Applications and limits of convex optimization

by

Linus Hamilton

Submitted to the Department of Applied Mathematics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Applied Mathematics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2022

Author . . . .
Department of Applied Mathematics
April 25, 2022

Certified by . . . .
Ankur Moitra
Norbert Wiener Professor of Mathematics
Thesis Supervisor

Accepted by . . . .
Jonathan Kelner
Chairman, Department Committee on Graduate Theses

Applications and limits of convex optimization

by

Linus Hamilton

Submitted to the Department of Applied Mathematics
on April 25, 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Applied Mathematics

Abstract

Every algorithmic learning problem becomes vastly more tractable when reduced to a convex program, yet few can be simplified this way. At the heart of this thesis are two hard problems with unexpected convex reformulations. The Paulsen problem, a longstanding open problem in operator theory, was recently resolved by Kwok et al [40]. We use a convex program due to Barthe to present a dramatically simpler proof with an accompanying efficient algorithm that also achieves a better bound. Next, we examine the related operator scaling problem, whose fastest known algorithm uses convex optimization in non-Euclidean space. We expose a fundamental obstruction to such techniques by proving that, under realistic noise conditions, hyperbolic space admits no analogue of Nesterov's accelerated gradient descent. Finally, we generalize Bresler's structure learning algorithm from Ising models to arbitrary graphical models. We compare our results to a recent convex programming reformulation of the same problem. Notably, in variants of the problem where one only receives partial samples, our combinatorial algorithm is almost unaffected, whereas the convex approach fails to get off the ground.

Thesis Supervisor: Ankur Moitra
Title: Norbert Wiener Professor of Mathematics

Acknowledgments

I am indebted to a great many people and groups. Here are a few.

First and foremost, to my advisor Ankur Motira: thank you for giving me interesting and approachable problems when I felt directionless. Thank you for tolerating me doing essentially nothing productive in the first 6 months of the pandemic, and in general, for being a friendly and understanding person. And thank you for giving me countless pieces of advice both for my PhD and for what comes next. I truly treasure the years I've spent at MIT under your mentorship.

I thank the Hertz Foundation for their generous Hertz Fellowship financial support and knowledgeable community.

I thank my boyfriend Daniel Palumbo, for editing, for keeping me physically as well as mentally healthy, and for energy-restoring hugs. Any errors introduced after his proofread are my own fault.

I thank the living group ET (et.mit.edu), for housing a steady flow of friends as well as grad students to share ideas with.

I thank my old tutor Michael Brin, for exposing me to interesting and advanced math throughout middle and high school.

And I thank Mom and Dad for, among 10000 other things, that monthly cheese subscription.

Contents

Chapter 1

Introduction

Modern machine learning owes its life to nonconvex optimization. Throwing gradient descent at a problem as complicated as a neural network's loss function will never work in theory, but engineers do not seem to care.

However, nonconvex optimization is expensive in both time and energy [27]. Also, despite the apparent generality of stochastic gradient descent, toolbox of tricks are frequently required to avoid overfitting or bursting into a slurry of NaNs [42]. In rare happy occasions though, we can formulate a learning problem in terms of *convex* optimization. For example, consider matrix completion, the problem of filling in the missing entries in a matrix to make it low-rank. Matrix completion was inspired by the Netflix Prize [14], which tasked researchers with filling in the unknown entries of a matrix N whose ij^{th} entry is how many stars user i rates movie j. It is reasonable to guess that N is *approximately* low-rank, if its entries depend only on a limited number of factors such as movie genre, number of explosions, etc.

Though matrix completion is NP-complete, Candes and Tao [14] showed that, under realistic incoherence conditions, it can be solved *exactly* by minimizing the 'nuclear norm' of the matrix. Since the nuclear norm of a matrix is a convex function of the entries, Candes and Tao's result is a miraculous case of a non-convex problem being reformulated as a convex program.

The thrust of this thesis is twofold. We will see two more cases of hard problems – graphical model structure learning and the Paulsen problem – solved using fortuitous convex formulations. Yet, we will also see some fundamental limits of convex optimization. In the case of graphical model structure learning, the delicate construction of the 'miraculous' convex program renders it inflexible to small perturbations of the problem. Combinatorial approaches, on the other hand, can easily adapt to such changes. Meanwhile for the Paulsen problem, we will explore a generalization which ties into the far-reaching problem of convex optimization in non-Euclidean manifolds. We will prove a new and fundamental lower bound on how efficient convex optimization can be in negatively curved geometries.

1.1 The Paulsen problem

The Paulsen problem was once thought to be "one of the most intractable problems [in frame theory]" [17, 16]. To state the problem, we need the following definition:

Definition 1. We say that a set of vectors $v_1, v_2, \ldots, v_n \in \mathbb{R}^d$ is an *equal norm Parseval frame* if

$$\sum_{i=1}^{n} v_i v_i^T = I \text{ and } \|v_i\|^2 = \frac{d}{n} \text{ for each } i.$$

Alternatively, we say that it is an *ε-nearly equal norm Parseval frame* if

$$(1-\varepsilon)I \preceq \sum_{i=1}^{n} v_i v_i^T \preceq (1+\varepsilon)I \text{ and } (1-\varepsilon)\frac{d}{n} \leq \|v_i\|^2 \leq (1+\varepsilon)\frac{d}{n} \text{ for each } i.$$

Equal-norm Parseval frames are a generalization of orthonormal bases. Indeed, when $n = d$, they are the same thing.

Let us list a few examples of equal-norm Parseval frames. The vertices of any Platonic solid work, because the high degree of symmetry of these solids guarantees that the vertices are in isotropic position. The union of two equal-norm Parseval frames, scaled appropriately, is also equal-norm Parseval. The examples end there. Only a few other algebraic constructions are known [59].

Signal decoding algorithms perform especially well for equal-norm Parseval frames that also satisfy the Grassmannian property, meaning the closest pair of vectors is as far apart as possible [48]. Such frames also see application in quantum information theory [45]. Unfortunately, they are difficult to construct.

Holmes and Paulsen [31] observed that large frames with random vectors are, with high probability, *ε-nearly* equal norm Parseval. They attempted to construct a nearly-optimal frame by first finding a large Grassmannian frame, and then correcting it to be equal-norm Parseval. But is this actually possible? Given an ε-nearly equal norm Parseval frame, can we always nudge each vector by a small distance to make it exactly equal-norm Parseval? This question became known as the Paulsen problem [6]. Let us formally state it now:

Question 1. *Let $v_1, \ldots, v_n \in \mathbb{R}^d$ be an arbitrary ε-nearly equal norm Parseval frame. Does there necessarily exist an exactly equal norm Parseval frame $w_1, \ldots, w_n \in \mathbb{R}^d$, such that the total squared distance $\sum \|v_i - w_i\|^2$ is at most poly(d, ε)? What is the best upper bound on the total squared distance?*[1]

1.1.1 Prior work

How does one nudge a frame so that it satisfies both the equal-norm and Parseval equalities? The most vanilla method is gradient descent. Casazza, Fickus, and Mixon [18] showed that when n and d are relatively prime, gradient descent achieves total squared distance $O(d^{42}n^{14}\varepsilon^2)$. Unfortunately this does not resolve the Paulsen problem because the bound depends on n.

Kwok, Lau, Lee and Ramachandran [40] gave the first bound that was polynomial in ε and d. Through a tour-de-force utilizing operator scaling, connections to dynamical systems and ideas from smoothed analysis, they proved that the squared distance can be bounded by $O(\varepsilon d^{13/2})$.

Their proof works by defining a differential equation that continuously nudges a frame in order to simultaneously correct the $\sum v_i v_i^T = I$ and $\|v_i\|^2 = \frac{d}{n}$ conditions. The flow does not have an obvious interpretation as a gradient descent. They show that naively applying this flow amasses total squared distance $O(\varepsilon d^2 n)$. However, by first perturbing the vectors randomly, the total squared distance loses its dependence on n and goes down to $O(\varepsilon d^{13/2})$.

1.1.2 A connection to convex optimization

The space of possible vector-nudging strategies is dauntingly large. We will follow the truism ‘restrictions breed creativity’ and restrict ourselves to a simple strategy: pick a linear transformation A and then move each vector v_i to $\sqrt{d/n}\frac{Av_i}{\|Av_i\|}$. Now our task is twofold: (a) prove that it is always possible to make an equal-norm Parseval frame this way, and (b) prove that if A is chosen well, then the total squared distance is

[1] Or to word it confusingly: is a nearly equal norm Parseval frame always nearly an equal norm Parseval frame?

small. For our first task, we will appeal to a connection between the Paulsen problem and convex optimization first discovered by Barthe.

Problem 2. *Given vectors $v_1, \dots, v_n$, find a square matrix A such that the scaled unit vectors $\widehat{Av_1}, \dots, \widehat{Av_n}$ are Parseval.*

Barthe [5] analyzed this problem and found an unexpected relationship between A and the solution to a convex program. Hardt and Moitra [30] also studied this convex optimization problem, giving necessary and sufficient criteria for its solvability, as well as proving a strong convexity condition that implies an efficient solution.[2] It turns out the problem can always be solved as long as not too many of the vectors v_i lie in a low-dimensional subspace.

1.1.3 Our contribution

The first resolution of the Paulsen problem, Kwok et al's tour-de-force paper, was 104 pages long and highly complex. Our contribution is a dramatically simpler proof, that also yields a much better bound and comes with a simple efficient algorithm.

Theorem 3. *For each ε-nearly equal norm Parseval frame $v_1, v_2, \dots, v_n \in \mathbb{R}^d$, there exists an exactly equal norm Parseval frame $w_1, w_2, \dots, w_n$ with total squared distance $\sum \|v_i - w_i\|^2 = O(\varepsilon d^2)$.*

Specifically: let $v_1, v_2, \dots, v_n \in \mathbb{R}^d$ be an ε-nearly equal norm Parseval frame with vectors in general position[3]. Then using Barthe's work, one can efficiently find a positive semidefinite matrix A such that $\sqrt{\frac{d}{n}}\widehat{Av_1}, \dots, \sqrt{\frac{d}{n}}\widehat{Av_n}$ is an equal-norm Parseval frame. We prove in Chapter 2 that the total squared distance $\sum \left\|v_i - \sqrt{\frac{d}{n}}\widehat{Av_i}\right\|^2$ is at most $O(\varepsilon d^2)$.

It is easy to construct examples requiring $O(\varepsilon d)$ total squared distance [17]. It would be interesting to close the gap between $O(\varepsilon d)$ and $O(\varepsilon d^2)$.

[2] Barthe's paper as well as Hardt and Moitra's use the term "radial isotropic position" rather than "Parseval frame." Vectors are said to be in radial isotropic position iff their scaled unit vectors are Parseval.

[3] Vectors not in general position are no obstruction. Just randomly nudge them by an arbitrarily small amount before applying this theorem.

1.2 Convex optimization in curved geometries

1.2.1 A generalization of our Paulsen strategy

The operator scaling problem, defined by Gurvits [29], is a multidimensional generalization of our linear-transformation-based strategy for handling the Paulsen problem.

Problem 4 (Operator Scaling)**.** *Given matrices $V_1, \ldots, V_n : \mathbb{R}^d \to \mathbb{R}^n$ where $d > n$, find matrices $S : \mathbb{R}^n \to \mathbb{R}^n$ and $A : \mathbb{R}^d \to \mathbb{R}^d$ so that, setting $W_i = SV_iA$, we have*

$$\sum W_i W_i^T = I_n \quad \text{and} \quad \sum W_i^T W_i = \frac{m}{d} I_d.$$

The two equality conditions here should spark déjà vu back to the definition of an equal-norm Parseval frame. Kwok et al [40] noticed that Problem 2, our linear-transformation-based strategy for the Paulsen problem, can be reduced to operator scaling. Say we want to solve Problem 2 for a certain list of vectors $v_1, \ldots, v_n \in \mathbb{R}^d$. Construct an equivalent instance of operator scaling by defining each V_i as the matrix whose i^{th} column is v_i and whose other columns are zero. Why is this equivalent? Well, let (S, A) be the solution to this operator scaling problem. The matrix A is the solution to the original instance of Problem 2, taking the role of transforming each vector v_i via the same linear transformation. Then, S scales each resulting vector to have unit norm. The two equality conditions in the operator scaling problem mandate the unit norm and Parseval conditions respectively.

The fastest known algorithm for operator scaling uses convex optimization [3]. However, rather than the typical convex optimization in Euclidean space, this algorithm optimizes a *geodesically convex* function defined over a specific space of PSD matrices.

1.3 Geodesically convex optimization

The first and most fundamental question about convex optimization is: what is the fastest way to do it? In this work we consider *first-order black-box* convex optimization. This means that there is an unknown function f promised to be convex, and an algorithm can query any point x to learn $f(x)$ and $\nabla f(x)$. The goal of the algorithm is to find a point x such that $f(x) - f(x^\star) < \varepsilon$, where $x^\star$ denotes the minimizer of f.

In 1983, Nesterov solved this problem for Euclidean space with his breakthrough discovery of accelerated gradient descent [43]. This technique is also known as adding a momentum term. For an α-smooth and β-strongly convex function f whose minimum is distance r from the origin, the task of finding a point x satisfying $f(x) - f(x^\star) < \varepsilon$ takes $O(\beta/\alpha \cdot \log(r/\varepsilon))$ queries for ordinary gradient descent. But accelerated gradient descent needs just $O(\sqrt{\beta/\alpha} \cdot \log(r/\varepsilon))$ queries, a quadratic improvement.

Nesterov's method is provably optimal [43], thus completing our understanding in Euclidean space. But what about in curved spaces?

In other manifolds, the natural analogue of 'convex' is called 'geodesically convex':

Definition 2. A function $f : M \to \mathbb{R}$ is geodesically convex if it is convex along every geodesic. In other words, whenever $x(t) : \mathbb{R} \to M$ parametrizes a constant-speed geodesic, the function $f(x(t))$ must be convex.

In Euclidean space, a function is convex iff its Hessian (i.e. its matrix of second derivatives) is everywhere semipositive definite. The properties α-smooth and β-strongly convex are, likewise, equivalent to the Hessian having all eigenvalues at most α and at least β. Pleasingly, this turns out to be true in curved manifolds as well [55, 1, 54]. To define the Hessian in curved geometries, we use the property that all manifolds look like Euclidean space if you zoom in far enough. Formally, given an n-dimensional manifold M and a point $x \in M$, the *exponential map*, $exp_x : \mathbb{R}^n \to M$, maps x to the origin, maps straight lines through the origin to geodesics through x, and preserves angles at x. Pulling a function $f : M \to \mathbb{R}$ back through the exponential map allows us to compute its derivatives.

Optimization in curved spaces sees application in the fastest known algorithms for computing Brascamp-Lieb constants [28, 3] and solving related problems like the null cone problem [12, 10, 11]. In machine learning, it arises in matrix completion

[13, 51, 57], dictionary learning [19, 49], robust subspace recovery [65], mixture models [32] and optimization under orthogonality constraints [24]. In statistics, some basic problems like estimating the shape of an elliptical distribution [61, 26] or estimating matrix normal models [52, 4] are best viewed through the lens of geodesic convexity.

In recent years there has been significant effort to adapt the key tools and ideas in convex optimization to the Riemannian setting. This includes giving new deterministic [63], stochastic [35, 53], variance-reduced [47, 62], projection-free [60], adaptive [34] and saddle-point escaping [20, 50] first-order methods. Many new ingredients are needed because the traditional analyses in the Euclidean setting rely on the linear structure. Still, one of the key challenges has remained elusive thus far:

> *Is there a Nesterov-like accelerated gradient method for geodesically convex functions on a Riemannian manifold?*

This question is particularly natural in settings where the curvature is non-positive, since it inherits many useful properties of Euclidean space such as having unique geodesics between any pair of points. There has been notable partial progress. Zhang and Sra [64] were among the first to clearly articulate this question. They gave a method that achieves Nesterov-like acceleration *if you start sufficiently close to the optimum.* Since then, the aim has been to develop methods that achieve *global* acceleration. Ahn and Sra [2] gave a partial answer by giving a method that converges strictly faster than gradient descent and eventually accelerates when close enough to the optimum. Martínez-Rubio [41] gave a method that achieves global acceleration but at the expense of having hidden constants that depend exponentially on the diameter of the space.

1.3.1 Our contribution

We prove that under a realistic noise assumption, acceleration is impossible even in the simplest of settings where we want to minimize a distance squared function in the hyperbolic plane. Our proof assumes that the gradient oracle returns an answer that has just an exponentially small amount of noise. In comparison, in the Euclidean setting it is possible to achieve Nesterov-like acceleration with an inverse polynomial amount of noise [23].

Theorem 5. *Given access to a δ-noisy gradient oracle, any algorithm for finding a point within distance $r/5$ of the minimum of a 1-strongly convex and $O(r)$-smooth function in the hyperbolic plane that succeeds with probability at least $2/3$ must make at least*

$$\Omega\left(\frac{r}{\log r + \log 1/\delta}\right)$$

queries in expectation. Here r is a bound on how far the optimum is from the origin.

An accelerated method would require just $O(\sqrt{r}\log(r))$ queries. See Chapter 3 for the precise definition of δ-noisy and full version of the theorem.

Remark. While it may at first seem like a limitation to restrict to functions whose condition number depends on the radius, we show in Appendix A that in the hyperbolic plane this is inevitable in the sense that *every* geodesically convex function has a condition number that is at least linear in the radius.

The key intuition is short and simple: *In negatively curved spaces, the volume of a ball grows so fast that information about the past gradients is not useful in the future.* Indeed for discrete approximations to the hyperbolic plane, it is not hard to make this intuition precise. Take an infinite regular binary tree – a shape which embeds isometrically into the hyperbolic plane. Suppose we are optimizing the distance-squared function, starting from an unknown point in this tree distance r away from the optimum. Each query can only tell us how far we are from the optimum and what direction along the tree to go. This is only $O(\log r)$ bits of information, so we need $O(r/\log r)$ queries. Of course in the actual hyperbolic plane, the direction of the gradient holds much more information than just one of the three possible directions along a tree. This is roughly why we require the noise hypothesis for our result. This intuition also helps clarify why existing acceleration results need to assume that you are already within a constant neighborhood of the optimum or depend badly on the radius.

Subsequently, Criscitiello and Boumal [20] improved upon our result by discarding the noise hypothesis. They replace the noise obstacle with a sequence of bump functions, thereby preventing any deterministic algorithm from achieving acceleration in a negatively curved space.

1.4 Graphical models

Our final example of the application and limits of convex optimization comes from graphical models, also known as Markov random fields. These are a popular model for defining high-dimensional distributions of correlated discrete random variables. Graphical models are named as such because they use a graph to encode conditional dependencies among a collection of random variables. More precisely, the distribution is described by an undirected graph $G = (V, E)$ where to each of the n nodes $u \in V$ we associate a random variable X_u which takes on one of k_u different states.

