our scheme. The first part of the result claims that any pair of minmax optimal x and maxmin optimal y forms a saddle point, a well-known fact. The second part claims that if the minimizer for the maxmin optimal y is unique, then it must also be minmax optimal and thereby constitute a saddle-point with y.

Lemma 7.8.5. *Consider the minmax problem* $\min_{x\in\mathcal{X}}\max_{y\in\mathcal{Y}} h(x,y)$*, and assume that*

$$\min_{x\in\mathcal{X}}\max_{y\in\mathcal{Y}} h(x,y) = \max_{y\in\mathcal{Y}}\min_{x\in\mathcal{X}} h(x,y).$$

Then, for every pair (x^*,y^*) *such that* $x^* \in \arg\min_{x\in\mathcal{X}}\max_{y\in\mathcal{Y}} h(x,y)$ *and* $y^* \in \arg\max_{y\in\mathcal{Y}}\min_{x\in\mathcal{X}} h(x,y)$ *constitutes a saddle-point. Furthermore, if the minimizer* $x^o(y^*)$ *of* $\min_{x\in\mathcal{X}} h(x,y^*)$ *is unique, then* $x^* = x^o(y^*)$ *is the unique minmax optimal solution.*

Proof. Since minmax and maxmin costs are assumed to be equal, by the definition of x^* and y^*, we have

$$\begin{aligned} h(x,y^*) &\geq \max_{y'\in\mathcal{Y}}\min_{x'\in\mathcal{X}} h(x',y') \\ &= \min_{x'\in\mathcal{X}}\max_{y'\in\mathcal{Y}} h(x',y') \geq h(x^*,y), \end{aligned} \tag{7.38}$$

for all x in $\mathcal{X}$ and y in $\mathcal{Y}$. Upon substituting x^* for x and y^* for y, we get that x^* is a minimizer of $h(x,y^*)$ and y^* a maximizer of $h(x^*,y)$. Therefore, (x^*,y^*) forms a saddle-point and $h(x^*,y^*) = \min_{x\in\mathcal{X}}\max_{y\in\mathcal{Y}} h(x,y)$.

Turning now to the second part, suppose that x', too, is minmax optimal. Then, using (7.38) with $x = x'$ and $y = y^*$, we get that x' must be a minimizer of $h(x,y^*)$ as well. But since this minimizer is unique, x' must coincide with x^o.

□

Proof of Lemma 7.5.2

Denoting

$$c_P(z,Q) := \sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\log\frac{\sum_{x'\in\mathcal{X}} g_{z,Q,P}(x')}{g_{z,Q,P}(x)},$$

we begin by observing the concavity of $c_P(z,Q)$. Recall the notations $\mathcal{G} = \{z \geq 0, Q \in \mathbb{R}^{|\mathcal{X}|} : g_{z,Q,P}(x) \geq 0 \quad \forall x \in \mathcal{X}\}$ and $g_{z,Q,P}(x) = (1-z^2/2)P(x) + z\sqrt{Q(x)P(x)}$.

Lemma 7.8.6. *The function* $c_P(z,Q)$ *is concave in* Q *for each fixed* z *and is concave in* z *for each fixed* Q*, over the set* $\mathcal{G}$*.*

Proof. For the first part, (7.35) yields that for every $(z, Q) \in \mathcal{G}$,

$$\sum_{x\in\mathcal{X}} g_{z,Q,P}(x) \log \frac{\sum_{x'\in\mathcal{X}} g_{z,Q,P}(x')}{g_{z,Q,P}(x)} = \min_{\ell\in\Lambda} \sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\ell(x).$$

Also, for every fixed z, the function $g_{z,Q,P}(x)$ is concave in Q, and thereby $\sum_{x\in\mathcal{X}} g_{z,Q,P}(x)\ell(x)$, is concave in Q. Thus, since the minimum of concave functions is concave, $c_P(z, Q)$ is concave in Q for a fixed z. Similarly, we can show concavity in z for a fixed Q since $g_{z,Q,P}(x)$ is concave in z, too, for every fixed Q. □

We now complete the proof of Lemma 7.5.2. We will show that for any (z, Q) which is feasible for optimization problem (7.9), we can find a feasible (z, Q') with Q' satisfying (7.11), and

$$c_P(z, Q) \le c_P(z, Q').$$

Indeed, consider $Q'(x) := Q(A_i)/|A_i|$ for all $x \in \mathcal{X}$. The remainder of the proof is divided into two parts, the first proving the feasibility of Q' and the second proving $c_P(z, Q) \le c_P(z, Q')$.

