UNIVERSITY OF CALIFORNIA SAN DIEGO

Doing it for the Bit: Applications of Quantization in Data Science and Signal Processing

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Mathematics

by

Eric Lybrand

Committee in charge:

Professor Rayan Saab, Chair
Professor Kamalika Chaudhuri
Professor Alex Cloninger
Professor Todd Kemp
Professor Bhaskar Rao

2021

The dissertation of Eric Lybrand is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California San Diego

2021

DEDICATION

To my family, my friends, and all of the people who helped me along the way.

You know who you are!

EPIGRAPH

I witness with pleasure the supreme achievement of memory, which is the masterly use it makes of innate harmonies when gathering to its fold the suspended and wandering tonalities of the past.

—Vladimir Nabokov

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Before all else, I am grateful for my family for their relentless and emphatic support. Thank you all for giving me strength, bringing me back down to earth, and reminding me that the most important question to ask is “why choose?”

The entirety of this thesis has been made possible by the continued and careful guidance of Rayan Saab. I extend my thanks to him for his support and his mentorship through the ups and the downs of graduate school.

I also would like to thank Ted Shifrin, Jason Cantarella, and Neil Lyall for their roles in shaping my career as an undergraduate at UGA. I am the mathematician I am today because of those three, and the many others who helped me along the way.

I would be remiss if I did not extend my gratitude for the innumerably many friends I have made along the way. In no particular order, and without claim to completeness, I am beyond thankful for Mallory, Fred, Courtney, Nish, Sasha, Christian, Fernán, Miguel, the Elmores, the Campbells, Anna, Debbie, Pieter, and Jacqueline. Your friendship cannot be overstated.

Chapter 2, in full, is joint work with Rayan Saab, and it has been submitted for publication as it may appear in the Journal of Machine Learning Research, 2021. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in full, is joint work with Rayan Saab and has been published in Information and Inference, 2019. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in full, is joint work with Mark Iwen, Aaron Nelson, and Rayan Saab and has been published in 13th International conference on Sampling Theory and Applications (SampTA). IEEE, 2019. The dissertation author was the primary investigator and author of this paper.

VITA

2011-2015	Bachelor of Science in Mathematics, *summa cum laude*, University of Georgia
2015-2021	Doctor of Philosophy in Mathematics, University of California San Diego

PUBLICATIONS

E. Lybrand, R. Saab, *Quantization for Low-Rank Matrix Recovery*, Information and Inference: A Journal of the IMA 8.1, 2018.

H. Huang and T. Kemp and Y. Ling and X. Luo and E. Lybrand and R. Smith and J. Wang. "Random Matrices with Independent Diagonals." preprint, 2018.

M. A. Iwen, E. Lybrand, A.A. Nelson, R. Saab, *New Algorithms and Improved Guarantees for One-Bit Compressed Sensing on Manifolds*, in: Proceedings of the 17th International Conference on Sampling Theory and Applications (SampTA), 2019.

E. Lybrand and R. Saab. "A Greedy Algorithm for Quantizing Neural Networks." preprint, 2020.

E. Lybrand, A. Ma, and R. Saab. "On the Number of Faces and Radii of Cells Induced by Gaussian Spherical Tessellations." preprint, 2020.

ABSTRACT OF THE DISSERTATION

Doing it for the Bit: Applications of Quantization in Data Science and Signal Processing

by

Eric Lybrand

Doctor of Philosophy in Mathematics

University of California San Diego, 2021

Professor Rayan Saab, Chair

This thesis explores topics in quantization in the context of data science and digital signal processing. It is already well-known that quantization plays a critical role in signal processing given the ubiquity of digital devices. Within the past few years quantization has been considered as a means of model selection and compression in data science. Chapter 1 gives an introduction to quantization and an overview of the later chapters in the dissertation.

The second chapter considers the problem of quantizing a pre-trained neural network with a data-driven approach. We propose a novel algorithm for quantizing the weights of a hidden unit in a given layer using a greedy path following algorithm which is equivalent to running a dynamical system. We prove that this dynamical system is stable when quantizing a single layer

neural network, or equivalently the first layer of a multi-layer network, when the training data are Gaussian. We then give bounds on the generalization error of the learned quantization and describe how these bounds may be tightened when the feature data lay in a low-dimensional subspace.

In the third chapter, we generalize a previous result for recovering sparse vectors to the setting of low-rank matrices. Specifically, we show that low-rank matrices may be robustly recovered from $\Sigma\Delta$ quantized subgaussian linear measurements. Using these techniques we show that the reconstruction error associated with our methods decays polynomially with the oversampling factor, and that one can leverage this result to obtain root-exponential accuracy by optimizing over the choice of quantization scheme. Additionally, if a random encoding of the quantized measurements is used then the reconstruction bound exhibits a near-optimal exponential rate-distortion trade-off.

In the fourth chapter, we generalize previous work to prove that signal recovery is possible when given $\Sigma\Delta$ quantized linear measurements of signals from a manifold where the manifold is only approximately known through a Geometric Multi-Resolution Analysis. In addition to proving error bounds that outperform those in the Memoryless Scalar Quantization approach taken in the aforementioned work, we also extend the ensemble of linear measurements beyond the Gaussian case.

Chapter 1

Introduction

The process by which we map the continuum to a discrete, and usually finite, set is known as quantization. Quantization forms the very bedrock of all of modern day computing. Computers, mobile devices, and servers all compute, communicate, and record data in bits. All of these technologies play an increasingly important role in our daily activities, and their computing capabilities continue to grow at surprising rates much like Moore's law has predicted since 1975. Only recently however have we seen a new trend emerging in modern computing, namely that which is referred to as big data. Large datasets and complex machine learning models like neural networks have enabled state-of-the-art results on tasks like image recognition, self-driving vehicles, and beating world-class champions in games like Go [11, 19, 24, 26]. Interestingly, and perhaps alarmingly, the amount of compute required to train these state-of-the-art models has been growing at a faster rate than what Moore's law dictates that hardware can keep up with [1]. In a similar vein, there are increasingly many signal processing contexts like astronomy and healthcare where the goal is to understand the structure of a particular object, be it through a telescope or MRI, by taking many measurements across space and time. Storing and computing with these measurements can pose significant challenges due to the sheer amount of data collected. As we enter into this new era of data extravagance, it is worth pausing to consider whether we

can be more efficient with our bits.

The goal of this thesis is to contribute novel results in quantization within the context of data science and signal processing. Specifically, we will look at quantization of multilayer perceptrons and convolutional neural networks, and we will also look at quantization in signal acquisition and recovery problems that employ techniques from compressed sensing. Since quantization inevitably induces an approximation error, or distortion, we will consider the trade-offs between minimizing the quantization error and also minimizing the number of bits, or the rate, we use to encode our signals or machine learning models. The underlying theme for all of these projects is that overparameterization and redundancy are crucial ingredients for better quantized representations. How we quantify overparameterization or redundancy will depend on the context of the problem. Since different quantizers can leverage redundancy to achieve better rate-distortion trade-offs, we will first provide an overview of three common quantizers in the context of frame theory.

1.1 Illustrated Example: Quantized Frame Theory

Classical sampling theory concerns itself with reconstructing a bandlimited function $f : \mathbb{R} \to \mathbb{R}$, for example, from point samples $\{f(x_i)\}_{i\in\mathbb{Z}}$. The fundamental result in this context is the Shannon-Nyquist theorem which states that perfect reconstruction is possible via convolution with a sinc kernel provided the sampling rate exceeds twice the bandwidth of f [25]. Frame theory generalizes sampling theory by considering linear samples rather than pointwise samples. Note that we can reduce to recovering vectors instead of functions over $\mathbb{R}$ since ultimately we are concerned with digitizing these samples onto hardware. Frame theory is centered around frame expansions of a vector $x \in \mathbb{R}^N$, where a frame $\{\varphi_1, \ldots, \varphi_m\} \subset \mathbb{R}^N$ is nothing more than a collection of vectors which spans $\mathbb{R}^N$. It is notationally convenient to consider the matrix $\Phi \in \mathbb{R}^{m\times N}$ whose rows are φ_i^T. Given a frame Φ, we say $\Psi \in \mathbb{R}^{N\times m}$ is an associated dual frame

if for all $x \in \mathbb{R}^N$ we have $\Psi\Phi x = x$. Note that because $\Phi \in \mathbb{R}^{m\times N}$ has rank N there are infinitely many dual frames. This observation will come in handy later on in our discussion. In the context of frame theory then, our samples are the inner products $\{\varphi_i^T x\}_{i\in[m]}$ and we recover x from these samples by calculating $\sum_{i=1}^m \langle \varphi_i, x\rangle \psi_i$. It is common in the frame theory context to refer to the operators Φ and Ψ as analysis and synthesis operators, respectively. Common examples of frames include harmonic frames which are submatrices of a Fourier matrix, Gabor frames which are normalized time-frequency shifts of a window function, and wavelet frames which, similar to Gabor frames, are designed to be well-localized in both time and frequency domains [2]. Of course, reconstructing x from directly applying the synthesis and analysis operators assumes that we have infinite-precision and noiseless frame coefficients $\{\varphi_i^T x\}_{i\in[m]}$. Both of these assumptions fail to be met in practice. For this reason we instead work with noiseless quantized, or encoded, measurements where the analog measurements Φx, are mapped to a finite discrete set $\mathscr{S}$ which we will refer to as a codebook.

We can slightly generalize this discussion in a manner similar to that in [3, 22] by viewing the application of the analysis operator Φ as an intermediate step of the encoder. A natural question to consider is for $\|x\| \leq 1$ and codebook $\mathscr{S} \subset \mathbb{R}^m$ how to design an encoder-decoder pair $(\mathscr{E}, \mathscr{D})$ with the properties that $\mathscr{E} : \mathbb{R}^N \to \mathscr{S}$ so that $\|x - \mathscr{D}(\mathscr{E}(x))\|$ is well-controlled. Specifically, in the context of frame theory we will assume that the decoder is some synthesis frame $\Psi \in \mathbb{R}^{N\times m}$. The synthesis problem concerns itself with the special case when the synthesis operator Ψ is fixed, and the practitioner is allowed to design the encoding map $\mathscr{E} : \mathbb{R}^N \to \mathscr{S}$ however he or she pleases. The goal, of course, is then to minimize or approximately minimize the synthesis distortion $D_s(\Phi, \mathscr{Q}) := \sup_{\|x\|\leq 1} \|x - \Psi\mathscr{E}(x)\|$. Naturally for a given codebook $\mathscr{S}$ the optimal choice of $\mathscr{E}$ is $\mathscr{E}^*(x;\mathscr{S}) := \arg\min_{q\in\mathscr{S}} \|x - \Psi q\|$. However, finding an efficient implementation of this quantization scheme for general Ψ and general codebooks reduces to solving a combinatorial optimization problem which is NP-Hard. Further, it is often desired that the quantization scheme exhibit a distortion which decays quickly with m, though what

the optimal decay is in terms of m is unknown for general Ψ except in a few special cases [22]. Fast distortion decay as a function of m allows the practitioner to make a trade off between high resolution codebooks and oversampling as a means of minimizing the distortion. Having this trade off is important because, depending on the context, oversampling could be much cheaper in terms of time and resources than allocating more bits to the codebook. This is the case, for example, in the context of encoding audio signals [14]. In the extreme case fast distortion decay as a function of oversampling allows the practitioner to work with the one-bit codebook $\{-1,1\}^m$ while ensuring a high-fidelity reconstruction of x.

Related to the synthesis problem is the analysis problem where we acquire measurements of the signal x through some analysis frame Φ whose structure is imposed upon us by the physics of the problem. That is, the encoding function must be of the form $\mathscr{E}(x) = \mathscr{Q}(\Phi x)$, where $\mathscr{Q} : \mathbb{R}^m \to \mathscr{S}$. Though the encoder is constrained to be of a particular form, the practitioner is allowed to choose any synthesis frame Ψ to decode the quantized measurements so long as $\Psi\Phi = I$. In other words, the practitioner is allowed to construct $\mathscr{Q}$ and Ψ however they please to minimize $D_a(\Psi, \mathscr{Q}) := \sup_{\|x\|\leq 1} \|x - \Psi\mathscr{Q}(\Phi x))\|$.

Both synthesis and analysis problems focus on the design of an encoding or quantization map but constrain the designer in one of either the encoding or decoding steps. The analysis problem allows the designer to choose among the infinitely many dual frame operators Ψ to use for decoding but constrains the encoding function in such a way that the practitioner encodes linear measurements of the signal x through a fixed analysis operator Φ. On the other hand, the synthesis problem imposes upon the designer a fixed synthesis frame Ψ for decoding but allows the designer an arbitrary choice encoding function. To get a better idea of the subtleties and trade offs that arise in designing synthesis frames–that is, linear decoders–and quantization mappings we will take a look at the following three examples from the perspective of the analysis problem. Our discussion will start by looking at two extreme examples of quantization maps $\mathscr{Q}$ and discuss their trade offs in computational complexity and how well they control distortion. We

will then end our discussion by introducing a third quantization scheme which enjoys a favorable compromise between these two extreme examples.

1.1.1 Vector Quantization

Knowing that we are sampling signals $x \in \mathbb{R}^N$ with $\|x\| \leq 1$, we can take $\mathscr{S}$ to be an ε-net of the ℓ_2-ball $B_2^N \subset \mathbb{R}^N$. To digitize this net, we could approximate the net elements with their best b-bit floating point representations, for example. Given measurements $y = \Phi x$ we can then use the quantizer $\mathscr{Q}_{VQ}(y) := \arg\min_{q \in \mathscr{S}} \|\Psi y - q\|$. To decode the quantized measurements, we simply return the quantized bit $\mathscr{D}_{VQ}(q) := q$. The perks of this quantization scheme is that it explicitly bounds, by construction, the distortion by ε. This fine control over the distortion comes with a heavy price though. A quick volumetric argument [28] shows that $|\mathscr{S}| \geq \varepsilon^{-N}$. From this there are two unfortunate implications. The first is that searching through the codebook $\mathscr{S}$ will be prohibitively costly when N is large and ε is small. Furthermore, the sheer amount of memory for storing the codebook $\mathscr{S}$ nearly defeats the purpose of quantizing the frame coefficients. Indeed, a back of the envelope calculation shows that storing $\mathscr{S}$ with b-bits per entry of the vectors would require at least $b \cdot N \cdot \varepsilon^{-N}$ bits in total. Nevertheless, the codes for the quantized samples would require at least $-N\log(\varepsilon)$ bits. We can write the distortion $D_{VQ}(x) := \|x - \mathscr{D}_{VQ}(\mathscr{Q}_{VQ}(\Phi x)))\|$ as a function of the rate $R_{VQ} := -N\log(\varepsilon)$ to see how the error decays as we allocate more bits, or equivalently more points in the net $\mathscr{S}$, to the quantization. Doing so gives us $D_{VQ} \leq e^{-R_{VQ}/N}$. In other words, vector quantization attains an exponential decrease in distortion as a function of rate. As mentioned before though, this rate-distortion relationship hides the memory footprint of storing the codebook $\mathscr{S}$ and the computational complexity for encoding and decoding. Furthermore, increasing the number of frame measurements, or the redundancy, m has no effect on the distortion and only increases the computational burden of encoding.

To avoid the memory overhead of storing a large and unstructured codebook one can instead consider more structured codebooks that arise from lattices. The simplest example of a

lattice codebook is a uniform grid over the unit hypercube $\mathscr{A}^N = \{-1 + \frac{j}{M} : j \in \{0, 1, ..., 2M\}\}^N$ where $M \in \mathbb{N}$ determines the resolution of the grid. Quantizing vectors to this lattice simply reduces to rounding each coordinate to the nearest element in $\mathscr{A}$. This lattice quantizer is known as Memoryless Scalar Quantization which we will detail in the section below. Other well-known lattice quantizers include root lattices, the Coxeter-Todd lattice in $\mathbb{R}^{12}$, the Barnes-Wall lattice in $\mathbb{R}^{16}$, and the Leech lattice in $\mathbb{R}^{24}$ [10]. These aforementioned lattices admit fast quantizers because of the group symmetries inherent in the lattice structure, and these symmetries allow the practitioner to store only a small generating set of the lattice rather than the entire lattice. Working with lattice quantizers in $\mathbb{R}^m$ for general m does have some drawbacks. For example, the last three aforementioned lattices exist only in the special cases when $m = 12, 16, 24$. Of course, if one were in the setting where m were an integer multiple of any of these three integers then one could perform block lattice quantization where one quantizes blocks of coordinates of size $12, 16$, or 24. One would run into problems though in high dimensions, or in the oversampling regime, since the quantization errors for each block would accumulate and lead to poor distortion decay similar to that seen in Memoryless Scalar Quantization detailed in the following subsection. Outside of these special scenarios one must work instead with more generic lattices like root lattices. Certain lattice quantizers like the Leech lattice are known to be optimal quantizers with regards to mean squared error in particular scenarios such as when the number of samples m is not too large and the samples are uniformly distributed in space [10]. Of course, this need not be case, especially if x is coming from a structured set like k-sparse vectors in $\mathbb{R}^N$.

In fact, both the lattice quantizer paradigm and the ε-net quantizer have in mind the scenario where the distribution of measurements Φx is uniform across some bounded set. As we alluded to before, there are many natural scenarios where the structure of the underlying signal class x is being drawn from may induce biases in the distribution of the measurements Φx. To account for these biases one could take a more adaptive, data-driven approach and learn a codebook to quantize to. That is, given $L \gg 1$ pre-recorded analog measurements

$\{\Phi x_j\}_{j\in[L]} \subset \mathbb{R}^m$, one could then cluster these measurements into k-clusters using an algorithm such as Lloyd's algorithm [21] or its generalization the Linde-Buzo-Gray algorithm [20]. These of course are only heuristic algorithms which attempt to solve the NP-hard problem of constructing an optimal codebook. In fact, solving the k-means clustering problem to global optimality is itself a NP-Hard problem [6]. Nevertheless, it is known that as $L \to \infty$ that Lloyd's algorithm produces a consistent limiting quantizer in the sense that the learned clusters from the empirical data approach those that would be learned as though one were to use the true distribution [23]. Additionally, some weak global convergence results have been shown [9]. However, these convergence results rely on asymptotically increasing the number of empirical data used and letting the algorithms run for an indefinite number of iterations. Both of these cannot be met in practice. To further add to these challenges, one must worry about dealing with the curse of dimensionality when $m \gg 1$ and having a large enough empirical sample that approximates the true underlying distribution well-enough. Lastly, for signals such as audio signals it may be expensive to collect analog measurements $\{\Phi x_j\}_{j\in[L]}$ to use for learning the codebook.

1.1.2 Memoryless Scalar Quantization

Many of the issues that vector quantization faces disappear if we restrict our attention to scalar quantizers. Given a quantization alphabet $\mathscr{A} \subset \mathbb{R}$, the set $\{0,1\}$ being the canonical example, we set $\mathscr{S} := \mathscr{A}^m$. The simplest scalar quantizer is one that rounds to the closest quantization element $\mathscr{Q}_{MSQ} : \mathbb{R} \to \mathscr{A}$ by $\mathscr{Q}_{MSQ}(x) := \arg\min_{z\in\mathscr{A}} |x-z|$. This kind of quantization is known as Memoryless Scalar Quantization, or MSQ. To quantize vectors, we simply apply this quantizer to each of the vector's coordinates. In contrast to vector quantization, the memory overhead for storing $\mathscr{A}$ is quite small. Indeed, for the choice of $\mathscr{A} = \{0,1\}$ we only require 1 bit. The computational complexity for quantizing a vector in $\mathbb{R}^N$ is a mere N flops, which is optimal in the sense that any quantizer must look at each entry of the vector at least once. With a slight abuse of notation, define $\mathscr{Q}_{MSQ} : \mathbb{R}^m \to \mathscr{A}^m$ via $\mathscr{Q}_{MSQ}(y)_i = \arg\min_{z\in\mathscr{A}} |y_i - z|$. Various

algorithms exist for decoding MSQ measurements depending on whether the decoded signal $x^\sharp$ is chosen to satisfy $\mathscr{Q}_{MSQ}(\Phi x^\sharp) = \mathscr{Q}_{MSQ}(\Phi x)$ or not [22]. Choosing to enforce this constraint is known as consistent reconstruction. Consistent reconstruction can be attained by a linear program which runs in time proportional to matrix multiplication by Φ [27]. Unlike vector quantization, the rate of MSQ grows with the number of measurements and is equal to $R_{MSQ} := m$ in the one-bit case. With all of these perks, MSQ pays greatly for its simplicity by having a poor rate-distortion trade-off. It is known that any decoder $\mathscr{D}_{MSQ}$, consistent or not, using MSQ quatized frame coefficients cannot have distortion $D_{MSQ}(x) := \|x - \mathscr{D}_{MSQ}(\mathscr{Q}_{MSQ}(\Phi x))\|$ decay faster than, up to constants, m^{-1} [12]. In other words, the best rate-distortion trade-off for MSQ can be no better than $D_{MSQ} \lesssim R_{MSQ}^{-1}$. This terribly slow distortion decay can be quite limiting in practice. As an extreme but insightful example, bounding the distortion by 10^{-16} would require on the order of 10^{16} one-bit measurements, or 1250 terabytes of data. At this point, one wonders what is gained from these one-bit measurements to begin with. We conclude this section by noting there are a few works which achieve this optimal rate-distortion trade-off in particular contexts such as random frames [4, 5, 12].

1.1.3 $\Sigma\Delta$ Quantization

So far we have seen two extreme examples of quantization. Vector quantization attains an exponential decrease in distortion as a function of rate but with a very heavy computational burden for encoding. MSQ on the other hand has order optimal computational complexity but has a very poor rate-distortion trade-off. Noise shaping quantizers like $\Sigma\Delta$ quantizers serve as a compromise between these two extremes. The breakthrough comes from the following observation. Vector quantization's greatest strength and its greatest weakness is that it requires looking at all of the entries of the vector Φx simultaneously. Similarly, MSQ's simplicity is a double-edged sword. Since MSQ only looks at one entry of the vector Φx at a time it has optimal computational complexity for encoding, yet it suffers a horrible rate-distortion relationship. In order to get better

rate-distortion decay than MSQ but maintain a low computational complexity for encoding, ΣΔ quantizers quantize the entries of a vector sequentially by using a state variable to "remember" the quantization error from previous quantization steps. Given a quantization alphabet $\mathscr{A} \subset \mathbb{R}$, and a scalar quantizer $\mathscr{Q}(x) := \arg\min_{z\in\mathscr{A}} |x-z|$, r^{th} order ΣΔ quantizers often take the form of the following dynamical system:

$$\begin{aligned} u_{-r+1}, \ldots, u_0 &:= 0 \in \mathbb{R}, \\ \mathscr{Q}_{\Sigma\Delta}^{(r)}(y_t) &:= q_t := \mathscr{Q}\left(y_t + (h \star u)_t\right), \\ u_t &:= (h \star u)_t + y_t - q_t, \end{aligned}$$

where $h \in \mathbb{R}^r$ is a parameter of the quantization scheme and $(h\star u)_t = \sum_{j=1}^{r} h_j u_{t-j}$ is the convolution of h and u. From this it is clear that the encoding step for a r^{th} order ΣΔ quantizer requires on the order of rm operations, which is a modest increase from the computational complexity for encoding with MSQ. One cannot choose h willy-nilly, however. One must carefully choose h as a function of $\mathscr{A}$, r, and $\|y\|_\infty$ to ensure that the quantizer does not "saturate". In other words, ΣΔ quantizers must be carefully designed so that the state variables u_i are uniformly bounded. Such quantization schemes are called stable ΣΔ quantizers. Designing provably stable ΣΔ quantizers is a challenge in itself, but they nevertheless exist [7, 8, 13, 17]. In fact, [7] constructed a family of stable ΣΔ quantizers for all orders r where h comes from an r^{th} order difference filter, of which the simplest to describe is the case when $r = 1$. There, we have $\mathscr{A} = \{-1, 1\}$,

$$\begin{aligned} u_0 &:= 0 \in \mathbb{R}, \\ \mathscr{Q}_{\Sigma\Delta}(y_t) &:= q_t := \operatorname{sign}\left(y_t + u_{t-1}\right), \\ u_t &:= u_{t-1} + y_t - q_t. \end{aligned} \tag{1.1.1}$$

Provided $|y_t| \leq 1$ for all i, the state variable uniformly satisfies $|u_t| < 1$. The equation (1.1.1) implies that the state variable u obeys the state equation $y - q = Du$, where $D \in \mathbb{R}^{m \times m}$ is the backwards difference operator $(Du)_i = u_i - u_{i-1}$. In general, the work [7] gives stable $\Sigma\Delta$ quantization schemes where the state equations are

$$y - q = D^r u. \tag{1.1.2}$$

In the setting where $y = \Phi x$, decoding the quantized frame coefficients $q = \mathscr{Q}_{\Sigma\Delta}^{(r)}(\Phi x)$ from a stable $\Sigma\Delta$ scheme is typically achieved by applying a dual frame Ψ to the state equations (1.1.2) which yields the transformed state equations

$$\Psi\Phi x - \Psi q = \Psi D^r u \implies x - \Psi q = \Psi D^r u.$$

Using the stability of the quantization scheme, namely that $\|u\|_\infty$ is bounded, and bounding a suitable matrix norm of ΨD^r bounds the distortion [22]. There are many examples of Φ and Ψ where it is known that the distortion from decoding stable r^{th} order $\Sigma\Delta$ quantized frame coefficients decays, up to constants which may depend on r, like m^{-r}. See [15, 16, 17] for example. In the one-bit case, the rate required for m $\Sigma\Delta$ measurements is m. In other words, the rate-distortion trade-off for stable $\Sigma\Delta$ quantizers is $D_{\Sigma\Delta} \lesssim R_{\Sigma\Delta}^{-r}$. As an aside, Güntürk constructed another family of stable one-bit $\Sigma\Delta$ quantizers in [13] which uses sparse filters h whose support need not be localized unlike the difference filters used in [7]. Nevertheless, in the special case of bandlimited functions, the quantization scheme proposed in [13] attains an exponential rate-distortion trade-off provided one chooses the order of the scheme as a function of the sampling rate. Both schemes [7, 13] give dramatic improvements over the rate-distortion decay from MSQ. If we return to our extreme example that we posed at the end of the previous subsection, bounding the distortion by 10^{-16} requires on the order of $10^{16/r}$ one-bit $\Sigma\Delta$ quantized measurements. In the very modest case where we use only a second order $\Sigma\Delta$ quantizer we would need on the order

Table 1.1.1: Computational complexities and rate-distortion trade-offs for the three quantizers in the context of quantized frame theory.

Overview of Frame Quantization Methods

Method	Encoding (flops)	Decoding (flops)	Codebook (bits)	Rate-Distortion
VQ (ε-net)	$O(\varepsilon^{-N}Nm)$	$O(1)$	$O(\varepsilon^{-N}bN)$	$D \lesssim e^{-R/N}$
MSQ	$O(m)$	$O(Nm)$	$O(1)$	$D \lesssim R^{-1}$
$\Sigma\Delta$ (order r)	$O(rm)$	$O(Nm)$	$O(1)$	$D \lesssim R^{-r}$

of 10^8 one bit measurements, or equivalently 12.5 megabytes of data. Compared to the 1250 terabytes of data required for such a distortion bound from MSQ measurements, the number of one-bit $\Sigma\Delta$ measurements is paltry.

Table 1.1.1 summarizes the trade-offs between vector quantization, MSQ, and $\Sigma\Delta$ quantization in the context of frame theory. This table only scratches the surface when it comes to a full analysis of these various methods. For example, our analysis so far has not considered the more realistic setting where we have noisy quantized measurements $y = \mathscr{Q}(\Phi x + \eta)$, where $\|\eta\|_\infty$ is bounded. Further, this discussion does not consider these methods in the context of imperfect hardware which implements the quantizers. We will remark that $\Sigma\Delta$ quantizers are known to be robust to measurement noise as well as imperfect implementations of the scalar quantizer $\mathscr{Q}$ [7].

1.2 Concluding Remarks

The above discussion highlights many of the key challenges in designing encoding and decoding algorithms. Importantly, it gives explicit examples of quantization schemes where increasing the redundancy, namely the number of frame coefficients m, of the underlying signal Φx leads to better distortion bounds. As mentioned before, we will see this theme throughout this thesis. In the second chapter we will look at quantizing the weights of pretrained neural networks. Similar to the dual frame reconstruction for $\Sigma\Delta$ quantized frame coefficients, here we design a quantizer in such a way so that for a given data matrix $X \in \mathbb{R}^{m\times N_0}$ where samples are stored as rows, and a weight matrix $W \in \mathbb{R}^{N_0\times N_1}$, we will find a quantized weight matrix $Q \in \mathscr{A}^{N_0\times N_1}$ so

that $\|XW - XQ\|$ is small. In other words, we approach neural network quantization through the lens of the synthesis problem where the empirical data X form the synthesis operator. Redundancy in this context then depends on the intrinsic dimension of the feature data and the number of samples m used to learn the quantization. In the third and fourth chapters we consider $\Sigma\Delta$ quantizers in the context of compressed sensing where the goal is to recover a low-dimensional signal $x \in \mathbb{R}^N$ from quantized linear measurements. These two latter chapters take the approach of the analysis problem but we allow ourselves to use non-linear yet computationally tractable decoders. The third chapter considers the scenario where we receive $\Sigma\Delta$ quantized measurements of a low rank matrix $X \in \mathbb{R}^{n_1 \times n_2}$ and we decode these measurements by solving a convex program to enforce consistency. We then prove that if the analysis operator comes from a random subgaussian ensemble then, with high probability on the draw of this operator, uniform recovery is achieved through solving this convex program. Furthermore, we show that the recovery is robust to measurement noise and the low-rank assumption. Redundancy for recovering rank k matrices in chapter 3 is parameterized in terms of how many additional samples one takes beyond the information theoretic lower bound $k(n_1 + n_2)$. Chapter 3 ends with numerical simulations to illustrate the performance of our proposed algorithm and to compare it to the theoretical guarantees established earlier in the chapter. Finally, chapter 4 considers the setting where again we receive $\Sigma\Delta$ quantized linear measurements of signals coming from an unknown manifold in $\mathbb{R}^N$. We assume that we have a Geometric Multiresolution Analysis (GMRA) to this manifold which, intuitively, is a sequence of affine approximations to the manifold. These quantized measurements are then decoded using a convex program that leverages the GMRA to approximate the manifold. We then prove that provided the linear measurements come from a subgaussian ensemble, a partial random circulant ensemble, or a bounded orthonormal ensemble, that with high probability uniform recovery is achieved from our decoder. This enlarged class of analysis operators extends beyond the Gaussian case that a previous work considered where the encoder uses MSQ [18]. We further show that our distortion gracefully decays as a function of the oversampling factor and the

GMRA approximation error. Notably, our distortion bound decays faster than that in [18] as a function of the oversampling factor. Oversampling in this context is parameterized in terms of geometric quantities that characterize the complexity of the manifold and the GMRA. Chapter 4 ends with numerical comparisons of our algorithm against the prior work that uses MSQ.

Bibliography

[1] Open AI. *AI and Compute*, 2019 (accessed December 4, 2020). "https://openai.com/blog/ai-and-compute/".

[2] Peter G. Casazza and Gitta Kutyniok. *Finite Frames Theory and Applications*. Springer Science and Business Media, 2012.

[3] Evan Chou and C Sinan Güntürk. Distributed noise-shaping quantization: I. beta duals of finite frames and near-optimal quantization of random measurements. *Constructive Approximation*, 44(1):1–22, 2016.

[4] Zoran Cvetkovic. Resilience properties of redundant expansions under additive noise and quantization. *IEEE Transactions on Information Theory*, 49(3):644–656, 2003.

[5] Zoran Cvetkovic and Martin Vetterli. On simple oversampled a/d conversion in l/sup 2/(r). *IEEE Transactions on Information Theory*, 47(1):146–154, 2001.

[6] Sanjoy Dasgupta. *The hardness of k-means clustering*. Department of Computer Science and Engineering, University of California . . . , 2008.

[7] Ingrid Daubechies and Ron DeVore. Approximating a bandlimited function using very coarsely quantized data: A family of stable sigma-delta modulators of arbitrary order. *Annals of mathematics*, 158(2):679–710, 2003.

[8] Percy Deift, Felix Krahmer, and C Sınan Güntürk. An optimal family of exponentially accurate one-bit sigma-delta quantization schemes. *Communications on Pure and Applied Mathematics*, 64(7):883–919, 2011.