The first graphical model was physics' Ising model [9]. There the n nodes represent particles which can each be either spin-up or spin-down. The potential energy of a state $x = (x_1, x_2, \ldots, x_n)$ of such a particle system, which we will denote by $H_\theta(x)$,[4] depends both on individual particles and on pairwise interactions between them. Because of this, $H_\theta(x)$ can be written in the form

$$H_\theta(x) := -\sum_i \theta^i(x_i) - \sum_{i_1,i_2} \theta^{i_1,i_2}(x_i, x_2).$$

Here the θ^i and θ^{i_1,i_2} are functions taking as input the state at node i (respectively, at nodes i_1, i_2) and outputting a contribution to the potential energy. The underlying graph G of this model has an edge between two nodes i_1, i_2 if these particles interact, i.e. if θ^{i_1,i_2} is not identically zero.

In physics, Ising models follow the Boltzmann distribution: the probability of the system being in state x is proportional to $\exp(-H_\theta(x))$.[5] A crucial property of this distribution is that for any particle u, if we observe the states of its neighbors $N(u)$, then it is not possible to gain even more information about the state of u by observing any other particles except for u itself [36]. In other words, letting $N(u)$ denote the set of u's neighbors and X the distribution of the system, the conditional mutual information $I(X_u; X_{G\setminus N(u)\cup\{u\}}|X_{N(u)})$ is zero. We will call this the *neighbor property.*

Graphical models generalize Ising models in two ways. They allow nodes to have

[4] H is for Hamiltonian. (No relation.)

[5] In physics there is also a temperature term, but we will disregard this.

more than two states, and they allow more than pairwise interactions. The potential energy function of a graphical model is given by

$$H_\theta(x) = -\sum_{i_1<i_2<\cdots<i_\ell} \theta^{i_1 i_2 \ldots i_\ell}(x_{i_1}, x_{i_2}, \ldots, x_{i_\ell}).$$

Here the $\theta^{i_1 i_2 \ldots i_\ell}$ are functions $[k_{i_1}] \times \cdots \times [k_{i_\ell}] \to \mathbb{R}$ taking as input the states at nodes $i_1, \ldots, i_\ell$. The $\theta^{i_1 i_2 \ldots i_\ell}$ must be identically zero if these nodes are not a clique in the underlying graph G. As before, the probability of a state x is proportional to $\exp(-H_\theta(x))$, so

$$\mathbf{Pr}(\vec{X} = x_1, x_2, \ldots, x_n) = \exp\left(\sum_{i_1<i_2<\cdots<i_\ell} \theta^{i_1 i_2 \ldots i_\ell}(x_{i_1}, x_{i_2}, \ldots, x_{i_\ell}) - C\right),$$

where C is a constant to normalize the total probability to 1. It turns out that *every* distribution of system states of G with the neighbor property can be written in this form, as long as every configuration has positive probability. [36]

In this paper, we will be primarily concerned with the *structure learning problem*. Given samples from a graphical model, our goal is to learn the underlying graph G with high probability.

1.4.1 Historical progress

It turns out the difficulty of structure learning hinges on the maximum degree of G. Per Santhanam [46], when a graphical model's underlying graph has n nodes and maximum degree d, structure learning requires a number of samples that scales like $\exp(d)\log(n)$. So recovering the graph requires time at least $\approx \exp(d) \cdot n\log(n)$.

Bresler, Mossel, and Sly were the first to give a learning algorithm for general graphical models [8]. Their idea is that, using the neighbor property, one can verify a guess for a node u's neighborhood. Unfortunately, although their method is near-optimal in sample complexity, running their algorithm requires exhaustively searching over all possible neighborhoods for each node, which takes time $O(n^d)$.

In a 2014 breakthrough specific to the Ising model, Bresler [7] replaced the exhaustive search with a greedy algorithm. For each node u, his algorithm grows a candidate neighborhood S so as to minimize the empirical $I(X_u; X_{G\setminus S\cup\{u\}}; X_S)$. Superfluous vertices may get added to S, but they are pruned in the final step. This algorithm remains near-optimal in sample complexity, and brings the running time down to a comfortable $O(n^2 \log n)$, albeit with doubly-exponential dependence on the maximum degree d.

In 2016, three different groups made simultaneous progress on the problem.

Vuffray et al [58] unveiled a totally different approach. Whereas all previous attacks on structural learning were combinatorial, their paper relied on analysis and convex optimization. They found a convex function f whose minimizer is related to the energy parameter θ. Their method improves the dependence on d to singly-exponential, at the cost of raising the dependence on n to $O(n^4)$ (though they conjecture $O(n^2)$ is possible).

Simultaneously, Kilvans and Meka [37] invented a new structure learning algorithm using a sparse multiplicative weight update algorithm. Their method works for either of the two generalizations of Ising model – non-binary states, or r-wise interactions – but not (yet) both at once. Their algorithm has running time just $\tilde{O}(n^2)$ for pairwise interactions and $n^O(r)$ for r-wise interactions, with singly-exponential dependence on d. They can also replace the bounded-degree requirement with a bound on the ℓ_1 norm of each node's interactions.

Also simultaneously, Koehler, Moitra, and I generalized Bresler's method to arbitrary graphical models. In Chapter 4, we will show how, given an arbitrary n-node graphical model with maximum degree d and at most r-wise interactions, we can reconstruct its graph in time $O(n^r)$, with doubly-exponential dependence on d. (Recall that for the Ising model there are only pairwise interactions so there $r = 2$.) We will also showcase the limits of the convex optimization approach. The existence of Vuffray et al's convex function f is a minor miracle, but by the same token their method is brittle to minor perturbations of the problem. For example, if one can only observe samples of the graphical model with a random 1/2 of the nodes revealed, then our combinatorial approach works with no change, whereas convex optimization collapses.

Chapter 2

The Paulsen problem

2.1 The convex program

In this section, we will showcase the connection Barthe found between Parseval frames and convex optimization [5].

Theorem 6. *Given vectors $v_1, \ldots, v_n$ in general position (i.e. no $k+1$ live in any k-dimensional subspace), there is a positive semidefinite linear transformation A so that the scaled unit vectors $\widehat{Av_1}, \ldots, \widehat{Av_n}$ form a Parseval frame.*

Theorem 7. *Given vectors $v_1, \ldots, v_n \in \mathbb{R}^d$ in general position, the function*

$$f(t_1, t_2, \ldots, t_n) = \log\det\left(\sum e^{t_i} v_i v_i^T\right) - \frac{d}{n}\sum t_i$$

is convex, attains a minimum, and furthermore, at any t attaining the minimum, the positive semidefinite linear transformation

$$A = \left(\sum e^{t_i} v_i v_i^T\right)^{-1/2}$$

makes the vectors $\widehat{Av_1}, \ldots, \widehat{Av_n}$ a Parseval frame.

That the function is convex can be deduced from known properties of logdet. That it attains a minimum is, perhaps surprisingly, not straightforward, and depends crucially on the vectors being in general position. Hardt and Moitra [30] show that f fails to attain a minimum iff too many of the v_i are contained within a subspace. They use this to present an algorithm to determine, given a set of vectors, whether many of them live in the same subspace. They also prove a strong convexity condition implying that solving the convex program is efficient.

As for the final claim – that A puts the vectors in radial isotropic position – I will sketch Barthe's proof.

Proof. Let t minimize

$$f(t_1, t_2, \ldots, t_n) = \log\det\left(\sum e^{t_i} v_i v_i^T\right) - \frac{d}{n}\sum t_i.$$

Then $\nabla f(t) = 0$. Using the fact that the derivative of $\log\det X$ is $(X^{-1})^T$, this means

$$\operatorname{Tr}\left(\left(\sum e^{t_i} v_i v_i^T\right)^{-1} e^{t_i} v_i v_i^T\right) - d/n = 0 \quad \text{for all } i.$$

Now set $A = \left(\sum e^{t_i} v_i v_i^T\right)^{-1/2}$ as per the theorem. Note that A is positive semidefinite. The gradient condition becomes

$$\operatorname{Tr}\left(A^2 e^{t_i} v_i v_i^T\right) - d/n = 0.$$

By rearranging this equality, we can pinpoint $\|Av_i\|$:

$$\begin{aligned} d/n &= \operatorname{Tr}\left(A^2 e^{t_i} v_i v_i^T\right) \\ &= e^{t_i} \operatorname{Tr}\left(v_i^T A^2 v_i\right) \quad \text{(using the identity } \operatorname{Tr}(XYZ) = \operatorname{Tr}(YZX)) \qquad (2.1) \\ &= e^{t_i} \|Av_i\|^2 \end{aligned}$$

Now since we know $\|Av_i\|$, we can tackle the Parseval condition. This is the isotropy condition we want to prove:

$$\sum \frac{Av_i(Av_i)^T}{\|Av_i\|^2} =^? \frac{n}{d} I.$$

Rearrange, using the fact that A is positive semidefinite:

$$\begin{aligned} \sum \frac{Ae^{t_i} v_i v_i^T A}{d/n} &=^? \frac{n}{d} I \\ \sum e^{t_i} v_i v_i^T &=^? A^{-2} \\ \left(\sum e^{t_i} v_i v_i^T \right)^{-1/2} &=^? A \end{aligned} \tag{2.2}$$

And this is exactly our definition of A, so the equality is true. □

2.2 Resolving the Paulsen problem

With this machinery in place, our algorithm to solve the Paulsen problem is not hard to guess:

Input vectors $v_1, v_2, \ldots, v_n \in \mathbb{R}^d$ which form an ε-nearly equal norm Parseval frame.

Apply small perturbations η_i so that $v_1 + \eta_1, \ldots, v_n + \eta_n$ are in general position. (If they were already in general position, we can just set $\eta_i = 0$.) Set $u_1, u_2, \ldots, u_n$ as the resulting vectors scaled to norm $\sqrt{d/n}$.

Finally, find a positive semidefinite linear transformation A using theorem 6, and output the equal-norm Parseval frame $w_1, \ldots, w_n := \sqrt{\frac{d}{n}} \widehat{Au_1}, \ldots, \sqrt{\frac{d}{n}} \widehat{Au_n}$.

Theorem 8. *This algorithm yields an equal-norm Parseval frame $w_1, \ldots, w_n$ such that the total squared distance $\sum \|v_i - w_i\|^2$ is at most $14\varepsilon d^2 = O(\varepsilon d^2)$.*

To prove this, we will first show that we did not lose much by going from v_i to u_i: that is, $\sum \|v_i - u_i\|^2$ is small, and that the u_i are a 4ε-nearly equal norm Parseval frame. This is a relatively straightforward calculation, and as we go along we will quantify how small we need the perturbations to be. The meat of the proof is Lemma 1, where we prove that $\sum \|u_i - w_i\|^2 = O(\varepsilon d^2)$.

Proof. Let $V = v_1, v_2, \ldots, v_n$, and similarly for U and W. First we want to bound the squared distance between V and U. We can upper bound

$$\|v_i - u_i\|^2 \leq \Big(\sqrt{\frac{d}{n}} - \sqrt{(1-\varepsilon)\frac{d}{n}}\Big)^2 + \gamma \leq \frac{\varepsilon d}{n},$$

where $\gamma \leq \|\eta_i\|^2 + 2\|\eta_i\|$ is a term that depends on the perturbation and if $\gamma \leq \frac{(1-\sqrt{1-\varepsilon})\varepsilon d}{n}$ then the last inequality holds. Now summing over all pairs of vectors we get that $\|V - U\|^2 \leq \varepsilon d$.

Next we observe that the vectors in U are still a nearly equal norm Parseval frame. After adding the perturbations, if $\|\eta_i\| \leq \sqrt{d/n} \cdot \frac{\varepsilon}{2}$ for each i, then each vector v_i was scaled by a factor between $\sqrt{1-2\varepsilon}$ and $\sqrt{1+2\varepsilon}$. Also if we take $\|\eta_i\| \leq \frac{\varepsilon}{2n}$ for each i, then we conclude that the vectors in U are a 4ε-nearly equal norm Parseval frame.

What remains is to bound the total squared distance between U and W. In Lemma 1, we show that $\|U - W\|^2 \leq 6\varepsilon d^2$.

Finally, for any three vectors a, b and c we have the triangle-like inequality

$$\|a - c\|_2^2 \leq 2\Big(\|a - b\|_2^2 + \|b - c\|_2^2\Big).$$

This follows from the parallelogram identity which says that $\|x + y\|_2^2 + \|x - y\|_2^2 = 2\|x\|_2^2 + 2\|y\|_2^2$ for any vectors x and y . We can then substitute $x = a - b$ and $y = b - c$ and omit the $\|x - y\|_2^2$ term to get the inequality above. In any case when we apply this for all triples of vectors v_i, u_i and w_i we have

$$\|V - W\|^2 \leq 2\,\|V - U\|^2 + 2\,\|U - W\|^2 \leq 14\varepsilon d^2.$$

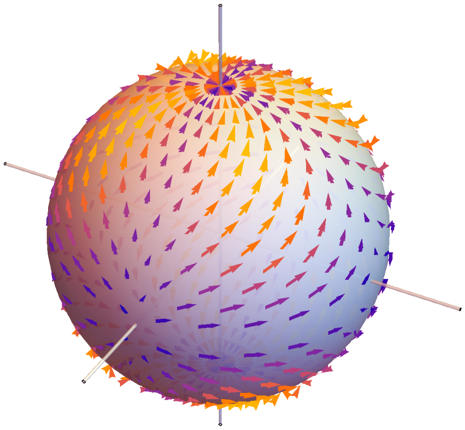
This completes the proof. □

Lemma 1. *With U, A and W as defined*[1] *in the proof of Theorem 8, we have that $\|U - W\|^2 \leq 6\varepsilon d^2$.*

Let us first motivate the proof. The mapping from U to W, namely $u_i \to \sqrt{\frac{d}{n}}\widehat{Au_i}$,

[1] To be precise, U is a 4ε-nearly equal-norm Parseval frame; each u_i has norm $\sqrt{d/n}$; and A is a positive semidefinite matrix such that the vectors $w_i = \sqrt{\frac{d}{n}}\widehat{Au_i}$ form an equal-norm Parseval frame.

is a function from the d-dimensional radius-$\sqrt{\frac{d}{n}}$ sphere to itself. As A is positive semidefinite, we can work in an orthonormal eigenbasis $e_1, e_2, \ldots, e_d$. In this basis, the mapping from U to W should pull vectors towards the $\pm e_j$ with larger eigenvalues, and push vectors away from the $\pm e_j$ with smaller eigenvalues.



Visualization of a possible mapping if e_1, e_2, e_3 are in order of biggest to smallest eigenvalue.

But the Parseval condition is equivalent to $\sum_i (w_i^T e_j)^2 = 1$ for all j. This equality almost holds for U and exactly holds for W. So intuitively, there cannot be *too* much movement towards the $\pm e_j$ with larger eigenvalues, or else $\sum_i (w_i^T e_j)^2$ would overshoot its target of 1. So, since all the movement is in the same direction – away from smaller eigenvalues and towards larger ones – there cannot be too much movement overall.

Without further ado, the proof.

Proof. First we introduce a notion of majorization:

Definition 3. For d element sequences x and y we say $x \succeq y$ if for all $1 \leq j \leq d$ we have $\sum_{i=1}^{j} x_i \geq \sum_{i=1}^{j} y_i$ and moreover $\sum_{i=1}^{d} x_i = \sum_{i=1}^{d} y_i$.

Next we introduce a notion of distance, similar to the Wasserstein distance, but for vectors that are not necessarily nonnegative:

Definition 4. If $x \succeq y$, we define

$$\mathcal{T}(x,y) := \sum_{j=1}^{d} j(y_j - x_j).$$

The following alternative definition is equivalent and will be convenient for us:

Fact 9. *If $x \succeq y$ then*

$$\mathcal{T}(x,y) = \sum_{j=1}^{d} \sum_{i=1}^{j} x_i - y_i.$$

Proof. Since by assumption $\sum_{j=1}^{d} x_j = \sum_{j=1}^{d} y_j$ we can write

$$\mathcal{T}(x,y) = \sum_{j=1}^{d} j(y_j - x_j) + \sum_{j=1}^{d} (d+1)(x_j - y_j) = \sum_{j=1}^{d} (d+1-j)(x_j - y_j) = \sum_{j=1}^{d} \sum_{i=1}^{j} x_i - y_i$$

which completes the proof. □

Next we relate $\mathcal{T}(x,y)$ and $\|x-y\|_1$ specifically when $x \succeq y$. What makes this bound subtle is that the vector $x-y$ can (and will) have negative entries. What will make $\mathcal{T}$ easier to work with is that it, unlike $\|x-y\|_1$, it is *linear.*

Lemma 2. *If $x \succeq y$ then $\frac{\|x-y\|_1}{2} \leq \mathcal{T}(x,y)$.*

Before we proceed to the proof of the above lemma let us introduce a kind of Wasserstein distance $\mathcal{W}(x,y)$ on the integers from 1 to d. It will not exactly be the usual definition because we allow x and y to have negative entries and so there is no way to interpret them as a distribution. Nevertheless we define:

Definition 5.

$$\mathcal{W}(x,y) := \sup_f \sum_{j=1}^{d} f(j)(x_j - y_j) \text{ s.t. } f : \{1,2,\cdots,d\} \to \mathbb{R} \text{ and } |f(i) - f(j)| \leq |i-j| \text{ for all } i,j$$

Next we give an alternative characterization of $\mathcal{W}(x,y)$:

Fact 10. *If* $\sum_{j=1}^{d} x_j = \sum_{j=1}^{d} y_j$ *then*

$$\mathcal{W}(x,y) = \sum_{j=1}^{d} \Big| \Big(\sum_{i=1}^{j} x_i \Big) - \Big(\sum_{i=1}^{j} y_i \Big) \Big|.$$

Proof. First, since $\sum_{j=1}^{d} x_j = \sum_{j=1}^{d} y_j$ we can assume without loss of generality that $f(d) = 0$. Now we can rewrite f in terms of its increments, starting from $f(d)$, as $f(j) = \sum_{i=i}^{d-1} \delta_i$ where each δ_i must be between -1 and 1. It now follows that

$$\sum_{j=1}^{d} f(j)(x_j - y_j) = \sum_{i=1}^{d-1} \delta_i \sum_{j=1}^{i} x_j - y_j$$

and the optimal choice for each δ_i is to set it to the sign of $\sum_{j=1}^{i} x_j - y_j$. From this it follows that

$$\mathcal{W}(x,y) = \sum_{j=1}^{d} \Big| \Big(\sum_{i=1}^{j} x_i \Big) - \Big(\sum_{i=1}^{j} y_i \Big) \Big|$$

which completes the proof. □

Now we are ready to prove Lemma 2:

Proof. Using Fact 9 we have

$$\mathcal{T}(x,y) = \sum_{j=1}^{d} \Big(\sum_{i=1}^{j} x_i \Big) - \Big(\sum_{i=1}^{j} y_i \Big) = \sum_{j=1}^{d} \Big| \Big(\sum_{i=1}^{j} x_i \Big) - \Big(\sum_{i=1}^{j} y_i \Big) \Big| = \mathcal{W}(x,y)$$

where the second equality follows from the assumption $x \succeq y$ and the last equality follows from Fact 10. Finally, in Definition 5 we can set $f(j)$ to be the sign of $x_j - y_j$ divided by two. This now implies $\mathcal{W}(x,y) \geq \frac{\|x-y\|_1}{2}$ which completes the proof. □

Lastly we define some useful sequences of helper vectors. Throughout, we will work in an orthonormal basis where A is diagonal and its diagonal entries are sorted $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$. Such a basis exists since A is positive semidefinite. In this basis, let $u_i^{\circ 2}$ denote the result of entrywise squaring u_i and define $w_i^{\circ 2}$ similarly. By construction, for each i, the sums of the entries in $u_i^{\circ 2}$ and in $w_i^{\circ 2}$ are both d/n.