Feasibility of (z, Q') From the feasibility of (z, Q), for all symbols x in A_i and for all i in $[M_P]$, $g_{z,Q,P}(x) \ge 0$, whereby

$$\begin{aligned}
\sum_{x\in A_i} g_{z,Q,P}(x) &= \sum_{x\in A_i} \left(1 - \frac{z^2}{2}\right) P(x) + z \sum_{x\in A_i} \sqrt{Q(x)P(x)} \\
&= \left(1 - \frac{z^2}{2}\right) P(A_i) + z \sum_{x\in A_i} \sqrt{Q(x)P(x)} \\
&\ge \left(1 - \frac{z^2}{2}\right) P(A_i) + z\sqrt{Q'(A_i)P(A_i)} \\
&= |A_i| g_{z,Q',P}(x) \\
&\ge 0,
\end{aligned}$$

where the first inequality is by Cauchy-Schwarz inequality, the positivity of z, and the assumption that $P(x) = P(A_i)/|A_i|$ for every x in A_i, and the final identity uses definition of Q'. This proves the feasibility of (z, Q') for the optimization problem (7.9).

Proof of optimality Denoting by $\Pi(A_1)$ the set of all permutations of the elements of A_1, let Q^π be the distribution given by

$$Q^\pi(x) = \begin{cases} Q(\pi(x)), & \forall x \in A_1 \\ Q(x), & \text{otherwise.} \end{cases}$$

Then, the distribution $\overline{Q} = (1/|\Pi(A_1)|) \cdot \sum_{\pi \in \Pi(A_1)} Q^\pi$ satisfies

$$\overline{Q}(x) = \begin{cases} \frac{1}{|A_1|} \cdot Q(A_1), & \forall x \in A_1 \\ Q(x), & \text{otherwise.} \end{cases}$$

Since by Lemma 7.8.6 $c_P(z, Q)$ is concave in Q for every fixed z, we get

$$c_P(z, \overline{Q}) \geq \frac{1}{|\Pi(A_1)|} \cdot \sum_{\pi \in \Pi(A_1)} c_P(z, Q^\pi).$$

Furthermore, note that $g_{z,Q^\pi,P}(x) = g_{z,Q,P}(\pi(x))$ since $P(x) = P(A_1)/|A_1|$ for every x in A_1, and thereby $c_P(z, Q^\pi) = c_P(z, Q)$ for every $\pi \in \Pi(A_1)$. Therefore, combining the observations above, we obtain $c_P(z, \overline{Q}) \geq c_P(z, Q)$.

Repeating this argument by iteratively using permutations of A_i for $i \geq 2$, we obtain the required inequality

