Scalable Inference Algorithms for Determinantal Point Processes

Alireza Rezaei

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2020

Reading Committee:

Shayan Oveis Gharan, Chair

James R. Lee

Kevin Jamieson

Program Authorized to Offer Degree:

Computer Science and Engineering

University of Washington

Abstract

Scalable Inference Algorithms for
Determinantal Point Processes

Alireza Rezaei

Chair of the Supervisory Committee:

Assistant Professor Shayan Oveis Gharan

Computer Science and Engineering

Determinantal Point Processes (DPPs) are probability distributions on subsets of a collection of points that tend to generate diverse configurations of points. This feature makes them suitable as a probabilistic model of diversity. Recently this idea has been exploited extensively in subset selection problems, where given a large set of items such as images, documents, or any other form of collected data, the goal is to select a small, yet diverse and representative subset. However, with the rapid growth of datasets size, in order to utilize DPPs for real-world tasks, we need to design new primitives and inference algorithms that can be run efficiently in these settings.

This thesis focuses on two inference tasks for DPPs: In the first part, we study sampling algorithms for DPPs and offer efficient MCMC based algorithms which can be applied in both discrete and continuous domains. In the second part, we consider the problem of determinant maximization which is equivalent to the Maximum a Posteriori encoding for DPPs, and present scalable algorithms in a distributed setting which assumes the

input data are arbitrarily split among numerous nodes.

TABLE OF CONTENTS

Page

LIST OF FIGURES

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Shayan Oveis Gharan for his endless support and generous mentorship over the years. Throughout my PhD, Shayan patiently paved the way, helped me broaden my knowledge, find my direction, and most importantly enjoy the research. His enthusiasm for research and incredible energy along the way has been the main source of inspiration for me, and without his guidance and constant feedback this PhD would have not been achievable.

This dissertation is based on joint work with great collaborators, a special thanks to Nima Anari, Sepideh Mahabadi, and Piotr Indyk. I want to further express my appreciation to Kevin Jamieson, James R. Lee, and Marina Meila, for devoting their time to serve on my committee and their valuable advice and suggestions to improve my dissertation.

I was privileged to be part of UW theory group: I am deeply grateful to our great faculty, especially Anna Karlin, Paul Beame, Anup Rao, and Thomas Rothvoss for their immense support and valuable courses. Also, a big thank you to my wonderful friends especially Kira, Farzam, Siva, Mert, and Xin for being a willing listener for my ideas and their help and suggestions.

Going back to my time at Sharif university, I want to thank all my undergrad mentors especially Saeed Akbari for teaching me the basics of research and his endless support at the beginning of my academic life.

Life away from home has not been easy without caring friends: a special shout out to Hessam for being a brother to me, and Koosha who made grad life much more enjoyable with all the fun conversations in our tea time.

Finally to my family; to my dear brother Abbas for being my first teacher and forming my interest in math, to Maman and Baba for their unconditional love and support, and keeping me motivated especially over the past years that I could not visit them. I save the final thank you to the love of my life, Reihane; I cannot say how thankful I am for your constant support, love, and encouragement.

Chapter 1

INTRODUCTION

Determinantal Point Processes (DPPs) are probabilistic models of repulsion which were first introduced in quantum physics to model negative interactions among particles [87]; in this context, they represent a probability distribution on the configuration of particles in the space with higher probabilities assigned to states that points are spread out all over the the space. There has been a lot of efforts over last decades to understand the mathematical properties of DPPs, e.g. see [85, 86, 114, 112]. Most importantly to us, Lyons [85] shows that this family of distributions fulfill negative correlation, and its stronger form *negative association* which roughly explains why diverse subsets are more probable under DPPs. Moreover, polynomial time (approximate) algorithms were developed for several basic inference tasks of DPPs in different settings including sampling, learning, marginalization, maximum a posteriori (MAP) inference, etc., see [72] for more details.

Given these algorithmic advances, and the better understanding of DPPs repulsive characteristics, researchers in the ML community initiated studying them as probabilistic models of *diversity* and *fairness*. In particular, DPPs gained a lot of attention for the task of *diverse subset selection*; given a large collection of items, the goal is to choose a small representative subset. In this context, the main quality for a representative subset is its diversity. To get a better sense of this task, consider the following simplified scenario for online shopping: Retailers with a large inventory should pick a small subset of their items which are more likely to engage their customers. To maximize this likelihood,

especially in the lack of adequate knowledge of customers' needs, this selected subset not only should contain highly rated products, but also needs to be diverse to attract a wide range of users. There is long line of work using DPPs to capture diversity in subset selection problems, started by the work of Taskar and Kuelsza [69], who used DPPs for more accurate pose estimation. Since then, DPPs have found many applications in variety of practical applications including video summarization [93, 52], document summarization [72, 71, 27], diverse image annotation [118], tweet timeline generation [119], neural network learning [121, 42], object detection [79], and several others.

In order to utilize DPPs in the aforementioned applications, we need efficient primitives for their basic inference tasks, including learning, sampling, computing marginal probabilities, etc. Although as stated for many of these tasks polynomial time methods are already developed, their computational complexity makes them inefficient in many real world situations that we are dealing with huge inputs. Moreover with persistent growth in data sizes centralized algorithms are no longer suitable choices for many tasks in machine learning. To overcome these challenges, we need to develop techniques and methods which are scalable in distributed settings.

In the first part of this thesis we study the problem of sampling from variations of DPPs. In particular, we focus on k-DPPs, which are DPPs restricted to subsets of a certain size. We present efficient MCMC sampling algorithms that can be applied in discrete and continuous domains.

In the second part we visit the Maximum A Posteriori (MAP) decoding of DPPs. In this problem which is also known as determinant maximization, the goal is to find the most probable (diverse) subset under a DPP distribution. We offer a practically efficient and nearly optimal algorithm in the framework of composable core-sets; In this setting the data is distributed across several units, and the algorithm should first independently summarize each part of the data to shrink the size significantly, and then solve the

problem on the union of these summaries in a centralized fashion.

1.1 Determinantal Point Processes and Diversity

Formally, a discrete point process is a distribution on the subsets of a ground set, known as the domain. There are multiple alternative formulations for determinantal point processes. We mainly refer to the following definition:

Definition 1.1. A point process on domain $[n] = \{1, \ldots, n\}$ is determinantal if there is a positive semi-definite (PSD) matrix $L \in \mathbb{R}^{n \times n}$ such that for each $S \subseteq [n]$,

$$\mathbb{P}(S) \propto \det(L_S),$$

where L_S is the submatrix of L whose rows and columns are indexed by S. Matrix L is known as the ensemble matrix of the DPP.

To see how diverse subsets of items are more probable under DPPs, however, it is more instructive to look at their geometric interpretation which follows from the following elementary fact.

Fact 2. *Let S be the k-dimensional parallelepiped created by rows of matrix $V \in \mathbb{R}^{k \times d}$. Then*

$$\mathrm{VOL}(S)^2 = \det(VV^\intercal).$$

Therefore, let $V = \{v_1, \ldots, v_n\}$ be a set of vectors. A DPP with respect to set V is a probability distribution supported on the subsets of the set $\{1, \ldots, n\}$ where the probability assigned to every subset S is proportional to

$$\mathrm{VOL}(\text{parallelepiped formed by} \{v_i\}_{i \in S})^2.$$

So the probability of choosing a single element $S = \{i\}$ under this distribution is proportional to $\|v_i\|^2$, and for a pair of elements $\{i, j\}$ this probability is proportional to the squared of the area of the parallelogram formed by v_i and v_j. So intuitively, this volume is higher for subsets of vectors that

1. have higher norms.

2. are directionally far from each other. In particular, with fixing the lengths, the highest volume is achieved by a set of orthogonal vectors.

In applications, each $v \in V$ is representing an item in a feature space, e.g. a product in our online shopping example. In this space, the similarity between two items is captured by the dot product between their corresponding vectors. Also, the length of a vector can be thought of the "quality" of the item, i.e. the chance that the product engages a user. In this settings, high quality and diverse subsets of items are more preferred since item 1 ensures high quality items are picked, item 2 gives higher chance to diverse subsets. Therefore, the model balances the diversity of a set versus the quality of its elements, as desired.

1.1.1 Sampling from k-DPPs

To capture real world restrictions, several variation of DPPs with additional constraints are introduced. A cardinality constraint on the returned set is such a required restriction in many settings. For example, for using DPPs for diversifying search results the size of the returned subset is expected to be in a certain range, whereas in standard DPPs, there is no guarantee on the size of the sampled set. A well-known extension of DPPs which allows an explicit control over the size is k-DPP.

Definition 1.3. For an integer k and a DPP μ defined on $2^{[n]}$, the truncation of μ to subsets of size k is called a k-DPP, i.e. denoting this k-DPP by μ_k for any $S \subset [n]$, we have

$$\mu_k(S) \begin{cases} = 0 & \text{if} |S| \neq k \\ \propto \mu(S) & \text{otherwise.} \end{cases}$$

In the first part of the thesis we focus on sampling algorithms from k-DPPs. Most of the previous work on sampling from k-DPPs is focused on spectral methods. However, these

method typically need matrix V in the input, which makes them inefficient when the target DPP is represented by the ensemble matrix L[1]. Given this restriction, [35] suggested using Markov chain techniques, which are very appealing in this context because of their simplicity and efficiency. In chapter 4, we devise an efficient MCMC based algorithm in the discrete case. Next, in chapter 5, we extend our ideas to continuous domain and design MCMC based algorithms for specific families of k-DPPs on continuous domains.

1.1.2 Our Contributions

Sampling from discrete k-DPPs and k-homogeneous strongly Rayleigh measures

A natural Markov Chain Monte Carlo (MCMC) algorithm for the problem is given by the Metropolis-Hasting method. Let π be a k-DPP given by an ensemble matrix $L \in \mathbb{R}^{n\times n}$. The resulting algorithm is a Markov chain which denoted by $\mathcal{M}_\pi$ can be described as follows. The state space of $\mathcal{M}_\pi$ is $\text{supp}\{\pi\}$, i.e. subsets of $[n]$ of size k with non-zero probability under μ. If S is the current state of the chain, the chain moves as below: first choose an element $i \in S$ and $j \notin S$ uniformly and independently at random. Then letting $T = S \setminus \{i\} \cup \{j\}$,

i) If $T \in \text{supp}\{\pi\}$, move to T with probability $\frac{1}{2}\min\{1, \pi(T)/\pi(S)\}$;

ii) Otherwise, stay in S.

From classical results on Markov chain, it turns out that starting from any subset $S(|S| = k$ the chain distribution on states converges to π. To show it is efficient, one needs to upper bound this convergence rate, widely known as the *mixing* time of the chain. There has been several attempts [65, 83, 109] to upper bound the mixing time of $\mathcal{M}_\pi$ for a given k-DPP π and partial results are obtained; but, to the best of our knowledge this question is still open for arbitrary k-DPPs.

---

[1] In this case, first a Cholesky decomposition has to be carried out which runs in time $w(n^2)$.

In the main result of chapter 4 we upper bound the total variation mixing time of π (see definition 2.10 for formal definition).

Theorem 1.4. *For any k-DPP distribution $\mu : 2^{[n]} \to \mathbb{R}_+$, and any starting state $S \in \text{supp}\{\mu\}$, the mixing time of the chain started at S is bounded by $\tilde{O}(kn) \cdot \log \frac{1}{\pi(S)}$.*[2]

To prove the above theorem, we appeal to properties of a broader family of probability distributions known as Strongly Rayleigh (SR) measures. These generalization of DPPs, are introduced and deeply studied in the work of [20]. Most importantly to us it is shown in [20] that

1. unlike DPPs, SR measures are closed under truncation.

2. similar to DPPs, they satisfy the strongest form of negative dependence, a.k.a. negative association.

In fact, using them (crucially item 2), we are able to prove theorem 1.4 for the broader family of k-homogeneous SR measures, i.e. truncation of SR measures to subsets of size k. In order to use $\mathcal{M}_\pi$ to efficiently draw an approximate sample from the k-homogeneous SR measure π, we also need a "proper" starting state and an oracle to compute ratio $\frac{\pi(S)}{\pi(T)}$ for adjacent pair of states to simulate the chain. These oracles can be straight-forwardly obtained for k-DPPs; the time complexity of the resulting method for k-DPPs is then given by the following theorem.

Theorem 1.5. *Given an ensemble matrix L of a k-DPP π, there is an algorithm that generates an approximate sample of π with $\tilde{O}(nk^4)$ arithmetic operations.*

---

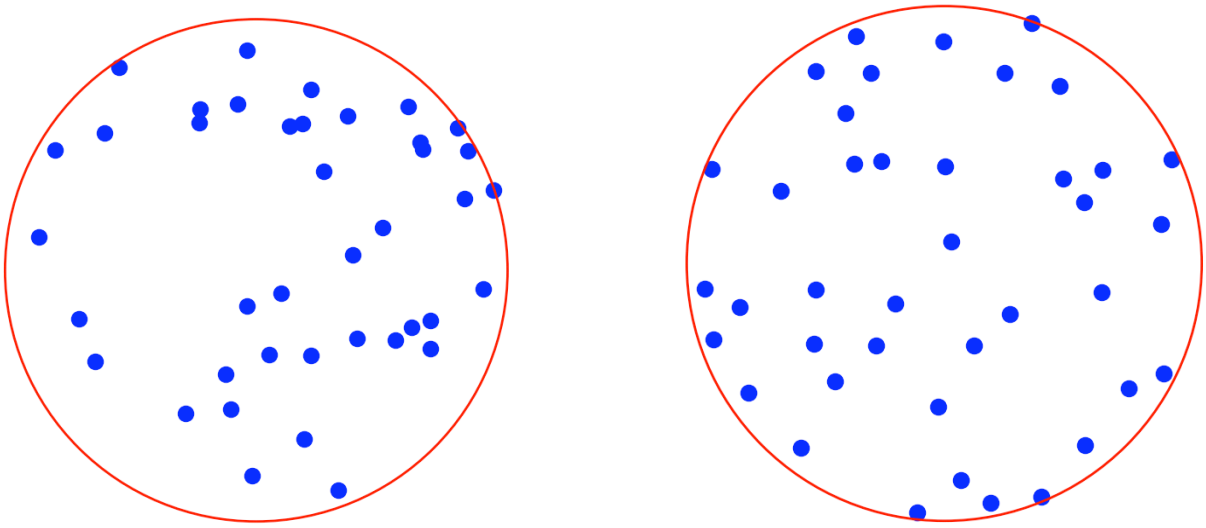
[2]The $\tilde{O}$ notation indicates that some log factors are hidden in the bound.

Sampling from Continuous k-DPPs

So far we only considered DPPs defined on finite sets, but they can also be extended to continuous spaces. In fact, DPPs were originally introduced to model physical particles in continuous spaces. Given $\mathcal{C} \subseteq \mathbb{R}^d$ and positive semi-definite kernel function $L : \mathcal{C} \times \mathcal{C} \to \mathbb{R}^d$ (under some mild conditions), the DPP defined by L is a probability distribution over finite subsets of $\mathcal{C}$ that the probability density function for every such subset $S \subset \mathcal{C}$ is proportional to $\det(L_S)$ where L_S denotes a $|S| \times |S|$ matrix defined by $L(x, y)$ for $x, y \in S$. Similar to the discrete case, a continuous k-DPP is a truncation of a continuous DPP.

An analogy of the geometric insight explained for the discrete case also holds here; the samples generated from a continuous DPPs tends to be more uniformly spread in the space. This is also illustrated in figure section 1.1.2 where 40 points drawn from a uniform distribution are visualized against same number of points generated by a DPP with a Gaussian kernel. This feature makes these distributions an appealing probabilistic model of diversity in continuous domains, for example samples from continuous DPPs can be employed in learning of generative mixture models [54, 107, 74]. In [2] they are used for the initialization step of k-means clustering, and more recently they have found applications in tuning the hyper-parameters of deep networks [42]. Also, see [75] for their applications in statistics and [17] for connections to repulsive systems.

On the algorithmic side, however, there has been less positive results due to the computational challenges arising in continuous domains. In chapter 5, we study the problem of sampling from continuous k-DPPs. The previous efforts has been mostly focused on approximating the continuous kernel by a finite (low) rank, then extending the spectral sampling algorithms for discrete DPPs to continuous domains. Although the idea can yield practical heuristics, as we explain in chapter 5 there are major obstacles to obtain provable guarantees.



(a) Uniform samples

(b) Points Sampled from a DPP

Figure 1.1: Diversity of points sampled from a DPP with a Gaussian kernel versus uniform samples (samples from a Poisson process).

Our result leads to the first class of rigorously analyzed efficient algorithms to generate random samples of continuous k-DPPs. We again follow an MCMC approach, and analyze a Gibbs sampling algorithm for k-DPPs, which was suggested as an efficient heuristic for the problem by [54]. Let π be continuous k-DPP defined by a kernel $L : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$. If the current state of the Gibbs sampler is $\{x_1, \ldots, x_k\}$, the next state will be determined as follows: A point $x_i \in \{x_1, \ldots x_k\}$ is chosen uniformly at random, and is replaced by $y \in \mathcal{C}$ sampled from the conditional distribution whose PDF is defined by

$$f(y) \propto \det_L(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_k),$$

which is the determinant of the $k \times k$ matrix obtained by restricting L to points

$$\{x_1 \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_k\}.$$

Note that the same chain can be defined for discrete settings as well.

Our main contribution is that the chain, for both discrete and continuous k-DPPs, mixes rapidly in time which is only a function of k and independent of the domain.

Theorem 1.6. *If we run the Gibbs sampler for a k-DPP π, starting from an arbitrary distribution μ_0, then the chain mixes after*

$$C_{\mu_0} \cdot \tilde{O}(k^4)$$

steps where the constant C_{μ_0} only depends on μ_0 and π.

In the discrete case, this constant can be easily bounded to get a polynomial time sampling algorithm. Although the final running time turns out suboptimal with respect to the algorithm of chapter 4, the method seems more suitable for distributed models. In this setting, the Gibbs sampler can yield a sub-linear (in terms of domain size) sampling algorithm.

Corollary 1.7. *Given access to n^δ processors for some $\delta > 0$, the Gibbs sampler can be used to generate an approximate sample of a k-DPP defined on domain of size n in time $O(n^{1-\delta}) \cdot \mathsf{poly}(k)$.*

On the other hand, to extend this result onto a polynomial time algorithm for continuous k-DPPs, we need to find a proper starting distribution and also an oracle to run the chain, i.e. a polynomial time method which given any state can take one step of the chain. We are not able to address this step in full generality; instead, we analyze a natural rejection sampler for this task; we analyze the number of generated samples and show that when the spectrum of the eigenvalues of the kernel is not concentrated on the largest k values, the method is efficient. Putting the pieces together, we obtain an efficient algorithm for sampling from k-DPPs defined by spherical Gaussians, a.k.a RBF kernels which are widely popular in practice.

Theorem 1.8. *Let $\mathcal{G}$ be a spherical Gaussian with standard deviation $\sigma = O(1)$ on the unit sphere given by $G_\sigma(x, y) = \exp(\|x - y\|^2/2\sigma^2)$. Also let $k \leq \exp(d/4)$, then an approximate sample of the continuous k-DPP defined by $\mathcal{G}_\sigma$ can be obtained using $\mathsf{poly}(d, k)$ operations.*

1.2 Determinant Maximization

In part II, we consider the MAP encoding problem for DPPs and k-DPPs which is known as determinant maximization; given a DPP defined by ensemble matrix L, the goal is to find the principal submatrix with the highest determinant. Given fact 4, the problem can also be phrased as a volume maximization problem, stated below.

Definition 1.9 (k-volume (k-determinant) maximization)**.** Given a set of vectors $V = \{v_1, \ldots, v_n\} \subset \mathbb{R}^d$ and an integer $k \leq d$, find a subset $S \subset V$ of size k which maximizes the k-dimensional volume of the parallelogram formed by these vectors.

Recall that the parallelogram defined by $v_1, \ldots, v_k$ is the set $\{\sum_{i=1}^{k} \alpha_i v_i \,|\, 0 \leq \alpha_i \leq 1\}$. On the hardness side, it is proved that even approximating the optimum value for determinant maximization up to an exponential factor of 2^{-ck}, for some constant $c > 0$ is NP-hard [32]. On the other hand, this lower bound was matched qualitatively by a recent paper of [99], who gave an algorithm with e^k-approximation guarantee.

This problem is also releted the well-known *submodular maximization* problem as the volume function is a submodular objective in the following sense. Define $f : 2^{[n]} \to \mathbb{R}_+$ by $f(S) = \text{VOL}(\text{Parallelogram}(S))$. Then, one can observe that $\log f$ is a sub-modular function. There is a long line of research on approximation algorithms for submodular maximization, started with the seminal work of [98], who shows a simple greedy procedure achieves a constant factor approximation solution. However, since only the logarithm of the volume is a submodular objective, these results do not directly translate into multiplicative bounds for the determinant maximization. Therefore, to design optimal approximation algorithms with multiplicative gaurantees for volume maximization, relying on the submodularity property is not sufficient.

In part II of the thesis we present our work on determinant maximization in a distributed computing framework known as *composable core-sets*.

1.2.1 Composable Core-sets

In many practical applications of DPPs, the input data is huge and the processing can not be executed on a single machine; this necessities studying algorithms in distributed, streaming, and parallel models of computation One popular such framework in this context is *composable core-sets* introduced in [61], is essentially a variation of the MAP reduced model which allows the processing units to work independently without any need to additional communication during their computations.

Definition 1.10. (composable core-sets) A function $c(V)$ that maps any $V \subseteq \mathbb{R}^d$ into its subset is called a core-set function. This core-set function is an α-composable core-set of size t for the function $f(.)$[3] [61, 4], if for any collection of sets $V_1, \ldots, V_p \subset \mathbb{R}^d$, we have

$$f(c(V_1) \cup \ldots \cup c(V_p)) \geq \frac{1}{\alpha} \cdot f(V_1 \cup \ldots \cup V_p).$$

Suppose that we want to solve an optimization problem on a large data set distributed across several computing units; in the first step of this framework, each machine independently runs the core-set method which produce a small "representative" of its data. Next this procedure then can be repeated for the union of the core-sets, until the data is small enough for a centralized machine to carry out the final algorithm on the entire set and outputs the final value. To formulate determinant maximization in the above language, we assume input vectors are distributed across machines, and for $S \subset \mathbb{R}^d$ the value $f(S)$ indicates the optimum of determinant maximization on S.

1.2.2 Our Contributions

Composable core-sets for determinant maximization has not been studied before, but there has been several efforts for the family of sub-modular functions, which as mentioned earlier also includes the log det function. In particular, [95] showed that a simple

[3] In this setting, function f can indicate the optimum of a maximization problem, e.g. determinant maximization in our problem.

greedy method generates core-sets that guarantee an approximation factor of $\min(k, m)$ for m being the number of chunks of the data, and k being the target cardinality. It is also shown in [61] that under the general assumption of submodularity, this bound can not be improved beyond $\frac{\sqrt{k}}{\log k}$. Chapter 7 examines two well-known heuristics, greedy and local-search methods to construct core-sets, and analyzes their approximation guarantee.

Analyzing Greedy and Local Search Heuristics

Greedy approaches are very appealing in sub-modular maximization problems and lead to constant factor approximation in many settings; so one might expect them to perform well for the determinant maximization, too. In particular, consider the following greedy algorithm: start with an empty set S, and for k iterations, chose an element $i \notin S$ that maximizes $\det(L_{S \cup i})$. This method has been previously analyzed by [31], who shows its approximation factor is $k!$. We analyze this method to generate composable core-sets, i.e. each machine uses this method to pick a subset of size k of its input, and prove the following.

Theorem 1.11. *The greedy method outputs $2^{O(k^2)}$-composable core-sets for k-determinant maximization.*

Next, we analyze a local-search algorithm. Let S be the output of the greedy algorithm. This algorithm start with S and iteratively does the following: if there exists $i \in S$ and $j \notin S$ that swapping them increases the volume with a constant factor, swap them, and terminate if no such a pair can be found. We show this method gives a significant improvement over the greedy method.

Theorem 1.12. *The local-search methods gives $O(2^{2k})$-composable core-sets for k-determinant maximization.*

Optimal Composable Core-sets via Spectral Spanner

Chapter 6 studies the problem from a more theoretical perspective, and aims to find optimal bounds for the composable core-sets for determinant maximization. In fact, To achieve that, we introduce the notion of spectral k-spanners, based on a generalization of the PSD relationship on symmetric matrices; For two $d \times d$ symmetric matrices A, B, we write $A \preceq_k B$ iff the sum of the smallest $d - k + 1$ eigenvalues of $B - A$ is nonnegative. In particular, note that for $k = d$, we recover the well-known notion of $\preceq$, and for $k < d$ we get a weaker relation on A and B, i.e. $A \preceq B$ implies $A \preceq_k B$ but the other direction does not necessarily hold.

Definition 1.13 (spectral k-spanner)**.** For a set of vectors $V \subset \mathbb{R}^d$, we say a set $U \subset V$ is an α-spectral k-spanner, for $k \leq d$ if for all $v \in V$ there is a probability distribution μ_v supported on U such that

$$vv^\intercal \preceq_k \alpha \cdot \mathbb{E}_{u \sim \mu_v} \left[uu^\intercal \right].$$

We show that spectral spanners can be directly used as composable core-sets for the (k)-determinant maximization. Indeed, it turns out they produce almost optimal composable core-sets for a broader family of spectral optimization programs including variations of the experimental design task. For determinant maximization, we achieve the following bound.

Theorem 1.14. *Spectral spanners can produce $O(k)^k$-composable core-sets for k-determinant maximization of size k.*

We also provide an almost matching information theoretic lower-bound, and show that with linear size core-sets, the exponent k in this bound can not be improved asymptotically.

Finally, we present a polynomial time algorithm to find optimal spectral k-spanners, and hence optimal compsoable core-sets for (k)-determinant maximization.

1.3 Organization

In chapter 2 we give a quick overview of main probabilistic and linear algebraic tools that are used throughout the thesis. As alluded to, the primary objects of study in this thesis are determinantal point processes; Chapter 3 formally introduces DPPs, strongly Rayleigh measures and their basic properties. Part I deals with MCMC based sampling algorithms from k-DPPs; in chapter 4, we present our sampling method for discrete k-DPPs and homogeneous strongly Rayleigh measures. Next, in chapter 5 we study the problem for continuous k-DPPs. The subject of the second part of the thesis is the problem of determinant maximization in the framework of composable core-sets. Although, we defined the problem in the context of DPPs, this part is self-contained and no prior knowledge of DPPs is needed to read that; in chapter 6, we offer an almost theoretically optimal construction of composable core-sets that can also be applied to a wider range of optimization problems. Next, in chapter 7, we visit two popular efficient heuristics for the problem and rigorously analyze them. In particular, we present a local-search method that can be executed very efficiently in practice and its theoretical bound is very close the optimal bound obtained in chapter 6.

Chapter 2

PRELIMINARIES

2.1 Linear Algebra

Let $\mathbb{R}^d$ denote the d-dimensional Euclidian space. Throughout this manuscript, all vectors that we consider are column based and sitting in $\mathbb{R}^d$, unless otherwise specified. For a vector v, we use notation $v(i)$ to denote its i_{th} coordinate and use $\|v\|$ to denote its ℓ_2 norm, i.e. $\|v\| = \sum_{i=1}^{d} v(i)^2$. Vector v is called a unit vector if $\|v\| = 1$. We use $e_1, \ldots, e_d$ to denote standard unit vectors, that is for any i, e_i is a vector whose i_{th} coordinate is 1 and its other coordinates are zero. For two vectors u, v, we use $\langle u, v \rangle$ to denote their inner-product which is given by $\sum_{i=1}^{d} u(i)v(i)$. u, v are orthogonal if $\langle u, v \rangle = 0$. Vectors $v_1, \ldots, v_k$ are called orthonormal if for any i, $\|v_i\| = 1$, and for any $i \neq j$, $\langle v_i, v_j \rangle = 0$.

For a set of vectors V, we let $\langle V \rangle$ denote the linear subspace spanned by vectors of V. We also use $S^{\perp}$ to denote the linear subspace orthogonal to S, for a linear subspace S, i.e. $S^{\perp} = \{a | \, \forall v \in S, \, \langle v, a \rangle = 0\}$.

Notation $\langle , \rangle$ is used to denote Frobenius inner product of matrices, for matrices $A, B \in \mathbb{R}^{d \times d}$

$$\langle A, B \rangle = \sum_{i=1}^{d} \sum_{j=1}^{d} A_{i,j} B_{i,j} = \text{tr}(AB^{\intercal})$$

where $A_{i,j}$ denotes the entry of matrix A in row i and column j and tr denotes the trace operator which for a matrix A is defined by $\text{tr}(A) \sum_{i=1}^{d} A_{i,i}$.

A matrix $A \in \mathbb{R}^{d \times d}$ is symmetric if for any $1 \leq i, j \leq d$, we have $A_{ij} = A_{ji}$. The set

of all symmetric $d \times d$ matrices is denoted by $\mathbb{S}_d$. Matrix A is a Positive Semi-definite (PSD) matrix denoted by $A \succeq 0$ if it is symmetric and for any vector v, we have $v^\intercal Av = \langle A, vv^\intercal \rangle \geq 0$. For PSD matrices A, B we write $A \preceq B$ if $B - A \succeq 0$. We also denote the set of $d \times d$ PSD matrices by $\mathbb{S}_d^+$.

2.1.1 Eigenvalues

Let $A \in \mathbb{R}^{d \times d}$ and let $\lambda_1 \ldots, \lambda_d$ be its eigenvalues with corresponding eigen-vectors $v_1, \ldots, v_d$, i.e. for any i, $Av_i = \lambda_i v_i$. If A is a symmetric matrix, its eigenvalues are real values, and can be characterized by the following theorem, knows as min-max characterization of eigenvalues.

Theorem 2.1 (Min-max Characterization of Eigenvalues)**.** *Let $A \in \mathbb{S}_d$ with eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$. Then*

$$\lambda_k = \max\{\min_{x \in U} \frac{x^\intercal Ax}{\|x\|^2} \mid U \text{ is a k-dimensional linear subspace}\},$$

or

$$\lambda_k = \min\{\max_{x \in U} \frac{x^\intercal Ax}{\|x\|^2} \mid U \text{ is a } (d-k+1)\text{-dimensional linear subspace}\},$$

The following theorem known as Cauchy interlacing theorem shows the relation between eigenvalues of a symmetric matrix and eigenvalues of its submatrices.

Theorem 2.2 (Cauchy Interlacing Theorem)**.** *Let $A \in \mathbb{S}_d$ be a symmetric matrix, and B be an $m \times m$ principal submatrix of A, i.e. B is obtained by picking out the elements of A in rows and columns both indexed by a subset S of size m. Then for any $1 \leq i \leq m$, we have*

$$\lambda_{d-m+i}(A) \leq \lambda_i(B) \leq \lambda_i(A)$$

We also use the following lemma which is an easy implication of min-max characterization to bound eigenvalues of summation of two matrices in terms of the summation their eigenvalues.

Lemma 2.3. *Let $A, B \in \mathbb{S}_d$ be two symmetric matrices. Then $\lambda_{i+j-d}(A+B) \geq \lambda_i(A) + \lambda_j(B)$ for any i, j with $i + j - d > 0$.*

Proof. Following the min-max characterization of eigenvalues, let S_A and S_B be two i-dimensional and j-dimensional linear subspaces for which we have

$$\lambda_i(A) = \min_{x \in S_A} \frac{x^\intercal A x}{\|x\|^2} \quad \text{and} \quad \lambda_j(B) = \min_{x \in S_B} \frac{x^\intercal B x}{\|x\|^2}$$

Then let $S = S_A \cap S_B$. The dimension of S is at least $i + j - d$, and by min-max characterization of eigenvalues we have

$$\lambda_{i+j-d}(A+B) \geq \min_{x \in S} \frac{x^\intercal (A+B) x}{\|x\|^2} \geq \min_{x \in S_A} \frac{x^\intercal A x}{\|x\|^2} + \min_{x \in S_B} \frac{x^\intercal B x}{\|x\|^2} = \lambda_i(A) + \lambda_j(B),$$

hence the proof is complete. □

2.1.2 Determinant

There are several ways to define the notion of *determinant*. Let $A \in \mathbb{R}^{d \times d}$ be a square matrix with rows $a_1, \ldots, a_d$. A textbook definition of determinant is as follows: Determinant of $d \times d$ matrices is a function $\det : \mathbb{R}^d \times \mathbb{R}^d \ldots \mathbb{R}^d (d \text{ A tuple of } d \text{ vectors of } \mathbb{R}^d) \to \mathbb{R}$ with the following three properties:

1. det is multilinear, meaning that fixing all rows except one , the det function is linear with respect to the changing row, i.e. for any i and for any vector $r \in \mathbb{R}^d$,

$$\det(a_1, \ldots, a_i + r, \ldots, a_d) = \det(a_1, \ldots, a_i, \ldots, a_d) + \det(a_1, \ldots, r, \ldots, a_d).$$

2. Swapping any pair of rows only negates the value of the determinant, i.e. for any $1 \leq i, j \leq d$,

$$\det(\ldots, a_i, \ldots a_j, \ldots) = -\det(\ldots, a_j, \ldots, a_i, \ldots).$$

3. The value of the determinant for the identity matrix is $+1$, i.e. letting $e_1, \ldots, e_d$ be the standard basis vectors we have $\det(e_1, \ldots, e_d) = 1$.

We use the notion of *determinant* of a subset of vectors as a measure of their diversity. This is more clear from a geometric point of view; in particular, we use the following fact which relates determinant to the more geometric concept of volume.

Fact 4. *Let $V \in \mathbb{R}^{k\times d}$ and let $v_1, \ldots, v_k$ denote the rows. Then* $\det(VV^\intercal)$ *is equal to the square of the k-dimensional volume of the parallelepiped spanned by vectors $v_1, \ldots, v_k$ which is the set $\{\sum_{i=1}^k \alpha_i v_i \,|\, \forall i, 0 \le \alpha_i \le 1\}$.*

So for example, for two vectors u and v, the area of the parallelepiped formed by u and v is equal to the root squared of

$$\left|\det\begin{pmatrix} \|u\|^2 & \langle u, v\rangle \\ \langle v, u\rangle & \|v\|^2 \end{pmatrix}\right|$$

Note that when $k = d$, we have that $\det(VV^\intercal) = \det(V^\intercal V)$ where the second term can also be written as $\det(\sum_{i=1}^d v_i v_i^\intercal)$. For $k < d$, this clearly does not hold as the second determinant is zero; the Cauchy-Binet identity can be used to generalize the above in this regime. For $S, T \subseteq [d]$, Let $A_{S,T}$ denote the $|S| \times |T|$ submatrix formed by intersecting the rows and columns corresponding to S, T respectively.

Fact 5 (Cauchy-Binet identity)**.** *For any integer $k \le d$, $B \in \mathbb{R}^{k\times d}$, and $C \in \mathbb{R}^{d\times k}$,*

$$\det(BC) = \sum_{S \in \binom{[d]}{k}} \det(B_{[k],S} C_{S,[k]}), \tag{2.1}$$

When setting $B = C = V$, each term in the RHS of (2.1) corresponds to a $k \times k$ submatrix of $V^\intercal V$. For a $d \times d$ matrix A, define $\det_k(A)$ as the summation of the determinant of $k \times k$ principal submatrices of A, i.e.

$$\det_k(A) = \sum_{S \in \binom{[d]}{k}} \det A_{S,S}.$$

Then, the Cauchy-Binet identity indicates that for vectors $v_1, \ldots, v_k \in \mathbb{R}^d$, $\det_k(\sum_{i=1}^k vv_i^\intercal)$ is equal to the square of the k-dimensional volume of the parallelepiped spanned by $v_1, \ldots, v_k$.

Determinant can also be expressed in terms of eigenvalues for symmetric matrices.

Fact 6. *Let A be a $d \times d$ symmetric matrix with eigenvalues $\lambda_1, \ldots, \lambda_d$, then*

$$\det(A) = \prod_{i=1}^{d} \lambda_i.$$

For specific family of matrices, there are easier formulas to compute determinant. In particular, we use the following formula for the determinant of lower-triangular matrices.

Fact 7. *Let $A \in \mathbb{R}^d \times d$ be a lower-triangular matrix, i.e. for any $j > i$, $A_{ij} = 0$. Then*

$$\det(A) = \prod_{i=1}^{d} A_{ii}.$$

2.1.3 Matrix Norms

Throughout the thesis, we work with different norms for matrices. The ℓ_2-norm of matrix A, denoted by $\|A\|_2$ or just $\|A\|$ denotes $\max_{\|x\|_2=1} \|Ax\|_2$. For symmetric matrices, it is straight-forward that the ℓ_2-norm is equal to the largest eigenvalue.

The Frobenius norm A is denoted by $\|A\|_F$ and is defined by

$$\|A\|_F = \sqrt{\sum_i \sum_j A_{ij}^2}.$$

For two matrices A, B, if we define the inner-product $\langle A, B \rangle$ as $\sum_{i=1}^d \sum_{j=1}^d A_{ij} B_{ij}$, then $\|A\|_F = \sqrt{\langle A, A \rangle}$. We also use the following identity which relates Frobenius norm to singular values.

Fact 8. *For any matrix* $A \in \mathbb{R}^{d\times d}$,

$$\|A\|_F^2 = \sum_{i=1}^{d} \sigma_i(A)^2.$$

Finally, $\|A\|_\infty = \max_{i,j} |A_{i,j}|$ denotes the ℓ_∞ norm of matrix A.

2.2 Markov Chain on Finite Spaces

In this section we give a high level overview of Markov chains defined on finite spaces and their mixing times. We refer readers to [82, 97] for more details. A Markov chain $\mathcal{M}$ can be specified by a triplet (Ω, P, π) where Ω denotes the state space of the chain, and $P : \Omega \times \Omega \to \mathbb{R}_+$ is called its transition kernel, i.e. $P(x, y)$ is the probability that chain moves to state y if its current state is x. So by definition for any $x \in \Omega$, we have $\sum_{y\in\Omega} P(x, y) = 1$. It is also referred to as the transition matrix of the chain, i.e. P can be identified by an $\Omega \times \Omega$ matrix where P_{xy} denotes $P(x, y)$. Finally $\pi : \Omega \to \mathbb{R}_+$ is called the *stationary measure* of the chain.