Additionally, we prove in Claim 1 that $w_i^{\circ 2} \succeq u_i^{\circ 2}$ for each i. Now we have

$$\|U - W\|^2 = \sum_{i=1}^{n}\sum_{j=1}^{d}\Big((u_i)_j - (w_i)_j\Big)^2 \le \sum_{i=1}^{n}\sum_{j=1}^{d}\Big|(u_i)_j^2 - (w_i)_j^2\Big| = \sum_{i=1}^{n}\|u_i^{\circ 2} - w_i^{\circ 2}\|_1$$

where the first inequality follows because for any real values a and b with the same sign we have $(a-b)^2 \le |a^2-b^2|$. Applying Lemma 2 along with further manipulations gives

$$\|U - W\|^2 \le 2\sum_{i=1}^{n}\mathcal{T}(w_i^{\circ 2}, u_i^{\circ 2}) = 2\mathcal{T}\Big(\sum_{i=1}^{n} w_i^{\circ 2}, \sum_{i=1}^{n} u_i^{\circ 2}\Big) = 2\sum_{j=1}^{d} j\Big(\sum_{i=1}^{n}((u_i)_j)^2 - ((w_i)_j)^2\Big).$$

Since W is Parseval, we have that $\sum_{i=1}^{n}((w_i)_j)^2 = 1$ for all j. And because U is 4ε-nearly Parseval, we have $\sum_{i=1}^{n}((u_i)_j)^2 \le 1 + 4\varepsilon$ which gives

$$\|U - W\|^2 \le 2\sum_{j=1}^{d} j \cdot 4\varepsilon = 4\varepsilon d(d+1) \le 6\varepsilon d^2$$

completing the proof. □

Claim 1. *With $u_i^{\circ 2}$ and $w_i^{\circ 2}$ as defined in Lemma 1, we have $w_i^{\circ 2} \succeq u_i^{\circ 2}$.*

Proof. Recall that by construction, $u_i^{\circ 2}$ and $w_i^{\circ 2}$ are nonnegative and the sum of their entries is the same. Thus showing that for any $1 \le j < d$, the inequality $\sum_{k=1}^{j}(w_i^{\circ 2})_k \ge \sum_{k=1}^{j}(u_i^{\circ 2})_k$ holds follows from showing instead that

$$\frac{\sum_{k=1}^{j}(w_i^{\circ 2})_k}{\sum_{k=j+1}^{d}(w_i^{\circ 2})_k} \ge \frac{\sum_{k=1}^{j}(u_i^{\circ 2})_k}{\sum_{k=j+1}^{d}(u_i^{\circ 2})_k}.$$

Now to complete the proof we observe that

$$\begin{aligned}\frac{\sum_{k=1}^{j}(w_i^{\circ 2})_k}{\sum_{k=j+1}^{d}(w_i^{\circ 2})_k} &= \frac{\sum_{k=1}^{j}\lambda_k^2((w_i)_k)^2}{\sum_{k=j+1}^{d}\lambda_k^2((w_i)_k)^2} \\ &\ge \frac{\sum_{k=1}^{j}\lambda_j^2((u_i)_k)^2}{\sum_{k=j+1}^{d}\lambda_j^2((u_i)_k)^2} = \frac{\sum_{k=1}^{j}((u_i)_k)^2}{\sum_{k=j+1}^{d}((u_i)_k)^2} = \frac{\sum_{k=1}^{j}(u_i^{\circ 2})_k}{\sum_{k=j+1}^{d}(u_i^{\circ 2})_k}\end{aligned}$$

where the inequality follows from the assumption $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_d \ge 0$. □

2.3 An Algorithm for the Paulsen Problem

Every step of the proof of Theorem 8 is straightforward to implement algorithmically, except for the step where we compute the transformation A that places the set of vectors U in radial isotropic position. Fortunately, Hardt and Moitra [30] gave an algorithm for computing A. We will state a special case of their main theorem, which is sufficient for our purposes.

Theorem 11. *[30] Let $\delta > 0$ and $\alpha > 0$. Suppose $U = u_1, u_2, \ldots, u_n \in \mathbb{R}^d$ has the property that every set of d vectors are linearly independent. Then there is an algorithm to find a linear transformation A so that, setting $w_i = \sqrt{\frac{d}{n}}\widehat{Au_i}$ as usual, we have*

$$\sum_{i=1}^{n} w_i w_i^T = I + J$$

where $\|J\|_\infty \leq \delta$ — i.e. the largest entry of J in absolute value is at most δ. The running time is polynomial in $1/\alpha$, $\log 1/\delta$ and L where L is an upper bound on the bit complexity of U.

By combining their algorithm with our proof of Theorem 8 we get:

Corollary 12. *Suppose $V = v_1, v_2, \ldots, v_n \in \mathbb{R}^d$ is an ε-nearly equal norm Parseval frame. Furthermore suppose $n > d$. Then given $\gamma > 0$, there is an algorithm to compute a γ-nearly equal norm Parseval frame W with*

$$\|V - W\|^2 \leq 14\varepsilon d^2$$

whose running time is polynomial in $\log 1/\gamma$ and L where L is an upper bound on the bit complexity of V.

Proof. We perturb V as in the proof of Theorem 8 and run the algorithm in Theorem 11 on U with $\delta = \frac{\gamma\varepsilon}{d^3}$ so that the output is a $\frac{\gamma\varepsilon}{d}$-nearly equal norm Parseval frame. Note that our bound on the squared distance between V and W in Lemma 1 used the fact that W was an equal norm Parseval frame. But it is easy to see that the slack in the bounds we used can accommodate a $\frac{\gamma\varepsilon}{d}$–nearly equal norm Parseval frame instead. $\square$

Chapter 3

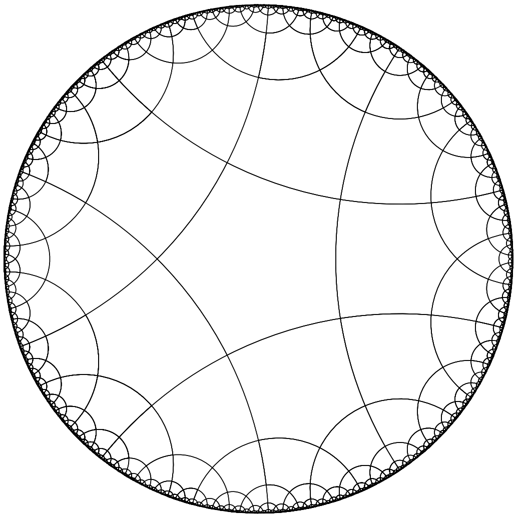
There is no acceleration in the hyperbolic plane

3.1 The Hyperbolic Plane

This section serves as an introduction to the hyperbolic plane $\mathbb{H}^2$, establishing the important facts we use for our proof as well as intuition for our main result. The first and most important fact about the hyperbolic plane is that it is very large:

Fact 13. *[15] The circumference and area of a hyperbolic circle are both exponential in its radius.*

For illustration, consider a tiling of the hyperbolic plane with congruent equilateral pentagons. As you can see, the number of pentagons at distance r from the origin grows exponentially in r.



This gives intuition for our result. If you are attempting to minimize a function whose minimum lies somewhere within a ball of radius r, the hyperbolic plane forces you to search over a much larger area than any fixed dimension of Euclidean space would. This inherently makes it harder to exploit information from past queries about the function value and gradient, when you are interested in reaching a point far away from the origin.

3.1.1 Why Pirates Don't Search for Treasure in the Hyperbolic Plane

The purpose of this subsection is to provide intuition for our main theorem and is not required to understand our results. Imagine a pirate who has buried treasure in the desert somewhere at distance 100 away from her. She does not remember exactly where the treasure is, but is in possession of a compass which points towards it. The compass' reading has error, though: on the order of 10^{-16} degrees. (This setting is analogous to an algorithm able to make noisy queries to the gradient of some function.) In the Euclidean plane, the pirate could easily find the treasure: take a compass bearing, walk 100 steps, and dig. An error of 10^{-16} degrees would literally be subatomic.

However, if the pirate attempted this strategy in the hyperbolic plane, she would end up at distance just over 190 from her treasure. She would have started to walk away from the treasure after just a few steps! (Specifically, a constant number of steps that scales with $\log(1/10^{-16})$.) Therefore, she would have to repeatedly look at her compass every few steps in order to make progress towards the treasure. This is why gradient descent has only linear convergence in hyperbolic spaces. Of course, this falls short of explaining why no algorithm converges faster. I am indebted to the video game HyperRogue [39] for giving intuition about the hyperbolic plane. One level of this game features a similar scenario with pirates and compasses.

3.2 Optimization with a Noisy Gradient Oracle

In this section we define the main model we will be interested in. Moreover we recall convergence bounds in the Euclidean case, particularly those that continue to hold in the presence of a small amount of noise, as a point of comparison.

Definition 6. The radius-r gradient optimization model is as follows: There is an unknown differentiable function f whose minimum is within distance r of the origin. An algorithm may query points x within distance $1000r$ of the origin.[1] Upon querying x, the algorithm learns $f(x)$ and the gradient $\nabla f(x)$.

Definition 7. In the noisy version of the model, instead of learning the exact values of $f(x)$ and $\nabla f(x)$, the algorithm receives $f(x)+z_1$ and $\nabla f(x)+z_2$ where z_1 and z_2 are noise. We do not require the noise to be of a specific form such as Gaussian, uniform, etc – we only require that the noise terms for different queries are independent.

Definition 8. We say noise is *c-non-concentrated* if on any query, the probability distribution function of the noise is everywhere bounded above by c. We say noise is *C-precise* if the noise term never has magnitude larger than C.

[1] The number 1000 is arbitrary and only affects constants inside big-O notation. The reason for this assumption is as follows: if an algorithm were to query a point x superexponentially far away from the origin and learn $f(x)$ + noise, then $f(x)$ would be so large that the noise term could be disregarded completely. Thus, removing this assumption would require a more refined noise model, such as multiplicative noise.

In Euclidean space, a small amount of noise does not preclude acceleration. As a point of comparison, we restate the key result (Theorem 7) from [23]:

Theorem 14 (from [23])**.** *There is an algorithm that, given C-precise noisy oracle access to an L-smooth μ-strongly convex function f and its gradient, along with a starting point at distance d from the minimum of f, outputs after k oracle queries a point x_k such that*

$$f(x_k) - \min f \leq O\left(Ld \cdot \min\left(1/k^2, \exp(-k/2\sqrt{\mu/L})\right) + \min\left(k \cdot \mathit{poly}(C), \sqrt{L/\mu}\right)\right).$$

(The bound in the original paper is more precise, making big-O constants explicit and using a more refined notion of precision.) This theorem implies that accelerated gradient descent works in Euclidean space even in the presence of noise, provided that the noise has magnitude at most some inverse polynomial in r. Contrast our main result: in hyperbolic space, accelerated gradient descent is impossible even with *exponentially* small noise.

3.3 The Noisy Gradient Task in the Hyperbolic Plane

In this paper we will prove lower bounds for minimizing arguably the simplest geodesically convex function, the distance squared function.

Fact 15. *In the hyperbolic plane, the distance squared function $x \mapsto d(x, x^\star)^2$ is geodesically convex and its minimum is $x^\star$. At distance r from $x^\star$, this function is 1-strongly convex and $(r/\tanh r)$-smooth.*

The strong convexity and smoothness come from the formula for its Hessian given in [25].

Without noise, an algorithm could locate $x^\star$ exactly in *one* query, because any gradient is guaranteed to both point exactly at $x^\star$ and indicate the distance to $x^\star$. What about with noisy gradients?

Within the hyperbolic disc of radius r centered at the origin, our function f is $O(r)$-smooth and 1-strongly convex. (As an aside, in Theorem 35 we show that for any β-smooth and α-strongly convex function in the hyperbolic disk we must have $\beta/\alpha = \Omega(r)$). If Nesterov-like acceleration in the hyperbolic plane were possible we should be able to locate $x^\star$ to within distance 1 in time $O(\sqrt{r})$. Unfortunately, as we will show, this task is impossible. Even worse, it is impossible to get any polynomial factor speedup in the convergence.

Theorem 16. *In the radius-r noisy gradient optimization model in the hyperbolic plane, if queries receive noisy answers with c-non-concentrated C-precise noise, then any algorithm that can find a point within distance $r/5$ of the minimum of the function that succeeds with probability at least $2/3$ must make at least*

$$\Omega\left(\frac{r}{\log r + \log C + \log c}\right)$$

queries. This is true even if the function is guaranteed to be 1-strongly convex and $O(r)$-smooth at every point within distance r from the origin.

To prove this result, we will generalize the noisy gradient model to any setting in which an agent makes queries and receives probabilistic answers over a discrete set of possibilities. In this general setting, we will prove a lower bound on the number of queries needed to determine the state of the world.

3.3.1 Noisy Query Games

Definition 9. A *noisy query game* is a tuple $(n, Q, \mathcal{X}, f)$, with which the following one-player game is played: A secret number $i^\star$ is chosen uniformly at random from $\{1, 2, \ldots, n\}$. The player's goal is to determine $i^\star$. To do so, the player may make a query $q \in Q$ and receive an observation $X \in \mathcal{X}$. The observation is sampled using some probability distribution function $f_{q,i^\star}(x)$. The player wins when they can guess $i^\star$ with probability at least $2/3$.

We remark that for us $\mathcal{X}$ will be a region in Euclidean space and we will use $|\mathcal{X}|$

to denote its volume. The noisy game broadly seems to be a natural model for a class of noisy learning tasks. In particular, it generalizes the noisy gradient task in the hyperbolic plane, as we show in the following comment:

Comment 17. Consider the noisy gradient task in which we place $n = e^{\Theta(r)}$ points equally in a circle of radius r in the hyperbolic plane, so that the points are distance $\geq r/2$ apart. (This is possible because circles are exponentially large – see [15], page 92 – so greedily picking points one at a time and removing a ball of radius $r/2$ around each runs out of volume only after exponentially many steps.) The secret number $i^\star$ corresponds to one of these points $x^\star$. Define the function $f(x) = \mathrm{dist}(x, x^\star)^2$ whose optimum is $x^\star$. The player makes queries in $Q :=$ a region in the hyperbolic plane with radius $O(r)$, and receives noisy gradient observations. Now the player can win if and only if they can locate the optimum of $f(x)$, among the discrete set of possibilities, with probability at least 2/3.

We now state our lower bound for noisy query games:

Theorem 18. *In a noisy query game $(n, Q, \mathcal{X}, f)$, suppose the noise is c-non-concentrated, i.e. all probability distribution functions $f_{q,i^\star}$ are everywhere bounded above by some constant c. Then for the player to be able to guess $i^\star$ with probability at least 2/3, the player must make at least $\Omega(\frac{\log n}{\log(c|\mathcal{X}|)})$ queries.*

Theorem 18 is difficult to prove directly, because the player's knowledge is a posterior distribution over the options $\{1, 2, \ldots, n\}$ which can change in complicated ways. To overcome this obstacle and prove Theorem 18, we will define an easier 'transparent' version of the noisy query game, where the player's knowledge is a subset of $\{1, 2, \ldots, n\}$ representing which options could possibly be the correct one. Then we will show that even in the easier version of the game, the player needs many queries in expectation to succeed.

3.3.2 The Transparent Noisy Query Game

In a noisy query game, observations are sampled from probability distribution functions $f_{q,i}$ on a space $\mathcal{X}$. One way to sample an observation from $f_{q,i}$ is to sample uniformly from the region under its graph. Let $G_{q,i}$ denote this region. Note that the volume of $G_{q,i}$ must be 1, because probabilities always sum to 1. In the transparent noisy query game, we answer a query q by telling the player a point (x, y) uniformly sampled from the graph region $G_{q,i^\star}$. In the normal query game the player only learns x, so the normal version can only be harder.

The key question is: How does the player's knowledge update when she receives an observation (x, y)? For any option i whose graph area $G_{q,i}$ does not include the point (x, y), the player learns that i cannot possibly be correct. For the rest of the options, the player learns nothing. This is because observations are sampled uniformly and each $G_{q,i}$ has unit area, so by Bayes' rule the player's posterior on $i^\star$ remains uniform over all remaining options. (Here we have assumed that the prior is uniform at the beginning.)

Indeed, this convenient property is the reason we defined this transparent version: It allows us to easily analyze the player's progress by tracking only the number of remaining possible options, rather than the messy details of what happens to the posterior distribution.

Lemma 3. *Suppose all the $f_{q,i}$ are c-non-concentrated distributions. Then in the transparent noisy query game, a query decreases the logarithm of the number of possible remaining options by at most $\log(c|\mathcal{X}|)$ in expectation.*

Proof. Let m be the number of options remaining before the query. For convenience, use the notation $N(x, y)$ for the number of graph areas $G_{q,i}$, among the m remaining options, that contain (x, y). If the player receives the query result (x, y), they would be left with $N(x, y)$ remaining options. So after the query, the expected number of options remaining is

$$\mathbb{E}_{i^\star}\left[\int\limits_{G_{q,i^\star}} \log\left(N(x,y)\right)\ dxdy\right],$$

where the expectation is taken uniformly at random from among the m remaining options. Moving the expectation inside the integral sign and using the assumption that all graph areas are contained within $\mathcal{X} \times [0, c]$, we get that the above expectation is equal to:

$$\int_{\mathcal{X}\times[0,c]} \frac{N(x,y)}{m} \log\left(N(x,y)\right) dxdy$$

Since each graph has area 1, the integral $\int_{\mathcal{X}\times[0,c]} N(x,y)$ is m. So by Jensen's inequality, subject to this restriction, the above quantity is minimized when N is constant over the entire domain $\mathcal{X} \times [0, c]$. The minimum value is

$$\int_{\mathcal{X}\times[0,c]} \frac{m/c|\mathcal{X}|}{m} \log\left(m/c|\mathcal{X}|\right) dxdy = \log\left(m/c|\mathcal{X}|\right) = \log m - \log\left(c|\mathcal{X}|\right).$$

So the expectation of the logarithm of the number of possible options left decreases by at most $\log\left(c|\mathcal{X}|\right)$, as desired. □

Now we can prove our main lower bound for the noisy query game:

Proof of Theorem 18. Let n_i denote the number of possible remaining options after i steps. Thus we have $n_0 = n$. Now let X be a random variable that represents the cumulative progress the algorithm has made. In particular let

$$X = \sum_{i=1}^{T} \log n_{i-1} - \log n_i.$$

Applying Lemma 3 and Markov's bound we have that $X \leq 3\mathbb{E}[X]$ with probability at least 2/3. If the algorithm succeeds at being able to determine $i^\star$ after T steps we must have $n_T = 1$. Putting everything together we have

$$0 = \log n_T = \log n - X \geq \log n - 3|T| \log(c|\mathcal{X}|)$$

and rearranging completes the proof. □

3.3.3 Proof of the Main Theorem

Our main result now follows easily from the machinery of noisy query games:

Proof of Theorem 16. Suppose we want to minimize the function $f(x) = \mathrm{dist}(x, x^\star)^2$. This function is 1-strongly convex and $2r$-smooth within distance r of the origin. (As mentioned earlier, [25] shows the eigenvalues of the Hessian are 1 and $r/\tanh r \leq r+1$.) First we apply the reduction in Comment 17 so that we have $n = e^{\Theta(r)}$ points with pairwise distance at least $r/2$. Moreover $x^\star$ is among them and corresponds to the secret number $i^\star$ in the noisy query game.