$$c_P(z, Q') \geq c_P(z, Q).$$

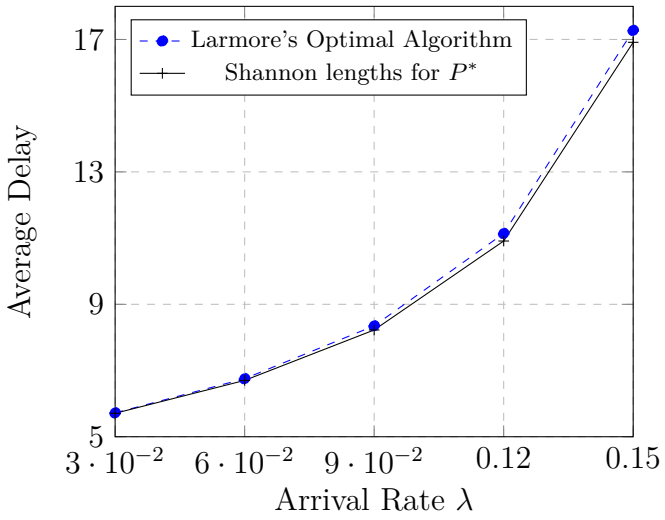
□

7.9 Concluding Remarks

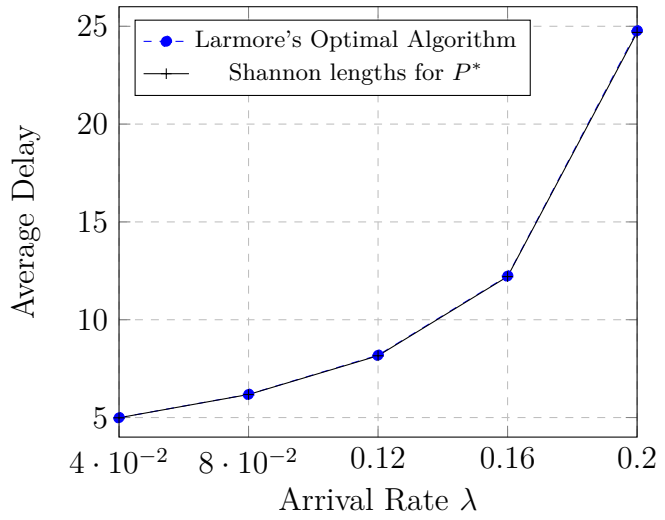
In this chapter, we studied the source coding problem where the goal is to minimize

$$\mathbb{E}[L] + \frac{\mathbb{E}[L^2]}{2\mathbb{E}[L]}$$

subject to the constraint that the code lengths satisfy Kraft's inequality. We saw that this problem differs from the standard source coding problem where the goal is to minimize $\mathbb{E}[L]$, and the classic source-coding solutions such as deploying Shannon codes may be suboptimal. Our main result was a structural result showing that the optimal code lengths for the relaxed version of the problem are Shannon lengths for tilting of the original distribution. Our recipe to prove this result was to linearize the cost function in terms of length by first expressing as the optimal value of a quadratic maximization problem over a new variable. Then, we use a variational formula for the L_2 norm of a random variable to linearize the cost. We believe that our approach can be used to prove similar structural results for other source coding problems, thereby gaining computational insights into solving them, as we saw with the application of our recipe for the problem of designing source codes with minimum delay.



(a) Comparison of proposed codes with Larmore's Algorithms [56] for the distribution $P(1) = 0.5$, and $P(i) = \frac{0.5}{255} \quad \forall i \in \{2, \cdots 256\}$.



(b) Comparison of proposed codes with Larmore's Algorithms [56] for the distribution $P(1) = 0.6$, and $P(i) = \frac{0.4}{255} \quad \forall i \in \{2, \cdots 256\}$.

Figure 7.7: Comparison of proposed codes with Larmore's Algorithms

Bibliography

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy." ACM, 2016, pp. 308–318.

[2] J. Acharya, C. L. Canonne, P. Mayekar, and H. Tyagi, "Information-constrained optimization: can adaptive processing of gradients help?" *arXiv preprint arXiv:2104.00979*, 2021.

[3] J. Acharya, C. L. Canonne, and H. Tyagi, "General lower bounds for interactive high-dimensional estimation under information constraints," *arXiv preprint arXiv:2010.06562*, 2020.

[4] J. Acharya, C. De Sa, D. J. Foster, and K. Sridharan, "Distributed Learning with Sublinear Communication," *arXiv:1902.11259*, 2019.

[5] A. Agarwal, P. L. Bartlett, P. Ravikumar, and M. J. Wainwright, "Information-Theoretic Lower Bounds on the Oracle Complexity of Stochastic Convex Optimization," *IEEE Transactions on Information Theory*, vol. 5, no. 58, pp. 3235–3249, 2012.