[9] Maria Emelianenko, Lili Ju, and Alexander Rand. Nondegeneracy and weak global convergence of the lloyd algorithm in r^d. *SIAM Journal on Numerical Analysis*, 46(3):1423–1441, 2008.

[10] Jerry D Gibson and Khalid Sayood. Lattice quantization. In *Advances in electronics and electron physics*, volume 72, pages 259–330. Elsevier, 1988.

[11] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[12] Vivek K Goyal, Martin Vetterli, and Nguyen T Thao. Quantized overcomplete expansions in ir/sup n: analysis, synthesis, and algorithms. *IEEE Transactions on Information Theory*, 44(1):16–31, 1998.

[13] C Sinan Güntürk. One-bit sigma-delta quantization with exponential accuracy. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 56(11):1608–1630, 2003.

[14] C Sinan Güntürk. Mathematics of analog-to-digital conversion. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 65(12):1671–1696, 2012.

[15] C Sinan Güntürk, Mark Lammers, Alex Powell, Rayan Saab, and Özgür Yilmaz. Sigma delta quantization for compressed sensing. In *2010 44th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2010.

[16] C Sinan Güntürk, Mark Lammers, Alex Powell, Rayan Saab, and Özgür Yilmaz. Sobolev duals of random frames. In *2010 44th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–5. IEEE, 2010.

[17] C Sinan Güntürk, Mark Lammers, Alexander M Powell, Rayan Saab, and Ö Yılmaz. Sobolev duals for random frames and $\sigma\delta$ quantization of compressed sensing measurements. *Foundations of Computational mathematics*, 13(1):1–36, 2013.

[18] Mark A Iwen, Felix Krahmer, Sara Krause-Solberg, and Johannes Maly. On recovery guarantees for one-bit compressed sensing on manifolds. *arXiv preprint arXiv:1807.06490*, 2018.

[19] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[20] Yoseph Linde, Andres Buzo, and Robert Gray. An algorithm for vector quantizer design. *IEEE Transactions on communications*, 28(1):84–95, 1980.

[21] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[22] Alexander Powell, Rayan Saab, and Ozgur Yilmaz. Quantization and finite frames, editor =.

[23] Michael Sabin and Robert Gray. Global convergence and empirical consistency of the generalized lloyd algorithm. *IEEE Transactions on information theory*, 32(2):148–155, 1986.

[24] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[25] Claude Elwood Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[26] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[27] Jan van den Brand. A deterministic linear program solver in current matrix multiplication time. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 259–278. SIAM, 2020.

[28] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Chapter 2

A Greedy Algorithm for Quantizing Neural Networks

We propose a new computationally efficient method for quantizing the weights of pre-trained neural networks that is general enough to handle both multi-layer perceptrons and convolutional neural networks. Our method deterministically quantizes layers in an iterative fashion with no complicated re-training required. Specifically, we quantize each neuron, or hidden unit, using a greedy path-following algorithm. This simple algorithm is equivalent to running a dynamical system, which we prove is stable for quantizing a single-layer neural network (or, alternatively, for quantizing the first layer of a multi-layer network) when the training data are Gaussian. We show that under these assumptions, the quantization error decays with the width of the layer, i.e., its level of over-parametrization. We provide numerical experiments, on multi-layer networks, to illustrate the performance of our methods on MNIST and CIFAR10 data.

2.1 Introduction

Deep neural networks have taken the world by storm. They outperform competing algorithms on applications ranging from speech recognition and translation to autonomous vehicles and even games, where they have beaten the best human players at, e.g., Go (see, [27, 14, 35, 36]). Such spectacular performance comes at a cost. Deep neural networks require a lot of computational power to train, memory to store, and power to run (e.g., [18, 23, 16, 7]). They are painstakingly trained on powerful computing devices and then either run on these powerful devices or on the cloud. Indeed, it is well-known that the expressivity of a network depends on its architecture [2]. Larger networks can capture more complex behavior [8] and therefore, for example, they generally learn better classifiers. The trade off, of course, is that larger networks require more memory for storage as well as more power to run computations. Those who design neural networks for the purpose of loading them onto a particular device must therefore account for the device's memory capacity, processing power, and power consumption. A deep neural network might yield a more accurate classifier, but it may require too much power to be run often without draining a device's battery. On the other hand, there is much to be gained in building networks directly into hardware, for example as speech recognition or translation chips on mobile or handheld devices or hearing aids. Such mobile applications also impose restrictions on the amount of memory a neural network can use as well as its power consumption.

This tension between network expressivity and the cost of computation has naturally posed the question of whether neural networks can be compressed without compromising their performance. Given that neural networks require computing many matrix-vector multiplications, arguably one of the most impactful changes would be to quantize the weights in the neural network. In the extreme case, replacing each 32-bit floating point weight with a single bit would reduce the memory required for storing a network by a factor of 32 and simplify scalar multiplications in the matrix-vector product. It is not clear at first glance, however, that there even exists a procedure

for quantizing the weights that does not dramatically affect the network's performance.

2.1.1 Contributions

The goal of this paper is to propose a framework for quantizing neural networks without sacrificing their predictive power, and to provide theoretical justification for our framework. Specifically,

- We propose a novel algorithm in (2.4.1) and (2.4.2) for sequentially quantizing layers of a pre-trained neural network in a data-dependent manner. This algorithm requires no retraining of the network, requires tuning only 2 hyperparameters—namely, the number of bits used to represent a weight and the radius of the quantization alphabet—and has a run time complexity of $O(Nm)$ operations per layer. Here, N is the ambient dimension of the inputs, or equivalently, the number of features per input sample of the layer, while m is the number of training samples used to learn the quantization. This $O(Nm)$ bound is optimal in the sense that any data-dependent quantization algorithm requires reading the Nm entries of the training data matrix. Furthermore, this algorithm is parallelizable across neurons in a given layer.

- We establish upper bounds on the relative training error in Theorem 2.5.1 and the generalization error in Theorem 2.5.2 when quantizing the first layer of a neural network that hold with high probability when the training data are Gaussian. Additionally, these bounds make explicit how the relative training error and generalization error decay as a function of the overparametrization of the data.

- We provide numerical simulations in Section 2.6 for quantizing networks trained on the benchmark data sets MNIST and CIFAR10 using both multilayer perceptrons and convolutional neural networks. We quantize all layers of the neural networks in these

numerical simulations to demonstrate that the quantized networks generalize very well even when the data are not Gaussian.

2.2 Notation

Throughout the paper, we will use the following notation. Universal constants will be denoted as C, c and their values may change from line to line. For real valued quantities x, y, we write $x \lesssim y$ when we mean that $x \leq Cy$ and $x \propto y$ when we mean $cy \leq x \leq Cy$. For any natural number $m \in \mathbb{N}$, we denote the set $\{1, \ldots, m\}$ by $[m]$. For column vectors $u, v \in \mathbb{R}^m$, the Euclidean inner product is denoted by $\langle u, v \rangle = u^T v = \sum_{j=1}^m u_j v_j$, the ℓ_2-norm by $\|u\|_2 = \sqrt{\sum_{j=1}^m u_j^2}$, the ℓ_1-norm by $\|u\|_1 = \sum_{j=1}^m |u_j|$, and the ℓ_∞-norm by $\|u\|_\infty = \max_{j \in [m]} |u_j|$. $B(x, r)$ will denote the ℓ_2-ball centered at x with radius r and we will use the notation $B_2^m := B(0, 1) \subset \mathbb{R}^m$. For a sequence of vectors $u_t \in \mathbb{R}^m$ with $t \in \mathbb{Z}$, the backwards difference operator Δ acts by $\Delta u_t = u_t - u_{t-1}$. For a matrix $X \in \mathbb{R}^{m \times N}$ we will denote the rows using lowercase characters x_t and the columns with uppercase characters X_t. For two matrices $X, Y \in \mathbb{R}^{m \times N}$ we denote the Frobenius norm by $\|X - Y\|_F := \sqrt{\sum_{i,j} |X_{i,j} - Y_{i,j}|^2}$. Φ will denote a L-layer neural network, or multilayer perceptron, which acts on data $x \in \mathbb{R}^{N_0}$ via

$$\Phi(x) := \varphi \circ A^{(L)} \circ \cdots \circ \varphi \circ A^{(1)}(x).$$

Here, $\varphi : \mathbb{R} \to \mathbb{R}$ is a rectifier which acts on each component of a vector, $A^{(\ell)}$ is an affine operator with $A^{(\ell)}(v) = v^T W^{(\ell)} + b^{(\ell)T}$ and $W^{(\ell)} \in \mathbb{R}^{N_\ell \times N_{\ell+1}}$ is the ℓ^{th} layer's weight matrix, $b^{(\ell)} \in \mathbb{R}^{N_{\ell+1}}$ is the bias.

2.3 Background

While there are a handful of empirical studies on quantizing neural networks, the mathematical literature on the subject is still in its infancy. In practice there appear to be three different paradigms for quantizing neural networks. These include quantizing the gradients during training, quantizing the activation functions, and quantizing the weights either during or after training. [15] presents an overview of these different paradigms. Any quantization that occurs during training introduces issues regarding the convergence of the learning algorithm. In the case of using quantized gradients, it is important to choose an appropriate codebook for the gradient prior to training to ensure stochastic gradient descent converges to a local minimum. When using quantized activation functions, one must suitably modify backpropagation since the activation functions are no longer differentiable. Further, enforcing the weights to be discrete during training also causes problems for backpropagation which assumes no such restriction. In any of these cases, it will be necessary to carefully choose hyperparameters and modify the training algorithm beyond what is necessary to train unquantized neural networks. In contrast to these approaches, our result allows the practitioner to train neural networks in any fashion they choose and quantizes the trained network afterwards. Our quantization algorithm only requires tuning the number of bits that are used to represent a weight and the radius of the quantization alphabet. We now turn to surveying approaches similar to ours which quantize weights after training.

A natural question to ask is whether or not for every neural network there exists a quantized representation that approximates it well on a given data set. It turns out that a partial answer to this question lies in the field of discrepancy theory. Ignoring bias terms for now, let's look at quantizing the first layer. There we have some weight matrix $W \in \mathbb{R}^{N_0 \times N_1}$ which acts on input $x \in \mathbb{R}^{N_0}$ by $x^T W$ and this quantity is then fed through the rectifier. Of course, a layer can act on a collection of $m > 0$ inputs stored as the rows in a matrix $X \in \mathbb{R}^{m \times N_0}$ where now the rectifier acts componentwise. Focusing on just one neuron w, or column of W, rather than viewing the

matrix vector product Xw as a collection of inner products $\{x_i^T w\}_{i\in[m]}$, we can think about this as a linear combination of the columns of X, namely $\sum_{t\in[N_0]} w_t X_t$. This elementary linear algebra observation now lends the quantization problem a rather elegant interpretation: is there some way of choosing quantized weights q_t from a fixed alphabet $\mathscr{A}$, such as $\{-1,0,1\}$, so that the walk $Xq = \sum_{t=1}^{N_0} q_t X_t$ approximates the walk $Xw = \sum_{t=1}^{N_0} w_t X_t$?

As we mentioned above, the study of the existence of such a q when $Xw = 0$ has a rather rich history from the discrepancy theory literature. [37] in Corollary 18 was able to prove the following surprising claim. There exists an absolute constant $c > 0$ so that given N vectors $X_1,\ldots,X_N \in \mathbb{R}^m$ with $\sup_{t\in[N]} \|X_t\|_2 \leq 1$ there exists a vector $q \in \{-1,1\}^N$ so that $\|Xw - Xq\|_\infty = \|Xq\|_\infty \leq c\log(m)$. What makes this so remarkable is that the upper bound is *independent* of N, or the number of vectors in the walk. Spencer further remarks that János Komlós has conjectured that this upper bound can be reduced to simply c. The proof of the Komlós conjecture seems to be elusive except in special cases. One special case where it is true is if we require $N < m$ and now allow $q \in \{-1,0,1\}^N$. Theorem 16 in [37] then proves that there exists universal constants $c \in (0,1)$ and $K > 0$ so that for every collection of vetors $X_1,\ldots,X_N \in \mathbb{R}^m$ with $\max_{i\in[N]} \|X_i\|_2 \leq 1$ there is some $q \in \{-1,0,1\}^N$ with $|\{i \in [N] : q_i = 0\}| < cN$ and $\|Xq\|_\infty \leq K$.

Spencer's result inspired others to attack the Komlós conjecture and variants thereof. [3] was able to prove a variant of Spencer's result for vectors X_t chosen from an ellipsoid. In the special case where the ellipsoid is the unit ball in $\mathbb{R}^m$, Banaszczyk's result says for any $X_1,\ldots,X_N \in B_2^m$ there exists $q \in \{-1,1\}^N$ so that $\|Xq\|_2 \leq \sqrt{m}$. This bound is tight, as it is achieved by the walk with $N = m$ and when the vectors X_t form an orthonormal basis. Later works consider a more general notion of boundedness. Rather than controlling the infinity or Euclidean norm one might instead wonder if there exists a bit string q so that the quantized walk never leaves a sufficiently large convex set containing the origin. The first such result, to the best of our knowledge, was proven by [12]. Giannopoulos proved there that for any origin-symmetric convex set $K \subset \mathbb{R}^m$ with standard Gaussian measure $\gamma(K) \geq 1/2$ and for any

collection of vectors $X_1, \ldots, X_m \in B_2^m$ there exists a bit string $q \in \{-1,1\}^m$ so that $Xq \in c\log(m)K$. Notice here that the number of vectors in this result is equal to the dimension. [4] strengthened this result by allowing the number of vectors to be arbitrary and further showed that, under the same assumption $\gamma(K) \geq 1/2$, there exists a $q \in \{-1,1\}^N$ which satisfies $Xq \in cK$. Scaling the hypercube appropriately, this immediately implies that the bound in Spencer's result can be reduced from $c\log(m)$ to $c\sqrt{1+\log(m)}$. Though the above results were formulated in the special case when $w = 0$, a result by [28] proves that results in this special case naturally extend to results in the *linear discrepancy* case when $\|w\|_\infty \leq 1$, though the universal constant scales by a factor of 2.

While all of these works are important contributions towards resolving the Komlós conjecture, many important questions remain, particularly pertaining to their applicability to our problem of quantizing neural networks. Naturally the most important question remains on how to construct such a q given X, w. A naïve first guess towards answering both questions would be to solve an integer least squares problem. That is, given a data set X, a neuron w, and a quantization alphabet $\mathscr{A}$, such as {-1, 1}, solve

$$\begin{aligned} \underset{q}{\text{minimize}} \quad & \|Xw - Xq\|_2^2 \\ \text{subject to} \quad & q_i \in \mathscr{A}, \ \ i = 1, \ldots, m. \end{aligned} \tag{2.3.1}$$

It is well-known, however, that solving (2.3.1) is NP-Hard. See, for example, [1]. Nevertheless, there have been many iterative constructions of vectors $q \in \{-1,1\}^N$ which satisfy the bounds in the aforementioned works. A non-comprehensive list of such works includes [5, 29, 34, 19, 11]. Constructions of q which satisfy the bound in the result of [4] include the works of [9, 6]. These works also generalize to the linear discrepancy setting. In fact, [6] prove a much more general result which allows the use of more arbitrary alphabets other than $\{-1,1\}$. Their algorithm is random though, so their result holds with high probability on the execution of the algorithm. This is in contrast, as we will see, with our result which will hold with high probability on the

draw of Gaussian data. Beyond this, the computational complexity of the algorithms in [9, 6] prohibit their use in quantizing deep neural networks. For [9], this consists of looping over $O(N_0^5)$ iterations of solving a semi-definite program and computing a Cholesky factorization for a $N_0 \times N_0$ matrix. The Gram-Schmidt walk algorithm in [6] has a run-time complexity of $O(N_0(N_0+m)^\omega)$, where $\omega \geq 2$ is the exponent for matrix multiplication. These complexities are already quite restrictive and only give the run-time for quantizing a single neuron. As the number of neurons in each layer is likely to be large for deep neural networks, these algorithms are simply infeasible for the task at hand. As we will see, our algorithm in comparison has a run-time complexity of $O(N_0 m)$ per neuron which is optimal in the sense that any data driven approach towards constructing q will require one pass over the $N_0 m$ entries of X. Using a norm inequality on Banasczyzk's bound, the result in [5] guarantees for $\|w\|_\infty \leq 1$ the existence of a q such that $\|Xw - Xq\|_2 \leq c\sqrt{m\log(m)}$. Provided w is a generic vector in the hypercube, namely that $\|w\|_2 \propto \sqrt{N_0}$, then a simple calculation shows that with high probability on the draw of Gaussian data X with entries having variance $1/m$ to ensure that the columns are approximately unit norm, the Gram-Schmidt walk achieves a relative error bound of $\frac{\|Xw-Xq\|_2}{\|Xw\|_2} \lesssim \sqrt{m\log(m)/N_0}$. As we will see in Theorem 2.5.1, our relative training error bound for quantizing neurons in the first layer decays like $\log(N_0)\sqrt{m/N_0}$. In other words, to achieve a relative error of less than ε in the overparametrized regime where $N_0 \gg m$, the Gram-Schmidt walk requires on the order of $\frac{m^3\log^3(m)}{\varepsilon^6}$ floating point operations as compared to our algorithm which only requires on the order of $\frac{m^2}{\varepsilon^2}$ floating point operations.

With no quantization algorithm that is both competitive from a theoretical perspective and computationally feasible, we turn to surveying what has been done outside the mathematical realm. Perhaps the simplest manner of quantizing weights is to quantize each weight within each neuron independently. The authors in [33] consider precisely this set-up in the context of convolutional neural networks (CNNs). For each weight matrix $W^{(\ell)} \in \mathbb{R}^{N_\ell \times N_{\ell+1}}$ the quantized weight matrix $Q^{(\ell)}$ and optimal scaling factor α_ℓ are defined as minimizers of $\|W^{(\ell)} - \alpha Q\|_F^2$ subject to the

constraint that $Q_{i,j} \in \{-1,1\}$ for all i,j. It turns out that the analytical solution to this optimization problem is $Q_{i,j}^{(\ell)} = \text{sign}(W_{i,j}^{(\ell)})$ and $\alpha_\ell = \frac{1}{mn}\sum_{i,j}|W_{i,j}^{(\ell)}|$. This form of quantization has long been known to the digital signal processing community as Memoryless Scalar Quantization (MSQ) because it quantizes a given weight independently of all other weights. While MSQ may minimize the Euclidean distance between two weight matrices, we will see that it is far from optimal if the concern is to design a matrix Q which approximates W on an overparameterized data set.

In a similar vein, [39] consider learning a quantized factorization of the weight matrix $W = XDY$, where the matrices X,Y are ternary matrices with entries $\{-1,0,1\}$ and D is a full-precision diagonal matrix. While in general this is a NP-hard problem authors use a greedy approach for constructing X,Y,D inspired by the work of [25]. They provide simulations to demonstrate the efficacy of this method on a few pre-trained models, yet no theoretical analysis is provided for the efficacy of this framework. We would like to remark that the work [26] gives a framework for computing factorizations of W when $\text{rank}(W) = r$ as $W = SA \in \mathbb{R}^{n\times m}$, where $S \in \{-1,1\}^{n\times r}, A \in \mathbb{R}^{r\times m}$. The reason this work is intruiging is that it does offer a means for compressing the weight matrix W by storing a smaller analog matrix A and a binarized matrix S though it does not offer nearly as much compression as if we were to replace W by a fully quantized matrix Q. Indeed, the matrix A is not guaranteed to be binary or admit a representation with a low-complexity quantization alphabet. Nevertheless, [26] give conditions under which such a factorization exists and propose an algorithm which provably constructs S,A using semi-definite programming. They extend this analysis to the case when W is the sum of a rank r matrix and a sparse matrix but do not establish robustness of their factorization to more general noise models.

Extending beyond the case where the quantization alphabet is fixed a priori, [13] propose learning a codebook through vector quantization to quantize only the dense layers in a convolutional neural network. This stands in contrast to our work where we quantize all layers of a network. They consider clustering weights using k-means clustering and using the centroids of the clusters as the quantized weights. Moreover, they consider three different methods of clustering,

which include clustering the neurons as vectors, groups of neurons thought of as sub-matrices of the weight matrix, and quantizing the neurons and the successive residuals between the cluster centers and the neurons. Beyond the fact that this work does not consider quantizing the convolutional layers, there is the additional shortcoming that clustering the neurons or groups thereof requires choosing the number of clusters in advance and choosing a maximal number of iterations to stop after. Our algorithm gives explicit control over the alphabet in advance, requires tuning only the radius of the quantization alphabet, and runs in a fixed number of iterations. Similar to the above work, we make special mention of Deep Compression by [18]. Deep Compression seems to enjoy compressing large networks like AlexNet without sacrificing their empirical performance on data sets like ImageNet. There, authors consider first pruning the network and quantizing the values of the (scalar-valued) weights in a given layer using k-means clustering. This method applies both to fully connected and convolutional layers. An important drawback of this quantization procedure is that the network must be retrained, perhaps multiple times, to fine tune these learned parameters. Once these parameters have been fine tuned, the weight clusters for each layer are further compressed using Huffman coding. We further remark that quantizing in this fashion is sensitive to the initialization of the cluster weights.

2.4 Algorithm and Intuition

Going forward we will consider neural networks without bias vectors. This assumption may seem restrictive, but in practice one can always use MSQ with a big enough bit budget to control the quantization error for the bias. Even better, one may simply embed the m dimensional data/activations x and weights w into an $m+1$ dimensional space via $x \mapsto (x,1)$ and $w \mapsto (w,b)$ so that $w^T x + b = (w,b)^T (x,1)$. In other words, the bias term can simply be treated as an extra dimension to the weight vector, so we will henceforth ignore it. Given a trained neural network Φ with its associated weight matrices $W^{(\ell)}$ and a data set $X \in \mathbb{R}^{m \times N_0}$, our goal is to construct

quantized weight matrices $Q^{(\ell)}$ to form a new neural network $\widetilde{\Phi}$ for which $\|\Phi(X)-\widetilde{\Phi}(X)\|_F$ is small. For simplicity and ease of exposition, we will focus on the extreme case where the weights are constrained to the ternary alphabet $\{-1,0,1\}$, though there is no reason that our methods cannot be applied to arbitrary alphabets.

Our proposed algorithm will quantize a given neuron independently of other neurons. Beyond making the analysis easier this has the practical benefit of allowing us to easily parallelize quantization across neurons in a layer. If we denote a neuron as $w \in \mathbb{R}^{N_\ell}$, we will sucessively quantize the weights in w in a greedy data-dependent way. Let $X \in \mathbb{R}^{m\times N_0}$ be our data matrix, and let $\Phi^{(\ell-1)}, \widetilde{\Phi}^{(\ell-1)}$ denote the analog and quantized neural networks up to layer $\ell-1$ respectively.

In the first layer, the aim is to achieve $Xq = \sum_{t=1}^{N_0} q_t X_t \approx \sum_{t=1}^{N_0} w_t X_t = Xw$ by selecting, at the t^{th} step, q_t so the running sum $\sum_{j=1}^{t} q_j X_j$ tracks its analog $\sum_{j=1}^{t} w_j X_j$ as well as possible in an ℓ_2 sense. That is, at the t^{th} iteration, we set

$$q_t := \underset{p\in\{-1,0,1\}}{\arg\min} \| \sum_{j=1}^{t} w_j X_j - \sum_{j=1}^{t-1} q_j X_j - pX_t\|_2^2.$$

It will be more amenable to analysis, and to implementation, to instead consider the equivalent dynamical system where we quantize neurons in the first layer using

$$\begin{aligned} u_0 &:= 0 \in \mathbb{R}^m, \\ q_t &:= \underset{p\in\{-1,0,1\}}{\arg\min} \|u_{t-1} + w_t X_t - pX_t\|_2^2, \\ u_t &:= u_{t-1} + w_t X_t - q_t X_t. \end{aligned} \tag{2.4.1}$$

One can see, using a simple substitution, that $u_t = \sum_{j=1}^{t}(w_j X_j - q_j X_j)$ is the error vector at the t^{th} step. Controlling it will be a main focus in our error analysis. An interesting way of thinking about (2.4.1) is by imagining the analog, or unquantized, walk as a drunken walker and the quantized walk is a concerned friend chasing after them. The drunken walker can stumble with

step sizes w_t along an avenue in the direction X_t but the friend can only move in steps whose lengths are encoded in the alphabet $\mathscr{A}$.

In the subsequent hidden layers, we follow a slightly modified version of (2.4.1). Letting $Y := \Phi^{(\ell-1)}(X)$, $\widetilde{Y} := \widetilde{\Phi}^{(\ell-1)}(X) \in \mathbb{R}^{m\times N_\ell}$, we quantize neurons in layer ℓ by

$$\begin{aligned} u_0 &:= 0 \in \mathbb{R}^m, \\ q_t &:= \operatorname*{arg\,min}_{p\in\{-1,0,1\}} \|u_{t-1} + w_t Y_t - p\widetilde{Y}_t\|_2^2, \\ u_t &:= u_{t-1} + w_t Y_t - q_t \widetilde{Y}_t. \end{aligned} \tag{2.4.2}$$

We say the vector $q \in \mathbb{R}^{N_\ell}$ is the quantization of w. In this work, we will provide a theoretical analysis for the behavior of (2.4.1) and leave analysis of (2.4.2) for future work. To that end, we re-emphasize the critical role played by *the state variable* u_t defined in (2.4.1). Indeed, we have the identity $\|Xw - Xq\|_2 = \|u_{N_0}\|_2$. That is, the two neurons w, q act approximately the same on the batch of data X only provided the state variable $\|u_{N_0}\|_2$ is well-controlled. Given bounded input $\{(w_t, X_t)\}_t$, systems which admit uniform upper bounds on $\|u_t\|_2$ will be referred to as *stable*. When the X_t are random, and in our theoretical considerations they will be, we remark that this is a much stronger statement than proving convergence to a limiting distribution which is common, for example, in the Markov chain literature. For a broad survey of such Markov chain techniques, one may consult [31]. The natural question remains: is the system (2.4.1) stable? Before we dive into the machinery of this dynamical system, we would like to remark that there is a concise form solution for q_t. Denote the greedy ternary quantizer by $\mathscr{Q} : \mathbb{R} \to \{-1, 0, 1\}$ with

$$\mathscr{Q}(z) = \operatorname*{arg\,min}_{p\in\{-1,0,1\}} |z - p|.$$

Then we have the following.

Lemma 2.4.1. In the context of (2.4.1), we have for any $X_t \neq 0$ that

$$q_t = \mathscr{Q}\left(w_t + \frac{X_t^T u_{t-1}}{\|X_t\|_2^2}\right). \tag{2.4.3}$$

Proof. This follows simply by completing a square. Provided $X_t \neq 0$, we have by the definition of q_t

$$\begin{aligned} q_t &= \operatorname*{arg\,min}_{p\in\{-1,0,1\}} \|u_{t-1} + (w_t - p)X_t\|_2^2 = \operatorname*{arg\,min}_{p\in\{-1,0,1\}} (w_t-p)^2 + 2(w_t-p)\frac{X_t^T u_{t-1}}{\|X_{t-1}\|_2^2} \\ &= \operatorname*{arg\,min}_{p\in\{-1,0,1\}} \left((w_t - p) + \frac{X_t^T u_{t-1}}{\|X_{t-1}\|_2^2}\right)^2 - \left(\frac{X_t^T u_{t-1}}{\|X_{t-1}\|_2^2}\right)^2. \end{aligned}$$

Because the former term is always non-negative, it must be the case that the minimizer is $\mathscr{Q}\left(w_t + \frac{X_t^T u_{t-1}}{\|X_t\|_2^2}\right)$. □

Any analysis of the stability of (2.4.1) must necessarily take into account how the vectors X_t are distributed. Indeed, one can easily cook up examples which give rise to sequences of u_t which diverge rapidly. For the sake of illustration consider restricting our attention to the case when $\|X_t\|_2 = 1$ for all t. The triangle inequality gives us the crude upper bound

$$\|X(w-q)\|_2 \leq \sum_{t=1}^{N_0} |w_t - q_t| \|X_t\|_2 = \|w-q\|_1.$$

Choosing q to minimize $\|w-q\|_1$, or any p-norm for that matter, simply reduces back to the MSQ quantizer where the weights within w are quantized independently of one another, namely $q_t = \mathscr{Q}(w_t)$. It turns out that one can effectively attain this upper bound by adversarially choosing X_t to be orthogonal to u_{t-1} for all t. Indeed, in that setting we have exactly the MSQ quantizer

$$\begin{aligned} q_t &:= \mathscr{Q}\left(w_t + X_t^T u_{t-1}\right) = \mathscr{Q}(w_t), \\ u_t &:= u_{t-1} + (w_t - q_t)X_t. \end{aligned}$$

Consequentially, by repeatedly appealing to orthogonality,

$$\|u_t\|_2^2 = \|u_{t-1} + (w_t - q_t)X_t\|_2^2 = \|u_{t-1}\|_2^2 + (w_t - q_t)^2\|X_t\|_2^2 = \sum_{j=1}^{t}(w_j - q_j)^2.$$

Thus, for generic vectors w, and adversarially chosen X_t, the error $\|u_t\|_2$ scales like $\sqrt{t}$. Importantly, this adversarial construction requires knowledge of u_{t-1} at "time" t, in order to construct an orthogonal X_t. In that sense, this extreme case is rather contrived. In an opposite (but also contrived) extreme case, all of the X_t are equal, and therefore X_t is parallel to u_{t-1} for all t, the dynamical system reduces to a first order greedy $\Sigma\Delta$ quantizer []

$$\begin{aligned} q_t &= \mathscr{Q}\left(w_t + X_t^T u_{t-1}\right) = \mathscr{Q}\left(w_t + \sum_{j=1}^{t-1} w_j - q_j\right), \\ u_t &= u_{t-1} + (w_t - q_t)X_t. \end{aligned} \tag{2.4.4}$$

Here, when $w_t \in [-1,1]$, one can show by induction that $\|u_t\|_2 \leq 1/2$ for all t, a dramatic contrast with the previous scenario. For more details on $\Sigma\Delta$ quantization, see for example [21, 10].

Recall that in the present context the signal we wish to approximate is not the neuron w itself, but rather Xw. The goal therefore is not to minimize the error $\|w-q\|_2$ but rather to minimize $\|X(w-q)\|_2$, which by construction is the same as $\|u_{N_0}\|_2$. Algebraically that means carefully selecting q so that $w-q$ is in or very close to the kernel, or null-space, of the data matrix X. This immediately suggests how overparameterization may lead to better quantization. Given m data samples stored as rows in $X \in \mathbb{R}^{m\times N_0}$, having $N_0 \gg m$ or alternatively having $\dim(\text{Span}\{x_1,\ldots,x_m\}) \ll N_0$ ensures that the kernel of X is large, and one may attempt to design q so that the vector $w-q$ lies as close as possible to the kernel of X.

2.5 Main Results

We are now ready to state our main result which shows that (2.4.1) is stable when the input data X are Gaussian. The proofs of the following theorems are deferred to Section 2.9, as the proofs are quite long and require many supporting lemmata.

Theorem 2.5.1. Suppose $X \in \mathbb{R}^{m\times N_0}$ has independent columns $X_t \sim \mathcal{N}(0,\sigma^2 I_{m\times m})$, $w \in \mathbb{R}^{N_0}$ is independent of X and satisfies $w_t \in [-1,1]$ and $\text{dist}(w_t,\{-1,0,1\}) > \varepsilon$ for all t. Then, with probability at least $1-C\exp(-cm\log(N_0))$ on the draw of the data X, if q is selected according to (2.4.1) we have that

$$\frac{\|Xw-Xq\|_2}{\|Xw\|_2} \lesssim \frac{\sqrt{m}\log(N_0)}{\|w\|_2}, \tag{2.5.1}$$

where $C,c>0$ are constants that depend on ε in a manner that is made explicit in the statement of Theorem 2.9.1.

Proof. Without loss of generality, we'll assume $\sigma = 1/\sqrt{m}$ since this factor appears in both numerator and denominator of (2.5.1). Theorem 2.9.1 guarantees with probability at least $1-Ce^{-cm\log(N_0)}$ that $\|u_{N_0}\|_2 = \|Xw-Xq\|_2 \lesssim \sqrt{m}\log(N_0)$. Using Lemma 2.8.2, we have $\|Xw\|_2 \gtrsim \|w\|_2$ with probability at least $1-2\exp(-c_{norm}m)$. Combining these two results gives us the desired statement. □

For generic vectors w we have $\|w\|_2 \propto \sqrt{N_0}$, so in this case Theorem 2.5.1 tells us that up to logarithmic factors the relative error decays like $\sqrt{m/N_0}$. As it stands, this result suggests that it is sufficient to have $N_0 \gg m$ to obtain a small relative error. In Section 2.9, we address the case where the feature data X_t lay in a d-dimensional subspace to get a bound in terms of d rather than m. In other words, this suggests that the relative training error depends not on the number of training samples m but on the intrinsic dimension of the features d. See Lemma 2.9.2 for details.