In the setting of Theorem 16, a player makes queries within a certain region of the hyperbolic plane, and learns the (noisy) function value and gradient at their query point. They are tasked with finding a point within distance $r/5$ of $x^\star$. Because the n points have pairwise distance at least $r/2$, doing so requires figuring out which of the n points is $x^\star$. So the player must win the query game, which by Theorem 18, takes at least $\frac{\log n}{\log(|X|c)}$ queries.

We picked $n = e^{\Theta(r)}$ above, and the value of c is stated in Theorem 16's assumptions. But what is $|\mathcal{X}|$, i.e. the volume containing all query answers? Since the player's queries are restricted to a region in the hyperbolic plane of radius $O(r)$, the true answer to their query is a function value in the interval $[0, O(r^2)]$ along with a gradient in the disk $B(0, O(r)) \subseteq \mathbb{R}^2$. (Recall that the gradient lives in $\mathbb{R}^2$, not the hyperbolic plane.) By assumption, the noise causes error at most C, so the observed query answer must lie in

$$[-C, O(r^2) + C] \times B(0, O(r) + C) \subset \mathbb{R}^3.$$

This is a compact set whose volume is a polynomial in r and C. In particular we have $|\mathcal{X}| \leq O(r^4 C^3)$. Therefore overall the player needs at least

$$\frac{\log n}{\log(|\mathcal{X}|c)} = \frac{r)}{\log(c) + O(\log r + \log C)}$$

queries. This completes the proof. □

3.3.4 Removing the noise hypothesis

In 2021, Criscitiello and Boumal [21] improved upon our result by replacing the noise hypothesis with a carefully-constructed sequence of bump functions.

Their key strategy is to answer queries so that, after every query, there are a great many smooth and strongly convex functions $f_1, \ldots, f_N$ consistent with all queries so far, yet the minima of the f_i are far apart. In their 'Pièce de résistance' lemma, they show how, given any query, one can modify each f_i by a small bump function and then adversarially return a query result consistent with many different f_i. Their construction is very technical so we will not attempt to sketch it here.

One open problem still remains. Criscitiello and Boumal proved that acceleration is impossible for any *deterministic* algorithm. This leaves open a very slim crack through which a randomized algorithm might slip. It would be satisfying to seal this crack.

Chapter 4

Learning graphical models

In 2015, Bresler [7] gave a simple greedy algorithm to learn the structure of a bounded degree Ising model. For each node u, the algorithm grows a set S of guesses for u's neighbors. Let X_S denote the random variable representing the joint state of nodes in S. Then the key fact enabling the greedy algorithm is the following:

Fact 19. *For every node u, for any set $S \subseteq V \setminus \{u\}$ that does not contain all of u's neighbors, there is a node $v \neq u$ which has non-negligible conditional mutual information (conditioned on X_S) with u.*

Bresler's algorithm uses this fact to repeatedly find a node $v \notin S$ with large correlation with u and add it to S. Since there is only so much information about u we can possibly acquire, this algorithm must halt in an amount of time depending on the bound on $I(u; v|X_S)$.

Fact 19 is simultaneously surprising and not surprising. When S contains all the neighbors of u, then X_u has zero conditional mutual information (again conditioned on X_S) with any other node because X_u only depends on X_S. Conversely shouldn't we expect that if S does not contain the entire neighborhood of u, that there is some neighbor that has nonzero conditional mutual information with u? The difficulty is that the influence of a neighbor on u can be cancelled out indirectly by the other neighbors of u. The key fact above tells us that it is impossible for the influences to all cancel out. But is this fact only true for Ising models or is it an instance of a more general phenomenon that holds for any graphical model?

4.0.1 Our Techniques

In this chapter, we give a vast generalization of Bresler's [7] lower bound on the conditional mutual information. We prove that it holds in general graphical models with higher order interactions provided that we look at sets of nodes. More precisely we prove, in a graphical model with up to r-wise interactions, the following fundamental fact:

> For every node u, for any set $S \subseteq V \setminus \{u\}$ that does not contain all of u's neighbors, there is a set I of at most $r-1$ nodes which does not contain u where X_u and X_I have non-negligible conditional mutual information (conditioned on X_S).

Remark 20. It is necessary to allow I to be a set of size $r-1$. For any integer r, there are graphical models where for every node u and every set of nodes I of size at most $r-2$, the mutual information between X_u and X_I is zero.

The starting point of our proof is a more conceptual approach to lower bounding the mutual information. Let $N(u)$ denote the neighbors of u. What makes proving such a lower bound challenging is that even though $I(X_u; X_{N(u)}) > 0$, this alone is not enough to conclude that $I(X_u; X_j) > 0$ for some $j \in N(u)$. Indeed for general distributions we can have that the mutual information between a variable and a set of variables is positive, but every pair of variables has zero mutual information. The distribution produced by a general graphical model can be quite unwieldy (e.g. it is computationally hard to sample from) so what we need is a technique to tame it to make sure these types of pathologies cannot arise.

Our approach goes through a two-player game that we call the GuessingGame between Alice and Bob. Alice samples a configuration $X_1, X_2, \dots X_n$ and reveals I and X_I for a randomly chosen set of u's neighbors with $|I| \leq r-1$. Bob's goal is to guess X_u with non-trivial advantage over its marginal distribution. We give an explicit strategy for Bob that achieves positive expected value. Our approach is quite general because we base Bob's guess on the contribution of X_I to the overall clique potentials that X_u participates in, in a way that the expectation over I yields an unbiased estimator of the total clique potential. The fact that the strategy has positive expected value is then immediate, and all that remains is to prove a quantitative lower bound on it using the law of total variance. From here, the intuition is that if the mutual information $I(X_u; X_I)$ were zero for all sets I then Bob could not have positive expected value in the GuessingGame. This can be made precise and yields a lower bound on the mutual information. We can extend the argument to work with conditional mutual information by exploiting the fact that there are many clique potentials that do not get cancelled out when conditioning.

4.0.2 Our Results

Recall that $N(u)$ denotes the neighbors of u. We require certain conditions (Definition 12) on the clique potentials to hold, which we call α, β-non-degeneracy, to ensure that the presence or absence of each hyperedge can be information-theoretically determined from few samples (essentially that no clique potential is too large and no non-zero clique potential is too small). Under this condition, we prove:

Theorem 21. *Fix any node u in an α, β-non-degenerate graphical model of bounded degree and a subset of the vertices S which does not contain the entire neighborhood of u. Then taking I uniformly at random from the subsets of the neighbors of u not contained in S of size $s = \min(r-1, |N(u) \setminus S|)$, we have $\mathbb{E}_I[I(X_u; X_I | X_S)] \geq C$.*

See Theorem 27 which gives the precise dependence of C on all of the constants, including α, β, the maximum degree D, the order of the interactions r and the upper bound K on the number of states of each node. We remark that C is exponentially small in D, r and β and there are examples where this dependence is necessary [46].

Next we apply our structural result within Bresler's [7] greedy framework for structure learning to obtain our main algorithmic result. We obtain an algorithm for learning graphical models on bounded degree graphs with a logarithmic number of samples, which is information-theoretically optimal [46]. More precisely we prove:

Theorem 22. *Fix any α, β-non-degenerate graphical model on n nodes with r-order interactions and bounded degree. There is an algorithm for learning G that succeeds with high probability given $C' \log n$ samples and runs in time polynomial in n^r.*

Remark 23. An $r-1$-sparse parity with noise is a graphical model with order r interactions. This means if we could improve the running time to $n^{o(r)}$ this would yield the first $n^{o(k)}$ algorithm for learning k-sparse parities with noise, which is a long-standing open question. The best known algorithm of Valiant [56] runs in time $n^{0.8k}$.

See Theorem 29 for a more precise statement. The constant C' depends doubly exponentially on D. In the special case of Ising models with no external field, Vuffray et al. [58] gave an algorithm based on convex programming that reduces the dependence on D to singly exponential. In 4.5 we will explain how to generalize their convex program to arbitrary graphical models, though it is unclear whether their efficiency bounds generalize as well. In contrast, in greedy approaches based on mutual information like the one we consider here, doubly-exponential dependence on D seems intrinsic. As in Bresler's [7] work, we construct a superset of the neighborhood that contains roughly $1/C$ nodes where C comes from Theorem 21. Recall that C is exponentially small in D. Then to accurately estimate conditional mutual information when conditioning on the states of this many nodes, we need doubly exponential in D many samples.

However, there is a distinct advantage to greedy based methods. Since we only ever need to estimate the conditional mutual information on a constant sized sets of nodes and when conditioning on a constant sized set of other nodes, we can perform structure learning with partial observations. More precisely, if for every sample from a graphical model, we are allowed to specify a set J of size at most a constant C'' where all we observe is X_J we can still learn the structure of the graphical model. We call such queries C''-bounded queries.

Theorem 24. *Fix any α, β-non-degenerate graphical model on n nodes with r-order interactions and bounded degree. There is an algorithm for learning G with C''-*

bounded queries that succeeds with high probability given $C' \log n$ samples and runs in time polynomial in n^r.

See Theorem 32 for a more precise statement. This natural scenario arises when it is too expensive to obtain a sample where the states of all nodes are known. The only other results we are aware of for learning with bounded queries work only for Gaussian graphical models [22]. We also consider a model where the state of each node is erased (i.e. we observe a '?' instead of its state) independently with some fixed probability p. See Theorem 33 for a precise statement. *The fact that we can straightforwardly obtain algorithms for these alternative settings demonstrates the flexibility of greedy, information-theoretic approaches to learning.*

In concurrent and independent work and using a different approach, Klivans and Meka [38] gave an algorithm for learning graphical models with r-order interactions and maximum degree D with a non-degeneracy assumption corresponding to a bound on the ℓ_1-norm of the derivatives of the clique potentials. Under our non-degeneracy assumptions, their algorithm runs in time n^r and has sample complexity $2^{D^r}(nr)^r$ and under related but stronger assumptions than we use here, their sample complexity improves to $2^{D^r} r^r \log n$.

4.1 Preliminaries

For reference, all fundamental parameters of the graphical model (max degree, etc.) are defined in the next two subsections. In terms of these fundamental parameters, we define additional parameters γ and δ in (4.4), $C(\gamma, K, \alpha)$ and $C'(\gamma, K, \alpha)$ in Theorem 26 and Theorem 27 respectively, and τ in (4.8) and L in (4.9).

4.1.1 Graphical models and the Canonical Form

We study the problem of structure learning for graphical models. Formally, a graphical model is specified by a hypergraph $\mathcal{H} = (V, H)$ where each hyperedge $h \in H$ is a set of at most r vertices and $V = [n]$. To each node i, we associate a random variable X_i which can take on one of k_i different states/spins/colors so that $X_i \in [k_i]$. Let K be an upper bound on the maximum number of states of any node. To each hyperedge $h = (i_1, i_2, \cdots i_\ell)$ we associate an ℓ-order tensor $\theta^{i_1 i_2 \cdots i_\ell}$ with dimensions $k_{i_1} \times \cdots k_{i_\ell}$ which represents the clique interaction on these nodes. When $(i_1, i_2, \cdots i_\ell)$ are not a hyperedge in $\mathcal{H}$ we define $\theta^{i_1 i_2 \cdots i_\ell}$ to be the zero tensor.

The potential energy of the model being in state $X = (X_1, \ldots, X_n)$ is given by

$$H_\theta(X) = -\sum_{\ell=1}^{r} \sum_{i_1 < i_2 < \cdots < i_\ell} \theta^{i_1 \cdots i_\ell}(x_{i_1}, \ldots, x_{i_\ell}). \tag{4.1}$$

The joint probability of the model being in state $X = (X_1, \ldots, X_n)$ is proportional to $\exp(-H_\theta(X))$, and is therefore given by

$$\mathbf{Pr}(X = x) = \exp\left(\sum_{\ell=1}^{r} \sum_{i_1 < i_2 < \cdots < i_\ell} \theta^{i_1 \cdots i_\ell}(x_{i_1}, \ldots, x_{i_\ell}) - C\right) \tag{4.2}$$

where C is a constant normalizing the total probability to one. For notational convenience, even when $i_1, \ldots, i_\ell$ are not sorted in increasing order, we define $\theta^{i_1 \cdots i_\ell}(a_1, \ldots, a_\ell) = \theta^{i'_1 \cdots i'_\ell}(a'_1, \ldots, a'_\ell)$ where the $i'_1, \ldots, i'_\ell$ are the sorted version of $i_1, \ldots, i_\ell$ and the $a'_1, \ldots, a'_\ell$ are the corresponding copies of $a_1, \ldots, a_\ell$.

The parameterization above is not unique. It will be helpful to put it in a normal form as below. A *tensor fiber* is the vector given by fixing all of the indices of the tensor except for one; this generalizes the notion of row/column in matrices. For example for any $1 \leq m \leq \ell$, $i_1 < \ldots < i_m < \ldots i_\ell$ and $a_1, \ldots, a_{m-1}, a_{m+1}, \ldots a_\ell$ fixed, the corresponding tensor fiber is the set of elements $\theta^{i_1 \cdots i_\ell}(a_1, \ldots, a_m, \ldots, a_\ell)$ where a_m ranges from 1 to k_{i_m}.

Definition 10. We say that the weights θ are in *canonical form*[1] if for every tensor $\theta^{i_1 \cdots i_\ell}$, the sum over all of the *tensor fibers* of $\theta^{i_1 \cdots i_\ell}$ is zero.

Moreover we say that a tensor with the property that the sum over all tensor fibers is zero is a *centered tensor*. Hence having a graphical model in canonical form

[1] This is the same as writing the log of the probability mass function according to the *Efron-Stein decomposition* with respect to the uniform measure on colors; this decomposition is known to be unique. See e.g. Chapter 8 of [44]

just means that all of the tensors corresponding to its clique potentials are centered. Next we prove that every graphical model can be put in canonical form:

Claim 2. *Every graphical model can be put in canonical form.*

Proof. We will recenter the tensors one by one without changing the law in (4.2). Starting with an arbitrary parameterization, observe that if the sum along some tensor fiber is $s \neq 0$, we can subtract s/k_{i_m} from each of the entries in the tensor fiber, so the sum over the tensor fiber is now zero, and add s to $\theta^{i_{\sim m}}(a_{\sim m})$ without changing the law of X in (4.2). Here $i_{\sim m}$ is our notation for $i_1, \ldots, i_{m-1}, i_{m+1}, \ldots i_\ell$. By iterating this process from the tensors representing the highest-order interactions down to the tensors representing the lowest-order interactions[2], we obtain the desired canonical form. □

4.1.2 Non-Degeneracy

We let $G = (V, E)$ be the graph we obtain from $\mathcal{H}$ by replacing every hyperedge with a clique. Let d_i denote the degree of i in G and let D be a bound on the maximum degree. Let $N(i)$ denote the neighborhood of i. Then as usual G encodes the independence properties of the graphical model. Our goal is to recover the structure of G with high probability. In order to accomplish this, we will need to ensure that edges and hyperedges are non-degenerate.

Definition 11. We say that a hyperedge h is maximal if no other hyperedge of strictly larger size contains h.

Informally, we will require that every edge in G is contained in some non-zero hyperedge, that all maximal hyperedges have at least one parameter bounded away from zero and that no entries are too large. More formally:

Definition 12. We say that a graphical model is α,β-non-degenerate if

[2] We treat C as the lowest order interaction, so when we are subtracting from the 1-tensors (vectors) θ^i to recenter them, we add the corresponding amount to C.

(a) Every edge (i, j) in the graph G is contained in some hyperedge $h \in H$ where the corresponding tensor is non-zero.

(b) Every maximal hyperedge $h \in H$ has at least one entry lower bounded by α in absolute value.

(c) Every entry of $\theta^{i_1 i_2 \cdots i_\ell}$ is upper bounded by a constant β in absolute value.

We will refer to a hyperedge h with an entry lower bounded by α in absolute value as α-*nonvanishing*. Each of the above non-degeneracy conditions is imposed in order to make learning G information-theoretically possible. If an edge (i, j) were not contained in any hyperedge with a non-zero tensor then we could remove the edge and not change the law in (4.2). If a hyperedge contains only entries that are arbitrarily close to zero, we cannot hope to learn that it is there. We require α-nonvanishing just for maximal hyperedges so that it is still possible to learn G. Finally if we did not have an upper bound on the absolute value of the entries, the probabilities in (4.2) could become arbitrarily skewed and there could be nodes i where X_i only ever takes on a single value.

4.1.3 Bounds on Conditional Probabilities

First we review properties of the conditional probabilities in a graphical model as well as introduce some convenient notation which we will use later on. Fix a node u and its neighborhood $U = N(u)$. Then for any $R \in [k_u]$ we have

$$P(X_u = R | X_U) = \frac{\exp(\mathcal{E}^X_{u,R})}{\sum_{B=1}^{k_u} \exp(\mathcal{E}^X_{u,B})} \tag{4.3}$$

where we define

$$\mathcal{E}^X_{u,R} = \sum_{\ell=1}^{r} \sum_{i_2 < \cdots < i_\ell} \theta^{u i_2 \cdots i_\ell}(R, X_{i_2}, \cdots, X_{i_\ell})$$

and $i_2, \ldots, i_\ell$ range over elements of the neighborhood U; when $\ell = 1$ the inner sum is just $\theta^u(R)$. To see that the above is true, first condition on $X_{\sim u}$, and see that the probability for a certain X_u is proportional to $\exp(\mathcal{E}^X_{u,R})$, which gives the right hand side of (4.3). Then apply the tower property for conditional probabilities.

Therefore if we define (where $|T|_{max}$ denotes the maximum entry of a tensor T)

$$\gamma := \sup_u \sum_{\ell=1}^{r} \sum_{i_2 < \cdots < i_\ell} |\theta^{ui_2 \cdots i_\ell}|_{max} \leq \beta \sum_{\ell=1}^{r} \binom{D}{\ell-1}, \qquad \delta := \frac{1}{K}\exp(-2\gamma) \tag{4.4}$$

then for any R

$$P(X_u = R|X_U) \geq \frac{\exp(-\gamma)}{K\exp(\gamma)} = \frac{1}{K}\exp(-2\gamma) = \delta. \tag{4.5}$$

Observe that if we pick any node i and consider the new graphical model given by conditioning on a fixed value of X_i, then the value of γ for the new graphical model is non-increasing.