[6] N. Agarwal, A. T. Suresh, F. X. X. Yu, S. Kumar, and B. McMahan, "cpSGD: Communication-efficient and differentially-private distributed SGD," *Advances in Neural Information Processing Systems*, pp. 7564–7575, 2018.

[7] N. Ailon and B. Chazelle, "Approximate nearest neighbors and the fast johnson-lindenstrauss transform," *Proceedings of the ACM symposium on Theory of computing (STOC'06)*, pp. 557–563, 2006.

[8] E. Akyol and K. Rose, "On constrained randomized quantization," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3291–3302, July 2013.

[9] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," *Advances in Neural Information Processing Systems*, pp. 1709–1720, 2017.

[10] B. T. Bacinoglu and E. Uysal-Biyikoglu, "Scheduling status updates to minimize age of information with an energy harvesting sensor," *Proceedings of the IEEE International Symposium on Information Theory, 2017 (ISIT' 17)*, pp. 1122–1126, 2017.

[11] M. B. Baer, "Source coding for quasiarithmetic penalties," *IEEE transactions on information theory*, vol. 52, no. 10, pp. 4380–4393, 2006.

[12] S. Bhambay, S. Poojary, and P. Parag, "Differential encoding for real-time status updates," *Proceedings of the IEEE Wireless Communications and Networking Conference, 2017 (WCNC' 17)*, pp. 1–6, 2017.

[13] S. Boucheron, G. Lugosi, and P. Massart, *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

[14] M. Braverman, A. Garg, T. Ma, H. L. Nguyen, and D. P. Woodruff, "Communication lower bounds for statistical estimation problems via a distributed data processing inequality," *Proceedings of ACM Symposium on the Theory of Computing (STOC' 16)*, pp. 1011–1020, 2016.

[15] S. Bubeck, "Convex optimization: Algorithms and complexity," *Foundations and Trends® in Machine Learning*, vol. 8, no. 3-4, pp. 231–357, 2015.

[16] L. L. Campbell, "A coding theorem and rényi's entropy," *Information and control*, vol. 8, no. 4, pp. 423–429, 1965.

[17] W.-N. Chen, P. Kairouz, and A. Özgür, "Breaking the communication-privacy-accuracy trilemma," *arXiv preprint arXiv:2007.11707*, 2020.

[18] T. M. Cover and J. A. Thomas, *Elements of Information Theory. 2nd edition.* John Wiley & Sons Inc., 2006.

[19] I. Csiszár and P. Narayan, "Capacity of the gaussian arbitrarily varying channel," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 18–26, 1991.

[20] P. Davies, V. Gurunathan, N. Moshrefi, S. Ashkboos, and D. Alistarh, "Distributed variance reduction with optimal communication," *arXiv e-prints*, pp. arXiv–2002, 2020.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *Proceedings of the conference on computer vision and pattern recognition*.

[22] A. S. Drud, "Conopt-a large-scale grg code," *ORSA Journal on computing*, vol. 6, no. 2, pp. 207–216, 1994.

[23] J. C. Duchi, "Introductory lectures on stochastic optimization," 2017, Available Online http://stanford.edu/~jduchi/PCMIConvex/Duchi16.pdf.

[24] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Privacy aware learning," *Journal of the ACM (JACM)*, vol. 61, no. 6, pp. 1–57, 2014.

[25] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and Y. Zhang, "Optimality guarantees for distributed statistical estimation," *arXiv:1405.0782*, 2014.

[26] J. C. Duchi and R. Rogers, "Lower bounds for locally private estimation via communication complexity," in *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, ser. Proceedings of Machine Learning Research, A. Beygelzimer and D. Hsu, Eds., vol. 99. PMLR, 2019, pp. 1161–1191. [Online]. Available: http://proceedings.mlr.press/v99/duchi19a.html

[27] I. Dumer, "Covering spheres with spheres," *Discrete & Computational Geometry*, vol. 38, no. 4, pp. 665–679, Dec 2007.