Let μ_0 be a probability distribution on Ω. If we start $\mathcal{M}$ from a state sampled from μ_0 and take one step of the chain, the next state is a random variable with distribution $\mu_0 P$. Similarly for any integer m, the resulting state after taking m steps of the chain has distribution $\mu_m = \mu_0 P^m$. It is guaranteed that under some mild conditions starting the chain from any step and taking sufficiently large number of steps, the resulting distribution on the states converges to the unique stationary measure of the chain. Due to this property, Markov chains are widely used to generate samples from probability distributions.

The chain $\mathcal{M}$ is said to be an irreducible chain if any pair of states $x, y \in \Omega$ there exists an integer t so that $P^t_{xy} > 0$. Moreover, an irreducible chain is an aperiodic chain if for any pair of states x, y there exists integer t so that for any integer $m \geq t$, we have $P^m_{xy} > 0$. The following theorem formalizes the previous paragraph.

Theorem 2.9 (Convergence Theorem [82]). *If $\mathcal{M}$ is an irreducible and aperiodic Markov chain with stationary measure π, then there exists constants $0 < \alpha < 1$ and $C > 0$ such that for any starting distribution μ_0,*

$$\|\mu_0 P^t - \pi\|_{(\mathrm{TV})} \leq C\alpha^t.$$

In the above for two probability distributions $\mu, \nu : \Omega \to \mathbb{R}_+$, the total variation distance between μ and ν is defined as

$$\|\mu - \nu\|_{(\mathrm{TV})} = \frac{1}{2} \cdot \|\mu - \nu\|_{(1)} = \frac{1}{2} \cdot \sum_{x \in \Omega} |\mu(x) - \nu(x)|.$$

2.2.1 Mixing Time

In order to use Markov chains as efficient algorithms for sampling, one needs to bound the number of steps before convergence of the chain, a.k.a. *mixing time*.

Definition 2.10 (Mixing Time). For a state $x \in \Omega$ and $\epsilon > 0$, the total variation mixing time of a chain started at x with transition probability matrix P and stationary distribution π is defined as follows:

$$\tau_x(\epsilon) := \min\{t : \|\mathbb{1}_x P^t - \pi\|_{(\mathrm{TV})} \leq \epsilon\}$$

where $\mathbb{1}_x P^t$ shows the distribution of the chain started at state x at time t.

Note that in general the mixing can be defined for a starting distribution, rather than a single starting state.

A Markov chain $\mathcal{M} = (\Omega, P, \pi)$ is reversible if for any pair of states $x, y \in \Omega$, $\pi(x)P(x,y) = \pi(y)P(y,x)$ which is also known as the *detailed balanced* condition. The chain $\mathcal{M}$ is said to be a *lazy* chain if for any state $x \in \Omega$, $P(x,x) \geq \frac{1}{2}$. We equip the space of all functions $f : \Omega \to \mathbb{R}$ with the standard inner product for $L^2(\pi)$,

$$\langle f, g \rangle_\pi := \mathbb{E}_\pi [f \cdot g] = \sum_{x \in \Omega} \pi(x) f(x) g(x).$$

In particular, $\|f\|_\pi = \sqrt{\langle f, f\rangle_\pi}$. For a function $f \in L^2(\pi)$, the *Dirichlet form* $\mathcal{E}_\pi(f, f)$ is defined as follows

$$\mathcal{E}_\pi(f, f) := \frac{1}{2} \sum_{x,y\in\Omega} (f(x) - f(y))^2 P(x, y)\pi(x),$$

and the *Variance* of f is

$$\mathrm{var}_\pi(f) := \|f - \mathbb{E}_\pi[f]\|_\pi^2 = \sum_{x\in\Omega} (f(x) - \mathbb{E}_\pi[f])^2 \pi(x).$$

Next, we overview classical spectral techniques to upper bound the mixing time of Markov chains.

Definition 2.11 (Poincaré Constant)**.** The *Poincaré constant* of the chain is defined as the largest constant λ which satisfies the following,

$$\lambda \cdot \mathrm{var}_\pi(f) \leq \mathcal{E}_\pi(f, f),$$

for all functions $f : \Omega \to \mathbb{R}$.

We will use the following result by Diaconis and Stroock [41] to bound the mixing time of $\mathcal{M}$. The following classical result shows that if $\pi(.)$ is a uniform distribution, the chain mixes in time $\log(\Omega)/\lambda$.

Theorem 2.12 ([40])**.** *For any reversible irreducible lazy Markov chain (Ω, P, π) with Poincarë constant λ and $\epsilon > 0$,*

$$\tau(\epsilon) \leq \frac{1}{\lambda} \cdot \log\left(\frac{1}{\epsilon \cdot \pi_{\min}}\right),$$

where $\pi_{\min} := \min_{x\in\Omega} \pi(x)$.

The above bound is not strong enough for our particular application in sampling from a k-DPP or a strongly Rayleigh distribution in chapter 4. This is because $\pi_{\min}$ can be arbitrarily smaller than the probability of the starting state. Instead, we use the following slight generalization of the above theorem.

Theorem 2.13 ([41, Prop 3]). *For any reversible irreducible lazy Markov chain (Ω, P, π) with Poincaré constant λ, for any $\epsilon > 0$, and any state $x \in \Omega$,*

$$\tau_x(\epsilon) \leq \frac{1}{\lambda} \cdot \log\left(\frac{1}{\epsilon \cdot \pi(x)}\right)$$

It is easy to see that for any transition probability matrix P, the second largest eigenvalue of P is $1 - \lambda$. If P is a lazy chain, then $1 - \lambda$ is also the second largest eigenvalue of P in absolute value.

2.3 Markov Chain on General State Spaces

Markov chains can also be defined on continuous domains, rather than a finite state space. Here, we give a short overview, but for a complete account we refer to [84]. Let $(\Omega, \mathcal{B})$ be a measurable space; in the most general setting, a Markov chain is defined by the triple $(\Omega, \mathcal{B}, \{P_x\}_{x\in\Omega})$, where for every $x \in \Omega$, $P_x : \mathcal{B} \to \mathbb{R}_+$ is a probability measure on $(\Omega, \mathcal{B})$. Also, for every fixed $B \in \mathcal{B}$, $P_x(B)$ is a measurable function in terms of x. In this setting starting from a distribution μ_0, after one step the distribution μ_1 would be given by

$$\mu_1(B) = \int_\Omega P_x(B) d\mu_0(x), \ \forall B \in \mathcal{B}.$$

From now on, assume $\Omega \subset \mathbb{R}^k$ and $\mathcal{B}$ is the standard Borel σ-algebra. In our setting, we can assume the transition probabilities are given by a kernel *transition kernel* $P : \Omega \times \Omega \to \mathbb{R}_+$ where for any measurable $A \subset \Omega$, we can write

$$P_x(A) = \int_A P(x, y) dy.$$

In this notation, we use $P(x, B)$ and $P_x(B)$ interchangeably. $P^n(x, .)$ would also denote the probability distribution of the states after n steps of the chain started at x. Similar to the discrete setting, we can define the *stationary measure* for the chain. A probability distribution π on Ω is stationary if and only if for every measurable set B, we have

$$\pi(B) = \int_\Omega \int_B P(x, y) dy d\pi(x).$$

We call $\mathcal{M}$ *ϕ-irreducible* for a probability measure ϕ if for any set $B \in \mathcal{B}$ with $\phi(B) > 0$, and any state x, there is $t \in \mathbb{N}$ such that $P^t(x, B) > 0$. It is called *strongly ϕ-irreducible* if for any $B \subseteq \Omega$ with non-zero measure and $x \in \Omega$, there exists $t \in \mathbb{N}$ such that for any $m \geq t$, $P^m(x, B) > 0$. We say $\mathcal{M}$ is *reversible* with respect to a measure π if for any two sets A and B we have

$$\int_B \int_A P(y, x) dx d\pi(y) = \int_A \int_B P(x, y) dy d\pi(x).$$

In particular, reversibility with respect to a measure, implies it is a stationary measure. The following lemma also shows π is the unique stationary measure, and as the number of steps increases, the chain approaches to the unique stationary measure.

Lemma 2.14 ([39]). *If π is a stationary measure of $\mathcal{M}$, and $\mathcal{M}$ is strongly π-irreducible. Then for any other distribution μ which is absolutely continuous with respect to π, $\lim_{n\to\infty} |P^n(\mu, .) - \pi|_{TV} = 0$.*

2.3.1 Mixing Time

From now on, assume $\mathcal{M} = (\Omega, P, \pi)$ is a chain with state space Ω, probability transition function P, and a unique stationary measure π. Let us describe some results about mixing time in the Markov chains defined on continuous spaces. But before that we need to setup some notation. Let $L^2(\Omega, \pi)$ be the space of functions on Ω with finite ℓ_2-norm with respect to π, i.e. $\int_\Omega |f(x)| \, d\pi(x) < \infty$. The inner product in this space is defined as

$$\langle f, g \rangle_\pi = \int_\Omega f(x) g(x) d\pi(x).$$

Then P is an operator that for any function $f \in L^2(\Omega, \pi)$ and $x \in \Omega$,

$$(Pf)(x) = \int_\Omega P(x, y) f(y) dy.$$

In particular, $\mathcal{M}$ being reversible is equivalent to P being self-adjoint, i.e. for any pair of functions f, g, $\langle Pg, f \rangle_\pi = \langle Pf, g \rangle_\pi$. For a reversible chain $\mathcal{M}$ and a function $f \in L^2(\Omega, \pi)$,

the Dirichlet form $\mathcal{E}_P(f,f)$ is defined as

$$\mathcal{E}_P(f,f) = \frac{1}{2}\int_\Omega \int_\Omega (f(x)-f(y))^2 P(x,y) d\pi(x) dy.$$

We also define the *Variance* of f with respect to π as

$$\mathrm{var}_\pi(f) := \int_\Omega (f(x) - \mathbb{E}_\pi(f))^2 d\pi(x).$$

We may drop the subscript if the underlying stationary distribution is clear in the context. Similar to the discrete case, one way for upperbounding the mixing time of a chain on a continuous state space is to use is to its spectral gap which is also known as *Poincaré Constant*.

Definition 2.15 (Poincaré Constant)**.** . The Poincaré constant of the chain the largest value of $\lambda > 0$ for which the following holds

$$\lambda \cdot \mathrm{var}(f) \leq \mathcal{E}_P(f,f)$$

for any function $f \in L^2(\Omega,\pi)$.

In chapter 5 , we use the following theorem to upperbound the mixing time of the chain relevant to us.

Theorem 2.16 ([67])**.** *For any lazy, reversible, strongly π-irreducible Markov chain $M = (\Omega, P, \pi)$, if $\lambda > 0$, then the distribution of the chain started from μ (which is absolute continuous with respect to π) is*

$$\|P^t(\mu,.) - \pi\|_{TV} \leq \frac{1}{2}(1-\lambda)^t \sqrt{\mathrm{var}\left(\frac{f_\mu}{f_\pi}\right)}.$$

For the sake of completeness, we include a proof of the above theorem which is an extension of the proof of the analogous discrete result in [48]. We need the following simple lemma known as Mihail's identity.

Lemma 2.17 (Mihail's identity, [48]). *For any reversible irreducible Markov chain* $\mathcal{M} = (\Omega, P, \pi)$*, and any function* f *in* $L^2(\pi)$*,*

$$\mathrm{var}(f) = \mathrm{var}(Pf) + \mathcal{E}_{P^2}(f, f).$$

Proof of Theorem 2.16. First of all, one can easily verify that if a chain is lazy and irreducible, then it is strongly-irreducible. Combining it with Lemma 2.14 would guarantee the uniqueness of the stationary measure. Let $\mu_0 = \mu$ be the starting distribution and define $\mu_t = P^t(\mu, .)$ be the distribution at time t. For distributions μ_t and π, let f_{μ_t} and f_π denote the density functions of the distribution, and set $f_t := \frac{f_{\mu_t}}{f_\pi}$, we have

$$(Pf_t)(x) = \int_\Omega P(x,y)\frac{f_{\mu_t}(y)}{f_\pi(y)}dy = \int_\Omega \frac{P(y,x)f_{\mu_t}(y)}{f_\pi(x)}dy = \frac{f_{\mu_{t+1}}}{f_\pi}(x) = f_{t+1}(x)$$

which implies

$$\mathrm{var}(Pf_t) = \mathrm{var}(f_{t+1}) \tag{2.2}$$

So applying Mihail's identity on $\frac{f_{\mu n}}{f_\pi}$ and using (2.2) , we conclude

$$\mathrm{var}(f_t) = \mathrm{var}(f_{t+1}) + \mathcal{E}_{P^2}(f_t, f_t). \tag{2.3}$$

Now, note that P^2 has the same stationary distribution π, so its Poincaré constant is at most

$$\lambda(P^2) \leq \frac{\mathcal{E}_{P^2}(f_t, f_t)}{\mathrm{var}(f_t)}.$$

Combining this with (2.3), and using induction we can deduce

$$\mathrm{var}(f_t) \leq (1 - \lambda(P^2))^t \,\mathrm{var}(f_0).$$

Note that, since P is the kernel for a lazy chain, it has no negative values in its spectrum, implying $1 - \lambda(P^2) = (1 - \lambda(P))^2$. So in order to complete the proof it is enough show

$$4\|\mu_t - \pi\|_{TV}^2 \leq \mathrm{var}(f_t).$$

This can be seen using an application of Cauchy-Schwarz's inequality. We have

$$\begin{aligned} 4\|\mu_t - \pi\|_{TV}^2 &= \left(\int_\Omega |f_{\mu_t}(x) - f_\pi(x)| dx\right)^2 \\ &= \left(\int_\Omega f_\pi(x) \left|\frac{f_{\mu_t}(x)}{f_\pi(x)} - 1\right| dx\right)^2 \\ &\leq \int_\Omega f_\pi(x) \left|\frac{f_{\mu_t}(x)}{f_\pi(x)} - 1\right|^2 dx = \text{var}(\frac{f_{\mu_t}}{f_\pi}) \end{aligned}$$

The last identity uses that $\mathsf{EEE}_\pi \frac{f_{\mu_t}}{f_\pi} = 1$. This completes the proof. □

In order to take advantage of Theorem 2.16, we need to lowerbound the Poicaré constant of our chain. This can be done by lowerbounding the *Ergodic Flow* of the chain.

Definition 2.18 (Ergodic Flow)**.** For a chain $\mathcal{M} = (\Omega, P, \pi)$, the ergodic flow $Q : \mathcal{B} \to [0,1]$ is defined by

$$Q(B) = \int_B \int_{\Omega \setminus B} P(u,v) dv f_\pi(u) du.$$

The *conductance* of a set B is defined by, $\phi(B) := \frac{Q(B)}{\pi(B)}$, and the conductance of the chain is

$$\phi(\mathcal{M}) = \min_{0 < \pi(B) \leq \frac{1}{2}} \phi(B).$$

The following theorem which is an extension of the Cheeger's inequality for the Markov chains on a continuous space, relates the spectral gap to conductance.

Theorem 2.19 ([77])**.** *For a chain $\mathcal{M}$ defined on a general state space with spectral gap λ we have*

$$\frac{\phi(\mathcal{M})^2}{8} \leq \lambda \leq 2\phi(\mathcal{M}).$$

Chapter 3

DETERMINANAL POINT PROCESSES: DEFINITION AND BASIC PROPERTIES

Determinantal Point Processes (DPPs) are central objects of study in this thesis. In this chapter, we formally define these probability distributions and their variants, and describe their basic properties.

Let $\mathcal{Y}$ be a mathematical space such as the real line, a ball or a discrete set. A point process on *domain* $\mathcal{Y}$ defines a probability distribution on collection of points located on $\mathcal{Y}$. For simplicity, we first focus on the discrete case where $\mathcal{Y}$ is a discrete and finite set of elements. A point process on such $\mathcal{Y}$ represents a probability distribution over $2^{\mathcal{Y}}$, which denotes the set of subsets of $\mathcal{Y}$. In this case, we sometimes refer to elements of $\mathcal{Y}$ as *items*.

3.1 Discrete Determinantal Point Processes

Without loss of generality assume $\mathcal{Y} = [N] = \{1, 2, \ldots, N\}$. A discrete DPP can be defined as follows.

Definition 3.1 (Discrete DPP)**.** A discrete DPP on the set of elements $[N]$ is a point process μ that can be identified by a PSD matrix $L \in \mathbb{R}^{N \times N}$ such that for every $S \subseteq [N]$,

$$\mu(S) \propto \det(L_S) \tag{3.1}$$

where $L_S = [L_{ij}]_{i,j \in S}$ is the principal submatrix of L indexed by S. The set $[N]$ is referred to as the domain of μ, and the matrix L is also called the *ensemble* matrix for μ. Sometimes it is also referred to as its *kernel*.

Note that, we need L to be PSD to ensure all probabilities, which are proportional to principal minors, are non-negative and the distribution is well-defined. To obtain the probability of each subset with the above characterization, one also needs to obtain the normalization constant of eq. (3.1), i.e. $\sum_{S\subseteq[N]} \det(L_S)$. This constant can be easily computed as follows

$$\sum_{S\subseteq[N]} \det(L_S) = \det(L+I).$$

In the above I denotes the $N \times N$ identity matrix. So if μ is a DPP defined by kernel L, then for any subset S of elements, the probability of choosing S is equal to $\frac{\det(L_S)}{\det(L+I)}$.

In this manuscript, we stick to definition 3.1 for DPPs. However, there is an alternative formulation of DPPs, sometimes used in the literature that we include as well to give more context to readers.

3.1.1 An Alternative Formulation by Marginal Kernels

DPPs can be described in terms of marginal probabilities of items. Let $\boldsymbol{S} \subseteq [N]$ denote a random subset drawn according to a point process μ. One can use eq. (3.1) to show that μ is a DPP if and only if there is a PSD matrix $K \in \mathbb{R}^{N\times N} \preceq I$ such that for every subset A of elements, we have

$$\mathbb{P}_\mu [A \subseteq \boldsymbol{S}] = \det(K_A). \tag{3.2}$$

This matrix K is called the marginalization kernel for μ. Note that, The condition $K \preceq I$ is essential to guarantee that marginal probabilities are at most 1. So in this formulation, the probability that an element $i \in [N]$ is selected in the DPP sample is equal to the corresponding diagonal element of K_{ii}. The kernel of μ, L, and K are related by the following formula: $K = L(L+I)^{-1}$. For a complete account of this formulation, see [72].

3.1.2 DPPs and Diversity

As alluded to in the introduction, determinants, and in particular DPPs in machine learning are mostly used to model diversity. As we saw, the relationship between the volume and determinant can illustrate this tendency of DPPs toward diverse subsets of items; in summary, let μ be a DPP defined over set $[N]$ via kernel L. Since L is a PSD matrix, it follows from elementary linear algebra that it can be written as $L = VV^\intercal$ for some matrix V. In this setting, each row of V can be viewed as a feature vector, and the probability assigned by μ to each subset of items is proportional to the squared of the volume of the corresponding feature vectors. So samples generated from μ tend to span larger volumes, thus achieve higher probabilities.

A more systematic way to formalize diversity is through the notion of *negative correlation*.

Negative Correlation

Let $i, j \in [N]$ be two elements of the domain. We say they are negatively correlated with respect to a point process μ, if

$$\mathbb{P}_{S \sim \mu}[i, j \in S] \leq \mathbb{P}_{S \sim \mu}[i \in S] \cdot \mathbb{P}_{S \sim \mu}[j \in S] \tag{3.3}$$

Or equivalently for a set S sampled from distribution μ, the probability of having j in S is smaller that its conditional probability when i is present in S. The above formalizes a repulsive relation between these two elements. As it turns out, when μ is determinantal point processes, this relation holds for any pair of elements.

Fact 2 (Pairwise Negative Correlation in DPPs)**.** *Let μ be a DPP defined on domain $[N]$. Then, any pair of elements are negatively correlated, i.e. eq. (3.3) holds for any pair $i, j \in [N]$.*

Proof. To verify that eq. (3.3) holds for all pairs, we use the representation of μ by its marginal kernel. Let K denote the marginal kernel. The followings for an arbitrary pair

$i, j \in [N]$ are immediate from eq. (3.2).

$$\begin{aligned} \mathbb{P}_S \sim \mu\,[i \in S] &= K_{i,i} \\ \mathbb{P}_S \sim \mu\,[j \in S] &= K_{j,j} \end{aligned}$$

and

$$\mathbb{P}_S \sim \mu\,[i, j \in S] = \det \begin{pmatrix} K_{i,i} & K_{i,j} \\ K_{i,j} & K_{j,j} \end{pmatrix} = K_{i,i}K_{j,j} - K_{i,j}^2$$

Now, it is easy to verify eq. (3.3) for i and j as replacing the RHS of above equations in eq. (3.3), it reduces to

$$\mathbb{P}_S \sim \mu\,[i \in S] \cdot \mathbb{P}_S \sim \mu\,[j \in S] - \mathbb{P}_S \sim \mu\,[i, j \in S] = K_{i,j}^2 \geq 0,$$

which completes the proof. □

Therefore, in a DPP, all pairs of elements prefer not to be selected together, which resembles a notion of diversity among items.

3.1.3 Basic Primitives for DPPs

DPPs are not the only family of probabilistic models which can theoretically characterize the notion of diversity, but they are one of the most popular ones. A key factor which makes them widely used is practice is that several basic operations and inference tasks for these family of probability distributions are tractable, and admit efficient polynomial time algorithm. We give a short introduction of a few of basic tasks for DPPs.

Marginalization. Given a probability distribution μ, and a subset A of the sample space, the marginal property of A, is the probability that it is contained in a sample generated from μ, i.e. $\mathbb{P}_S \sim \mu\,[A \subset S]$. As explained in the previous section, marginal probabilities of a DPP defined by kernel L can be expressed as principal minors of another PSD matrix defined by $K = L(L + I)^{-1}$. In other words, for any set A, the marginal probability of A is proportional to $\det(K_A)$ which suggests that given the DPP kernel, marginals can be computed in time required to perform a matrix inversion.

Conditioning. For a point process μ on domain $[N]$ and a subset A of the domain, conditioning of μ on subset A, defines a new point process on domain $[N] \setminus A$ that for any subset $B \subset [N] \setminus A$ is defined by $\mathbb{P}_S \sim \mu[S = B \cup A | A \in S]$. We denote this distribution by $\mu\big|_A$. A nice structural property of determinantal point processes is that they are closed under conditioning. In fact as we will explain later, this in fact holds for a more general family of probability distributions known as strongly Rayleigh measures. For DPPs, this can be understood more simply from a geometric point of view.

As stated, letting $L = VV^\intercal$ implies that the probability that the DPP defined by L assigned to each subset is proportional to the square of the volume spanned by the parallelepiped formed by the corresponding rows of V. For any subset $S \in [N]$ let V_S denotes rows of V indexed by S. Also, for any set of vectors U, let $\text{VOL}(U)$ indicate the volume of the spanned parallelepiped. We ignore the formal proof here, but it is straight-forward to verify the following: for any set $S \subset [N] \setminus A$,

$$\mu\big|_A(S) \propto \mu(S \cup A) \propto \text{VOL}(V_A) \cdot \text{VOL}(\{\Pi_{\langle V_A \rangle^\perp}(v)\}_{v \in V_S}).$$

A formal proof This implies that $\mu\big|_A$ is in fact the DPP defined by the projection of the vectors in $[N] \setminus A$ onto the space orthogonal to A.

3.2 k-DPPs

DPPs are probabilistic models which are widely used in practice to generate a diverse sample of items. A limitation of DPPs in practice is that they can assign a non-zero probability to every subset of items, and they do not offer any way to control the size of the sampled set. For example, in order to use DPPs to select a diverse subset of results to display on a search engine, the cardinality of the returned subset needs to be fixed.

Moreover, the way the diversity of subsets of items with different sizes is compared in DPPs, is not what one naturally expect in the following sense: As we discussed, from a geometric perspective, the diversity in DPPs is measured in terms of the notion of volume; For two subset of items represented by sets of vectors A and B, the correspond-

ing DPP assigns a higher probability to A if it spans a larger volume in the underlying space. When A and B contain the same number of items, this seems a reasonable criteria for diversity. However, it might not be a good candidate for comparing diversity when these subsets have very different number of elements. To see that, $|A| > |B|$ and $\text{VOL}(A) > \text{VOL}(B)$, thus the DPP assigns a higher probability to A. Now let see what happens when we scale all the feature vectors by factor of $\alpha < 1$; the volume of a subset of size k, scales with a factor of α^k which implies the volume of the subsets of larger reduces by a larger factor. As a result, in the new DPP, B might have a higher probability. This seems an undesirable property of the model, as one not expect that the relative diversity changes with a scaling of feature vectors.

These observations suggest that to improve DPPs for modeling diversity, we need to add some form of normalization or untangle the notion of size and diversity of a set. In [70], they take the latter approach and introduce the notion of k-DPPs by conditioning a DPP on returning subsets of size k.

Definition 3.3 (k-DPPs)**.** For an integer $k < n$, and a DPP μ, the k-DPP μ_k can be defined as a probability distributions over subsets of size k S of the domain we have:

$$\mu_k(S) \propto \begin{cases} \mu(S) & |S| = k \\ 0 & |S| \neq k \end{cases}$$

In other words, μ_k is a k-DPP defined by kernel L if for every subset S,

$$\mu_k(S) = \begin{cases} \frac{\det(L_S)}{\sum_{S':|S'|=k} \det(L_{S'})} & |S| = k \\ 0 & |S| \neq k \end{cases}$$

Despite the fact that, k-DPPs are obtained from DPPs by imposing a simple cardinality constraint, their mathematical properties can be very different. In particular, note that a k-DPP is not an instance of DPPs, and one can observe that even the uniform distribution on subsets of size k which is a very elementary k-DPP can not be expressed as a DPP. Therefore, in order to be able to use k-DPPs in practice, we at least need to have algorithms for the basic tasks. In the next part, we review some of the basic primitives

for k-DPPs. For more details we refer readers to [72].

3.2.1 *Algorithms for Basic Tasks for k-DPPs*

Normalization: To compute the normalization constant of a k-DPP distribution, we need to compute

$$\sum_{S':|S'|=k} \det(L_{S'}).$$

This can be done by examining the characteristic polynomial of L which is a uni-variate polynomial defined by

$$p_L(t) = \det(tI - L) = \prod_{i=1}^{n}(t - \lambda_i), \tag{3.4}$$

where $\lambda_1 \dots \lambda_n$ are eigenvalues of L. On the other hand, by definition of the determinant, one can see that,

$$\det(tI - L) = \sum_{i=0}^{n} t^{n-i}(-1)^i \sum_{S \in \binom{[N]}{i}} \det(L_S). \tag{3.5}$$

Setting the coefficient of t^{n-k} equal eq. (3.4) and eq. (3.5), one can see

$$\sum_{S':|S'|=k} \det(L_{S'}) = \sum_{S \in \binom{[N]}{k}} \prod_{i \in S} \lambda_i,$$

which is also known as the elementary symmetric polynomial of degree k Now, as shown in [70], given the eigen-decomposition of L, one can compute the above in time $O(Nk + k^2)$.

Marginalization. Fix a subset A with $|A| < k$. We are interested in computing $\mathbb{P}_S \sim \mu_k [A \subset S]$ which is the probability that a sample generated from our k-DPP μ_k contains elements of A. If we denote the normalization constant of μ_k by Z_k, by definition of μ_k we can write this probability as

$$\mathbb{P}_S \sim \mu_k [A \subset S] = \frac{1}{Z_k} \cdot \sum_{Y \in \binom{[N] \setminus A}{k-|A|}} \det(L_{A \cup Y}) \tag{3.6}$$

$$= \frac{\det(L + I)}{Z_k} \cdot \sum_{Y \in \binom{[N] \setminus A}{k-|A|}} \mathbb{P}_S \sim \mu [S = A \cup Y], \tag{3.7}$$

where the second equality holds as the normalizer of the DPP μ is $\det(L+I)$ as explained. Now, as we explained, the distribution of μ conditioning of containing A can be represented as another DPP, and the kernel of this projected DPP can be computed by essentially a matrix inversion. Let μ_A denote this distribution. We can compute section 3.2.1 using the following equation

$$\sum_{Y \in \binom{[N]\setminus A}{k-|A|}} \mathbb{P}_{S \sim \mu}[S = A \cup Y] = \sum_{Y \in \binom{[N]\setminus A}{k-|A|}} \mathbb{P}_{S \sim \mu_A}[S = Y]$$

Now, one can easily see the RHS of the above can be computed by computing the normalizer of μ_A and also the normalizer of the $(k-|A|)$-DPP obtained from μ_A.

3.3 Strongly Rayleigh Measures

As alluded to, the key property of DPPs and k-DPPs which makes them plausible probabilistic models to capture diversity is negative correlation, and its strongest form, *negative association*. DPPs inherit this property from a more general family of distributions known as Strongly Rayleigh (SR) distributions, which were first introduced and deeply studied in the work of [20]. In this part, we define these distributions, and review their properties which are useful to us in studying DPPs and k-DPPs. In order to define SR measures, we first need to introduce the notion of *generating polynomial* for a point process.

Definition 3.4. Let μ be a discrete point process on a domain with n elements. The generating polynomial associated with μ is a multi-affine polynomial p_μ over n variables $z_1, \ldots, z_n$ which is defined as

$$p_\mu(z_1, \ldots, z_n) = \sum_{S \in \text{supp}\{\mu\}} \mu(S) \boldsymbol{z}^S$$

where we use the notation $\boldsymbol{z}^S$ to denote $\prod_{i \in S} z_i$. We also say p_μ is *homogeneous* polynomial of degree k if each monomial is of degree k, i.e. any set in the support of μ has exactly k elements.

SR measures are defined as point processes whose generating polynomials belong to a specific family of polynomials called *real stable* polynomials. For a complex number

$z \in \mathbb{C}$ let $\mathrm{Im}(z)$ show its imaginary part. A polynomial p with real-valued coefficients defined over variables $z_1, \ldots, z_n$ is called a real stable polynomial if whenever for all values of z_i for $1 \leq i \leq n$, we have $\mathrm{Im}(z_i) > 0$, then we can deduce $p(z_1, \ldots, z_n) \neq 0$. For example the polynomial $p(\boldsymbol{z}) = \sum_{i=1}^{n} p_i z_i$ for $p_i \in \mathbb{R}_+$ is real stable. This can be easily verified by noting that $\mathrm{Im}(p(\boldsymbol{z})) > 0$, if $\mathrm{Im}(z_i) > 0$ for all i. With this algebraic notion, SR measures can be formally defined as:

Definition 3.5 (Strongly Rayleigh distribution)**.** A point process μ is a strongly Rayleigh (SR) measure if and only if its generating polynomial p_μ is a real stable polynomial.

Similar to the way we instantiate k-DPPs from DPPs, we can define k-homogeneous SR measures as point processes whose generating polynomial is real stable and also homogeneous of degree k.
Perhaps the simplest examples of SR measures are product distributions which can be defined as follows:
Product measures. Let $[N]$ be a ground space and for each $i \in [N]$ consider an independent Bernoulli distribution with probability $q_i \in [0, 1]$. That is the probability assigned to each subset $S \subseteq [N]$ is $\mu(S) = \prod_{i \in S} q_i \prod_{j \notin S} (1 - q_j)$.

Verifying that a product measure is an SR measures is straight-forward: Note that in this case the generating polynomial is given by $p_\mu(z_1, \ldots, z_n) = \prod_{i \in [N]} (q_i z_i + (1 - q_i))$, and this is a real stable polynomial as for any root of this polynomial we should have $q_i z_i + (1 - q_i) = 0$ for some i which implies $\mathrm{Im}(z_i)$ should be zero. However, for more complex distributions such as DPPs, it might seem a very hard task to check whether it is an SR measure by verifying the above criteria for real stability of its generating polynomial. However, it turns out that this complex criteria for multivariate polynomials can be reduced to a much simpler one for single variable polynomials. In particular, the following alternative criteria for real stability is given in [20].

Lemma 3.6 ([20])**.** *A polynomial $p(z_1, \ldots, z_n)$ with real coefficients is real stable if and only if*

for any $e \in \mathbb{R}^n_{>0}$ with positive coordinates and $x \in \mathbb{R}^n$, the univariate polynomial $p(x + te)$ has only real roots.

The above characterization can be used to conclude that DPPs are also belong to the family of SR measures.

Theorem 3.7 ([20]). *Any DPP is an SR measure.*

They also show that SR measures are closed under truncation. Combining that with the above, we get that k-DPPs are also instances of SR measures.

Corollary 3.8 ([20]). *Any k-DPP is an SR measure.*

This already shows the advantage of considering DPPs as SR measures for us, as it allows us to apply SR properties for k-DPPs. On the other hand, the above is not true for DPPs, that is to say k-DPPs can not be viewed as a DPP, and we can not extend DPPs properties directly to k-DPPs. In the rest of this section, we describe some properties of SR measures which are more important for us. We begin with negative association and, then explain some closure properties of SR measures. In the rest of section, let $\mu : 2^{[N]}$ be an SR measure whose generating polynomial is p_μ.

3.3.1 Negative Association

Earlier, we defined negative correlation for point processes and observe that DPPs satisfy this condition. Negative association is a stronger form of negative correlation which was introduced in [89]. We say an event $A \subseteq 2^{[N]}$ is increasing if it is closed upward under containment, i.e., if $S \in A$, and $S \subset T$ then $T \in A$. Moreover, we say a function $f : 2^{[N]} \to \mathbb{R}_+$ is increasing if it is the indicator function of an increasing event. We say μ is negatively associated if for any pair of increasing functions $f, g : 2^{[N]} \to \mathbb{R}_+$ which are depending on disjoint sets of coordinates, we have

$$\mathbb{E}_\mu [f] \cdot \mathbb{E}_\mu [g] \geq \mathbb{E}_\mu [f \cdot g] .$$

To see negative association implies negative correlation for any pair of elements $i, j \in [N]$, set f and g as functions indicating whether i and j belong to the set, respectively. It is straight-forward to see these two are closed upward under containment and also $\mathbb{E}_\mu[f] = \mathbb{P}_\mu[i \in S]$ and $\mathbb{E}_\mu[g] = \mathbb{P}_\mu[j \in S]$ which completes the proof. Building on [89], [20] proved that any strongly Rayleigh distribution is negatively associated.

Theorem 3.9 ([20]). *Any strongly Rayleigh probability distribution is negatively associated.*

This immediately implies DPPs and k-DPPs also satisfy negative association which is their main property that we exploit to develop our sampling algorithm in chapter 4 and chapter 5.

3.3.2 Closure Properties

In a brilliant sequence of papers Borcea and Brändén introduced a complete characterization of operators which preserve real stability of polynomials [19, 18]. Some instances of these operators can be naturally explained as actions on the corresponding SR measure. Here we focus on those operators and state the resulting closure properties for SR measures.

Conditioning. For any $1 \leq i \leq n$, let X_i be the random variable indicating whether i is in a sample of μ. We use $\mu\big|_i := \{\mu \mid X_i = 1\}$ to denote the conditional measure on sets that contain i and $\mu\big|_{\bar{i}} := \{\mu \mid X_i = 0\}$, to denote the conditional measure on sets that do not contain i. It is shown that that strongly Rayleigh distributions are closed under conditioning.

Theorem 3.10 (Theorem 2.5 of [20]). . *For any SR distribution μ and any $1 \leq i \leq n$, $\mu\big|_i$ and $\mu\big|_{\bar{i}}$ are also SR.*

Projection. For a subset $S \subseteq [N]$ define the projection of μ S, $\mu\big|_S$ as measure on 2^S which

the probability that assigns to any $A \subseteq S$ is

$$\mu\big|_S(A) = \sum_{B \subseteq [N]: B \cap S = A} \mu(B).$$

From the properties of real stable polynomials, it is easy to see that $\mu\big|_S$ is also SR.

Truncation. For an integer $1 \leq m \leq N$, the truncation of μ to subsets of size m is the conditional measure on subsets of size m. In other words, if we denote it by μ_m, for any $S \subseteq [N]$ we have

$$\mu_m(S) \propto \begin{cases} 0 & |S| \neq m \\ \mu(S) & \text{otherwise} \end{cases}$$

3.4 DPPs on a Continuous Domain

Recall that, discrete k-DPPs are in fact random processes that select a subset of k points from a finite ground set, each with probability proportional to the determinant of the corresponding sub-matrix of the kernel. Similarly, continuous k-DPPs can be defined over a continuous domain via a PSD operator under some conditions which are covered in section 3.4.1. In order to give a mathematically precise definition of continuous k-DPPs, we need to introduce some operator theory nuances that we ignore, since the following simpler definition of continuous k-DPPs suffices to understand this thesis. Interested readers can found more details about continuous DPPs in [57].

Definition 3.11 (continuous k-DPP)**.** Let $\Lambda \subseteq \mathbb{R}^d$ be a closed subset of the d-dimensional euclidian space, and let $L : \Lambda \times \Lambda \to \mathbb{R}$ be a PSD operator defined on this domain. Continuous k-DPP defined on the ground set Λ by the ensemble kernel L is a distribution π supported on set of k-points in Λ that satisfies the following condition: For any mutually disjoint family of compact subsets $D_1, \ldots, D_k \subseteq \Lambda$,

$$\mathbb{P}_\pi\left[\text{one point is selected from each } D_i\right] \propto \int_{D_1} \cdots \int_{D_k} \det_L(x_1, \ldots, x_k) dx_1 \ldots dx_k$$

where $\det_L(x_1, \ldots, x_k)$ refers to the determinant of the $k \times k$ matrix given by $\{L(x_i, x_j\}_{1 \leq i,j \leq k}$.

Examples. Here, we present some of the PSD operators which are commonly used as

kernels.