4.1.4 Lower Bounds for Conditional Mutual Information

As in Bresler's work on learning Ising models [7], certain information theoretic quantities will play a crucial role as a progress measure in our algorithms. Specifically, we will use the functional

$$\nu_{u,I|S} := \mathbb{E}_{R,G}\Big[\mathbb{E}_{X_S}\Big[\Big|\mathbf{Pr}(X_u = R, X_I = G|X_S) - \mathbf{Pr}(X_u = R|X_S)\,\mathbf{Pr}(X_I = G|X_S)\Big|\Big]\Big]$$

where R is a state drawn uniformly at random from $[k_u]$, G is an $|I|$-tuple of states drawn independently uniformly at random from $[k_{i_1}] \times [k_{i_2}] \times \ldots \times [k_{i_{|I|}}]$ where $I = (i_1, i_2, \ldots i_{|I|})$. This will be used as a *proxy* for conditional mutual information which can be efficiently estimated from samples. The following lemma is a version of Lemma 5.1 in [7] that works over non-binary alphabets.

Lemma 4. *Fix a set of nodes S. Fix a node u and a set of nodes I that are not contained in S. Then*

$$\sqrt{\frac{1}{2}I(X_u; X_I|X_S)} \geq \nu_{u,I|S}.$$

Proof.

$$
\begin{aligned}
\sqrt{\frac{1}{2}I(X_u;X_I|X_S)} &= \sqrt{\frac{1}{2}\mathbb{E}_{X_S=x_S}[I(X_u;X_I|X_S=x_S)]} \\
&\geq \mathbb{E}_{X_S=x_S}\left[\sqrt{\frac{1}{2}I(X_u;X_I|X_S=x_S)}\right] \\
&= \mathbb{E}_{X_S=x_S}\left[\sqrt{\frac{1}{2}D_{KL}(\mathbf{Pr}(X_u,X_I|X_S=x_S)||\,\mathbf{Pr}(X_u|X_S=x_S)\,\mathbf{Pr}(X_I|X_S=x_S))}\right] \\
&\geq \mathbb{E}_{X_S}[\sup_{R,G}[|\,\mathbf{Pr}(X_u=R,X_I=G|X_S)-\mathbf{Pr}(X_u=R|X_S)\,\mathbf{Pr}(X_I=G|X_S)|]] \\
&\geq \mathbb{E}_{X_S}[\mathbb{E}_{R,G}[|\,\mathbf{Pr}(X_u=R,X_I=G|X_S)-\mathbf{Pr}(X_u=R|X_S)\,\mathbf{Pr}(X_I=G|X_S)|]] \\
&= \nu_{u,I|S}
\end{aligned}
$$

where the first inequality follows from Jensen's inequality, and the second inequality follows from Pinsker's inequality. □

4.1.5 No Cancellation

In this subsection we will show that a clique interaction of order s cannot be completely cancelled out by clique interactions of lower order.

Lemma 5. *Let $T^{1\cdots s}$ be a centered tensor of dimensions $d_1 \times \cdots \times d_s$ and suppose there exists at least one entry of $T^{1\cdots s}$ which is lower bounded in absolute value by a constant κ. For any $\ell < s$ and $i_1 < \cdots < i_\ell$ let $T^{i_1\cdots i_\ell}$ be an arbitrary centered tensor of dimensions $d_{i_1} \times \cdots \times d_{i_\ell}$. Define*

$$
T(a_1,\ldots,a_s) = \sum_{\ell=1}^{s}\sum_{i_1<\cdots<i_\ell} T^{i_1\cdots i_\ell}(a_{i_1},\ldots,a_{i_\ell}) \tag{4.6}
$$

and suppose the entries of T are bounded by a constant μ. Then for any ℓ and $i_1 < \cdots < i_\ell$, the entries of $T^{i_1\cdots i_\ell}(a_{i_1},\ldots,a_{i_\ell})$ are bounded above by $\mu\ell^\ell$.

Proof. The sum over all values of indices $a_1,\ldots,a_s$ on the right hand side is zero, so the same must hold for the left hand side. Assume for contradiction that every entry of T is upper bounded by μ, to be optimized later. For each m from 1 to s, consider summing over all of the indices except a_m, which is held fixed. Using that the sum over tensor fibers is zero, we observe that the right hand side of (4.7) is just

$$T^{i_m}(a_1) \prod_{m' \neq m} d_{m'}$$

and the left hand side is strictly bounded in norm by $\mu \prod_{m' \neq m} d_{m'}$ so $|T^{i_m}(a_m)| < \mu$ for all a_m. We have proven this for all m from 1 to s.

Now we proceed by induction, assuming that t indices are fixed. We will show that the entries of the t-tensors are bounded above by $\mu g(t)$ for $g(t) = 2^{t(t+1)/2}$ and have already proven this for $t = 1$. Now suppose we fix $a_1, \ldots, a_t$. We rearrange (4.7) to get

$$\begin{aligned} T(a_1, \ldots, a_s) &- \sum_{\ell=1}^{t-1} \sum_{\{i_1 < \cdots < i_\ell\} \subset [t]} T^{i_1 \cdots i_\ell}(a_{i_1}, \ldots, a_{i_\ell}) \\ &= T^{1 \cdots t}(a_1, \ldots, a_t) + \sum_{\ell=1}^{s} \sum_{\{i_1 < \cdots < i_\ell\} \not\subset [t]} T^{i_1 \cdots i_\ell}(a_{i_1}, \ldots, a_{i_\ell}). \end{aligned}$$

When we fix indices $a_1, \ldots, a_t$ and sum over the others, all but the first term on the rhs vanishes, and by applying the triangle inequality on the lhs and the induction hypothesis we get that

$$d_{t+1} \cdots d_s \left(\mu + \sum_{\ell=1}^{t-1} \binom{t}{\ell} \mu g(\ell) \right) > d_{t+1} \cdots d_s T^{ui_2 \cdots i_t}(a_1, \ldots, a_t)$$

so taking $g(t)$ such that $g(0) = 1$ and

$$g(t) \geq \sum_{\ell=0}^{t-1} \binom{t}{\ell} g(\ell)$$

and in particular $g(t) = t^t$ works, because

$$t^t = (1 + (t-1))^t = \sum_{\ell=0}^{t} \binom{t}{\ell} (t-1)^\ell \geq \sum_{\ell=0}^{t-1} \binom{t}{\ell} \ell^\ell.$$

Thus we get that all the entries of $T^{i_1\cdots i_\ell}(a_{i_1},\dots,a_{i_\ell})$ are bounded above by $\mu\ell^\ell$, which completes the proof. $\square$

We are now ready to restate the above result in a more usable form:

Lemma 6. *Let $T^{1\cdots s}$ be a centered tensor of dimensions $d_1\times\cdots\times d_s$ and suppose there exists at least one entry of $T^{1\cdots s}$ which is lower bounded in absolute value by a constant κ. For any $\ell<s$ and $i_1<\cdots<i_\ell$ let $T^{i_1\cdots i_\ell}$ be an arbitrary centered tensor of dimensions $d_{i_1}\times\cdots\times d_{i_\ell}$. Let*

$$T(a_1,\dots,a_s)=\sum_{\ell=1}^{s}\sum_{i_1<\cdots<i_\ell}T^{i_1\cdots i_\ell}(a_{i_1},\dots,a_{i_\ell}). \tag{4.7}$$

Then the sum over all the entries of T is 0, and there exists an entry of T of absolute value lower bounded by κ/s^s.

Proof. We apply the previous lemma with $\mu=\kappa/s^s$, and get that all the entries of $T^{1\cdots s}$ are bounded in absolute value by μs^s, giving a contradiction. $\square$

4.2 The Guessing Game

Here we introduce a game-theoretic framework for understanding mutual information in general graphical models. The GuessingGame is defined as follows:

---

1. Alice samples $X=(X_1,\dots,X_n)$ and $X'=(X'_1,\dots,X'_n)$ independently from the graphical model
2. Alice samples R uniformly at random from $[k_u]$
3. Alice samples a set I of size $s=\min(r-1,d_u)$ uniformly at random from the neighbors of u
4. Alice tells Bob I, X_I and R
5. Bob wagers w with $|w|\le\gamma K\binom{D}{r-1}$

6. Bob gets $\Delta = w\mathbb{1}_{X_u=R} - w\mathbb{1}_{X'_u=R}$

Bob's goal is to guess X_u given knowledge of the states of some of u's neighbors. The graphical model (including all of its parameters) are common knowledge. The intuition is that if Bob can obtain a positive expected value, then there must be some set I of neighbors of u which have non-zero mutual information. In this section, will show that there is a simple, explicit strategy for Bob that yields positive expected value.

4.2.1 A Good Strategy for Bob

Here we will show an explicit strategy for Bob that has positive expected value. Our analysis will rest on the following key lemma:

Lemma 7. *There is a strategy for Bob that wagers at most $\gamma K\binom{D}{r-1}$ in absolute value that satisfies*

$$\mathbb{E}_{I,X_I}[w|X_{\sim u}, R] = \mathcal{E}^X_{u,R} - \sum_{B\neq R} \mathcal{E}^X_{u,B}.$$

Proof. First we explicitly define Bob's strategy. Let

$$\Phi(R, I, X_I) = \sum_{\ell=1}^{s} C_{u,\ell,s} \sum_{i_1<i_2<\cdots<i_\ell} \mathbb{1}_{\{i_1\cdots i_\ell\}\subseteq I}\theta^{ui_1\cdots i_\ell}(R, X_{i_1}, \dots, X_{i_\ell})$$

where $C_{u,\ell,s} = \frac{\binom{d_u}{s}}{\binom{d_u-\ell}{s-\ell}}$. Then Bob wagers

$$w = \Phi(R, I, X_I) - \sum_{B\neq R} \Phi(B, I, X_I).$$

Notice that the strategy only depends on X_I because all terms in the summation where $\{i_1 \cdots i_\ell\}$ are not a subset of I have zero contribution.

The intuition behind this strategy is that the weighting term satisifes

$$C_{u,\ell,s} = \frac{1}{\mathbf{Pr}[\{i_1, \dots i_\ell\} \subset I]}.$$

Thus when we take the expectation over I and X_I we get

$$\mathbb{E}_{I,X_I}[\Phi(R,I,X_I)|X_{\sim u},R] = \sum_{\ell=1}^{r} \sum_{i_2<\cdots<i_\ell} \theta^{ui_2\cdots i_\ell}(R, X_{i_2}, \cdots, X_{i_\ell}) = \mathcal{E}^X_{u,R}$$

and hence $\mathbb{E}_{I,X_I}[w|X_{\sim u},R] = \mathcal{E}^X_{u,R} - \sum_{B\neq R} \mathcal{E}^X_{u,B}$. To complete the proof, notice that $C_{u,\ell,s} \leq \binom{D}{r-1}$ which using the definition of γ implies that $|\Phi(R,I,X_I)| \leq \gamma\binom{D}{r-1}$ for any state B, and thus Bob wagers at most the desired amount (in absolute value). □

Now we are ready to analyze the strategy:

Theorem 25. *There is a strategy for Bob that wagers at most $\gamma K\binom{D}{r-1}$ in absolute value which satisfies*

$$\mathbb{E}[\Delta] \geq \frac{4\alpha^2\delta^{r-1}}{r^{2r}e^{2\gamma}}.$$

Proof. We will use the strategy from Lemma 7. First we fix $X_{\sim u}$, $X'_{\sim u}$ and R. Then we have

$$\mathbb{E}_{I,X_I}[\Delta|X_{\sim u}, X'_{\sim u}, R] = \mathbb{E}_{I,X_I}[w|X_{\sim u},R]\Big(\mathbf{Pr}[X_u = R|X_{\sim u},R] - \mathbf{Pr}[X'_u = R|X'_{\sim u},R]\Big)$$

which follows because $\Delta = r\mathbb{1}_{X_u=R} - r\mathbb{1}_{X'_u=R}$ and because r and X_u do not depend on $X'_{\sim u}$ and similarly X'_u does not depend on $X_{\sim u}$. Now using (4.3) we calculate:

$$\begin{aligned}\mathbf{Pr}[X_u = R|X_{\sim u},R] - \mathbf{Pr}[X'_u = R|X'_{\sim u},R] &= \frac{\exp(\mathcal{E}^X_{u,R})}{\sum_B \exp(\mathcal{E}^X_{u,B})} - \frac{\exp(\mathcal{E}^{X'}_{u,R})}{\sum_B \exp(\mathcal{E}^{X'}_{u,B})} \\ &= \frac{1}{D}\Big(\sum_{B\neq R} \exp(\mathcal{E}^X_{u,R} + \mathcal{E}^{X'}_{u,B}) - \exp(\mathcal{E}^X_{u,B} + \mathcal{E}^{X'}_{u,R})\Big)\end{aligned}$$

where $D = \Big(\sum_B \exp(\mathcal{E}^X_{u,B})\Big)\Big(\sum_B \exp(\mathcal{E}^{X'}_{u,B})\Big)$. Thus putting it all together we have

$$\mathbb{E}_{I,X_I}[\Delta|X_{\sim u}, X'_{\sim u}, R] = \frac{1}{D}\Big(\mathcal{E}^X_{u,R} - \sum_{B\neq R} \mathcal{E}^X_{u,B}\Big)\Big(\sum_{B\neq R} \exp(\mathcal{E}^X_{u,R} + \mathcal{E}^{X'}_{u,B}) - \exp(\mathcal{E}^X_{u,B} + \mathcal{E}^{X'}_{u,R})\Big).$$

Now it is easy to see that

$$\sum_{\text{distinct } R,G,B} \mathcal{E}^X_{u,B} \left(\sum_{G \neq R} \exp(\mathcal{E}^X_{u,R} + \mathcal{E}^{X'}_{u,G}) - \exp(\mathcal{E}^X_{u,G} + \mathcal{E}^{X'}_{u,R}) \right) = 0$$

which follows because when we interchange R and G the entire term multiplies by a negative one and so we can pair up the terms in the summation so that they exactly cancel. Using this identity we get

$$\mathbb{E}_{I,X_I}[\Delta | X_{\sim u}, X'_{\sim u}] = \frac{1}{k_u D} \sum_R \sum_{B \neq R} \Big(\mathcal{E}^X_{u,R} - \mathcal{E}^X_{u,B} \Big) \Big(\exp(\mathcal{E}^X_{u,R} + \mathcal{E}^{X'}_{u,B}) - \exp(\mathcal{E}^X_{u,B} + \mathcal{E}^{X'}_{u,R}) \Big)$$

where we have also used the fact that R is uniform on k_u. And finally using the fact that $X_{\sim u}$ and $X'_{\sim u}$ are identically distributed we can sample $Y_{\sim u}$ and $Z_{\sim u}$ and flip a coin to decide whether we set $X_{\sim u} = Y_{\sim u}$ and $X'_{\sim u} = Z_{\sim u}$ or vice-versa. Now we have

$$\mathbb{E}_{I,X_I}[\Delta | Y_{\sim u}, Z_{\sim u}] = \frac{1}{2k_u D} \sum_R \sum_{B \neq R} \Big(\mathcal{E}^Y_{u,R} - \mathcal{E}^Y_{u,B} - \mathcal{E}^Z_{u,R} + \mathcal{E}^Z_{u,B} \Big) \Big(\exp(\mathcal{E}^Y_{u,R} + \mathcal{E}^Z_{u,B}) - \exp(\mathcal{E}^Y_{u,B} + \mathcal{E}^Z_{u,R}) \Big).$$

With the appropriate notation it is easy to see that the above sum is strictly positive. Let $a_{R,B} = \mathcal{E}^Y_{u,R} + \mathcal{E}^Z_{u,B}$ and $b_{R,B} = \mathcal{E}^Z_{u,R} + \mathcal{E}^Y_{u,B}$. With this notation:

$$\mathbb{E}_{I,X_I}[\Delta | Y_{\sim u}, Z_{\sim u}] = \frac{1}{2Dk_u} \sum_R \sum_{B \neq R} \Big(a_{R,B} - b_{R,B} \Big) \Big(\exp(a_{R,B}) - \exp(b_{R,B}) \Big).$$

Since $\exp(x)$ is a strictly increasing function it follows that as long as $a_{R,B} \neq b_{R,B}$ for some term in the sum, the sum is positive. In Lemma 8 we prove that the expectation over Y and Z of this sum is at least $\frac{4\alpha^2\delta^{r-1}}{r^{2r}e^{2\gamma}}$, which completes the proof. □

4.2.2 A Quantitative Lower Bound

Here we prove a quantitative lower bound on the sum that arose in the proof of Theorem 25. More precisely we show:

Lemma 8.

$$\mathbb{E}_{Y,Z}\Big[\sum_{R}\sum_{B\neq R}\Big(\mathcal{E}^Y_{u,R}-\mathcal{E}^Y_{u,B}-\mathcal{E}^Z_{u,R}+\mathcal{E}^Z_{u,B}\Big)\Big(\exp(\mathcal{E}^Y_{u,R}+\mathcal{E}^Z_{u,B})-\exp(\mathcal{E}^Y_{u,B}+\mathcal{E}^Z_{u,R})\Big)\Big] \geq \frac{4\alpha^2\delta^{r-1}}{r^{2r}e^{2\gamma}}.$$

Proof. Setting $a = \mathcal{E}^Y_{u,R} + \mathcal{E}^Z_{u,B}$ and $b = \mathcal{E}^Y_{u,B} + \mathcal{E}^Z_{u,R}$, letting $D' = K^3 \exp(2\gamma) \geq D$, and taking an expectation over the randomness in Y and Z, we have

$$\begin{aligned}
\mathbb{E}_{Y,Z}\Big[\sum_{R}\sum_{R\neq B}(a-b)(e^a-e^b)\Big] &= \mathbb{E}[\sum_{R}\sum_{R\neq B}(a-b)\int_b^a e^x dx] \\
&\geq \mathbb{E}[\sum_{R}\sum_{R\neq B}(a-b)^2 e^{-2\gamma}] \geq \frac{1}{e^{2\gamma}}\sum_{R}\sum_{R\neq B}\mathrm{Var}[a-b]
\end{aligned}$$

where the inequality follows from the fact that $a, b \geq -2\gamma$. In the following claim, we give a more convenient expression for the above quantity.