Our next result shows that the quantization error is well-controlled in the span of the training data so that the quantized weights generalize to new data.

Theorem 2.5.2. Define X, w and q as in the statement of Theorem 2.5.1 and further suppose that $N_0 \gg m$. Let $X = U\Sigma V^T$ be the singular value decomposition of X, and let $z = Vg$ where $g \sim \mathcal{N}(0, \sigma_z^2 I_{m\times m})$ is drawn independently of X, w. In other words, suppose z is a Gaussian random variable drawn from the span of the training data x_i. Then with probability at least $1 - Ce^{-cm\log(N_0)} - 3\exp(-c''m)$ we have

$$|z^T(w-q)| \lesssim \left(\frac{\sigma_z m}{\sigma(\sqrt{N_0}-\sqrt{m})}\right)\sigma m\log(N_0). \tag{2.5.2}$$

Proof. To begin, notice that the error bound in Theorem 2.9.1 easily extends to the set $X^T(B_1^{N_0}) := \{y \in \mathbb{R}^{N_0} : y = \sum_{i=1}^m a_i x_i,\ \|a\|_1 \leq 1\}$ with a simple yet pessimistic argument. With probability at least $1 - Ce^{-cm\log(N_0)}$, for any $y \in X^T(B_1^{N_0})$ one has

$$\begin{aligned} |y^T(w-q)| &= \left|\sum_{i=1}^m a_i x_i^T(w-q)\right| \leq \sum_{i=1}^m |a_i||x_i^T(w-q)| \\ &\lesssim \sum_{i=1}^m |a_i|\sigma m\log(N_0) \leq \sigma m\log(N_0). \end{aligned} \tag{2.5.3}$$

Now for z as defined in the statement of this theorem define $p := \alpha^* z$, where

$$\begin{aligned} \alpha^* := \arg\max_{\alpha\geq 0} \quad & \alpha \\ \text{subject to} \quad & \alpha z \in X^T(B_1^{N_0}). \end{aligned}$$

If it were the case that $\alpha^* > 0$ then we could use (2.5.3) to get the bound

$$|z^T(w-q)| = \frac{1}{\alpha^*}\|p^T X(w-q)\|_2 \lesssim \frac{\sigma m\log(N_0)}{\alpha^*}.$$

So, it behooves us to find a strictly positive lower bound on α^*. By the assumption that $z = Vg$,

there exists $h \in \mathbb{R}^m$ so that $X^T h = z$. Since $N_0 > m$, X^T is injective almost surely and therefore h is unique. Setting $v := \|h\|_1^{-1} h$, observe that $X^T v = \|h\|_1^{-1} z$ and $\sum_{i=1}^m v_i = 1$. It follows that $\alpha^* \geq \|h\|_1^{-1}$. To lower bound $\|h\|_1^{-1}$, note

$$\begin{aligned}\|h\|_1^{-1}\|z\|_2 = \|X^T v\|_2 &\geq \min_{\|y\|_1=1} \|X^T y\|_2 \geq \left(\min_{\|\eta\|_2=1} \|X^T \eta\|_2\right) \min_{\|y\|_1=1} \|y\|_2 \\ &\gtrsim \sigma(\sqrt{N_0} - \sqrt{m}) \min_{\|y\|_1=1} \|y\|_2 = \frac{\sigma(\sqrt{N_0} - \sqrt{m})}{\sqrt{m}}. \end{aligned} \tag{2.5.4}$$

The penultimate inequality in the above equation follows directly from well-known bounds on the singular values of isotropic subgaussian matrices that hold with probability at least $1 - 2\exp(-c'm)$ (see [38]). To make the argument explicit, note that $X^T = \sigma G$, where $G \in \mathbb{R}^{N_0 \times m}$ is a matrix whose rows are independent and identically distributed gaussians with $\mathbb{E}[g_i g_i^T] = I_{m\times m}$ and are thus isotropic. Using Lemma 2.8.2 we have with probability at least $1 - \exp(-c_{norm} m/4)$ that $\|z\|_2 = \|Vg\|_2 = \|g\|_2 \lesssim \sigma_z \sqrt{m}$. Substituting in (2.5.4), we have

$$\|h\|_1^{-1} \gtrsim \frac{\sigma(\sqrt{N_0} - \sqrt{m})}{\sigma_z m}.$$

Therefore, putting it all together, we have with probability at least $1 - Ce^{-cm\log(N_0)} - 3\exp(-c''m)$

$$|z^T(w-q)| \lesssim \left(\frac{\sigma_z m}{\sigma(\sqrt{N_0} - \sqrt{m})}\right) \sigma m \log(N_0).$$

□

Remark 1. In the special case when $\sigma_z = \sigma\sqrt{N_0/m}$, i.e. when $\mathbb{E}[\|z\|_2^2 | V] = \mathbb{E}\|x_i\|_2^2 = \sigma^2 N_0$ and $N_0 \gg m$, the bound in Theorem 2.5.2 reduces to

$$\frac{\sigma\sqrt{N_0 m}}{\sigma(\sqrt{N_0} - \sqrt{m})} \sigma m \log(N_0) \lesssim \sigma m^{3/2} \log(N_0).$$

Furthermore, when the row data are normalized in expectation, or when $\sigma^2 = N_0^{-1}$, this bound

becomes $\frac{m^{3/2}\log(N_0)}{\sqrt{N_0}}$.

Remark 2. Under the low-dimensional assumptions in Lemma 2.9.2, the bound (2.5.2) and the discussion in Remark 1 apply when m is replaced with d.

Our technique for showing the stability of (2.4.1), i.e., the boundedness of $\|X(w-q)\|_2$, relies on tools from drift analysis. Our analysis is inspired by the works of [32] and [17]. Given a real valued stochastic process $\{Y_t\}_{t\in\mathbb{N}}$, those authors give conditions on the increments $\Delta Y_t := Y_t - Y_{t-1}$ to uniformly bound the moments, or moment generating function, of the iterates Y_t. These bounds can then be transformed into a bound in probability on an individual iterate Y_t using Markov's inequality. Recall that we're interested in bounding the state variable u_t induced by the system (2.4.1) which quantizes the first layer of a neural network. In situations like ours it is natural to analyze the increments of u_t since the innovations (w_t, X_t) are jointly independent. To invoke the results of [32, 17] we'll consider the associated stochastic process $\{\|u_t\|_2^2\}_{t\in[N_0]}$. Beyond the fact that our intent is to control the norm of the state variable, it turns out that stability analyses of vector valued stochastic processes typically involve passing the process through a real-valued and oftentimes quadratic function known as a Lyapunov function. There is a wide variety of stability theorems which require demonstrating certain properties of the image of a stochastic process under a Lyapunov function. For example, Lyapunov functions play a critical role in analyzing Markov chains as detailed in [30]. However, there are a few details which preclude us from using one of these well-known stability results for the process $\{\|u_t\|_2^2\}_{t\in[N_0]}$. First, even though the innovations (w_t, X_t) are jointly independent the increments

$$\Delta\|u_t\|_2^2 = (w_t - q_t)^2\|X_t\|_2^2 + 2(w_t - q_t)\langle X_t, u_{t-1}\rangle \tag{2.5.5}$$

have a dependency structure encoded by the bit sequence q. In addition to this, the bigger challenge in the analysis of (2.4.1) is the discontinuity inherent in the definition of q_t. Addressing this discontinuity in the analysis requires carefully handling the increments on the events where

q_t is fixed.

Based on our prior discussion, towards the end of Section 2.4, it would seem that for generic data sets the stability of (2.4.1) lies somewhere in between the behavior of MSQ and $\Sigma\Delta$ quantizers, and that behavior crucially depends on the "dither" terms $X_t^T u_{t-1}$. For the sake of analysis then, we will henceforth make the following assumptions.

Assumption 1. The sequence $(w_t, X_t)_t$ defined on the probability space $(\Omega, \mathscr{F}, \mathbb{P})$ is adapted to the filtration $\mathscr{F}_t$. Further, all X_t and w_t are jointly independent.

Assumption 2. $\|W^{(\ell)}\|_\infty = \sup_{i,j} |W_{i,j}^{(\ell)}| \leq 1$.

Assumption 1's stipulation that the X_t are independent of the weights is a simplifying relaxation, and our proof technique handles the case when the w_t are deterministic. The joint independence of the X_t could be realized by splitting the global population of training data into two populations where one is used to train the analog network and another to train the quantization. In the hypotheses of Theorem 2.5.1 it is also assumed that the entries of the weight vector w_t are sufficiently separated from the characters of the alphabet $\{-1, 0, 1\}$. We want to remark that this is simply an artifact of the proof. In succinct terms, the proof strategy relies on showing that the moment generating function of the increment $\Delta\|u_t\|_2^2$ is strictly less than 1 conditioned on the event that $\|u_{t-1}\|_2$ is sufficiently large. In the extreme case where the weights are already quantized to $\{-1, 0, 1\}$, this aforementioned event is the empty set since the state variable u_t is identically the zero function. As such, the conditioning is ill-defined. To avoid this technicality, we assume that the neural network we wish to quantize is not already quantized, namely $\text{dist}(w_t, \{-1, 0, 1\}) > \varepsilon$ for some $\varepsilon > 0$ and for all $t \in [N_\ell]$. The proof technique could easily be adapted to the case where the w_t are deterministic and this hypothesis is violated for $O(1)$ weights with only minor changes to the main result, but we do not include these modifications to keep the exposition as clear as possible. Assumption 2 is quite mild, and can be realized by scaling all neurons in a given layer by $\|W\|_\infty^{-1}$. Choosing the ternary vector q according to the scaled neuron $\|W\|_\infty^{-1} w$, any bound of

the form $\|X(\|W\|_\infty^{-1}w-q)\|_2 \leq \alpha$ immediately gives the bound $\|X(w-\|W\|_\infty q)\|_2 \leq \alpha\|W\|_\infty$. In other words, at run time the network can use the scaled ternary alphabet $\{-\|W\|_\infty, 0, \|W\|_\infty\}$.

2.6 Numerical Simulations

We present two stylized examples which show how our proposed quantization algorithm affects classification accuracy on two benchmark data sets. In the following tables and figures, we'll refer to our algorithm as Greedy Path Following Quantization, or GPFQ for short. We look at classifying digits from the MNIST data set using a multilayer perceptron as well as classifying images from the CIFAR10 data set using a convolutional neural network. We trained both networks using Keras with the Tensorflow backend on a 2014 iMac with 32GB of RAM. Beyond this our hardware was limited, so we did not train on a GPU and therefore the depths and widths of both networks were kept modest. Note that our aim here is no not to match state of the art results in training the unquantized neural networks. Rather, our goal is to demonstrate that given a trained neural network, our quantization algorithm yields a network that performs similarly. Below, we mention our design choices for the sake of completeness, and to demonstrate that our quantization algorithm does not require any special engineering beyond what is customary in neural network architectures. We have made our code available on GitHub at `https://github.com/elybrand/quantized_neural_networks`.

Our implementations for these simulations differ from the presentation of the theory in a few ways. First, we do not restrict ourselves to the particular ternary alphabet of $\{-1,0,1\}$. In practice, it is much more useful to replace this with the equispaced alphabet $\mathscr{A} = \alpha \times \{-1+\frac{2j}{M-1} : j \in \{0,1,\ldots,M-1\}\} \subset [-\alpha,\alpha]$, where M is fixed in advance and α is chosen by cross-validation. Of course, this includes the ternary alphabet $\{-\alpha,0,\alpha\}$ as a special case. The intuition behind choosing the alphabet's radius α is to better capture the dynamic range of the true weights. For this reason we choose for every layer $\alpha_\ell = C_\alpha \text{median}(\{|W_{i,j}^{(\ell)}|\}_{i,j})$ where the constant C_α is fixed

for all layers and is chosen by cross-validation. Thus, the cost associated with allowing general alphabets $\mathscr{A}$ is storing a floating point number for each layer (i.e., α_ℓ) and $N_\ell \times N_{\ell+1}$ bit strings of length $\log_2(2M+1)$ per layer as compared to $N_\ell \times N_{\ell+1}$ floats per layer in the unquantized setting.

2.6.1 Multilayer Perceptron with MNIST

We trained a multilayer perceptron to classify MNIST digits (28×28 images) with two hidden layers. The first layer has 500 neurons, the second has 300 neurons, and the output layer has 10 neurons. We also used batch normalizations [22] after each hidden layer and chose the ReLU activation function for all layers except the last where we used softmax. We trained the unquantized network on the full training set of $60,000$ digits without any preprocessing. 20% of the training data was used as validation during training. We then tested on the remaining $10,000$ images not used in the training set. We used categorical cross entropy as our loss function during training and the Adam optimizer—see [24]—for 100 epochs with a minibatch size of 128. After training the unquantized model we used $25,000$ samples from the training set to train the quantization. We used the same data to quantize each layer rather than splitting the data for each layer. For this experiment we restricted the alphabet to be ternary and only cross-validated over the alphabet scalar $C_\alpha \in \{1,2,\ldots,10\}$. The results for each choice of C_α are displayed in Figure 2.6.1a. As a benchmark we compared against a network quantized using MSQ, so each weight was quantized to the element of $\mathscr{A}$ that is closest to it. As we see in Figure 2.6.1a, the MSQ quantized network exhibits a high variability in its performance as a function of the alphabet scalar, whereas the GPFQ quantized network exhibits more stable behavior. Indeed, for a number of consecutive choices of C_α the performance of the GPFQ quantized network was close to its unquantized counterpart. To illustrate how accuracy was affected as subsequent layers were quantized, we ran the following experiment. First, we chose the best alphabet scalar C_α for each of the MSQ and GPFQ quantized networks separately. We then measured the test accuracy

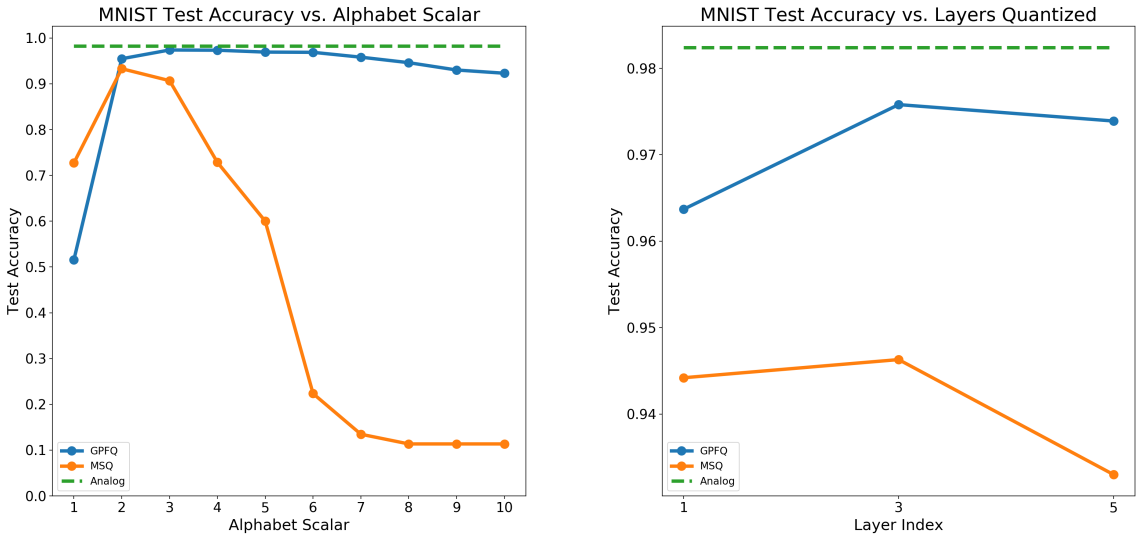
as each subsequent layer of the network was quantized, leaving the later ones unchanged. The results for MSQ and GPFQ are shown in Figure 2.6.1b. Figure 2.6.1b demonstrates that GPFQ is able to “error correct" in the sense that quantizing a later layer can correct for errors introduced when quantizing previous ones. We also remark that in this setting we replace 32 bit floating point weights with $\log_2(3)$ bit weights. Consequentially, we have compressed the network by a factor of approximately 20, and yet the drop in test accuracy for GPFQ was minimal. Further, this quick calculation assumes we use $\log_2(3)$ bits to represent those weights which are quantized to zero. However, there are other important consequences for setting weights to zero. From a hardware perspective, the benefit is that forward propagation requires less energy due to there being fewer connections between layers. From a software perspective, multiplication by zero is an incredibly stable operation.

2.6.2 Convolutional Neural Network with CIFAR10

Even though our theory was phrased in the language of multilayer perceptrons it is easy to rephrase it using the vocabulary of convolutional neural networks. Here, neurons are kernels and the data are patches from the full images or their feature data in the hidden layers. These patches have the same dimensions as the kernel. Matrix convolution is defined in terms of Hilbert-Schmidt inner products between the kernel and these image patches. In other words, if we were to vectorize both the kernel and the image patches then we could take the usual inner product on vectors and reduce back to the case of a multilayer perceptron. This is exactly what we do in the quantization algorithm. Since every channel of the feature data has its own kernel we quantize each channel’s kernel independently.

We trained a convolutional neural network to classify images from the CIFAR10 data set with the following architecture

$$2 \times 32C3 \to MP2 \to 2 \times 64C3 \to MP2 \to 2 \times 128C3 \to 128FC \to 10FC.$$



(a) (b)

Figure 2.6.1: Comparison of GPFQ and MSQ quantized network performance on MNIST. Figure 2.6.1a plots the top-1 accuracy against scalars C_α. Figure 2.6.1b plots how each quantized network behaves as layers are quantized using the best C_α for each network. Only fully-connected layer indices are plotted.

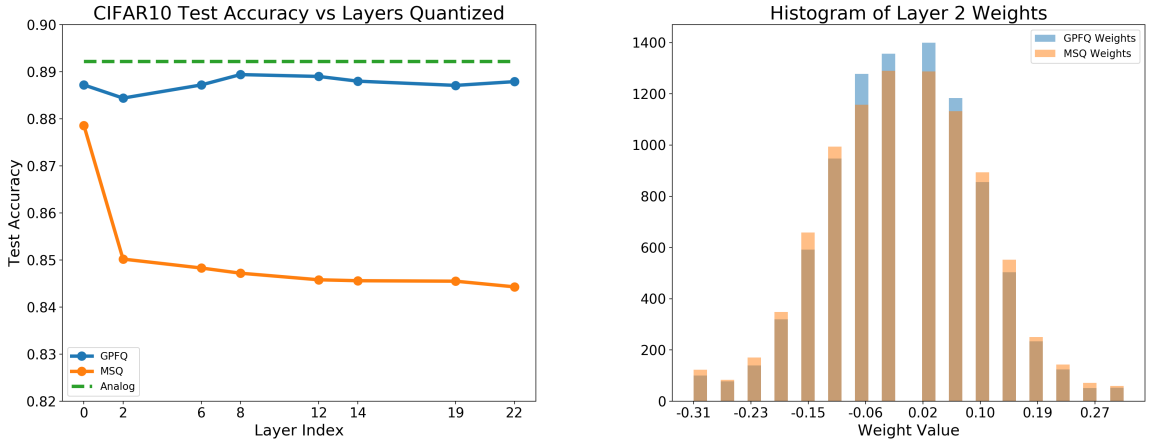
Here, $2 \times N$ C3 denotes two convolutional layers with N kernels of size 3×3, MP2 denotes a max pooling layer with kernels of size 2×2, and FC denotes a fully connected layer. Not listed in the above schematic are batch normalization layers which we place before every convolutional and fully connected layer except the first. During training we also use dropout layers after the max pooling layers and before the final output layer. We use the ReLU function for every layer's activation function except the last layer where we use softmax. We preprocess the data by dividing the pixel values by 255 which normalizes them in the range $[0,1]$. We augment the data set with width and height shifts as well as horizontal flips for each image. Finally, we train the network to minimize categorical cross entropy using stochastic gradient descent with a learning rate of 10^{-4}, momentum of 0.9, and a minibatch size of 64 for 400 epochs. For more information on dropout layers and pooling layers see, for example, [20] and [40], respectively.

We trained the unquantized network on the full set of $50,000$ training images. For training the quantization we only used the first $5,000$ images from the training set. As we did with the multilayer perceptron on MNIST, we cross-validated the alphabet scalars C_α over the range $\{2,3,4,5,6\}$ and chose the best scalar for the benchmark MSQ network and the best GPFQ quantized network separately. Additionally, we cross-validated over the number of elements in the quantization alphabet, ranging over the set $M \in \{3,4,8,16\}$ which corresponds to the set of bit budgets $\{\log_2(3),2,3,4\}$. The results of these experiments are shown in Table 2.6.1. In particular, the table shows that the performance of GPFQ degrades gracefully as the bit budget decreases, while the performance of MSQ drops dramatically. In this experiment, the best bit budget for both MSQ and GPFQ networks was 4 bits, or 16 characters in the alphabet. We plot the test accuracies for the best MSQ and the best GPFQ quantized network as each layer is quantized in Figure 2.6.2a. Both networks suffer from a drop in test accuracy after quantizing the second layer, but (like in the first experiment) GPFQ recovers from this dip in subsequent layers while MSQ does not. Finally, to illustrate the difference between the two sets of quantized weights in this layer we histogram the weights in Figure 2.6.2b.

Table 2.6.1: This table documents the test accuracies for the analog and quantized neural networks on CIFAR10 data for the various choices of alphabet scalars C_α and bit budgets.

CIFAR10 Top-1 Test Accuracy

Bits	C_α	Analog	GPFQ	MSQ
$\log_2(3)$	2	0.8922	**0.7487**	0.1347
	3	0.8922	0.7350	0.1464
	4	0.8922	0.6919	0.0991
	5	0.8922	0.5627	0.1000
	6	0.8922	0.3515	0.1000
2	2	0.8922	0.7522	0.2209
	3	0.8922	**0.8036**	0.2800
	4	0.8922	0.7489	0.1742
	5	0.8922	0.6748	0.1835
	6	0.8922	0.5365	0.1390
3	2	0.8922	0.7942	0.4173
	3	0.8922	0.8670	0.3754
	4	0.8922	**0.8710**	0.5014
	5	0.8922	0.8567	0.5652
	6	0.8922	0.8600	0.5360
4	2	0.8922	0.8124	0.4525
	3	0.8922	0.8778	0.7776
	4	0.8922	0.8879	0.8443
	5	0.8922	**0.8888**	0.8291
	6	0.8922	0.8810	0.7831



(a) (b)

Figure 2.6.2: Figure 2.6.2a plots top-1 test accuracy as layers are quantized successively for the best MSQ and GPFQ networks according Table 2.6.1. Only fully-connected and convolutional layer indices are plotted. Figure 2.6.2b histograms the weights for the MSQ and GPFQ networks at the second layer.

2.7 Future Work

Despite all of the analysis that has gone into proving stability of quantizing the first layer of a neural network using the dynamical system (2.4.1) and isotropic Gaussian data, there are still many interesting and unanswered questions about the performance of this quantization algorithm. The above experiments suggest that our theory can be generalized to account for non-Gaussian feature data which may have hidden dependencies between them. Beyond the subspace model we consider in Lemma 2.9.2, it would be interesting to extend the results to apply in the case of a manifold structure, or clustered feature data, whose intrinsic complexities can be used to improve the upper bounds in Theorem 2.5.1 and Theorem 2.5.2. Furthermore, it would be desirable to extend the analysis to address quantizing all of the hidden layers. As we showed in the experiments, our set-up naturally extends to the case of quantizing convolutional layers. Another extension of this work might consider modifying our quantization algorithm to account for other network models like recurrent networks. Finally, we observed in Theorem 2.5.1 that the relative training error for learning the quantization decays like $\log(N_0)\sqrt{m/N_0}$. We

also observed in the discussion at the end of Section 2.4 that when all of the feature data X_t were the same our quantization algorithm reduced to a first order greedy $\Sigma\Delta$ quantizer. Higher order $\Sigma\Delta$ quantizers in the context of oversampled finite frame coefficients and bandlimited functions are known to have quantization error which decays *polynomially* in terms of the oversampling rate. One wonders if there exist extensions of our algorithm, perhaps with a modest increase in computational complexity, that achieve faster rates of decay for the relative quantization error. We leave all of these questions for future work.

2.8 Proofs: Supporting Lemmata

This section presents supporting lemmata that characterize the geometry of the dynamical system (2.4.1), as well as standard results from high dimensional probability which we will use in the proof of the main technical result, Theorem 2.9.1, which appears in Section 2.9. Outside of the high dimensional probability results, the results of Lemmas 2.8.3, 2.8.4 and 2.8.5 consider the behavior of the dynamical system under arbitrarily distributed data.

Lemma 2.8.1. [38] Let $g \sim N(0,\sigma^2)$. Then for any $\alpha > 0$

$$\mathbb{P}(g \geq \alpha) \leq \frac{\sigma}{\alpha\sqrt{2\pi}} e^{-\frac{\alpha^2}{2\sigma^2}}.$$

Lemma 2.8.2. [38] Let $g \sim N(0, I_{m\times m})$ be an m-dimensional standard Gaussian vector. Then there exists some universal constant $c_{norm} > 0$ so that for any $\alpha > 0$

$$\mathbb{P}\left(\left|\|g\|_2 - \sqrt{m}\right| \geq \alpha\right) \leq 2e^{-c_{norm}\alpha^2}.$$

Lemma 2.8.3. Suppose that $|w_t| < 1/2$. Then

$$\{X_t \in \mathbb{R}^m : q_t = 1\} = B\left(\frac{1}{1-2w_t}u_{t-1}, \frac{1}{1-2w_t}\|u_{t-1}\|_2\right),$$

$$
\begin{aligned}
&:= B(\tilde{u}_{t-1}, \|\tilde{u}_{t-1}\|_2), \\
\{X_t \in \mathbb{R}^m : q_t = -1\} &= B\left(\frac{-1}{1+2w_t} u_{t-1}, \frac{1}{1+2w_t}\|u_{t-1}\|_2\right), \\
&:= B(\hat{u}_{t-1}, \|\hat{u}_{t-1}\|_2)
\end{aligned}
$$

Proof. When $q_t = 1$, (2.4.3) implies that

$$
\frac{X_t^T}{\|X_t\|_2^2} u_{t-1} \geq \frac{1}{2} - w_t \iff (1-2w_t)\|X_t\|_2^2 - 2X_t^T u_{t-1} \leq 0. \tag{2.8.1}
$$

Since $|w_t| < 1/2$, $1-2w_t > 0$. After dividing both sides of (2.8.1) by this factor, and recalling that $\tilde{u}_{t-1} := (1-2w_t)^{-1} u_{t-1}$, we may complete the square to get the equivalent inequality

$$
\|X_t - \tilde{u}_{t-1}\|_2^2 \leq \|\tilde{u}_{t-1}\|_2^2.
$$

An analogous argument shows the claim for the level set $\{X_t : q_t = -1\}$. □

Remark 3. When $w > \frac{1}{2}$ or $w < -\frac{1}{2}$, the algebra in the proof tells us that the set of $X_t's$ for $q_t = 1$ (resp. $q_t = -1$) is actually the complement of $B(\tilde{u}_{t-1}, \|\tilde{u}_{t-1}\|_2)$, (resp. the complement of $B(\hat{u}_{t-1}, \|\hat{u}_{t-1}\|_2)$). For the special case when $w_t = \pm 1/2$ these level sets are half-spaces.

Lemma 2.8.4. Suppose $0 < w_t < 1$, and X_t a random vector in $\mathbb{R}^m \setminus \{0\}$. Then

$$
\begin{aligned}
&\mathbb{P}_{X_t}\left((w_t - q_t)^2 + 2(w_t - q_t)\frac{X_t^T u_{t-1}}{\|X_t\|_2^2} > \alpha \middle| \mathscr{F}_{t-1}\right) \\
&= \begin{cases}
\mu_y\left(\frac{\alpha-(w_t+1)^2}{2(w_t+1)}, \frac{\alpha-(w_t-1)^2}{2(w_t-1)}\right) & \alpha < -w_t - w_t^2 \\
\mu_y\left(\frac{\alpha-w_t^2}{2w_t}, \frac{\alpha-(w_t-1)^2}{2(w_t-1)}\right) & -w_t - w_t^2 \leq \alpha \leq w_t - w_t^2 \\
0 & \alpha > w_t - w_t^2
\end{cases}
\end{aligned}
$$

where μ_y is the probability measure over $\mathbb{R}$ induced by the random variable $y := \frac{X_t^T u_{t-1}}{\|X_t\|_2^2}$.

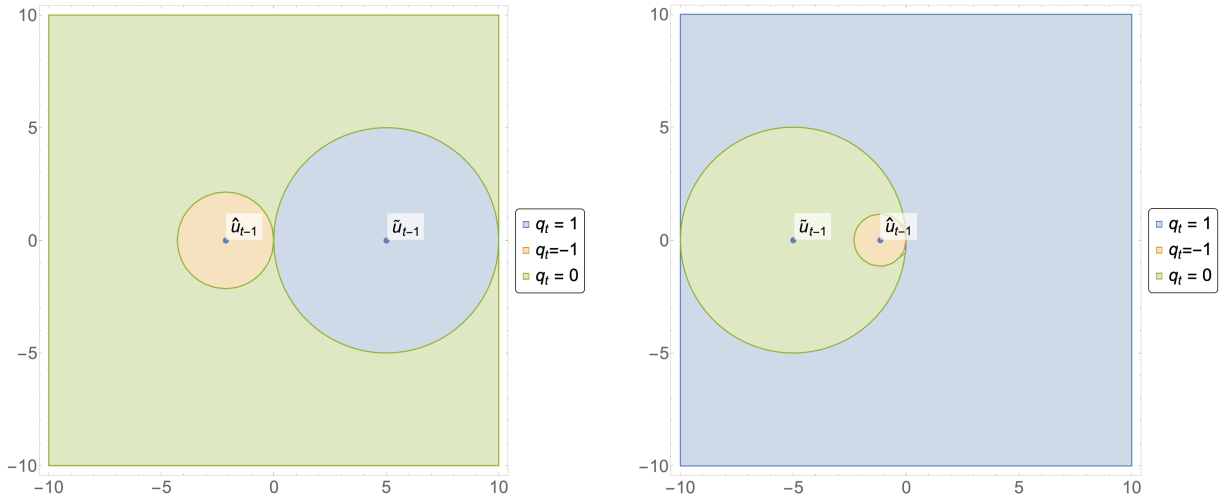


Figure 2.8.1: Visualizations of the level sets when $u_{t-1} = 3e_1$, $q_t = 1$ (blue), $q_t = -1$ (orange), and $q_t = 0$ (green) when $w_t = 0.2$ (left) and $w_t = 0.8$ (right). These are the regions that must be integrated over when calculating the moment generating function of the increment $\Delta\|u_t\|_2^2$.