- Gaussian quality and similarity kernel: Let $\phi(x) = \exp(\frac{-\|x-a\|^2}{\sigma^2})$ be a multi-variate Gaussian distribution with mean $a \in \mathbb{R}^d$ and variance $\sigma > 0$. Also let $K : \mathbb{R}^d \times \mathbb{R}^d$ be a Gaussian kernel defined by $K(x,y) = \exp(\frac{-\|x-y\|^2}{\nu^2})$. Then one can use $L(x,y) = \phi(x)K(x,y)\phi(y)$ to define a k-DPP over any subset of $\mathbb{R}^d$. Intuitively speaking, here ϕ gives a quality score to each point based on its distance from the center a and K captures diversity of selected points in the k-DPP. This is why ϕ is called the quality kernel and K is the similarity kernel.

- Polynomial kernel: Another widely used family of kernels are polynomial kernels. For an integer ℓ, the polynomial kernel of degree ℓ is defined by
$$P(x,y) = (1 + \langle x, y\rangle)^{\ell}.$$

More generally, in the next part we describe minimal requirements for a function to define a k-DPP on a continuous domain.

3.4.1 Necessary Conditions for a Continuous DPP Kernel

Let $L : \Lambda \times \Lambda \to \mathbb{R}$ be a continuous function, and let $\mathcal{T}_L$ be its corresponding Hilbert-Schmidt integral operator which for any function $f \in L^2(\Lambda)$ is given by
$$\mathcal{T}_L(f)(x) = \int_C L(x,y)f(y)dy.$$
We may abuse the notation and use L to also represent the corresponding integral operator. Function L needs to satisfy the following to define a DPP.

1. L is a symmetric function, i.e. for any x, y, $L(x,y) = L(y,x)$. This also implies L is self-adjoint which means for any $f, g \in L^2(\Lambda)$, we have $\langle Lf, g\rangle = \langle f, Lg\rangle$.

2. $\mathcal{T}_L$ is a Hilbert-Schmidt kernel which means $\int_\Lambda \int_\Lambda \|L(x,y)\|^2 dxdy < \infty$. And most importantly

3. L satisfies Mercer's condition: Any restriction of L to a finite domain gives a PSD matrix, i.e. for any n, and $x_1, \ldots, x_n \in \Lambda$ and any set of values $c_1, \ldots, c_n \in \mathbb{R}$, we have $\sum_{i,j=1}^{n} L(x_i, x_j) c_i c_j \geq 0$. This along with continuity of L implies that $\mathcal{T}_L$ is a PSD operator. Therefore, there is a function f that maps any point in Λ to some Hilbert space such that for any $x, y \in \Lambda$:
$$L(x, y) = \langle f(x), f(y) \rangle.$$

One important consequence of the above conditions is the Mercer's theorem which essentially is a generalization of the eigenvalue decomposition for PSD matrices in the continuous domain.

Theorem 3.12 (Mercer's theorem)**.** *For any functions L satisfying the conditions* (1)-(3) *there is a countable system of eigenspaces and eigenvalues, i.e. there are non-negative eigenvalues $\lambda_1, \lambda_2 \ldots$, and $\{\phi_i\}_{i=1}^{\infty} \subset L^2(\Lambda)$ where for any x and y*
$$L = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(y).$$

Part I

SAMPLING FROM K-DPPS AND SR MEASURES

Chapter 4

MCMC ALGORITHMS FOR SAMPLING FROM DISCRETE K-DPPS AND HOMOGENEOUS SR MEASURES

4.1 Introduction

Although, k-DPP are obtained from DPPs by a simple truncation operation, in contrast to DPPs, mathematics of k-DPP can be very different. In particular, it turns out that a k-DPP distribution may not be representable as a DPP. Perhaps, the simplest example is the k-uniform distribution over a set of n elements. Although the uniform distribution over n elements is a DPP, for any $2 \leq k \leq n-2$, the corresponding k-DPP is not a DPP [72, Section 5]. To study sampling algorithms for k-DPPs, we appeal to properties of a broader family of probability distributions known as Strongly Rayleigh (SR) measures (see definition 3.5 for the definition). These generalization of DPPs, are introduced and deeply studied in the work of [20]. Most importantly to us it is shown in [20] that

1. unlike DPPs, SR measures are closed under truncation.

2. similar to DPPs, they satisfy strongest form of negative dependence, a.k.a. negative association.

These negative dependence properties were recently exploited to design approximation algorithms [102, 105, 7].

We use these properties (crucially item 2) to study sampling algorithms for k-homogeneous

SR measures[1], which are a generalization of k-DPPs. We prove that the "natural" Metropolis-Hastings Markov Chain defined on the support of these distributions *mixes* rapidly. Let π be k-homogeneous SR measure on domain $[N]$, i.e. π is the truncation of a SR measure to subsets of size k from $[N]$. We analyze the mixing time of the following Markov chain: The state space of $\mathcal{M}_\pi$ is $\text{supp}(\pi)$ and the transition probability kernel P_π is defined as follows. We may drop the subscript if π is clear from the context. For a set $S \subseteq [N]$ and $i \in [N]$, let

$$\begin{aligned} S - i &= S \setminus \{i\}, \\ S + i &= S \cup \{i\}. \end{aligned}$$

In any state S, choose an element $i \in S$ and $j \notin S$ uniformly and independently at random, and let $T = S - i + j$; then

i) If $T \in \text{supp}\{\pi\}$, move to T with probability $\frac{1}{2}\min\{1, \pi(T)/\pi(S)\}$;

ii) Otherwise, stay in S.

Our main contribution is to analyze the mixing time of the above chain.

4.1.1 Sampling from Discrete k-DPPs

Generating a sample from a k-DPP is a fundamental computational task with many practical applications [66, 35, 72]. Moreover, as pointed out, DPPs can have geometric interpretation as well: Given a set of vectors, the probability that a k-DPP assign to a subset of vectors size k is proportional to the volume of k-dimensional parallelogram formed by them. With this terminology, the problem of sampling from k-DPPs is also known as the k-volume sampling.

Definition 4.1 (k-volume Sampling)**.** We are given a matrix $V \in \mathbb{R}^{n \times m}$, an integer k, and we want to choose a set $S \subseteq [n]$ of k rows of X with probability proportional to

[1] As explained, these are truncation of SR measures to subsets of size k.

$\det(V_{S,[m]}, V_{S,[m]}^{\mathsf{T}})$, where $V_{S,[m]}$ is the submatrix of V with rows indexed by elements of S.

Therefore, if L is the ensemble matrix of a given k-DPP π, and $L = VV^{\mathsf{T}}$ is the Cholesky decomposition of L, then the k-volume sampling problem on X is equivalent to the problem of generating a random sample of π. This is a well-studied problem due to its connections to low-rank approximations of matrices, e.g. see [66, 23, 35, 36, 37]; it is shown in [35] that generating a sample of k rows of V from this distribution and projecting all other rows of onto their span gives a $(k+1)$-approximation to the nearest rank-k matrix to V under the Frobenius norm.

The first type of algorithms developed for sampling from discrete k-DPPs are spectral methods, e.g. see [58, 35, 72]. These methods can generate exact samples from the distribution. However, they are only efficient when this matrix V is known, otherwise a time consuming preprocess is required to decompose L- especially for high rank matrices - and obtain V which makes them inefficient in terms of time and memory for many DPP applications. In particular, when the input k-DPP is given by its ensemble kernel L, the spectral method of [35] needs $O(knm^{\omega} \log n)$, for $\omega \approx 2.37$ being the constant for matrix multiplication, to execute[2].

Given the limitation of spectral techniques, it was asked by [35] to generate random samples of a k-DPP using Markov chain techniques. Markov chain techniques are very appealing in this context because of their simplicity and efficiency.

There has been several attempts [65, 83, 109] to upper bound the mixing time of the Markov chain $\mathcal{M}_{\pi}$ for a k-DPP π; but, to the best of our knowledge this question is still open[3]. Here, we design the first rigorously analyzed MCMC algorithms for sampling

[2]We remark that the algorithms in [35] are almost linear in n when V is given.

[3]We remark that [65] claimed to have a proof of the rapid mixing time of a similar Markov chain. As it is pointed out in [109] the coupling argument of [65] is ill-defined. To be more precise, the chain specified

from discrete k-DPPs by upper-bounding the mixing time of $\mathcal{M}_\pi$.

4.1.2 *Results*

It is straight-forward to see that $\mathcal{M}_\pi$ is reversible and $\pi(.)$ is the stationary distribution of the chain. In addition, Brändén showed that the support of a (homogeneous) strongly Rayleigh distribution is the set of bases of a matroid [24, Cor 3.4]; so $\mathcal{M}_\pi$ is irreducible. Lastly, since we stay in each state S with probability at least $1/2$, $\mathcal{M}_\pi$ is a lazy chain.

In our main theorem, we analyze the "spectral gap" of $\mathcal{M}_\pi$ and combining with the above facts conclude that for any state $S \in \text{supp}(\pi)$, if we start $\mathcal{M}_\pi$ from a state S, then after $\text{poly}(N, k, \log(\frac{1}{\epsilon \cdot \pi(S)}))$ steps we obtain an ϵ-approximate sample of the input homogeneous SR distribution. Formally we prove the following theorem.

Theorem 4.2. *For any strongly Rayleigh k-homogeneous probability distribution $\pi : 2^{[N]} \to \mathbb{R}_+$, $S \in \text{supp}\{\pi\}$ and $\epsilon > 0$,*

$$\tau_S(\epsilon) \leq \frac{1}{C_\pi} \cdot \log\left(\frac{1}{\epsilon \cdot \pi(S)}\right),$$

where $P^t_\pi(S, .)$ be the distribution of $\mathcal{M}_\pi$ started at S at time t and

$$C_\pi := \min_{S,T:\, P(S,T)>0} \max(P(S,T), P(T,S)) \tag{4.1}$$

is at least $\frac{1}{2kn}$ by construction.

See section 2.2 for the definition of mixing time. We remark that the homogeneity assumption is necessary for the above theorem. The mixing time of Markov chain is closely related to their Poincaré constant and the relationship is given by the classical theorem theorem 2.13 of Diaconis and Stroock. Using that, to prove theorem 4.2 we only need to prove the following theorem.

Theorem 4.3. *For any k-homogeneous strongly Rayleigh distribution $\pi : 2^{[N]} \to \mathbb{R}_+$, the*

in Algorithm 1 of [65] may not mix in a polynomial time of n. The chain specified in Algorithm 2 of [65] is similar to $\mathcal{M}_\pi$, but the statement of Theorem 2 which upper bounds its mixing time is clearly incorrect even when $k = 1$.

Poincaré constant of the chain $\mathcal{M}_\pi$ is at least

$$\lambda \geq C_\pi.$$

Discrete k-DPPs are special cases of k-homogeneous SR measures. So, we get the following corollary.

Corollary 4.4. *For any k-DPP π, $S \in \text{supp}(\pi)$ and $\epsilon > 0$,*

$$\tau_S(\epsilon) \leq \frac{1}{C_\pi} \cdot \log\left(\frac{1}{\epsilon \cdot \pi(S)}\right).$$

Suppose we have access to a set $S \in \text{supp}\{\pi\}$ such that $\pi(S) \geq \exp(-n)$. In addition, we are given an oracle such that for any set $T \in \binom{N}{k}$, it returns $\pi(T)$ if $T \in \text{supp}(\pi)$ and zero otherwise. Then, by the above theorem we can generate an ϵ-approximate sample of k-homogeneous SR measures with at most $\text{poly}(N, k, \log(1/\epsilon))$ oracle calls. For k-DPPs, we provide these oracles and conclude $\mathcal{M}_\pi$ can be used to efficiently generate an approximate sample of π. More precisely we prove the following theorem.

Theorem 4.5. *Given kernel L of a k-DPP π, for any $\epsilon > 0$, there is an algorithm that generates an ϵ-approximate sample of π in time $\text{poly}(k)O(n \log(n/\epsilon))$.*

Note that in order to run the Markov chain $\mathcal{M}_\pi$, it is only enough to have an algorithm that given any pair of subsets $T, S \in \binom{[N]}{k}$, computes the ratio $\frac{\pi(T)}{\pi(S)}$. For k-DPPs it only requires computing the determinant of two $k \times k$ matrices which can be done in time $O(k^3)$. Therefore, to obtain an actual algorithm and prove theorem 4.5 it remains to design an oracle to find a proper starting state; We need to generate a set $S \in \text{supp}(\pi)$ such that $\pi(S)$ is bounded away from zero, perhaps by an exponentially small function of n, k. We use the greedy algorithm 4.1 to find such a set, and we show that, in time $O(n)\text{poly}(k)$, it returns a set S such that

$$\pi(S) \geq \frac{1}{k!|\text{supp}\{\pi\}|} \geq n^{-2k}. \tag{4.2}$$

Algorithm 4.1 Greedy Algorithm for Selecting the Starting State of $\mathcal{M}_\pi$

1: $S \leftarrow \emptyset$
2: **for** $i = 1$ to k **do**
3: Among all elements $j \notin S$ pick the one maximizing $\det(L_{S+j})$ and let $S \leftarrow S + j$.
4: **end for**
5: Return S.

It remains to analyze Algorithm 4.1. This problem is already studied by [31] in the context of maximum volume submatrix problem. In the maximum volume submatrix problem, given a matrix $X \in \mathbb{R}^{n\times m}$, we want to choose a subset S of k rows of X maximizing $\det(X_{S,[m]}X_{S,[m]}^\intercal)$. Equivalently, given a matrix $L = XX^\intercal$, we want to choose $S \subseteq [n]$ of size k maximizing $\det(L_S)$. Note that if L is an ensemble matrix of a k-DPP π, then

$$\max_{|S|=k} \pi(S) = \frac{\max_{|S|=k} \det(L_S)}{\sum_{|S|=k} \det(L_S)} \geq \frac{1}{|\mathrm{supp}(\pi)|} \geq n^{-k}.$$

The maximum volume submatrix problem is NP-hard to approximate within a factor c^k for some constant $c > 1$ [32]. Numerous approximation algorithm are given for this problem [31, 32, 99]. It was shown in [31, Thm 11] that choosing the rows of X greedily gives a $k!$ approximation to the maximum volume submatrix problem. Algorithm 4.1 is equivalent to the greedy algorithm of [31]; it is only described in the language of ensemble matrix L. Therefore, it returns a set S such that

$$\pi(S) \geq \frac{\max_{|T|=k} \det(L_T)}{k! \sum_{|T|=k} \det(L_T)} \geq \frac{1}{k!|\mathrm{supp}(\pi)|},$$

which establishes eq. (4.2). The complete algorithm can be summarized as follows:

Algorithm 4.2 An MCMC algorithm to generate approximate samples from k-DPPs

1: **Input:** A k-DPP π on $[N]$ defined by kernel $L \in \mathbb{R}^{N \times N}$.
2: Use algorithm 4.1 to find a starting subset S.
3: **for** $i = 1$ to $\tau_S(\epsilon) = O(nk \log \frac{1}{\epsilon})$ **do** Choose an element $i \in S$ and $j \notin S$ uniformly and independently at random, and let $T = S - i + j$; then
   i) If $T \in \text{supp}(\pi)$, move to T, i.e. set $S \leftarrow T$, with probability $\frac{1}{2} \min\{1, \pi(T)/\pi(S)\}$.
   ii) Otherwise, stay in S.
4: **end for**
5: Return S.

Overall Running Time. Note that the number of steps as bounded by corollary 3.8 is $O(nk \log \frac{1}{\epsilon})$, and the complexity of each step is $O(k^3)$. So the total running time of the algorithm to generate an ϵ-approximate sample is $O(nk^4 \log \frac{1}{\epsilon})$.

4.1.3 Proof Overview

In the rest of the paper we prove theorem 4.2. To prove theorem 4.2, we lower bound the spectral gap, a.k.a. the Poincaré constant of the chain $\mathcal{M}_\pi$ and prove theorem 4.3. To lower bound the spectral gap, we use an extension of the seminal work of [47]. Feder and Mihail showed that the *bases exchange* graph of the bases of a *balanced matroid* is an *expander*. This directly lower bounds the spectral gap by Cheeger's inequality. A matroid is called balanced if the matroid and all of its minors satisfy the property that the uniform distribution of the bases is negatively associated (see section 3.3.1 for the definition).

Our proof can be seen as a *weighted* variant of [47]. As we mentioned earlier, the support of a homogeneous strongly Rayleigh distribution corresponds to the bases of a matroid. Our proof shows that if a distribution μ over the bases of a matroid and all of its con-

ditional measures are negatively associated, then the MCMC algorithm mixes rapidly. To show that μ satisfies the aforementioned property we simply appeal to the negative dependence theory of strongly Rayleigh distributions developed in [20]. Although our proof can be written in the language of [47], we work with the more advanced chain decomposition idea of [63] to prove a tight bound on the Poincaré constant; see section 4.2 for the details.

We remark that the decomposition idea of [63] can be used to lower bound the *log-Sobolev* constant of $\mathcal{M}_\mu$. However, it turns out that in our case, the log-Sobolev constant may be no larger than $\frac{1}{-\log(\min_{S\in\text{supp}\{\mu\}}\mu(S))}$. Since the latter quantity is not necessarily lower-bounded as a function of k, n, the L_2 mixing time of the chain may be unbounded.

4.2 Decomposable Markov Chains

Let $\mathcal{M}$ be a Markov chain on a finite space identified by (Ω, P, π) which respectively denote the state space, transition probability matrix, and the stationary distribution. Recall that a $\mathcal{M}$ is reversible if for any pair of states $x, y \in \Omega$, $\pi(x)P(x,y) = \pi(y)P(y,x)$. In this chapter we only work with reversible Markov chains.

Our main tool to lower bound the Poincaré constant of $\mathcal{M}_\pi$ is the decomposable Markov chain technique due to Jerrum, Son, Tetali and Vigoda [63]. Roughly speaking, they consider Markov chains that can be decomposed into "projection" and "restriction" chains. They lower bound the Poincaré constant of the original chain assuming certain properties of these projection/restriction chains.

Let $\Omega_0 \cup \Omega_1$ be a decomposition of the state space of a Markov chain (Ω, P, π) into two disjoint sets[4]. For $i \in \{0,1\}$ let

$$\bar{\pi}(i) = \sum_{x\in\Omega_i} \pi(x),$$

[4]Here, we only focus on decomposition into two disjoint sets, although the technique of [63] is more general.

and let $\bar{P} \in \mathbb{R}^{2\times 2}$ be

$$\bar{P}(i,j) = \bar{\pi}(i)^{-1} \sum_{x\in\Omega_i, y\in\Omega_j} \pi(x)P(x,y).$$

The Markov chain $(\{0,1\}, \bar{P}, \bar{\pi})$ is called a projection chain. Let $\bar{\lambda}$ be the Poincaré constant of this chain.

We can also define a restriction Markov chain on each Ω_i as follows. For each $i \in \{0,1\}$,

$$P_i(x,y) = \begin{cases} P(x,y) & \text{if } x \neq y, \\ P(x,x) + \sum_{z\notin\Omega_i} P(x,z) & \text{if } x = y. \end{cases}$$

In other words, for any transition from x to a state outside of Ω_i, we remain in x. Observe that in the stationary distribution of the restriction chain, the probability of x is proportional to $\pi(x)$. Let λ_i be the Poincaré constant of the chain $(\Omega_i, P_i, .)$. Now, we are ready to explain the main result of [63].

Theorem 4.6 ([63, Cor 3]). *If for any distinct $i,j \in \{0,1\}$, and any $x \in \Omega_i$,*

$$\bar{P}(i,j) = \sum_{y\in\Omega_j} P(x,y), \tag{4.3}$$

then the Poincaré constant of (Ω, P, π) is at least $\min\{\bar{\lambda}, \lambda_0, \lambda_1\}$.

Note that the projection chain in this case is a Markov chain with only two states. The spectral gap of Markov chains with two states can be easily calculated. In particular, we use the following.

Fact 7. *The Poincaré constant of any reversible two state chain with $\Omega = 0,1$ and $P(0,1) = c \cdot \pi(1)$ is c.*

Proof. Consider any function f. Since $\mathrm{var}(f)$ is shift-invariant, we can assume $\mathbb{E}_\pi[f] = 0$, i.e., $\pi(0)f(0) = -\pi(1)f(1)$. Since $\frac{\mathcal{E}_\pi(f,f)}{\mathrm{var}_\pi(f)}$ is invariant under the scaling of f, we can assume $f(0) = \pi(1)$ and $f(1) = -\pi(0)$. Since the chain is reversible $P(1,0) = c \cdot \pi(0)$. Plugging this unique f into the ratio we obtain $\lambda = c$. □

4.3 *Inductive Argument*

In this section we prove theorem 4.3. Throughout this section we fix a strongly Rayleigh distribution π, and we let Ω, P be the state space and the transition probability matrix of $\mathcal{M}_\pi$.

We prove theorem 4.3 by induction on $|\text{supp}(\pi)|$. If $|\text{supp}(\pi)| = 1$, then there is nothing to prove. To do the induction step, we will use theorem 4.6. So, let us first start by defining the restriction chains. Without loss of generality, perhaps after renaming, let n be an element such that $0 < \mathbb{P}_S \sim \pi[n \in S] < 1$. Let $\Omega_0 = \{S \in \text{supp}(\pi) : n \notin S\}$ and $\Omega_1 = \{S \in \text{supp}(\pi) : n \in S\}$. Note that both of these sets are nonempty. Observe that the restricted chain $(\Omega_0, P_0, .)$ is the same as $\mathcal{M}_{\pi|_{\overline{n}}}$ and $(\Omega_1, P_1, .)$ is the same as $\mathcal{M}_{\pi|_n}$. In addition, by theorem 3.10, $\pi|_{\overline{n}}$ and $\pi|_n$ are strongly Rayleigh, and also clearly $C_{\pi|_n}, C_{\pi|_{\overline{n}}} \geq C_\pi$. So, we can use the induction hypothesis to lower bound $\lambda_0, \lambda_1 \geq C_\pi$.

It remains to lower bound the Poincaré constant of the projection chain and to prove equation (4.3). Unfortunately, P does not satisfy (4.3). So, we use an idea of [63]. We construct a new Markov kernel $\hat{P}$ satisfying (4.3) such that (i) $\hat{P}$ has the same stationary distribution. (ii) The Poincaré constant of $\hat{P}$, $\hat{\lambda}$ lower-bounds λ. Then we use theorem 4.6 to lower bound $\hat{\lambda}$.

To make sure that $\hat{P}$ satisfies (i), (ii), it is enough that for all distinct states $x, y \in \Omega$,

$$\pi(x)\hat{P}(x,y) = \pi(y)\hat{P}(y,x), \tag{4.4}$$

$$\hat{P}(x,y) \leq P(x,y). \tag{4.5}$$

Equation (4.4) implies (i), i.e., that π is also the stationary distribution of $\hat{P}$. By an application of the comparison method [40] (i) together with (4.5) implies (ii), i.e.,

$$\hat{\lambda} \leq \lambda. \tag{4.6}$$

So, to prove the induction step, it is enough to show that

$$\hat{\lambda} \geq C_\pi. \tag{4.7}$$

Lemma 4.8. *There is a transition probability matrix $\hat{P} : \Omega \times \Omega \to \mathbb{R}_+$ such that*

1) $\hat{P}$ satisfies (4.4), (4.5).

2) For any $i \in \{0,1\}$ and any distinct states $x, y \in \Omega_i$, $\hat{P}(x,y) = P(x,y)$.

3) The Poincaré constant of the chain $(\Omega, \hat{P}, \pi)$ projected onto Ω_0, Ω_1 is at least $\bar{\hat{\lambda}} \geq C_\pi$,

4) For any state $x \in \text{supp}(\pi)$ and distinct $i, j \in \{0,1\}$,

$$\bar{\hat{P}}(i,j) = \sum_{y \in \Omega_j} \hat{P}(x,y).$$

Before proving the above lemma, we use it to finish the proof of the induction. By part (2), $\hat{P}$ agrees with P on the projection chains. Therefore, the Poincaré constants of the chains $(\Omega_0, \hat{P}_0, .)$ and $(\Omega_1, \hat{P}_1, .)$ are at least $\hat{\lambda}_0, \hat{\lambda}_1 \geq C_\pi$. So, by parts (3) and (4) we can invoke theorem 4.6 for $\hat{P}$ and we get that

$$\hat{\lambda} \geq \min\{\bar{\hat{\lambda}}, \hat{\lambda}_0, \hat{\lambda}_1\} \geq C_\pi.$$

This proves (4.7). As we discussed earlier, part (1) implies (4.6) which completes the induction.

4.3.1 *Proof of lemma 4.8*

In the rest of this section we prove lemma 4.8. Note that the main challenge in proving the lemma is part (4). The transition probability matrix P already satisfies part (1)-(3). The key to proving part (4) is to construct a fractional perfect matching between the states of Ω_0 and Ω_1; see the following lemma for the formal definition. This idea originally was used in [47] and it was later extended in [62].

Lemma 4.9. *There is a function $w : \{\{x,y\} : x \in \Omega_0, y \in \Omega_1\} \to \mathbb{R}_+$ such that $w_{\{x,y\}} > 0$*

only if $P(x,y) > 0$ *and*

$$\begin{aligned} \sum_{y\in\Omega_1} w_{\{x,y\}} &= \frac{\pi(x)}{\pi(\Omega_0)} \quad \forall x \in \Omega_0, \\ \sum_{x\in\Omega_0} w_{\{x,y\}} &= \frac{\pi(y)}{\pi(\Omega_1)} \quad \forall y \in \Omega_1. \end{aligned} \tag{4.8}$$

We use the negative association property of the strongly Rayleigh distributions to prove the above lemma. But before that let us prove lemma 4.8.

Proof of lemma 4.8. We use w to construct $\hat{P}$. For any $i,j \in \{0,1\}$ and $x \in \Omega_i$ and $y \in \Omega_j$ where $x \neq y$, we let

$$\hat{P}(x,y) = \begin{cases} \frac{C_\pi}{\pi(x)}\pi(\Omega_i)\pi(\Omega_j)w_{\{x,y\}} & \text{if } i \neq j, \\ P(x,y) & \text{otherwise.} \end{cases}$$

We also set $\hat{P}(x,x) = 1 - \sum_{y\neq x\in\Omega} \hat{P}(x,y)$ for any $x \in \Omega$. Note that by definition part (2) is satisfied. First we verify part (1). If $i \neq j$, then

$$\hat{P}(x,y)\pi(x) = C_\pi \pi(\Omega_i)\pi(\Omega_j)w_{\{x,y\}} = \hat{P}(y,x)\pi(y),$$

and if $i = j$ the same identity holds because $\hat{P}(x,y) = P(x,y)$. This proves (4.4). To see (4.5), let $x \in \Omega_i, y \in \Omega_j$ be two distinct states. First note that WLOG we can assume $i \neq j$ and $P(x,y) \neq 0$; otherwise clearly $\hat{P}(x,y) = P(x,y)$. So we have

$$\begin{aligned} \hat{P}(x,y) &= \frac{C_\pi}{\pi(x)} \cdot \pi(\Omega_0)\pi(\Omega_1)w_{\{x,y\}} \\ &\leq \frac{\max(P(x,y),P(y,x))}{\pi(x)}\pi(\Omega_i)\pi(\Omega_j)w_{\{x,y\}} \\ &\leq \max(P(x,y),P(y,x)) \cdot \frac{\min(\pi(x),\pi(y))}{\pi(x)} \leq P(x,y). \end{aligned}$$

The first inequality follows by the definition of C_π (see (4.1)), and the second inequality follows by the fact that $w_{\{x,y\}} \leq \frac{\pi(x)}{\pi(\Omega_0)}$ and $w_{\{x,y\}} \leq \frac{\pi(y)}{\pi(\Omega_1)}$, and the last inequality follows by the detailed balanced condition. This completes the proof of part (1).

Next, we prove part (3). By the definition of $\hat{P}$, for distinct $i, j \in \{0,1\}$ we have

$$\begin{aligned}\bar{\hat{P}}(i,j) &= \frac{1}{\pi(\Omega_i)} \sum_{x\in\Omega_i, y\in\Omega_j} \pi(x)\hat{P}(x,y) \\ &= \frac{C_\pi}{\pi(\Omega_i)} \sum_{x\in\Omega_i, y\in\Omega_j} \pi(\Omega_i)\pi(\Omega_j)w(x,y) \\ &= C_\pi \cdot \pi(\Omega_j) \sum_{x\in\Omega_i} \frac{\pi(x)}{\pi(\Omega_i)} = C_\pi \cdot \pi(\Omega_j),\end{aligned}$$

where the second to last equality follows by (4.8). By Fact 7, the Poincaré constant of $\bar{\hat{P}} = C_\pi$. This proves part (3).

Finally we prove part (4). Fix distinct $i, j \in \{0,1\}$ and $z \in \Omega_i$. We have,

$$\sum_{y\in\Omega_j} \hat{P}(z,y) = \frac{C_\pi}{\pi(z)}\pi(\Omega_i)\pi(\Omega_j) \sum_{y\in\Omega_j} w_{\{z,y\}} = C_\pi \cdot \pi(\Omega_j),$$

where we used (4.8). On the other hand, by the definition of $\hat{P}$ we know that

$$\bar{\hat{P}}(i,j) = \frac{1}{\pi(\Omega_i)} \sum_{x\in\Omega_i, y\in\Omega_j} \pi(x)\bar{\hat{P}}(x,y) = C_\pi \cdot \pi(\Omega_j) \sum_{x\in\Omega_i} \frac{\pi(x)}{\pi(\Omega_i)} = C_\pi \cdot \pi(\Omega_j),$$

where the second equality follows by (4.8). This completes the proof of part (4) and lemma 4.8. □

It remains to prove lemma 4.9. For a set $A \subseteq \Omega$ let

$$N(A) = \{y \in \Omega \setminus A : \exists x \in A, P(x,y) > 0\}.$$

To prove lemma 4.9 we use a maximum flow-minimum cut argument. To prove the claim we need to show that the support graph of the transition probability matrix P_π satisfies Hall's condition. This is proved in the following lemma using the negative association property of strongly Rayleigh measures. The proof is simply an extension of the proof of [47, Lem 3.1].

Lemma 4.10. *For any $A \subseteq \Omega_1$,*

$$\frac{\pi(N(A))}{\pi(\Omega_0)} \geq \frac{\pi(A)}{\pi(\Omega_1)}.$$

Proof. Let $R \sim \pi$ be a random set. Recall that $\Omega_0 = \{S \in \text{supp}\{\pi\} : n \notin S\}$ and $\Omega_1 = \{S \in \text{supp}\{\pi\} : n \in S\}$. Let g be a random variable indicating whether $n \in R$. Let

f be an indicator random variable which is 1 if there exists $T \in A$ such that $R \supseteq T \setminus \{n\}$. It is easy to see that f and g are two increasing functions which depend on two disjoint sets of elements. By the negative association property, theorem 3.9, we can write

$$\mathbb{P}_\pi \left[f(R) = 1 | g(R) = 0 \right] \geq \mathbb{P}_\pi \left[f(R) = 1 | g(R) = 1 \right].$$

The lemma follows by the fact that the LHS of the above inequality is $\frac{\pi(N(A))}{\pi(\Omega_0)}$ and the RHS is $\frac{\pi(A)}{\pi(\Omega_1)}$. □

Proof of lemma 4.9. Let G be a bipartite graph on $\Omega_0 \cup \Omega_1$ where there is an edge between $x \in \Omega_1$ and $y \in \Omega_0$ if $P(x, y) > 0$. We prove the lemma by showing there is a unit flow from Ω_1 to Ω_0 such that the amount of flow going out of any $x \in \Omega_1$ is $\frac{\pi(x)}{\pi(\Omega_1)}$, and the incoming flow to any $y \in \Omega_0$ is $\frac{\pi(y)}{\pi(\Omega_0)}$. Then, we simply let $w_{\{x,y\}}$ be the flow on the edge connecting x to y.

Add a source s and a sink t. For any $x \in \Omega_1$ add an arc (s, x) with capacity $c_{s,x} = \pi(x)/\pi(\Omega_1)$. Similarly, for any $y \in \Omega_0$ add an arc (y, t) with capacity $c_{y,t} = \pi(y)/\pi(\Omega_0)$. Let the capacity of any other edge in the graph be ∞. Since the sum of the capacities of all edges leaving s is 1, to prove the lemma, it is enough to show that the maximum flow is 1. Equivalently, by the max-flow min-cut theorem, it suffices to show that the value of the minimum cut separating s and t is at least 1. Let $B, \overline{B}$ be an arbitrary s-t cut, and assume that $s \in B$ and $t \in \overline{B}$. Let $B_0 = \Omega_0 \cap B$ and $B_1 = \Omega_1 \cap B$. For disjoint $X, Y \subseteq \Omega$, let $c(X, Y) = \sum_{x \in X, y \in Y} c_{x,y}$. We have

$$\begin{aligned} c(B, \overline{B}) &\geq c(s, \Omega_1 \setminus B_1) + c(B_0, t) \\ &= \frac{\pi(\Omega_1 \setminus B_1)}{\pi(\Omega_1)} + \frac{\pi(B_0)}{\pi(\Omega_0)} \\ &= 1 - \frac{\pi(B_1)}{\pi(\Omega_1)} + \frac{\pi(B_0)}{\pi(\Omega_0)} \geq 1 - \frac{\pi(N(B_1))}{\pi(\Omega_0)} + \frac{\pi(B_0)}{\pi(\Omega_0)}, \end{aligned} \tag{4.9}$$

where the inequality follows by lemma 4.9. If there are any edge from B_1 to $\Omega_0 \setminus B_0$, then $c(B, \overline{B}) = \infty$ and we are done. Otherwise, $N(B_1) \subseteq B_0$. Therefore, $\pi(N(B_1)) \leq \pi(B_0)$, and the RHS of the above inequality is at least 1. So, $c(B, \overline{B}) \geq 1$ as desired. □

Chapter 5

A POLYNOMIAL TIME MCMC METHOD FOR SAMPLING FROM CONTINUOUS K-DPPS

In this chapter, we study the Gibbs sampling algorithm for discrete and continuous k-DPPs. We show that in both cases, the spectral gap of the chain is bounded by a polynomial of k and it is independent of the size of the domain. As an immediate corollary, we obtain sublinear time algorithms for sampling from discrete k-DPPs given access to polynomially many processors. In the continuous setting, our result leads to the first class of rigorously analyzed efficient algorithms to generate random samples of continuous k-DPPs. We achieve this by showing that the Gibbs sampler for a large family of continuous k- DPPs can be simulated efficiently when the spectrum is not concentrated on the top k eigenvalues.

5.1 Introduction

The exact definition of continuous DPPs was presented in section 3.4; as stated, the major difference with discrete DPPs is that these distributions are defined on a subset of $\mathbb{R}^d$ via a PSD integral operator instead of a PSD matrix. A simple and widely used example of such kernels is symmetric Gaussian also known as radial basis function (RBF) kernel. For a parameter $\sigma > 0$, a symmetric Gaussian kernel $g_\sigma : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is defined by $g_\sigma(x, y) = \exp(\frac{-\|x-y\|^2}{2\sigma^2})$.

Continuous DPPs naturally arise in several areas of Physics and Math; To name a few examples, eigenvalues of random matrices [88, 50], zero-set of Gaussian analytic functions

[106] are families of DPPs; also, see [75] for applications in statistics and [17] for connections to repulsive systems. However, their applications is not limited to just theoretical fields. Recently they have found several applications in machine learning as well; In [2], approximate samples from continuous k-DPPs are drawn to generate initial seeds for the k-means clustering algorithm. As they observed the diversity of DPP samples leads to better recovery of the underlying ground truth clusters. They have also been used for learning and tuning parameters of different learning models, e.g. learning parameters of generative mixture models [107, 74], tuning the hyper-parameters of a deep network [42].

The wide range of applications of continuous DPPs and k-DPPs motivates designing efficient learning and inference primitives for them. As explained in the previous chapter, in the discrete setting several efficient algorithms have been discovered for sampling [57, 83, 37, 6], marginalization [21], conditioning [72], and many other inference tasks. On the other hand, in the continuous domain, despite previous efforts [111, 76, 54, 56], there has been much less progress. In this work we study sampling algorithms for continuous k-DPPs. The sampling task for these distribution can be formally formulated as follows:

Sampling from continuous k-DPPs. Let π be a continuous k-DPP on domain $C \subseteq \mathbb{R}^d$ with kernel L. Recall that μ is a distribution on the subsets of size k of C, that we call k-points of C and a sample $x_1, \ldots, x_k \in C$ is generated from π if for any collection of non-overlapping sets $B_1, \ldots, B_k \subset C$, the probability that there exists a permutation $\sigma : \{1, \ldots, k\} \to \{1, \ldots, k\}$ that for any $1 \leq i \leq k$, $x_i \in B_{\sigma(i)}$ is proportional to

$$\int_{B_1} \cdots \int_{B_k} \det_L(z_1, \ldots, z_k) dz_1 \ldots dz_k.$$

Here we do not need to find the normalizer of this distribution, as we are only interested in (approximate) samples from the distribution. Note that, devising such algorithms in full generality is not well-defined, since such an algorithm would depend on how the

input kernel L is represented. Therefore, the main question is that, in what settings the sampling can be done efficiently. We propose an MCMC based approach by analyzing the so-called *Gibbs sampler chain* for continuous k-DPPs. Our main contribution is to show that this chain mixes in polynomial time in total variation distance, which implies simulating it yields a polynomial time algorithm for sampling from continuous k-DPPs. Next, we show that, given a "conditional sampling oracle" for a kernel L (definition 5.3), one can simulate the chain efficiently. Finally, we suggest that a simple rejection sampler can be used as the oracle to run the chain efficiently for several kernels of interest.

5.1.1 previous work

As explained in the previous section, the first type of sampling algorithms proposed for the discrete DPPs were spectral algorithms. The basic idea is introduced by [57] which given the eigen-decomposition of the kernel leads to a two-step sampling algorithm: Firstly, a set of eigen-vectors of the kernel is generated from a probability distribution driven from the eigenvalues. In the second step, a subset of points in the domain is sampled recursively based on selected eigen-vectors in the previous step. Although, a natural generalization of this scheme provides a theoretically correct and exact sampling method for continuous DPPs, there are several challenges to turn it into a practical algorithm:

1. A general continuous kernel does not have a finite eigen-decomposition representation. As suggested by [76] and [54], a heuristic is to find a finite rank approximation of the original kernel. [54] applies Nyström method, and random Fourier feature transform to find a low rank approximation of the kernel. However, to the best of our knowledge, there is no universal bound on the total variation distance of the approximated kernel and true underlying DPP kernel, because generally speaking, to project the DPP kernel onto a lower dimensional space, these methods minimize the error with respect to a matrix norm, rather than the DPP distribution.