[28] F. Faghri, I. Tabrizian, I. Markov, D. Alistarh, D. Roy, and A. Ramezani-Kebrya, "Adaptive gradient quantization for data-parallel sgd," *Advances in Neural Information Processing Systems*, 2020.

[29] V. Feldman, C. Guzmán, and S. Vempala, "Statistical query algorithms for mean vector estimation and stochastic convex optimization," *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'17)*, pp. 1265–1277, 2017.

[30] G. D. Forney, "Coset codes. i. introduction and geometrical classification," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 1123–1151, 1988.

[31] R. Fourer, D. M. Gay, and B. Kernighan, *Ampl*. Boyd & Fraser Danvers, MA, 1993, vol. 117.

[32] R. G. Gallager, *Information theory and reliable communication*. Springer, 1968, vol. 2.

[33] V. Gandikota, D. Kane, R. K. Maity, and A. Mazumdar, "vqsgd: Vector quantized stochastic gradient descent," *arXiv preprint arXiv:1911.07971*, 2019.

[34] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Springer Science & Business Media, 2012, vol. 159.

[35] A. Ghosh, R. K. Maity, and A. Mazumdar, "Distributed newton can communicate less and resist byzantine workers," *Advances in Neural Information Processing Systems*, 2020.

[36] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM review*, vol. 47, no. 1, pp. 99–131, 2005.

[37] A. M. Girgis, D. Data, S. Diggavi, P. Kairouz, and A. T. Suresh, "Shuffled model of federated learning: Privacy, communication and accuracy trade-offs," *arXiv preprint arXiv:2008.07180*, 2020.

[38] V. Gupta, A. Ghosh, M. Derezinski, R. Khanna, K. Ramchandran, and M. Mahoney, "Localnewton: Reducing communication bottleneck for distributed learning," *arXiv preprint arXiv:2105.07320*, 2021.

[39] R. Hadad and U. Erez, "Dithered quantization via orthogonal transformations," *IEEE Transactions on Signal Processing*, vol. 64, pp. 5887–5900, 11 2016.

[40] M. K. Hanawal and R. Sundaresan, "Guessing revisited: A large deviations approach," *IEEE Transactions on Information Theory*, vol. 57, no. 1, pp. 70–78, 2011.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the conference on computer vision and pattern recognition*.

[42] Q. He, D. Yuan, and A. Ephremides, "Optimal link scheduling for age minimization in wireless systems," *IEEE Transactions on Information Theory*, vol. 64, no. 7, pp. 5381–5394, 2018.

[43] T. Holenstein, "Parallel repetition: Simplification and the no-signaling case," *Theory of Computing*, vol. 5, no. 8, pp. 141–172, 2009.

[44] K. J. Horadam, *Hadamard matrices and their applications*. Princeton university press, 2012.

[45] X. Hu, L. Prashanth, A. György, and C. Szepesvári, "(Bandit) Convex Optimization with Biased Noisy Gradient Oracles," *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS' 16)*, pp. 819–828, 2016.

[46] Z. Huang, W. Yilei, K. Yi *et al.*, "Optimal sparsity-sensitive bounds for distributed mean estimation," *Advances in Neural Information Processing Systems*, pp. 6371–6381, 2019.

[47] B. Hughes and P. Narayan, "Gaussian arbitrarily varying channels," *IEEE Transactions on Information Theory*, vol. 33, no. 2, pp. 267–284, 1987.

[48] P. A. Humblet, "Source coding for communication concentrators," *Ph. D. Dissertation, MIT*, 1978.

[49] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[50] S. Kaul, R. Yates, and M. Gruteser, "On piggybacking in vehicular networks," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*. IEEE, 2011, pp. 1–5.

[51] ——, "Real-time status: How often should one update?" in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 2731–2735.