Proof. Let A_b denote the event that $q_t = b$ for $b \in \{-1, 0, 1\}$. Then by the law of total probability

$$\begin{aligned}
&\mathbb{P}\left((w_t - q_t)^2 + 2(w_t - q_t)y > \alpha \middle| \mathscr{F}_{t-1}\right) \\
&= \sum_{b \in \{-1,0,1\}} \mathbb{P}\left((w_t - b)^2 + 2(w_t - b)y > \alpha \text{ and } A_b \middle| \mathscr{F}_{t-1}\right).
\end{aligned}$$

Therefore, we need to look at each summand in the above sum. Well, $q_t = 0$ precisely when $-1/2 - w_t \leq y \leq 1/2 - w_t$. So we have

$$\begin{aligned}
&\mathbb{P}\left(w_t^2 + 2w_t y > \alpha \text{ and } A_0 \middle| \mathscr{F}_{t-1}\right) = \mathbb{P}\left(y > \frac{\alpha - w_t^2}{2w_t} \text{ and } -1/2 - w_t \leq y \leq 1/2 - w_t \middle| \mathscr{F}_{t-1}\right) \\
&= \begin{cases}
\mu_y(-1/2 - w_t, 1/2 - w_t) & \alpha < -w_t - w_t^2 \\
\mu_y\left(\frac{\alpha - w_t^2}{2w_t}, 1/2 - w_t\right) & -w_t - w_t^2 \leq \alpha \leq w_t - w_t^2 \\
0 & \alpha > w_t - w_t^2
\end{cases}.
\end{aligned}$$

Next, $q_t = 1$ precisely when $y > 1/2 - w_t$. Noting that $w_t - 1 < 0$, we have

$$\mathbb{P}\left((w_t - 1)^2 + 2(w_t - 1)y > \alpha \text{ and } A_1 \middle| \mathscr{F}_{t-1}\right)$$

$$= \mathbb{P}\left(y < \frac{\alpha - (w_t - 1)^2}{2(w_t - 1)} \text{ and } y > 1/2 - w_t \middle| \mathscr{F}_{t-1}\right)$$
$$= \begin{cases} \mu_y\left(1/2 - w_t, \frac{\alpha-(w_t-1)^2}{2(w_t-1)}\right) & \alpha \leq w_t - w_t^2 \\ 0 & \alpha > w_t - w_t^2 \end{cases}.$$

Finally, $q_t = -1$ precisely when $y < -1/2 - w_t$. So we have

$$\mathbb{P}\left((w_t+1)^2 + 2(w_t+1)y > \alpha \text{ and } A_{-1} \middle| \mathscr{F}_{t-1}\right)$$
$$= \mathbb{P}\left(y > \frac{\alpha - (w_t+1)^2}{2(w_t+1)} \text{ and } y < -1/2 - w_t \middle| \mathscr{F}_{t-1}\right)$$
$$= \begin{cases} \mu_y\left(\frac{\alpha-(w_t+1)^2}{2(w_t+1)}, -1/2 - w_t\right) & \alpha \leq -w_t - w_t^2 \\ 0 & \alpha > -w_t - w_t^2 \end{cases}.$$

Summing these three piecewise functions yields the result. □

Lemma 2.8.5. When $-1 < w_t < 0$, we have

$$\mathbb{P}_{X_t}\left((w_t - q_t)^2 + 2(w_t - q_t)\frac{X_t^T u_{t-1}}{\|X_t\|_2^2} > \alpha \middle| \mathscr{F}_{t-1}\right)$$
$$= \begin{cases} \mu_y\left(\frac{\alpha-(w_t+1)^2}{2(w_t+1)}, \frac{\alpha-(w_t-1)^2}{2(w_t-1)}\right) & \alpha < w_t - w_t^2 \\ \mu_y\left(\frac{\alpha-(w_t+1)^2}{2(w_t+1)}, \frac{\alpha-w_t^2}{2w_t}\right) & w_t - w_t^2 \leq \alpha \leq -w_t - w_t^2 \\ 0 & \alpha > -w_t - w_t^2 \end{cases}.$$

Corollary 2.8.6. If $\|X_t\|_2^2 \leq B$ with probability 1, then $\Delta\|u_t\|_2^2 \leq B/4$ with probability 1.

Proof. Using (2.4.1), this follows from the identity

$$\Delta\|u_t\|_2^2 = \|X_t\|_2^2\left((w_t - q_t)^2 + 2(w_t - q_t)\frac{X_t^T u_{t-1}}{\|X_t\|_2^2}\right) \leq B\left((w_t - q_t)^2 + 2(w_t - q_t)\frac{X_t^T u_{t-1}}{\|X_t\|_2^2}\right).$$

Applying Lemma 2.8.4 (or Lemma 2.8.5) on the latter quantity with $\alpha = |w_t| - w_t^2$ and recognizing that $|w_t| - w_t^2 \leq 1/4$ when $w_t \in [-1, 1]$ yields the claim. □

2.9 Proofs: Core Lemmata

We start by proving our main result, Theorem 2.9.1, and its extension to the case where feature vectors live in a low-dimensional subspace, Lemma 2.9.2. The proof of Theorem 2.9.1 relies on bounding the moment generating function of $\Delta\|u_t\|_2^2 \Big| \mathscr{F}_{t-1}$, which in turn requires a number of results, referenced in the proof and presented thereafter. These lemmas carefully deal with bounding the above moment generating function on the events where q_t is fixed. Given u_{t-1} and $q_t = b$, Lemma 2.8.3 tells us the set of directions X_t which result in $q_t = b$ and these are the relevant events one needs to consider when bounding the moment generating function. Lemma 2.9.3 handles the case when $q_t = 0$, Lemma 2.9.4 handles the case when $q_t = 1$, and Lemma 2.9.5 handles the case when $q_t = -1$.

Theorem 2.9.1. Suppose that for $t \in \mathbb{N}$, the vectors $X_t \sim \mathscr{N}(0, \sigma^2 I_{m\times m})$ are independent and that $w_t \in [-1,1]$ are i.i.d. and independent of X_t, and define the event

$$A_\varepsilon := \{\text{dist}(w_t, \{-1,0,1\}) < \varepsilon\}.$$

Then there exist positive constants c_{norm}, C_λ, and C_{sup}, such that with $\lambda := \frac{C_\lambda}{C_{\text{sup}}^2 \sigma^2 m \log(N_0)}$, and $\rho, \varepsilon \in (0,1)$ satisfying $\tilde{\rho} := \rho + e^{C_\lambda/4}\mathbb{P}(A_\varepsilon) < 1$, the iteration (2.4.2) satisfies

$$\mathbb{P}\left(\|u_t\|_2^2 > \alpha\right) \leq \tilde{\rho}^t e^{-\lambda\alpha} + \frac{1-\tilde{\rho}^t}{1-\tilde{\rho}} e^{\frac{C_\lambda}{4} + \lambda(\beta-\alpha)} + 2e^{-\log(N_0)(c_{norm} C_{\text{sup}}^2 m - 1)}. \tag{2.9.1}$$

Above, $C > 0$ is a universal constant and $\beta := C\frac{e^{8C_\lambda}\sigma^2 m^2 \log^2(N_0)}{\rho^2 \varepsilon^2}$.

Proof. The proof technique is inspired by [17]. Define the events

$$U_t := \left\{ \sup_{j\in\{1,\ldots,t\}} \|X_j\|_2 \leq C_{\text{sup}}\sigma\sqrt{m}\left(\sqrt{\log(N_0)} + 1\right)\right\}.$$

Using a union bound and Lemma 2.8.2, we see that $U_{N_0}^C$ happens with low probability since

$$\mathbb{P}\left(\sup_{t\in[N_0]}\|X_t\|_2 > C_{\sup}\sigma\sqrt{m}\left(\sqrt{\log(N_0)}+1\right)\right) \leq 2N_0 e^{-c_{norm}C_{\sup}^2 m\log(N_0)}$$
$$= 2e^{-\log(N_0)(c_{norm}C_{\sup}^2 m-1)}.$$

We can therefore bound the probability of interest with appropriate conditioning.

$$\mathbb{P}\left(\|u_t\|_2^2 \geq \alpha\right) \leq \mathbb{P}\left(\|u_t\|_2^2 \geq \alpha \middle| U_{N_0}\right)\mathbb{P}(U_{N_0}) + \mathbb{P}(U_{N_0}^C).$$

Looking at the first summand, for any $\lambda > 0$, we have by Markov's inequality

$$\begin{aligned}\mathbb{P}\left(\|u_t\|_2^2 \geq \alpha \middle| U_{N_0}\right)\mathbb{P}(U_{N_0}) &\leq e^{-\lambda\alpha}\mathbb{E}[e^{\lambda\|u_t\|_2^2}\middle| U_{N_0}]\mathbb{P}(U_{N_0}) \\ &= e^{-\lambda\alpha}\mathbb{E}[e^{\lambda\|u_t\|_2^2}\mathbb{1}_{U_{N_0}}] \\ &= e^{-\lambda\alpha}\mathbb{E}\left[e^{\lambda\|u_{t-1}\|_2^2}e^{\lambda\Delta\|u_t\|_2^2}\mathbb{1}_{U_{N_0}}\right] \\ &= e^{-\lambda\alpha}\mathbb{E}\left[\mathbb{E}\left[e^{\lambda\|u_{t-1}\|_2^2}e^{\lambda\Delta\|u_t\|_2^2}\mathbb{1}_{U_{N_0}}\middle|\mathscr{F}_{t-1}\right]\right].\end{aligned}$$

We expand the conditional expectation given the filtration into a sum of two parts

$$\begin{aligned}\mathbb{E}\left[e^{\lambda\|u_{t-1}\|_2^2}e^{\lambda\Delta\|u_t\|_2^2}\mathbb{1}_{U_{N_0}}\middle|\mathscr{F}_{t-1}\right] &= \mathbb{E}\left[e^{\lambda\|u_{t-1}\|_2^2}e^{\lambda\Delta\|u_t\|_2^2}\mathbb{1}_{U_{N_0}}\mathbb{1}_{A_\varepsilon^C \text{ and } \|u_{t-1}\|_2^2\geq\beta}\middle|\mathscr{F}_{t-1}\right] \\ &+ \mathbb{E}\left[e^{\lambda\|u_{t-1}\|_2^2}e^{\lambda\Delta\|u_t\|_2^2}\mathbb{1}_{U_{N_0}}\mathbb{1}_{A_\varepsilon \text{ or } \|u_{t-1}\|_2^2<\beta}\middle|\mathscr{F}_{t-1}\right].\end{aligned}$$

Towering expectations, the expectation over X_t of the first summand is bounded above by $\rho e^{\lambda\|u_{t-1}\|_2^2}\mathbb{1}_{U_{t-1}}$ for all w_t on the event A_ε^C using Lemmas 2.9.3, 2.9.4, and 2.9.5. Therefore, the same bound is also true for the expectation over w_t. As for the second term, we have

$$\mathbb{E}\left[e^{\lambda\|u_{t-1}\|_2^2}e^{\lambda\Delta\|u_t\|_2^2}\mathbb{1}_{U_{N_0}}\mathbb{1}_{A_\varepsilon \text{ or } \|u_{t-1}\|_2^2<\beta}\middle|\mathscr{F}_{t-1}\right] =$$

$$\mathbb{E}\left[e^{\lambda\|u_{t-1}\|_2^2}e^{\lambda\Delta\|u_t\|_2^2}\mathbb{1}_{U_{N_0}}\mathbb{1}_{\|u_{t-1}\|_2^2<\beta}\middle|\mathscr{F}_{t-1}\right]+\mathbb{E}\left[e^{\lambda\|u_{t-1}\|_2^2}e^{\lambda\Delta\|u_t\|_2^2}\mathbb{1}_{U_{N_0}}\mathbb{1}_{A_\varepsilon \text{ and } \|u_{t-1}\|_2^2\geq\beta}\middle|\mathscr{F}_{t-1}\right]$$

For both terms, we can use the uniform bound on the increments as proven in Corollary 2.8.6. The first term we can bound by $e^{\lambda C_{\sup}^2\sigma^2 m\log(N_0)/4}e^{\lambda\beta}\leq e^{C_\lambda/4}e^{\lambda\beta}$. As for the second, expecting over the draw of w_t gives us

$$\begin{aligned}\mathbb{E}\left[e^{\lambda\|u_{t-1}\|_2^2}e^{\lambda\Delta\|u_t\|_2^2}\mathbb{1}_{U_{N_0}}\mathbb{1}_{A_\varepsilon \text{ and } \|u_{t-1}\|_2^2\geq\beta}\middle|\mathscr{F}_{t-1}\right]&\leq e^{\lambda C_{\sup}^2\sigma^2 m\log(N_0)/4}e^{\lambda\|u_{t-1}\|_2^2}\mathbb{1}_{U_{t-1}}\mathbb{P}(A_\varepsilon)\\&\leq e^{C_\lambda/4}e^{\lambda\|u_{t-1}\|_2^2}\mathbb{1}_{U_{t-1}}\mathbb{P}(A_\varepsilon).\end{aligned}$$

Therefore, we have

$$\begin{aligned}&\mathbb{P}\left(\|u_t\|_2^2>\alpha\right)\\&\leq e^{-\lambda\alpha}\left(\left(\rho+e^{C_\lambda/4}\mathbb{P}\left(A_\varepsilon\right)\right)\mathbb{E}[e^{\lambda\|u_{t-1}\|_2^2}\mathbb{1}_{U_{t-1}}]+e^{\lambda\beta+C_\lambda/4}\right)+2e^{-\log(N_0)(c_{norm}C_{\sup}^2m-1)}\\&=e^{-\lambda\alpha}\left(\tilde{\rho}\mathbb{E}[e^{\lambda\|u_{t-1}\|_2^2}\mathbb{1}_{U_{t-1}}]+e^{\lambda\beta+C_\lambda/4}\right)+2e^{-\log(N_0)(c_{norm}C_{\sup}^2m-1)}.\end{aligned}$$

Proceeding inductively on $\mathbb{E}[e^{\lambda\|u_{t-1}\|_2^2}]$ yields the claim. □

Remark 4. To simplify the bound in (2.9.1), assuming we have $\tilde{\rho},\varepsilon\propto 1$ and $\alpha\gtrsim\beta\propto\sigma^2m^2\log^2(N_0)$ we have

$$\begin{aligned}\mathbb{P}\left(\|u_t\|_2^2\geq\alpha\right)&\leq e^{-\lambda\alpha}+e^{\frac{C_\lambda}{4}+\lambda(\beta-\alpha)}+2e^{-\log(N_0)(c_{norm}C_{\sup}^2m-1)},\\&=e^{-\frac{C_\lambda m\log(N_0)}{C_{\sup}^2}}+e^{\frac{C_\lambda}{4}-C'\frac{C_\lambda m\log(N_0)}{C_{\sup}^2}}+2e^{-\log(N_0)(c_{norm}C_{\sup}^2m-1)},\\&\leq Ce^{-cm\log(N_0)}.\end{aligned}$$

This matches the bound on the probability of failure we give in Theorem 2.5.1.

Lemma 2.9.2. Suppose $X=ZA$ where $Z\in\mathbb{R}^{m\times d}$ satisfies $Z^TZ=I$, and $A\in\mathbb{R}^{d\times N_0}$ has i.i.d. $\mathscr{N}(0,\sigma^2)$ entries. In other words, suppose the feature data X_t are Gaussians drawn from a

d-dimensional subspace of $\mathbb{R}^m$. Then with the remaining hypotheses as Theorem 2.9.1 we have with probability at least $1 - Ce^{-cd\log(N_0)} - 3\exp(-c''d)$

$$\|Xw - Xq\|_2 \lesssim \sigma d \log(N_0).$$

Proof. We will show that running the dynamical system (2.4.1) with X_t is equivalent to running a modified version of (2.4.1) with the columns of A, denoted A_t. Then we can apply the result of Theorem 2.9.1. By definition, we have

$$\begin{aligned} u_0 &:= 0 \in \mathbb{R}^m, \\ q_t &:= \mathscr{Q}\left(w_t + \frac{X_t^T u_{t-1}}{\|X_t\|_2^2}\right), \\ u_t &:= u_{t-1} + w_t X_t - q_t X_t. \end{aligned}$$

In anticipation of subsequent applications of change of variables, let $Z = U\Sigma V^T \in \mathbb{R}^{m\times d}$ be the singular value decomposition of Z where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{d\times d}$ are orthogonal matrices and $\Sigma \in \mathbb{R}^{m\times d}$ decomposes as

$$\Sigma = \begin{bmatrix} I_{d\times d} \\ 0 \end{bmatrix}.$$

Since u_t is a linear combination of $X_1, \ldots, X_t$ for all t it follows that u_t is in the column space of Z. In other words, $u_t = Z(Z^TZ)^{-1}Z^Tu_t := Z\eta_t$. We may rewrite the above dynamical system in terms of A_t, η_t as

$$\begin{aligned} u_0 &:= 0 \in \mathbb{R}^m, \\ q_t &:= \mathscr{Q}\left(w_t + \frac{A_t^T Z^T Z\eta_{t-1}}{\|ZA_t\|_2^2}\right) \end{aligned}$$

$$= \mathscr{Q}\left(w_t + \frac{A_t^T \eta_{t-1}}{\|A_t\|_2^2}\right)$$

$$Z\eta_t := Z\eta_{t-1} + w_t ZA_t - q_t ZA_t$$

$$\iff \eta_t = \eta_{t-1} + w_t A_t - q_t A_t$$

So, in other words, we've reduced to running (2.4.1) but now with the state variables $\eta_{t-1} \in \mathbb{R}^d$ in place of u_{t-1} and with A_t in place of X_t. Applying the result of Theorem 2.9.1 yields the claim. □

Lemma 2.9.3. Let $X_t \sim \mathcal{N}(0, \sigma^2 I_{m\times m})$ and $\text{dist}(w_t, \{-1,0,1\}) \geq \varepsilon$. Define the event

$$U := \left\{ \|X_t\|_2 \leq C_{\text{sup}} \sigma \sqrt{m \log(N_0)} \right\},$$

and set $\lambda := \frac{C_\lambda}{C_{\text{sup}}^2 \sigma^2 m \log(N_0)}$, where $C_\lambda \in (0, \frac{c_{norm}}{12})$ is some constant and c_{norm} is as in Lemma 2.8.2. Then there exists a universal constant $C > 0$ so that with $\beta := \frac{Ce^{C_\lambda} \sigma^2 m \log(N_0)}{\rho \varepsilon \sigma}$

$$\mathbb{E}\left[e^{\lambda \Delta \|u_t\|_2^2} \mathbb{1}_{q_t=0} \mathbb{1}_U \mathbb{1}_{\|u_{t-1}\|_2 \geq \beta} \middle| \mathscr{F}_{t-1} \right] \leq \rho.$$

Proof. Recall $\Delta\|u_t\|_2^2 = \|u_t\|_2^2 - \|u_{t-1}\|_2^2 = (w_t - q_t)^2 \|X_t\|_2^2 + 2(w_t - q_t)\langle X_t, u_{t-1}\rangle$. Let us first consider the case when $|w_t| < \frac{1}{2}$. We will further assume that $w_t > 0$, since there is the symmetry between $\hat{u}_{t-1}$ and $\tilde{u}_{t-1}$ under the mapping $w_t \to -w_t$. Before embarking on our calculus journey, let us make some key remarks. First, on the event U, we can bound the increment above by $\Delta\|u_t\|_2^2 \leq (w_t - q_t)^2 C_{\text{sup}}^2 \sigma^2 m \log(N_0) + 2(w_t - q_t)\langle X_t, u_{t-1}\rangle$. So, it behooves us to find an upper bound for $\mathbb{E}\left[e^{2\lambda w_t \langle X_t, u_{t-1}\rangle} \mathbb{1}_U \mathbb{1}_{q_t=0} \mathbb{1}_{\|u_{t-1}\|_2 \geq \beta} | \mathscr{F}_{t-1} \right]$. Since the exponential function is non-negative, we can always upper bound this expectation by removing the indicator on U. In other words,

$$\mathbb{E}\left[e^{2\lambda w_t \langle X_t, u_{t-1}\rangle} \mathbb{1}_U \mathbb{1}_{q_t=0} \mathbb{1}_{\|u_{t-1}\|_2 \geq \beta} \middle| \mathscr{F}_{t-1} \right] \leq \mathbb{E}\left[e^{2\lambda w_t \langle X_t, u_{t-1}\rangle} \mathbb{1}_{q_t=0} \mathbb{1}_{\|u_{t-1}\|_2 \geq \beta} \middle| \mathscr{F}_{t-1} \right]. \quad (2.9.2)$$

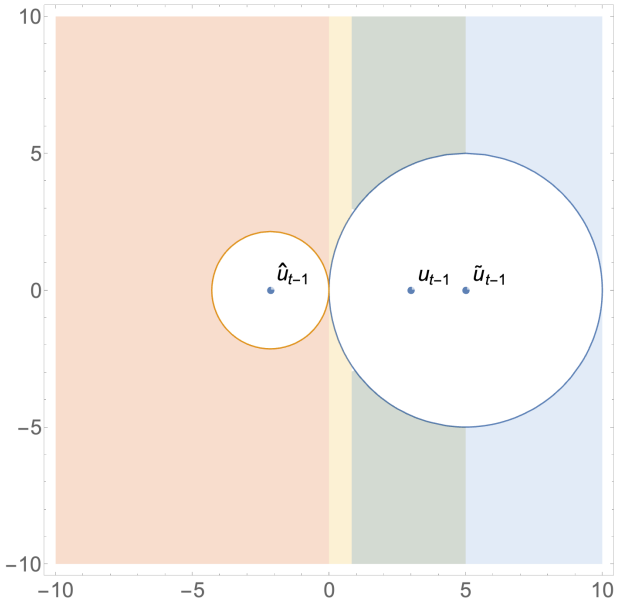


Figure 2.9.1: The regions of integration involved in the calculations in Lemma 2.9.3 for the particular case when $w_t = 0.3$ and $u_t = 3e_1$. The region in red corresponds to equation (2.9.3), yellow to region R as in equation (2.9.9), green to region S as in equation (2.9.15), and blue to region T as in equation (2.9.12).

Since we're indicating on an event where $\|u_{t-1}\|_2 \geq \beta$, we will need to handle the events where $\langle X_t, u_{t-1}\rangle > 0$ with some care, since without an *a priori* upper bound on $\|u_{t-1}\|$ the moment generating function restricted to this event could explode. Therefore, we'll divide the region of integration into 4 pieces which are depicted in Figure 2.9.1. Because of the abundance of notation in the following arguments, we will denote $\mathbb{1}_\beta := \mathbb{1}_{\|u_{t-1}\|_2 \geq \beta}$.

Let's handle the easier event first, namely where $\langle X_t, u_{t-1}\rangle \leq 0$. Here, we have

$$\mathbb{E}\left[e^{2\lambda\langle X_t, u_{t-1}\rangle}\mathbb{1}_\beta \mathbb{1}_{q_t=0}\mathbb{1}_{\langle X_t, u_{t-1}\rangle<0}\middle|\mathscr{F}_{t-1}\right] =$$
$$(2\pi\sigma^2)^{-m/2}\mathbb{1}_\beta \int_{B(\hat{u}_{t-1}, \|\hat{u}_{t-1}\|)^C \cap \{\langle x, u_{t-1}\rangle \leq 0\}} e^{2\lambda w_t \langle x, u_{t-1}\rangle} e^{\frac{-1}{2\sigma^2}\|x\|_2^2}\, dx. \tag{2.9.3}$$

By rotational invariance, we may assume without loss of generality that $u_{t-1} = \|u_{t-1}\|_2 e_1$, where $e_1 \in \mathbb{R}^m$ is the first standard basis vector. In that case, the constraint $\langle X_t, u_{t-1}\rangle < 0$ is equivalent to $X_{t,1} < 0$, where $X_{t,1}$ is the first component of X_t. Using Lemma 2.8.3, it follows that the set of X_t

for which $q_t = 0$ and $X_{t,1} < 0$ is simply $\{x \in \mathbb{R}^m : x_1 \leq 0\} \cap B(-\|\hat{u}_{t-1}\|_2 e_1, \|\hat{u}_{t-1}\|_2)^C$, where the negative sign here comes from the fact that $\hat{u}_{t-1} = -(1+2w_t)u_{t-1}$. That means we can rewrite (2.9.3) as

$$(2\pi\sigma^2)^{-m/2}\mathbb{1}_\beta \int_{B(-\|\hat{u}_{t-1}\|_2 e_1, \|\hat{u}_{t-1}\|)^C \cap \{x_1 \leq 0\}} e^{2\lambda w_t \|u_{t-1}\| x_1 - \frac{x_1^2}{2\sigma^2}} e^{\frac{-1}{2\sigma^2}\sum_{j\geq 2} x_j^2}\, dx.$$

Perhaps surprisingly, we can afford to use the crude upper bound on this integral by simply removing the constraint that $x \in B(-\|\hat{u}_{t-1}\|_2 e_1, \|\hat{u}_{t-1}\|)^C$. Iterating the univariate integrals then gives us

$$\begin{aligned}
&(2\pi\sigma^2)^{-m/2}\mathbb{1}_\beta \int_{B(-\|\hat{u}_{t-1}\|_2 e_1, \|\hat{u}_{t-1}\|)^C \cap \{x_1 \leq 0\}} e^{2\lambda w_t \|u_{t-1}\| x_1 - \frac{x_1^2}{2\sigma^2}} e^{\frac{-1}{2\sigma^2}\sum_{j\geq 2} x_j^2}\, dx \\
&\leq (2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta \int_{-\infty}^{0} e^{2\lambda w_t \|u_{t-1}\| x_1 - \frac{x_1^2}{2\sigma^2}}\, dx_1 \int_{\mathbb{R}^{m-1}} (2\pi\sigma^2)^{-\frac{m-1}{2}} e^{\frac{-1}{2\sigma^2}\sum_{j\geq 2} x_j^2}\, dx_2 \ldots dx_m \\
&= (2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta \int_{-\infty}^{0} e^{2\lambda w_t \|u_{t-1}\| x_1 - \frac{x_1^2}{2\sigma^2}}\, dx_1 = (2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta \int_0^{\infty} e^{-2\lambda w_t \|u_{t-1}\| x_1 - \frac{x_1^2}{2\sigma^2}}\, dx_1.
\end{aligned} \tag{2.9.4}$$

We complete the square and use a change of variables to reformulate (2.9.4) as

$$\begin{aligned}
&(2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta e^{2\sigma^2\lambda^2 w_t^2 \|u_{t-1}\|_2^2} \int_0^{\infty} e^{-\frac{1}{2\sigma^2}\left(x_1 + 2\sigma^2\lambda w_t \|u_{t-1}\|_2\right)^2}\, dx_1 \\
&= (2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta e^{2\sigma^2\lambda^2 w_t^2 \|u_{t-1}\|_2^2} \int_{2\sigma^2\lambda w_t \|u_{t-1}\|_2}^{\infty} e^{-\frac{x_1^2}{2\sigma^2}}\, dx_1.
\end{aligned} \tag{2.9.5}$$

Since the lower limit of integration is positive and large when $\|u_{t-1}\|_2$ is, we can use a tail bound as in Lemma 2.8.1 to upper bound (2.9.5) by

$$\frac{\mathbb{1}_\beta \sigma}{2\sigma^2\lambda w_t \|u_{t-1}\|_2 \sqrt{2\pi}} \leq \frac{\mathbb{1}_\beta \sigma}{\sigma^2\lambda\varepsilon\|u_{t-1}\|_2} = \frac{\mathbb{1}_\beta C_{\sup}^2 \sigma m \log(N_0)}{C_\lambda \varepsilon \|u_{t-1}\|_2}, \tag{2.9.6}$$

where the first inequality follows from $|w_t| \geq \varepsilon$ and the equality follows from $\lambda = \frac{C_\lambda}{C_{\sup}^2 \sigma^2 m \log(N_0)}$.

Now we handle the moment generating function on the event that $\langle X_t, u_{t-1}\rangle \geq 0$. Again, using rotational invariance to assume $u_{t-1} = \|u_{t-1}\|_2 e_1$, we have by Lemma 2.8.3 that the event to integrate over is $\{x \in \mathbb{R}^m : x_1 \geq 0\} \cap B(\|\tilde{u}_{t-1}\|_2 e_1, \|\tilde{u}_{t-1}\|_2)^C$. Notice that iterating the integrals gives us

$$\begin{aligned}
&\mathbb{E}\left[e^{2\lambda\langle X_t, u_{t-1}\rangle}\mathbb{1}_\beta \mathbb{1}_{q_t=0}\mathbb{1}_{\langle X_t, u_{t-1}\rangle \geq 0}\middle|\mathscr{F}_t-1\right] \\
&\quad = (2\pi\sigma^2)^{-m/2}\mathbb{1}_\beta \int_{B(\tilde{u}_{t-1}, \|\tilde{u}_{t-1}\|)^C \cap \{x_1 \geq 0\}} e^{2\lambda w_t \|u_{t-1}\| x_1 - \frac{1}{2\sigma^2}\|x\|_2^2}\, dx \\
&\quad = (2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta \int_0^\infty e^{2\lambda w_t \|u_{t-1}\| x_1 - \frac{x_1^2}{2\sigma^2}} \int_{B\left(0, \sqrt{(2x_1\|\tilde{u}_{t-1}\|_2 - x_1^2)^+}\right)^C} (2\pi\sigma^2)^{-\frac{m-1}{2}} e^{-\frac{1}{2\sigma^2}\sum_{j=2}^m x_j^2}\, dx_2 \ldots dx_m dx_1, \quad (2.9.7)
\end{aligned}$$

with the notation $(z)^+ = \max\{z, 0\}$ for $z \in \mathbb{R}$. Consequentially, we can rephrase (2.9.7) into a more probabilistic statement. Below, let $\gamma_j \sim \mathscr{N}(0,1)$ denote i.i.d. standard normal random variables. Then (2.9.7) is equal to

$$(2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta \int_0^\infty e^{2\lambda w_t \|u_{t-1}\| x_1 - \frac{x_1^2}{2\sigma^2}} \mathbb{P}\left(\sigma^2 \sum_{j=1}^{m-1} \gamma_j^2 \geq 2x_1\|\tilde{u}_{t-1}\|_2 - x_1^2\right) dx_1. \quad (2.9.8)$$

The probability appearing in (2.9.8) will decay exponentially provided $2x_1\|\tilde{u}_{t-1}\|_2 - x_1^2$ is sufficiently large. To that end, we will divide up this half-space into the following regions. Let $C_0 \geq 16$ be a constant and define the sets $R := \{x \in \mathbb{R}^m : 0 \leq x_1 \leq \frac{C_0\sigma^2 m}{\|\tilde{u}_{t-1}\|_2}\}$, $S := \{x \in \mathbb{R}^m : \frac{C_0\sigma^2 m}{\|\tilde{u}_{t-1}\|_2} \leq x_1 \leq \|\tilde{u}_{t-1}\|_2\}$, and $T := \{x \in \mathbb{R}^m : \|\tilde{u}_{t-1}\|_2 \leq x_1\}$. Figure 2.9.1 gives a visual depiction of this decomposition. Then we have

$$\begin{aligned}
\mathbb{E}\left[e^{2\lambda\langle X_t, u_{t-1}\rangle}\mathbb{1}_\beta \mathbb{1}_{q_t=0}\mathbb{1}_{\langle X_t, u_{t-1}\rangle \geq 0}\middle|\mathscr{F}_t-1\right] &= (2\pi\sigma^2)^{-m/2}\mathbb{1}_\beta \int_{B(\tilde{u}_{t-1}, \|\tilde{u}_{t-1}\|)^C \cap R} e^{2\lambda w_t \|u_{t-1}\| x_1 - \frac{1}{2\sigma^2}\|x\|_2^2}\, dx \\
&\quad + (2\pi\sigma^2)^{-m/2}\mathbb{1}_\beta \int_{B(\tilde{u}_{t-1}, \|\tilde{u}_{t-1}\|)^C \cap S} e^{2\lambda w_t \|u_{t-1}\| x_1 - \frac{1}{2\sigma^2}\|x\|_2^2}\, dx
\end{aligned}$$

$$+ (2\pi\sigma^2)^{-m/2}\mathbb{1}_\beta \int_{B(\tilde{u}_{t-1},\|\tilde{u}_{t-1}\|)^C \cap T} e^{2\lambda w_t \|u_{t-1}\| x_1 - \frac{1}{2\sigma^2}\|x\|_2^2}\, dx.$$

For the integral over R, we will use the naïve upper bound $\mathbb{P}\left(\sigma^2 \sum_{j=1}^{m-1} \gamma_j^2 \geq 2x_1 \|\tilde{u}_{t-1}\|_2 - x_1^2\right) \leq 1$. This gives us

$$\begin{aligned}
&(2\pi\sigma^2)^{-m/2}\mathbb{1}_\beta \int_{B(\tilde{u}_{t-1},\|\tilde{u}_{t-1}\|)^C \cap R} e^{2\lambda w_t \|u_{t-1}\| x_1 - \frac{1}{2\sigma^2}\|x\|_2^2}\, dx \\
&\leq (2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta \int_0^{\frac{C_0\sigma^2 m}{\|\tilde{u}_{t-1}\|_2}} e^{2\lambda w_t \|u_{t-1}\| x_1 - \frac{1}{2\sigma^2}x_1^2}\, dx_1 \\
&= (2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta e^{2\lambda^2\sigma^2 w_t^2 \|u_{t-1}\|_2^2} \int_{-2\lambda w_t \sigma^2 \|u_{t-1}\|_2}^{\frac{C_0\sigma^2 m}{\|\tilde{u}_{t-1}\|_2} - 2\lambda w_t \sigma^2 \|u_{t-1}\|_2} e^{-\frac{1}{2\sigma^2}x_1^2}\, dx_1. \qquad (2.9.9)
\end{aligned}$$