2. Even given a proper low rank approximation of the kernel with small error, implementing the second phase of the algorithm is not tractable in general, as it requires computationally integrating certain functionals of the eigenvectors over a continuous space. To bypass this, [54] suggests an analytical approach which first computes a dual kernel by analytically integrating the functionals. Such a method can only be employed if the eigenvectors of the approximated kernel are well-understood and integrable.

Another type of algorithms which give fast, and practical sampling algorithms for discrete DPPs and k-DPPs are MCMC based methods. In the previous chapter, we show how a Metropolis-Hastings algorithm can be used to generate approximate samples from discrete k-DPPs in time $O(n)\mathsf{poly}(k)$, where n is the size of the domain, However, prior to this work, such an MCMC algorithm with a provable guarantee is not known for the continuous setting; in an attempt [57] provides empirical evidence that Gibbs sampling is an efficient algorithm to generate samples from continuous k-DPPs in many cases. However, they do not provide any rigorous justification.

It is also worth mentioning that [56] claims to devise an algorithm to generate exact samples for specific kernels (including Gaussian), yet a careful look at their method would reveal a major flaw in their argument [1].

5.1.2 Our Results

First, we formally define the lazy Gibbs sampler chain that we use for sampling from continuous k-DPPs. Let π be a k-DPP defined by a kernel $L : C \times C \to \mathbb{R}$ for $C \in \mathbb{R}^d$. Given a state $\{x_1, \ldots, x_k\}$, the Gibbs sampler $\mathcal{M}$ at each step evolves mas follows: With probability half stays at the current state. Otherwise, a point $x_i \in \{x_1, \ldots, x_k\}$ is chosen uniformly at random, and is replaced by $y \in C$ sampled from the conditional distribution

[1] The distribution that they consider as the conditional distribution of the k-DPP is in fact equivalent to our notion of conditional distribution of the kernel (see Definition 5.3)

whose PDF, f , is defined by

$$f(y) \propto \det_L(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_k).$$

Our main contribution is that the above-defined Gibbs sampler mixes rapidly in a time which is only a function of k, in both discrete and continuous settings.

We prove the following bound on the mixing time of the Gibbs sampler for continuous k-DPPs and, its analogue for discrete k-DPPs.

Theorem 5.1. *Let $\mathcal{M}$ be the Gibbs sampler for a k-DPP π. If we run the chain starting from an arbitrary distribution μ_0, for any $\epsilon > 0$ we have*

$$\tau_{\mu_0}(\epsilon) \leq O(k^4) \cdot \log\left(\frac{\mathrm{var}_\pi(\frac{f_{\mu_0}}{f_\pi})}{\epsilon}\right).$$

In the above theorem, f_π and f_{μ_0} refer to the probability density functions for π and μ_0, respectively. Moreover, $\tau_{\mu_0}(\epsilon)$ denotes the mixing time for the chain started from μ_0. To prove this theorem, we analyze the *conductance* of the Gibbs samplers k-DPPs (see theorems 5.6 and 5.11), and apply the well-known connection between the conductance and mixing time.

Applications for Discrete k-DPPs. In this case, to find a proper starting state of the chain, we can use the greedy algorithm for determinant maximization (algorithm 4.1) which returns a state, a subset of C of size k S, where $\pi(S) \geq \frac{1}{k!}$; starting from such state S, the chain generates ϵ-approximate samples after $\tilde{O}(k^5)$ steps. Moreover, for a k-DPP over N elements, one can note that to simulate one step of the chain, it is enough to compute the determinant of at most N $k \times k$ submatrices. Therefore, using the Gibbs sampler, approximate samples from a k-DPP can be generated in time $O(N) \cdot \mathsf{poly}(k)$. As a sequential algorithm, this algorithm does not improve the running time of the Metropolis-Hastings algorithm presented in chapter 4. However, since the mixing time is independent of N, it can lead to sublinear time sampling algorithms in distributed models of computation. The following corollary is an immediate naive consequence of

this fact.

Corollary 5.2. *Given access to $N\delta$ processors for some $\delta > 0$, an approximate sample of a k-DPP defined on domain of size N can be generated in time $O(N^{1-\delta}) \cdot \mathsf{poly}(k)$.*

On the other hand, for continuous k-DPPs, to turn the above result into an efficient algorithm, finding a good starting distribution μ_0 which makes the log variance term in the bound of theorem 5.1 polynomially small is more elusive. We also need to have an algorithm to simulate the Gibbs sampler. To do both of these, we require the DPP kernel to be presented to us by a conditional sampling oracle, defined as follows.

Definition 5.3. For a kernel $L : C \times C \to \mathbb{R}$ and a subset $S \subset C$, we define (S, L)-conditional distribution to be a distribution on C defined by the PDF function $f : C \to \mathbb{R}_+$ given by

$$f(x) \propto \det_L(S \cup x\}),$$

and zero if $x \in S$. We denote this distribution by $\mathrm{CD}_L(S)$. We say an algorithm is an $\mathrm{CD}_L(i)$ oracle for an integer i, if given any $S \subset C$ ($|S| = i$), it returns a sample from the $\mathrm{CD}_L(S)$.

It is straight-forward to see that taking a step of the Gibbs sampler of the k-DPP from the state $x_1, \ldots, x_k$ defined by L is equivalent to removing a point x_i, for some $1 \leq i \leq k$, and generating a sample from $\mathrm{CD}_L(\{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k\})$. Therefore simulating the chain can be done by a $\mathrm{CD}_L(k-1)$ oracle call. We also show these oracles are enough to find a proper starting distribution to get an algorithm with the following guarantee. We prove the following.

Theorem 5.4. *Let π be a k-DPP defined by a kernel $L : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. Given $CD_L(i)$ oracles for all $0 \leq i \leq k-1$, we can generate ϵ-approximate samples from π with*

$$O(k^5 \log \frac{k}{\epsilon}).$$

oracle calls.

Therefore, the task of sampling from a continuous k-DPP boils down to sampling from conditional 1-DPPs ($\mathrm{CD}_L(.)$ distributions), which seems a simpler problem.

Applications for Continuous k-DPPs. To construct the conditional sampling oracles, we use a simple rejection sampler similar to the one suggested at [76], with uniform distribution on the domain as the proposal distribution. Analyzing the rejection sampler and combining that with theorem 5.4, we get the following.

Theorem 5.5. *Let L be a kernel on a bounded closed domain C, and suppose that we have access to an oracle which can generate uniform samples from C. For any integer k and any $\epsilon > 0$, an ϵ-approximate sample from the k-DPP defined by L can be generated by*

$$O(k^5 \log \frac{1}{\epsilon}) \frac{M \cdot \mathrm{VOL}(C)}{\sum_{i=k}^{\infty} \lambda_i} \tag{5.1}$$

oracle calls in expectation where $\lambda_0 \geq \lambda_1 \geq \cdots$ are eigen-values of L and $M = \sup_x L(x,x)$.

For some of the widely used kernels such as a Gaussian kernel, the $L(x,x)$ is a constant for all x and so $\mathrm{tr}(L) = \int_C L(x,x) \propto \mathrm{VOL}(C)$ and the numerator in eq. (5.1) becomes proportional to $\mathrm{tr}(C)$. Therefore in this setting, we get an efficient algorithm for sampling from k-DPPs with "moderately decaying" spectrum. We further analyze the running time for Gaussian kernels defined on a sphere. The details can be found in section 5.5.2.

5.1.3 Techniques

Our first contribution is to analyze the Gibbs sampler chain in the discrete setting. We prove for a k-DPP defined on N points, the spectral gap of the Gibbs sampler chain is a polynomial in $1/k$ and *independent* of N. So, up to logarithmic factors in N, the chain mixes in time polynomial in k. This result on its own could be of interest in designing distributed algorithms for sampling from discrete k-DPPs. This is because given access to m processors, one can generate the next step of the Gibbs sampler in time $O(N/m)$.

Secondly, we lift the above proof to the continuous setting using a natural discretization

of the underlying space. To prove the mixing time, we need to make sure that the logarithm of the variance of the starting distribution with respect to the stationary distribution of the chain, i.e., the k-DPP, is polynomially small in k and the dimension of the ambient space. We use a simple randomized greedy algorithm for this task: We start from the empty set; assuming we have chosen $x_1, \ldots, x_i$ we sample x_{i+1} from $CD_L(\{x_1, \ldots, x_i\})$, where as usual L is the underlying kernel. We show that the distribution governing the state output by this algorithm is our desired starting distribution.

Lastly, we use our main theorem to generate samples from a k-DPP defined on a spherical Gaussian kernel on $\mathbb{S}^{d-1}$. To run the above algorithm we need to construct the $CD_L(i)$ oracles for all $0 \leq i \leq k-1$. Given the point $\{x_1, \ldots, x_i\}$, we use the classical rejection sampling algorithm to choose x_{i+1}; namely, we generate a uniformly random point on the unit sphere and we accept it with probability $\frac{\det_L(x_1,\ldots,x_{i+1})}{\det_L(x_1,\ldots,x_i)}$. We use the distribution of the eigenvalues of the spherical Gaussian kernel [90] to bound the expected number of proposals in the rejection sampler.

5.2 Notaions

Let $\mathbb{R}^d$ denote the d-dimensional euclidean space. Whenever, we consider $C \subset \mathbb{R}^d$ as measurable space, our measure is the standard Lebesgue measure. The $\mathrm{VOL}(C)$ denotes the d-dimensional volume of C with respect to the standard measure. Throughout the chapter, we only consider k-DPPs which are defined by continuous Hilbert-Schmidt kernels which satisfy the properties stated in section 3.4.1. If π is a probability distribution, we use f_π to refer to the corresponding probability density function (PDF). We use small letters to refer to elements of C and small bold letters to refer to its subsets, e.g. $\boldsymbol{x} = \{x_1, \ldots, x_k\}$ indicates a state of the Gibbs sampler for a k-DPP. For $y \in \mathbb{R}^d$ and a finite subset $S \in \mathbb{R}^d$, we may use $S + y$ to indicate $S \cup \{y\}$ and $S - y$ to indicate $S \setminus \{y\}$. Moreover, for $S = \{x_1, \ldots, x_m\}$ we use $\det_L(x_1, \ldots, x_m)$ or $\det_L(S)$ to refer to determinant of the $m \times m$ submatrix where the ij_{th} entry is $L(x_i, x_j)$. Whenever, the kernel is clear

from the context, we may drop the subscript. For two expression A and B, we write $A \lesssim B$ to denote $A \leq O(B)$.

5.3 *Gibbs Sampling for Discrete k-DPP*

In this section we analyze the conductance of the Gibbs sampler for a discrete k-DPP, and show that it is bounded by $\Omega\left(\frac{1}{k^2}\right)$. Recall that the conductance of a time reversible chain $\mathcal{M} = (\Omega, P, \pi)$ is defined by

$$\Phi(\mathcal{M}) = \min_{S \subset \Omega : \pi(S) \leq \frac{1}{2}} \frac{Q(S, \overline{S})}{\pi(S)},$$

where for $x, y \in \Omega$, $Q(y, x) = Q(x, y) = \pi(x)P(x, y)$ and $Q(S, \overline{S}) = \sum_{\substack{x \in S \\ y \notin S}} Q(x, y)$. We prove the following.

Theorem 5.6. *Let $\mathcal{M}$ be the Gibbs sampler chain for an arbitrary discrete k-DPP, then for a constant $C < 20$ we have*

$$\phi(\mathcal{M}) \geq \frac{1}{Ck^2}$$

In the rest of this section, we fix $\mathcal{M} = (\Omega, P, \pi)$ to be the Gibbs-sampler chain on a k-DPP defined on a set of N elements.

Before discussing the details of the proof let us first fix a notation and recall fundamental properties of k-DPPs. For any element $1 \leq i \leq N$, define $\Omega_i, \Omega_{\bar{i}}$ be the set of all states in Ω that contain, do not contain i, respectively. Also define π_i and $\pi_{\bar{i}}$ to be the corresponding conditional distribution (stationary distribution of the restricted chains). i.e. for

$$\pi_i := \{\pi |\ i \text{ is chosen }\},\ \text{i.e. } \pi_i(\boldsymbol{x}) = \frac{\pi(\boldsymbol{x})}{\pi(\Omega_i)}, \forall \boldsymbol{x} \in \Omega_i$$

$$\pi_{\bar{i}} := \{\pi |\ i \text{ is not chosen }\},\ \text{i.e. } \pi_{\bar{i}}(\boldsymbol{x}) = \frac{\pi(\boldsymbol{x})}{\pi(\Omega_{\bar{i}})}, \forall \boldsymbol{x} \in \Omega_{\bar{i}}.$$

As explained in section 3.3.2, $\pi_i, \pi_{\bar{i}}$ can be identified with a $(k-1)$-DPP, k-DPP supported on $\Omega_i, \Omega_{\bar{i}}$, respectively. We define $\mathcal{M}_i = (\Omega_i, P_i, \pi_i), \mathcal{M}_{\bar{i}} = (\Omega_{\bar{i}}, P_{\bar{i}}, \pi_{\bar{i}})$ to be the *restricted* Gibbs samplers. So, it is straightforward to see that for any $\boldsymbol{x}, \boldsymbol{y} \in \Omega_i$ we get $P_i(\boldsymbol{x}, \boldsymbol{y}) =$

$\frac{k}{k-1} \cdot P(\boldsymbol{x}, \boldsymbol{y})$. and consequently for Q_i defined as Q for $\mathcal{M}_i$, we get

$$Q_i(\boldsymbol{x}, \boldsymbol{y}) = \frac{Q(\boldsymbol{x}, \boldsymbol{y})}{\pi(\Omega_i)}. \tag{5.2}$$

Unlike P_i, $P_{\bar{i}}$ is not obtained from scaling a restriction of P. In particular, Let $\boldsymbol{x}, \boldsymbol{y} \in \Omega_{\bar{i}}$ so that $P_{\bar{i}}(\boldsymbol{x}, \boldsymbol{y}) > 0$ (which implies $|\boldsymbol{x} \cap \boldsymbol{y}| = k - 1$). Then, setting $I = \boldsymbol{x} \cap \boldsymbol{y}$ and with a bit abuse of notation $\pi(I) = \sum_{j \in [n] \setminus I} \pi(I + j)$, i.e. $\pi(I) = \mathbb{P}_{\boldsymbol{z} \sim \pi}[I \subset \boldsymbol{z}]$, we have

$$P_{\bar{i}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{k} \cdot \frac{\pi(\boldsymbol{y})}{\pi(I) - \pi(i + I)} \tag{5.3}$$

whereas $P(\boldsymbol{x}, \boldsymbol{y}) = \frac{\pi(\boldsymbol{y})}{k \cdot \pi(I)}$. For any $\boldsymbol{x} \in \Omega_i$, define $N_{\bar{i}}(\boldsymbol{x})$ be the set of its neighbours in $\Omega_{\bar{i}}$, i.e.

$$N_{\bar{i}}(\boldsymbol{x}) = \{\boldsymbol{y} \in \Omega_{\bar{i}} | P(\boldsymbol{x}, \boldsymbol{y}) > 0\}.$$

We use the following lemma to relate $Q_{\bar{i}}$ to Q.

Lemma 5.7. *Let $A \subset \Omega_{\bar{i}}$ be an arbitrary subset. For a state $\boldsymbol{x} \in \Omega_i$, consider the following partitioning of $N_{\bar{i}}(\boldsymbol{x})$: $N_A = N_{\bar{i}}(\boldsymbol{x}) \cap A$ and $N_{\overline{A}} = N_{\bar{i}}(\boldsymbol{x}) \cap (\Omega_{\bar{i}} \setminus A)$. Then we have*

$$Q(\boldsymbol{x}, N_A) + Q(N_A, N_{\overline{A}}) \geq \pi(\Omega_{\bar{i}}) \cdot Q_{\bar{i}}(N_A, N_{\overline{A}}). \tag{5.4}$$

Proof. Note that $\boldsymbol{x} \cup N_A \cup N_{\overline{A}}$ is the set of all states containing elements in $\boldsymbol{x} - i$. So by definition of Q and $Q_{\bar{i}}$, we have

$$Q(\boldsymbol{x}, N_A) + Q(N_A, N_{\overline{A}}) = \frac{1}{k} \cdot \frac{\pi(\boldsymbol{x})\pi(N_A)}{\pi(\boldsymbol{x}) + \pi(N_A) + \pi(N_{\overline{A}})} + \frac{1}{k} \cdot \frac{\pi(N_{\overline{A}})\pi(N_A)}{\pi(\boldsymbol{x}) + \pi(N_A) + \pi(N_{\overline{A}})} \tag{5.5}$$

$$= \frac{\pi(N_A)}{k} \cdot \frac{\pi(\boldsymbol{x}) + \pi(N_{\overline{A}})}{\pi(\boldsymbol{x}) + \pi(N_A) + \pi(N_{\overline{A}})} \geq \frac{\pi(N_A)}{k} \cdot \frac{\pi(N_{\overline{A}})}{\pi(N_A) + \pi(N_{\overline{A}})} \tag{5.6}$$

$$= \pi(\Omega_{\bar{i}}) \cdot Q_{\bar{i}}(N_A, N_{\overline{A}}) \tag{5.7}$$

where the inequality follows simply because $\pi(N_A) \geq 0$. □

High level idea of the proof of Theorem 5.6. We follow a proof strategy similar to [89], which obtains analogue of our result in an unweighted setting and for the Metropolis-

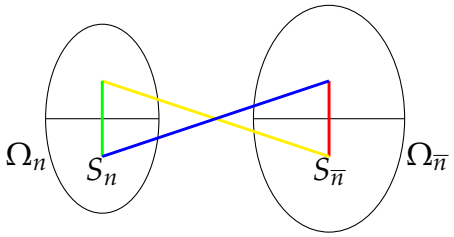


Figure 5.1: A schematic view of the restriction chains.
yellow, red, blue, and green edges correspond to
$Q(S_n, \Omega_n \setminus S_n), Q(S_{\overline{n}}, \Omega_{\overline{n}} \setminus S_{\overline{n}}), Q(S_n, \Omega_{\overline{n}} \setminus \Omega_{\overline{n}} \setminus S_{\overline{n}})$, and $Q(S_{\overline{n}}, \Omega_{\overline{n}} \setminus S_{\overline{n}})$, respectively

Hastings samplers. We use an inductive argument to prove the theorem. We need to prove $Q(S, \overline{S}) \geq \frac{\pi(S)}{Ck^2}$ for a subset $S \in \Omega$ with $\pi(S) \leq \frac{1}{2}$. Letting $S_n = S \cap \Omega_n$ and $S_{\overline{n}} = S \cap \Omega_{\overline{n}}$, we have

$$Q(S, \overline{S}) = Q(S_n, \Omega_n \setminus S_n) + Q(S_{\overline{n}}, \Omega_{\overline{n}} \setminus S_{\overline{n}}) + Q(S_n, \Omega_{\overline{n}} \setminus S_{\overline{n}}) + Q(S_{\overline{n}}, \Omega_n \setminus S_n). \quad (5.8)$$

We carry out the induction step by lowerbounding the RHS of the above term by term. In order to bound $Q(S_n, \Omega_n \setminus S_n)$ we use induction hypothesis on $\mathcal{M}_n$. To bound $Q(S_{\overline{n}}, \Omega_n \setminus S_{\overline{n}})$, we combine the induction hypothesis on $\mathcal{M}_{\overline{n}}$ with Lemma 5.7. It remains to bound the other two terms which correspond to the contribution of the edge across $(\Omega_n, \Omega_{\overline{n}})$. To do that, we crucially use negative association of π. In particular, we use lemma 4.10. For any set $A \in \Omega_n$, let $N_{\overline{n}}(A) = \{\boldsymbol{y} \in \Omega_{\overline{n}} : \exists \boldsymbol{x} \in A,\ P(\boldsymbol{x}, \boldsymbol{y}) > 0\}$ denote the set of neighbors of A in $\Omega_{\overline{n}}$. The following restates that lemma.

Lemma 5.8. *For any subset $A \subseteq \Omega_n$,*

$$\pi_{\overline{n}}(N_{\overline{n}}(A)) \geq \pi_n(A).$$

The lemma lower bounds the vertex expansion of S_n in Ω_n and similarly vertex expansion of $S_{\overline{n}}$ in Ω_n. Later we show how to use it to bound the edge expansion which is our quantity of interest.

Proof of Theorem 5.6. We induct on $k+n$. So, assume, the conductance of the Gibbs sampler for any $(k-1)$-DPP over $n-1$ elements is at most $\frac{1}{C(k-1)^2}$ and the conductance is at most $\frac{1}{Ck^2}$ for any k-DPP over any $n-1$ elements.

Fix a set $S \subset \Omega$ where $\pi(S) \leq \frac{1}{2}$. We need to show $Q(S, \overline{S}) \geq \frac{\pi(S)}{Ck^2}$. First, consider a simple case where $\pi_n(S) \leq \frac{1}{2}$ and $\pi_{\overline{n}}(S) \leq \frac{1}{2}$. By induction hypothesis we have $Q_n(S_n, \Omega_n \setminus S_n) \geq \frac{\pi_n(S_n)}{c(k-1)^2}$. Moreover, by adding up (5.2) for the edges across the cut $(S_n, \Omega_n \setminus S_n)$, we get $Q(S_n, \Omega_n \setminus S_n) = \frac{(k-1)\pi(\Omega_n)}{k} \cdot Q_n(S_n, \Omega_n \setminus S_n)$. So combining them we have

$$Q(S_n, \Omega_n \setminus S_n) \geq \frac{\pi(S_n)}{Ck^2}. \tag{5.9}$$

Now, we use induction on $\mathcal{M}_{\overline{n}}$ along with Lemma 5.7. The induction hypothesis implies

$$Q_{\overline{n}}(S_{\overline{n}}, \Omega_{\overline{n}} \setminus S_{\overline{n}}) \geq \frac{\pi_{\overline{n}}(S_{\overline{n}})}{ck^2} = \frac{\pi(S_{\overline{n}})}{\pi(\Omega_{\overline{n}}) \cdot ck^2}$$

So to prove the theorem in this case, it is enough to show the following and add it up with (5.9).

$$Q(S_{\overline{n}}, \Omega_{\overline{n}} \setminus S_{\overline{n}}) + Q(S_{\overline{n}}, \Omega_n \setminus S_n) + Q(S_n, \Omega_{\overline{n}} \setminus S_{\overline{n}}) \geq \pi(\Omega_{\overline{n}}) \cdot Q_{\overline{n}}(S_{\overline{n}}, \Omega_{\overline{n}} \setminus S_{\overline{n}}). \tag{5.10}$$

To see that, it is enough to apply Lemma 5.7 and add up (5.4) for all $\boldsymbol{x} \in \Omega_n$, where subset $A \subset \Omega_{\overline{n}}$ in the lemma is determined as follows: if $\boldsymbol{x} \in S_n$ then set $A = S_{\overline{n}}$, otherwise set $A = \Omega_{\overline{n}} \setminus S_n$. Note that, doing that the RHS of the result will be exactly $\pi(\Omega_{\overline{n}}) \cdot Q_{\overline{n}}(S_{\overline{n}}, \Omega_{\overline{n}} \setminus \Omega_{\overline{n}})$, because any edge $\boldsymbol{yz}$ of that will only show up in (5.4) by having $\boldsymbol{x} = \boldsymbol{y} \cap \boldsymbol{z} + n$.

So we focus on the case $\max\{\pi_n(S_n), \pi_{\overline{n}}(S_{\overline{n}})\} > \frac{1}{2}$. Since $\pi(S) \leq \frac{1}{2}$, we have $\min\{\pi_n(S_n), \pi_{\overline{n}}(S_{\overline{n}})\} \leq \frac{1}{2}$. So, without loss of generality, perhaps by considering $\overline{S}$ instead of S, we may assume $\pi_n(S_n) > \frac{1}{2}$ and $\pi_{\overline{n}}(S) \leq \frac{1}{2}$. Our goal is to prove

$$Q(S, \overline{S}) \geq \frac{1}{Ck^2} \cdot \min\{1 - \pi(S), \pi(S)\} \tag{5.11}$$

For every $\boldsymbol{x} \in \Omega_n$, let $N_{\overline{n},S}(\boldsymbol{x}) := N_{\overline{n}}(\boldsymbol{x}) \cap S_{\overline{n}}$, and $N_{\overline{n},\overline{S}}(\boldsymbol{x}) := N_{\overline{n}}(\boldsymbol{x}) \cap (\Omega_{\overline{n}} \setminus S_{\overline{n}})$ be a partitioning of $N_{\overline{n}}(\boldsymbol{x})$, so for every subset $T \in N_{\overline{n}}(\boldsymbol{x})$ we have

$$Q(\boldsymbol{x}, T) = \frac{1}{2k} \cdot \frac{\pi(\boldsymbol{x})\pi(T)}{\pi(\boldsymbol{x}) + \pi(N_{\overline{n},S}(\boldsymbol{x})) + \pi(N_{\overline{n},\overline{S}}(\boldsymbol{x}))} \tag{5.12}$$

Now, define $S_{\text{leave}} \subset S_n$ to be

$$S_{\text{leave}} = \{\boldsymbol{x} \in S_n : \ \pi(\boldsymbol{x}) + \pi(N_{\overline{n},S}(\boldsymbol{x})) < \pi(N_{\overline{n},\overline{S}}(\boldsymbol{x}))\},$$

in other words, $S_{\text{leave}} \in S_n$ is the subset of states so that, if the chain takes one step from S_{leave} by removing and resampling element n, then with probability at least $\frac{1}{2}$ it leaves S and enters $N_{\overline{n},\overline{S}}(\boldsymbol{x})$. We also let $S_{\text{stay}} = S_n \setminus S_{\text{leave}}$. On the other hand, starting from S_{stay} and by resampling n, the chain with probability at least half stays in S. It is straight-forward to see

$$Q(S_{\text{leave}}, \Omega_{\overline{n}} \setminus S_{\overline{n}}) \geq \frac{\pi(S_{\text{leave}})}{4k} \tag{5.13}$$

To see that, note that definition of S_{leave} and setting $T = \Omega_{\overline{n}} \setminus S_{\overline{n}}$ in (5.12) implies that for any $\boldsymbol{x} \in S_{\text{leave}}$, we have $Q(\boldsymbol{x}, \Omega_{\overline{n}} \setminus S_{\overline{n}}) \geq \frac{\pi(\boldsymbol{x})}{4k}$. To get (5.13), it suffices to sum up this over all states of S_{leave}. The bound (5.13) shows that $Q(S_{\text{leave}}, \overline{S}) \gg \frac{\pi(S_{\text{leave}})}{k^2}$. So roughly speaking, to prove the theorem, it suffices to show $\phi(S_{\text{stay}}) \cup S_{\overline{n}}) \geq \frac{1}{Ck^2}$. consider two cases: if $\pi_n(S_{\text{stay}}) \lessapprox \frac{1}{2}$, we essentially use the same argument as in the case $\pi_n(S_n), \pi_{\overline{n}}(S_{\overline{n}}) \leq \frac{1}{2}$. Otherwise we combine the induction with Lemma 5.8 to bound the expansion.

- **Case 1:** $\pi_n(S_{\textbf{stay}}) \leq \frac{1}{2} + \frac{1}{4k}$**.** We show $Q(S, \overline{S}) \geq \frac{\pi(S)}{Ck^2}$. To do that, we use the induction hypothesis on $\mathcal{M}_n$, and the following claim which is the stronger version of (5.10).

  Claim 5.9.

  $$Q(S_{\overline{n}}, \overline{S}) + Q(S_n, \Omega_{\overline{n}} \setminus S_{\overline{n}}) - \frac{1}{2} Q(S_{leave}, \Omega_{\overline{n}} \setminus S_{\overline{n}}) \geq \pi(\Omega_{\overline{n}}) \cdot Q_{\overline{n}}(S_{\overline{n}}, \Omega_{\overline{n}} \setminus S_{\overline{n}}) \tag{5.14}$$

  Proof. The claim is implied by combining the summation of (5.15),(5.16), and (5.17) over $\Omega_n \setminus S_n$, S_{stay} and S_{leave}, respectively. Let $\boldsymbol{x} \in \Omega_n \setminus S_n$. Then by applying Lemma 5.7 for $\boldsymbol{x}$ and $A = S_{\overline{n}}$, we get

  $$Q(N_{\overline{n},S}(\boldsymbol{x}), \{\boldsymbol{x}\} \cup N_{\overline{n},\overline{S}}(\boldsymbol{x})) \geq \pi(\Omega_{\overline{n}}) \cdot Q_{\overline{n}}(N_{\overline{n},S}(\boldsymbol{x}), N_{\overline{n},\overline{S}}(\boldsymbol{x})) \tag{5.15}$$

  Similarly if $\boldsymbol{x} \in S_n$, by applying Lemma 5.7 for $\boldsymbol{x}$ and $A = \Omega_{\overline{n}} \setminus S_{\overline{n}}$, we have

  $$Q(\boldsymbol{x} \cup N_{\overline{n},S}(\boldsymbol{x}, N_{\overline{n},\overline{S}}(\boldsymbol{x})) \geq \pi(\Omega_{\overline{n}}) \cdot Q_{\overline{n}}(N_{\overline{n},S}(\boldsymbol{x}), N_{\overline{n},\overline{S}}(\boldsymbol{x})). \tag{5.16}$$

Finally, for $\boldsymbol{x} \in S_{\text{leave}}$, we can show

$$Q(N_{\overline{n},S}(\boldsymbol{x}), N_{\overline{n},\overline{S}}(\boldsymbol{x})) + \frac{1}{2}Q(\boldsymbol{x}, N_{\overline{n},\overline{S}}(\boldsymbol{x})) \geq \pi(\Omega_{\overline{n}}) \cdot Q_{\overline{n}}(N_{\overline{n},S}(\boldsymbol{x}), N_{\overline{n},\overline{S}}(\boldsymbol{x})). \tag{5.17}$$

To see that, first note that the LHS is equal to

$$\frac{\pi(N_{\overline{n},\overline{S}}(\boldsymbol{x}))}{2k \cdot (\pi(\boldsymbol{x}) + \pi(N_{\overline{n},S}(\boldsymbol{x})) + \pi(N_{\overline{n},\overline{S}}(\boldsymbol{x})))} \cdot \left(\pi(N_{\overline{n},S}(\boldsymbol{x})) + \frac{\pi(\boldsymbol{x})}{2}\right)$$

which since $\pi(\boldsymbol{x}) + \pi(N_{\overline{n},S}(\boldsymbol{x})) < \pi(N_{\overline{n},\overline{S}}(\boldsymbol{x}))$ for $\boldsymbol{x} \in S_{\text{leave}}$, we get the lower-bound of

$$\frac{1}{2k} \cdot \frac{\pi(N_{\overline{n},S}(\boldsymbol{x}))\pi(N_{\overline{n},\overline{S}}(\boldsymbol{x}))}{\pi(N_{\overline{n},S}(\boldsymbol{x})) + \pi(N_{\overline{n},\overline{S}}(\boldsymbol{x}))}\pi(\Omega_{\overline{n}}) = \pi(\Omega_{\overline{n}}) \cdot Q_{\overline{n}}(N_{\overline{n},S}(\boldsymbol{x}), N_{\overline{n},\overline{S}}(\boldsymbol{x}))$$

for the LHS. □

In particular, we use the above claim to get

$$\begin{aligned}
Q(S,\overline{S}) &= Q(S_n, \Omega_n \setminus S_n) + Q(S_{\overline{n}}, \Omega_{\overline{n}} \setminus S_{\overline{n}}) + Q(S_n, \Omega_{\overline{n}} \setminus S_{\overline{n}}) + Q(S_{\overline{n}}, \Omega_n \setminus S_n) \\
&\geq Q(S_n, \Omega_n \setminus S_n) + \frac{1}{2}Q(S_{\text{leave}} + \Omega_{\overline{n}} \setminus S_{\overline{n}}) + \pi(\Omega_{\overline{n}})Q_{\overline{n}}(S_{\overline{n}}, \overline{S_{\overline{n}}}) \quad \text{By Claim 5.9} \\
&\geq \frac{\pi(\Omega_n) - \pi(S_n)}{Ck(k-1)} + \frac{1}{2}Q(S_{\text{leave}}, \Omega_{\overline{n}} \setminus S_{\overline{n}}) + \frac{\pi(S_{\overline{n}})}{Ck^2} \quad \text{induction Hyp. on } \mathcal{M}_n \text{ and } \mathcal{M}_{\overline{n}} \\
&\geq \frac{\pi(\Omega_n) - \pi(S_{\text{leave}}) - \pi(S_{\text{stay}})}{Ck(k-1)} + \frac{\pi(S_{\text{leave}})}{8k} + \frac{\pi(S_{\overline{n}})}{Ck^2}
\end{aligned} \tag{5.18}$$

where the last inequality follows by (5.13) and the fact that $S_n = S_{\text{leave}} \cup S_{\text{stay}}$. To finish the proof, we need to show the RHS of the above is at least $\frac{\pi(S)}{Ck^2}$. To see that note that since $\frac{\pi(S_{\text{leave}})}{8k} \geq \pi(S_{\text{leave}}) \cdot \left(\frac{1}{Ck^2} + \frac{1}{Ck(k-1)}\right)$ for sufficiently large k, it suffices to show $\frac{\pi(\Omega_n) - \pi(S_{\text{stay}})}{Ck(k-1)} \geq \frac{\pi(S_{\text{stay}})}{Ck^2}$, which can be directly verified for $\pi_n(S_{\text{stay}}) \leq \frac{1}{2} + \frac{1}{4k}$.

- **Case 2:** $\pi_n(S_{\textbf{stay}}) > \frac{1}{2} + \frac{1}{4k}$**.** We prove

$$Q(S,\bar{S}) \geq \frac{1 - \pi(S)}{Ck^2}.$$

Lemma 5.8 states that the vertex expansion of S_{stay} is proportional to $\pi_n(S_{\text{stay}}) - \pi_{\overline{n}}(S_{\overline{n}})$(which is positive in this case by the assumption). We use it to bound $Q(S,\overline{S})$ by relating vertex expansion of S_{stay} to $Q(S,\overline{S})$. In particular, we show the

following claim.

Claim 5.10.

$$Q(S_{stay}, \Omega_{\overline{n}} \setminus S_{\overline{n}}) + Q(S_{\overline{n}}, \Omega_{\overline{n}} \setminus S_{\overline{n}}) \geq \frac{\pi(\Omega_{\overline{n}})}{2k} \cdot \left(\pi_n(S_{stay}) - \pi_{\overline{n}}(S_{\overline{n}})\right)$$

Proof. Note that for any $\boldsymbol{x} \in S_{\text{stay}}$, since $\pi(N_{\overline{n},\overline{S}}(\boldsymbol{x})) \leq \pi(\boldsymbol{x}) + \pi(N_{\overline{n},S}(\boldsymbol{x}))$, we have

$$Q(\boldsymbol{x}, N_{\overline{n},\overline{S}}(\boldsymbol{x})) + Q(N_{\overline{n},S}(\boldsymbol{x}), N_{\overline{n},\overline{S}}(\boldsymbol{x})) = \frac{1}{2k} \cdot \frac{\pi(N_{\overline{n},\overline{S}}(\boldsymbol{x})) \cdot [\pi(\boldsymbol{x}) + \pi(N_{\overline{n},S}(\boldsymbol{x}))]}{\pi(\boldsymbol{x}) + \pi(N_{\overline{n},S}(\boldsymbol{x})) + \pi(N_{\overline{n},\overline{S}}(\boldsymbol{x}))} \geq \frac{1}{2k} \cdot \frac{\pi(N_{\overline{n},\overline{S}}(\boldsymbol{x}))}{2},$$

To complete the proof, it is enough to sum up the above over S_{stay} to get the following

$$\begin{aligned} Q(S_{\text{stay}}, \Omega_{\overline{n}} \setminus S_{\overline{n}}) + Q(S_{\overline{n}}, \Omega_{\overline{n}} \setminus S_{\overline{n}}) &\geq \sum_{x \in S_{\text{stay}}} \frac{\pi(N_{\overline{n},\overline{S}}(\boldsymbol{x}))}{4k} \geq \pi\left(\bigcup_{\boldsymbol{x} \in S_{\text{stay}}} N_{\overline{n},\overline{S}}(\boldsymbol{x})\right). \\ &\geq \pi(N_{\overline{n}}(S_{\text{stay}})) - \pi(S_{\overline{n}}) \\ &\geq \pi(\Omega_{\overline{n}}) \cdot \left(\pi_n(S_{\text{stay}}) - \pi_n(S_{\overline{n}})\right) \quad \text{By Lemma 5.8} \end{aligned}$$

□

Claim 5.10 and (5.18) implies $Q(S, \overline{S}) \geq \max\{L_1, L_2\}$ defined as above

$$L_1 := \frac{\pi(S_1)}{8k} + \frac{\pi(\Omega_n) - \pi(S_{\text{leave}}) - \pi(S_{\text{stay}})}{Ck(k-1)} + \frac{\pi(S_{\overline{n}})}{Ck^2} \qquad \text{By (5.18)}$$

$$L_2 := \frac{\pi(\Omega_{\overline{n}})}{4k} \cdot \left(\pi_n(S_{\text{stay}}) - \pi_{\overline{n}}(S_{\overline{n}})\right) \qquad \text{By Claim 5.10.}$$

So we need to prove $\max\{L_1, L_2\} \geq \frac{1-\pi(S)}{Ck^2}$. To prove that, we show that $L_1 + \frac{L_2}{k-1} \geq (1 + \frac{1}{k-1}) \cdot \frac{1-\pi(S)}{Ck^2}$. Replacing values of L_1 and L_2 in the above and simplifying the resulting inequality, we need to show

$$\frac{\pi(S_{\text{leave}})}{8k} + \frac{\pi(S_{\overline{n}})}{Ck^2} + \frac{\pi(\Omega_{\overline{n}})}{4k(k-1)} \cdot \left(\pi_n(S_{\text{stay}}) - \pi_{\overline{n}}(S_{\overline{n}})\right) \geq \frac{\pi(\Omega_{\overline{n}}) - \pi(S_{\overline{n}})}{Ck(k-1)}.$$

Ignoring the $\frac{\pi(S_1)}{8k}$ term and rearranging the other terms, it is enough to show

$$\frac{\pi(\Omega_{\overline{n}})}{4k(k-1)} \cdot (\pi_{\overline{n}}(S_{\text{stay}}) - \pi_{\overline{n}}(S_{\overline{n}})) \geq \frac{\pi(\Omega_{\overline{n}})}{Ck(k-1)} \cdot (1 - \frac{2k-1}{k} \cdot \pi_{\overline{n}}(S_{\overline{n}})).$$

The above can be verified for $C > 16$, by noting that by assumption $\pi_n(S_{\text{stay}}) \geq \frac{1}{2} + \frac{1}{4k}$ and $\pi_{\overline{n}}(S_{\overline{n}}) \leq \frac{1}{2}$.