Claim 3.

$$\sum_{R}\sum_{R\neq B}\mathrm{Var}[a-b] = 4k_u\sum_{R}\mathrm{Var}[\mathcal{E}^Y_{u,R}].$$

Proof. Using the fact that $a - b = (\mathcal{E}^Y_{u,R} - \mathcal{E}^Y_{u,B}) + (\mathcal{E}^Z_{u,B} - \mathcal{E}^Z_{u,R})$ we have that

$$\begin{aligned}
\sum_{R}\sum_{R\neq B}\mathrm{Var}[a-b] &= \sum_{R}\sum_{B\neq R}\mathrm{Var}[(\mathcal{E}^Y_{u,R} - \mathcal{E}^Y_{u,B}) + (\mathcal{E}^Z_{u,B} - \mathcal{E}^Z_{u,R})] \\
&= 2\sum_{R}\sum_{B\neq R}\mathrm{Var}[(\mathcal{E}^Y_{u,R} - \mathcal{E}^Y_{u,B})] \\
&= 2\sum_{R}\sum_{B\neq R}\Big(2\,\mathrm{Var}[\mathcal{E}^Y_{u,R}] - 2\,\mathrm{Cov}\Big(\mathcal{E}^Y_{u,R}, \mathcal{E}^Y_{u,B}\Big)\Big) \\
&= 2\sum_{R}\Big(2(k_u-1)\,\mathrm{Var}[\mathcal{E}^Y_{u,R}] - 2\,\mathrm{Cov}\Big(\mathcal{E}^Y_{u,R}, \sum_{B\neq R}\mathcal{E}^Y_{u,B}\Big)\Big) \\
&= 2\sum_{R}\Big(2(k_u-1)\,\mathrm{Var}[\mathcal{E}^Y_{u,R}] - 2\,\mathrm{Cov}\Big(\mathcal{E}^Y_{u,R}, -\mathcal{E}^Y_{u,R}\Big)\Big) \\
&= 4k_u\sum_{R}\mathrm{Var}[\mathcal{E}^Y_{u,R}]
\end{aligned}$$

where the second to last equality follows from the fact that the tensors are centered which gives $\sum_R \mathcal{E}^Y_{u,R} = 0$ for any Y. This completes the proof. □

Now we can complete the proof by appealing to the law of total variance. By assumption there is a maximal hyperedge $J = \{u, j_1 \dots j_s\}$ containing u with $|J| \leq r$, such that θ^{uJ} is α-nonvanishing. Then we have

$$\sum_R \mathrm{Var}[\mathcal{E}^Y_{u,R}] \geq \sum_R \mathrm{Var}[\mathcal{E}^Y_{u,R}|Y_{\sim J}] = \sum_R \mathrm{Var}[T(R, Y_{j_1}, \dots, Y_{j_s})|Y_{\sim J}]$$

where the tensor T is defined by treating $Y_{\sim J}$ as fixed as follows:

$$T(R, Y_{j_1}, \dots, Y_{j_s}) = \sum_{\ell=2}^{r} \sum_{i_2 < \cdots < i_\ell} \theta^{u i_2 \cdots i_\ell}(R, Y_{i_2}, \cdots, Y_{i_\ell}).$$

Now we claim there is a choice of R, G and G' so that $|T(R, G) - T(R, G')| > \alpha/r^r$. This follows because from Lemma 6 we have that T is α/r^r-nonvanishing. Hence there is a choice of R and G so that $|T(R, G)| > \alpha/r^r$. Because T is centered there must be a G' so that $T(R, G')$ has the opposite sign.

Finally for this choice of R we have

$$\mathrm{Var}[T(R, Y_{j_1}, \dots, Y_{j_s})|Y_{\sim J}] \geq \frac{\alpha^2 \delta^{r-1}}{2r^{2r}}$$

which follows from the fact that $\mathbf{Pr}(Y_{J\setminus u} = G)$ and $\mathbf{Pr}(Y_{J\setminus u} = G')$ are both lower bounded by δ^{r-1} and the following elementary lower bound on the variance:

Claim 4. *Let Z be a random variable such that $Pr(Z = a) \geq p$ and $Pr(Z = b) \geq p$, then*

$$\mathrm{Var}(Z) \geq \frac{p}{2}(a - b)^2.$$

Proof.

$$\mathrm{Var}(Z) \geq p(a - \mathbb{E}[Z])^2 + p(\mathbb{E}[Z] - b)^2 \geq p\Big(a - \frac{a+b}{2}\Big)^2 + p\Big(b - \frac{a+b}{2}\Big)^2 = \frac{p}{2}(a - b)^2.$$

□

Putting this all together we have

$$\mathbb{E}_{Y,Z}\Big[\sum_{R}\sum_{R\neq B}(a-b)(e^a-e^b)\Big] \geq \frac{4\alpha^2\delta^{r-1}}{r^{2r}e^{2\gamma}}$$

which is the desired bound. This completes the proof. □

4.3 Implications for Mutual Information

In this section we show that Bob's strategy implies a lower bound on the mutual information between node u and a subset I of its neighbors of size at most $r-1$. We then extend the argument to work with conditional mutual information as well.

4.3.1 Mutual Information in graphical models

Recall that the goal of the GUESSINGGAME is for Bob to use information about the states of nodes I to guess the state of node u. Intuitively, if X_I conveys no information about X_u then it should contradict the fact that Bob has a strategy with positive expected value. We make this precise below. Our argument proceeds in two steps. First we upper bound the expected value of any strategy.

Lemma 9. *For any strategy,*

$$\mathbb{E}[\Delta] \leq \gamma K\binom{D}{r-1}\mathbb{E}_{I,X_I,R}\Big[|\,\mathbf{Pr}[X_u=R|X_I]-\mathbf{Pr}[X_u=R]|\Big].$$

Proof. Intuitively this follows because Bob's optimal strategy given I, X_I and R is to guess

$$w = \operatorname{sgn}(\mathbf{Pr}[X_u=R|X_I]-\mathbf{Pr}[X_u=R])\gamma K.$$

More precisely, we have

$$\begin{aligned}\mathbb{E}[\Delta] &= \mathbb{E}_{I,X_I,R}\Big[\mathbb{E}_{X_{\sim I},X'}\Big[r\mathbb{1}_{X_u=R} - r\mathbb{1}_{X'_u=R}\Big| I, X_I, R\Big]\Big] \\ &= \mathbb{E}_{I,X_I,R}\Big[r\,\mathbf{Pr}[X_u = R|X_I] - r\,\mathbf{Pr}[X'_u = R]\Big] \\ &= \mathbb{E}_{I,X_I,R}\Big[r\,\mathbf{Pr}[X_u = R|X_I] - r\,\mathbf{Pr}[X_u = R]\Big] \\ &\le \gamma K\binom{D}{r-1}\mathbb{E}_{I,X_I,R}\Big[|\,\mathbf{Pr}[X_u = R|X_I] - \mathbf{Pr}[X_u = R]|\Big]\end{aligned}$$

which completes the proof. □

Next we lower bound the mutual information using (essentially) the same quantity. We prove

Lemma 10.

$$\sqrt{\frac{1}{2}I(X_u; X_I)} \ge \frac{1}{K^r}\mathbb{E}_{X_I,R}\Big[|\,\mathbf{Pr}(X_u = R|X_I) - \mathbf{Pr}(X_u = R)|\Big].$$

Proof. Applying Lemma 4 with $S = \emptyset$ we have that

$$\begin{aligned}\sqrt{\frac{1}{2}I(X_u; X_I)} &\ge \mathbb{E}_{R,G}\Big[|\,\mathbf{Pr}(X_u = R, X_I = G) - \mathbf{Pr}(X_u = R)\,\mathbf{Pr}(X_I = G)|\Big] \\ &= \mathbb{E}_{R,G}\Big[\,\mathbf{Pr}(X_I = G)|\,\mathbf{Pr}(X_u = R|X_I = G) - \mathbf{Pr}(X_u = R)|\Big] \\ &= \frac{1}{\prod_{i\in I} k_i}\sum_G \mathbf{Pr}(X_I = G)\mathbb{E}_R[|\,\mathbf{Pr}(X_u = R|X_I = G) - \mathbf{Pr}(X_u = R)|] \\ &\ge \frac{1}{K^r}\mathbb{E}_{R,X_I}\Big[|\,\mathbf{Pr}(X_u = R|X_I) - \mathbf{Pr}(X_u = R)|\Big]\end{aligned}$$

where R and G are uniform (as in the definition of $\nu_{u,I|S}$). □

Now appealing to Lemma 9, Lemma 10 and Theorem 25 we conclude:

Theorem 26. *Fix a non-isolated vertex u contained in at least one α-nonvanishing maximal hyperedge. Then taking I uniformly at random from the subsets of the neigh-*

bors of u of size $s = \min(r-1, deg(u))$,

$$\mathbb{E}_I\left[\sqrt{\frac{1}{2}I(X_u; X_I)}\right] \geq \mathbb{E}_I[\nu_{u,I|\emptyset}] \geq C(\gamma, K, \alpha)$$

where explicitly

$$C(\gamma, K, \alpha) := \frac{4\alpha^2\delta^{r-1}}{r^{2r}K^{r+1}\binom{D}{r-1}\gamma e^{2\gamma}}.$$

4.3.2 Extensions to Conditional Mutual Information

In the previous subsection, we showed that X_u and X_I have positive mutual information. Here we show that the argument extends to conditional mutual information when we condition on X_S for any set S that does not contain all the neighbors of u. The main idea is to show that there is a setting of X_S where the hyperedges do not completely cancel out each other in the new graphical model we obtain by conditioning on X_S.

More precisely fix a set of nodes S that does not contain all the neighbors of u and let I be chosen uniformly at random from the subsets of neighbors of u of size $s = \min(r-1, |N(u) \setminus S|)$. Then we have

$$\begin{aligned}\mathbb{E}_I[\sqrt{\frac{1}{2}I(X_u; X_I|X_S)}] &= \mathbb{E}_I[\sqrt{\frac{1}{2}\mathbb{E}_{X_S=x_S}[I(X_u; X_I|X_S = x_S)]}] \\ &\geq \mathbb{E}_{I,X_S=x_S}\left[\sqrt{\frac{1}{2}I(X_u; X_I|X_S = x_S)}\right]\end{aligned}$$

which follows from Jensen's inequality. Now conditioned on $X_S = x_S$ the resulting distribution is again a graphical model and γ does not increase.

Definition 13. Let E be the event that conditioned on $X_S = x_S$, node u is contained in at least one α/r^r-nonvanishing maximal hyperedge.

Lemma 11. $\mathbf{Pr}(E) \geq \delta^d$

Proof. When we fix $X_S = x_S$ we obtain a new graphical model where the underlying hypergraph is

$$\mathcal{H}' := ([n] \setminus S, H') \text{ where } H' = \{h \setminus S | h \in H).$$

For notational convenience let $\phi(h)$ be the image of a hyperedge h in $\mathcal{H}$ in the new hypergraph $\mathcal{H}'$. What makes things complicated is that a hyperedge in $\mathcal{H}'$ can have numerous preimages. The crux of our argument is in how to select the right one to show is α/r^r-nonvanishing. First we observe that u is contained in at least one non-empty hyperedge in $\mathcal{H}'$. This is because by assumption S does not contain all the neighbors of u. Hence there is some neighbor $v \notin S$. Since v is a neighbor of u it means that there is a hyperedge $h \in H$ that contains both u and v. In particular $\phi(h)$ contains u and is nonempty.

Now that we know u is not isolated in $\mathcal{H}'$, let h^* be a hyperedge in $\mathcal{H}$ that contains u and where $\phi(h^*)$ is maximal. Now let $f_1, f_2, \dots f_p$ be the preimages of $\phi(h^*)$ so that without loss of generality f_1 is maximal in $\mathcal{H}$. Now let $J = \cup_{i=1}^{p} f_i \setminus \{u\}$. In particular, J is the set of neighbors of u that are contained in at least one of $f_1, f_2, \dots f_p$. Finally let $J_1 = J \cap S := \{i_1, i_2, \dots i_s\}$ and let $J_2 = J \setminus S := \{i'_1, i'_2, \dots i'_{s'}\}$. We can now define

$$T(R, a_1, \dots, a_s, a'_1, \dots, a'_{s'}) = \sum_{i=1}^{p} \theta^{f_i}$$

which is the clique potential we get on hyperedge $\phi(h^*)$ when we fix each index in $J_1 \subseteq S$ to their corresponding value.

Suppose for the purposes of contradiction that all the entries of T are strictly bounded in absolute value by α/r^r. Then applying Lemma 5 in the contrapositive we see that the entries of f_1 are strictly bounded above in absolute value by α, but f_1 is maximal and thus α-nonvanishing, which yields a contradiction. Thus there is some setting $a_1^*, \dots, a_s^*$ such that the tensor

$$T'(R, a'_1, \dots, a'_{s'}) = T(R, a_1^*, \dots, a_s^*, a'_1, \dots, a'_{s'})$$

has at least one entry with absolute value at least α/r^r. Under this setting, $\phi(h^*)$ is α/r^r-nonvanishing and by construction maximal in $\mathcal{H}'$ and thus we would be done. All that remains is to lower bound the probability of this setting. Since J_1 is a subset of the neighbors of u we have $|J_1| \leq d$. Thus the probability that $(X_{i_1}, \dots, X_{i_s}) = (a_1^*, \dots, a_s^*)$ is bounded below by $\delta^s \geq \delta^d$, which completes the proof. □

Now we are ready to prove a lower bound on conditional mutual information:

Theorem 27. *Fix a vertex u such that all of the maximal hyperedges containing u are α-nonvanishing, and a subset of the vertices S which does not contain the entire neighborhood of u. Then taking I uniformly at random from the subsets of the neighbors of u not contained in S of size $s = \min(r-1, |N(u) \setminus S|)$,*

$$\mathbb{E}_I\left[\sqrt{\frac{1}{2}I(X_u; X_I|X_S)}\right] \geq E_I[\nu_{u,I|S}] \geq C'(\gamma, K, \alpha)$$

where explicitly

$$C'(\gamma, K, \alpha) := \frac{4\alpha^2\delta^{r+d-1}}{r^{2r}K^{r+1}\binom{D}{r-1}\gamma e^{2\gamma}}.$$

Proof. We have

$$\mathbb{E}_{I,X_S}\left[\sqrt{\frac{1}{2}I(X_u; X_I|X_S)}\right] \geq \mathbb{E}_{I,X_S=x_S}\left[\sqrt{\frac{1}{2}I(X_u; X_I|X_S = x_S)}\mathbb{1}_E\right] \geq \delta^d C(\gamma, K, \alpha)$$

where the last inequality follows by invoking Lemma 11 and applying Theorem 26 to the new graphical model we get by conditioning on $X_S = x_S$. □

4.4 Applications

4.4.1 Learning graphical models

We now employ the greedy approach of Bresler [7] which was previously used to learn Ising models on bounded degree graphs. Let $x^{(1)}, \ldots, x^{(m)}$ denote a collection of independent samples from the underlying graphical model. Let $\widehat{\mathbf{Pr}}$ denote the empirical distribution so that

$$\widehat{\mathbf{Pr}}(X = x) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{x^{(i)}=x}.$$

Let $\widehat{\mathbb{E}}$ denote the expectation under this distribution, i.e. the sample average.

In our algorithm, we will need estimates for $\nu_{u,i|S}$ which we obtain in the usual way by replacing all expectations over X with sample averages:

$$\widehat{\nu}_{u,i|S} := \mathbb{E}_{R,G}\widehat{\mathbb{E}}_{X_S}[|\widehat{\mathbf{Pr}}(X_u = R, X_i = G|X_S) - \widehat{\mathbf{Pr}}(X_u = R|X_S)\widehat{\mathbf{Pr}}(X_i = G|X_S)|].$$

Also we define τ (which will be used as a thresholding constant) as

$$\tau := C'(\gamma, k, \alpha)/2 \tag{4.8}$$

and L, which is an upper bound on the size of the superset of a neighborhood of u that the algorithm will construct,

$$L := (8/\tau^2)\log K = (32/C'(\gamma, k, \alpha)^2)\log K. \tag{4.9}$$

Then the algorithm MrfNbhd at node u is:

1. Fix input vertex u. Set $S := \emptyset$.
2. While $|S| \leq L$ and there exists a set of vertices $I \subset [n] \setminus S$ of size at most $r - 1$ such that $\widehat{\nu}_{u,I|S} > \tau$, set $S := S \cup I$.
3. For each $i \in S$, if $\widehat{\nu}_{u,i|S\setminus i} < \tau$ then remove i from S.
4. Return set S as our estimate of the neighborhood of u.

The algorithm will succeed provided that $\widehat{\nu}_{u,I|S}$ is sufficiently close to the true value $\nu_{u,I|S}$. This motivates the definition of the event A:

Definition 28. We denote by $A(\ell, \varepsilon)$ the event that for all u, I and S with $|I| \leq r-1$ and $|S| \leq \ell$ simultaneously,

$$\left|\nu_{u,i|S} - \widehat{\nu}_{u,i|S}\right| < \varepsilon.$$

We let A denote the event $A(L, \tau/2)$.

The proof of the following technical lemma is left to an appendix.

Lemma 12. *Fix a set S with $|S| \leq \ell$ and suppose that for any set $T \supseteq S$ with $|T \setminus S| \leq r$, that*

$$|\widehat{\mathbf{Pr}}(X_T = x_T) - \mathbf{Pr}(X_T = x_T)| < \sigma.$$

If $\sigma \leq \varepsilon K^{-\ell} \frac{\delta^\ell}{5}$ then for any I with $|I| \leq r - 1$,

$$\left|\nu_{u,i|S} - \widehat{\nu}_{u,i|S}\right| < \varepsilon.$$

Lemma 13. *Fix ℓ, ε and $\omega > 0$. If the number of samples satisfies*

$$m \geq \frac{15K^{2\ell}}{\varepsilon^2 \delta^{2\ell}} \Big(\log(1/\omega) + \log(\ell + r) + (\ell + r) \log(nK) + \log 2 \Big)$$

then $\mathbf{Pr}(A(\ell, \varepsilon)) \geq 1 - \omega$.

Proof of Lemma 13. Fix ℓ, ε and $\omega > 0$. Let m denote the number of samples. By Hoeffding's inequality, for any set T,

$$\mathbf{Pr}[|\widehat{\mathbf{Pr}}(X_T = x_T) - \mathbf{Pr}(X_T = x_T)| > \sigma] \leq 2\exp(-2\sigma^2 m)$$

and taking the union bound over all possibly x_T for T with $|T| \leq \ell + r$, of which there are at most

$$\sum_{i=1}^{\ell+r} \binom{n}{i} K^i \leq \sum_{i=1}^{\ell+r} (nK)^i \leq (\ell + r)(nK)^{\ell+r}$$

many, we find the probability that $|\widehat{\mathbf{Pr}}(X_T = x_T) - \mathbf{Pr}(X_T = x_T)| > \sigma$ for any such T is at most

$$(\ell + r)(nK)^{\ell+r} 2 \exp(-2\sigma^2 m).$$

Therefore taking

$$m \geq \frac{\log(1/\omega) + \log(\ell + r) + (\ell + r) \log(nK) + \log 2}{2\sigma^2} \tag{4.10}$$

ensures this probability is at most ω.

Now applying Lemma 12 and substituting $\sigma = \varepsilon K^{-\ell}\frac{\delta^{\ell}}{5}$ into (4.10), we see that the result holds if

$$m \geq \frac{15K^{2\ell}}{\varepsilon^2\delta^{2\ell}}\Big(\log(1/\omega) + \log(\ell + r) + (\ell + r)\log(nK) + \log 2\Big).$$

□

Lemma 14. *Assume that the event A holds. Then every time a node i is added to S in Step 2 of the algorithm, the mutual information $I(X_u; X_S)$ increases by at least $\tau^2/8$.*

Proof. For a particular iteration of Step 2, let I denote the newly added set of nodes, and S the set of candidate neighbors before adding I. Then we must show for $Q = \tau^2/8$ that

$$I(X_u; X_{S\cup\{I\}}) \geq I(X_u; X_S) + Q$$

which by the chain rule for expectation is equivalent to

$$I(X_u; X_I|X_S) \geq Q.$$

Applying Lemma 4 and the fact that event A holds, we see

$$\sqrt{\frac{1}{2}\cdot I(X_u; X_I|X_S)} \geq \frac{1}{2}\nu_{u,I|S} \geq \frac{1}{2}\left(\widehat{\nu}_{u,i|S} - \tau/2\right).$$

Thus the algorithm only adds node i to S if $\widehat{\nu}_{u,i|S} \geq \tau$, so the chain of inequalities implies that

$$I(X_u; X_i|X_S) \geq \frac{1}{2}(\tau - \tau/2)^2 = \tau^2/8.$$

□

Lemma 15. *If event A holds then at the end of Step 2, S contains all of the neighbors of u.*

Proof. Step 2 ended either because $|S| > L$ or because there was no set of nodes

$I \subset [n] \setminus S$ with $\widehat{\nu}_{u,I|S} > \tau$. First we rule out the former possibility. Whenever a new element is added to S, the quantity $I(X_u; X_S)$ increases by at least $\tau^2/8$. But

$$I(X_u; X_S) \leq H(X_u) \leq \log K$$

because X_u takes on at most K states. Thus if $|S| > L$ then

$$\log K \geq I(X_u; X_S) > L(\tau^2/8) = \log K$$

which gives a contradiction.