The upper limit of integration is negative since $\|u_{t-1}\|_2^2 \geq \frac{3C_0 C_{\sup}^2 \sigma^2 m \log(N_0)}{2C_\lambda \varepsilon} \geq \frac{C_0|1-2w_t|}{2\lambda w_t}$. Under this assumption, we can upper bound the integral with a Riemann sum. As the maximum of the integrand occurs at the upper limit of integration, we bound (2.9.9) with

$$(2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta \frac{e^{\frac{-1}{2\sigma^2}\left(\frac{C_0^2\sigma^4 m^2}{\|\tilde{u}_{t-1}\|_2^2} - \frac{4C_0\sigma^4 m\lambda w_t \|u_{t-1}\|_2}{\|\tilde{u}_{t-1}\|_2}\right)} C_0\sigma^2 m}{\|\tilde{u}_{t-1}\|_2}. \qquad (2.9.10)$$

Recognizing that $\frac{\|u_{t-1}\|_2}{\|\tilde{u}_{t-1}\|_2} = |1-2w_t| \leq 3$ and recalling that $\lambda = \frac{C_\lambda}{C_{\sup}^2 \sigma^2 m \log(N_0)}$ we can further upper bound by

$$\frac{\mathbb{1}_\beta e^{2C_0\lambda\sigma^2 m w_t |1-2w_t|} C_0\sigma m}{\|\tilde{u}_{t-1}\|_2\sqrt{2\pi}} \leq \frac{\mathbb{1}_\beta 3C_0 e^{\frac{6C_0C_\lambda}{C_{\sup}^2\log(N_0)}}\sigma m}{\|u_{t-1}\|_2}. \qquad (2.9.11)$$

As was the case for R, we can use the bound $\mathbb{P}\left(\sigma^2 \sum_{j=1}^{m-1} \gamma_j^2 \geq 2x_1 \|\tilde{u}_{t-1}\|_2 - x_1^2\right) \leq 1$ over T too. Completing the square in the exponent as we usually do gives us

$$(2\pi\sigma^2)^{-m/2}\mathbb{1}_\beta \int_{B(\tilde{u}_{t-1},\|\tilde{u}_{t-1}\|)^C \cap T} e^{2\lambda w_t \|u_{t-1}\| x_1 - \frac{1}{2\sigma^2}\|x\|_2^2}\, dx$$

$$\leq (2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta e^{2\lambda^2 w_t^2\sigma^2\|u_{t-1}\|_2^2}\int_{\|\tilde{u}_{t-1}\|_2-2\lambda w_t\sigma^2\|u_{t-1}\|_2}^{\infty} e^{\frac{-x_1^2}{2\sigma^2}}\,dx_1. \tag{2.9.12}$$

Since $\lambda < \frac{1}{6\sigma^2} \leq \frac{1}{2\sigma^2 w_t|1-2w_t|}$ the lower limit of integration is positive, so we can use a Gaussian tail bound as in Lemma 2.8.1 to bound (2.9.12) by

$$\begin{aligned}&\frac{\mathbb{1}_\beta\sigma}{\sqrt{2\pi}\left(\|\tilde{u}_{t-1}\|_2-2\lambda w_t\sigma^2\|u_{t-1}\|_2\right)}e^{\frac{-1}{2\sigma^2}\left(\|\tilde{u}_{t-1}\|_2^2-4\lambda w_t\sigma^2\|u_{t-1}\|_2\|\tilde{u}_{t-1}\|_2\right)}\\&=\frac{\mathbb{1}_\beta\sigma}{\sqrt{2\pi}\left(\|\tilde{u}_{t-1}\|_2-2\lambda w_t\sigma^2\|u_{t-1}\|_2\right)}e^{\frac{-\|u_{t-1}\|_2^2}{2\sigma^2}\left(\frac{1}{|1-2w_t|^2}-\frac{4\lambda w_t\sigma^2}{|1-2w_t|}\right)}.\end{aligned} \tag{2.9.13}$$

As $\lambda < \frac{1}{12\sigma^2} \leq \frac{1}{4w_t\sigma^2|1-2w_t|}$ the exponent appearing in (2.9.13) is negative. Bounding the exponential by 1 then gives us the upper bound

$$\begin{aligned}\frac{\mathbb{1}_\beta\sigma}{\sqrt{2\pi}\left(\|\tilde{u}_{t-1}\|_2-2\lambda w_t\sigma^2\|u_{t-1}\|_2\right)}&=\frac{\mathbb{1}_\beta\sigma}{\|u_{t-1}\|_2\left(\frac{1}{|1-2w_t|}-\frac{2w_tC_\lambda\sigma^2}{C_{\text{sup}}^2\sigma^2 m\log(N_0)}\right)}\\&\leq\frac{\mathbb{1}_\beta\sigma}{\|u_{t-1}\|_2\left(\frac{1}{3}-\frac{2C_\lambda}{C_{\text{sup}}^2 m\log(N_0)}\right)}.\end{aligned} \tag{2.9.14}$$

Now, for S we can use the exponential decay of the probability appearing in (2.9.8). To make the algebra a bit nicer, we can upper-bound this probability by $\mathbb{P}\left(\sigma^2\sum_{j=1}^{m-1}\gamma_j^2 \geq x_1\|\tilde{u}_{t-1}\|\right)$ since on S we have $0 \leq x_1 \leq \|\tilde{u}_{t-1}\|_2$. Setting $\nu := \frac{1}{\sigma\sqrt{m-1}}\sqrt{x_1\|\tilde{u}_{t-1}\|}$, Lemma 2.8.2 tells us for $x_1 \geq \frac{C_0\sigma^2 m}{\|\tilde{u}_{t-1}\|}$

$$\mathbb{P}\left(\sqrt{\sum_{j=1}^{m-1}\gamma_j^2} \geq \sqrt{m-1}\nu\right) \leq 2\exp(-c_{norm}(\nu-1)^2(m-1)).$$

To simplify our algebra, we remark that for any $c > 0$,

$$e^{-c(m-1)(z-1)^2} \leq e^{\frac{-c}{2}(m-1)z^2},$$

provided $z \geq 4$. By our choice of C_0, this happens to be the case on S, as $\frac{C_0\sigma^2 m}{\|\tilde{u}_{t-1}\|} \leq x_1 \leq \|\hat{u}_{t-1}\|$

and so

$$\nu^2 \geq \frac{x_1\|\tilde{u}_{t-1}\|_2}{\sigma^2 m} \geq C_0.$$

This gives us the upper bound on the probability

$$\mathbb{P}\left(\sigma^2\sum_{j=1}^{m-1}\gamma_j^2 \geq 2x_1\|\tilde{u}_{t-1}\|_2 - x_1^2\right) \leq 2\exp(-c_{norm}(m-1)\nu^2/2) = 2\exp\left(\frac{-c_{norm}x_1\|\tilde{u}_{t-1}\|_2}{2\sigma^2}\right).$$

Consequentially, we can bound the integral over S as follows

$$\begin{aligned}
&(2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta \int_{B(\tilde{u}_{t-1},\|\tilde{u}_{t-1}\|)^C\cap S} e^{2\lambda w_t\|u_{t-1}\|x_1 - \frac{x_1^2}{2\sigma^2}}\mathbb{P}\left(\sigma^2\sum_{j=1}^{m-1}\gamma_j^2 \geq 2x_1\|\tilde{u}_{t-1}\|_2 - x_1^2\right)\,dx_1 \\
&\leq 2\cdot(2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta \int_{\frac{C_0\sigma^2 m}{\|\tilde{u}_{t-1}\|}}^{\|\tilde{u}_{t-1}\|} e^{2\lambda w_t\|u_{t-1}\|x_1 - \frac{x_1^2}{2\sigma^2} - \frac{c_{norm}x_1\|\hat{u}_{t-1}\|_2}{2\sigma^2}}\,dx_1 \\
&= 2\cdot(2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta \int_{\frac{C_0\sigma^2 m}{\|\tilde{u}_{t-1}\|}}^{\|\tilde{u}_{t-1}\|} e^{\left(2\lambda w_t\|u_{t-1}\|_2 - \frac{c_{norm}\|\tilde{u}_{t-1}\|}{2\sigma^2}\right)x_1 - \frac{x_1^2}{2\sigma^2}}\,dx_1. \qquad (2.9.15)
\end{aligned}$$

Setting $2\zeta := c_{norm}\|\tilde{u}\|_2 - 4\lambda\sigma^2 w_t\|u_{t-1}\|$, we have that (2.9.15) is equal to

$$\begin{aligned}
2\cdot(2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta \int_{\frac{C_0\sigma^2 m}{\|\tilde{u}_{t-1}\|}}^{\|\tilde{u}_{t-1}\|} e^{\frac{-2\zeta x_1}{2\sigma^2} - \frac{x_1^2}{2\sigma^2}}\,dx_1 &= 2\cdot(2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta e^{\frac{\zeta^2}{2\sigma^2}} \int_{\frac{C_0\sigma^2 m}{\|\tilde{u}_{t-1}\|}}^{\|\tilde{u}_{t-1}\|} e^{\frac{-1}{2\sigma^2}(x_1+\zeta)^2}\,dx_1 \\
&\leq 2\cdot(2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta e^{\frac{\zeta^2}{2\sigma^2}} \int_{\frac{C_0\sigma^2 m}{\|\tilde{u}_{t-1}\|}+\zeta}^{\infty} e^{\frac{-x_1^2}{2\sigma^2}}\,dx_1. \qquad (2.9.16)
\end{aligned}$$

We remark that $\zeta > 0$ if $-2\lambda w_t + \frac{c_{norm}}{2|1-2w_t|\sigma^2} > 0$ which holds since $\lambda < \frac{c_{norm}}{12\sigma^2} < \frac{c_{norm}}{4w_t|1-2w_t|\sigma^2}$. Therefore, the lower limit of integration is positive and we can use a Gaussian tail bound as in Lemma 2.8.1 to upper bound (2.9.16) by

$$\frac{\mathbb{1}_\beta 2\sigma e^{\frac{-1}{2\sigma^2}\left(\frac{C_0^2\sigma^4 m^2}{\|\tilde{u}_{t-1}\|^2} + 2\frac{C_0\sigma^2 m\zeta}{\|\tilde{u}_{t-1}\|_2}\right)}}{\sqrt{2\pi}\left(\frac{C_0\sigma^2 m}{\|\tilde{u}_{t-1}\|}+\zeta\right)} \leq \frac{\mathbb{1}_\beta 2\sigma}{\sqrt{2\pi}\zeta} = \frac{\mathbb{1}_\beta 4\sigma}{\sqrt{2\pi}\|u_{t-1}\|_2\left(\frac{c_{norm}}{|1-2w_t|} - 4\lambda\sigma^2 w_t\right)} \qquad (2.9.17)$$

$$\leq \frac{\mathbb{1}_\beta 4\sigma}{\|u_{t-1}\|_2\left(\frac{c_{norm}}{3}-\frac{4C_\lambda}{C_{\sup}^2 m\log(N_0)}\right)}. \tag{2.9.18}$$

Putting it all together, and remembering to add back in the factor $e^{\lambda C_{\sup}^2\sigma^2 m\log(N_0)w_t^2}=e^{C_\lambda w_t^2}\leq e^{C_\lambda}$ we have previously ignored, we've bound $\mathbb{E}\left[e^{\lambda\Delta\|u_t\|_2^2}\mathbb{1}_\beta\mathbb{1}_{q_t=0}\mathbb{1}_U\mathbb{1}_{\|u_{t-1}\|_2\geq\beta}\middle|\mathscr{F}_{t-1}\right]$ from above with

$$\underbrace{\frac{\mathbb{1}_\beta e^{C_\lambda}C_{\sup}^2\sigma m\log(N_0)}{C_\lambda\varepsilon\|u_{t-1}\|_2}}_{(2.9.6)}+\underbrace{\frac{\mathbb{1}_\beta 3C_0 e^{\frac{6C_0C_\lambda}{C_{\sup}^2\log(N_0)}+C_\lambda}\sigma m}{\|u_{t-1}\|_2}}_{(2.9.11)}+\underbrace{\frac{\mathbb{1}_\beta e^{C_\lambda}\sigma}{\|u_{t-1}\|_2\left(\frac{1}{3}-\frac{2C_\lambda}{C_{\sup}^2 m\log(N_0)}\right)}}_{(2.9.14)}$$
$$+\underbrace{\frac{\mathbb{1}_\beta 4\sigma e^{C_\lambda}}{\|u_{t-1}\|_2\left(\frac{c_{norm}}{3}-\frac{4C_\lambda}{C_{\sup}^2 m\log(N_0)}\right)}}_{(2.9.17)}\lesssim\frac{\mathbb{1}_\beta e^{C_\lambda}\sigma m\log(N_0)}{\|u_{t-1}\|_2\varepsilon}.$$

So, when $|w_t|<1/2$ and $\|u_{t-1}\|_2\geq\beta\gtrsim\frac{\sigma m\log(N_0)}{\rho\varepsilon}$ the claim follows.

Now, let's consider the case when $w_t\geq 1/2$. Then it must be, by Lemma 2.8.3, that $X_t\in B(\tilde{u}_{t-1},\|\tilde{u}_{t-1}\|_2)\cap B(\hat{u}_{t-1},\|\hat{u}_{t-1}\|_2)^C$. By non-negativity of the exponential function, we can always upper-bound the moment generating function by instead integrating over $X_t\in B(\hat{u}_{t-1},\|\hat{u}_{t-1}\|_2)^C\cap\{x_1\leq 0\}$. Pictorially, one can see this by looking at the subfigure on the right in Figure 2.8.1. In this scenario, we're integrating over the region in green. The upper bound we're proposing is derived by ignoring the constraint from the blue region on the left half-space. Using this upper bound we can retrace through the steps we took to bound the integrals over R,S, and T with only minor modifications and obtain the desired result. By symmetry, an analogous approach will work for $w_t\leq -1/2$. □

Lemma 2.9.4. With the same hypotheses as Lemma 2.9.3,

$$\mathbb{E}\left[e^{\lambda\Delta\|u_t\|_2^2}\mathbb{1}_\beta\mathbb{1}_{q_t=1}\mathbb{1}_U\mathbb{1}_{\|u_{t-1}\|_2\geq\beta}\middle|\mathscr{F}_{t-1}\right]\leq\rho.$$

Proof. To begin, let's consider the case when $w_t < 1/2$. Recalling that $\tilde{u}_{t-1} = \frac{1}{1-2w_t}u_{t-1}$, and arguing as we did at the beginning of the proof of Lemma 2.9.3, Lemma 2.8.3 tells us

$$\begin{aligned}
&\mathbb{E}\left[e^{\lambda\Delta\|u_t\|_2^2}\mathbb{1}_\beta \mathbb{1}_{q_t=1}\mathbb{1}_U \mathbb{1}_{\|u_{t-1}\|_2\geq\beta}\middle|\mathscr{F}_t-1\right] \\
&\leq (2\pi\sigma^2)^{-m/2}\mathbb{1}_\beta e^{\lambda C_{\sup}^2\sigma^2 m\log(N_0)(w_t-1)^2}\int\limits_{B(\tilde{u}_{t-1},\|\tilde{u}_{t-1}\|_2)} e^{2\lambda(w_t-1)x^T u_{t-1}}e^{\frac{-1}{2\sigma^2}\|x\|_2^2}\,dx.
\end{aligned}$$

As before, we have denoted $\mathbb{1}_\beta := \mathbb{1}_{\|u_{t-1}\|_2\geq\beta}$ for conciseness. Using rotational invariance, we may assume that $u_{t-1} = \|u_{t-1}\|_2 e_1$. Just as we did in Lemma 2.9.3, expressing this integral as nested iterated integrals gives us the probabilistic formulation

$$\frac{\mathbb{1}_\beta e^{\lambda C_{\sup}^2\sigma^2 m\log(N_0)(w_t-1)^2}}{\sqrt{2\pi}\sigma}\int_0^{2\|\tilde{u}_{t-1}\|_2} e^{2\lambda(w_t-1)\|u_{t-1}\|x_1-\frac{1}{2\sigma^2}x_1^2}\mathbb{P}\left(\sigma^2\sum_{j=1}^{m-1}\gamma_j^2 \leq 2x_1\|\tilde{u}_{t-1}\|_2 - x_1^2\right)\,dx_1,$$

where, as before, the $\gamma_j \sim \mathscr{N}(0,1)$ are i.i.d. standard normal random variables. So, consider decomposing the above integral into the following two pieces. Set $R := \{x\in\mathbb{R}^m : 0\leq x_1 \leq \frac{C_1\sigma^2 m}{\|\tilde{u}_{t-1}\|_2}\}$ and $S := \{x\in\mathbb{R}^m : \frac{C_1\sigma^2 m}{\|\tilde{u}_{t-1}\|_2}\leq x_1 \leq 2\|\tilde{u}_{t-1}\|_2\}$ where $C_1\in(0,1)$ is a fixed constant. Then on R we have by Lemma 2.8.2

$$\begin{aligned}
&\mathbb{P}\left(\sum_{j=1}^{m-1} g_j^2 \leq (m-1)\left(\frac{1}{\sigma^2(m-1)}(2x_1\|\tilde{u}_{t-1}\|_2 - x_1^2)\right)\right) \\
&\leq \mathbb{P}\left(\sum_{j=1}^{m-1} g_j^2 \leq (m-1)\left(\frac{1}{\sigma^2(m-1)}(2x_1\|\tilde{u}_{t-1}\|_2)\right)\right) \leq 2e^{-c(1-C_1)^2(m-1)}.
\end{aligned}$$

Setting aside the factor $e^{\lambda C_{\sup}^2\sigma^2 m\log(N_0)(w_t-1)^2}$ for the moment, we have that the integral over R is equal to

$$(2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta\int_0^{\frac{C_1\sigma^2 m}{\|\tilde{u}_{t-1}\|_2}} e^{2\lambda(w_t-1)\|u_{t-1}\|x_1-\frac{1}{2\sigma^2}x_1^2}\mathbb{P}\left(\sigma^2\sum_{j=1}^{m-1}\gamma_j^2 \leq 2x_1\|\tilde{u}_{t-1}\|_2 - x_1^2\right)\,dx$$

$$\leq (2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta 2e^{-c(1-C_1)^2(m-1)}\int_0^{\frac{C_1\sigma^2 m}{\|\tilde{u}_{t-1}\|_2}} e^{2\lambda(w_t-1)\|u_{t-1}\|x_1-\frac{1}{2\sigma^2}x_1^2}dx_1$$

$$= (2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta 2e^{-c(1-C_1)^2(m-1)}e^{2\sigma^2\lambda^2(w_t-1)^2\|u_{t-1}\|_2^2}\int_{2\lambda\sigma^2(1-w_t)\|u_{t-1}\|_2}^{\frac{C_1\sigma^2 m}{\|\tilde{u}_{t-1}\|_2}+2\lambda\sigma^2(1-w_t)\|u_{t-1}\|_2} e^{-\frac{1}{2\sigma^2}x_1^2}dx_1. \tag{2.9.19}$$

We remark that the lower limit of integration is strictly positive. Therefore, using a Riemann approximation to the integral and knowing that the maximum of the integral occurs at the lower limit of integration bounds (2.9.19) above by

$$\mathbb{1}_\beta 2e^{-c(1-C_1)^2(m-1)}\frac{C_1\sigma m}{\|\tilde{u}_{t-1}\|_2\sqrt{2\pi}}. \tag{2.9.20}$$

On S, we use the bound $\mathbb{P}\left(\sum_{j=1}^{m-1} g_j^2 \leq (m-1)\left(\frac{1}{\sigma^2(m-1)}(2x_1\|\tilde{u}_{t-1}\|_2 - x_1^2)\right)\right) \leq 1$ to get

$$(2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta \int_{\frac{C_1\sigma^2 m}{\|\tilde{u}_{t-1}\|_2}}^{2\|\tilde{u}_{t-1}\|_2} e^{2\lambda(w_t-1)\|u_{t-1}\|x_1-\frac{1}{2\sigma^2}x_1^2}\mathbb{P}\left(\sum_{j=1}^{m-1} g_j^2 \leq (m-1)\left(\frac{2x_1\|\tilde{u}_{t-1}\|_2 - x_1^2}{\sigma^2(m-1)}\right)\right) dx_1$$

$$\leq (2\pi\sigma^2)^{-1/2}\mathbb{1}_\beta \int_{\frac{C_1\sigma^2 m}{\|\tilde{u}_{t-1}\|_2}}^{2\|\tilde{u}_{t-1}\|_2} e^{2\lambda(w_t-1)\|u_{t-1}\|x_1-\frac{1}{2\sigma^2}x_1^2}dx_1$$

$$= \mathbb{1}_\beta e^{2\sigma^2\lambda^2(w_t-1)^2\|u_{t-1}\|_2^2}\int_{\frac{C_1\sigma^2 m}{\|\tilde{u}_{t-1}\|_2}+2\lambda\sigma^2(1-w_t)\|u_{t-1}\|_2}^{2\|\tilde{u}_{t-1}\|_2+2\lambda\sigma^2(1-w_t)\|u_{t-1}\|_2}(2\pi\sigma^2)^{-1/2}e^{-\frac{1}{2\sigma^2}x_1^2}dx_1. \tag{2.9.21}$$

Since the lower limit of integration is strictly positive, we can use a Gaussian tail bound as in Lemma 2.8.1 to upper bound (2.9.21) by

$$\frac{\mathbb{1}_\beta\sigma e^{-\frac{1}{2\sigma^2}\left(\frac{C_1^2\sigma^4m^2}{\|\tilde{u}_{t-1}\|_2^2}+4\lambda(1-w_t)C_1\sigma^4m\frac{\|u_{t-1}\|_2}{\|\tilde{u}_{t-1}\|_2}\right)}}{\left(\frac{C_1\sigma^2 m}{\|\tilde{u}_{t-1}\|_2}+2\lambda\sigma^2(1-w_t)\|u_{t-1}\|_2\right)\sqrt{2\pi}} \leq \frac{\mathbb{1}_\beta\sigma}{\sqrt{2\pi}2\lambda\sigma^2(1-w_t)\|u_{t-1}\|}. \tag{2.9.22}$$

To summarize, we have shown, at least when $w_t < 1/2$, that

$$\begin{aligned}
&\mathbb{E}\left[e^{\lambda\Delta\|u_t\|_2^2}\mathbb{1}_\beta\mathbb{1}_{q_t=1}\mathbb{1}_U\middle|\mathscr{F}_{t-1}\right]\\
&\leq \underbrace{\mathbb{1}_\beta e^{\lambda C_{\sup}^2\sigma^2 m\log(N_0)(w_t-1)^2}2e^{-c(1-C_1)^2(m-1)}\frac{C_1\sigma m}{\|\tilde{u}_{t-1}\|_2\sqrt{2\pi}}}_{(2.9.20)}+\underbrace{\frac{\mathbb{1}_\beta e^{\lambda C_{\sup}^2\sigma^2 m\log(N_0)(w_t-1)^2}\sigma}{\sqrt{2\pi}2\lambda\sigma^2(1-w_t)\|u_{t-1}\|}}_{(2.9.22)}\\
&\leq \frac{\mathbb{1}_\beta e^{\lambda C_{\sup}^2\sigma^2 m\log(N_0)(w_t-1)^2}}{\|u_{t-1}\|_2}\left(\frac{2\sigma m|1-2w_t|}{\sqrt{2\pi}}+\frac{\sigma}{\sqrt{2\pi}2\lambda\sigma^2(1-w_t)}\right)\\
&\leq \frac{\mathbb{1}_\beta e^{\lambda C_{\sup}^2\sigma^2 m\log(N_0)(w_t-1)^2}}{\|u_{t-1}\|_2}\left(6\sigma m+\frac{\sigma}{\lambda\sigma^2\varepsilon}\right)\\
&\leq \frac{\mathbb{1}_\beta e^{4C_\lambda}\sigma m\log(N_0)}{\|u_{t-1}\|_2}\left(\frac{6}{\log(N_0)}+\frac{1}{C_\lambda\varepsilon}\right)\\
&\lesssim \frac{\mathbb{1}_\beta\sigma m\log(N_0)}{\|u_{t-1}\|_2\varepsilon}.
\end{aligned}\tag{2.9.23}$$

Therefore, when $\|u_{t-1}\|_2 \geq \beta \gtrsim \frac{Ce^{C_\lambda}\sigma m\log(N_0)}{\rho\varepsilon}$, (2.9.23) is bounded above by ρ as desired.

Now, let's consider the case when $w_t > 1/2$. In this scenario, we can express the expectation as

$$\begin{aligned}
&\mathbb{E}\left[e^{\lambda\Delta\|u_t\|_2^2}\mathbb{1}_\beta\mathbb{1}_{q_t=1}\mathbb{1}_U\middle|\mathscr{F}_{t-1}\right]\\
&\leq e^{\lambda C_{\sup}^2\sigma^2 m\log(N_0)(w_t-1)^2}(2\pi\sigma^2)^{-m/2}\mathbb{1}_\beta\int_{B(\tilde{u}_{t-1},\|\tilde{u}_{t-1}\|_2)^C}e^{2\lambda(w_t-1)x^Tu_{t-1}}e^{\frac{-1}{2\sigma^2}\|x\|_2^2}\,dx.
\end{aligned}$$

Using the exact same approach as in the proof of Lemma 2.9.3, we can partition the domain of integration into the following pieces:

$$\begin{aligned}
\int_{B(\tilde{u}_{t-1},\|\tilde{u}_{t-1}\|_2)^C} e^{2\lambda(w_t-1)x^Tu_{t-1}}e^{\frac{-1}{2\sigma^2}\|x\|_2^2}\,dx \quad &= \int_{B(\tilde{u}_{t-1},\|\tilde{u}_{t-1}\|_2)^C\cap\{x_1\leq-\|\tilde{u}_{t-1}\|\}} e^{2\lambda(w_t-1)x^Tu_{t-1}}e^{\frac{-1}{2\sigma^2}\|x\|_2^2}\,dx\\
&+\int_{B(\tilde{u}_{t-1},\|\tilde{u}_{t-1}\|_2)^C\cap\{-\|\tilde{u}_{t-1}\|\leq x_1\leq\frac{-C\sigma^2 m}{\|\tilde{u}_{t-1}\|}\}} e^{2\lambda(w_t-1)x^Tu_{t-1}}e^{\frac{-1}{2\sigma^2}\|x\|_2^2}\,dx
\end{aligned}$$

$$+ \int_{B(\tilde{u}_{t-1},\|\tilde{u}_{t-1}\|_2)^C \cap \{-\frac{-C\sigma^2 m}{\|\tilde{u}_{t-1}\|} \le x_1 \le 0\}} e^{2\lambda(w_t-1)x^T u_{t-1}} e^{\frac{-1}{2\sigma^2}\|x\|_2^2}\, dx$$

$$+ \int_{B(\tilde{u}_{t-1},\|\tilde{u}_{t-1}\|_2)^C \cap \{0 \le x_1\}} e^{2\lambda(w_t-1)x^T u_{t-1}} e^{\frac{-1}{2\sigma^2}\|x\|_2^2}\, dx.$$

The same arguments from the proof of Lemma 2.9.3 apply here with only minor modifications. Namely, an argument exactly like that given for (2.9.3) gives us

$$\int_{B(\tilde{u}_{t-1},\|\tilde{u}_{t-1}\|_2)^C \cap \{0 \le x_1\}} e^{2\lambda(w_t-1)x^T u_{t-1}} e^{\frac{-1}{2\sigma^2}\|x\|_2^2}\, dx \le \frac{\sigma}{\lambda\sigma^2\varepsilon\|u_{t-1}\|}.$$

Similarly, the chain of logic used to derive (2.9.14) gives us

$$\int_{B(\tilde{u}_{t-1},\|\tilde{u}_{t-1}\|_2)^C \cap \{x_1 \le -\|\tilde{u}_{t-1}\|\}} e^{2\lambda(w_t-1)x^T u_{t-1}} e^{\frac{-1}{2\sigma^2}\|x\|_2^2}\, dx \le \frac{\sigma}{\|u_{t-1}\|_2\left(\frac{1}{3} - \frac{2C_\lambda}{C_{\sup}^2 m\log(N_0)}\right)}.$$

Calculations for the derivation of (2.9.17) give us

$$\int_{B(\tilde{u}_{t-1},\|\tilde{u}_{t-1}\|_2)^C \cap \{-\|\tilde{u}_{t-1}\| \le x_1 \le \frac{-C\sigma^2 m}{\|\tilde{u}_{t-1}\|}\}} e^{2\lambda(w_t-1)x^T u_{t-1}} e^{\frac{-1}{2\sigma^2}\|x\|_2^2}\, dx \le \frac{4\sigma}{\|u_{t-1}\|_2\left(\frac{c_{norm}}{3} - \frac{4C_\lambda}{C_{\sup}^2 m\log(N_0)}\right)}.$$

Finally, the same reasoning that was used to derive (2.9.11) gives us

$$\int_{B(\tilde{u}_{t-1},\|\tilde{u}_{t-1}\|_2)^C \cap \{-\frac{-C\sigma^2 m}{\|\tilde{u}_{t-1}\|} \le x_1 \le 0\}} e^{2\lambda(w_t-1)x^T u_{t-1}} e^{\frac{-1}{2\sigma^2}\|x\|_2^2}\, dx \le \frac{3C_0 e^{\frac{6C_0C_\lambda}{C_{\sup}^2\log(N_0)}}\sigma m}{\|u_{t-1}\|_2}.$$

Following the remainder of the proof of Lemma 2.9.3 in this scenario gives us the result when $w_t > 1/2$. □

Lemma 2.9.5. With the same hypotheses as Lemma 2.9.3

$$\mathbb{E}\left[e^{\lambda\Delta\|u_t\|_2^2}\mathbb{1}_{q_t=-1}\mathbb{1}_U\mathbb{1}_{\|u_{t-1}\|_2\ge\beta}\middle|\mathscr{F}_t-1\right] \le \rho.$$

Proof. The proof is effectively the same as that for Lemma 2.9.4. □

Acknowledgements

This chapter, in full, is joint work with Rayan Saab, and it has been submitted for publication as it may appear in the Journal of Machine Learning Research, 2021. The dissertation author was the primary investigator and author of this paper. This work was supported in part by National Science Foundation Grant DMS-2012546 and a UCSD senate research award.

Bibliography

[1] Miklós Ajtai. The shortest vector problem in l2 is np-hard for randomized reductions. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 10–19, 1998.

[2] Pierre Baldi and Roman Vershynin. The capacity of feedforward neural networks. *Neural networks*, 116:288–311, 2019.

[3] Wojciech Banaszczyk. A beck—fiala-type theorem for euclidean norms. *European Journal of Combinatorics*, 11(6):497–500, 1990.

[4] Wojciech Banaszczyk. Balancing vectors and gaussian measures of n-dimensional convex bodies. *Random Structures & Algorithms*, 12(4):351–360, 1998.

[5] Nikhil Bansal. Constructive algorithms for discrepancy minimization. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 3–10. IEEE, 2010.

[6] Nikhil Bansal, Daniel Dadush, Shashwat Garg, and Shachar Lovett. The gram-schmidt walk: a cure for the banaszczyk blues. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 587–597, 2018.

[7] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in neural information processing systems*, pages 3123–3131, 2015.

[8] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[9] Daniel Dadush, Shashwat Garg, Shachar Lovett, and Aleksandar Nikolov. Towards a constructive version of banaszczyk's vector balancing theorem. *arXiv preprint arXiv:1612.04304*, 2016.

[10] Ingrid Daubechies and Ron DeVore. Approximating a bandlimited function using very coarsely quantized data: A family of stable sigma-delta modulators of arbitrary order. *Annals of mathematics*, 158(2):679–710, 2003.

[11] Ronen Eldan and Mohit Singh. Efficient algorithms for discrepancy minimization in convex sets. *arXiv preprint arXiv:1409.2913*, 2014.

[12] Apostolos A Giannopoulos. On some vector balancing problems. *Studia Mathematica*, 122(3):225–234, 1997.

[13] Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115*, 2014.

[14] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[15] Yunhui Guo. A survey on methods and theories of quantized neural networks. *arXiv preprint arXiv:1808.04752*, 2018.

[16] Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *International Conference on Machine Learning*, pages 1737–1746, 2015.

[17] Bruce Hajek. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied probability*, pages 502–525, 1982.

[18] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149, conference paper at ICLR*, 2016.