[52] J. Konečný and P. Richtárik, "Randomized distributed mean estimation: Accuracy vs. communication," *Frontiers in Applied Mathematics and Statistics*, vol. 4, p. 62, 2018.

[53] S. B. Korada and R. L. Urbanke, "Polar codes are optimal for lossy source coding," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1751–1768, 2010.

[54] A. Kosta, N. Pappas, V. Angelakis *et al.*, "Age of information: A new concept, metric, and tool," *Foundations and Trends® in Networking*, vol. 12, no. 3, pp. 162–259, 2017.

[55] A. Lapidoth, "On the role of mismatch in rate distortion theory," *IEEE Transactions on Information Theory*, vol. 43, no. 1, pp. 38–47, 1997.

[56] L. L. Larmore, "Minimum delay codes," *SIAM Journal on Computing*, vol. 18, no. 1, pp. 82–94, 1989.

[57] K. Liang and Y. Wu, "Improved communication efficiency for distributed mean estimation with side information," *arXiv preprint arXiv:2102.02525*, 2021.

[58] C.-Y. Lin, V. Kostina, and B. Hassibi, "Achieving the fundamental convergence-communication tradeoff with differentially quantized gradient descent," *arXiv preprint arXiv:2002.02508*, 2020.

[59] C. Ling, S. Gao, and J. Belfiore, "Wyner-ziv coding based on multidimensional nested lattices," *IEEE Transactions on Communications*, vol. 60, no. 5, pp. 1328–1335, 2012.

[60] L. Liu, "Polar codes and polar lattices for efficient communication and source quantization," *Ph.D. Thesis*, 2016.

[61] L. Liu and C. Ling, "Polar lattices are good for lossy compression," *CoRR*, vol. abs/1501.05683, 2015.

[62] Y. Lyubarskii and R. Vershynin, "Uncertainty principles and vector quantization," *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3491–3501, 2010.

[63] A. Mahajan and D. Teneketzis, "Optimal design of sequential real-time communication systems," *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 5317–5338, 2009.

[64] E. Martinian and M. Wainwright, "Low density codes achieve the rate-distortion bound," *Proceedings of the IEEE Data Compression Conference*, pp. 153–162, 2006.

[65] P. Mayekar, P. Parag, and H. Tyagi, "Optimal lossless source codes for timely updates," *Proc. International Symposium of Information Theory*, pp. 1246–1250, 2018.

[66] P. Mayekar, A. Theertha Suresh, and H. Tyagi, "Wyner-ziv estimators: Efficient distributed mean estimation with side-information," *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics (AISTATS' 21)*, pp. 3502–3510, 2021.

[67] P. Mayekar and H. Tyagi, "Limits on gradient compression for stochastic optimization," *Proceedings of the IEEE International Symposium of Information Theory (ISIT' 20)*, 2020.

[68] ——, "RATQ: A universal fixed-length quantizer for stochastic optimization," *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS' 20)*, pp. 1399–1409, 2020.

[69] ——, "RATQ: A universal fixed-length quantizer for stochastic optimization," *Transactions on Information Theory*, 2021.

[70] A. Nemirovsky, "Information-based complexity of convex programming," 1995, Available Online http://www2.isye.gatech.edu/ne-mirovs/Lec_EMCO.pdf.

[71] A. Nemirovsky and D. B. Yudin, "Problem complexity and method efficiency in optimization." *Wiley series in Discrete Mathematics and Optimization*, 1983.

[72] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.

[73] ——, "Introductory lectures on convex optimization: A basic course," *Springer Science and Business Media*, vol. 87, 2013.

[74] Y. Oohama, "Gaussian multiterminal source coding," *IEEE Transactions on Information Theory*, vol. 43, no. 6, pp. 1912–1923, 1997.

[75] V. Ostromoukhov, R. D. Hersch, and I. Amidror, "Rotated dispersed dither: A new technique for digital halftoning," *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '94)*, pp. 123–130, 1994.

[76] G. Pisier, "Remarques sur un résultat non publié de b. maurey," *Séminaire Analyse fonctionnelle (dit" Maurey-Schwartz")*, pp. 1–12, 1981.