□

5.4 *Gibbs Sampling for Continuous k-DPP*

In this section we analyze the mixing time of Gibbs samplers for continuous k-DPPs. Let $\mathcal{M}$ be the Gibbs sampler for a k-DPP defined by a continuous kernel L. In section 5.4.1, we show $\phi(M) \gtrsim \frac{1}{k^2}$. Therefore, Gibbs sampling is an efficient method to generate samples from a continuous k-DPP provided that: We have access to an $\mathrm{CD}_L(k-1)$ oracle to simulate the chain, and we can find a *proper* starting distribution. In subsection 5.4.2, we show access to conditional oracles sampling is also enough to find the proper starting distributions.

As alluded to before, throughout the section $L : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a continuous kernel which satisfies the Mercer's condition and also $\int\int |L(x,y)|^2 dxdy < \infty$ which also implies the partition function $Z = \int \cdots \int \det_L(x_1, \ldots, x_k) dx_k \ldots dx_1 < \infty$.

5.4.1 *Conductance of $\mathcal{M}$*

Theorem 5.11. *Let $\mathcal{M}$ be the Gibbs sampler for a k-DPP defined by kernel L, then*

$$\phi(\mathcal{M}) \gtrsim \frac{1}{k^2}. \tag{5.19}$$

Proof. Recall that by Theorem 5.6 the conductance of a Gibbs sampler for any discrete k-DPP is at least $\Omega(\frac{1}{k^2})$. The key observation is that this bound is independent of the number of states. Therefore, we can obtain this bound for arbitrarily fine discretizations of $\mathcal{M}$, and with a limiting argument extend it to $\mathcal{M}$.

For simplicity, we assume $d = 1$. It is straight-forward to extend the argument to higher dimensions. Let us denote the state space by Ω. Fix a measurable subset $S \subset \Omega$ with $\pi(S) \leq \frac{1}{2}$. Our goal is to prove $\phi(S) = \frac{Q(S,\bar{S})}{\pi(S)} \geq \Omega(\frac{1}{k^2})$. Without loss of generality, we can only consider restriction of Ω and S to a bounded set. To see that, note that if we set $\Omega_n = \binom{[-n,n]}{k}$, then clearly, $\lim_{n\to\infty} \frac{Q(S\cap\Omega_n,\bar{S}\cap\Omega_n)}{\pi(S\cap\Omega_n)} = \phi(S)$, and so for large values of n, $\frac{Q(S\cap\Omega_n,\bar{S})}{\pi(S\cap\Omega_n)} = \Theta(\phi(S))$. So suppose that $\Omega = \binom{[0,1]}{k}$. For an integer n, we consider a

discretization $\mathcal{M}_n$ of $\mathcal{M}$ defined as follows. We use n in subscript to denote quantities related to $\mathcal{M}_n$. We partition $[0,1]$ into intervals of length $\frac{1}{n}$, and identify each interval with an element in the ground set of $\mathcal{M}_n$, so $\Omega_n = \binom{[n]}{k}$. $\mathcal{M}_n$ is defined by a kernel L_n characterized below. For $i \in [n]$ let $I_i = [\frac{i-1}{n}, \frac{i}{n}]$. For any $i, j \in [n]$, we define $L_n(i,j) = \int_{I_i}\int_{I_j} L(u,v)dudv$, be the accumulative value of L over $I_i \times I_j$. One can easily see L_n is a PSD matrix, as L is a PSD operator. Moreover, L and consequently $\det_L$ is a continuous function on a closed domain, so it is uniform continuous, implying for any $\epsilon > 0$, there exists an integer $n(\epsilon)$ so that for all $n > n(\epsilon)$ and any two states $\{x_1, \ldots, x_k\}$ and $\{y_1, \ldots, y_k\}$ with $|y_i - x_i| \leq \frac{1}{n}$, we have $|\det_L(x_1, \ldots, x_k) - \det_L(y_1, \ldots, y_k)| \leq \epsilon$. Now, note that $f_\pi(y_1, \ldots, y_k) = \frac{\det_L(y_1,\ldots,y_k)}{\frac{1}{k!}\int \det_L(x_1,\ldots,x_k)dx_1\ldots dx_k}$. So, using the simple fact that for any two sequences of numbers $\{a_n\}$ and $\{b_n\}$,

$$\left(\lim_{n\to\infty} a_n = a\right) \wedge \left(\lim_{n\to\infty} b_n = b \neq 0\right) \implies \lim_{n\to\infty} \frac{a_n}{b_n} = \frac{a}{b} \tag{5.20}$$

we get that for any $\epsilon > 0$, there exists an integer $m(\epsilon)$, where $m(\epsilon)$ depends on $n(\epsilon)$, such that

$$\forall n \geq m(\epsilon), \forall \{t_1, \ldots, t_k\} \in \binom{[n]}{k} : \left|\pi_n(t_1, \ldots, t_k) - \pi(\prod_{i=1}^{k} I_{t_i})\right| \leq \frac{\epsilon}{n^k} \tag{5.21}$$

We define a set $S_n \subset \Omega_n$ corresponding to S for any n, so that

$$\lim_{n\to\infty} \phi_n(S_n) = \phi(S). \tag{5.22}$$

Clearly, the above proves the theorem as by Theorem 5.6, we know that $\phi_n(S_n) \gtrsim \frac{1}{k^2}$ for any n. In what follows, we use $A \subset B$ to denote both of $A - B$ and $B - A$ have Lebesgue measure zero. Also, define

$$S_n = \left\{ \{t_1, \ldots, t_k\} \in \binom{[n]}{k} \;\middle|\; I_{t_1} \times \cdots \times I_{t_k} \subset S \right\}.$$

Following (5.20), to prove (5.22), it is enough to argue that $\lim_{n\to\infty} Q_n(S_n, \overline{S_n}) = Q(S, \overline{S})$, and $\lim_{n\to\infty} \pi_n(S_n) = \pi(S)$. We first show the latter. This follows by (5.21) and that

$$\lim_{n\to\infty} \mu\left(\cup_{\{t_1,\ldots,t_k\}\in S_n} \prod_{i=1}^{k} I_{t_i}\right) = \mu(S) \tag{5.23}$$

for μ being the Lebesgue measure.

It remains to see $\lim_{n\to\infty} Q_n(S_n, \overline{S_n}) = Q(S, \overline{S})$. First, note that $[0,1]^{k-1}$ is a closed set, so

for any $\delta > 0$ and $\epsilon > 0$, there exists an integer $n(\delta, \epsilon)$ so that for any $n > n(\delta, \epsilon)$, and points $x_1, \ldots, x_k, x_{k+1}$ and $y_1, \ldots, y_k, y_{k+1}$ with $|x_i - y_i| \leq \frac{1}{n}$, and $\int_0^1 \det_L(x_1, \ldots, x_{k-1}, \tau) d\tau \geq \delta$, we have

$$\left| \frac{\det_L(x_1, \ldots, x_k) \det_L(x_1, \ldots, x_{k-1}, x_{k+1})}{\int_0^1 \det_L(x_1, \ldots, x_{k-1}, \tau) d\tau} - \frac{\det_L(y_1, \ldots, y_k) \det_L(y_1, \ldots, y_{k-1}, y_{k+1})}{\int_0^1 \det_L(y_1, \ldots, y_{k-1}, \tau) d\tau} \right| \leq \epsilon.$$

Therefore, similar to the case for π_n, it follows that for any $\epsilon, \delta > 0$, there exists integer $m(\delta, \epsilon)$ depending on $n(\delta, \epsilon)$ so that for any $n \geq m(\delta, \epsilon)$ and for all $t_1, \ldots, t_{k-1}, s, t \in \binom{[n]}{k+1}$ with $\sum_{i=1}^n \pi_n(t_1, \ldots, i) \geq \frac{\delta}{n^{k-1}}$

$$\left| Q_n(\{t_1, \ldots, t_{k-1}, t\}, \{t_1, \ldots, t_{k-1}, s\}) - Q(I_t \times \prod_{i=1}^{k-1} I_{t_i}, I_s \times \prod_{i=1}^{k-1} I_{t_i}) \right| \leq \frac{\epsilon}{n^{k+1}}. \tag{5.24}$$

Now, combining the above equation with (5.23), and noting ϵ and δ can be chosen arbitrary close to zero, we obtain $\lim_{n \to \infty} Q_n(S_n, \overline{S_n}) = Q(S, \overline{S})$, which completes the proof. □

Combining the theorem with Theorem 2.19, we get that $\lambda_{\mathcal{M}} \gtrsim \frac{1}{k^4}$, where $\lambda_{\mathcal{M}}$ is the poincarë constant of $\mathcal{M}$. Moreover, clearly the above argument implies the chain is π-strongly irreducible as well. So we can apply Theorem 2.16 to obtain the following corollary.

Corollary 5.12. *Let π be the k-DPP defined by L. If μ is an arbitrary starting distribution, then*

$$\tau_\mu(\epsilon) \leq O(k^4) \cdot \log \left(\frac{\mathrm{var}_\pi(\frac{f_\mu}{f_\pi})}{\epsilon} \right).$$

5.4.2 Finding a Warm Start

In this subsection we propose a simple greedy algorithm that given access to $\mathrm{CD}_L(i)$ oracles for a kernel L and any $0 \leq i \leq k-1$, returns a proper starting state such that starting the corresponding Gibbs sampler yields the guarantee of theorem 5.4. The algorithm is described in algorithm 5.3 which is essentially the continuous version of a greedy algorithm analyzed at [37] for approximate volume sampling. We can prove the following guarantee for the distribution of the output of the algorithm.

Algorithm 5.3 Choosing a starting state

Input: $\mathrm{CD}_L(i)$ oracles of L for $0 \leq i \leq k-1$.

1: Let $S = \emptyset$.

2: **for** i from 0 to $k-1$ **do**

3: Use the $\mathrm{CD}_L(i)$ oracle to generate a sample x_i and add x_i to S.

4: **end for**

return S.

Lemma 5.13. *Let μ_0 be the probability distribution of the output of Algorithm 5.3. Also let f_{μ_0} and f_π denote the PDF for μ_0 and π. Then for any $\{x_1, \ldots, x_k\} \subset \Omega$,*

$$f_{\mu_0}(\{x_1, \ldots, x_k\}) \leq (k!)^2 f_\pi(\{x_1, \ldots, x_k\}).$$

The proof essentially is a continuous extension of the argument appeared in [37].

Proof. For any $x \in \mathbb{R}^d$, let f_x be the corresponding feature map, i.e. $f_x : \mathcal{H} \to \mathbb{R}$ for some Hilbert space $\mathcal{H}$ and for any $x, y \in \mathbb{R}^d$, $L(x,y) = \langle f_x, f_y \rangle$. Fix $\boldsymbol{x} = \{x_1, \ldots, x_k\}$, and let S_k be the set of all permutations of $\{x_1, \ldots, x_k\}$. Also, for any $\sigma \in S_k$ and for any $1 \leq i \leq k-1$, define $H_\sigma^i = \langle f_{\sigma(1)}, \ldots, f_{\sigma(i)} \rangle$. By definition of algorithm 5.3 we have

$$f_\nu(\boldsymbol{x}) = \sum_{\sigma \in S_k} \left[\frac{\|f_{\sigma(1)}\|^2}{\int \|f_y\|^2 dy} \cdot \frac{d(f_{\sigma(2)}, H_\sigma^1)^2}{\int d(f_y, H_\sigma^1)^2 dx} \cdots \frac{d(f_{\sigma(k)}, H_\sigma^{k-1})^2}{\int d(f_y, H_\sigma^{k-1})^2 dy} \right].$$

In the above the range of all integrals is $\mathbb{R}^d$. Note that they are well-defined since our kernel is continuous. For any $1 \leq i \leq k-1$, let $H_*^i = \arg\min_{H = \langle f_{y_1}, \ldots, f_{y_i} \rangle} \int d(f_y, H)^2 dy$, where $y_1 \ldots, y_i$ range over $\mathbb{R}^d$. Note that, the minimum of the quantity is defined since L is continuous on a closed set. Combining with the above, and noting that for any σ, $\det(x_1, \ldots, x_k) = \|f_{\sigma(1)}\|^2 \cdot d(f_{\sigma(2)}, H_\sigma^1)^2 \ldots d(f_{\sigma(k)}, H_\sigma^{k-1})^2$, we obtain

$$\begin{aligned} f_\nu(\boldsymbol{x}) &\leq k! \cdot \frac{\det(x_1, \ldots, x_k)}{\int \|f_y\|^2 dy \cdot \int d(f_y, H_*^1)^2 dy \cdot \int d(f_y, H_*^{k-1} dy} \\ &\leq k! \cdot \frac{f_\pi(\boldsymbol{x}) \cdot \int \cdots \int_C \det(y_1, \ldots, y_k) dy_k \ldots dy_1}{k! \cdot \int \|f_y\|^2 dy \cdot \int d(f_y, H_*^1)^2 dx \cdots \int d(f_y, H_*^{k-1})^2 dy}. \end{aligned}$$

So, rearranging the above to show $\frac{f_v(\boldsymbol{x})}{f_\pi(\boldsymbol{x})} \leq (k!)^2$, it suffices to show

$$\frac{\int \cdots \int \det(y_1, \ldots, y_k) dy_k \ldots dy_1}{\int \|f_y\|^2 dy \cdot \int d(f_y, H_*^1)^2 dx \cdots \int d(f_y, H_*^{k-1})^2 dy} \leq (k!)^2. \tag{5.25}$$

To proof the above, we use induction on k. For $k = 1$, the statement is obvious as for any $y \in \mathbb{R}^d$, $\det(y) = L(y, y) = \|f_y\|^2$. It is straight-forward to see, applying the above claim will prove the induction step, and completes the proof.

Claim 5.14.

$$\int \cdots \int \det(y_1, \ldots, y_k) dy_k \ldots dy_1 \leq k^2 \left(\int d(f_y, H_*^{k-1})^2 dy \right) \left(\int \cdots \int \det(y_1, \ldots, y_{k-1}) dy_{k-1} \ldots dy_1 \right) \tag{5.26}$$

Proof of Claim 5.14. For any $\boldsymbol{y} = \{y_1, \ldots, y_k\} \subset \mathbb{R}^d$, let $G_{\boldsymbol{y}}$ be a $(k-1)$-dimensional linear subspace of $\langle f_{y_1} \ldots, f_{y_k} \rangle$ which contains the projection of $H_*^{(k-1)}$ onto $\langle f_{y_1} \ldots, f_{y_k} \rangle$. Now, for any $\boldsymbol{y}$, we apply the following lemma.

Lemma 5.15 (Lemma 2 of [37])**.** *Let S be a set of k vectors, and H be any $(k-1)$-dimesnsional subspace of $\langle S \rangle$. Then*

$$\text{VOL}(S) \leq \sum_{v \in S} d(v, H) \text{VOL}(S - v),$$

where volume of a set of vectors, refer to the volume of the parallelopiped spanned by them.

Using the above lemma, and noting for a set of points $V \subset \mathbb{R}^d$, $\det_L(V) = \text{VOL}(\{f_v\}_{v \in V})$ we can get

$$\begin{aligned} \det(\boldsymbol{y}) &\leq \left(\sum_{i=1}^k d(f_{y_i}, G_{\boldsymbol{y}}) \sqrt{\det(\boldsymbol{y} - y_i)} \right)^2 \\ &\leq k \left(\sum_{i=1}^k d(f_{y_i}, G_{\boldsymbol{y}})^2 \det(\boldsymbol{y} - y_i) \right) \quad \text{Cauchy-Schwarz Inequality.} \end{aligned}$$

By integrating the above over $\mathbb{R}^d \times \mathbb{R}^d \dots \mathbb{R}^d$ (k times), we get

$$\begin{aligned}
\int \cdots \int \det(\boldsymbol{y}) d\boldsymbol{y} &\leq k \int_{\mathbb{R}^d} \cdots \int \sum_{i=1}^{k} d(f_{y_i}, G_{\boldsymbol{y}})^2 \det(\boldsymbol{y} - y_i) d\boldsymbol{y} \\
&\leq k^2 \int_{y \in \mathbb{R}^d} \int_{z_1 \in \mathbb{R}^d} \cdots \int_{z_{k-1} \in \mathbb{R}^d} d(f_y, G_{z+y})^2 \det(z) dz dy \ \text{(setting } z = \{z_1, \dots, z_{k-1}\}) \\
&\leq \int_{y \in \mathbb{R}^d} \int_{z_1 \in \mathbb{R}^d} \cdots \int_{z_{k-1} \in \mathbb{R}^d} d(f_y, H_*^{k-1})^2 \det(z) dz dy \\
&= \left(\int d(f_y, H_*^{k-1})^2 dy \right) \left(\int_{z_1 \in \mathbb{R}^d} \cdots \int_{z_{k-1} \in \mathbb{R}^d} \det(z) dz \right),
\end{aligned}$$

where in the third inequality, the fact $d(f_y, G_{z+y}) \leq d(f_y, H_*^{k-1})$ holds because $f_y \in \langle f_{z_1}, \dots, f_{z_{k-1}}, f_y \rangle$, and G_{z+y} contains the projection of H_*^{k-1} onto this space. Thus, the proof of the claim and the lemma is complete. □

□

We use this lemma to bound $\text{var}_\pi(\frac{f_{\mu_0}}{f_\pi})$ which appears in our bound for the mixing time. In particular, we use it to prove the following.

Corollary 5.16. *Let $\mathcal{M}$ be the Gibbs sampler for the k-DPP defined by kernel $L : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. Given access to $CD_L(i)$ oracles all $0 \leq i \leq k-1$, algorithm 5.3 returns a state of $\mathcal{M}$ from a distribution μ_0 where*

$$\log \text{var}_\pi(\frac{f_{\mu_0}}{f_\pi}) \leq O(k \log k) \tag{5.27}$$

Moreover, the algorithm only uses k oracle accesses.

Proof. First of all, clearly the algorithm use each $\text{CD}_L(i)$ oracle for $1 \leq i \leq k-1$ once. So letting μ_0 be the distribution of the output of the algorithm. The proof straight-forwardly follows by applying Lemma 5.15. More precisely,

$$\text{var}_\pi(\frac{f_{\mu_0}}{f_\pi}) = \mathbb{E}_\pi \left[\left(\frac{f_{\mu_0}(\boldsymbol{x})}{f_\pi(\boldsymbol{x})} \right)^2 \right] - 1 \leq (k!)^4 \cdot \mathbb{E}_\pi [1] = (k!)^4$$

, which implies eq. (5.27) after applying logarithm to both sides. □

To prove theorem 5.4, it is enough to combine the above corollary and corollary 5.12: The

resulting algorithm is to first find a starting state using algorithm 5.3 and then run the chain for $O(k^5 \log k)$ steps using the given $\mathrm{CD}_L(k-1)$ oracle.

Remark 17. It is straight-forward to use a similar discretization argument to prove Theorem 5.11, and consequently corollary 5.12 when the domain of the kernel is restricted to a closed subset $C \subset \mathbb{R}^d$ which can be nicely discretized as in Theorem 5.11. In particular, we assume C is an sphere in the next section. More precisely, C could be any closed subset which its interior has also the same measure.

5.5 A Simple Conditional Sampler

theorem 5.4 reduces sampling from the continuous k-DPPs to having access to conditional samplers $\mathrm{CD}_L(i)$ (for $0 \leq i \leq k-1$). To implement these conditional samplers, we consider a simple rejection sampler described in algorithm 5.4. Let C be the domain of the k-DPP. The algorithm assumes that C is bounded and we have an oracle to generate uniform samples from C.

Algorithm 5.4 An Algorithm for Conditional Sampling

Input: A set of k points $x_1, \ldots, x_k \in C$.

Output: A sample from $\mathrm{CD}_L(\{x_1, \ldots, x_k\})$.

1: Let M be a number such that $M > \sup_{z \in C} L(z, z)$.
2: **while** A sample is accepted. **do**
3: Draw a uniform sample x from C and a uniform number u from $[0, 1]$.
4: If $u \leq \frac{\det_L(x_1, \ldots, x_k, x)}{M \cdot \det_L(x_1, \ldots, x_k)}$, accept and return x.
5: **end while**

Correctness of the algorithm. We want to show that algorithm 5.4 generate a sample of $\mathrm{CD}_L(\{x_1, \ldots, x_k\})$. Let Φ denote the distribution of the output and f_ϕ be its PDF. It suffices to show that for any $z \in C$, $f_\phi(z) \propto \det_L(x_1, \ldots, x_k, z)$. By the definition of the

algorithm, it is enough to verify $\frac{\det_L(x_1,\ldots,x_k,z)}{M\cdot\det_L(x_1,\ldots,x_k)} \leq 1$ which follows from $\frac{\det_L(x_1,\ldots,x_k,z)}{\det_L(x_1,\ldots,x_k)} \leq L(z,z)$ and $M > L(z,z)$. The former holds, since if we write the PSD matrix given by restricting L to $x_1,\ldots,x_k,z$ as the inner product of a set of $k+1$ vectors, then by definition $L(z,z)$ is the norm squared of the vector corresponding to z and the ratio $\frac{\det_L(x_1,\ldots,x_k,z)}{\det_L(x_1,\ldots,x_k)}$ is equal to the squared of distance of that vector from the plane spanned by vectors corresponding to $x_1, x_2, \ldots, x_k$.

Therefore, it remains to analyze the running time,

5.5.1 *Analyzing the Running Time of algorithm 5.4*

Let T be a random variable which indicates the expected number of uniform samples generated from C until the algorithm terminates. Our goal is to bound $\mathbb{E}[T]$. As we saw in the preliminaries, for the kernels that we are considering, the associated integral operator has a discrete spectrum of eigenvalues. So, let $\lambda_0 \geq \lambda_1 \geq \ldots$ be eigenvalues of L. The following relates $\mathbb{E}[T]$ to the eigenvalues.

Lemma 5.18. *For any set of points $x_1,\ldots,x_k$ as the input of algorithm 5.4, we have*

$$\mathbb{E}[T] \leq \frac{M\cdot \mathrm{VOL}(C)}{\sum_{i=k}^{\infty}\lambda_i}.$$

Proof. Let μ be the uniform distribution on C and $\boldsymbol{x} = \{x_1,\ldots,x_k\}$. We also use $\boldsymbol{x}+z$ to denote $\{x_1,\ldots,x_k,z\}$. The probability that the algorithm accepts and outputs the sample generated in the current step is

$$\mathbb{P}_{\substack{z\sim\mu\\ u\sim[0,1]}}\left[u \leq \frac{\det_L(\boldsymbol{x}+z)}{M\cdot\det_L(\boldsymbol{x})}\right] = \mathbb{E}_{z\sim\mu}\left[\frac{\det_L(\boldsymbol{x}+z)}{M\cdot\det_L(\boldsymbol{x})}\right].$$

So T forms a geometric distribution and $\mathbb{E}[T] = \frac{M\cdot\det_L(\boldsymbol{x})}{\mathbb{E}_{y\sim\mu}[\det_L(\boldsymbol{x}+y)]}$. Since $\boldsymbol{x}$ is fixed, it is enough to show $\mathbb{E}_{y\sim\mu}\left[\frac{\det_L(\boldsymbol{x}+y)}{\det_L(\boldsymbol{x})}\right] \geq \frac{\sum_{i=k}^{\infty}\lambda_i(L)}{\mathrm{VOL}(C)}$ to prove the lemma. By Mercer theorem, for any $x\in C$, there exists a function (feature map) $f_x : C \to \mathbb{R}$ such that for any $y\in C$, $L(x,y) = \langle f_x, f_y\rangle$. Now, for any $y\in C$, define $\mathcal{E}(y) = \Pi_{\langle f_{x_1},\ldots,f_{x_k}\rangle^{\perp}}(f_y)$, be the projection of f_y onto the space orthogonal to functions corresponding to $x_1,\ldots,x_k$. Then, by definition

$\frac{\det_L(\boldsymbol{x}+y)}{\det_L(\boldsymbol{x})} = \|\mathcal{E}(y)\|^2$, where recall that $\boldsymbol{x} = \{x_1, \ldots, x_k\}$. It implies

$$\mathbb{E}_{y\sim\mu}\left[\frac{\det_L(\boldsymbol{x}+y)}{\det_L(\boldsymbol{x})}\right] = \mathbb{E}_{y\sim\mu}\left[\|\mathcal{E}(y)\|^2\right] = \frac{\mathrm{tr}(\mathcal{E})}{\mathrm{VOL}(C)} \tag{5.28}$$

for the kernel $\mathcal{E} : C \times C \to \mathbb{R}$ defined by $\mathcal{E}(x,y) = \langle \mathcal{E}(x), \mathcal{E}(y)\rangle$. Now, note that, $\mathrm{tr}(\mathcal{E}) = \sum_{i=0}^{\infty} \lambda_i(\mathcal{E})$. Moreover, it follows from the definition of $\mathcal{E}$ that, $L - \mathcal{E}$ is associated to an PSD operator of rank at most k. So $\mathrm{tr}(\mathcal{E}) \geq \sum_{j=k}^{\infty} \lambda_j(L)$ which completes the proof. □

Using this algorithm as $\mathrm{CD}_L(.,1)$ oracles for L and combining that with theorem 5.4 immediately implies theorem 5.5. Next, we analyze the bound of lemma 5.18 more precisely for special kernels defined on a sphere, and show it gives an efficient sampling algorithm for k-DPPs defined by spherical Gaussian.

5.5.2 *Complexity of algorithm 5.4 for Spherical Kernels*

Let $\mathbb{S}^{d-1}$ denote the $(d-1)$-dimensional unit sphere, and let $f : [-1,1] \to \mathbb{R}$ be a continuous function. Consider a kernel $K_f : \mathbb{S}^{d-1} \times \mathbb{S}^{d-1} \to \mathbb{R}$ which can be defined by $K_f(x,y) = f(\langle x,y\rangle)$ for any $x, y \in \mathbb{S}^{d-1}$. For example, consider a *spherical* Gaussian kernel (a.k.a RBF kernel) defined by $\mathcal{G}(x,y) = \exp(-\frac{\|x-y\|^2}{2\sigma^2})$ for some scalar σ. In our setting, it is generated by taking $f(u) = \exp((-1+2u)/\sigma^2)$. As an another example, consider the polynomial kernel which is defined by $P(x,y) = (1+\langle x,y\rangle)^b$, where b is an integer known as the degree of the kernel. It is obtained by letting $f(u) = (1+u)^b$. The eigenvalues and eigen-functions of such kernels has been studied before, e.g. see [90].

Theorem 5.19 ([90]). *Let K be a kernel defined on $\mathbb{S}^{d-1} \times \mathbb{S}^{d-1}$ defined as above. Then for any $\ell \geq 0$, the associated operator to K has an eigenvalue λ_ℓ with multiplicity $N(d,\ell) = \frac{(2\ell+d-2)(\ell+d-3)!}{\ell!(d-2)!}$ given by*

$$\lambda_\ell = \mathrm{VOL}(\mathbb{S}^{d-2}) \int_{-1}^{1} f(\tau) P_\ell(d;\tau)(1-\tau^2)^{\frac{d-3}{2}} d\tau$$

where $P_\ell(;.)$ is the Legendre polynomial of degree ℓ in dimension d.

The above integral formula for eigenvalues turns out to be computable or easy to bound

for several kernels. In particular, [90] gives explicit formula for spherical Gaussians. The following lemma states its implication for bounding the complexity of Algorithm algorithm 5.4. Note that generating a uniform sample from a d-dimensional sphere can be done in $O(d)$ time, so combining the lemma with theorem 5.5 yields an efficient algorithm for sampling from spherical Gaussians on a sphere. Recall that T denotes the number of samples generated from C in a run of Algorithm algorithm 5.4, we prove the following.

Lemma 5.20. *Let $\mathcal{G}_\sigma$ be a spherical Gaussian kernel on the unit sphere given by $\mathcal{G}_\sigma(x,y) = \exp(-\|x-y\|^2/2\sigma^2)$ for $x,y \in \mathbb{S}^{d-1}$. Also let $k \leq \exp(\frac{d}{4})$, and set t to be the smallest integer that $\frac{d^t}{t!} \geq 2k$. Then for any set of k points as the input of algorithm 5.4, we have*

$$\mathbb{E}[T] \leq e^{\frac{2}{\sigma^2}} \cdot \sigma^{2t} \cdot t!.$$

Moreover, if $\sigma \lesssim \frac{1}{\sqrt{\log k}}$, then $\mathbb{E}[T] = O(1)$.

To prove the above we use the following special instance of theorem 5.19.

Lemma 5.21 ([90])**.** *Let $\mathcal{G}_\sigma$ be the Gaussian kernel with variance σ^2 restricted to the unit sphere with the uniform measure, i.e. for any $x,y \in \mathbb{S}^{d-1}$: we have $\mathcal{G}_\sigma(x,y) = \frac{\exp(-\|x-y\|^2/\sigma^2)}{\mathrm{VOL}(S^{d-1})}$. For any integer $k \geq 0$, $\mathcal{G}_\sigma$ has an eigenvalue μ_k with multiplicity $N(d,k) = \frac{(2k+d-2)(k+d-3)!}{k!(d-2)!}$ where*

$$\left(\frac{2e}{\sigma^2}\right)^k \cdot \frac{A_1}{(2k+d-2)^{k+\frac{d-1}{2}}} \leq \mu_k \leq \left(\frac{2e}{\sigma^2}\right)^k \cdot \frac{A_2}{(2k+d-2)^{k+\frac{d-1}{2}}}, \tag{5.29}$$

for $A_1 = e^{-\frac{2}{\sigma^2}-\frac{1}{12}} \frac{1}{\sqrt{\pi}} (2e)^{\frac{d}{2}-1} \Gamma\left(\frac{d}{2}\right)$ and $A_2 = A_1 \cdot e^{\frac{1}{12}+\frac{1}{\sigma^4}}$.

Proof of lemma 5.20. Let $\lambda_0 \geq \lambda_1 \ldots$ be eigenvalues of $\mathcal{G}_\sigma$. Note that $\mathcal{G}_\sigma(x,x) = 1$ for all x. Combining that with Lemma 5.1 and the fact that we are considering the kernel with respect to the uniform measure [2] we get

$$\mathbb{E}[T] \leq \frac{1}{\sum_{j=k}^{\infty} \lambda_j}.$$

[2]The kernel, we are considering in the paper is not normalized by the uniform measure. Note that, after normalizing the volume term cancels out.

We first prove the second part by showing if $\sigma \leq \frac{1}{2\sqrt{\log k}}$, then $\sum_{j=k}^{\infty} \lambda_j \geq \Omega(1)$. Using the Cauchy-Schwarz inequality we have $k \cdot \sum_{i=0}^{k-1} \lambda_i^2 \geq \left(\sum_{i=0}^{k-1} \lambda_i\right)^2 = \left(1 - \sum_{j=k}^{\infty} \lambda_j\right)^2$. We show $\sum_{i=0}^{k-1} \lambda_i^2 \leq \frac{1}{k^2}$ which implies $\sum_{j=k}^{\infty} \lambda_j \geq (1 - 1/\sqrt{k})$ which completes the proof. To see that, note that

$$\sum_{i=0}^{k-1} \lambda_i^2 \leq \operatorname{tr}(\mathcal{G}_\sigma^2) = \mathbb{E}_{x,y\sim\mu}\left[e^{-\|x-y\|^2/2\sigma^2}\right],$$

where μ is the uniform measure on the sphere. Fix $x \in \mathbb{S}^{d-1}$. It follows from basic concentration inequalities for Gaussian measures that $\mathbb{E}_{y\sim\mu}\left[e^{-\|x-y\|^2/2\sigma^2}\right] \leq e^{-1/2\sigma^2}$ which implies the bound on the trace and finishes the proof of this case.

So from now on, we only need to prove for any σ

$$\sum_{j=k}^{\infty} \lambda_j \gtrsim \frac{e^{\frac{-2}{\sigma^2}}}{t! \cdot \sigma^{2t}}. \tag{5.30}$$

Let $\mu_0 > \mu_1 > \dots$ be distinct eigenvalues of the kernel given by lemma 5.21 where for any j, the multiplicity of μ_j is $n_j = N(d, j)$. It suffices to show $\frac{n_t \mu_t}{2} \geq \frac{e^{\frac{-2}{\sigma^2}}}{t! \cdot \sigma^{2t}}$ where we are using the fact that for any j, $n_j \geq \frac{d^j}{j!}$, and so $n_t \geq 2k$. Now using $n_t \geq \frac{d^t}{t!}$, and the bound on μ_t by Lemma 5.21, we get

$$\begin{aligned}
n_t \mu_t &\gtrsim \frac{d^t}{t!} \cdot \frac{e^{\frac{-2}{\sigma^2}} (2e)^{t+\frac{d}{2}} \Gamma(\frac{d}{2})}{\sigma^{2t} \cdot (2t+d)^{t+\frac{d+1}{2}}} \\
&\gtrsim \frac{d^t}{t!} \cdot \frac{e^{\frac{-2}{\sigma^2}} (2e)^t \cdot d^{\frac{d+1}{2}}}{\sigma^{2t} (2t+d)^{t+\frac{d+1}{2}}} && \text{Sterling's approximation} \\
&\geq \frac{e^{\frac{-2}{\sigma^2}} (2e)^t}{\sigma^{2t} \cdot t! \cdot (1+\frac{2t}{d})^{t+\frac{d+1}{2}}} \gtrsim \frac{e^{\frac{-2}{\sigma^2}} 2^t}{\sigma^{2t} \cdot t! \cdot e^{\frac{2t^2}{d}}} && \text{by } (1+2t/d) \leq e^{2t/d}.
\end{aligned}$$

Noting that $k \leq \exp(d/4)$ implies $t \leq \frac{d}{4}$ and $\exp(2t/d) \leq 2$, completes the proof of (5.30).

□

Note that a direct consequence of the above lemma is that in the case $k = \text{poly}(d)$ and $\sigma = \Omega(1)$, the running time of the algorithm is polynomial in terms of σ, d.

Our final algorithm for sampling from continuous k-DPPs is described in algorithm 5.5. As explained, it assumes that we have an oracle that generate uniform samples from the domain.

Algorithm 5.5 Gibbs Sampler for Continuous k-DPPs

Input: A kernel $L : C \times C \to \mathbb{R}$ along with an oracle which generates uniform samples form C.

Let $S = \emptyset$.
Let M be a number such that $M > \sup_{z \in C} L(z, z)^3$.
for i from 0 to $k - 1$ **do**
    while A sample is accepted **do**
        Draw a uniform sample x from C and a uniform number u from $[0, 1]$.
        If $u \leq \frac{\det_L(S \cup x)}{M \cdot \det_L(S)}$, accept x and set $S = S \cup x$.
    end while
end for
Let $\tau = \tilde{O}(k^5 \log \frac{1}{\epsilon})$.
for τ iterations **do**
    Let $S = \{x_1, \ldots, x_k\}$ and pick an uniform random integer $0 \leq i \leq k - 1$. Set $S = S - x_i$
    while A sample is accepted **do**
        Draw a uniform sample x from C and a uniform number u from $[0, 1]$.
        If $u \leq \frac{\det_L(S \cup x)}{M \cdot \det_L(S)}$, accept x and set $S = S \cup x$.
    end while
end for
return S.

Combining lemma 5.20 and theorem 5.4, we obtain the following guarantee for Gaussian kernels.

5.6 Experimental Results

We implement our algorithm and evaluate the mixing time for various kernels and parameters to empirically confirm our results. In particular, we consider the two family of kernels:

1. Spherical Gaussian given by $L(x, y) = \exp(-\|x - y\|^2/\sigma^2)$ for parameter σ. In all experiments, we let the domain be the d-dimensional unit ball.

2. Polynomial kernel defined by $K(x, y) = (1 + \langle x, y \rangle)^b$ for some parameter b which is also known as the degree of the kernel. In our experiments, we let the domain be the unit hypercube in $\mathbb{R}^d$.

Simulation Setup: For a fixed kernel, we use the rejection sampler described in algorithm 5.4 as the conditional sampler of the kernel. To do the sampling from the continuous k-DPP defined by the kernel, we first run algorithm 5.3 to find a starting state. Then we start simulating the chain; At each step, one of the k current points is chosen uniformly and replaced by the point returned by the rejection sampler. The pseudo-code of the method is presented in Algorithm algorithm 5.5. Finally, to evaluate the mixing time, we use the following criteria.

Empirical Mixing: We employ the multivariate extension of the Gelman and Rubin multiple sequence method [25]. To be consistent with that, instead of k-subsets, we work with k-tuples as the state space by randomly labeling points in each step. So each state can be represented by a $k \times d$ matrix. We run $m = 10$ copies of our algorithm independently. We consider each column separately as the projection of the state onto a coordinate of the ambient space, and at each step compute its associated multivariate Potential Scale Reduction Factor (PSRF) over these m runs. We set the first time that the average of these d PSRF values drops below $\alpha = 1.1$, as our empirical measure for the mixing time. For any fixed kernel, we repeat this process 10 times and report the average

as the (empirical) mixing time.