[19] Nicholas JA Harvey, Roy Schwartz, and Mohit Singh. Discrepancy without partial colorings. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2014)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

[20] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[21] Hi Inose, Y Yasuda, and Jun Murakami. A telemetering system by code modulation-δ-σmodulation. *IRE Transactions on Space Electronics and Telemetry*, (3):204–209, 1962.

[22] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[23] Yong-Deok Kim, Eunhyeok Park, Sungjoo Yoo, Taelim Choi, Lu Yang, and Dongjun Shin. Compression of deep convolutional neural networks for fast and low power mobile applications. *arXiv preprint arXiv:1511.06530, conference paper at ICLR*, 2016.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[25] Tamara G Kolda and Dianne P O'leary. A semidiscrete matrix decomposition for latent semantic indexing information retrieval. *ACM Transactions on Information Systems (TOIS)*, 16(4):322–346, 1998.

[26] Richard Kueng and Joel A Tropp. Binary component decomposition part ii: The asymmetric case. *arXiv preprint arXiv:1907.13602*, 2019.

[27] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[28] László Lovász, Joel Spencer, and Katalin Vesztergombi. Discrepancy of set-systems and matrices. *European Journal of Combinatorics*, 7(2):151–160, 1986.

[29] Shachar Lovett and Raghu Meka. Constructive discrepancy minimization by walking on the edges. *SIAM Journal on Computing*, 44(5):1573–1582, 2015.

[30] Mikhail Menshikov, Serguei Popov, and Andrew Wade. *Non-homogeneous random walks: Lyapunov function methods for near-critical stochastic systems*, volume 209. Cambridge University Press, 2016.

[31] Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

[32] Robin Pemantle and Jeffrey S Rosenthal. Moment conditions for a sequence with negative drift to be uniformly bounded in lr. *Stochastic Processes and their Applications*, 82(1):143–155, 1999.

[33] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pages 525–542. Springer, 2016.

[34] Thomas Rothvoss. Constructive discrepancy minimization for convex sets. *SIAM Journal on Computing*, 46(1):224–234, 2017.

[35] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[36] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever,

Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[37] Joel Spencer. Six standard deviations suffice. *Transactions of the American mathematical society*, 289(2):679–706, 1985.

[38] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[39] Peisong Wang and Jian Cheng. Fixed-point factorized networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4012–4020, 2017.

[40] Juyang Weng, Narendra Ahuja, and Thomas S Huang. Cresceptron: a self-organizing neural network which grows adaptively. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 1, pages 576–581. IEEE, 1992.

Chapter 3

Quantization for Low-Rank Matrix Recovery

We study Sigma-Delta ($\Sigma\Delta$) quantization methods coupled with appropriate reconstruction algorithms for digitizing randomly sampled low-rank matrices. We show that the reconstruction error associated with our methods decays polynomially with the oversampling factor, and we leverage our results to obtain root-exponential accuracy by optimizing over the choice of quantization scheme. Additionally, we show that a random encoding scheme, applied to the quantized measurements, yields a near-optimal exponential bit-rate. As an added benefit, our schemes are robust both to noise and to deviations from the low-rank assumption. In short, we provide a full generalization of analogous results, obtained in the classical setup of bandlimited function acquisition, and more recently, in the finite frame and compressed sensing setups to the case of low-rank matrices sampled with sub-Gaussian linear operators. Finally, we believe our techniques for generalizing results from the compressed sensing setup to the analogous low-rank matrix setup is applicable to other quantization schemes.

3.1 Introduction

Let $\mathscr{M} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$ be a linear map that acts on matrices X to produce measurements

$$y = \mathscr{M}(X) = \sum_{i=1}^{m} \langle X, A_i \rangle e_i, \tag{3.1.1}$$

where the vectors e_i are the standard basis vectors for $\mathbb{R}^m$, and each A_i is a matrix in $\mathbb{R}^{n_1 \times n_2}$, $i \in \{1, ..., m\}$. Here, the inner product is the standard Hilbert-Schmidt inner product given by $\langle Y, Z \rangle = \sum_{i,j} Y_{ij} Z_{ij}$. Note that for every linear operator $\mathscr{M}$ as above, there exists an $m \times (n_1 n_2)$ matrix $A_{\mathscr{M}}$, such that for all $X \in \mathbb{R}^{n_1 \times n_2}$

$$\mathscr{M}(X) = A_{\mathscr{M}} \vec{X}.$$

Here $\vec{X} \in \mathbb{R}^{n_1 n_2}$, the vectorized version of the matrix X, is obtained by stacking the columns of X. Low-rank matrix recovery is concerned with approximating a rank k matrix X from y, knowing the operator $\mathscr{M}$. It is primarily interesting in the regime where $m \ll n_1 n_2$, and many recent results propose recovery algorithms and prove recovery guarantees when $m \geq Ck \max\{n_1, n_2\}$ where $C > 0$ is some absolute constant [7, 8, 23, 33]. For example, when the entries of the matrix representation $A_{\mathscr{M}}$ are independent Gaussian or sub-Gaussian random variables, and $y = \mathscr{M}(X) + e$ with $\|e\|_2 \leq \varepsilon$, one can solve the convex optimization problem

$$X^\sharp := \arg\min_Z \|Z\|_* \quad \text{subject to} \quad \|\mathscr{M}(Z) - y\|_2 \leq \varepsilon. \tag{3.1.2}$$

Then, with high probability on the draw of $\mathscr{M}$, and uniformly for all $n_1 \times n_2$ matrices X, we have

$$\|X^\sharp - X\|_F \leq C \left(\frac{\sigma_k(X)_*}{\sqrt{k}} + \varepsilon \right), \tag{3.1.3}$$

as shown in, e.g., [7]. Above, $\|\cdot\|_F$ denotes the Frobenius norm $\|X\|_F = \sqrt{\sum_{i,j} A_{i,j}^2}$ on matrices induced by the Hilbert-Schmidt inner product, $\|Z\|_*$ denotes the nuclear norm of Z, i.e., the sum of its singular values, and

$$\sigma_k(Z)_* := \min_{\text{rank}(V)=k} \|Z - V\|_*$$

denotes the error, measured in the nuclear norm, associated with the best rank k approximation of a matrix.

3.1.1 Background and prior work

Low-rank matrix recovery has seen a wide range of applications, ranging from quantum state tomography [17] and collaborative filtering [31] to sensor localization [32] and face recognition [6], to name a few.

Nuclear norm minimization was proposed by Fazel in [14] as a means of finding a matrix of minimial rank in a given convex set. Fazel motivates this through the observation that nuclear norm minimization is the convex relaxation of the rank minimization problem, which has been shown to be NP-hard. Since then, and with the advent of compressed sensing, there has been much work on recovering low-rank matrices from linear measurements. For example, [33] considers recovering low-rank matrices given random linear measurements, and establishes recovery guarantees given that the sampling scheme satisfies the matrix restricted isometry property, which we define in the subsequent section. Perhaps not surprisingly, this analysis closely follows that of sparse vector recovery under ℓ_1 minimization, as it is known that random ensembles of linear maps satisfy the matrix restricted isometry property with high probability [16, 33]. This led to a flurry of papers on nuclear norm minimization for matrix recovery in various contexts, see [6, 29, 9] for example.

While the theoretical results on nuclear norm minimization have been promising, convex optimization practically necessitates the use of digital computers for recovering the underlying

matrix. It behooves the theory, therefore, to take into account that the measurements must be converted to bits so that numerical solvers can handle them. Indeed, quantization is the necessary step in data acquisition by which measurements taking values in the continuum are mapped to discrete sets. Without any claim to comprehensiveness, we are aware of the following developments on quantization in the low-rank matrix completion setting, i.e., the setting where one quantizes a random subset of the entries of the matrix directly.

Davenport and coauthors in [11] consider recovering a rank k matrix $X \in \mathbb{R}^{n_1 \times n_2}$ given 1-bit measurements of a subset of the entries, sampled according to a distribution that *may depend on the entries*. They recover an estimate $\hat{X}$ of the sampled matrix through maximum likelihood estimation with a nuclear norm constraint and derive error bounds which decay, as a function of the number of measurements m, like $O(m^{-1/2})$.

Shortly thereafter Cai and Zhou in [4] consider reconstruction given 1-bit measurements of the entries under more general sampling schemes of the indices. Unlike the argument in [11], Cai and Zhou impose a max-norm constraint on the maximum likelihood estimation to enforce the low-rank condition. Under this regime the scaled Frobenius norm error decay is also $O(m^{-1/2})$.

Bhaskar and Javanmard in [2] modify the optimization problem of [11] so that it now imposes an exact rank constraint in place of the nuclear norm. This yields a non-convex problem with associated computational challenges. Nevertheless, assuming one can solve this hard optimization problem, they obtain an error estimate that decays like $O(m^{-4})$, at the added cost of a much increased constant that scales like $n^7 k^3$.

Proceeding towards more general quantization alphabets, [28] consider low-rank matrix completion via nuclear norm penalized maximum liklihood estimation given quantized measurements of the entries with unknown quantization bin boundaries. They propose an optimization procedure which learns the quantization bin boundaries and recovers the matrix in an alternating fashion. No theoretical guarantees are given to delineate the relationship between the number of measurements and the reconstruction error.

Authors in [27] propose a low-rank matrix recovery algorithm given quantized measurements of the entries from a finite alphabet under some sampling distribution of the indices. As the aforementioned schemes have done, they propose a maximum liklihood estimation but with a nuclear norm constraint to enforce the low-rank condition. Specifically, given $m \geq C \max\{n_1, n_2\} \log(\max\{n_1, n_2\})$ measurements where $C > 0$ is a universal constant, they show that the scaled Frobenius error decays like m^{-1}.

In contrast to the above works, we study the quantization problem in the low-rank matrix recovery setting given linear measurements of the form (3.1.1), where the matrices A_i are sub-Gaussian.

3.1.2 Contributions

To the best of our knowledge, we provide the first theoretical guarantees of low-rank matrix recovery from $\Sigma\Delta$ quantized sub-Gaussian linear measurements. Our result holds for stable $\Sigma\Delta$ quantizers (defined in Section 3.2.2) of arbitrary order and our bounds apply to the particular case of 1-bit quantization; that is, we can recover scaling information in this setting. Thus, we generalize a result from [34] that recovers sparse vectors from quantized noisy measurements so that it now applies to the low-rank matrix setting, as shown in Theorem 4.4.1. Our main tool for achieving this extension is a modification of the technique of Oymak et al. [30] for converting compressed sensing results to the low-rank matrix setting. We show that the reconstruction error under constrained nuclear norm minimization is bounded by

$$\|X^\sharp - X\|_F \leq C \left(\left(\frac{m}{\ell} \right)^{-r+1/2} \beta + \frac{\sigma_k(X)_*}{\sqrt{k}} + \sqrt{\frac{m}{\ell}} \varepsilon \right)$$

thus showing that our reconstruction scheme is robust to noise and to the low-rank assumption. Above, r denotes the order of the $\Sigma\Delta$ scheme and β the step-size of the associated alphabet (see Section 3.2.2), ℓ is of order $k \max\{n_1, n_2\}$, and $\frac{m}{\ell}$ denotes the oversampling factor. Note that in the

case of rank k matrices, with no measurement noise, our reconstruction error decays polynomially fast, namely as m^{-r}, thereby greatly improving on the rates obtained in the works cited above. Furthermore, by optimizing over the order of the $\Sigma\Delta$ reconstruction scheme, we show in Corollary 3.3.2 that our procedure attains root-exponential accuracy with respect to the oversampling factor. This generalizes the error decay seen in [34] for vectors.

The robustness of the main result extends beyond quantization. We show in Corollary 3.3.3 that we can further reduce the total number of bits, by encoding the quantized measurements using a discrete Johnson-Lindenstrauss [22] embedding into a lower dimensional space. The resulting dramatic reduction in bit-rate is coupled with only a small increase in reconstruction error. This, in turn yields an exponentially decaying, i.e., optimal, relationship between number of bits and reconstruction error.

Finally, we remark that the techniques used herein can be used to derive analogous results for other quantization schemes that share certain properties of $\Sigma\Delta$ quantization. Namely, suppose one is given a quantization map $\mathscr{Q}$ and a bijective linear map $T : \mathbb{R}^n \to \mathbb{R}^n$ which satisfy $\|T(y-\mathscr{Q}(y))\| < C$ for some norm $\|\cdot\|$ and some constant C that may depend on the quantization technique but not on the dimensions. Then, the proof of Theorem 4.4.1, with a suitably altered decoder, can likely be modified to produce an analogous result for the new quantization scheme.

3.2 Preliminaries

3.2.1 Notation

For $x \in \mathbb{R}^n$, let $\mathrm{supp}(x)$ denote the set of indices i for which x_i is non-zero, and $\Sigma_k^n := \{x \in \mathbb{R}^n, |\mathrm{supp}(x)| \leq k\}$ be the set of all k-sparse vectors in $\mathbb{R}^n$. For a matrix $A \in \mathbb{R}^{n_1 \times n_2}$, we will denote its singular values by $\sigma_i(A)$ for $i = 1, ..., n$ where $n := \min\{n_1, n_2\}$ and $\sigma_1(A) \geq \sigma_2(A) \geq ... \geq \sigma_n(A)$. We will require the definitions of the well known restricted isometry property (RIP), both for linear operators acting on sparse vectors and for linear operators acting on low-rank

matrices.

Definition 3.2.1 (*vector-RIP* (e.g., [5]))**.** We say a linear operator $\Phi : \mathbb{R}^n \to \mathbb{R}^m$ satisfies the vector-RIP of order k and constant δ_k, if for all $x \in \Sigma_k^n$,

$$(1-\delta_k)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1+\delta_k)\|x\|_2^2.$$

Definition 3.2.2 (*matrix-RIP*)**.** We say a linear operator $\mathscr{M} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$ satisfies the matrix-RIP of order k and constant δ_k, if for all matrices X of rank k or less we have,

$$(1-\delta_k)\|X\|_F^2 \leq \|\mathscr{M}(X)\|_2^2 \leq (1+\delta_k)\|X\|_F^2.$$

Definition 3.2.3 (Restriction [30])**.** Let $\mathscr{M} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$ be a linear operator and assume without loss of generality that $n_1 \leq n_2$. Given a pair of matrices U and V with orthonormal columns, define $\mathscr{M}_{U,V}$, the (U,V) restriction of $\mathscr{M}$ by [1]

$$\begin{aligned} \mathscr{M}_{U,V} : \mathbb{R}^{n_1} &\to \mathbb{R}^m \\ x &\mapsto \mathscr{M}(U\,\mathrm{diag}(x)V^*). \end{aligned}$$

3.2.2 Preliminaries on ΣΔ quantization

ΣΔ quantizers were first proposed in the context of digitizing oversampled band-limited functions by [20], and their mathematical properties have been studied since. In this band-limited context, the ΣΔ quantizer takes in a sequence of point evaluations of the function sampled at a rate exceeding the critical Nyquist rate and produces a sequence of *quantized* elements, i.e., elements from a finite set. So, the ΣΔ quantizer is associated with this finite set, say $\mathscr{A} \subset \mathbb{R}$ (called the

[1] Here, given a vector $x \in \mathbb{R}^n$, $\mathrm{diag}(x) = X$ is a diagonal matrix in $\mathbb{R}^{n\times n}$ with $X_{ii} = x_i$ for $i \in \{1,...,n\}$.

quantization alphabet), and also with a *scalar quantizer*

$$\begin{aligned} Q_{\mathscr{A}} &: \mathbb{R} \to \mathscr{A} \\ z &\mapsto \arg\min_{v \in \mathscr{A}} |v - z|. \end{aligned} \tag{3.2.1}$$

ΣΔ schemes build on scalar quantization by incorporating a state variable sequence u, which is recursively updated. In an rth order ΣΔ scheme, a function, say ρ_r, of r previous values of u and the current measurement are fed into the scalar quantizer to produce an element from $\mathscr{A}$. For example, in the band-limited context the measurements are simply the pointwise evaluations of the function. Defining $u_i = 0$ for $i \leq 0$, and denoting the measurements by y_i we have the recursion:

$$q_i = Q_{\mathscr{A}}(\rho_r(u_{i-1}, ...u_{i-r+1}, y_i)) \tag{3.2.2}$$

$$(\Delta^r u)_i = y_i - q_i. \tag{3.2.3}$$

Here $(\Delta u)_i := u_i - u_{i-1}$, and $\Delta^r u := \Delta(\Delta^{r-1} u)$. Thus, the rth order ΣΔ quantizer updates the state variables as a solution to an rth order difference equation. To give a concrete example, the simplest 1st order ΣΔ scheme operates by running the following recursion:

$$q_i = Q_{\mathscr{A}}(y_i + u_{i-1}) \tag{3.2.4}$$

$$u_i = u_{i-1} + y_i - q_i. \tag{3.2.5}$$

Usually, the alphabet $\mathscr{A}$ associated with ΣΔ quantizers is of the form

$$\mathscr{A} := \{\pm(j - 1/2)\beta, j = 1, ..., L\}.$$

We refer to such an $\mathscr{A}$ as a $2L$-level alphabet with step-size β. In particular, when $L = 1$, we have a 1-bit alphabet.

For reasons related to building a circuit that implements the $\Sigma\Delta$ quantization scheme and bounding the reconstruction error, an important consideration is the so-called stability of the $\Sigma\Delta$ scheme. A stable rth order $\Sigma\Delta$ scheme produces *bounded* state variables with

$$\|u\|_\infty < \gamma(r), \tag{3.2.6}$$

whenever $\|y\|_\infty$ is bounded above. Above, $\gamma(r)$ is some constant which may depend on r. For example, for the $2L$-level alphabet described above, coupled with a particular choice of ρ_r and $\Sigma\Delta$ order r it is sufficient to choose $L \geq 2\lceil \frac{\|y\|_\infty}{\beta} \rceil + 2^r + 1$ to guarantee (3.2.6) holds with $\gamma(r) = \beta/2$ [1, 10]. Note that with such a choice the size of the alphabet grows exponentially as a function of the $\Sigma\Delta$ order. On the other hand, given a fixed alphabet, [10] constructed the first family of functions ρ_r with associated stability constants $\gamma(r)$. Subsequently, the dependence on r was improved upon by [19] and [12] via different constructions of ρ_r. In these papers it was shown that $\Sigma\Delta$ quantized measurements of a band-limited function f, sampled at a rate λ times the critical Nyquist rate, can be used to obtain an approximation $\hat{f}$ of f satisfying

$$\|\hat{f} - f\|_\infty \leq C\gamma(r)\lambda^{-r}.$$

By optimizing the right hand side above, i.e., $\gamma(r)\lambda^{-r}$ as a function of r, [19] and [12] obtain the error rates

$$\|\hat{f} - f\|_\infty \leq Ce^{-c\lambda}$$

where $c < 1$ is a known constant depending on the family of schemes.

Outside of the band-limited context, $\Sigma\Delta$ schemes were proposed and studied for quantizing finite-frame coefficients [21, 1, 25, 26] as well as compressed sensing coefficients [18, 15, 3, 34].

In both these contexts, given a linear map $\Phi : \mathbb{R}^n \mapsto \mathbb{R}^m$, absent noise, one obtains measurements

$$y = \Phi x$$

of a vector $x \in \mathscr{X} \subset \mathbb{R}^n$ and quantizes using an rth order stable $\Sigma\Delta$ scheme. To ensure boundedness of the resulting state variable, typically one has $\mathscr{X} \subset \{x \in \mathbb{R}^n : \|\Phi x\|_\infty < 1\}$. One may also enforce additional restrictions on elements of $\mathscr{X}$, such as k-sparsity. Here, as before, one runs a stable rth order $\Sigma\Delta$ quantization scheme

$$Q_{\Sigma\Delta}^{(r)} : \mathbb{R}^m \to \mathscr{A}^m. \tag{3.2.7}$$

Writing the $\Sigma\Delta$ state equations (3.2.2) in matrix-vector form yields

$$y - q = D^r u \tag{3.2.8}$$

where $D \in \mathbb{R}^{m\times m}$ is the lower bi-diagonal difference matrix with 1 on the main diagonal and -1 on the sub-diagonal. In analogy with the band-limited case, here one defines the oversampling factor as the ratio of the number of measurements m to the minimal number m_0 needed to ensure that Φ is injective (or stably invertible) on $\mathscr{X}$. For example $\lambda := \frac{m}{n}$ in the finite-frames setting when $\mathscr{X}$ is the Euclidean ball, and $\lambda := \frac{m}{k \log n/k}$ in the compressed sensing context when $\mathscr{X}$ is the intersection of the Euclidean ball with the set of k-sparse vectors in $\mathbb{R}^n$. As in the band-limited context, one wishes to bound the reconstruction error as a function of λ. A typical result states that provided Φ satisfies certain assumptions, there exists a reconstruction map

$$\mathscr{D} : \mathscr{A}^m \to \mathbb{R}^n \tag{3.2.9}$$

such that for all $x \in \mathscr{X}$ and $\hat{x} := \mathscr{D}(Q_{\Sigma\Delta}^{(r)}(\Phi x))$,

$$\|x - \hat{x}\|_2 \leq C\lambda^{-\alpha(r-1/2)}$$

where $\alpha \leq 1$ is a parameter that, in the case of random measurements, controls the probability with which the result holds. Most relevant to this work [34] proposes recovering *arbitrary*, that is, not necessarily strictly sparse, vectors in $\mathbb{R}^n$ from their *noisy* $\Sigma\Delta$-quantized compressed sensing measurements by solving a convex optimization problem. In particular, one obtains the approximation $\hat{x}$ from $q := Q_{\Sigma\Delta}^{(r)}(\Phi x + e)$, where $\|e\|_\infty \leq \varepsilon$ via

$$\begin{aligned}(\hat{x}, \hat{\nu}) := \arg\min_{(z,\nu)} \|z\|_1 \ \text{ subject to } \|D^{-r}(\Phi z + \nu - q)\|_2 \leq \gamma(r)\sqrt{m} \\ \text{and } \ \|\nu\|_2 \leq \varepsilon\sqrt{m}. \end{aligned} \tag{3.2.10}$$

Then, [34] shows that the reconstruction error due to quantization decays polynomially in the number of measurements, while maintaining stability and robustness against noise in the measurements and deviations from sparsity. Specifically, defining

$$\sigma_k(x) := \arg\min_{v \in \Sigma_k^n} \|x - v\|_1,$$

the following theorem holds.

Theorem 3.2.4. *[34] Let k, ℓ, m, n be integers, and let $P_\ell : \mathbb{R}^m \to \mathbb{R}^\ell$ be the projection onto the first ℓ coordinates. Let $D^{-r} = U\Sigma V^*$ be the singular value decomposition of D^{-r} and let Φ be an $m \times n$ matrix such that $\frac{1}{\sqrt{\ell}} P_\ell V^* \Phi$ has the vector-RIP of order k and constant $\delta_k < 1/9$. Then, for all $x \in \mathbb{R}^n$ satisfying $\|\Phi x\|_\infty \leq \mu < 1$ and all e, $\|e\|_\infty \leq \varepsilon < 1 - \mu$, the solution $\hat{x}$ of* (3.2.10) *with*

$q = Q_{\Sigma\Delta}^{(r)}(\Phi x + e)$ *satisfies*

$$\|\hat{x} - x\|_2 \leq C \left(\left(\frac{m}{\ell}\right)^{-r+1/2} \beta + \frac{\sigma_k(x)}{\sqrt{k}} + \sqrt{\frac{m}{\ell}} \varepsilon \right). \tag{3.2.11}$$

Above C does not depend on m, ℓ, n.

The proof of Theorem 3.2.4 reveals that a more general statement is true. Indeed, it turns out that the only assumptions on $\hat{x}$ needed are that it satisfies the constraints in (3.2.10) and that $\|\hat{x}\|_1 \leq \|x\|_1$. Moreover, the only assumption needed on q is that it satisfies the state variable equations (3.2.8), and need not belong to $\mathscr{A}^m$. We will use this generalization in proving our main result, and we state it below for convenience.

Theorem 3.2.5. *Let* $k, \ell, m, n, P_\ell, V^*, \Phi$ *be as above. The following is true for all* $x \in \mathbb{R}^n$ *and* $e \in \mathbb{R}^m$ *with* $\|\Phi x\|_\infty \leq \mu < 1$ *and* $\|e\|_\infty \leq \varepsilon < 1 - \mu$ *:*
Suppose q *is any vector which satisfies the relation* $\Phi x + e - D^r u = q$ *and* $\|u\|_\infty \leq \gamma(r) < \infty$. *Suppose further that* $\hat{x} \in \mathbb{R}^n$ *is feasible to* (3.2.10) *and satisfies* $\|\hat{x}\|_1 \leq \|x\|_1$. *Then*

$$\|\hat{x} - x\|_2 \leq C \left(\left(\frac{m}{\ell}\right)^{-r+1/2} \beta + \frac{\sigma_k(x)}{\sqrt{k}} + \sqrt{\frac{m}{\ell}} \varepsilon \right), \tag{3.2.12}$$

where C does not depend on m, ℓ, n.

3.2.3 Preliminaries on low-rank recovery

A key idea in our proof is relating low-rank matrix recovery to sparse vector recovery, as was first done in [30], where the following useful lemmas were presented.

Lemma 3.2.6 ([30])**.** *If* $\mathscr{M}$ *satisfies the matrix-RIP of order k and constant* δ_k*, then for all unitary* U, V, $\mathscr{M}_{U,V}$ *satisfies the vector-RIP of order k and constant* δ_k.

Lemma 3.2.7 ([30])**.** *Suppose* $W \in \mathbb{R}^{n_1 \times n_2}$ *admits the singular value decomposition* $U_W \Sigma_W V_W^*$, *and suppose* $X_0 \in \mathbb{R}^{n_1 \times n_2}$ *admits the singular value decomposition* $U_{X_0} \Sigma_{X_0} {V_{X_0}}^*$. *Suppose that*

$\|X_0 + W\|_* \leq \|X_0\|_*$ *and assume without loss of generality that* $n_1 \leq n_2$. *Then, there exists* $X_1 = U_W \operatorname{diag}(z) V_W^*$ *for some* $z \in \mathbb{R}^{n_1}$ *such that* $\|X_1 + W\|_* \leq \|X_1\|_*$. *In particular, the choice* $X_1 := -U_W \Sigma_{X_0} V_W^*$ *yields the inequality.*

3.2.4 Preliminaries on Probabalistic Tools

Many of the classical compressed sensing results involve sampling a sparse signal with a Gaussian linear operator. It has been noticed, however, that only a handful of the special features of the Gaussian distribution are needed for these results to hold. Examples of such features include super-exponential tail decay, the existence of a moment generating function, and moments which grow "slowly", see [37, 16] for example. A class of distributions which enjoy these features is the sub-Gaussian class, which we define below.

Definition 3.2.8. Let X be a real-valued random variable. We say X is a sub-Gaussian random variable with parameter K if for all $t \geq 0$

$$\mathbb{P}[|X| > t] \leq \exp(1 - t^2/K).$$

We say that a linear operator $\mathscr{M}$ is sub-Gaussian if its associated matrix $A_{\mathscr{M}}$ has entries drawn independently and identically from a sub-Gaussian distribution.

The tail decay property in Definition 3.2.8 is equivalent to the j-th root of the j-th moment of a sub-Gaussian random variable X growing like $\sqrt{j}$, or when $\mathbb{E}X = 0$ equivalent to the moment generating function existing over all of $\mathbb{R}$. See [37, 16] for the details.

In the course of proving our main result, we will need to show a certain sub-Gaussian linear operator satisfies the matrix-RIP. Our proof of such will require a technique known as chaining. Talagrand makes the following definition in [36].

Definition 3.2.9. Given a metric space (T, d), an admissible sequence of T is a collection of subsets of T, $\{T_s : s \geq 0\}$, such that for all $s \geq 0$, $|T_s| \leq 2^{2^s}$, and $|T_0| = 1$. The γ_2 functional is

defined by

$$\gamma_2(T,d) = \inf \sup_{t\in T} \sum_{s=0}^{\infty} 2^{s/2} d(t,T_s)$$

where the infimum is taken with respect to all admissible sequences of T.

It is common, given the unwieldy definition above, to control the γ_2 functional with the well-known Dudley integral [13]. In our case, we will consider a set of matrices $\mathscr{S} \subset \mathbb{R}^{m\times n_1 n_2}$ equipped with the operator norm $\|A\|_{2\to 2} = \sup_{\|x\|_2=1} \|Ax\|_2$. With this, we have for some universal constant $c > 0$

$$\gamma_2(\mathscr{S}, \|\cdot\|_{2\to 2}) \leq c \int_0^{d_{2\to 2}(\mathscr{S})} \sqrt{\log(N(\mathscr{S}, \|\cdot\|_{2\to 2}; u))}\, du$$

where $d_{2\to 2}(\mathscr{S}) = \sup_{A\in\mathscr{S}} \|A\|_{2\to 2}$. is the operator norm radius of the set $\mathscr{S}$ and $N(\mathscr{S}, \|\cdot\|_{2\to 2}; u)$ is the covering number of $\mathscr{S}$ with radius u.

The following useful lemma from [24] will allow us to easily control the matrix-RIP of the linear operators $P_\ell V^* \mathscr{M}$ where $\mathscr{M}$ is sub-Gaussian.

Lemma 3.2.10 ([24])**.** *Let $\mathscr{S}$ be a set of matrices, and let ξ be a sub-Gaussian random vector with independent and identically distributed (i.i.d.) mean zero, unit variance entries with parameter K. Set*

$$\begin{aligned}
\mu &= \gamma_2(\mathscr{S}, \|\cdot\|_{2\to 2})\Big(\gamma_2(\mathscr{S}, \|\cdot\|_{2\to 2}) + d_F(\mathscr{S})\Big) \\
\nu_1 &= d_{2\to 2}(\mathscr{S})\Big(\gamma_2(\mathscr{S}, \|\cdot\|_{2\to 2}) + d_F(\mathscr{S})\Big) \\
\nu_2 &= d^2_{2\to 2}(\mathscr{S}).
\end{aligned}$$

where $d_F(\mathscr{S}) = \sup_{A\in\mathscr{S}} \|A\|_F$ is the radius of the set $\mathscr{S}$ with respect to the Frobenius norm.

Then for all $t > 0$,

$$\mathbb{P}\left[\sup_{A\in\mathscr{S}}\left|\|A\xi\|_2^2 - \mathbb{E}\|A\xi\|_2^2\right| \geq c_1\mu + t\right] \leq 2\exp\left(-c_2\min\left\{\frac{t^2}{\nu_1^2}, \frac{t}{\nu_2}\right\}\right)$$

where the constants $c_1, c_2 > 0$ depend only on K.

Lemma 3.2.11. *Let $\mathscr{M} : \mathbb{R}^{n_1\times n_2} \to \mathbb{R}^m$ be a mean zero, unit variance sub-Gaussian linear map with parameter K, $P_\ell : \mathbb{R}^m \to \mathbb{R}^\ell$ the projection map onto the first ℓ coordinates, and $V^* \in \mathbb{R}^{m\times m}$ a unitary matrix. Then there exist constants C_1, C_2 which may depend on K, such that for $\ell \geq C_1\frac{k(n_1+n_2+1)}{\delta_k^2}$, the operator $\frac{1}{\sqrt{\ell}}P_\ell V^*\mathscr{M}$ has the matrix-RIP with constant δ_k with probability exceeding $1-2e^{-C_2\ell}$.*

Proof. The proof will be an application of Lemma 3.2.10. To that end, observe that

$$\frac{1}{\sqrt{\ell}}P_\ell V^*\mathscr{M}(X) = \frac{1}{\sqrt{\ell}}P_\ell V^*\mathrm{diag}(\vec{X}^T)\xi$$

where $\mathrm{diag}(\vec{X}^T) = I_{m\times m}\otimes(\vec{X})^T$, and $\xi \in \mathbb{R}^{mn_1n_2}$ is a sub-Gaussian random vector [2]. Without loss of generality, we may assume that $\|X\|_F = 1$ by rescaling, if necessary. It behooves us then to consider

$$\mathscr{S} = \left\{\frac{1}{\sqrt{\ell}}P_\ell V^*\mathrm{diag}(\vec{X}^T) : X \in \mathbb{R}^{n_1\times n_2}, \|X\|_F = 1, \mathrm{rank}(X) \leq k\right\}.$$

Let $V_{i,\cdot}$ denote the i-th row of V. By direct calculation, we see that

$$d_F^2(\mathscr{S}) = \sup_X\frac{1}{\ell}\|P_\ell V^*\mathrm{diag}(\vec{X}^T)\|_F^2 = \sup_X\frac{1}{\ell}\|X\|_F^2\sum_{i=1}^m\sum_{j=1}^\ell|V_{i,j}|^2 \leq 1.$$

[2] Here, $\otimes$ refers to the Kronecker product of matrices.