Thus at the end of Step 2 we must have that there is no set of nodes $I \subset [n] \setminus S$ with $\widehat{\nu}_{u,I|S} > \tau$. Suppose for the purposes of contradiction that S does not contain all of the neighbors of u. Then by Theorem 27, there exists a subset of the neighbors such that $\nu_{u,I|S} \geq C'(\gamma, k, \alpha) = 2\tau$, and because event A holds we know $\widehat{\nu}_{u,I|S} > 2\tau - \tau/2 > \tau$, which gives us our contradiction and completes the proof of the lemma. □

Lemma 16. *If event A holds and if at the start of Step 3 S contains all neighbors of u, then at the end of Step 3 the remaining set of nodes are exactly the neighbors of u.*

Proof. If $A(\ell)$ holds, then during Step 3,

$$\widehat{\nu}_{u,i|S\setminus\{i\}} < \nu_{u,i|S} + \tau/2 \leq \sqrt{\frac{1}{2} I(X_u; X_i|X_S)} + \tau/2 = \tau/2$$

for all nodes i that are not neighbors of u. Thus all such nodes are pruned. Furthermore, by Theorem 27, $\widehat{\nu}_{u,i|S\setminus\{i\}} > \nu_{u,i|S\setminus\{i\}} - \tau/2 \geq 2\tau - \tau/2 = 3\tau/2$ for all neighbors of u and thus no neighbor is pruned. This completes the proof. □

Recall that $\gamma \leq \beta r D^r$, $\delta = e^{-2\gamma}/K$, $(C'(\gamma, K, \alpha))^{-1} = O(\frac{K^{r+1} r^{2r}}{\alpha^2 \delta^{2D}} D^{r-1} \gamma e^{-2\gamma})$ and $L = O(C'(\gamma, K, \alpha)^{-2})$.

Theorem 29. *Fix $\omega > 0$. Suppose we are given m samples from an α, β-non-degenerate graphical model with r-order interactions where the underlying graph has* *maximum degree at most D and each node takes on at most K states. Suppose that*

$$m \geq \frac{60K^{2L}}{\tau^2\delta^{2L}}\Big(\log(1/\omega)+\log(L+r)+(L+r)\log(nK)+\log 2\Big).$$

Then with probability at least $1-\omega$, MrfNbhd *when run starting from each node u recovers the correct neighborhood of u, and thus recovers the underlying graph G. Furthermore, each run of the algorithm takes $O(mLn^r)$ time.*

Proof. Set $\ell = L$ and $\varepsilon = \tau/2$ in Lemma 13. Then event A occurs with probability at least $1-\omega$ for our choice of m. Now by Lemma 15 and Lemma 16 the algorithm returns the correct set of neighbors of u for every node u.

To analyze the running time, observe that when running algorithm MrfNbhd at a single node u, the bottleneck is Step 2, in which there are at most L steps and in each step the algorithm must loop over all subsets of the vertices in $[n]\setminus S$ of size $r-1$, of which there are $\sum_{\ell=1}^{r-1}\binom{n}{\ell} = O(n^{r-1})$ many. Running the algorithm at all nodes thus takes $O(mLn^r)$ time. □

Remark 30. Note that when we plug in the values of γ and δ we get that the overall sample complexity of our algorithm in terms of D and r is doubly exponential in D^r.

4.4.2 Learning with Bounded Queries

In many situations, it is too expensive to obtain full samples from a graphical model (e.g. this could involve needing to measure every potential symptom of a patient). Here we consider a model where we are allowed only partial observations in the form of a C-bounded query:

Definition 31. A *C-bounded query* to a graphical model is specified by a set S with $|S| \leq C$ and we observe X_S

Our algorithm MrfNbhd can be made to work with C-bounded queries instead of full observations by a simple change: instead of estimating all of the terms $\widehat{\nu}_{u,I|S}$ jointly from samples, we estimate each one individually by querying a fresh batch of

m' samples of $\{u\} \cup I \cup S$ every time the algorithm requires $\widehat{\nu}_{u,I|S}$. First we make an elementary observation about MRFNBHD:

Observation 1. *In Step 2,* MRFNBHD *only needs $\widehat{\nu}_{u,I|S}$ for all I with $|I| \leq r-1$. Similarly at Step 3,* MRFNBHD *only needs $\widehat{\nu}_{u,i|S\setminus i}$ for each $i \in S$.*

Thus the number of distinct terms $\widehat{\nu}_{u,I|S}$ which MRFNBHD needs is at most $L(r-1)n^{r-1}$ for Step 2 and R for Step 3, which in total is at most Lrn^{r-1}.

Lemma 17. *Fix a node u, a set S with $\ell = |S|$, a set I with $|I| \leq r-1$ and fix ε and $\omega > 0$. If the number of samples we observe of $X_{S\cup I\cup\{u\}}$ satisfies*

$$m' \geq \frac{15K^{2\ell}}{\varepsilon^2\delta^{2\ell}}\Big(\log(1/\omega) + \log(\ell+r) + (\ell+r)\log(nK) + \log 2\Big)$$

then

$$|\nu_{u,I|S} - \widehat{\nu}_{u,I|S}| < \varepsilon$$

with probability at least $1-\omega$.

Proof. This follows by the same Hoeffding and union bound as in proof of Lemma 13. □

Theorem 32. *Fix an α,β-non-degenerate graphical model with r-order interactions where the underlying graph has maximum degree at most D and each node takes on at most K states. The bounded queries modification to the algorithm returns the correct neighborhood of every vertex u using $m'Lrn^r$-bounded queries of size at most $L+r$ where*

$$m' = \frac{60K^{2L}}{\tau^2\delta^{2L}}\Big(\log(Lrn^r/\omega) + \log(L+r) + (L+r)\log(nK) + \log 2\Big),$$

with probability at least $1-\omega$.

Proof. Invoking Lemma 17 with $\omega' = \frac{\omega}{Lrn^r}$, $\varepsilon = \tau/2$ and $\ell = L$, we get that each query to $\widehat{\nu}_{u,I|S}$ fails (i.e. is wrong by at least $\tau/2$) with probability at most $\frac{\omega}{Lrn^r}$. We observed that Algorithm MRFNBHD makes at most Lrn^{r-1} queries of the form, $\widehat{\nu}_{u,I|S}$.

Therefore, by a union bound, with probability at least $1-\omega/n$, the bounded queries algorithm answers all of those queries to within tolerance $\tau/2$.

Now it follows as in Theorem 29 that the algorithm returns the correct neighborhood of node u with probability at least $1-\omega/n$, and taking the union bound over all nodes u it follows that the algorithm recovers the correct neighborhood of all nodes with probability at least $1-\omega$. This completes the proof. □

4.4.3 Learning with Random Erasures

Here we consider another variant where we do not observe full samples from a graphical model. Instead we observe partial samples where the state of each node is revealed independently with probability p and is otherwise replaced with a '?', and the choice of which nodes to reveal is independent of the sample. We can apply our algorithm in this setting, as follows.

Lemma 18. *With probability at least $1-\varepsilon$, if we take $N\frac{\ell \log n + \log \ell + \log N/\varepsilon}{p^2}$ samples then we will see each set S at least N times for every $|S| \le \ell$.*

Proof. Each sample has at least a p^ℓ chance of being observed, and there are at most ℓn^ℓ many different sets S. So by a union bound,

$$Pr[\text{exists unobserved } S \text{ after } t \text{ steps}] \le n^\ell (1-p^\ell)^t \le \varepsilon/N$$

if we take $t = \frac{\ell \log n + \log \ell + \log N/\varepsilon}{p^2}$. Repeating this N times, we see that with

$$Nt = N\frac{\ell \log n + \log \ell + \log N/\varepsilon}{p^2}$$

many samples, we see every S at least N times with probability at least $1-\varepsilon$. □

Lemma 19. *Fix ℓ, ε and $\omega > 0$. If the number of samples satisfies*

$$m \ge N\frac{\ell \log N + \log \ell + \log 2N/\omega}{p^2}$$

where

$$N = \frac{15K^{2\ell}}{\varepsilon^2\delta^{2\ell}}\Big(\log(2/\omega) + \log(\ell + r) + (\ell + r)\log(nK) + \log 2\Big)$$

then $\mathbf{Pr}(A(\ell,\varepsilon)) \geq 1 - \omega$.

Proof. Observe by Lemma 18, taking $\varepsilon = \omega/2$ that with probability at least $1-\omega/2$, for every set S with $|S| \leq \ell$ we see at least N samples revealing all of the members of S. Condition on this event; now the proof is exactly the same as Lemma 13 taking $\omega' = \omega/2$. Applying Hoeffding and Lemma 12 and taking the union bound, we see that event A holds with probability at least $\omega/2$. Therefore the total probability A occurs is at least $1 - \omega/2 - \omega/2 = 1 - \omega$. □

Theorem 33. *Fix $\omega > 0$. Suppose we are given m samples from an α, β-non-degenerate graphical model with r-order interactions where the underlying graph has maximum degree at most D and each node takes on at most K states. Suppose that*

$$m \geq N\frac{\ell \log n + \log L + \log 2N/\omega}{p^2}$$

where

$$N = \frac{60K^{2L}}{\tau^2\delta^{2L}}\Big(\log(2/\omega) + \log(L + r) + (L + r)\log(nK) + \log 2\Big).$$

Then with probability at least $1-\omega$, MrfNbhd *when run starting from each node u recovers the correct neighborhood of u, and thus recovers the underlying graph G. Furthermore, each run of the algorithm takes $O(mLn^r)$ time.*

Proof. By Lemma 19, given our assumption on m the event A occurs with probability at least $1-\omega$. Conditioned on event A, the algorithm returns the correct answer by the same argument as Theorem 29. □

4.5 Interaction screening

In 2016, Vuffray et al [58] discovered a convex programming approach to Ising model structure learning. Here we will exposit their method while simultaneously generalizing it to arbitrary graphical models.

Definition 14. In a graphical model, define the local energy $H_{\theta,u}(x)$ at a node u to be the contribution to the total energy $H_\theta(x)$ added by hyperedges that include u:

$$H_{\theta,u}(x) := -\sum_{\ell=1}^{r} \sum_{(i_1,i_2,\ldots,i_\ell)\ni u} \theta^{i_1\cdots i_\ell}(x_{i_1},\ldots,x_{i_\ell})$$

Take care that the total energy $H_\theta(x)$ is not the sum of the local energies $H_{\theta,u}(x)$, since each hyperedge $\theta^{i_1\cdots i_\ell}(x_{i_1},\ldots,x_{i_\ell})$ contributes to the energy at multiple different nodes.

Vuffray et al use the local energy in an Ising model to define a convex optimization problem. Here we generalize the approach to arbitrary graphical models.

Definition 15. Given a guess θ for the parameters of a graphical model, the Interaction Screening Objective $\mathcal{S}_u(\theta)$ is defined as the expected value of $\exp(H_{\theta,u}(x))$. In the context of sampling, use the notation $\overline{\mathcal{S}}_u(\theta)$ for the empirical mean of $\exp(H_{\theta,u}(x))$.

Observe $\mathcal{S}_u(\theta)$ is convex because it is a weighted sum of convex functions. Miraculously, it turns out the minimizer of this convex function is the true value of the graphical model's parameters at all cliques that include u. (Note that $\mathcal{S}_u(\theta)$ only depends on the parameters of θ for interactions involving u, so by finding its minimizer we learn the local structure of the graphical model around u.)

Theorem 34. *For a graphical model with energy parameter $\theta^\star$, the function $\mathcal{S}_u$ is minimized, among all θ in canonical form, by $\theta^\star$.*

Proof. Since $\mathcal{S}_u$ is convex, it suffices to show the gradient at $\theta^\star$ is zero. This requires some careful setup. There are very many partial derivatives $\frac{\partial}{\partial\theta^{i_1\cdots i_\ell}(x_1,\ldots,x_\ell)}\mathcal{S}_u(\theta^\star)$.

The partial derivatives with $u \notin (i_1,\ldots,i_\ell)$ must be zero, since $\mathcal{S}_u(\theta)$ does not even depend on these tensors $\theta^{i_1\cdots i_\ell}$.

But when $u \in (i_1,\ldots,i_\ell)$, the partial derivatives are *not* in general zero. Fortunately this does not spoil the theorem, as we care only about those θ in canonical form. Our task is to show that the gradient is zero in all directions that keep θ in canonical form.

Take an arbitrary tensor $\theta^{i_1 \cdots i_\ell}$ with $u \in (i_1, \ldots, i_\ell)$. We aim to show that for any direction of change in $\theta^{i_1 \cdots i_\ell}$ that keeps it in canonical form, the resulting partial derivative of $\mathcal{S}_u(\theta^\star)$ is zero. It will be convenient to slightly abuse partial derivative notation so that, for example, $\partial_{x,y}\, x^2 + y^3 + z = 2x\partial x + 3y^2 \partial y$.

$$\begin{aligned}
&\partial_{\theta^{i_1 \cdots i_\ell}} \mathcal{S}_u(\theta^\star) \\
&= \partial_{\theta^{i_1 \cdots i_\ell}} \mathbb{E}_x \left[\exp(H_{\theta,u}(x))\right] \\
&= \sum_x \mathbf{Pr}(\text{model is in state } x) \cdot \partial_{\theta^{i_1 \cdots i_\ell}} \exp(H_{\theta,u}(x)) \\
&= \sum_x \mathbf{Pr}(\text{model is in state } x) \cdot \exp(H_{\theta,u}(x)) \cdot \partial\theta^{i_1 \cdots i_\ell}(x_{i_1}, \ldots, x_{i_\ell}) \\
&= \sum_x \exp(-H_{\theta^\star}(x) - C + H_{\theta^\star,u}(x)) \cdot \partial\theta^{i_1 \cdots i_\ell}(x_{i_1}, \ldots, x_{i_\ell})
\end{aligned}$$

Behold: by the definition of the local energy $H_{\theta^\star,u}(x)$, every term that involves the node u cancels out in the exponent $-H_{\theta^\star}(x) - C + H_{\theta^\star,u}(x)$. This means that two different states x, x' that differ only at node u have equal values of $\exp(-H_{\theta^\star}(x) - C + H_{\theta^\star,u}(x))$. This motivates us to split the sum over states x into a double sum: one over the state x_u at u, and the other over the state $x_{\sim u}$ everywhere else.

$$\begin{aligned}
&\sum_{x_{\sim u}} \sum_{x_u} \exp(-H_{\theta^\star}(x) - C + H_{\theta^\star,u}(x)) \cdot \partial\theta^{i_1 \cdots i_\ell}(x_{i_1}, \ldots, x_{i_\ell}) \\
&= \sum_{x_{\sim u}} \exp(-H_{\theta^\star}(x) - C + H_{\theta^\star,u}(x)) \sum_{x_u} \partial\theta^{i_1 \cdots i_\ell}(x_{i_1}, \ldots, x_{i_\ell})
\end{aligned}$$

The inner sum is always 0, by the definition of canonical form. Thus the gradient is 0 in any direction that preserves canonical form. This completes the proof.

□

By finding the minimizer of $\mathcal{S}_u$ for each node u, one recovers the full structure of the graphical model. The canonical form constraint is no hindrance to gradient descent, since it is just a set of linear constraints. Alas, there is one obstacle: in the context of sample complexity, we can only estimate $\mathcal{S}_u$ using the empirical mean $\overline{\mathcal{S}}_u$. Nevertheless, Vuffray et al prove that, with the addition of an appropriately scaled ℓ_1 regularization term, one can use the convex program to reconstruct bounded-degree Ising models with high probability given enough samples. It is unclear whether their error bounds can generalize to arbitrary graphical models.

Observe that this convex programing method *cannot* solve the bounded query or random erasure variants of the problem.

Appendix A

Lower Bounds on the Condition Number in the Hyperbolic Plane

In Euclidean space, the function $f(x) = ||x||^2$ is 1-smooth and 1-strongly convex at every point. However, as we will show, in the hyperbolic plane geodesically convex functions always have a condition number that depends on the radius:

Theorem 35. *If f is a β-smooth, α-strongly convex function defined in a hyperbolic disk of radius r, then $\beta/\alpha \geq \Omega(r)$.*

Proof. First we will give the intuition for the proof. Consider a geodesic that dips a distance of 1 into the disk of radius r (see the picture below). On the one hand, due to α-strong convexity, the value of f must vary a large amount along this geodesic. But on the other hand, this geodesic is short, so by β-smoothness f cannot vary much. These two properties will give us a lower bound on the condition number.