[77] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (discus): design and construction," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 626–643, 2003.

[78] A. Ramezani-Kebrya, F. Faghri, and D. M. Roy, "Nuqsgd: Improved communication efficiency for data-parallel sgd via nonuniform quantization," *arXiv preprint arXiv:1908.06077*, 2019.

[79] A. Rényi, "On measures of entropy and information," *Proc. Fourth Berkeley Symposium on Mathematics Statistics and Probability, Vol. 1 (Univ. of Calif. Press)*, pp. 547–561, 1961.

[80] P. Richtárik and M. Takác, "Parallel coordinate descent methods for big data optimization, arxiv e-prints," *arXiv preprint arXiv:1212.0873*, 2012.

[81] S. M. Ross, *Stochastic processes*. Wiley, New York, 1996.

[82] M. Safaryan, E. Shulgin, and P. Richtárik, "Uncertainty principle for communication compression in distributed and federated learning and the search for an optimal compressor," *arXiv preprint arXiv:2002.08958*, 2020.

[83] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, "1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns," *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[84] M. Sion, "On general minimax theorems," *Pacific Journal of mathematics*, vol. 8, no. 1, pp. 171–176, 1958.

[85] N. Sommer, M. Feder, and O. Shalvi, "Low-density lattice codes," *IEEE Transactions on Information Theory*, vol. 54, no. 4, pp. 1561–1585, April 2008.

[86] P. Subramani, N. Vadivelu, and G. Kamath, "Enabling fast differentially private sgd via just-in-time compilation and vectorization," *arXiv preprint arXiv:2010.09063*, 2020.

[87] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," *IEEE Transactions on Information Theory*, vol. 63, no. 11, pp. 7492–7508, 2017.

[88] A. T. Suresh, F. X. Yu, S. Kumar, and H. B. McMahan, "Distributed mean estimation with limited communication," *Proceedings of the International Conference on Machine Learning (ICML' 17)*, vol. 70, pp. 3329–3337, 2017.

[89] S. Tatikonda and S. Mitter, "Control under communication constraints," *IEEE Transactions on Automatic Control*, vol. 49, no. 7, pp. 1056–1068, 2004.

[90] Y. Wu, "Lecture notes on information-theoretic methods for high-dimensional statistics," *Lecture Notes for ECE598YW (UIUC)*, vol. 16, 2017.

[91] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on information Theory*, vol. 22, no. 1, pp. 1–10, 1976.

[92] A. D. Wyner, "Random packings and coverings of the unit n-sphere," *The Bell System Technical Journal*, vol. 46, no. 9, pp. 2111–2118, 1967.

[93] Y. Yan, C. Ling, and X. Wu, "Polar lattices: where arıkan meets forney," *Proceedings of the IEEE International Symposium of Information Theory (ISIT' 13)*, pp. 1292–1296, 2013.

[94] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," *Proceedings of the IEEE International Symposium on Information Theory 2015, (ISIT' 15)*, pp. 3008–3012, 2015.

[95] B. Yu, "Assouad, fano, and le cam," in *Festschrift for Lucien Le Cam*. Springer, 1997, pp. 423–435.

[96] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1250–1276, 2002.

[97] J. Zhong, R. D. Yates, and E. Soljanin, "Timely lossless source coding for randomly arriving symbols," *arXiv:1810.01533*, 2018.

[98] J. Zhong and R. D. Yates, "Timeliness in lossless block coding," *2016 Data Compression Conference (DCC)*, pp. 339–348, 2016.

[99] J. Zhong, R. D. Yates, and E. Soljanin, "Backlog-adaptive compression: Age of information," *Proceedings of the IEEE International Symposium on Information Theory, 2017 (ISIT' 17)*, pp. 566–570, 2017.

[100] J. Ziv, "On universal quantization," *IEEE Transactions on Information Theory*, vol. 31, no. 3, pp. 344–347, 1985.