Likewise, by direct calculation,

$$d_{2\to 2}^2(\mathscr{S}) = \sup_X \frac{1}{\ell}\|P_\ell V^* \mathrm{diag}(\vec{X}^T)\|_{2\to 2}^2 = \sup_X \sup_{\|w\|_2=1} \frac{1}{\ell}\|P_\ell V^* \mathrm{diag}(\vec{X}^T)w\|_2^2$$
$$\leq \frac{1}{\ell}\sup_X \|X\|_F^2 = \frac{1}{\ell}.$$

Above, the last inequality follows from the Cauchy-Schwarz inequality, or alternatively from the fact that the largest singular value of $I \otimes \vec{X}^T$ is just the singular value of the vector $\vec{X}^T$, namely its Frobenius norm. Lemma 3.1 in [7] tells us the covering number $N(\mathscr{S}, \|\cdot\|_F; \varepsilon) \leq \left(\frac{9}{\varepsilon}\right)^{(n_1+n_2+1)k}$. Invoking Dudley's inequality and Hölder's inequality, and letting $\mathbb{1}_B$ denote the indicator function on the set B, that is, $\mathbb{1}_B(x) = 1$ for $x \in B$ and 0 otherwise, we get

$$\begin{aligned}
\gamma_2(\mathscr{S}, \|\cdot\|_{2\to 2}) &\leq c_1\sqrt{k(n_1+n_2+1)}\int_0^{\frac{1}{\sqrt{\ell}}} \sqrt{\log\left(\frac{9}{\varepsilon}\right)}\, d\varepsilon \\
&\leq c_1\sqrt{k(n_1+n_2+1)}\left\|\mathbb{1}_{[0,\frac{1}{\sqrt{\ell}}]}\right\|_{L^2([0,1])}\left\|\sqrt{\log\left(\frac{9}{\varepsilon}\right)}\right\|_{L^2([0,1])} \\
&= c_1\sqrt{\frac{k(n_1+n_2+1)}{\ell}}(\log(9)+1)^{1/2} := c_2\sqrt{\frac{k(n_1+n_2+1)}{\ell}}.
\end{aligned}$$

Putting it all together, we have

$$\begin{aligned}
\mu &\leq c_2^2\frac{k(n_1+n_2+1)}{\ell} + c_2\sqrt{\frac{k(n_1+n_2+1)}{\ell}} \leq c_3\sqrt{\frac{k(n_1+n_2+1)}{\ell}} \\
\nu_1 &\leq \frac{1}{\sqrt{\ell}} + c_2\frac{\sqrt{k(n_1+n_2+1)}}{\ell} \leq c_4\frac{1}{\sqrt{\ell}} \\
\nu_2 &= \frac{1}{\ell}.
\end{aligned}$$

So invoking Lemma 3.2.10 yields for all $t > 0$,

$$\mathbb{P}\left[\sup_X \left|\frac{1}{\ell}\|P_\ell V^* \mathscr{M}(X)\|_F^2 - \mathbb{E}\frac{1}{\ell}\|P_\ell V^* \mathscr{M}(X)\|_F^2\right| \geq c_5\mu + t\right] \leq 2\exp\left(-c_6 \min\left\{\frac{t^2}{\nu_1^2}, \frac{t}{\nu_2}\right\}\right), \tag{3.2.13}$$

where the supremum is taken over all $X \in \mathbb{R}^{n_1 \times n_2}$, $\|X\|_F = 1, \text{rank}(X) \leq k$. Note that by independence of the A_j, we have

$$\begin{aligned}
\mathbb{E}\frac{1}{\ell}\|P_\ell V^* \mathscr{M}(X)\|_F^2 &= \frac{1}{\ell}\mathbb{E}\sum_{i=1}^{\ell}\left(\sum_{j=1}^{m} V_{i,j}\langle A_j, X\rangle\right)^2 \\
&= \frac{1}{\ell}\sum_{i=1}^{\ell}\sum_{j=1}^{m} V_{i,j}^2 \mathbb{E}\langle A_j, X\rangle^2 \\
&= \frac{1}{\ell}\|X\|_F^2 \sum_{i=1}^{\ell}\sum_{j=1}^{m} V_{i,j}^2 = \|X\|_F^2.
\end{aligned}$$

Equation (3.2.13) now becomes

$$\mathbb{P}\left[\sup_X \left|\frac{1}{\ell}\|P_\ell V^* \mathscr{M}(X)\|_F^2 - \|X\|_F^2\right| \geq c_5\sqrt{\frac{k(n_1+n_2+1)}{\ell}} + t\right] \leq 2\exp\left(-c_6 \min\left\{c_4^{-2}t^2\ell, t\ell\right\}\right). \tag{3.2.14}$$

Choosing $t = \delta_k/2$ and recalling that $\ell \geq C_1 \frac{k(n_1+n_2+1)}{\delta_k^2}$ with $C_1 := 4c_5^2$, equation (3.2.14) reduces to

$$\mathbb{P}\left[\sup_X \left|\frac{1}{\ell}\|P_\ell V^* \mathscr{M}(X)\|_F^2 - \|X\|_F^2\right| \geq \delta_k\right] \leq 2\exp\left(-C_2\ell\right),$$

where $C_2 := c_6 \min\{c_4^{-2}\delta_k^2/4, \delta_k/2\}$. Therefore, with probability $1 - 2\exp(-C_2\ell)$, we have that

$$\left|\frac{1}{\ell}\|P_\ell V^* \mathscr{M}(X)\|_F^2 - \|X\|_F^2\right| \leq \delta_k = \delta_k\|X\|_F^2$$

for all $X \in \mathbb{R}^{n_1 \times n_2}$, $\|X\|_F = 1, \text{rank}(X) \leq k$. In other words, $\frac{1}{\ell} P_\ell V^* \mathscr{M}(X)$ has the matrix RIP with constant δ_k with high probability. $\square$

3.3 Recovery error guarantees

Herein, we present our main result on the recovery error guarantees for $\Sigma\Delta$-quantized sub-Gaussian measurements of approximately low-rank matrices. Specifically, our results pertain to reconstruction via the constrained nuclear-norm minimization

$$(\hat{X}, \hat{\nu}) := \arg\min_{(Z,\nu)} \|Z\|_* \text{ subject to } \|D^{-r}(\mathscr{M}(Z) + \nu - q)\|_2 \leq \gamma(r)\sqrt{m}$$
$$\text{and } \|\nu\|_2 \leq \varepsilon\sqrt{m} \tag{3.3.1}$$

where $\gamma(r)$ is the stability constant associated with the quantizer. As such, Theorem 4.4.1 is a generalization of Theorem 3.2.4 to the low-rank matrix case.

Theorem 3.3.1 (Error guarantees for stable $\Sigma\Delta$ quantizers)**.** *Let k, ℓ, and r be integers and let $\mathscr{M} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$ be a mean zero, unit variance sub-Gaussian linear operator with parameter K. Suppose that $m \geq \ell \geq c_1 k \max\{n_1, n_2\}$. Then, with probability exceeding $1 - c_2 e^{-c_3 \ell}$ on the draw of $\mathscr{M}$, the following holds for a stable $\Sigma\Delta$ quantizer with stability constant $\gamma(r)$:*

For all $X \in \mathbb{R}^{n_1 \times n_2}$, the solution $X^\sharp$ of (3.3.1) *where q is the $\Sigma\Delta$ quantization of $\mathscr{M}(X) + e$ with $\|e\|_\infty \leq \varepsilon$, satisfies*

$$\|X^\sharp - X\|_F \leq C(r) \left(\left(\frac{m}{\ell} \right)^{-r+1/2} \beta + \frac{\sigma_k(X)_*}{\sqrt{k}} + \sqrt{\frac{m}{\ell}} \varepsilon \right). \tag{3.3.2}$$

The constants c_1, c_2, c_3, C do not depend on the dimensions, but may depend on K and r.

Proof. Recall that by the ΣΔ state equations, we have

$$\|u\|_\infty = \|D^{-r}(\mathscr{M}(X) + e - q)\|_\infty \leq \gamma(r)\beta.$$

Consequently, by feasibility and optimality of $(X^\sharp, \nu^\sharp)$ respectively, we have $\|D^{-r}(\mathscr{M}(X^\sharp) + \nu^\sharp - q)\|_2 \leq \gamma(r)\beta\sqrt{m}$ and $\|X^\sharp\|_* \leq \|X\|_*$.

Define $W := X^\sharp - X$ and let $U_W \Sigma_W V_W^*$ be the singular value decomposition of W. Then, denoting by $U_X \Sigma_X V_X^*$ the singular value decomposition of X, we have by Lemma 3.2.7, with $X_1 = -U_W \Sigma_X V_W^*$ that

$$\|X_1 + W\|_* \leq \|X_1\|_*.$$

Moreover, defining

$$y_1 := D^{-r}(\mathscr{M}(X_1) + e) + u,$$

we have by the linearity of $\mathscr{M}$

$$\begin{aligned}
\|D^{-r}(\mathscr{M}(X_1 + W) + \nu^\sharp) - y_1\|_2 &= \|D^{-r}(\mathscr{M}(X_1 + W) + \nu^\sharp) - (D^{-r}(\mathscr{M}(X_1) + e) + u)\|_2 \\
&= \|D^{-r}(\mathscr{M}(W) + \nu^\sharp - e) - u\|_2 \\
&= \|D^{-r}(\mathscr{M}(X + W) + \nu^\sharp) - (D^{-r}(\mathscr{M}(X) + e) + u)\|_2 \\
&= \|D^{-r}(\mathscr{M}(X^\sharp) + \nu^\sharp - q)\|_2 \\
&\leq \gamma(r)\beta\sqrt{m}. \qquad (3.3.3)
\end{aligned}$$

Now, note that with x_1 denoting the vector composed of the diagonal entries of $-\Sigma_X$, we have

$$\begin{aligned}
y_1 &= D^{-r}(\mathscr{M}(U_W \operatorname{diag}(x_1) V_W^*) + e) + u \qquad (3.3.4) \\
&= D^{-r}(\mathscr{M}_{U_W, V_W} x_1 + e) + u \qquad (3.3.5) \\
&= (D^{-r}\mathscr{M})_{U_W, V_W} x_1 + D^{-r} e + u. \qquad (3.3.6)
\end{aligned}$$

Above, we defined $(D^{-r}\mathscr{M})(X) := \sum_{i=1}^{m} \langle X, A_i \rangle D^{-r} e_i$. Denoting by w the vector composed of the diagonal entries of Σ_W, (3.3.6) and (3.3.3) respectively yield the inequalities

$$\|(D^{-r}\mathscr{M})_{U_W,V_W} x_1 + D^{-r} e - y_1\|_2 \leq \gamma(r)\beta\sqrt{m}. \tag{3.3.7}$$

and

$$\|(D^{-r}\mathscr{M})_{U_W,V_W}(x_1 + w) + D^{-r} v^\sharp - y_1\|_2 \leq \gamma(r)\beta\sqrt{m}. \tag{3.3.8}$$

Additionally, we have that

$$\|x_1 + w\|_1 = \|X_1 + W\|_* = \|X^\sharp\|_* \leq \|X\|_* = \|x_1\|_1.$$

Thus, we have shown that the vector $x^\sharp := x_1 + w$ has a smaller ℓ_1 norm than x, and that it is feasible to (3.2.10) with $\mathscr{M}_{U_W,V_W}$ in place of Φ and y_1 in place of $D^{-r}q$. So, we are *almost* ready to apply Theorem 3.2.5 to $\mathscr{M}_{U_W,V_W}$ and conclude that

$$\|X^\sharp - X\|_F = \|W\|_F = \|w\|_2 \leq C(r)\left(\left(\frac{m}{\ell}\right)^{-r+1/2}\beta + \frac{\sigma_k(X)_*}{\sqrt{k}} + \sqrt{\frac{m}{\ell}}\varepsilon\right). \tag{3.3.9}$$

However, to do that, we must first show that $\frac{1}{\sqrt{\ell}}(P_\ell V^* \mathscr{M})_{(U_W,V_W)}$ has the vector-RIP of order k and constant $\beta < 1/9$. This, however, follows from Lemma 3.2.11 where it is established that $(\frac{1}{\sqrt{\ell}} P_\ell V^* \mathscr{M})$ has the required matrix-RIP, with high probability. By Lemma 3.2.6, this implies that $(\frac{1}{\sqrt{\ell}} P_\ell V^* \mathscr{M})_{U_W,V_W}$ has the vector-RIP of order k for all unitary pairs (U_W, V_W), so now we may apply Theorem 3.2.5 to obtain (3.3.9) and conclude the proof.

□

By finding the optimal quantization order r as a function of the oversampling factor, as is standard in the ΣΔ literature (e.g., [19], [25]), root-exponential error decay can attained. Corollary 3.3.2 is a precise statement to that effect. Its proof follows the same argument as Corollary 11

in [34], with only the oversampling factor λ changed to reflect the fact that we are dealing with matrices instead of vectors. Next, we show that the component of the reconstruction error that is due to quantization can be made to decay root-exponentially as a function of the oversampling factor.

Corollary 3.3.2 (Root-exponential quantization error decay)**.** *Let $\mathscr{M} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$ be a mean zero, unit variance sub-Gaussian linear operator with parameter K and $X \in \mathbb{R}^{n_1 \times n_2}$ a rank k matrix with $\|\mathscr{M}(X)\|_F \leq 1$. Denote by $Q^r_{\Sigma\Delta}$ the r^{th} order $\Sigma\Delta$ quantizer with alphabet $\mathscr{A}$ of step-size β and stability constant $\gamma(r) \leq C^r r^r \beta$. Then there exist constants $c, c_1, C_1, C_2 > 0$ that may depend on K, so that when*

$$\begin{aligned} \lambda &:= \frac{m}{\lceil ck \max(n_1, n_2) \rceil} \\ r &:= \left\lfloor \frac{\lambda}{2C_1 e} \right\rfloor^{1/2} \\ q &:= Q^r_{\Sigma\Delta}(\mathscr{M}(X)). \end{aligned}$$

the solution $\hat{X}$ to (3.3.1) *satisfies $\|\hat{X} - X\|_F \leq C_2 \beta e^{-c_1\sqrt{\lambda}}$.*

Next, Corollary 3.3.3 shows that by projecting the quantized measurements onto a subspace of dimension $L = Ck \max(n_1, n_2) \leq C'm$, where $C, C' > 0$ are absolute constants, we can obtain comparable reconstruction error guarantees to those of Theorem 4.4.1. In turn, this allows us to obtain a reconstruction with *exponentially* decaying quantization error, or distortion, as a function of the number of bits, or rate, used. We make this observation precise in Remark 5, thereby extending the analogous result for the vector case [35] to our matrix setting. We comment that, just like for sparse vectors, this exponentially decaying rate-distortion relationship is optimal for low-rank matrices over all possible encoding and decoding schemes.

Corollary 3.3.3 (Error guarantees with encoding)**.** *Let $\mathscr{M} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$ be a mean zero, unit variance sub-Gaussian linear operator with parameter K. Let $B : \mathbb{R}^m \to \mathbb{R}^L$ be a Bernoulli*

random matrix whose entries are ± 1. *Then there exist constants* $c_1, c_2, c_3, C_1, C_2 > 0$ *that may depend on* K *and* r, *so that whenever* $m \geq c_1 L \geq c_2 k \max(n_1, n_2)$ *the following is true with probability greater than* $1 - C_1 \exp(-c_3\sqrt{mL})$ *on the draw of* $\mathscr{M}$ *and* B:

Suppose $X \in \mathbb{R}^{n_1 \times n_2}$ *has rank* k, $\|\mathscr{M}(X)\|_\infty \leq \mu < 1$, *and* $q = Q^r_{\Sigma\Delta}(\mathscr{M}(X) + e)$ *with* $\|e\|_\infty \leq \varepsilon$ *for* $\varepsilon \in [0, 1-\mu)$. *Then the solution of*

$$(\hat{X}, \hat{\nu}) := \arg\min_{(Z,\nu)} \|Z\|_* \ \text{ subject to } \|BD^{-r}(\mathscr{M}(Z) + \nu - q)\|_2 \leq 3m\gamma(r)$$
$$\text{and } \ \|\nu\|_2 \leq \varepsilon\sqrt{m}. \tag{3.3.10}$$

satisfies

$$\|\hat{X} - X\|_F \leq C_2 \left(\left(\frac{m}{L}\right)^{-r/2+3/4} \beta + \sqrt{\frac{m}{L}}\varepsilon + \frac{\sigma_k(X)_*}{\sqrt{k}} \right).$$

Remark 5. Let $\alpha := \max_{a \in \mathscr{A}} \|a\|_\infty$. A simple calculation shows that one needs a rate of at most $\mathscr{R} = Lr \log_2(\alpha m)$ bits to store the encoded measurements. This demonstrates that in the noise free setting, and with rank k matrices, the distortion $\mathscr{D} := \|\hat{X} - X\|_F$ satisfies

$$\mathscr{D} \leq \left(\frac{1}{\alpha L} 2^{\frac{\mathscr{R}}{Lr}} \right)^{-r/2+3/4}. \tag{3.3.11}$$

That is, the distortion decays exponentially with respect to the rate provided $r \geq 2$.

The proof of the above corollary follows from a combination of Theorem 12 in [35], which we state below, and an argument similar to the proof of Theorem 4.4.1.

Theorem 3.3.4 ([35])**.** *Let* Φ *be a* $m \times n$ *sub-Gaussian matrix with mean zero and unit variance entries with parameter* K, *and let* B *be a* $L \times m$ *Bernoulli matrix with* ± 1 *entries. Moreover, let* $k \in \{1, .., \min\{m, n\}\}$.

Denote by $Q^r_{\Sigma\Delta}$ a stable rth-order scheme with $r > 1$, alphabet $\mathscr{A}$ and stability constant $\gamma(r)$. There exist positive constants C_1, C_2, C_3, C_4 and c_1 such that whenever $\frac{m}{C_2} \geq L \geq C_1 k \log(n/k)$ the following holds with probability greater than $1 - C_3 e^{-c_1\sqrt{mL}}$ on the draws of Φ and B:

Suppose that $x \in \mathbb{R}^n$, $e \in \mathbb{R}^m$ with $\|\Phi x\|_\infty \leq \mu < 1$ and that $q := Q^r_{\Sigma\Delta}(\Phi x + e)$ where $\|e\|_\infty \leq \varepsilon$ for some $0 \leq \varepsilon < 1 - \mu$. Then the solution $\hat{x}$ to

$$(\hat{x}, \hat{\nu}) := \arg\min_{(z,\nu)} \|z\|_1 \;\; \textit{subject to } \|BD^{-r}(\Phi(z) + \nu - q)\|_2 \leq 3m\gamma(r) \quad \textit{and} \;\; \|\nu\|_2 \leq \varepsilon\sqrt{m}. \tag{3.3.12}$$

satisfies

$$\|\hat{x} - x\|_2 \leq C_4 \left(\left(\frac{m}{L}\right)^{-r/2+3/4} \beta + \sqrt{\frac{m}{L}}\varepsilon + \frac{\sigma_k(x)}{\sqrt{k}} \right).$$

Remark 6. As before, the requirements on q can be relaxed so that it is any vector satisfying the relation $\Phi x + \omega + D^r u = q$ with $\|\Phi x\|_\infty \leq \mu < 1$, $\|\omega\|_\infty \leq \varepsilon < 1 - \mu$, and $\|u\|_\infty \leq \gamma(r) < \infty$.

Remark 7. If one restricts the scope of Theorem 3.3.4 to strictly sparse vectors, the constraint in (3.3.12) can be relaxed to $\|BD^{-r}(\Phi(z) + \nu - q)\|_2 \leq 3\sqrt{mL}\gamma(r)$ through a Johnson-Lindenstrauss embedding argument. See the proof of Theorem 16 in [35] for details.

Proof of Corollary 3.3.3. We know (see for example [35]) that with probability exceeding $1 - c_2 \exp(-c_1\sqrt{mL})$ that $\|B\|_{2\to 2} \leq \sqrt{L} + 2\sqrt{m}$. For such B, we have

$$\|Bu\|_2 \leq \|B\|_{2\to 2}\|u\|_\infty \leq 3m\gamma(r).$$

Define, as before, the following:

$$W = X^\sharp - X = U_W \Sigma_W V_W^*$$
$$X = U_X \Sigma_X V_X^*$$
$$X_1 = -U_W \Sigma_X V_W^*.$$

By Lemma 3.2.7, $\|X_1 + W\|_* \leq \|X_1\|_*$. Now, define

$$y_1 = BD^{-r}\Big(\mathscr{M}(X_1) + e\Big) + Bu.$$

By linearity of $\mathscr{M}$,

$$\begin{aligned}\left\|BD^{-r}\Big(\mathscr{M}(X_1+W)+\nu^\sharp\Big) - y_1\right\|_2 &= \left\|B\Big(D^{-r}\big(\mathscr{M}(X_1+W)+\nu^\sharp\big) - D^{-r}\big(\mathscr{M}(X_1)+e\big) - u\Big)\right\|_2 \\ &= \left\|BD^{-r}\Big(\mathscr{M}(X^\sharp)+\nu^\sharp - q\Big)\right\|_2 \leq 3m\gamma(r).\end{aligned}$$

Letting x_1 denote the vector of diagonal elements of $-\Sigma_X$ and w that of Σ_W, we have

$$y_1 = (BD^{-r}\mathscr{M})_{U_W,V_W}(x_1) + BD^{-r}e + Bu.$$

Just as in the proof of the main theorem, we remark that

$$\left\|BD^{-r}\Big(\mathscr{M}_{U_W,V_W}(x_1) + e\Big) - y_1\right\|_2 = \|Bu\|_2 \leq 3m\gamma(r).$$

In other words, both $x^\sharp := x_1 + w$ and x_1 are feasible to (3.3.12) with Φ set to M_{U_W,V_W} and q set to y_1. Moreover, we also have $\|x_1 + w\|_1 = \|X_1 + W\|_* \leq \|X_1\|_* = \|x_1\|_1$. The result now follows by Theorem 3.3.4.

□

3.4 Numerical Experiments

Herein, we present the results of a series of numerical experiments. The goal is to illustrate the performance of the algorithms studied in this paper and to compare their empirical performance to the error bounds (up to constants) predicted by the theory. All tests were performed in MATLAB using the CVX package. One thing worth noting is that, in the interest of numerical stability and computational efficiency, we modified the constraint in (3.3.1) to be

$$\sigma_\ell \|P_\ell V^*(\mathscr{M}(X) + \nu - q)\|_2 \leq \gamma(r)\sqrt{m},$$

where σ_ℓ is the ℓ-th singular value of D^{-r}. The motivation for this is that as r increases, D^r quickly becomes ill-conditioned. The analysis and conclusions of Theorem 4.4.1 remain unchanged with the above modification. The only additional cost is computing the singular value decomposition of D^{-r} before beginning the optimization. For a fixed value of m this needs to be done only once as the result can be stored and re-used.

To construct rank k matrices, we sampled $\alpha_1, \ldots, \alpha_k \sim \mathscr{N}(0,1)$, $u_1, \ldots, u_k \sim \mathscr{N}(0, I_{n_1 \times n_1})$, $v_1, \ldots, v_k \sim \mathscr{N}(0, I_{n_2 \times n_2})$, and set $X := \sum_{i=1}^k \alpha_i u_i v_i^*$. We note that under these conditions $\mathbb{E}\|X\|_F^2 = k \cdot n_1 \cdot n_2$. The measurements we collect are via a Gaussian linear operator $\mathscr{M}$ whose matrix representation consists of i.i.d. standard normal entries. For each experiment, we use a *fixed* draw of M.

First, we illustrate the decay of the reconstruction error, measured in the Frobenius norm, as a function of the order of the $\Sigma\Delta$ quantization scheme for $r = 1, 2$, and 3 in the noise-less setting. Experiments were run with the following parameters: $n_1 = n_2 = 20$, $\varepsilon = 0$, alphabet step-size $\beta = 1/2$, rank $k = 5$, and $\ell = 4 \cdot k \cdot n_1$. We let the over-sampling factor $\frac{m}{\ell}$ range from 5 to 60 by a step size of 5. The reconstruction error for a fixed over-sampling factor was averaged over 20 draws of X. The results are reported in Figure 3.4.1 for the three choices of r. As Theorem 4.4.1 predicts, the reconstruction error decays polynomially in the oversampling rate, with the

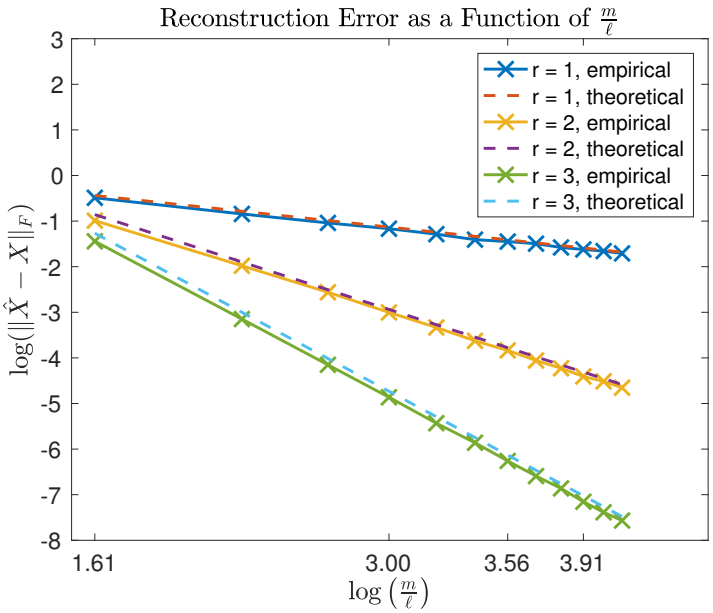


Figure 3.4.1: Log-log plot of the reconstruction error as a function of the oversampling factor $\frac{m}{\ell}$ for $r = 1, 2, 3$. Polynomial relationships will appear as linear relationships.

polynomial degree increasing with r.

To test the dependence on measurement noise, we considered reconstructing 20×20 matrices from measurements generated by a fixed draw of $\mathscr{M}$. For $\varepsilon \in \{0, 1/10, \ldots, 2\}$, we averaged our reconstruction error over 20 trials with noise vectors ν drawn from the uniform distribution on $(0,1)^m$ and normalized to have $\|\nu\|_\infty = \varepsilon$. The remaining parameters were set to the following values: $r = 1$, alphabet step-size $\beta = 1/2$, rank $k = 2$, $\ell = 4 \cdot k \cdot n_1$, and $m = 2\ell$. Figure 3.4.2 illustrates the outcome of this experiment, which agrees with the bound in Theorem 4.4.1.

The goal of the next experiment is to illustrate, in the context of encoding (Corollary 3.3.3), the exponential decay of distortion as a function of the rate, or equivalently of the reconstruction error as a function of the number of bits (i.e., rate) used. We performed numerical simulations for $\Sigma\Delta$ schemes of order 2 and 3. As before, our parameters were set to the following: $n_1 = n_2 = 20$, $\beta = 1/2$, rank of the true matrix $k = 5$, $L = 4 \cdot k \cdot n_1$, $\varepsilon = 0$, and let $\frac{m}{L}$ range from 5 to 60 by a

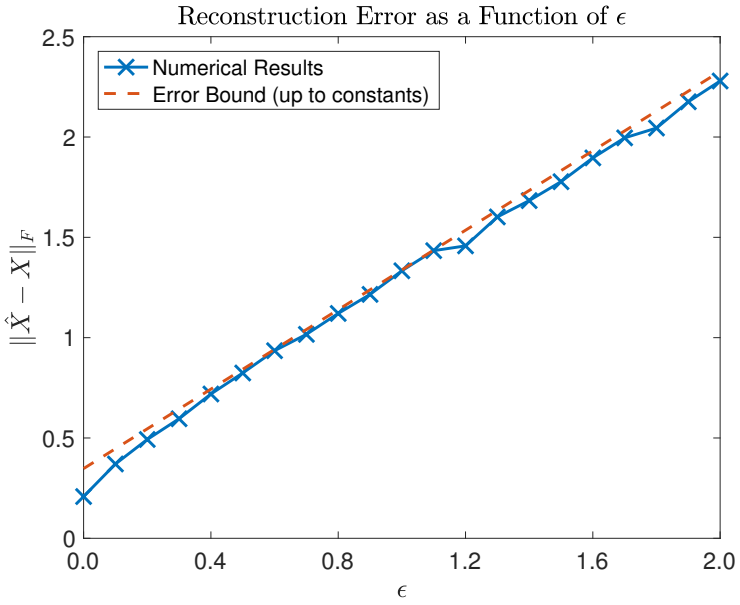


Figure 3.4.2: Reconstruction error as a function of the input noise. The rank of the true matrix is 2.

step size of 5. The rate is calculated to be $\mathscr{R} = L \cdot r \cdot \log(m)$. Again, the reconstruction error for a fixed over-sampling factor was averaged over 20 draws of X. The results are shown in Figures 3.4.3 and 3.4.4, respectively. The slopes of the lines (corresponding to the constant in the exponent in the rate-distortion relationship) which pass through the first and last points of each plot are -1.8×10^{-3} and -2.0×10^{-3} for $r = 2, 3$, respectively. It should further be noted that the numerical distortions decay much faster than the upper bound of (3.3.11). We suspect this to be due to the sub-optimal r-dependent constants in the exponent of (3.3.11), which are likely an artifact of the proof technique in [35]. Indeed, there is evidence in [35] that the correct exponent is $-r/2 + 1/4$ rather than $-r/2 + 3/4$. This is more in line with our numerical exerpiments but the proof of such is beyond the scope of this paper.

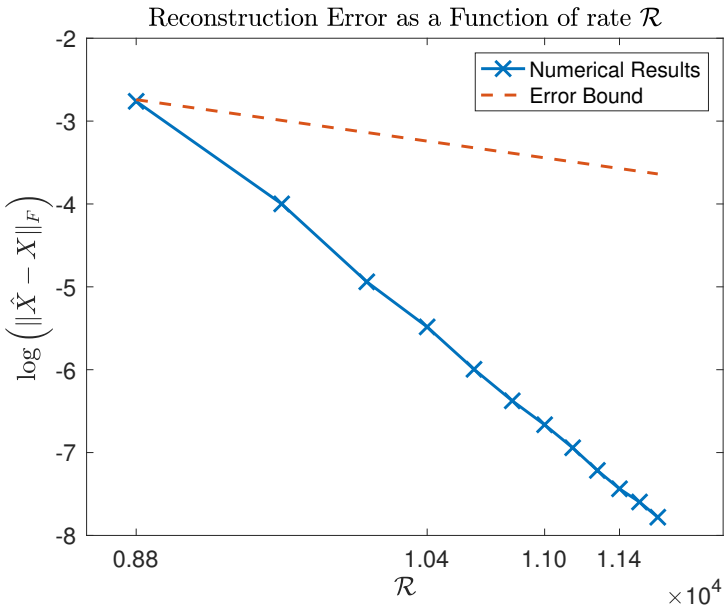


Figure 3.4.3: Log plot of the reconstruction error as a function of the bit rate $\mathscr{R} = Lr\log(m)$ for $r = 2$. Exponential relationships will appear as linear relationships.

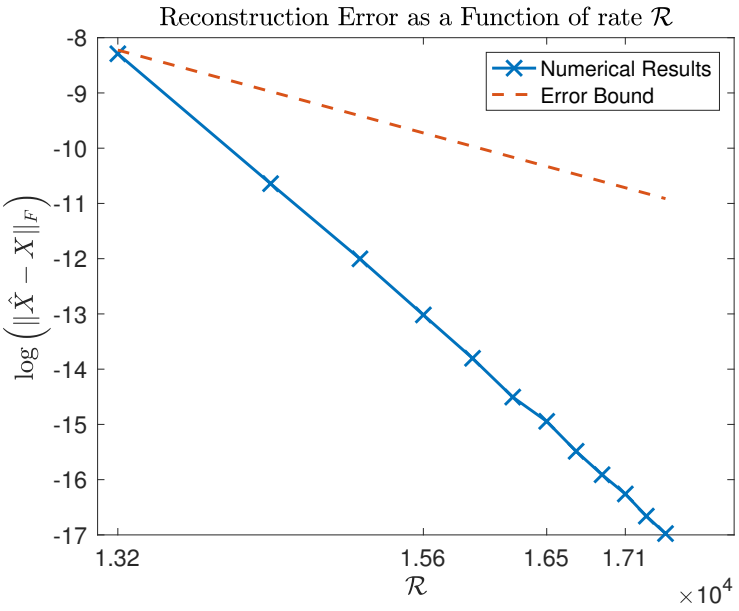


Figure 3.4.4: Log plot of the reconstruction error as a function of the bit rate $\mathscr{R} = Lr\log(m)$ for $r = 3$. Exponential relationships will appear as linear relationships.