Experiments: We use the above criteria to evaluate the empirical mixing time of the chain for the Gaussian and polynomial kernels, defined on the unit ball and unit hypercube, respectively. The results are demonstrated in Figure 5.2 and Figure 5.3. In the first experiment, we investigate the change of mixing time with respect to size parameter k; k varies from 5 to 40, and other parameters are fixed, $d = 40$, $\sigma = 1$, $b = 5$. As stated, our theoretical results guarantees an $O(k^4)$ dependency. However, our experiments demonstrated in Figure 5.2, shows a much smaller bound (roughly $O(k^2)$).

In the second experiment, we fix number of points $k = 10$, and values $\sigma = 1$ ($b = 5$) for the Gaussian (Polynomial) kernel, and vary the dimension from 5 to 50. As illustrated in Figure 5.3, the mixing time is quite unchanged with small fluctuations which corroborates independence of the mixing time from these parameters.

Finally, we look at the impact of b and σ on the mixing time. As shown in Figure 5.3, for fixed values of $k = 10$ and $d = 40$, the change in mixing time with respect to changes in σ and b seems negligible, as expected by our theoretical findings.

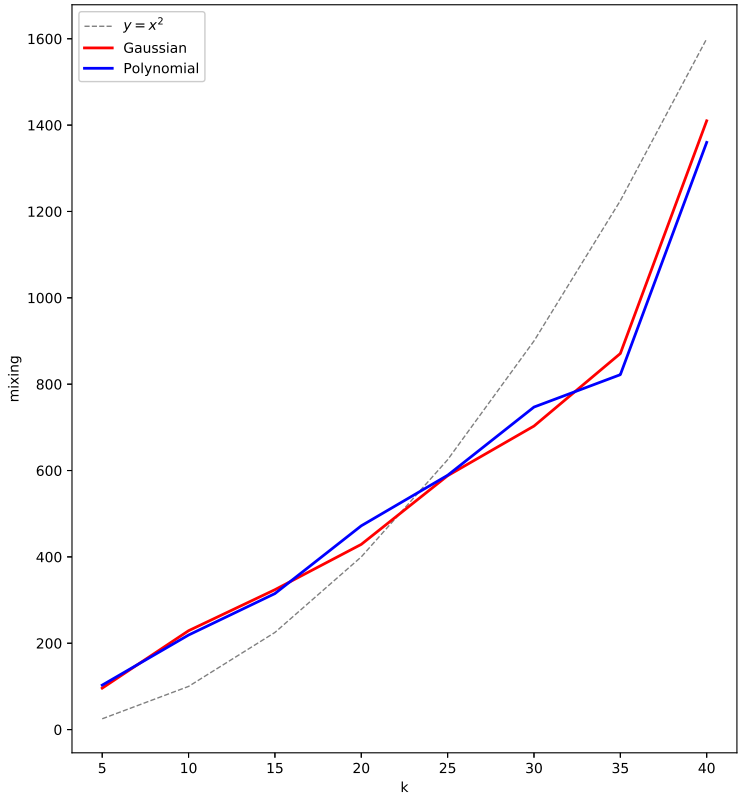


Figure 5.2: Empirical mixing time for different values of k while dimension and other parameters are fixed ($d = 40$, $\sigma = 1$ and $b = 5$)

.

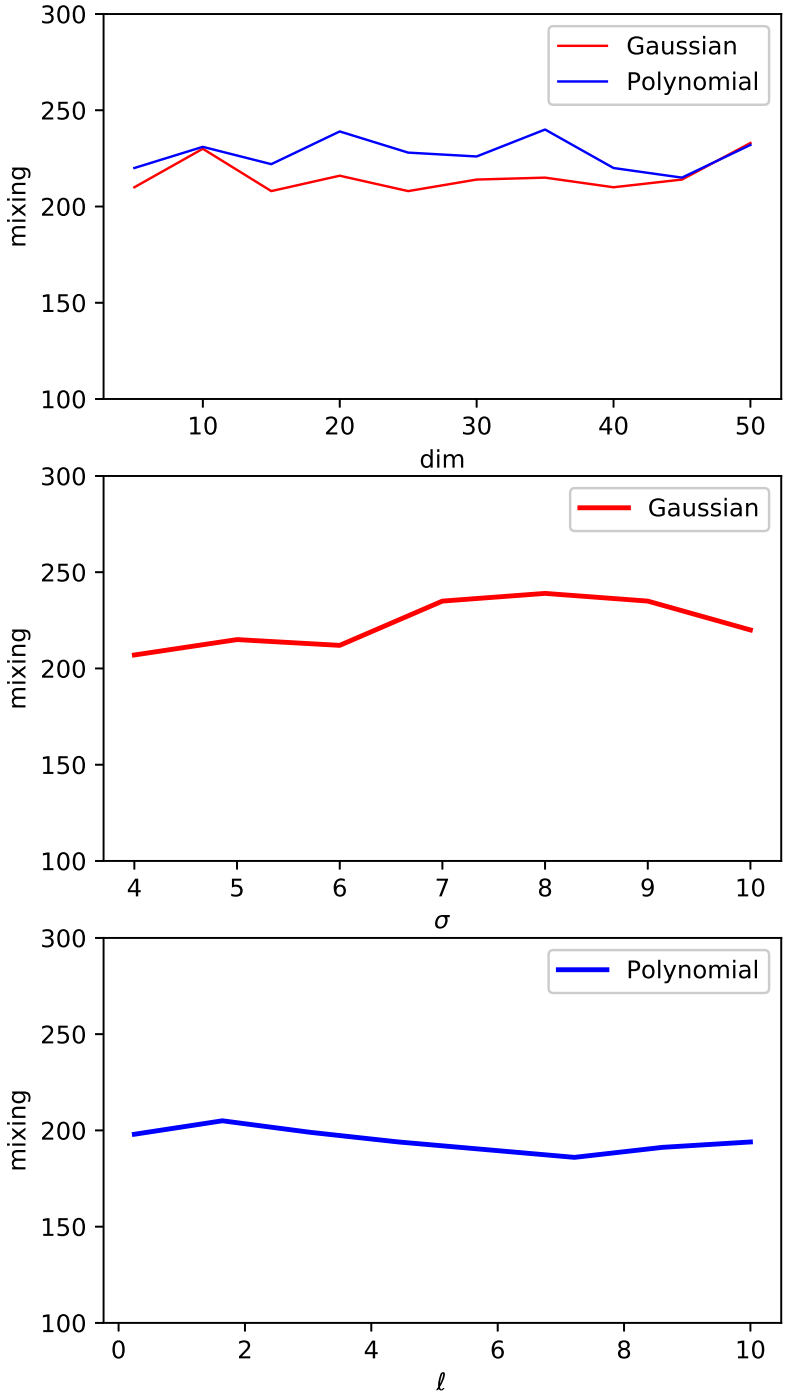


Figure 5.3: Plots of the empirical mixing time for a fixed k and varying σ (middle plot), b (bottom plot), and d (top plot).

Part II

COMPOSABLE CORE-SETS FOR DETERMINANT MAXIMIZATION

Chapter 6

OPTIMAL COMPOSABLE CORE-SETS FOR DETERMINANT MAXIMIZATION PROBLEMS VIA SPECTRAL SPANNERS

In this chapter, we study a generalization of classical combinatorial graph spanners to the spectral setting. Given a set of vectors $V \subseteq \mathbb{R}^d$, we say a set $U \subseteq V$ is an α-spectral k-spanner, for $k \leq d$, if for all $v \in V$ there is a probability distribution μ_v supported on U such that

$$vv^\intercal \preceq_k \alpha \cdot \mathbb{E}_{u \sim \mu_v} \left[uu^\intercal \right],$$

where for two matrices $A, B \in \mathbb{R}^{d \times d}$ we write $A \preceq_k B$ iff the sum of the bottom $d - k + 1$ eigenvalues of $B - A$ is nonnegative. In particular, $A \preceq_d B$ iff $A \preceq B$. We show that any set V has an $\tilde{O}(k)$-spectral spanner of size $\tilde{O}(k)$ and this bound is almost optimal in the worst case.

We use spectral spanners to study composable core-sets for spectral problems. We show that for many objective functions one can use a spectral spanner, independent of the underlying function, as a core-set and obtain almost optimal composable core-sets. For example, for the k-determinant maximization problem, we obtain an $\tilde{O}(k)^k$-composable core-set, and we show that this is almost optimal in the worst case.

Our algorithm is a spectral analogue of the classical greedy algorithm for finding (combinatorial) spanners in graphs. We expect that our spanners find many other applications in distributed or parallel models of computation. Our proof is spectral. As a side result of our techniques, we show that the rank of diagonally dominant lower-triangular matrices are robust under "small perturbations" which could be of independent interests.

6.1 Introduction

6.1.1 Spectral Spanners

Given a graph G with n vertices $\{1, \ldots, n\}$, we say a subgraph H is a α-(combinatorial) spanner if for every pair of vertices u, v of G,

$$\text{dist}_H(u, v) \leq \alpha \cdot \text{dist}_G(u, v),$$

where $\text{dist}_G(u, v)$ is the shortest path distance between u, v in G. It has been shown that for any α, G has an α-spanner with only $n^{1+O(1)/\alpha}$ many edges and that can be found efficiently [46]. Such a spanner can be found by a simple algorithm which repeatedly finds and adds an edge $f = (u, v)$ where $\text{dist}_H(u, v) > \alpha$. Combinatorial spanners have many applications in distributed computing [104, 49, 68], optimization [43, 9], etc.

We define and study a spectral generalization of this property. Given a set of vectors $V \subseteq \mathbb{R}^d$, we say a set $U \subseteq V$ is an α-spectral d-spanner of V if for any vector $v \in V$, there exists a probability distribution μ_v on the vectors in U such that

$$vv^\intercal \preceq \alpha \cdot \mathbb{E}_{u \sim \mu_v}\left[uu^\intercal\right] \quad \text{equiv} \quad \langle x, v\rangle^2 \leq \alpha \cdot \mathbb{E}_{u \sim \mu_v}\left[\langle x, u\rangle^2\right], \forall x \in \mathbb{R}^d.$$

To see that this is a generalization of the graph case, let $b_{u,v} = e_u - e_v$ be the vector corresponding to an edge $\{u, v\}$ of G, where e_u is the indicator vector of the vertex u. It is an exercise to show that for $V = \{b_e\}_{e \in E(G)}$ and for any α-combinatorial spanner H of G, the set $U = \{b_e\}_{e \in E(H)}$ is an α^2-spectral spanner of V.

The following theorem is a special case of our main theorem.

Theorem 6.1 (Main theorem for $k = d$)**.** *There is an algorithm that for any set of vectors $V \subseteq \mathbb{R}^d$ finds an $\tilde{O}(d)$-spectral d-spanner of size $\tilde{O}(d)$ in time polynomial in d and size of $|V|$* [1].

Our algorithm is a spectral generalization of the greedy algorithm mentioned above for finding combinatorial spanners.

---

[1] The asymptotic notation $\tilde{O}(f(n))$ hides logarithmic factors in $f(n)$.

We further study generalizations of our spectral spanners to weaker forms of PSD inequalities. For two matrices A, B we write $A \preceq_k B$ if for every projection matrix Π onto a $d-k+1$ dimensional linear subspace, $\langle A, \Pi \rangle \leq \langle B, \Pi \rangle$. For example, if $A \preceq_k B$, then sum of the top k eigenvalues of A is at most the sum of the top k eigenvalues of B. Analogously, we say $U \subseteq V$ is an α-spectral k-spanner of V, if for any $v \in V$, there is a distribution μ_v on U such that $vv^\intercal \preceq_k \alpha \cdot \mathbb{E}_{u \sim \mu_v} uu^\intercal$. In our main theorem we generalize the above statement to all $k \leq d$ and we show that to construct an $\tilde{O}(k)$ spectral k-spanner we only need to use $\tilde{O}(k)$ many vectors independent of the ambient dimension of the space.

Theorem 6.2 (Main)**.** *There is an algorithm that for any set of vectors $V \subseteq \mathbb{R}^d$ finds an $\tilde{O}(k)$-spectral k-spanner of size $\tilde{O}(k)$.*

Furthermore, for any $\epsilon > 0$ and $k \leq d$, there exists a set $V \subseteq \mathbb{R}^d$ of size $e^{\Omega(k^\epsilon)}$ such that any $k^{1-\epsilon}$-spectral spanner of V must have all vectors of V.

6.1.2 Composable core-sets

"Composable core-sets" are an efficient framework for solving optimization problems in massive data models. Our main application of spectral spanners is to design (composable) core-sets for spectral problems. A function $c(V)$ that maps $V \subseteq \mathbb{R}^d$ into its subset is called an α-composable core-set of size t for the function $f(\cdot)$ [4, 61], if for any collection of sets $V_1, \ldots, V_p \subset \mathbb{R}^d$, we have

$$f(c(V_1) \cup \ldots \cup c(V_p)) \geq \frac{1}{\alpha} \cdot f(V_1 \cup \ldots \cup V_p)$$

and $|c(V_i)| \leq t$ for any V_i. A composable core-set of a small size immediately yields a communication-efficient distributed approximation algorithm: if each set V_i is stored on a separate machine, then all machines can compute and transmit core-sets, $c(V_i)$'s, to a central server, which can then perform the final computation over the union. Similarly, core-sets make it possible to design a streaming algorithm which processes N vectors in

one pass using only $\sqrt{Nt}$ storage. This is achieved by dividing the stream of data into blocks of size $\sqrt{Nt}$, computing and storing a core-set for each block, and then performing the computation over the union.

We show that, for a given set $V_i \in \mathbb{R}^d$, an α-spectral spanner of V_i for a proper value of α provides a good core-set of V_is. Specifically, we show that for many (spectral) optimization problems, such as determinant maximization, D-optimal design or min-eigenvalue maximization, this approach leads to almost the best possible composable core-set in the worst case.

In what follows we discuss a specific application, to determinant maximization, in more detail.

Composable Core-sets for Determinant Maximization. Recall that given a set of vectors $V = \{v_1, \ldots, v_n\}$ and a parameter k, the k-DPP defined by these vectors is a distribution over subsets of V of size k that to any subset S with k elements assigns the following probability

$$\mathbb{P}(S) \sim \det_k(\sum_{v \in S} vv^T)$$

As discussed in chapter 3, this distribution formalizes a notion of diversity, as sets of vectors that are "substantially different" from each other are assigned higher probability. One can then find the "most diverse" k-subset in P by computing S that maximizes $\mathbb{P}(S)$, i.e., solving the *maximum a posteriori (MAP) decoding* problem:

$$\max_{S \subset P, |S|=k} \mathbb{P}(S).$$

This problem is also known as *k-determinant maximization*.

Here we use our results on spectral spanners to construct an almost optimal composable core-set for MAP problem. Before mentioning our result let us briefly discuss relevant previous work on this problem. The MAP problem is hard to approximate up to a factor of 2^{-ck} for some constant $c > 0$, unless P=NP. [31, 32]. This lower bound was matched

qualitatively by a recent paper of [99], who gave an algorithm with e^k-approximation guarantee. Since the data sets in the aforementioned applications can be large, there has been a considerable effort on developing efficient algorithms in distributed, streaming or parallel models of computation [93, 117, 103, 94, 91, 14]. All of these algorithms relied on the fact that the logarithm of the volume is a submodular function, which makes it possible to obtain multiplicative factor approximation algorithms (assuming some lower bound on the volume, as otherwise the logarithm of the volume can be negative). See Section 6.1.4 for an overview. However, this generality comes at a price, as multiplicative approximation guarantees for the logarithm of the volume translates into "exponential" guarantees for the volume, and necessitates the aforementioned extra lower bound assumptions. As a result, to the best of our knowledge, no multiplicative approximation factor algorithms were known before for this problem, for streaming, distributed or parallel models of computation.

We present the first (composable) core-set construction for the determinant maximization problem. Our main contributions are:

Theorem 6.3. *There exists a polynomial time algorithm for computing an $\tilde{O}(k)^k$-composable core-set of size $\tilde{O}(k)$, for the k-determinant maximization problem.*

Let us discuss the proof of the above theorem for the case $k = d$ using our main theorem 6.1. Given sets of vectors $V_1, \ldots, V_m$ let $U_1, \ldots, U_m$ be their $\tilde{O}(d)$-spectral spanners respectively. Let

$$S = \text{argmax}_{S \subseteq \cup_i V_i : |S| = d} \mathbb{P}(S).$$

Consider the matrix $A = \sum_{v \in S} \mathbb{E}_{u \sim \mu_v} \left[uu^T \right]$, that is we substitute each vector v in S by a convex combination of the vectors in the spectral spanner(s). Then, by definition of spectral spanner,

$$\frac{1}{\alpha} \sum_{v \in S} vv^T \preceq A.$$

Since determinant is a monotone function with respect to the Loewner order of PSD

matrices,

$$\frac{1}{\alpha^d}\mathbb{P}(S) = \det\left(\frac{1}{\alpha}\sum_{v\in S} vv^T\right) \leq \det(A).$$

The matrix A can be seen as a fractional solution to the determinant maximization problem. In fact [99] showed that A can be rounded to a set T of size $|T| = d$ such that $\det(A) \leq e^d \det(T)$. Therefore, we obtain an $(e\alpha)^d$ approximation for determinant maximization (see section 6.6.1 for more details).

The technique that we discussed above can be applied to many optimization problems. In general, if instead of the determinant, we wanted to maximize any function $f : \mathbb{R}^{d\times d} \to \mathbb{R}_+$, that is monotone on the Loewner order of PSD matrices, we can use the above approach to construct a fractional solution A supported on the spectral spanners such that $f(A)$ is at least the optimum (up to a loss that depends on α). Then, we can use randomized rounding ideas to round the the matrix A to an integral solution of f. See section 6.6.2 for further examples.

We complement the above theorem by showing the above guarantee is essentially the best possible.

Theorem 6.4. *Any composable core-sets of size at most k^β for the k-determinant maximization problem must incur an approximation factor of at least $(\frac{k}{\beta})^{k(1-o(1))}$, for any $\beta \geq 1$.*

Note that our lower bound of $(\frac{k}{\beta})^{k(1-o(1))}$ for the approximation factor achievable by composable core-sets is substantially higher than the approximation factor e^k of the best off-line algorithm, demonstrating a large gap between these two models.

6.1.3 Overview of the Techniques

In this part, we give a high level overview of the proof of theorem 6.2. Our proof has two steps: First, we solve the "full dimensional version of the problem, i.e., we construct an $\tilde{O}(d)$-spectral d-spanner of size $\tilde{O}(d)$ for a given set of vectors in $\mathbb{R}^d$ as promised in

theorem 6.1. Then, we reduce the "low dimensional" version of the problem, i.e., finding k-spanners for $k < d$, to the full dimensional version in a $\tilde{O}(k)$-dimensional space.

Step 1: Our high-level plan is to "augment" the classical greedy algorithm for finding combinatorial spanners in graphs to the spectral setting. First, we rewrite the combinatorial algorithm in spectral language.

Let G be a graph with vertex set $V(G)$ and edge set $E(G)$. Recall that for any edge $e = \{u, v\} \in E(G)$ $b_e = e_u - e_v$. As alluded to in the introduction, if H is an α-combinatorial spanner of G, then $U = \{b_e\}_{e \in E(H)}$ is an α^2-spectral spanner of $\{b_e\}_{e \in E(G)}$. The following algorithm gives an α-combinatorial spanner with $n^{1+O(1)/\alpha}$ edges: Start with an empty graph H. While there is an edge $f = \{u, v\}$ in G where $\text{dist}_H(u, v) > \alpha$, add it to H. One can observe that $\text{dist}_H(u, v) > \alpha$ iff, for any distribution μ on $E(H)$, $b_f b_f^T \not\preceq \alpha^2 \mathbb{E}_{e \sim \mu} \left[b_e b_e^T \right]$.

This observation suggests a natural algorithm in the spectral setting: At each step find a vector $v \in V$ such that for all μ supported on the set of vectors already chosen in the spanner, $vv^T \not\preceq \alpha \cdot \mathbb{E}_{u \sim \mu} [uu^\intercal]$, and add it to the spanner. We can implement such an algorithm in polynomial time, but we cannot directly bound the size of the spectral spanner that such an algorithm constructs using our current techniques.

So, we take a detour. First, we solve a seemingly easier problem by changing the order of quantifiers in the definition of the spectral spanner. For $V \subseteq \mathbb{R}^d$, a subset $U \subseteq V$ is a weak α-spectral spanner of V, if for all $v \in V$ and $x \in \mathbb{R}^d$ there is a distribution $\mu_{v,x}$ on U such that

$$\langle v, x \rangle^2 \leq \alpha \cdot \mathbb{E}_{u \sim \mu_{v,x}} \left[\langle u, x \rangle^2 \right] \quad \text{equiv} \quad \langle v, x \rangle^2 \leq \alpha \cdot \max_{u \in U} \langle u, x \rangle^2.$$

To find a weak spectral spanner, we use the analogue of the greedy algorithm: Let U be the set of vectors already chosen; while there is a vector $v \in V$ and $x \in \mathbb{R}^d$ such that $\langle x, v \rangle^2 > \alpha \cdot \max_{u \in U} \langle u, x \rangle^2$ we add $\text{argmax}_v \langle x, v \rangle^2$ to U.

We prove that for $\alpha = \tilde{O}(d)$ the above algorithm stops in $\tilde{O}(d)$ steps. Suppose that the algorithm finds vectors $u_1, \ldots, u_m$ together with corresponding "bad" directions $x_1, \ldots, x_m$, where x_i being a bad direction for u_i means that

$$\langle u_i, x_i \rangle^2 > \alpha \langle u_j, x_i \rangle^2, \forall 1 \leq i \leq m, \forall 1 \leq j < i. \tag{6.1}$$

We need to show that $m = \tilde{O}(d)$. We consider the matrix $M \in \mathbb{R}^{m \times m}$ where $M_{i,j} = \langle u_i, x_j \rangle$. By the above constraints M is diagonally dominant and approximately lower triangular matrix. But since M has rank at most d as it is the inner product matrix of vectors lying in $\mathbb{R}^d$, we conclude that $m = \tilde{O}(d)$. Note that in the extreme case, where M is truly lower triangular the latter fact obviously holds because then $\text{rank}(M) = m$. As a side result, we also show that the rank of lower triangular matrices is robust under small perturbations, (see lemma 6.17).

The above argument shows that the spectral greedy algorithm gives a weak spectral spanner for $\alpha = \tilde{O}(d)$ of size $\tilde{O}(d)$. To finish the proof of theorem 6.1 we need to find a (strong) α-spectral spanner from our weak spanner. We use a duality argument to show that any weak spectral spanner is indeed an α-spectral spanner. Let U be a weak spectral spanner. To verify that U is an α-spectral spanner, we need to find a distribution μ_v for any $v \in V$ supported on U such that $vv^T \preceq \alpha \cdot \mathbb{E}_{u \sim \mu_v}[uu^\intercal]$. We can find the best distribution μ_v using an SDP with variables p_u for all $u \in U$ denoting $\mathbb{P}_{\mu_v}(u)$. Instead of directly bounding the primal, we write down the dual of the SDP and use hyperplane separating theorem to show that indeed such a distribution exists.

It was pointed to us by an anonymous reviewer that one can use approximate John's ellipsoid [13] to find an $O(d)$-weak-spectral d-spanner of size $\tilde{O}(d)$. This improves the guarantees of our algorithm by a $\log d$ factor. We discuss the details at the end of Section 6.4.1. Let us briefly mention the advantages of our algorithm over the John's ellipsoid method: First, in finding the weak spanner one can tune the value of α in (6.1) based on the structure of the given data points and the ideal size of the core-set. We also expect that in many real world applications, one can use our algorithm to obtain $\text{polylog}(d)$-

spectral d-spanners of size $\tilde{O}(d)$. Secondly, to implement our algorithm we only need to solve linear programs with $O(d)$ many variables. This requires polynomially smaller amount of memory compared to the SDP solvers one needs to use to solve the John's ellipsoid. Lastly, our algorithm is easier to parallelize.

Step 2: To reduce the k-spanner problem to the "full dimensional" case, we use the greedy algorithm of [31] to find a set of vectors $W \subset V$ of size $\tilde{O}(k)$ such that for any $v \in V$,

$$v_{\langle W\rangle^\perp} v_{\langle W\rangle^\perp}^\intercal \preceq_k O(1) \cdot \mathbb{E}_w \left[ww^\intercal\right] \tag{6.2}$$

where the expectation is over the uniform distribution on W, and $v_{\langle W\rangle^\perp}$ represents the projection of v onto the space orthogonal to the linear subspace spanned by W. Then, we project all vectors in V onto the space $\langle W\rangle$, and we solve the full dimensional version, i.e., we find $U \subseteq V$ of size $\tilde{O}(|W|)$ such that for any $v \in V$, there exists a distribution μ_v supported on U which satisfies

$$v_{\langle W\rangle} v_{\langle W\rangle}^\intercal \preceq \tilde{O}(k) \cdot \mathbb{E}_{u\sim\mu_v} \left[u_{\langle W\rangle} u_{\langle W\rangle}^\intercal\right]. \tag{6.3}$$

Ideally on the RHS of the above, we need to have $uu^\intercal$ instead of $u_{\langle W\rangle} u_{\langle W\rangle}^\intercal$ which can be achieved by incurring an extra constant factor by applying (6.2). It is not hard to see from the above two equations that $U \cup W$ is an $\tilde{O}(k)$-spectral k-spanner for V.

It remains to find the set $W \subseteq V$ satisfying (6.2). We use the following algorithm: Let $W = \emptyset$. For $i = 1, \ldots, \tilde{O}(k)$, add $\operatorname{argmax}_{v\in V} \|v_{\langle U\rangle^\perp}\|$ to W. Intuitively, we greedily choose a set of vectors of size $\tilde{O}(k)$ to minimize the projection of the remaining vectors in the orthogonal space of $\langle W\rangle$.

To prove (6.2), we need to show

$$v_{\langle W\rangle^\perp} v_{\langle W\rangle^\perp}^\intercal \preceq_k O(1) \cdot \mathbb{E}_{w\sim\mu} \left[ww^\intercal\right].$$

Equivalently, after choosing the worst projection matrix onto a $d-k+1$ linear subspace,

it is enough to show

$$\|v_{\langle W\rangle^\perp}\|^2 \leq O(1) \cdot \sum_{i=k}^{d} \lambda_i(\mathbb{E}_{w\sim\mu}[ww^\intercal]). \tag{6.4}$$

To prove the above inequality, we use properties of the greedy algorithm to study singular values of the matrix obtained by applying the Gram-Schmidt process on the vectors in W.

Lower bounds. As we discussed in the intro, it is not hard to prove that the guarantee of theorem 6.1 is tight in the worst case. However, one might wonder if it is possible to design better composable core-sets for determinant maximization and related spectral problems. We show that for many such problems we obtain the best possible composable core-set in the worst case. Let us discuss the main ideas of theorem 6.4.

We consider the case $k = d$ for simplicity of exposition. For a set $V \subseteq \mathbb{R}^d$ and a linear transformation $Q \in \mathbb{R}^{d\times d}$, define $QV = \{Qv\}_{v\in V}$. Choose a set $V \subseteq \mathbb{R}^d$ of unit vectors such that for any distinct $u, v \in V$, $\langle u, v\rangle^2 \leq 1/d^{1-o(1)}$. This can be achieved with high probability by selecting points in V independently and uniformly at random from the unit sphere. Recall that the set V can have exponentially (in d) large number of vectors. Consider sets $A_1, \ldots, A_d$ and $B_1, \ldots, B_{d-1}$ in a $(2d-1)$-dimensional space such that:

- For each $1 \leq i \leq d$, let $A_i = R_i V$ where R_i is a rotation matrix which maps $\mathbb{R}^d$ to $\langle e_1, e_2, \ldots e_{d-1}, e_{(d-1)+i}\rangle$ and it maps a uniformly randomly chosen vector of V to $e_{(d-1)+i}$.
- For each $1 \leq i \leq d-1$, $B_i = \{Me_i\}$, where M is a "large" number.

Our instance of determinant maximization is simply $QA_1, \ldots, QA_d, QB_1, \ldots, QB_{d-1}$ for a random rotation matrix $Q \in \mathbb{R}^{(2d-1)\times(2d-1)}$.

Observe that the optimal set of $2d-1$ vectors contains $Qe_{(d-1)+i}$'s from QA_i's and QMe_i's from B_i's, and has value equal to $(M^{d-1})^2$. However, since Q is a random rotation, the

core-set function cannot determine which vector in QA_i was aligned with $Qe_{(d-1)+i}$. Recall that the core-set algorithm must find a core-set of A_i by only observing the vectors in A_i. Thus unless core-sets are exponentially large in d, there is a good probability that, for all i, the core-set for QA_i does not contain $Qe_{(d-1)+i}$. For a sufficiently large M, all vectors QMe_i from QB_i must be included in any solution with a non-trivial approximation factor. It follows that, with a constant probability, any core-set-induced solution is sub-optimal by at least a factor of $d^{\Theta(d)}$.

Organization. In section 6.3, we formally define spectral (d)-spanners and their generalization "spectral k-spanners". In section 6.4, we introduce our algorithm for finding spectral spanners and prove theorem 6.1. Then, in section 6.5, we generalize the result of theorem 6.1 for spectral k-spanners and show the reduction from $k < d$ to the full dimensional case of $k = d$, proving theorem 6.2. We mention applications of spectral spanners for designing composable core-sets for several optimization problems including k-determinant maximization in section 6.6. In particular we prove theorem 6.3. Finally in section 6.7, we present our lower bound results and prove theorem 6.4.

6.1.4 Related work

As mentioned earlier, multiple papers developed composable core-sets (or similar constructions) when the objective function is equal to the *logarithm* of the volume. In particular, [94] showed that core-sets obtained via a greedy algorithm guarantee an approximation factor of $\min(k, n)$. The approximation ratio can be further improved to a constant if the input points are assigned to set V_i uniformly at random [91, 14]. However, these guarantees do not translate into a multiplicative approximation factor for the volume objective function.[2]

[2]It is possible to show that the greedy method achieves composable core-sets with multiplicative approximation factor of $2^{O(k^2)}$. Since this bound is substantially larger than our bound obtained by spectral spanners, we do not include the proof in this paper.

Core-sets constructions are known for a wide variety of geometric and metric problems, and several algorithms have found practical applications. Some of those constructions are relevant in our context. In particular, core-sets for approximating the directional width [3] have functionality that is similar to weak spanners. However, the aforementioned paper considered this problem for low-dimensional points, and as a result, the core-sets size was exponential in the dimension. Another line of research [1, 61, 26] considered core-sets for maximizing *metric* diversity measures, such as the minimum inter-point distance. Those measures rely only on relationships between pairs of points, and thus have quite different properties from the volume-induced measure considered in this paper.

We also remark that one can consider generalizations of our problem to settings were we want to maximize the volume under additional constraints. Over the last few years several extensions were studied extensively and many new algorithmic ideas were developed [100, 8, 115, 44]. In this paper, we study composable core-sets for the basic version of the determinant maximization problem where no additional constraints are present.

6.2 Preliminaries

6.2.1 Linear Algebra

Throughout the section, all vectors that we consider are column based and sitting in $\mathbb{R}^d$, unless otherwise specified. For a vector v, we use notation $v(i)$ to denote its i_{th} coordinate and use $\|v\|$ to denote its ℓ_2 norm. Vectors $v_1, \ldots, v_k$ are called orthonormal if for any i, $\|v_i\| = 1$, and for any $i \neq j$, $\langle v_i, v_j \rangle = 0$. For a set of vectors V, we let $\langle V \rangle$ denote the linear subspace spanned by vectors of V. We also use $S^{\perp}$ to denote the linear subspace orthogonal to S, for a linear subspace S.

Notation $\langle , \rangle$ is used to denote Frobenius inner product of matrices, for matrices $A, B \in$

$\mathbb{R}^{d\times d}$

$$\langle A, B\rangle = \sum_{i=1}^{d}\sum_{j=1}^{d} A_{i,j}B_{i,j} = \mathrm{tr}(AB^\intercal)$$

where $A_{i,j}$ denotes the entry of matrix A in row i and column j.

Projection Matrices. A matrix $\Pi \in \mathbb{R}^{d\times d}$ is a projection matrix if $\Pi^2 = \Pi$. It is also easy to see that for any $v \in \mathbb{R}^d$,

$$\langle vv^\intercal, \Pi\rangle = v^\intercal \Pi v = \langle \Pi v, \Pi v\rangle = \|\Pi v\|^2.$$

For a linear subspace S, we let Π_S denote the matrix projecting vectors from $\mathbb{R}^d$ onto S which means for any vector v, $(\Pi_S)v$ is the projection of v onto S. If S is k-dimensional and $v_1, \ldots, v_k$ form an arbitrary orthonormal basis of S, then one can see that $\Pi_S = \left(\sum_{i=1}^{k} v_i v_i^\intercal\right)$. We also represent the set of all projection matrices onto k-dimensional subspaces by $\mathcal{P}_k$.

Fact 5. *For any vectors $u, v \in \mathbb{R}^d$ and any projection matrix $\Pi \in \mathbb{R}^{d\times d}$*

$$\langle (u+v)(u+v)^\intercal, \Pi\rangle \le 2\langle uu^\intercal + vv^\intercal, \Pi\rangle.$$

Proof. Let $a = \Pi u$ and $b = \Pi v$. Then since $\Pi^2 = \Pi$, we have $\langle uu^\intercal, \Pi\rangle = \|a\|^2$, $\langle vv^\intercal, \Pi\rangle = \|b\|^2$, and $\langle (u+v)(u+v)^\intercal, \Pi\rangle = \|a+b\|^2$. Now, the assertion is equivalent to

$$\|a+b\|^2 \le 2(\|a\|^2 + \|b\|^2)$$

which follows by Cauchy-Schwarz inequality, $\langle a, b\rangle \le \|a\|\|b\| \le (\|a\|^2 + \|b\|^2)/2$. □

For a symmetric Matrix A, $\lambda_1(A) \ge \ldots \ge \lambda_d(A)$ denotes the eigenvalues of A. We take advantage of the following simple lemma which is also known as extremal partial trace. A proof of it can be found in [116].

Lemma 6.6. *Let $L \in \mathbb{R}^{d\times d}$ be a symmetric matrix. Then for any integer $n \le d$,*

$$\min_{\Pi\in\mathcal{P}_n} \langle \Pi, L\rangle = \sum_{d-n+1}^{d} \lambda_i(L).$$

In particular, we use it to conclude that if $x_1, \ldots, x_n \in \mathbb{R}^d$ are orthonormal vectors, then

$$\sum_{i=1}^{n} x_i^\intercal L x_i \geq \sum_{d-n+1}^{d} \lambda_i(L).$$

For a matrix A, we use $\sigma_1(A) \geq \ldots \geq \sigma_d(A) \geq 0$ to denote singular values of A (for symmetric matrices they are the same as eigenvalues). Given a matrix, we use the following simple lemma to construct a symmetric matrix whose eigenvalues are the singular values of the input matrix and their negations.

Many of the matrices that we work with in this paper are not symmetric. Define a symmetrization operator $\mathcal{S}_d : \mathbb{R}^{d\times d} \to \mathbb{R}^{2d\times 2d}$ where for any matrix $A \in \mathbb{R}^{d\times d}$,

$$\mathcal{S}_d(A) = \begin{bmatrix} 0 & A \\ A^\intercal & 0 \end{bmatrix}.$$

When the dimension is clear in the context, we may drop the subscript d. The following fact is immediate.

Fact 7. *For any matrix $A \in \mathbb{R}^{d\times d}$, $\mathcal{S}(A)$ has eigenvalues $\sigma_1(A) \geq \ldots \geq \sigma_d(A) \geq -\sigma_d(A) \ldots \geq -\sigma_1(A)$.*

Proof. Let $u_1, \ldots, u_d$ and $v_1, \ldots, v_d$ be right and left singular vectors of A, respectively. Then we have $Au_i = \sigma_i v_i$ and $A^\intercal v_i = \sigma_i u_i$ for any $1 \leq i \leq m$. Now, it is easy to see $[v_i \; u_i]$ and $[-v_i \; u_i]$ are eigenvectors for $\mathcal{S}(A)$ with eigenvalues σ_i and $-\sigma_i$ for any $1 \leq i \leq m$. □

Determinant Maximization Problem. We use the notion of *determinant* of a subset of vectors as a measure of their diversity. From a geometric point of view, for a subset of vectors $V = \{v_1, \ldots, v_d\} \subset \mathbb{R}^d$, $\det(\sum_{i=1}^{d} v_i v_i^\intercal)$ is equal to the square of the volume of the parallelepiped spanned by V. For $S, T \subseteq [d]$, Let $A_{S,T}$ denote the $|S| \times |T|$ submatrix formed by intersecting the rows and columns corresponding to S, T respectively. The

notation $\det_k$ is a generalization of determinant and is defined by

$$\det_k(A) = \sum_{S \in \binom{[d]}{k}} \det A_{S,S}.$$

In particular, for vectors $v_1, \ldots, v_k \in \mathbb{R}^d$, $\det_k(\sum_{i=1}^k vv_i^\intercal)$ is equal to the square of the k-dimensional volume of the parallelepiped spanned by $v_1, \ldots, v_k$. The problem of k-determinant maximization is defined as follows.

Definition 6.8 (k-Determinant Maximization)**.** Let $V = \{v_1, \ldots, v_n\} \subset \mathbb{R}^d$ be a set of vectors, and let $M \in \mathbb{R}^{n \times n}$ be the Gram matrix obtained from A, i.e., $M_{i,j} = \langle v_i, v_j \rangle$. For an integer $k \leq d$, the goal of the k-determinant maximization problem is to choose a subset $S \subseteq V$ such that $|S| = k$ and the determinant of $M_{S,S}$ is maximized.