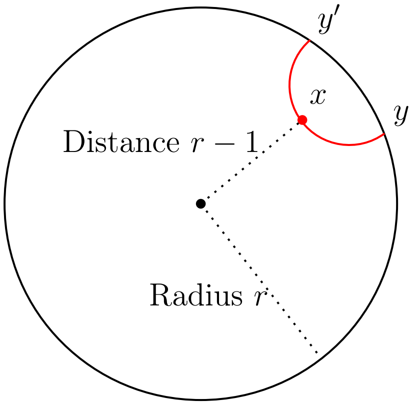
Now we proceed to the formal proof. Of all points at distance $r - 1$ from the center of the disk, let x be one at which f is minimal. Without loss of generality suppose that $f = 0$ at the center of the disk. By convexity and the minimality of x, we deduce that

$$f(y) \geq \frac{r}{r-1} f(x)$$

for all y on the circumference of the disk. By α-strong convexity, $f(x) \geq \Omega(\alpha r^2)$. Now draw a geodesic through x, as pictured, perpendicular to the geodesic between the center of the disk and x. This geodesic intersects the disk at two points y and y'. By β-smoothness,

$$\frac{1}{2}(f(y)+f(y')) \leq f(x) + O(\beta d(y,y')^2).$$

Finally, in hyperbolic geometry, the distance $d(y,y')$ is $O(1)$. See Lemma 20 for an explicit calculation justifying this. Finally combining the three inequalities in the previous paragraph gives $\beta/\alpha \geq \Omega(r)$, as desired. □



Lemma 20. *A hyperbolic triangle with two sides of length r, whose altitude between those sides has length $r-1$, has a third side of length $O(1)$.*

Proof. This is a straightforward calculation using formulas from [33]. Let c denote the length of the third side and A denote the measure of either angle adjacent to side c. (Those angles are equal because the triangle is isoceles.) The hyperbolic law of cosines from [33] gives $\cos A = \frac{(-1+\cosh c)\cosh r}{\sinh c \sinh r}$. The altitude length formula gives $\sin A = \frac{\sinh(r-1)}{\sinh r}$. Using $1-\sin^2 = \cos^2$ gives:

$$1 - \frac{\sinh^2(r-1)}{\sinh^2 r} = \frac{(-1+\cosh c)^2 \cosh^2 r}{\sinh^2 c \sinh^2 r}.$$

Rearranging the above expression, we get:

$$\left(1 - \frac{\sinh^2(r-1)}{\sinh^2 r}\right)\frac{\sinh^2 r}{\cosh^2 r} = \frac{(-1+\cosh c)^2}{\sinh^2 c}.$$

As $r \to \infty$ the left-hand side tends to $1 - \frac{1}{e^2} \approx 0.86$. Then solving the right-hand side for c gives a solution, unique in the reals, that is approximately 3.31, which is $O(1)$

as desired. □

Appendix B

Graphical models: proof of Lemma 12

Proof. Observe the left hand side of our desired inequality is bounded by

$$\begin{aligned} E_{R,G}\Big|&\widehat{\mathbb{E}}_{X_S}[|\widehat{\mathbf{Pr}}(X_u = R, X_I = G|X_S) - \widehat{\mathbf{Pr}}(X_u = R|X_S)\widehat{\mathbf{Pr}}(X_I = G|X_S)|] \\ &- \mathbb{E}_{X_S}[|\,\mathbf{Pr}(X_u = R, X_I = G|X_S) - \mathbf{Pr}(X_u = R|X_S)\,\mathbf{Pr}(X_I = G|X_S)|]\Big|. \end{aligned}$$

So it suffices if we can bound for every R and G

$$\begin{aligned}
&\Big|\widehat{\mathbb{E}}_{X_S}[|\widehat{\mathbf{Pr}}(X_u = R, X_I = G|X_S) - \widehat{\mathbf{Pr}}(X_u = R|X_S)\widehat{\mathbf{Pr}}(X_I = G|X_S)|] \\
&\qquad - \mathbb{E}_{X_S}[|\,\mathbf{Pr}(X_u = R, X_I = G|X_S) - \mathbf{Pr}(X_u = R|X_S)\,\mathbf{Pr}(X_I = G|X_S)|]\Big| \\
&= \Big|\sum_{x_S} |\widehat{\mathbf{Pr}}(X_u = R, X_I = G, X_S = x_S) - \widehat{\mathbf{Pr}}(X_u = R|X_S = x_S)\widehat{\mathbf{Pr}}(X_I = G, X_S = x_S)| \\
&\qquad - |\,\mathbf{Pr}(X_u = R, X_i = G, X_S = x_S) - \mathbf{Pr}(X_u = R|X_S = x_S)\,\mathbf{Pr}(X_I = G, X_S = x_S)|\Big| \\
&\leq \sum_{x_S} \Big||\widehat{\mathbf{Pr}}(X_u = R, X_I = G, X_S = x_S) - \widehat{\mathbf{Pr}}(X_u = R|X_S = x_S)\widehat{\mathbf{Pr}}(X_I = G, X_S = x_S)| \\
&\qquad - |\,\mathbf{Pr}(X_u = R, X_I = G, X_S = x_S) - \mathbf{Pr}(X_u = R|X_S = x_S)\,\mathbf{Pr}(X_I = G, X_S = x_S)|\Big| \\
&\leq \sum_{x_S} \Big|\widehat{\mathbf{Pr}}(X_u = R, X_I = G, X_S = x_S) - \widehat{\mathbf{Pr}}(X_u = R|X_S = x_S)\widehat{\mathbf{Pr}}(X_I = G, X_S = x_S) \\
&\qquad - \mathbf{Pr}(X_u = R, X_I = G, X_S = x_S) + \mathbf{Pr}(X_u = R|X_S = x_S)\,\mathbf{Pr}(X_I = G, X_S = x_S)\Big| \\
&\leq \sum_{x_S} |\widehat{\mathbf{Pr}}(X_u = R, X_I = G, X_S = x_S) - \mathbf{Pr}(X_u = R, X_I = G, X_S = x_S)| \\
&\qquad + \sum_{x_S} |\widehat{\mathbf{Pr}}(X_u = R|X_S = x_S)\widehat{\mathbf{Pr}}(X_I = G, X_S = x_S) - \mathbf{Pr}(X_u = R|X_S)\,\mathbf{Pr}(X_I = G, X_S = x_S)| \\
&\leq K^{|S|}\sigma + \sum_{x_S} |\widehat{\mathbf{Pr}}(X_u = R|X_S = x_S)\widehat{\mathbf{Pr}}(X_I = G, X_S = x_S) - \\
&\qquad\qquad\qquad \mathbf{Pr}(X_u = R|X_S = x_S)\,\mathbf{Pr}(X_I = G, X_S = x_S)|.
\end{aligned}$$

To bound the second term, observe

$$\begin{aligned}
&|\widehat{\mathbf{Pr}}(X_u = R|X_S = x_S)\widehat{\mathbf{Pr}}(X_I = G, X_S = x_S) - \mathbf{Pr}(X_u = R|X_S)\,\mathbf{Pr}(X_I = G, X_S = x_S)| \\
&\quad \leq \widehat{\mathbf{Pr}}(X_u = R|X_S = x_S)|\widehat{\mathbf{Pr}}(X_I = G, X_S = x_S) - \mathbf{Pr}(X_I = G, X_S = x_S)| \\
&\qquad\qquad + \mathbf{Pr}(X_I = G, X_S = x_S)|\widehat{\mathbf{Pr}}(X_u = R|X_S = x_S) - \mathbf{Pr}(X_u = R|X_S)| \\
&\quad \leq |\widehat{\mathbf{Pr}}(X_I = G, X_S = x_S) - \mathbf{Pr}(X_I = G, X_S = x_S)| + |\widehat{\mathbf{Pr}}(X_u = R|X_S = x_S) - \mathbf{Pr}(X_u = R|X_S)| \\
&\quad \leq \sigma + |\widehat{\mathbf{Pr}}(X_u = R|X_S = x_S) - \mathbf{Pr}(X_u = R|X_S)|
\end{aligned}$$

and furthermore

$$|\widehat{\mathbf{Pr}}(X_u = R|X_S = x_S) - \mathbf{Pr}(X_u = R|X_S = x_S)| = \left| \frac{\widehat{\mathbf{Pr}}(X_u = R, X_S = x_S)}{\widehat{\mathbf{Pr}}(X_S = x_S)} - \frac{\mathbf{Pr}(X_u = R, X_S = x_S)}{\mathbf{Pr}(X_S = x_S)} \right|$$

which in turn we can bound as

$$\begin{aligned}
&\leq \left| \frac{\widehat{\mathbf{Pr}}(X_u = R, X_S = x_S)}{\widehat{\mathbf{Pr}}(X_S = x_S)} - \frac{\mathbf{Pr}(X_u = R, X_S = x_S)}{\widehat{\mathbf{Pr}}(X_S = x_S)} \right| \\
&\quad + \left| \frac{\mathbf{Pr}(X_u = R, X_S = x_S)}{\widehat{\mathbf{Pr}}(X_S = x_S)} - \frac{\mathbf{Pr}(X_u = R, X_S = x_S)}{\mathbf{Pr}(X_S = x_S)} \right| \\
&\leq \frac{\sigma}{\delta^{|S|}} + \mathbf{Pr}(X_u = R, X_S = x_S) \left| \frac{\mathbf{Pr}(X_S = x_S) - \widehat{\mathbf{Pr}}(X_S = x_S)}{\widehat{\mathbf{Pr}}(X_S = x_S)\,\mathbf{Pr}(X_S = x_S)} \right| \leq \frac{\sigma}{\delta^{|S|}} + \frac{\sigma}{\delta^{|S|} - \sigma}.
\end{aligned}$$

Finally, if $\sigma < \varepsilon K^{-\ell} \frac{\delta^\ell}{5}$ then because $|S| \leq \ell$ and $\sigma < \delta^\ell/5 < \delta^\ell/2$

$$\begin{aligned}
K^{|S|}\sigma + \sum_{x_S} \left(\sigma + \frac{\sigma}{\delta^{|S|}} + \frac{\sigma}{\delta^{|S|} - \sigma} \right) &= K^{|S|}\sigma \left(2 + \frac{1}{\delta^{|S|}} + \frac{1}{\delta^{|S|} - \sigma} \right) \\
&< K^{|S|}\sigma \left(\frac{2}{\delta^{|S|}} + \frac{1}{\delta^{|S|}} + \frac{2}{\delta^{|S|}} \right) < \varepsilon.
\end{aligned}$$

This completes the proof. □

Bibliography

[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds.* Princeton University Press, 2009.

[2] Kwangjun Ahn and Suvrit Sra. From nesterov's estimate sequence to riemannian acceleration. pages 84–118, 2020.

[3] Zeyuan Allen-Zhu, Ankit Garg, Yuanzhi Li, Rafael Oliveira, and Avi Wigderson. Operator scaling via geodesically convex optimization, invariant theory and polynomial identity testing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 172–181, 2018.

[4] Carlos Améndola, Kathlén Kohn, Philipp Reichenbach, and Anna Seigal. Invariant theory and scaling algorithms for maximum likelihood estimation. *arXiv preprint arXiv:2003.13662*, 2020.

[5] Franck Barthe. On a reverse form of the brascamp-lieb inequality. *Inventiones mathematicae*, 134(2):335–361, 1998.

[6] Bernhard G Bodmann and Peter G Casazza. The road to equal-norm parseval frames. *Journal of Functional Analysis*, 258(2):397–420, 2010.

[7] Guy Bresler. Efficiently learning ising models on arbitrary graphs. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 771–782, 2015.

[8] Guy Bresler, Elchanan Mossel, and Allan Sly. Reconstruction of markov random fields from samples: Some observations and algorithms. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 343–356. Springer, 2008.

[9] Stephen G Brush. History of the lenz-ising model. *Reviews of modern physics*, 39(4):883, 1967.

[10] Peter Bürgisser, Cole Franks, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. Efficient algorithms for tensor scaling, quantum marginals, and moment polytopes. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 883–897. IEEE, 2018.

[11] Peter Bürgisser, Cole Franks, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. Towards a theory of non-commutative optimization: geodesic first and second order methods for moment maps and polytopes. *arXiv preprint arXiv:1910.12375*, 2019.

[12] Peter Bürgisser, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. Alternating minimization, scaling algorithms, and the null-cone problem from invariant theory. *arXiv preprint arXiv:1711.08039*, 2017.

[13] Léopold Cambier and P-A Absil. Robust low-rank matrix completion by riemannian optimization. *SIAM Journal on Scientific Computing*, 38(5):S440–S460, 2016.

[14] 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.

[15] James W Cannon, William J Floyd, Richard Kenyon, Walter R Parry, et al. Hyperbolic geometry. *Flavors of geometry*, 31:59–115, 1997.

[16] Peter G Casazza. The kadison–singer and paulsen problems in finite frame theory. In *Finite Frames*, pages 381–413. Springer, 2013.

[17] Peter G Casazza and Jameson Cahill. The paulsen problem in operator theory. *arXiv preprint arXiv:1102.2344*, 2011.

[18] Peter G Casazza, Matthew Fickus, and Dustin G Mixon. Auto-tuning unit norm frames. *Applied and Computational Harmonic Analysis*, 32(1):1–15, 2012.

[19] Anoop Cherian and Suvrit Sra. Riemannian dictionary learning and sparse coding for positive definite matrices. *IEEE transactions on neural networks and learning systems*, 28(12):2859–2871, 2016.

[20] Christopher Criscitiello and Nicolas Boumal. Efficiently escaping saddle points on manifolds. In *Advances in Neural Information Processing Systems*, pages 5987–5997, 2019.

[21] Christopher Criscitiello and Nicolas Boumal. Negative curvature obstructs acceleration for geodesically convex optimization, even with exact first-order oracles. *arXiv preprint arXiv:2111.13263*, 2021.

[22] Gautamd Dasarathy, Aarti Singh, Maria-Florina Balcan, and Jong H Park. Active learning algorithms for graphical model selection. In *Artificial Intelligence and Statistics*, pages 1356–1364. PMLR, 2016.

[23] Olivier Devolder, François Glineur, Yurii Nesterov, et al. First-order methods with inexact oracle: the strongly convex case. *CORE Discussion Papers*, 2013016:47, 2013.

[24] Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[25] Ricardo Ferreira and Joao Xavier. Hessian of the riemannian squared distance function on connected locally symmetric spaces with applications. In *Controlo 2006, 7th Portuguese conference on automatic control*, volume 2. Citeseer, 2006.

[26] Cole Franks and Ankur Moitra. Rigorous guarantees for tyler's m-estimator via quantum expansion. *arXiv preprint arXiv:2002.00071*, 2020.

[27] Eva García-Martín, Crefeda Faviola Rodrigues, Graham Riley, and Håkan Grahn. Estimation of energy consumption in machine learning. *Journal of Parallel and Distributed Computing*, 134:75–88, 2019.

[28] Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. Algorithmic and optimization aspects of brascamp-lieb inequalities, via operator scaling. *Geometric and Functional Analysis*, 28(1):100–145, 2018.

[29] Leonid Gurvits. Classical complexity and quantum entanglement. *Journal of Computer and System Sciences*, 69(3):448–484, 2004.

[30] Moritz Hardt and Ankur Moitra. Algorithms and hardness for robust subspace recovery. In *Conference on Learning Theory*, pages 354–375. PMLR, 2013.

[31] Roderick B Holmes and Vern I Paulsen. Optimal frames for erasures. *Linear Algebra and its Applications*, 377:31–51, 2004.

[32] Reshad Hosseini and Suvrit Sra. Matrix manifold optimization for gaussian mixtures. *Advances in Neural Information Processing Systems*, 28:910–918, 2015.

[33] Svante Janson. Euclidean, spherical and hyperbolic trigonometry. *Notes*, 2015.

[34] Hiroyuki Kasai, Pratik Jawanpuria, and Bamdev Mishra. Riemannian adaptive stochastic gradient algorithms on matrix manifolds. 97:3262–3271, 2019.

[35] Masoud Badiei Khuzani and Na Li. Stochastic primal-dual method on riemannian manifolds of bounded sectional curvature. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 133–140. IEEE, 2017.

[36] Ross Kindermann. Markov random fields and their applications. *American mathematical society*, 1980.

[37] Adam Klivans and Raghu Meka. Learning graphical models using multiplicative weights. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 343–354. IEEE, 2017.

[38] Adam Klivans and Raghu Meka. Learning graphical models using multiplicative weights. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 343–354. IEEE, 2017.

[39] Eryk Kopczyński, Dorota Celińska, and Marek Čtrnáct. Hyperrogue: Playing with hyperbolic geometry. In David Swart, Carlo H. Séquin, and Kristóf Fenyvesi, editors, *Proceedings of Bridges 2017: Mathematics, Art, Music, Architecture, Education, Culture*, pages 9–16, Phoenix, Arizona, 2017. Tessellations Publishing.

[40] Tsz Chiu Kwok, Lap Chi Lau, Yin Tat Lee, and Akshay Ramachandran. The paulsen problem, continuous operator scaling, and smoothed analysis. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 182–189, 2018.

[41] David Martinez-Rubio. Acceleration in hyperbolic and spherical spaces. *arXiv preprint arXiv:2012.03618*, 2020.

[42] Grégoire Montavon, Geneviève Orr, and Klaus-Robert Müller. *Neural networks: tricks of the trade*, volume 7700. springer, 2012.

[43] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[44] Ryan O'Donnell. *Analysis of boolean functions.* Cambridge University Press, 2014.

[45] Joseph M Renes, Robin Blume-Kohout, Andrew J Scott, and Carlton M Caves. Symmetric informationally complete quantum measurements. *Journal of Mathematical Physics*, 45(6):2171–2180, 2004.

[46] Narayana P Santhanam and Martin J Wainwright. Information-theoretic limits of selecting binary graphical models in high dimensions. *IEEE Transactions on Information Theory*, 58(7):4117–4134, 2012.

[47] Hiroyuki Sato, Hiroyuki Kasai, and Bamdev Mishra. Riemannian stochastic variance reduced gradient algorithm with retraction and vector transport. *SIAM Journal on Optimization*, 29(2):1444–1472, 2019.

[48] Thomas Strohmer and Robert W Heath Jr. Grassmannian frames with applications to coding and communication. *Applied and computational harmonic analysis*, 14(3):257–275, 2003.

[49] Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery over the sphere ii: Recovery by riemannian trust-region method. *IEEE Transactions on Information Theory*, 63(2):885–914, 2016.

[50] Yue Sun, Nicolas Flammarion, and Maryam Fazel. Escaping from saddle points on riemannian manifolds. In *Advances in Neural Information Processing Systems*, pages 7276–7286, 2019.

[51] Mingkui Tan, Ivor W Tsang, Li Wang, Bart Vandereycken, and Sinno Jialin Pan. Riemannian pursuit for big matrix recovery. In *International Conference on Machine Learning*, pages 1539–1547, 2014.

[52] Tiffany M Tang and Genevera I Allen. Integrated principal components analysis. *arXiv preprint arXiv:1810.00832*, 2018.

[53] Nilesh Tripuraneni, Nicolas Flammarion, Francis Bach, and Michael I Jordan. Averaging stochastic gradient descent on riemannian manifolds. pages 650–687, 2018.

[54] Constantin Udriste. *Convex functions and optimization methods on Riemannian manifolds*, volume 297. Springer Science & Business Media, 2013.

[55] user1952770. Geodesic convexity and the geometric hessian. Available online at https://mathoverflow.net/q/356224.

[56] Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and juntas. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 11–20. IEEE, 2012.

[57] Bart Vandereycken. Low-rank matrix completion by riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013.

[58] Marc Vuffray, Sidhant Misra, Andrey Lokhov, and Michael Chertkov. Interaction screening: Efficient and sample-optimal learning of ising models. *Advances in neural information processing systems*, 29, 2016.

[59] Shayne FD Waldron. Group frames. In *An Introduction to Finite Tight Frames*, pages 209–243. Springer, 2018.

[60] Melanie Weber and Suvrit Sra. Nonconvex stochastic optimization on manifolds via riemannian frank-wolfe methods. *arXiv preprint arXiv:1910.04194*, 2019.

[61] Ami Wiesel and Teng Zhang. *Structured robust covariance estimation.* Now Foundations and Trends, 2015.

[62] Hongyi Zhang, Sashank J Reddi, and Suvrit Sra. Riemannian svrg: Fast stochastic optimization on riemannian manifolds. *Advances in Neural Information Processing Systems*, 29:4592–4600, 2016.

[63] Hongyi Zhang and Suvrit Sra. First-order methods for geodesically convex optimization. In *Conference on Learning Theory*, pages 1617–1638, 2016.

[64] Hongyi Zhang and Suvrit Sra. An estimate sequence for geodesically convex optimization. In *Conference On Learning Theory*, pages 1703–1723, 2018.

[65] Teng Zhang. Robust subspace recovery by geodesically convex optimization. *arXiv preprint arXiv:1206.1386*, 60(11):5597–5625, 2012.