Acknowledgements

This chapter, in full, is joint work with Rayan Saab and has been published in Information and Inference, 2019. The dissertation author was the primary investigator and author of this paper. RS was supported in part by the NSF via DMS-1517204. Additionally, both authors acknowledge support from a UCSD senate research grant award.

Bibliography

[1] John J Benedetto, O Yilmaz, and Alexander M Powell. Sigma-delta quantization and finite frames. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 3, pages iii–937. IEEE, 2004.

[2] Sonia A Bhaskar and Adel Javanmard. 1-bit matrix completion under exact low-rank constraint. In *Information Sciences and Systems (CISS), 2015 49th Annual Conference on*, pages 1–6. IEEE, 2015.

[3] Petros T Boufounos, Laurent Jacques, Felix Krahmer, and Rayan Saab. Quantization and compressive sensing. In *Compressed Sensing and its Applications*, pages 193–237. Springer, 2015.

[4] Tony Cai and Wen-Xin Zhou. A max-norm constrained minimization approach to 1-bit matrix completion. *The Journal of Machine Learning Research*, 14(1):3619–3647, 2013.

[5] E. J. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59:1207–1223, 2006.

[6] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[7] Emmanuel J Candes and Yaniv Plan. Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. *Information Theory, IEEE Transactions on*, 57(4):2342–2359, 2011.

[8] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

[9] Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[10] I. Daubechies and R. DeVore. Approximating a bandlimited function using very coarsely quantized data: a family of stable sigma-delta modulators of arbitrary order. *Ann. Math.*, 158(2):679–710, 2003.

[11] Mark A Davenport, Yaniv Plan, Ewout Van Den Berg, and Mary Wootters. 1-bit matrix completion. *Information and Inference: A Journal of the IMA*, 3(3):189–223, 2014.

[12] P. Deift, C. S. Güntürk, and F. Krahmer. An optimal family of exponentially accurate one-bit sigma-delta quantization schemes. *Comm. Pure Appl. Math.*, 64(7):883–919, 2011.

[13] Richard M Dudley. The sizes of compact subsets of hilbert space and continuity of gaussian processes. *Journal of Functional Analysis*, 1(3):290–330, 1967.

[14] Maryam Fazel. *Matrix rank minimization with applications*. PhD thesis, PhD thesis, Stanford University, 2002.

[15] Joe-Mei Feng, Felix Krahmer, and Rayan Saab. Quantized compressed sensing for partial random circulant matrices. *arXiv preprint arXiv:1702.04711*, 2017.

[16] Simon Foucart and Holger Rauhut. *A mathematical introduction to compressive sensing*. Birkhäuser Basel, 2013.

[17] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert. Quantum State Tomography via Compressed Sensing. *Physical Review Letters*, 105(15):150401, October 2010.

[18] C Sinan Güntürk, Mark Lammers, Alex Powell, Rayan Saab, and Özgür Yilmaz. Sigma delta quantization for compressed sensing. In *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, pages 1–6. IEEE, 2010.

[19] C.S. Güntürk. One-bit sigma-delta quantization with exponential accuracy. *Comm. Pure Appl. Math.*, 56(11):1608–1630, 2003.

[20] H. Inose and Y. Yasuda. A unity bit coding method by negative feedback. *Proceedings of the IEEE*, 51(11):1524–1535, 1963.

[21] Mark Iwen and Rayan Saab. Near-optimal encoding for sigma-delta quantization of finite frame expansions. *Journal of Fourier Analysis and Applications*, 19(6):1255–1273, 2013.

[22] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

[23] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11(Jul):2057–2078, 2010.

[24] Felix Krahmer, Shahar Mendelson, and Holger Rauhut. Suprema of chaos processes and the restricted isometry property. *Communications on Pure and Applied Mathematics*, 67(11):1877–1904, 2014.

[25] Felix Krahmer, Rayan Saab, and Rachel Ward. Root-exponential accuracy for coarse quantization of finite frame expansions. *IEEE Transactions on Information Theory*, 58(2):1069–1079, 2012.

[26] Felix Krahmer, Rayan Saab, and Özgür Yilmaz. Sigma–delta quantization of sub-gaussian frame expansions and its application to compressed sensing. *Information and Inference: A Journal of the IMA*, 3(1):40–58, 2014.

[27] Jean Lafond, Olga Klopp, Eric Moulines, and Joseph Salmon. Probabilistic low-rank matrix completion on finite alphabets. In *Advances in Neural Information Processing Systems*, pages 1727–1735, 2014.

[28] Andrew S Lan, Christoph Studer, and Richard G Baraniuk. Matrix recovery from quantized and corrupted measurements. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 4973–4977. IEEE, 2014.

[29] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[30] Samet Oymak, Karthik Mohan, Maryam Fazel, and Babak Hassibi. A simplified approach to recovery conditions for low rank matrices. In *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, pages 2318–2322. IEEE, 2011.

[31] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 502–511. IEEE, 2008.

[32] Swati Rallapalli, Lili Qiu, Yin Zhang, and Yi-Chao Chen. Exploiting temporal stability and low-rank structure for localization in mobile networks. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 161–172. ACM, 2010.

[33] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

[34] Rayan Saab, Rongrong Wang, and Özgür Yılmaz. Quantization of compressive samples with stable and robust recovery. *Applied and Computational Harmonic Analysis*, 2016.

[35] Rayan Saab, Rongrong Wang, and Özgür Yılmaz. From compressed sensing to compressed bit-streams: practical encoders, tractable decoders. *IEEE Transactions on Information Theory*, 2017.

[36] M. Talagrand. *The Generic Chaining*. Springer-Verlag, 2005.

[37] Roman Vershynin. *Introduction to the non-asymptotic analysis of random matrices*, pages 210 – 268. Cambridge University Press, 2012.

Chapter 4

New Algorithms and Improved Guarantees for One-Bit Compressed Sensing on Manifolds

We study the problem of approximately recovering signals on a manifold from one-bit linear measurements drawn from either a Gaussian ensemble, partial circulant ensemble, or bounded orthonormal ensemble and quantized using $\Sigma\Delta$ or distributed noise shaping schemes. We assume we are given a Geometric Multi-Resolution Analysis, which approximates the manifold, and we propose a convex optimization algorithm for signal recovery. We prove an upper bound on the recovery error which outperforms prior works that use memoryless scalar quantization, requires a simpler analysis, and extends the class of measurements beyond Gaussians. Finally, we illustrate our results with numerical experiments.

4.1 Introduction

Compressed sensing [3, 5] demonstrates that structured high dimensional signals such as sparse vectors or low-rank matrices can be recovered from few random linear measurements. Recovery is typically formulated as a convex optimization problem whose minimizer cannot be expressed analytically and must be solved for using numerical algorithms running on digital devices. Thus, it is necessary to consider the effect of quantization in the design of the recovery algorithms. Indeed, sparse vector recovery and low-rank matrix recovery have been studied in the presence of various quantization schemes [7, 8, 11, 12, 13]. We look to extend these results to account for those structured signals that lie on a compact, low-dimensional submanifold of $\mathbb{R}^N$ for which we have a Geometric Multi-Resolution Analysis (GMRA) [1]. Our work is motivated by the results of Iwen et al. in [9] where they assume memoryless scalar quantized Gaussian measurements, and we provide better error bounds that hold for a wider class of measurement ensembles.

As in [9], a key component of our technique is the GMRA which approximates the manifold at various levels of refinement. At each level the GMRA is a collection of approximate tangent spaces about certain known "centers", and the quality of the approximation improves with every level. Unlike in [9], the quantization schemes that we use are $\Sigma\Delta$ or distributed noise shaping methods (see, e.g., [7, 8]) and the compressed sensing measurements that our results apply to include those drawn from Gaussian ensembles, partial circulant ensembles (PCE) or bounded orthonormal ensembles (BOE) (see [8] for precise definitions). Our proposed reconstruction method is summarized in Algorithm 1. This simple algorithm first finds a GMRA center that quantizes to a bit sequence close to the quantized measurements, where "closeness" is determined using a pseudo-metric that respects the quantization; it then optimizes over all points in the associated approximate tangent space to enforce, as much as possible, the consistency of the quantization. Using the results of [8] we prove that the quantization error associated with our

proposed reconstruction algorithm decays polynomially or exponentially as a function of the number of measurements, depending on the quantization scheme. This greatly improves on the sub-linear error decay associated with scalar quantization in [9].

4.2 Background

In [9], Iwen et al. study the case where measurements $y = Ax$ of a signal x on a manifold $K \subset \mathbb{S}^{N-1}$ are quantized via memoryless scalar quantization (MSQ). For a discrete set $\mathscr{A}$ and $\mathscr{Q}(x) := \arg\min_{z\in\mathscr{A}} |x-z|$, the measurements are

$$q_j = \mathscr{Q}(\langle a_j, x\rangle), \quad j = 1,\dots,m.$$

For example, one could take $\mathscr{A} = \{\pm 1\}$ and $\mathscr{Q}(\cdot) = \text{sign}(\cdot)$. [9] proposes an algorithm for recovering x from such measurements and shows that the associated error decays like $O(m^{-1/7})$. Such slow error decay, associated with MSQ, has also been seen in the context of sparse vector recovery in the compressed sensing literature. Indeed, it is known in that setting that the error under any reconstruction scheme using MSQ measurements cannot decay faster than $O(m^{-1})$ [6] (see also [2]). So, to acheive better error rates one must use more sophisticated quantization schemes. For example, in the sparse vector setting noise shaping techniques such as $\Sigma\Delta$ and distributed noise shaping leverage redundancy of the measurements to ensure error decay like $O(m^{-r})$ or $O(\beta^{-cm})$ for some parameters $r \in \mathbb{N}$, $\beta > 1$ that depend on the quantization scheme, e.g., [4, 14]. As we will also use these schemes, we now briefly describe them.

Each of the quantization methods mentioned above employs a state variable $u \in \mathbb{R}^m$ and quantizes measurements in a recursive fashion: $q_j = \mathscr{Q}\left(f(y_j,\dots,y_1,u_{j-1},\dots,u_1)\right)$, where f is some function designed for the quantization scheme. The state variable is then updated via the state relation $Ax - q = Hu$, where $H : \mathbb{R}^m \to \mathbb{R}^m$ is a lower-triangular Toeplitz matrix. Important for the analysis (and for practical reasons) is that H, f are chosen so that whenever $\|x\|_\infty$ is

bounded, we have $\|u\|_\infty < C$; the upper bound C is often referred to as the stability constant of the quantization scheme. For a more detailed explanation of these noise shaping techniques, the interested reader may refer for example to [8]. For the sake of expositional simplicity, we will only consider the $\Sigma\Delta$ setting, but our arguments work for distributed noise shaping under very minor adjustments.

4.3 Problem Formulation and Notation

For an integer ℓ, $[\ell] := \{1, \dots, \ell\}$. We use $\gtrsim$ and $\lesssim$ for inequalities that hold up to a constant; subscripts indicate the constant depends on a specified parameter. Let $K \subset B_2^N$ be a d-dimensional submanifold of the unit ℓ_2-ball in $\mathbb{R}^N$. We assume that we have a GMRA of K, which we make precise below. First, for a set $T \subset \mathbb{R}^N$ and $\rho > 0$, define

$$\text{tube}_\rho(T) := \left\{ x \in \mathbb{R}^N : \ \inf_{y \in T} \|x - y\|_2 \leq \rho \right\}.$$

Definition 4.3.1 ([10])**.** Let $J \in \mathbb{N}$ and $K_0, \dots, K_J \in \mathbb{N}$. A *GMRA* of K is a collection $\{(\mathscr{C}_j, \mathscr{P}_j)\}_{j \in [J]}$ of centers $\mathscr{C}_j = \{c_{j,k}\}_{k \in [K_j]}$ and affine projections

$$\mathscr{P}_j = \left\{ P_{j,k} \colon \mathbb{R}^N \to \mathbb{R}^N : \ k \in [K_j] \right\}$$

with the following properties:

1. **Affine Projections**. Every $P_{j,k}$ is an orthogonal projection onto some d-dimensional affine space which contains the center $c_{j,k}$.

2. **Dyadic Structure**. The number of centers at each level is bounded by $|\mathscr{C}_j| = K_j \leq C_C 2^{dj}$ for an absolute constant $C_C \geq 1$. Moreover, there exist $C_1 > 0$, $C_2 \in (0, 1]$ such that

   (a) $K_j \leq K_{j+1}$ for all $j \in [J-1]$,

(b) $\|c_{j,k_1} - c_{j,k_2}\|_2 > C_1 2^{-j}$ for all $j \in [J]$, $k_1 \neq k_2 \in [K_j]$,

(c) For each $j \in [J] \setminus \{0\}$ there exists a parent function $p_j \colon [K_j] \to [K_{j-1}]$ with

$$\|c_{j,k} - c_{j-1,p_j(k)}\|_2 \leq C_2 \min_{k' \in [K_{j-1}] \setminus \{p_j(k)\}} \|c_{j,k} - c_{j-1,k'}\|_2.$$

3. **Multiscale Approximation**. The projectors in $\mathscr{P}_j$ approximate K in the following sense:

(a) There exists $j_0 \in [J-1]$ such that $c_{j,k} \in \text{tube}_{C_1 2^{-j-2}}(K)$ for all $j \geq j_0$ and $k \in [K_j]$.

(b) For each $j \in [J]$ and $z \in \mathbb{R}^N$, let

$$c_{j,k_j(z)} \in \operatorname*{arg\,min}_{c_{j,k} \in \mathscr{C}_j} \|z - c_{j,k}\|_2.$$

Then for each $z \in K$ there exist $C_z, \widetilde{C}_z > 0$ so that $\|z - P_{j,k_j(z)} z\|_2 \leq C_z 2^{-2j}$ for all $j \in [J]$ and

$$\|z - P_{j,k'} z\|_2 \leq \widetilde{C}_z 2^{-j} \tag{4.3.1}$$

whenever $j \in [J]$ and $k' \in [K_j]$ satisfy

$$\|z - c_{j,k'}\|_2 \leq 16 \max\left\{ \|z - c_{j,k_j(z)}\|_2,\, C_1 2^{-j-1} \right\}.$$

Let $\{(\mathscr{C}_j, \mathscr{P}_{j,k})\}_j$ be a GMRA of a smooth compact manifold $K \subset (1-\mu) B_2^N$ for some $\mu \in (0,1)$, and define the scale-j GMRA approximation $\hat{K}_j := \{P_{j,k_j(z)} z : \ \|z\|_2 \leq 1\} \cap B_2^N$. We suppose that j_0 is large enough so that $\sup_{x \in K} \widetilde{C}_x 2^{-j_0} \leq \mu$ to ensure $\{P_{j,k'} x : \ x \in K, \ (j,k') \text{ as in 3.b of Definition 4.3.1}\} \subset B_2^N$, and further assume that $\text{tube}_{C_1 2^{-j_0-2}}(K) \subset B_2^N$ which ensures $\mathscr{C}_j \subset \hat{K}_j$ for $j \geq j_0$. The number of measurements required for our theoretical guarantees to hold will depend on two notions of complexity of K and the GMRA. For

$g \sim \mathcal{N}(0, I_N)$, define

$$w(S) := \mathbb{E} \sup_{x \in S} \langle g, x \rangle, \qquad \text{rad}(S) := \sup_{x \in S} \|x\|_2,$$

and, for $j \geq j_0$, define

$$C_K := \max\left\{C_1, \sup_{z \in K} \widetilde{C}_z\right\}, \qquad S := K \cup \hat{K}_j.$$

Now, let $\mathscr{Q}$ be a stable r^{th} order $\Sigma\Delta$ quantizer with stability constant $C(r)$ and associated alphabet $\mathscr{A}$. Let $x \in K$, $A \in \mathbb{R}^{m \times N}$ be a standard Gaussian matrix (or a matrix drawn from a PCE/BOE), $D_\varepsilon \in \mathbb{R}^{N \times N}$ a diagonal matrix with random signs (independent of A) along the diagonal, $\Phi := AD_\varepsilon$ and

$$q := \mathscr{Q}(\Phi x).$$

Our goal, given q and Φ*, is to approximate x and show that the associated error bounds decay fast as a function of m.*

A useful fact is that the binary embedding provided by $\Sigma\Delta$ quantization approximately preserves Euclidean distance via a related pseudo-metric on the quantized vectors, defined as follows. For $p \leq m$, define $\lambda := m/p =: r\widetilde{\lambda} - r + 1$ and $v \in \mathbb{R}^\lambda$ be the (row) vector whose j^{th} entry v_j is the j^{th} coefficient of the polynomial of $(1 + z + \ldots + z^{\widetilde{\lambda}})^r$. Set $\gamma = \|v\|_1/\|v\|_2$ and define $\widetilde{V} \colon \mathbb{R}^m \to \mathbb{R}^p$ by $\widetilde{V} = \frac{9}{8\|v\|_2\sqrt{p}} I_p \otimes v$, where $\otimes$ denotes to the Kronecker product. Then the pseudo-metric is given by $d_{\widetilde{V}}(x, y) := \|\widetilde{V}(x - y)\|_2$.

4.4 Main Results

We now present our recovery algorithm and its associated error guarantees.

Algorithm 1 Reconstruction Algorithm

Step 1: Find $c_{j,k'} \in \arg\min_{c_{j,k} \in \mathscr{C}_j} \|\widetilde{V}(\mathscr{Q}(\Phi c_{j,k}) - q)\|_2$.
Step 2: If $\|\widetilde{V}(\mathscr{Q}(\Phi c_{j,k'}) - q)\|_2 = 0$, set $x^\sharp = c_{j,k'}$; else

$$x^\sharp = \underset{z \in \mathbb{R}^N}{\arg\min} \left\| \widetilde{V}\left(\Phi z - q\right) \right\|_2 \text{ s.t. } z = P_{j,k'}(z),\ \|z\|_2 \leq 1.$$

Theorem 4.4.1. *Suppose we have a GMRA of $K \subset (1-\mu)B_2^N$ at level $j \geq j_0$. Let $\mathscr{Q}$ be a stable r^{th} order $\Sigma\Delta$ quantizer with $r = O(j)$, associated alphabet $\mathscr{A} = \{\pm(1-\alpha)\sqrt{m}\}$ where $\alpha^{-1} \gtrsim \mathrm{rad}^2(S-S)2^{2(j+1)}$, and*

$$m^{\frac{r-1}{r-1/2}} \gtrsim \gamma^2(1+\nu)^2 \log^4(N) j^2 2^{4j} \times \mathrm{rad}(S-S)^4 \max\left\{1, w^2(S-S)\mathrm{rad}^{-2}(S-S)\right\}. \tag{4.4.1}$$

Then with probability exceeding $1-e^{-\nu}$, for all $x \in K$, $x^\sharp$ from Algorithm 1 satisfies

$$\|x^\sharp - x\|_2 \lesssim_r \widetilde{C}_x 2^{-j}.$$

Proof. By the triangle inequality we have

$$\|x^\sharp - x\|_2 \leq \|x^\sharp - P_{j,k'}x\|_2 + \|P_{j,k'}x - x\|_2.$$

Both terms are bounded by $\widetilde{C}_x 2^{-j}$, the first by Lemma 4.4.3 and the second by Lemma 4.4.2 and Equation (4.3.1) from Definition 4.3.1. □

Remark 8. As Lemma 4.3 of [9] shows, $w(S-S) \lesssim w(K) + \sqrt{dj}$. This is a suitable bound for coarse GMRA scales, i.e. $j \lesssim \log(N)$. However, for $j \gtrsim \log(N)$ one can slightly modify the definition of S and use the bound $w(S-S) \lesssim (w(K)+1)\log(N)$ as proven in Lemma 4.5 of [9], albeit this requires some modifications to the proof of Theorem 4.4.1. Please see Remark 4.15 of

[9] for more details, as we shall leave the latter case for future work.

Lemma 4.4.2. *Under the assumptions of Theorem 4.4.1, with probability exceeding* $1-e^{-\nu}$*, for all* $x \in K$*, the center* $c_{j,k'}$ *chosen in Step 1 of Algorithm 1 satisfies*

$$\|x - c_{j,k'}\|_2 \leq 16 \max\left\{\|x - c_{j,k_j(x)}\|_2, C_1 \cdot 2^{-j-1}\right\}. \tag{4.4.2}$$

Proof. Theorem 5.2 and Remarks 3, 5 of [8] state that if

$$m^{\frac{r-1}{r-1/2}} \geq p \gtrsim \gamma^2(1+\nu)^2 \log^4(N) \frac{\max\{1, w^2(S-S)\mathrm{rad}^{-2}(S-S)\}}{\alpha^2}$$

and we choose $r = \lfloor \frac{\lambda}{2Ce} \rfloor^{1/2}$, where $\lambda = m^{\frac{r-1}{r-1/2}}/p$ and $C > 0$, then with probability exceeding $1-e^{-\nu}$

$$\begin{aligned} &\left| d_{\widetilde{V}}\left(\mathscr{Q}(\Phi c_{j,k}), q\right) - \|c_{j,k} - x\|_2 \right| \\ &\quad \lesssim \max\left\{\sqrt{\alpha}, \alpha\right\} \mathrm{rad}(S-S) + e^{-c\sqrt{\lambda}} \end{aligned}$$

for all $c_{j,k} \in \mathscr{C}_j$. Conditioning on this, we have

$$\begin{aligned} & d_{\widetilde{V}}\left(\mathscr{Q}(\Phi c_{j,k'}), q\right) \leq d_{\widetilde{V}}\left(\mathscr{Q}(\Phi c_{j,k_j(x)}), q\right) \\ \implies & \|c_{j,k'} - x\|_2 \lesssim_r \|c_{j,k_j(x)} - x\|_2 \\ & \qquad + \max\left\{\sqrt{\alpha}, \alpha\right\} \mathrm{rad}(S-S) + e^{-c\sqrt{\lambda}}. \end{aligned}$$

For $c_{j,k'}$ to satisfy (4.4.2) (i.e., part 3.b of Definition 4.3.1), it suffices to choose α small and λ large, so that $\max\{\sqrt{\alpha}, \alpha\}\mathrm{rad}(S-S) + e^{-c\sqrt{\lambda}} \leq 15 C_1 2^{-j-1}$. Choosing $\lambda \gtrsim j^2$ (hence, $r = O(j)$) and $\alpha^{-2} \gtrsim \mathrm{rad}^4(S-S) 2^{4(j+1)}$ realizes the above bound. □

Lemma 4.4.3. *Under the assumptions of Theorem 4.4.1, with probability exceeding* $1-e^{-\nu}$*, for all* $x \in K$*,* $x^\sharp$ *from Algorithm 1 satisfies* $\|x^\sharp - P_{j,k'}x\|_2 \lesssim_r \widetilde{C}_x 2^{-j}$.

Proof. By optimality of $x^\sharp$ and feasibility of $P_{j,k'}x$, the triangle inequality gives us

$$\begin{aligned} 0 &\leq \left\| \widetilde{V}\left(\Phi P_{j,k'}x - q\right)\right\|_2 - \left\| \widetilde{V}\left(\Phi x^\sharp - q\right)\right\|_2 \\ &\leq 2\left\| \widetilde{V}\left(\Phi P_{j,k'}x - q\right)\right\|_2 - \left\| \widetilde{V}\Phi\left(x^\sharp - P_{j,k'}x\right)\right\|_2. \end{aligned}$$

Define $q^* := \mathscr{Q}\left(\Phi P_{j,k'}x\right)$. Then

$$\begin{aligned} \left\| \widetilde{V}\Phi\left(x^\sharp - P_{j,k'}x\right)\right\|_2 &\leq 2\left\| \widetilde{V}\left(\Phi P_{j,k'}x - q^* + q^* - q\right)\right\|_2 \\ &\leq 2\left\| \widetilde{V}\left(\Phi P_{j,k'}x - q^*\right)\right\|_2 + 2d_{\widetilde{V}}(q,q^*). \end{aligned}$$

Lemma 4.5 of [8] states that $\|\widetilde{V}(\Phi P_{j,k'}x - q^*)\|_2 \lesssim_r e^{-c\sqrt{\lambda}} \lesssim_r 2^{-j}$, while Theorem 5.2 of [8] and (4.3.1) (via Lemma 4.4.2) imply $d_{\widetilde{V}}(q,q^*) \lesssim \widetilde{C}_x 2^{-j}$ with probability exceeding $1 - e^{-\nu}$. Therefore, we have

$$\left\| \widetilde{V}\Phi\left(x^\sharp - P_{j,k'}x\right)\right\|_2 \lesssim_r \widetilde{C}_x 2^{-j}. \tag{4.4.3}$$

By the definition of S, we have $x^\sharp, P_{j,k'}x \in S$. Equation (5.8) in the proof of Theorem 5.2 of [8] states that with probability exceeding $1 - e^{-\nu}$ $\left\| \widetilde{V}\Phi\left(x^\sharp - P_{j,k'}x\right)\right\|_2 \gtrsim \left\|x^\sharp - P_{j,k'}x\right\|_2 - 2^{-j}$. With (4.4.3), this yields $\|x^\sharp - P_{j,k'}x\|_2 \lesssim_r \widetilde{C}_x 2^{-j}$. □

Remark 9. If one does not choose r and α as in the proofs above, but works with r large enough and α small enough, a version of Theorem 4.4.1 with $\|x^\sharp - x\|_2 \lesssim \widetilde{C}_x 2^{-j} + \max\{\sqrt{\alpha}, \alpha\}\mathrm{rad}(S - S) + (m/p)^{-r+1/2}$ holds for m, p as in the proof of Lemma 4.4.2.

Remark 10. When ΣΔ quantization is replaced by distributed noise shaping, Theorem 4.4.1 holds with $j^2 2^{4j}$ in the lower bound (4.4.1) replaced by $j2^{4j}$, and Remark 9 holds with $\beta^{-m/p}$ replacing $(m/p)^{-r+1/2}$.

4.5 Numerical Simulations

To simulate Algorithm 1, we take $K = \mathbb{S}^2$ embedded in $\mathbb{R}^{20}$ and construct a GMRA up to level $j_{\max} = 15$ using 20,000 data points sampled uniformly from K. We randomly select a test set of 100 points $x \in K$ for use throughout all experiments. In each experiment (i.e., point in Figure 4.5.1), compressed sensing measurements $y = \Phi x = m^{-1/2} A D_\varepsilon x$ are taken for each test point, with $A \sim \mathcal{N}(0, I_{m \times N})$ and D_ε a diagonal $N \times N$ matrix of random ± 1s. We recover $x^\sharp$ from the r^{th} order $\Sigma\Delta$ measurements $\mathcal{Q}(y)$ via Algorithm 1 where, for practical reasons, the alphabet from Theorem 4.4.1 is modified to be $\mathcal{A} = \{\pm 1\}$. We vary $\lambda = m/p$ for fixed r, p, and refinement scale j. As in Remark 9, the reconstruction error decays as a function of λ until reaching a floor due to the refinement level of the GMRA.

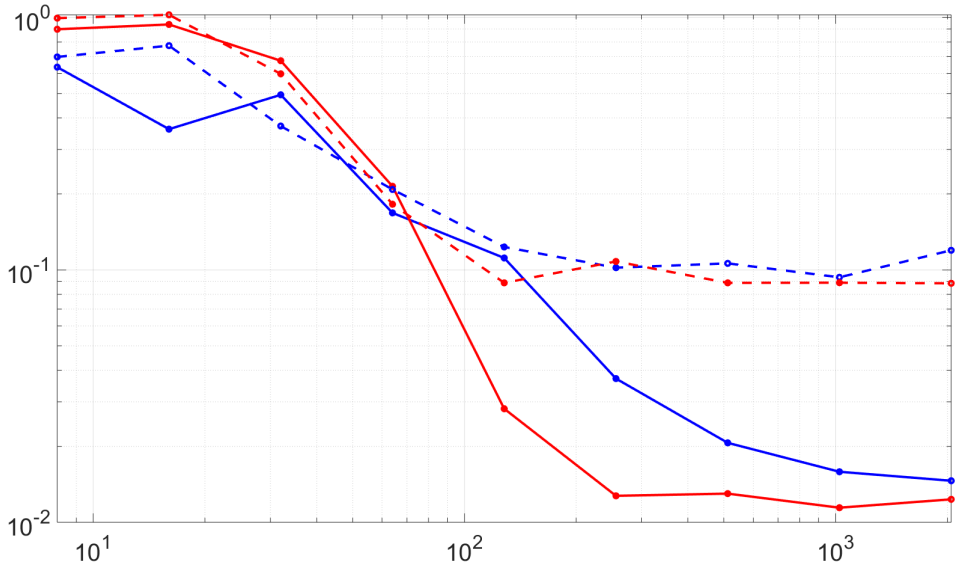


Figure 4.5.1: Log plot of average relative reconstruction error from Algorithm 1 as a function of $\lambda = m/p$ for $p = 10$. Solid lines correspond to $j = 12$; dashed lines to $j = 6$. Blue and red plots represent $r = 2, 4$ (resp.). For each j, the error decays as a function of λ until reaching a floor due the GMRA error.

4.6 Acknowledgements

We thank F. Krahmer, S. Krause-Solberg, and J. Maly for sharing their GMRA code, which they adapted from that provided by M. Maggioni. This chapter, in full, is joint work with Mark Iwen, Aaron Nelson, and Rayan Saab and has been published in 13th International conference on Sampling Theory and Applications (SampTA). IEEE, 2019. The dissertation author was the primary investigator and author of this paper. M. A. Iwen was supported in part by NSF CCF-1615489; R. Saab was supported in part by NSF DMS-1517204.

Bibliography

[1] William K Allard, Guangliang Chen, and Mauro Maggioni. Multi-scale geometric methods for data sets ii: Geometric multi-resolution analysis. *Applied and Computational Harmonic Analysis*, 32(3):435–462, 2012.

[2] Petros T. Boufounos, Laurent Jacques, Felix Krahmer, and Rayan Saab. Quantization and compressive sensing. *preprint arXiv:1405.1194*, 2014.

[3] Emmanuel J. Candes, Justin K. Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59(8):1207–1223, 2006.

[4] Evan Chou and C. Sinan Güntürk. Distributed noise-shaping quantization: I. beta duals of finite frames and near-optimal quantization of random measurements. *Constr. Approx.*, 44(1):1–22, 2016.

[5] David L. Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, 2006.

[6] Vivek K. Goyal, Martin Vetterli, and Nguyen T. Thao. Quantization of overcomplete expansions. In *Data Compression Conference, 1995. DCC'95. Proceedings*, pages 13–22. IEEE, 1995.

[7] C. Sinan Güntürk, Mark Lammers, Alex Powell, Rayan Saab, and Özgür Yilmaz. Sigma delta quantization for compressed sensing. In *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, pages 1–6. IEEE, 2010.

[8] Thang Huynh and Rayan Saab. Fast binary embeddings, and quantized compressed sensing with structured matrices. *preprint arXiv:1801.08639*, 2018.

[9] Mark A. Iwen, Felix Krahmer, Sara Krause-Solberg, and Johannes Maly. On recovery guarantees for one-bit compressed sensing on manifolds. *preprint arXiv:1807.06490*, 2018.

[10] Mark A. Iwen and Mauro Maggioni. Approximation of points on low-dimensional manifolds via random linear projections. *Inf. Inference*, 2(1):1–31, 2013.

[11] Laurent Jacques, Jason N. Laska, Petros T. Boufounos, and Richard G. Baraniuk. Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. *IEEE Trans. Inf. Theory*, 59(4):2082–2102, 2013.

[12] Eric Lybrand and Rayan Saab. Quantization for low-rank matrix recovery. *Inf. Inference*, 2017.

[13] Yaniv Plan and Roman Vershynin. One-bit compressed sensing by linear programming. *Comm. Pure Appl. Math.*, 66(8):1275–1297, 2013.

[14] Rayan Saab, Rongrong Wang, and Özgür Yılmaz. From compressed sensing to compressed bit-streams: practical encoders, tractable decoders. *IEEE Trans. Inf. Theory*, 64(9):6098–6114, 2018.