For any $S \subset [n] (|S| = k)$, if we let $V_S \in \mathbb{R}^{k \times d}$ be the matrix with $\{v_i\}_{i \in S}$ as its rows, then we have

$$\det(M_{S,S}) = \det(V_S V_S^\intercal) = \det(V_S^\intercal V_S) = \det_k(\sum_{v \in S} vv^\intercal), \tag{6.5}$$

where the last equality uses the Cauchy-Binet identity. The k-determinant maximization is also known as *maximum volume* k-simplex since $\det_k(\sum_{v \in S} vv^\intercal)$ is equal to the square of the volume spanned by $\{v_i\}_{i \in S}$. We also use the following identity which can be derived from the Cauchy-Binet formula (see fact 5) when the columns of $B \in \mathbb{R}^{d \times n}$ are v_is and $C = B^\intercal$.

$$\det(\sum_{i=1}^n v_i v_i^\intercal) = \sum_{S \in \binom{[n]}{d}} \det(\sum_{i \in S} v_i v_i^\intercal). \tag{6.6}$$

We use it to deduce the following simple lemma

Lemma 6.9. *For any set of vectors $v_1, \ldots, v_n \in \mathbb{R}^d$ and any integer $1 \leq k \leq d$,*

$$\det_k\left(\sum_{i=1}^n v_i v_i^\intercal\right) = \sum_{S \in \binom{[n]}{k}} \det_k\left(\sum_{i \in S} v_i v_i^\intercal\right)$$

Proof. For a set $T \subset [d]$ and any $1 \leq i \leq n$, let $v_{i,T} \in \mathbb{R}^k$ denote the restriction of v_i to its

coordinates in T. The proof can be derived as follows

$$\det_k\left(\sum_{i=1}^{n} v_i v_i^\intercal\right) = \sum_{T\in\binom{[d]}{k}} \det\left(\sum_{i=1}^{n} v_{i,T} v_{i,T}^\intercal\right) \qquad \text{By definition of } \det_k$$

$$= \sum_{T\in\binom{[d]}{k}} \sum_{S\in\binom{[n]}{k}} \det\left(\sum_{i\in S} v_{i,T} v_{i,T}^\intercal\right) \qquad \text{By (6.6)}$$

$$= \sum_{S\in\binom{[n]}{k}} \left(\sum_{T\in\binom{[d]}{k}} \det\left(\sum_{i\in S} v_{i,T} v_{i,T}^\intercal\right)\right) = \sum_{S\in\binom{[n]}{k}} \det_k\left(\sum_{i\in S} v_i v_i^\intercal\right) \qquad \text{By definition of } \det_k$$

□

We also use the following identities about the determinant of matrices. For a $d \times d$ matrix A, we have

$$\det(A) = \prod_{i=1}^{d} \sigma_i(A).$$

If A is lower(upper) triangular, i.e. $A_{i,j} = 0$ for $j > i (j < i)$, we have $\det(A) = \Pi_{i=1}^{d} A_{i,i}$.

6.2.2 *Core-sets*

The notion of core-sets has been introduced in [3]. Informally, a core-set for an optimization problem is a subset of the data with the property that solving the underlying problem on the core-set gives an approximate solution for the original data. This notion is somewhat generic, and many variations of core-sets exist.

The specific notion of *composable core-sets* was explicitly formulated in [61].

Definition 6.10 (α-Composable Core-sets)**.** A function $c(V)$ that maps the input set $V \subset \mathbb{R}^d$ into one of its subsets is called an α-composable core-set for a maximization problem with respect to a function $f\colon 2^{\mathbb{R}^d} \to \mathbb{R}$ if, for any collection of sets $V_1, \cdots, V_m \subset \mathbb{R}^d$, we have

$$f(c(V_1) \cup \cdots \cup c(V_m)) \geq \frac{1}{\alpha} f(V_1 \cup \cdots \cup V_m)$$

For simplicity, we will often refer to the set $c(P)$ as the core-set for P and use the term "core-set function" with respect to $c(\cdot)$. The *size* of $c(\cdot)$ is defined as the smallest number t such that $c(P) \leq t$ for all sets P (assuming it exists). Unless otherwise stated, whenever we use the term "core-set", we mean a composable core-set.

6.3 Spectral Spanners

In this section we introduce the notion of *spectral spanners* and review their properties. In the following, we define the special case of spectral spanners. Later in Definition 6.14, we introduce its generalization, *spectral k-spanners*.

Definition 6.11 (Spectral Spanner)**.** Let $V \subset \mathbb{R}^d$ be a set of vectors. We say $U \subseteq V$ is an α-spectral d-spanner for V if for any $v \in V$, there exists a probability distribution μ_v on the vectors in U so that

$$vv^\intercal \preceq \alpha \cdot \mathbb{E}_{u \sim \mu_v} [uu^\intercal]. \tag{6.7}$$

We study spectral spanners in Section 6.4, and propose polynomial time algorithms for finding $\tilde{O}(d)$-spectral spanners of size d. Considering (6.7) for all $v \in V$ implies that if $U \subseteq V$ is an α-spectral spanner of V, then for any probability distribution $\mu : V \to \mathbb{R}^+$, there exists a distribution $\tilde{\mu} : U \to \mathbb{R}^+$ such that

$$\mathbb{E}_{v \sim \mu} [vv^\intercal] \preceq \alpha \cdot \mathbb{E}_{u \sim \tilde{\mu}} [uu^\intercal]. \tag{6.8}$$

We crucially take advantage of this property in Section 6.6 to develop core-sets for the *experimental design* problem. Let $f : \mathcal{S}_d^+ \to \mathbb{R}^+$ be a monotone function such that $f(A) \leq f(B)$ if $A \preceq B$. Roughly speaking, we use monotonicity of f along (6.8) to reduce optimizing f on the set of all matrices of the form $\mathbb{E}_{v \sim \mu} [vv^\intercal]$ for some distribution μ, to optimizing it on distributions which are only supported on the small set U. A wide range of matrix functions used in practice lie in the category of monotone functions, e.g. determinant, trace. More generally one can see $\lambda_i(.)$ for any i is a monotone function, and consequently the same holds for any elementary symmetric polynomial of the eigen-

values. For polynomial functions of the lower-degree, e.g. trace, $\det_k$, the monotonicity can be guaranteed by weaker constraints. Therefore, one should expect to find smaller core-sets with better guarantees for those functions. Motivated by this, we introduce the notion of spectral k-spanners. Let us first define the notation $\preceq_k$ to generalize $\preceq$.

Definition 6.12 ($\preceq_k$ notation)**.** For two matrices $A, B \in \mathbb{R}^{d\times d}$, we say $A \preceq_k B$ if for any $\Pi \in \mathcal{P}_{d-k+1}$, we have $\langle A, \Pi\rangle \leq \langle B, \Pi\rangle$.

In particular note that $A \preceq_d B$ is equivalent to $A \preceq B$ and $A \preceq_1 B$ is the same as $\mathrm{tr}(A) \leq \mathrm{tr}(B)$, since $\mathcal{P}_1 = \mathbb{R}^d$ and $\mathcal{P}_d = I$. More generally, the following lemma can be used to check if $A \preceq_k B$.

Lemma 6.13. *Let $A, B \in \mathbb{R}^{d\times d}$ be two symmetric matrices. Then $A \preceq_k B$ if and only if $\sum_{i=k}^{d} \lambda_i(B - A) \geq 0$.*

Proof. Suppose that $A \preceq_k B$. Then by definition for any $\Pi \in P_{d-k+1}$, $\langle B - A, \Pi\rangle \geq 0$, so combining with Lemma 6.6, we get

$$0 \leq \min_{\Pi\in\mathcal{P}_{d-k+1}} \langle B - A, \Pi\rangle = \sum_{i=k}^{d} \lambda_i(B - A).$$

The other side can also be verified in the exactly reverse order. □

Now, we are ready to define spectral k-spanners.

Definition 6.14 (Spectral k-Spanner)**.** Let $V \subset \mathbb{R}^d$ be a set of vectors. We say $U \subseteq V$ is an α-spectral k-spanner for V if for any $v \in V$, there exists a probability distribution μ_v on the vectors in U so that

$$vv^\intercal \preceq_k \alpha \cdot \mathbb{E}_{u\sim\mu_v}\left[uu^\intercal\right]. \tag{6.9}$$

We may drop k, whenever it is clear from the context. Finally, we remark that spectral k-spanners have the composability property: If U_1, U_2 are α-spectral spanners of V_1, V_2

respectively, then $U_1 \cup U_2$ is an α-spectral spanner of $V_1 \cup V_2$. This property will be useful to construct composable core-sets.

We will prove the first part of Theorem 6.2 in Section 6.5. The second part of the theorem shows almost optimality of our results: We cannot get better than an $\Omega(d)$-spectral spanner in the worst case unless the spectral spanner has size *exponential* in d. Next, here we prove the second part of the theorem.

First, let us prove the claim for $k = d$. Let V be a set of $\frac{1}{2}e^{d^\epsilon/8}$ independently chosen random ± 1 vectors in $\mathbb{R}^d$. By Azuma-Hoeffding inequality and the union bound, we get that

$$\mathbb{P}\left[\forall u, v \in V : |\langle u, v\rangle| \leq \sqrt{\frac{1}{2}d^{1+\epsilon}}\right] \geq 1 - |V|^2 e^{-d^\epsilon/4} \geq 1/2.$$

So, let V be a set where for all $u, v \in V$, $\langle u, v\rangle^2 \leq \frac{1}{2}d^{1+\epsilon}$. We claim that any $d^{1-\epsilon}$-spectral spanner of V must have all V. Let U be such a spanner and suppose $v \in V$ is not in u. We observe that $vv^\intercal \not\preceq d^{1-\epsilon}\mathbb{E}_{u\sim\mu}\left[uu^\intercal\right]$ for any μ supported on U. This is because for any μ supported on U,

$$\mathbb{E}_{u\sim\mu}\left[\langle v, u\rangle^2\right] \leq \mathbb{E}_{u\sim\mu}\left[\frac{1}{2}d^{1+\epsilon}\right] \leq \frac{1}{2}d^{1+\epsilon} = \frac{1}{2d^{1-\epsilon}}d^2 = \frac{1}{2d^{1-\epsilon}}\langle v, v\rangle^2$$

as desired.

Now, let us extend the above proof to $k < d$. Firstly, we construct a set $V \subseteq \mathbb{R}^k$ of $\frac{1}{2}e^{k^\epsilon/8}$ independently chosen random ± 1 vectors in $\mathbb{R}^k$. By above argument V has no $k^{1-\epsilon}$-spectral k-spanner. Now define $V' \subseteq \mathbb{R}^d$ by appending $d - k$ zeros to each vector in V. It is not hard to see that any α-spectral k-spanner of V is also an α-spectral k-spanner of V'. Therefore, any $k^{1-\epsilon}$-spectral k-spanner of V' has all vectors of V'.

6.4 Spectral Spanners in Full Dimensional Case

In this section we prove theorem 6.2 for the case $k = d$. In this case we have a slightly better bound. So, indeed we will prove theorem 6.1. As alluded to in the introduction we design a greedy algorithm that can be seen as a spectral analogue of the classical greedy

algorithms for finding combinatorial spanners in graphs. The details of our algorithm is in Algorithm algorithm 6.6.

Algorithm 6.6 Spectral d-spanner(V,α): Finds an α-spectral d-spanner

Input: A set of vectors $V \subset \mathbb{R}^d$.

Output: A subset $U \subseteq V$ which is α-spectral d-spanners of V.

1: Let $U = \emptyset$.

2: **repeat**

3: For any $v \in V$, define a polytope

$$P_v = \{x|\ \forall u \in U, \langle x, v\rangle > \sqrt{\alpha}|\langle x, u\rangle|\}$$

.

4: Find a vector v such that P_v is nonempty and let x be any point in P_v.

5: Add $\text{argmax}_{u\in V}\langle u, x\rangle^2$ to U.

6: **until** There exists v such that P_v is nonempty

7: Output U.

Note that for any vector v we can test whether P_v is empty using a linear program. Therefore, the above algorithm runs in time polynomial in $|V|$ and d.

As alluded to in section 6.1.3, we first prove that our algorithm constructs a weak $\tilde{O}(d)$-spectral spanner. Let us recall the definition of weak spectral spanner.

Definition 6.15 (Weak Spectral Spanner)**.** A subset $U \subseteq V \subset \mathbb{R}^d$ is a weak α-spectral spanner of V, if for all $v \in V$ and $x \in \mathbb{R}^d$ there is a probability distribution $\mu_{v,x}$ on U such that

$$\langle v, x\rangle^2 \leq \alpha \cdot \mathbb{E}_{u\sim\mu_{v,x}}\left[\langle u, x\rangle^2\right] \quad \text{equiv} \quad \langle v, x\rangle^2 \leq \alpha \cdot \max_{u\in U}\langle u, x\rangle^2$$

In the rest of this section, we may call spectral spanners *strong* to emphasize its difference

from weak spectral spanners defined above. The rest of this section is organized as follows: In section 6.4.1 we prove that the output of the algorithm is a weak α-spectral spanner of size $O(d \log d)$ for $\alpha = \Omega(d \log^2 d)$. Then, in section 6.4.2 we prove that for any α, any weak α-spectral spanner is a strong α-spectral spanner.

6.4.1 Construction of a Weak Spectral Spanner

In this section we show that algorithm 6.6 returns an α-spectral d-spanner when α is sufficiently larger than d. At the end of this section we discuss an alternative algorithm for finding a weak spanner that achieves slightly better approximation guarantee. However, we believe that this algorithm is simpler to implement, easier to parallelize and can be tuned for practical applications.

Proposition 16. *There is a universal constant $C > 0$ such that for $\alpha \geq C \cdot d \log^2 d$, Algorithm algorithm 6.6 returns a weak α-spectral spanner of size $O(d \log d)$.*

First, we observe that for *any* α, the output of the algorithm is a weak α-spectral spanner. For the sake of contradiction, suppose the output set U is not a weak α-spectral spanner. So, there is a vector $v \in V$ and $x \in \mathbb{R}^d$ such that

$$\langle x, v \rangle^2 > \alpha \cdot \max_{u \in U} \langle x, u \rangle^2 \tag{6.10}$$

We show that P_v is non-empty, which implies U cannot be the output. We can assume $\langle x, v \rangle > 0$, perhaps by multiplying x by a -1. So the above equation is equivalent to $\langle x, v \rangle > \sqrt{\alpha} \cdot \max_{u \in U} |\langle u, x \rangle|$, which implies $x \in P_v$.

It remains to bound the size of the output set U. As alluded to in the introduction, the main technical part of the proof is to show that the rank of lower triangular matrices is robust under small perturbations. To bound the size of U we will construct such a matrix and we will use lemma 6.17 (see below) to bound its rank. Let $u_1, u_2, \ldots, u_m$ be the *sequence* of vectors added to our spectral spanner in the algorithm, i.e., u_i is the i-th vector added to the set U. By Step 2 of the algorithm for any u_i there exists a "bad"

vector $x_i \in \mathbb{R}^d$ such that

$$\langle u_i, x_i \rangle^2 > \alpha \cdot \max_{1 \le j < i} \langle u_j, x_i \rangle^2,$$

Furthermore, by construction, u_i is the vector with largest projection onto x_i, i.e., $u_i = \text{argmax}_{u \in V} \langle x_i, u \rangle^2$. Define inner product matrix $M \in \mathbb{R}^{m \times m}$

$$M_{ij} = \langle u_i, x_j \rangle.$$

By the above conditions on the vectors u_i, x_j, M is diagonally dominant and for all $1 \le i \le m$ and $1 \le j < i$ we have $M_{j,i} \le \frac{M_{i,i}}{\sqrt{\alpha}}$. So the assertion of the lemma 6.17 holds for M and $\epsilon = \frac{1}{\sqrt{\alpha}}$. By the lemma,

$$\text{rank}(M) \ge C \cdot \min \left\{ \frac{4\alpha}{\log^2 \alpha}, \frac{m}{\log m} \right\},$$

where C for some constant $C > 0$. But, it turns out that $\text{rank}(M) \le d$ as it can be written as the product of an $m \times d$ matrix and a $d \times m$ matrix. Setting $\alpha = \frac{d \log^2 d}{C}$, implies $|U| = m \le \frac{2d \log d}{C}$ for large enough d, as desired. It remains to prove the following lemma.

Lemma 6.17. *Let $M \in \mathbb{R}^{m \times m}$ be a diagonally dominant and approximately lower triangular matrix in the following sense*

$$M_{j,i} \le \epsilon \cdot M_{i,i}, \forall\, 1 \le j < i \le m, \tag{6.11}$$

Then, there is a universal constant $C > 0$ such that we have $\text{rank}(M) \ge C \cdot \min \left\{ \left(\frac{1}{\epsilon \log \frac{1}{\epsilon}} \right)^2, \frac{m}{\log m} \right\}$.

Proof. Without loss of generality, perhaps after scaling each column of M by its diagonal entry, we assume $M_{i,i} = 1$ for all i. Note that rank and (6.11) is invariant under scaling, so it is enough to prove the statement for such a matrix. Let M_s denote the top left $s \times s$ principal submatrix of M for some integer $s \le m$ that we specify later. Note that rank is monotonically decreasing under taking principal sub-matrices, so this operations does not increase the rank and showing the assertion of the lemma on $\text{rank}(M_s)$ proves the lemma. Furthermore, (6.11) is closed under taking principal sub-matrices. We can write $M_s = L + E$ such that

- $L \in \mathbb{R}^{s\times s}$ is a lower triangular matrix where $L_{i,i} = 1$ and $\left|L_{i,j}\right| \leq 1$, for any $1 \leq j \leq i \leq s$. In particular, $\|L\|_\infty = 1$.

- $\|E\|_\infty \leq \epsilon$ (note that we may further assume E is upper triangular, but we do not use it in our proof).

Let $\sigma_1(M_s) \geq \ldots \geq \sigma_s(M_s)$ denote singular values of M_s. Obviously, $\sigma_i(M_s) > 0$ implies $\text{rank}(M_s) \geq i$ for any $1 \leq i \leq s$. Considering this fact, let us give some intuition on why M_s has a large rank. Since L is lower-triangular with non-zero entries on the diagonal, it is a full rank matrix. Moreover, entries of E are much smaller than (diagonal) entries of L. Singular values of E are on average much smaller than those of L, so adding E to L can only make a small fraction of singular values of L vanish. This implies that $M_s = L + E$ must have a high rank. Now we make the argument rigorous.

Let $\mathcal{S}(M_s), \mathcal{S}(L), \mathcal{S}(E)$ be the symmetrized versions of M_s, L and E respectively (see section 6.2.1). By fact 7, to show $\sigma_i(M_s) > 0$ for some i, we can equivalently prove $\lambda_i(\mathcal{S}(M_s)) > 0$. We use Lemma lemma 2.3: Setting $A = \mathcal{S}(L)$ and $B = \mathcal{S}(E)$, for any pair of integers $\ell < k \leq s$ such that

$$\lambda_k(\mathcal{S}(L)) + \lambda_{2s-\ell}(\mathcal{S}(E)) > 0 \tag{6.12}$$

we have $\lambda_{k-\ell}(\mathcal{S}(M_s)) > 0$. So to prove the lemma, it suffices to find s, k and ℓ satisfying the above and $k - \ell \geq C \cdot \min\left\{\left(\frac{1}{\epsilon \log \frac{1}{\epsilon}}\right)^2, \frac{m}{\log m}\right\}$ for some constant C.

To find proper values of k and ℓ, we use the following two claims.

Claim 6.18. *For any $\ell \leq s$,*

$$\lambda_{2s-\ell}(\mathcal{S}(E)) \geq \lambda_{2s-\ell+1}(\mathcal{S}(E)) = -\sigma_\ell(E) \geq \frac{-\|E\|_F}{\sqrt{\ell}} \geq \frac{-\epsilon \cdot s}{\sqrt{\ell}}.$$

Claim 6.19. *For any $k < \frac{s}{2}$,*

$$\lambda_k(\mathcal{S}(L)) = \sigma_k(L) \geq \left(\frac{k-1}{s^2}\right)^{\frac{k-1}{s}}.$$

Therefore, to show (6.12) it is enough to show

$$s \log \frac{\sqrt{\ell}}{\epsilon s} > (k-1) \log \frac{s^2}{k-1}, \tag{6.13}$$

for $k, \ell \leq \frac{s}{2}$. We analyze the above in two cases. If $m \log m \leq \frac{1}{\epsilon^2}$, then one can see that for $s = m$, $k = \lfloor \frac{m}{4 \log m} \rfloor$ and $\ell = \lfloor \frac{m}{8 \log m} \rfloor$, and large enough m, (6.13) holds. It implies that in this case $\operatorname{rank}(M_s) \geq k - \ell \geq \frac{m}{8 \log m}$, thus we are done. Now suppose that $\frac{1}{\epsilon^2} \leq m \log m$. We set $s < m$ to be the largest integer such that $s \log s \leq \frac{1}{16\epsilon^2}$. Next, we let $\ell = \lfloor 4\epsilon^2 s^2 \rfloor$. Note that $s \log s \leq \frac{1}{16\epsilon^2}$ implies $\ell \leq \frac{s}{4 \log s}$. Now applying $\ell = \lfloor 4\epsilon^2 s^2 \rfloor$ into (6.13) turns it into

$$s > (k-1) \log \frac{s^2}{k-1}.$$

So, for $k = \lfloor \frac{s}{2 \log s} \rfloor$, the above is satisfied. Furthermore, in this case $k - \ell = \lfloor \frac{s}{2 \log s} \rfloor - \lfloor 4\epsilon^2 s^2 \rfloor \geq \frac{s}{4 \log s} \geq \frac{1}{256 \epsilon^2 \log^2 \frac{1}{\epsilon}}$, as s is the largest number such that $s \log s \leq \frac{1}{16\epsilon^2}$ and $\log s \leq 2 \log \frac{1}{\epsilon}$. So the lemma holds for $C \geq \frac{1}{256}$. □

Proof of Claim 6.18. By Fact fact 8 we know that

$$\sum_{i=1}^{s} \sigma_i(E)^2 = \|E\|_F^2 \leq \|E\|_\infty^2 \cdot s^2 \leq \epsilon^2 \cdot s^2.$$

Now, by Markov inequality we get $\sigma_\ell(E)^2 \leq \frac{\epsilon^2 s^2}{\ell}$. Therefore, the claim is proved. □

Proof of Claim 6.19. Since L is lower-triangular, we have that

$$\prod_{i=1}^{s} \sigma_i(L) = \det L = \prod_{i=1}^{s} L_{i,i} = 1, \tag{6.14}$$

It follows that for any $k \leq s$,

$$\prod_{i=1}^{k-1} \sigma_i(L) = \frac{1}{\prod_{j=k}^{s} \sigma_j(L)} \geq \frac{1}{\sigma_k(L)^{s-k+1}}. \tag{6.15}$$

Now, we use the Frobenius norm to prove an upper bound on the first $k-1$ singular values. By Fact fact 8,

$$\sum_{i=1}^{s} \sigma_i(L)^2 = \|L\|_F^2 \leq \|L\|_\infty \cdot s^2 = s^2, \tag{6.16}$$

By AM-GM inequality we get

$$\prod_{i=1}^{k-1} \sigma_i(L) \leq \left(\frac{\sum_{i=1}^{k-1} \sigma_i(L)^2}{k-1} \right)^{\frac{k-1}{2}} \leq \left(\frac{\|L\|_F^2}{k-1} \right)^{\frac{k-1}{2}} \leq \left(\frac{s^2}{k-1} \right)^{\frac{k-1}{2}}.$$

The above together with (6.15) proves $\sigma_k(L) \geq \left(\frac{k-1}{s^2}\right)^{\frac{k-1}{2(s-k+1)}}$. Noting that for $k \leq \frac{s}{2}$, $2(s-k+1) \geq s$ completes the proof of the claim. □

We would like to thank an anonymous reviewer for suggesting an alternative algorithm for finding weak spanners. It offers slightly better guarantees, and finds a weak d-spectral spanner of size $O(d \log d)$. However, as we argue, our algorithm can be made more efficient in practice, and in particular in a distributed setting.

An alternative Algorithm. For a subset $V \in \mathbb{R}^d$, define $\mathrm{sym}(V)$ to be the symmetric set $\mathrm{sym}(V) = V \cup \{-x | x \in V\}$. A direct application of the separating hyperplane theorem shows that a subset $U \subseteq V$ is a weak α-spectral spanner of V, if $\mathrm{conv}(\mathrm{sym}(V)) \subseteq \sqrt{\alpha} \cdot \mathrm{conv}(\mathrm{sym}(U))$ where conv refers to the convex hull of the set. Knowing this, we can apply the celebrated result of F. John [13] to get a weak d-spectral spanner. Letting the notation MVEE of a set denote the minimum volume ellipsoid enclosing the set, it implies that there exists a subset $U \subset V$ of size $O(d^2)$ and an ellipsoid E where $E = \mathrm{MVEE}(\mathrm{sym}(U)) = \mathrm{MVEE}(\mathrm{sym}(V))$ and $\frac{E}{\sqrt{d}} \subseteq \mathrm{conv}(\mathrm{sym}(U))$. Therefore, U is a weak d-spectral spanner of V with size $O(d^2)$. Moreover, the size of U can be reduced by using the result of [110] on approximating John's ellipsoids. In our case, it implies choosing a random set of $O(d \log d)$ points of U gives a weak d-spectral spanner of V with high probability.

Although, the approximation guarantee can be improved by a log factor in this algorithm, this improvement comes at a cost. First of all, finding the John's ellipsoid requires solving a semidefinite program with $O(d^2)$ variables whereas in Algorithm algorithm 6.6, we only need to solve linear programs with $O(d)$ many variables. This requires polynomially smaller amount of memory. Furthermore, note that the main computational task

of each step of algorithm 6.6 is to solve $|V|$ feasibility LPs where each of them has $\tilde{O}(d)$ variables and constraints. These LPs can be solved in parallel: having access to $O(|V|)$ many processors, our greedy algorithm runs in poly(d) time in PRAM model of computation. This extreme parallelism cannot be achieved using the above approach. Finally, in finding the weak spanner one can *tune* the value of α in (6.1) based on the structure of the given data points and the ideal size of the core-set, making the algorithm more suitable for applications.

6.4.2 *From Weak Spectral Spanners to Strong Spectral Spanners*

In this section, we prove that if U is a weak α-spectral spanner of V, then it is a strong α-spectral spanner of V. Combining with proposition 16 it proves theorem 6.1.

Lemma 6.20. *For any set of vectors $V \subset \mathbb{R}^d$, any weak α-spectral spanner of V is a strong α-spectral spanner of V.*

Proof. Let U be a weak α-spectral spanner of V. Fix a vector $v \in V$, we write a program to find a probability distribution $\mu_v : U \to \mathbb{R}^+$ such that $vv^T \preceq \frac{1}{\delta} \cdot \mathbb{E}_{u\sim\mu_v}\left[uu^T\right]$, for the largest possible δ. It turns out that this is a semi-definite program, where we have a variable $p_u = \mathbb{P}_{\mu_v}(u)$ to denote the probability of each vector $u \in U$, see (6.17) for details.

$$\begin{aligned} \max \quad & \delta \\ \text{s.t} \quad & \delta \cdot vv^T \preceq \mathbb{E}_{u\sim\mu_v}\left[uu^T\right] \\ & \mu_v \text{ is a distribution on } U \end{aligned} \tag{6.17}$$

To prove the lemma, it suffices to show the optimal of the program is at least $\frac{1}{\alpha}$. To do that, we analyze the dual of the program. We first show the set of feasible solutions of the program has a non-empty interior; this implies that the Slater condition is satisfied, and the duality gap is zero. Then we show any solution of the dual has value at least $1/\alpha$.

To see the first assertion, we let μ_v be equal to the uniform distribution on U and $\delta \leq \frac{1}{\alpha|U|}$. It is not hard to see that this is a feasible solution of the program since U is a weak α-spectral spanner.

Next, we prove the second statement. First we write down the dual.

$$\begin{aligned} \min \quad & \lambda \\ \text{s.t.} \quad & u^T X u \leq \lambda, \, \forall u \in U \\ & v^T X v \geq 1 \\ & X \succeq 0 \end{aligned}$$

Let (X, λ) be a feasible solution of the dual. Our goal is to show $\lambda \geq \frac{1}{\alpha}$. Let $E = \{x \in \mathbb{R}^d \mid x^T X x \leq \lambda\}$ be an ellipsoid of radius $\sqrt{\lambda}$ defined by X. The set E has the following properties:

- Convexity,
- Symmetry: If $x \in E$, then $-x \in E$,
- $U \subseteq E$: By the dual constraints $u^\intercal X u \leq \lambda$ for all $u \in U$.

Let $\bar{v} = v/\sqrt{\alpha}$. We claim that $\bar{v} \in E$. Note that if $\bar{v} \in E$ we obtain

$$\lambda \geq \bar{v}^\intercal X \bar{v} \geq \frac{1}{\alpha},$$

which completes the proof.

For the sake of contradiction suppose $\bar{v} \notin E$. We show that U is not a weak α-spectral spanner. By convexity of E there is a hyperplane separating $\bar{v}$ from E. So there is a vector $e \in \mathbb{R}^d$ such that

$$\langle v, e \rangle = \sqrt{\alpha} \cdot \langle \bar{v}, e \rangle \geq \sqrt{\alpha} \qquad \text{and} \qquad \forall x \in E, \, \langle x, e \rangle < 1.$$

Moreover, by symmetry of E, for any $x \in E$,

$$\langle x, e \rangle^2 \leq \max\{\langle x, e \rangle, \langle -x, e \rangle\}^2 < 1$$

Finally, since $U \subset E$, we obtain $\max_{u \in U} \langle u, e \rangle^2 < 1$. Therefore, $\langle v, e \rangle^2 \not\leq \alpha \max_{u \in U} \langle u, e \rangle^2$

which implies U is not a weak α-spectral spanner. □

6.5 *Construction of Spectral k-Spanners*

In this section we extend our proof on spectral d-spanner to spectral k-spanners for $k < d$, this proves our main theorem 6.2. Here is our high-level plan of proof: First we use the greedy algorithm of [31] for volume maximization to find an $\tilde{O}(k)$-dimensional linear subspace of $\mathbb{R}^d$ onto which input vectors have a "large" projection. Next, we apply theorem 6.1 to this $\tilde{O}(k)$-dimensional space to obtain the desired spectral k-spanner.

6.5.1 *Greedy Algorithm for Volume Maximization*

In this subsection, we prove the following statement.

Proposition 21. *For any set of vectors $V \subset \mathbb{R}^d$, and any $k < d$ and $m > 2k$, there is a set $U \subseteq V$ of size m such that for all $v \in V$ we have*

$$v_{U^\perp} v_{U^\perp}^\intercal \preceq_k 2m^{(\frac{2k}{m})} \cdot \mathbb{E}_{u\sim\mu}\left[uu^\intercal\right], \tag{6.18}$$

where $v_{U^\perp} = \Pi_{\langle U\rangle^\perp}(v)$ is the projection of v on the space orthogonal to the span of U and μ is the uniform distribution on the set U.

As we will see in the next subsection, for $m = \Theta(k \log k)$, the set U promised above will be a part of our spectral k-spanner. Roughly speaking, to obtain a spectral spanner of V, it is enough to additionally add a spectral spanner of $\{\Pi_{\langle U\rangle}(v)\}_{v\in V}$. In the next subsection, will use theorem 6.1 for the latter part.

First, we will describe an algorithm to find the set U promised in the proposition. Then, we will prove the correctness. We use the greedy algorithm of [31] for volume maximization to find the set U. The algorithm is described as algorithm 6.7. Given a set of vectors $V \subset \mathbb{R}^d$ and an integer $t \leq d$, the goal of this greedy algorithm is to find a subset $S(|S| = t)$ of vectors such that the volume of the t- dimensional parallelepiped spanned

by them is approximately maximized among all subsets of size t [3]. Let $U = \{u_1, \ldots, u_m\}$

Algorithm 6.7 Volume Maximization(V,m)

Let $U = \emptyset$
while $|U| < m$ **do**
    add $\text{argmax}_{v \in V} \|\Pi_{\langle U \rangle^\perp}(v)\|$ to U
end while
return U.

be the output of the algorithm and suppose u_i is the i-th vector added to the set, and μ be a uniform distribution on U. Fix a vector $v \in V$ for which we will verify the assertion of the proposition. Note that if $v \in U$ the statement obviously holds. So, assume $v \notin U$.

Fix a $\Pi \in \Pi_{d-k+1}$. Observe that $\langle v_{U^\perp} v_{U^\perp}^\intercal, \Pi \rangle \leq \|\Pi_{\langle U \rangle^\perp}(v)\|^2$. On the other hand, by lemma 6.6, $\left\langle \mathbb{E}_{u \sim \mu}\left[uu^\intercal\right], \Pi \right\rangle \geq \sum_{i=k}^{d} \lambda_i$ where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$ are eigenvalues of $\mathbb{E}_{u \sim \mu}\left[uu^\intercal\right]$. Therefore, to prove (6.18), it suffices to prove

$$\|\Pi_{\langle U \rangle^\perp}(v)\|^2 \leq 2m^{\frac{2k}{m}} \cdot \sum_{i=k}^{d} \lambda_i. \tag{6.19}$$

Define $\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_m$ to be an orthonormal basis of $\langle U \rangle$ obtained by the Gram-Schmidt process on $u_1, \ldots, u_m$, i.e., $\hat{u}_1 = \frac{u_1}{\|u_1\|}$, $\hat{u}_2 = \frac{\Pi_{\langle u_1 \rangle^\perp}(u_2)}{\|\Pi_{\langle u_1 \rangle^\perp}(u_2)\|}$ and so on. Define $M \in \mathbb{R}^m$ to be a matrix where the i_{th} column is the representation of u_i in the orthonormal basis formed by $\{\hat{u}_1, \ldots, \hat{u}_m\}$, i.e., for all $1 \leq i, j \leq m$,

$$M_{i,j} = \langle u_j, \hat{u}_i \rangle.$$

Note that $\mathbb{E}_{u \sim \mu}\left[uu^\intercal\right]$ is the same as $\frac{1}{m} MM^\intercal$ up to a rotation of the space. In other words, both matrices have the same set of non-zero eigenvalues. Since eigenvalues of $\frac{1}{m} MM^\intercal$ are the squares of the singular values of $\frac{1}{\sqrt{m}} M$, to prove (6.19) it is enough to show

$$\|\Pi_{\langle U \rangle^\perp}(v)\|^2 \leq 2m^{\frac{2k}{m}} \cdot \sum_{i=k}^{m} \sigma_i^2(m^{-1/2} M). \tag{6.20}$$

[3] The approximation factor is roughly $\frac{1}{t!}$

Since $v \in V$ and $v \notin U$ we get $\|\Pi_{\langle U\rangle^\perp}(v)\|^2 \leq \|\Pi_{\langle \hat{u}_1,\ldots,\hat{u}_{m-1}\rangle^\perp}(u_m)\|^2 = M_{m,m}^2$. So to prove (6.20), it suffices to show

$$M_{m,m}^2 \leq 2m^{\frac{2k}{m}} \cdot \sum_{i=k}^{m} \sigma_i^2(m^{-1/2}M) \tag{6.21}$$

Note that the above inequality can be seen just as a property of the matrix M. First, let us discuss properties of M that we will use to prove the above:

I) M is upper-triangular as $u_i \in \langle \hat{u}_1, \ldots, \hat{u}_i\rangle$.

II) By description of the algorithm, for any $i < j \leq m$ we have

$$M_{i,i}^2 = \|\Pi_{\langle \hat{u}_1,\ldots,\hat{u}_{i-1}\rangle^\perp}(u_i)\|^2 \geq \|\Pi_{\langle \hat{u}_1,\ldots,\hat{u}_{i-1}\rangle^\perp}(u_j)\|^2 = \sum_{\ell=i}^{j} M_{\ell,j}^2 \tag{6.22}$$

The following lemma completes the proof of proposition 21.

Lemma 6.22. *Let $M \in \mathbb{R}^{m\times m}$ satisfying (i) and (ii). For any $k < m/2$, we have*

$$M_{m,m}^2 \leq 2m^{\frac{2k}{m}} \sum_{i=k}^{m} \frac{1}{m}\sigma_i^2(M).$$

Proof. Here is the main idea of the proof. First, we use Cauchy-Interlacing theorem along with property (ii) to deduce σ_i cannot be much larger than $M_{i,i}$. Then, we combine it with the fact that M is upper triangular and so $\det(M) = \prod_{i=1}^m M_{i,i} = \prod_{i=1}^m \sigma_i$, to upper-bound $M_{m,m}^2$ by a multiple of $\sum_{i=k}^m \sigma_i^2(M)$.

First, we show for all $1 \leq i \leq m$,

$$\sigma_i^2(M) \leq (m-i+1)M_{i,i}^2. \tag{6.23}$$

Define M_i to be the $(m-i+1)\times(m-i+1)$ matrix obtained by removing the first $i-1$ rows and columns of M. First, Cauchy interlacing theorem tells us $\sigma_i(M) \leq \sigma_1(M_i)$. Secondly, by Fact 8 and property (ii) we have

$$\sigma_1(M_i)^2 \leq \sum_{j=1}^{m-i+1} \sigma_j(M_i)^2 = \|M_i\|_F^2 \leq (m-i+1)M_{i,i}^2.$$

This proves (6.23). Since M is upper-triangular,

$$\det(M)^2 = \prod_{i=1}^{m} M_{i,i}^2 = \prod_{i=1}^{m} \sigma_i^2(M) \leq \left(\frac{\sum_{j=k}^{m} \sigma_j(M)^2}{m-k+1}\right)^{m-k+1} \prod_{i=1}^{k-1} \sigma_i^2 =: \beta \prod_{i=1}^{k-1} \sigma_i^2$$

where the inequality follows by the AM-GM inequality and $\beta = (\frac{\sum_{j=k}^{m} \sigma_j(M)^2}{m-k+1})^{m-k+1}$. By (6.23),

$$\prod_{i=1}^{m} \sigma_i^2(M) \leq \beta \prod_{i=1}^{k-1} (m-i+1) M_{i,i}^2 \leq m^k \beta \prod_{i=1}^{k-1} M_{i,i}^2. \tag{6.24}$$

Using $\prod_{i=1}^{m} \sigma_i^2 = \prod_{i=1}^{m} M_{i,i}^2$ again, we get

$$\prod_{i=k}^{m} M_{i,i}^2 \leq m^k \beta.$$

Using property (ii) again, we have $M_{i,i}^2 \geq M_{m,m}^2$ for all i. Therefore, $\prod_{i=k}^{m} M_{i,i}^2 \geq M_{m,m}^{2(m-k+1)}$, we get

$$M_{m,m}^{2(m-k+1)} \leq m^k \beta$$

The lemma follows by raising both sides to $1/(m-k+1)$ and using that $m-k+1 \geq m/2$ since $k < m/2$. □

6.5.2 *Main algorithm*

In this section we prove Theorem 6.2. The details of our algorithm are described in algorithm 6.8.

First of all let us analyze the size of the output. By definition, algorithm 6.7 returns m vectors. Then, by theorem 6.1, algorithm 6.6 has size at most $O(m \log m)$. Since $m = O(k \log k)$, the size of the output is $|U| + |W| \leq O(k \log^2 k)$, as desired.

In the rest of this section we prove the correctness. Fix a vector $v \in V$, we need to find a distribution μ_v on $U \cup W$ such that $vv^\intercal \preceq_k \alpha \mathbb{E}_{u \sim \mu_v} [uu^\intercal]$ for some $\alpha = \tilde{O}(k)$.

First, by fact 5,

$$vv^\intercal \preceq_k 2(\Pi_{\langle U \rangle^\perp}(v)\Pi_{\langle U \rangle^\perp}(v)^\intercal + \Pi_{\langle U \rangle}(v)\Pi_{\langle U \rangle}(v)^\intercal)$$

Algorithm 6.8 Spectral-k-Spanner(V,α,k): Finds an α-Spectral k-Spanner

Input: A set of vectors $V \subset \mathbb{R}^d$, a parameter α and an integer $k \leq d$.

Output: A α-spectral k-spanner of V.

1: Set m, such that $m^{\frac{2k}{m}} = O(1)$.

2: Run Volume-Maximization(V,m) of algorithm 6.7 and let U be the output, i.e., the set of vectors satisfying (6.18).

3: Run Spectral-d-Spanner($\{\Pi_{\langle U\rangle}(v)\}_{v\in V}$,$\alpha$) of algorithm 6.6 and let W be the output of the corresponding spectral m-spanner.

return $U \cup \{v : \Pi_{\langle U\rangle}(v) \in W\}$.

So, it is enough to prove that

$$\Pi_{\langle U\rangle^\perp}(v)\Pi_{\langle U\rangle^\perp}(v)^\intercal + \Pi_{\langle U\rangle}(v)\Pi_{\langle U\rangle}(v)^\intercal \preceq_k (\alpha/2)\mathbb{E}_{u\sim\mu_v}\left[uu^\intercal\right] \tag{6.25}$$

We proceed by upper-bounding the LHS term by term. By proposition 21,

$$\Pi_{\langle U\rangle^\perp}(v)\Pi_{\langle U\rangle^\perp}(v)^\intercal \preceq_k O(1)\cdot\mathbb{E}_{u\sim\mu}\left[uu^\intercal\right] \tag{6.26}$$

where μ is the uniform distribution on U. So, to prove the above, it is enough to find a distribution μ_v on $U \cup W$ such that

$$\Pi_{\langle U\rangle}(v)\Pi_{\langle U\rangle}(v)^\intercal \preceq_k \alpha\mathbb{E}_{u\sim\mu_v}\left[uu^\intercal\right] \tag{6.27}$$

for some $\alpha = \tilde{O}(k)$. From now on, for any vector $v \in V$ we use $\hat{v}$ to denote $\Pi_{\langle U\rangle}(v)$.

By description of the algorithm, $\{\hat{v}\}_{v\in W}$ is an $O(m\log^2 m)$-spectral m-spanner for $\{\hat{v}\}_{v\in V}$. So, there exists a probability distribution ν_v on W such that

$$\hat{v}\hat{v}^\intercal \preceq O(m\log^2 m))\mathbb{E}_{w\sim\nu_v}\left[\hat{w}\hat{w}^\intercal\right] \quad \text{equiv} \quad \forall x \in \langle U\rangle : \langle x,\hat{v}\rangle^2 \leq O(m\log^2 m))\cdot\mathbb{E}_{w\sim\nu_v}\left[\langle\hat{w},x\rangle^2\right] \tag{6.28}$$

In fact the above holds for any $x \in \mathbb{R}^d$, as $\langle x,\hat{u}\rangle = \langle\Pi_{\langle U\rangle}(x),\hat{u}\rangle$ for any vector $u \in V$. Therefore, for any $\Pi \in \Pi_{d-k+1}$, by summing (6.28) up over an orthonormal basis of Π and noting $m = O(k\log k)$, we get

$$\langle\Pi,\hat{v}\hat{v}^\intercal\rangle \leq \tilde{O}(k)\cdot\langle\mathbb{E}_{w\sim\nu_v}\left[\hat{w}\hat{w}^\intercal,\Pi\right]\rangle,$$

which by definition implies

$$\hat{v}\hat{v}^\intercal \preceq_k \tilde{O}(k) \cdot \mathbb{E}_{w\sim\nu_v}\left[\hat{w}\hat{w}^\intercal\right]. \tag{6.29}$$

Therefore, to show (6.27) for $\alpha = \tilde{O}(k)$ it suffices to find a distribution μ_v on $U \cup W$ such that

$$\mathbb{E}_{w\sim\nu_v}\left[\hat{w}\hat{w}^\intercal\right] \preceq_k O(1) \cdot \mathbb{E}_{u\sim\mu_v}\left[uu^\intercal\right].$$

But, observe that for any $w \in W$, we can write

$$\hat{w}\hat{w}^\intercal \preceq_k 2\left(ww^\intercal + \Pi_{\langle U\rangle^\perp}(w)\Pi_{\langle U\rangle^\perp}(w)^\intercal\right) \preceq_k O(1)\cdot(ww^\intercal + \mathbb{E}_{u\sim\mu}\left[uu^\intercal\right])$$

where μ is the uniform distribution over U. The first inequality follows by fact 5 and the second inequality follows by equation (6.26) which holds for all vectors $v \in V$. Averaging out the above inequality with respect to the distribution ν_v completes the proof.

6.6 Applications

In this section we discuss applications of theorem 6.2 in designing composable core-sets. As we discussed in the intro, we show that for many problems spectral spanners provide almost the best possible composable core-set in the worst case. Next, we see that for any function f that is "monotone" on PSD matrices, spectral spanners provide a composable core-set for a *fractional* budgeted minimization problem with respect to f. Later, in section 6.6.1 and section 6.6.2 we see that for a large class of monotone functions the optimum of the fractional budgeted minimization problem is within a small factor of the optimum of the corresponding integral problem. So, spectral spanners provide almost optimal composable core-sets for several spectral budgeted minimization problems.

Let $V \subset \mathbb{R}^d$ be a set of vectors. For a function $f : \mathbb{S}_d^+ \to \mathbb{R}^+$ on PSD matrices and a positive integer B denoting the budget, the *fractional budgeted minimization* problem is to choose a mass B of the vectors of V, i.e., $\{s_v\}_{v\in V}$ where $\sum_v s_v \leq B$, such that $f(\sum_v s_v vv^T)$ is minimized. This can be modeled as a continuous optimization problem, see BM for details.

$$\begin{aligned} \inf \quad & f\left(\sum_{v\in V} s_v vv^T\right). \\ \text{s.t} \quad & \sum_{v\in V} s_v \le B \\ & s_v \ge 0,\ \forall v \in V \end{aligned}$$

BM

Definition 6.23 (k-monotone functions)**.** We say a function $f : \mathbb{S}_d^+ \to \mathbb{R}^+$ is *k-monotone* for an integer $1 \le k \le d$, if for all PSD matrices $A, B \succeq 0$, we have $A \preceq_k B$ implies $f(A) \ge f(B)$.

We say f is *vector k-monotone* if for all PSD matrices A, B and all vectors $v \in \mathbb{R}^d$, if $vv^\intercal \preceq_k B$, then $f(A + vv^\intercal) \ge f(A + B)$. Note that any k-monotone function is obviously vector k-monotone as $A + vv^\intercal \preceq_k A + B$.

We show that an algorithm for finding α-spectral k-spanners give an α-composable core-set function for the fractional budgeted minimization *for any* function f that is vector k-monotone. We emphasize that our composable core-set *does not* depend on the choice of f as long as it is vector monotone.

Proposition 24. *For any $1 \le k \le d$ and any vector k-monotone function $f : \mathbb{S}_d^+ \to \mathbb{R}^+$, algorithm 6.8 gives a $\beta(f, \tilde{O}(k))$-composable core-set of size $\tilde{O}(k)$ for the fractional budgeted minimization problem, BM(V,f,B), where for any $t > 0$,*

$$\beta(f, t) = \sup_{A \in \mathbb{S}_d^+} \frac{f(A)}{f(tA)}.$$

Proof. Let $V_1, V_2, \ldots, V_p$ be p given input sets for an arbitrary integer p, and let $\bigcup_{i=1}^p V_i = V$. For each $1 \le i \le p$, let U_i be the output of Spectral k-Spanner(V_i,k,α). By theorem 6.2, for $\alpha = \tilde{O}(k)$, $|U_i| \le \tilde{O}(k)$. Let $U = U_1 \cup \cdots \cup U_p$.

Fix a k-monotone function f and a budget $B > 0$ and let $\mathbf{s} = \{s_v\}_{v \in V}$ be a feasible

solution of BM(V, f, B). To prove the assertion we need to show that there exists a feasible solution $\tilde{\mathbf{s}}$ of BM(U, f, B) such that

$$f\left(\sum_{u\in U}\tilde{s}_u uu^T\right) \leq \beta(f,\alpha)\cdot f\left(\sum_{v\in V} s_v vv^T\right) \tag{6.30}$$

By composability property of spanners, U is an α-spectral k-spanner of V. Therefore, for any $v \in V$, there exists a probability distribution μ_v on U such that $vv^T \preceq_k \alpha \cdot \mathbb{E}_{u\sim\mu_v}[uu^\intercal]$. Say $V = \{v_1, \ldots, v_m\}$. It follows by vector k-monotonicity of f and by U bein an α-spectral k-spanner that

$$\begin{aligned} f\left(\sum_{i=1}^m s_{v_i} v_i v_i^\intercal\right) &\geq f\left(\alpha s_{v_1}\mathbb{E}_{u\sim\mu_{v_1}}[uu^\intercal] + \sum_{i=2}^m s_{v_i} v_i v_i^\intercal\right) \\ &\geq f\left(\sum_{i=1}^2 \alpha s_{v_i}\mathbb{E}_{u\sim\mu_{v_i}}[uu^\intercal] + \sum_{i=3}^m s_{v_i} v_i v_i^\intercal\right) \geq \cdots \geq f\left(\sum_{i=1}^m \alpha s_{v_i}\mathbb{E}_{u\sim\mu_{v_i}}[uu^\intercal]\right) \end{aligned}$$

Now, define $\tilde{\mathbf{s}}$ by $\tilde{s}_u = \sum_{v\in V} s_v \mathbb{P}_{\mu_v}[u]$ for any $u \in U$. It is straight-forward to see that this is a feasible solution of BM(V, f, B) since $\sum_{u\in U}\tilde{s}_u = \sum_{v\in V}\sum_{u\in U}\mathbb{P}_{\mu_v}[u] = \sum_{v\in V} s_v \leq B$. Therefore,

$$f\left(\sum_{v\in V} s_v vv^\intercal\right) \geq f\left(\sum_{v\in V}\alpha s_v \mathbb{E}_{u\sim\mu_v}[uu^\intercal]\right) = f\left(\sum_{u\in U}\alpha\tilde{s}_u uu^\intercal\right) \geq \beta(f,\alpha) f\left(\sum_{u\in U}\tilde{s}_u uu^\intercal\right),$$

This proves (6.30) as desired. □

In general, we may not solve BM efficiently if f is not a convex function. It turns out that if f is convex and has a certain reciprocal multiplicity property then the integrality gap of the program is small, so assuming further that f is (vector) k-monotone, by the above theorem we obtain a composable core-set for the corresponding integral budgeted minimization problems. In the next two sections we explain two such families of functions namely determinant maximization and optimal design.

6.6.1 Determinant Maximization

In this section, we use proposition 24 to prove theorem 6.3. Throughout this section, for an integer $1 \leq k \leq d$ we let $f : \mathbb{S}_d^+ \to \mathbb{R}^+$ be the map $A \mapsto -\det_k(A)^{1/k}$. It follows from theory of Hyperbolic polynomials that for any $1 \leq k \leq d$, $-\det_k(A)^{1/k}$ is a convex

function [53], so one can solve BM($-\det_k^{1/k}, V, k$) using convex programming. Furthermore, observe that BM($-\det_k^{1/k}, V, k$) gives a relaxation of k-determinant maximization problem. Nikolov [99] showed that any fractional solution can be rounded to an integral solution incurring only a multiplicative error of e.

Theorem 6.25 ([99]). *There is a randomized algorithm that for any set $V \subseteq \mathbb{R}^d$, $1 \leq k \leq d$, and any feasible solution x of BM($-\det_k^{1/k}, V, k$) outputs $S \subset V$ of size k such that*

$$\det_k\left(\sum_{v\in S} vv^\intercal\right) \geq e^{-k} \max_{T\in\binom{V}{k}} \det_k\left(\sum_{u\in T} uu^\intercal\right) \tag{6.31}$$

Proof. We include the proof for the sake of completeness. First, we explain the algorithm: For $1 \leq i \leq k$, choose a vector v with probability $\frac{x_v}{k}$ (with replacement) and call it u_i. It follows that,

$$\mathbb{E}\left[\det_k\left(\sum_{i=1}^k u_i u_i^\intercal\right)\right] = \sum_{S\in\binom{V}{k}} \frac{k!}{k^k}\Pi_{v\in S}x_v \det_k(\sum_{v\in S} vv^\intercal) \geq e^{-k}\cdot \sum_{S\in\binom{V}{k}} \det_k\left(\sum_{v\in S} x_v vv^\intercal\right)$$

where first equality holds, since we have $k!$ different orderings for selecting a fixed set S of k vectors, but by Cauchy-Binet identity the RHS is equal to $e^{-k}\det_k(\sum_{v\in V} x_v vv^\intercal)$ as desired. □

Note that the algorithm we discussed in the above proof may have an exponentially small probability of success but [99] also gives a de-randomization using the conditional expectation method. From now on, we do not need convexity. To use proposition 24 we need to verify that $-\det_k^{1/k}$ is (vector) k-monotone.

Lemma 6.26. *For any integer $1 \leq k \leq d$, the function $-\det_k^{1/k}$ is vector k-monotone.*

Proof. Equivalently, we show $-det_k$ is vector k-monotone. Fix $A \succeq 0$, and decompose it as $A = \sum_{a\in A} aa^\intercal$ where we abuse notation and also use A to denote the set of vectors in the decomposition of A. Also, fix a vector v and suppose $vv^\intercal \preceq_k B$ for some $B \succeq 0$. We

need to show $\det_k(A+vv^\intercal) \leq \det_k(A+B)$. By lemma 6.9,

$$\begin{aligned}\det_k(A+vv^\intercal) - \det_k(A) &= \sum_{S\in\binom{A}{k-1}} \det_k\left(\sum_{a\in S} aa^\intercal + vv^\intercal\right)\\ &= \sum_{S\in\binom{A}{k-1}} \det_{k-1}\left(\sum_{a\in S} aa^\intercal\right)\langle vv^\intercal, \Pi_{\langle S\rangle^\perp}\rangle.\end{aligned}$$

The second equality follows by the fact that $\det_k(\sum_{a\in S} aa^\intercal + vv^\intercal)$ is the same as the determinant of the $k\times k$ inner product matrix of all of these k vectors. Using Gram-Schmidt orthogonalization process the latter can be re-written as $\det_{k-1}(\sum_{a\in S} aa^\intercal)\langle vv^\intercal, \Pi_{\langle S\rangle^\perp}\rangle$. Since $vv^\intercal \preceq_k B$ for any such S we have $\langle vv^\intercal, \Pi_{\langle S\rangle^\perp}\rangle \leq \langle B, \Pi_{\langle S\rangle^\perp}\rangle$. Therefore,

$$\begin{aligned}\det_k(A+vv^T) &= \det_k(A) + \sum_{S\in\binom{A}{k-1}} \det_{k-1}\left(\sum_{a\in S} aa^\intercal\right)\langle vv^\intercal, \Pi_{\langle S\rangle^\perp}\rangle\\ &\leq \det_k(A) + \sum_{S\in\binom{A}{k-1}} \det_{k-1}\left(\sum_{a\in S} aa^\intercal\right)\langle B, \Pi_{\langle S\rangle^\perp}\rangle \leq \det_k(A+B).\end{aligned}$$

The last inequality follows by another application of Cauchy-Binet identity, lemma 6.9.

□

Now, we are ready to prove theorem 6.3. Let $V\subseteq \mathbb{R}^d$ and suppose we are given p subsets $V_1,\ldots,V_p$ such that $\bigcup_{i=1}^p V_i = V$. First, by proposition 24, spectral spanners give a $\beta(-\det_k^{1/k}, \tilde{O}(k))$-composable core-set of size $\tilde{O}(k)$ for the fractional budgeted minimization problem BM$(-\det_k^{1/k}, V, k)$. Observe that for any t,

$$\beta(-\det_k(.)^{1/k}, t) = \sup_{A\in\mathbb{S}_d^+} \frac{-\det_k(A)^{1/k}}{-\det_k(tA)^{1/k}} = \sup_{A\in\mathbb{S}_d^+} \frac{\det_k(A)^{1/k}}{t\det_k(A)^{1/k}} = \frac{1}{t}.$$

So, proposition 24 gives an $\tilde{O}(k)$-composable core-set for BM$(-\det_k^{1/k}, V, k)$. But, by theorem 6.25, the integrality gap of BM$(-\det_k^{1/k}, V, k)$ is e. Therefore, proposition 24 gives an $\tilde{O}(k)^k$-composable core-set for integral determinant maximization problem. This completes the proof of theorem 6.3.

6.6.2 *Experimental Design*

In this section we discuss another set of applications of proposition 24 to the problem of *experimental design* [108] Consider a noisy linear regression problem: Given n data points $v_1, v_2, \ldots, v_n \in \mathbb{R}^d$, we are interested in learning a vector $w \in \mathbb{R}^d$ from observations of the form $y_i = \langle v_i, w \rangle + \eta_i$ where the noise values η_i are i.i.d samples from a zero-mean Gaussian distribution. Suppose we are allowed to learn parameter w by only observing $k \ll n$ data points. Letting S be the set of k chosen data points and $\hat{w}$ be the maximum likelihood estimation of w, $w - \hat{w}$ has a d- dimensional zero-mean Gaussian distribution with covariance matrix $(\sum_{i \in S} v_i v_i^T)^{-1}$. In the *experimental design* problem the goal is to choose k data points where the corresponding covariance matrix minimizes a given function $f : \mathbb{S}_d^+ \to \mathbb{R}$. The formal definition of the problem is as follows.

Definition 6.27 (Experimental Design)**.** For $V \subset \mathbb{R}^d$ and $f : \mathbb{S}_d^+ \to \mathbb{R}^+$ and an integer B, the experimental design is the problem of finding

$$S^*(f, k) = \text{argmin}_{S \in \binom{V}{B}} f\left(\sum_{v \in S} v v^\intercal\right),$$

where S ranges over all multi-sets of size B.

Experimental design problem has applications to linear bandit [38, 59], diversity sampling [72], active learning [28], feature selection and matrix approximation [34, 11], sensor placement in wireless networks [64].

Note that for any function f, BM(V, f, B) is a continuous relaxation to the above problem. It is shown in [113, 5, 101] that there is a polynomial time algorithm that if f, in addition to being convex and monotone, has a " reciprocal multiplicity property", then for $B \gg d$, the solution of BM(V, f, B) can be rounded to an integer solution losing only a constant factor in the value.

We say a function $f : \mathbb{S}_d^+ \to \mathbb{R}^+$ is *regular* if it is convex, monotone and $f(tA) = f(A)/t$ for any $t > 0$ and $A \succeq 0$. Here are some examples of regular functions: Average

$A \mapsto \frac{\mathrm{tr}(A^{-1})}{d}$, Determinant $A \mapsto \det(A)^{-\frac{1}{d}}$, Min Eigenvalue $A \mapsto \|A^{-1}\|_2$, Variance, $A \mapsto \frac{1}{d}\langle \sum_{v \in V} vv^\intercal \Sigma^{-1} \rangle$. They prove the following.

Theorem 6.28 ([5]). *There exists a polynomial time algorithm that for any regular function $f : \mathbb{S}_d^+ \to \mathbb{R}^+$, $\epsilon < 1/3$ and $B \geq \frac{5d}{\epsilon^2}$. outputs a multi-set S such that*

$$f\left(\sum_{v \in S} vv^\intercal\right) \leq (1 + 8\epsilon)\,\mathrm{OPT}(BM(V, f, B)),$$

Combining it with proposition 24 for $k = d$ leads to the following corollary.

Corollary 6.29. *There exists a polynomial time algorithm which finds an $\tilde{O}(d)$-composable core-set of size $\tilde{O}(d)$ for the experimental design with any regular function and $k \geq Cd$ where C is a universal constant.*

We simply use proposition 24 and the fact that any regular function is monotone, and hence vector d-monotone. Since for any regular function f, $\beta(f, t) = 1/t$, we obtain an $\tilde{O}(d)$-composable core-set of size $\tilde{O}(d)$ for the fractional version of the experimental design problem. But then, by theorem 6.28 any α-composable core-set for the fractional experimental design problem is an $O(\alpha)$-composable core-set for (integer) experimental design. Again, we emphasize that given V, B, for any regular function our algorithm outputs exactly the same composable core-set.

In section 6.7 we show that for many examples of regular functions f, the above bound is almost optimal.

6.7 Lower Bound

In this section, we study lower-bounds on the approximation ratio and size of the composable core-sets for the k-determinant maximization and the experimental design problem. In particular, we prove theorem 6.4. We also prove the bound given by corollary 6.29 is optimal up to a logarithmic factor for some of the regular functions.

Figure 1: A Hard Input for Composable Core-sets

1. Set $m = \frac{d}{\log d}$ to have $d^{m/d} = O(1)$.
2. Consider a set $G \subset \mathbb{R}^{m+1}$ of $n = d^{\beta+2}$ vectors such that for every two vectors $p, q \in G$, we have
$$\langle p, q \rangle \leq O\left(\frac{\sqrt{\beta}\log d}{\sqrt{d}}\right) \tag{6.32}$$
3. Do the following for any $1 \leq i \leq d - m$:
Embed G into the space spanned by $e_1, \ldots, e_m$ and e_{m+i}, and call it G_i. Choose an index $\pi_i \in [n]$ uniformly at random. Construct X_i by rotating G_i using a rotation $R(\pi_i)\colon \mathbb{R}^d \to \mathbb{R}^d$ that maps the π_i-th vector in G_i to e_{m+i}, and that maps the rest of the vectors in G_i to points in $\langle e_1, \ldots, e_m, e_{m+i} \rangle$.
4. Choose a matrix Q uniformly at random from the Haar measure over the space of rotations in $\mathbb{R}^d$ (i.e., orthogonal $d \times d$ matrices).
5. Return $QX_1, \ldots, QX_{d-m}$ and $QY_1, \ldots, QY_m$ where $Y_i = Me_i$ for a large enough scalar M.

6.7.1 Construction of a Hard Input

Here, we describe a distribution over collection of vectors which turns out to be a"hard" input for composable core-sets in many spectral problems. We use that in the next subsection to establish our lower-bound results. The construction of the instance is described in Figure 1.

To construct the instance we need to start with a set of vectors G satisfying (6.32). The following lemma guarantees the existence of the set G.

Lemma 6.30 (Implied by [33])**.** *Let G be a set of $d^{\beta+2}$ vectors chosen independently and uniformly at random from the $(m+1)$-dimensional unit sphere for $m = \frac{d}{\log d}$ and $\beta \geq 1$. Then with with probability at least $1 - 1/d^3$, for every two vectors $p, q \in G$, we have $\langle p, q \rangle \leq O(\frac{\sqrt{\beta}\log d}{\sqrt{d}})$.*

Proof. Let $\epsilon = \frac{C_1\sqrt{\beta}\log d}{\sqrt{d}}$ for some constant C_1 that we specify later. For any two random vectors chosen uniformly at random from the $(m+1)$-dimensional unit sphere, their inner product is distributed as $\mathcal{N}(0,1)/\sqrt{m}$. Using Lemma 2.2 (b) from [33], the probability that their inner product is more than ϵ, is bounded by $e^{-\frac{\epsilon^2}{3}\cdot m}$ (note that this uses the fact that $\epsilon^2 m > 27$).

Thus, by union bound, the probability that for any pair of points in G, their inner product is bounded by ϵ, is at least $1 - d^{2\beta+4}\cdot e^{-\frac{\epsilon^2}{3}\cdot m} \geq 1 - d^{2\beta+4}\cdot e^{-\frac{\beta(C_1\log d)^2}{3d}\cdot\frac{d}{\log d}} \geq 1 - d^{-\beta C_1^2/3+2\beta+4}$. Setting $C_1 = 6$, this probability is at least $1 - 1/d^3$. □

The main property of the sets generated in Figure 1 that we use is the following.

Lemma 6.31. *Let c be an arbitrary core-set function. For any $i = 1 \ldots d-m$, the probability that the image of e_{m+i} under Q is in the core-set for QX_i is bounded by $\frac{|c(X_i)|}{|X_i|}$, i.e.,*

$$\mathbb{P}_{Q,\pi}[Qe_{m+i} \in c(QR(\pi_i)G_i)] \leq \frac{|c(X_i)|}{|X_i|}$$

Proof. From the right-translation-invariance of Haar measure, it follows that, for any fixed value of π_i, the distribution of $QR(\pi_i)G_i$ is the same as the distribution of QG_i. Therefore, the joint distribution of $(\pi_i, QR(\pi_i)G_i)$ is the same as of (π_i, QG_i), so it suffices to bound $\mathbb{P}_{Q,\pi}[(G_i)_{\pi_i} \in c(QG_i)]$, where $(G_i)_{\pi_i}$ denotes the π_i-th vector in G_i. Since π_i and QG_i are independent, the bound follows. □

6.7.2 *Lower-bounds for Composable Core-sets for Spectral Problems*

Consider the collection of sets generated by the procedure described in Figure 1. Without loss of generality we may assume $Q = I$, as rotation matrices do not change spectral quantities we are interested in. So let $X_1, \ldots, X_{d-m}$ and $Y_1, \ldots, Y_m$ be the output sets. We are only interested in polynomial size core-sets, so fix a core-set function c that maps any set in $\mathbb{R}^d$ to its subsets of size at most d^β for some constant $\beta \geq 1$. Using

Lemma 6.31 and union bound, the probability that for at least one $1 \leq i \leq d-m$ we have $e_{m+i} \in c(X_i)$, is at most $(d-m) \cdot \frac{|c(X_i)|}{|X_i|} \leq \frac{(d-m)d^\beta}{d^{\beta+2}} \leq 1/d$. So WLOG we can assume $(Q)e_{m+i} = e_{m+i} \notin c(X_i)$ for any $1 \leq i \leq d-m$. It implies the following assumption that we crucially use in the future proofs.

Assumption. For any $u \in \bigcup_{i=1}^{d-m} c(X_i)$,

$$\left\langle \Pi_{\langle e_{m+1},\ldots,e_d\rangle}, uu^\intercal \right\rangle = \sum_{j=1}^{d-m} \langle u, e_{m+j}\rangle^2 \leq O\left(\frac{\beta \log^2 d}{d}\right). \tag{6.33}$$

To see this, suppose $u \in c(X_i)$ for some i. Since $X_i \subset \langle e_1, \ldots, e_m, e_{m+i}\rangle$ by construction, we have $\langle u, e_{m+j}\rangle = 0$ for $j \neq i$. Moreover, we assumed $e_{m+i} \notin c(X_i)$, so $\langle u, e_{m+i}\rangle \leq O(\frac{\sqrt{\beta}\log d}{\sqrt{d}})$ by (6.32).

We also define

$$V = \left(\bigcup_{i=1}^{d-m} X_i\right) \cup \left(\bigcup_{j=1}^{m} Y_j\right) \quad \text{and} \quad U = \left(\bigcup_{i=1}^{d-m} c(X_i)\right) \cup \left(\bigcup_{j=1}^{m} c(Y_j)\right). \tag{6.34}$$

In what follows we assume (6.33) holds.

Proof of theorem 6.4. First let us proof the theorem for $k = d$. Consider the core-set function c and input sets $X_1, \ldots, X_{d-m}, Y_1, \ldots, Y_m$ explained above. Consider the optimal set of d vectors maximizing the determinant on the union of the input sets, V. It contains $e_{m+1}, \cdots, e_d$ from the sets $X_1, \cdots, X_{d-m}$, respectively, and the points $Me_1, \cdots, Me_m$ from the sets $Y_1, \cdots, Y_m$ respectively. The value of this solution is equal to $(M^m)^2$. At the same time, the optimal solution from the union of the core-sets U must contain the m vectors $Me_1, \cdots, Me_m$ from the sets $Y_1, \cdots, Y_m$, if we set M to be large enough. Any other set of $k-m = d-m$ vectors must be chosen from the union of core-sets $c(X_i)$. So by Hadamard inequality we get the optimum is at most $(M^m)^2 \cdot \max_{u \in U} \left(\langle \Pi_{\langle e_{m+1},\ldots,e_d\rangle^\perp}, uu^\intercal\rangle\right)^{d-m}$ which results in a value of at most $(\frac{M^m}{(\sqrt{d}/(O(\sqrt{\beta})\log d))^{d-m}})^2 = \frac{M^{2m}(O(\sqrt{\beta})\log d)^{2(d-m)}}{d^{d-m}}$ by assumption (6.33). Hence the approximation ratio is at least $(d/(O(\sqrt{\beta})\log d)^2)^{d-m}$. Noting $m = \frac{d}{\log d} = o(d)$ completes the proof for $k = d$.

To extend the above proof for smaller k, we can consider the same instance in $d' = k$

dimensions, and then append the vectors with $d - d'$ zeros. It is straight-forward to see this gives us the same result for any value of $k \leq d$, yielding Theorem 6.4. □

Now, we present our lower-bounds on the approximation ratio of composable core-sets for the experimental design problem. Again, we consider the aforementioned core-set function c and input sets $X_1, \ldots, X_{d-m}, Y_1, \ldots, Y_m$.

Proposition 32. *Composable core-sets of size at most d^β for the experimental design problem with respect to the function $A \mapsto \|A^{-1}\|_2$ and size parameter $B \geq Cd$ have an approximation factor of at least $O(\frac{d}{\beta \log^2 d})$, for any $\beta \geq 1$ and a universal constant C.*

Proof. Note that it is enough to show the same lower-bound for the corresponding fractional budget minimization problem (BM). Since as pointed out in section 6.6, the relaxation BM has constant integrality gap when the conditions of theorem 6.28 are satisfied (which is satisfied for large enough C). Therefore, we show for any B and $f = (A \mapsto \|A^{-1}\|_2)$,

$$\frac{\mathrm{OPT}(\mathrm{BM}(U, f, B))}{\mathrm{OPT}(\mathrm{BM}(V, f, B))} \geq \Omega\left(\frac{d}{\beta \log^2 d}\right)$$

where V and U are defined by (6.34). Let us first find an upper bound on the optimal on V. For simplicity we work with the reciprocal of f (note that for any $A \in \mathbb{S}_d^+$, $\frac{1}{f(A)} = \lambda_d(A)$). Picking $Me_i \in B_i$ for any $1 \leq i \leq m$, and $e_{m+i} \in A_i$ for any $1 \leq i \leq d - m$, all with multiplicity $\frac{B}{d}$, we can deduce

$$\frac{1}{\mathrm{OPT}(\mathrm{BM}(V, f, B))} \geq \lambda_d\left(\frac{B}{d} \cdot \left(\sum_{i=1}^{d-m} e_{m+i} e_{m+i}^\intercal + \sum_{j=1}^{m} M^2 e_j e_j^\intercal\right)\right) \geq \frac{B}{d},$$

for $M > 1$. So in order to prove the theorem, is suffices to show for any feasible solution $s \in \mathbb{R}^U$ of $\mathrm{BM}(U, f, B)$ (which means $\sum_{u \in U} s_u \leq B$), we have $\lambda_d\left(\sum_{u \in U} s_u u u^\intercal\right) \leq$

$O\left(\frac{\beta \cdot B \log^2 d}{d^2}\right)$. We have

$$\begin{aligned}
\lambda_d\left(\sum_{u\in U} s_u uu^\intercal\right) &\le \frac{1}{d-m}\sum_{i=m+1}^{d}\lambda_i\left(\sum_{u\in U} s_u uu^\intercal\right)\\
&\le \frac{1}{d-m}\left\langle \sum_{u\in U} s_u uu^\intercal, \Pi_{\langle e_{m+1},\ldots,e_d\rangle}\right\rangle && \text{By lemma 6.6}\\
&\le \frac{\sum_{u\in U} s_u}{d-m}\cdot O\left(\frac{\beta\log^2 d}{d}\right) && \text{By (6.33)}
\end{aligned}$$

which completes the proof, as $\sum_{u\in U} s_u \le B$. □

Proposition 33. *Composable core-sets of size at most d^β for the experimental design problem with respect to the function $A \mapsto \det(A)^{-1/d}$ and size parameter $B \ge Cd$ have an approximation factor of at least $O(\frac{d}{\beta \log^2 d})$, for any $\beta \ge 1$ and a universal constant C.*

Proof. Similar to the previous proposition, it is enough to show that for function $f = A \mapsto \det{}^{-1/d}$ and any B

$$\frac{\mathrm{OPT}(\mathrm{BM}(U,f,B))}{\mathrm{OPT}(\mathrm{BM}(V,f,B))} \ge \Omega\left(\frac{d}{\beta\log^2 d}\right),$$

where U and V are defined in (6.34). let us first find an upper bound on the optimum on V (or equivalently a lower-bound on its reciprocal). If we choose e_{m+i} from X_i for any $1 \le i \le d-m$ and Me_j from Y_j for any $1 \le j \le m$ with equal weights of B/d, we get $\frac{1}{\mathrm{OPT}\,\mathrm{BM}(V,f,B)} \ge \frac{B}{d}M^{2m/d}$. So in order to prove the theorem it suffices to show

$$\frac{1}{\mathrm{OPT}(\mathrm{BM}(U,f,B))} = \det\left(\sum_{u\in U} s_u uu^\intercal\right)^{1/d} \le BM^{2m/d}\cdot O\left(\frac{\beta\log^2 d}{d^2}\right)$$

for any feasible solution $s \in \mathbb{R}^U$, i.e. $\sum_{u\in U} s(u) \le B$. By Cauchy-Binet we know $\det\left(\sum_{u\in U} s_u uu^\intercal\right) = \left(\sum_{S\in\binom{U}{d}} \det(\sum_{u\in S} s_u uu^\intercal)\right)$. Taking M to be large enough implies the summation is dominated by terms containing all vectors $Me_1,\ldots,Me_m$. So letting

$H = \langle e_{m+1}, \ldots, e_d \rangle$, we have

$$\det\left(\sum_{u\in U} s_u uu^\intercal\right) = M^{2m}(1+o(1)) \cdot \sum_{S\in\binom{U}{d-m}} \det\left(\sum_{j=1}^{m} s_{e_j} e_j e_j^\intercal + \sum_{u\in S} s_u uu^\intercal\right)$$
$$= M^{2m}(1+o(1)) \cdot \prod_{j=1}^{m} s_{e_j} \cdot \sum_{S\in\binom{U}{d-m}} \det_{d-m}\left(\sum_{u\in S} s_u \Pi_H(u_i)\Pi_H(u_i)^\intercal\right)$$

Now, note that if $p, q \in U$ both belong to the same $c(X_i)$, then the corresponding determinant in the summation is zero as $p, q \in \langle e_1, \ldots, e_m, e_{m+i} \rangle$ by construction. So we have

$$\det\left(\sum_{u\in U} s_u uu^\intercal\right) = M^{2m}(1+o(1)) \cdot \prod_{j=1}^{m} s_{e_j} \cdot \sum_{\substack{(u_1,\ldots,u_{d-m}) \\ \in \\ c(X_1)\times\ldots,\times c(X_{d-m})}} \det_{d-m}\left(\sum_{i=1}^{d-m} s_{u_i} \Pi_H(u_i)\Pi_H(u_i)^\intercal\right)$$
$$\leq M^{2m}(1+o(1)) \cdot \prod_{j=1}^{m} s_{e_j} \cdot \left(\prod_{i=1}^{d-m} \sum_{u\in c(X_i)} s_u\right) \cdot \max_{S\in\binom{U}{d-m}} \det_{d-m}\left(\sum_{u\in S} \Pi_H(u)\Pi_H(u)^\intercal\right) \tag{6.35}$$

We can further simplify the above by combining Hadamard inequality and (6.33). It implies

$$\max_{S\in\binom{U}{d-m}} \det_{d-m}\left(\sum_{u\in S} \Pi_H(u)\Pi_H(u)^\intercal\right) \leq \max_{S\in\binom{U}{d-m}} \prod_{u\in S} \|\Pi_H(u)\|^2 \leq O\left(\frac{\beta \log^2 d}{d}\right)^{(d-m)} \tag{6.36}$$

Furthermore, by AM-GM inequality we get that $\prod_{j=1}^{m} s_{e_j} \cdot \left(\prod_{i=1}^{d-m} \sum_{u\in c(X_i)} s_u\right) \leq \left(\frac{\sum_{u\in U} s_u}{d}\right)^d \leq \frac{B^d}{d^d}$. Combining the above with (6.35) and (6.36) proves

$$\det\left(\sum_{u\in U} s_u uu^\intercal\right)^{1/d} \leq M^{2m/d} \cdot \frac{B}{d} \cdot O\left(\frac{\beta \log^2 d}{d}\right)^{(d-m)/d}.$$

Noting $d^{2m/d} = O(1)$ completes the proof. □

Chapter 7

COMPOSABLE CORE-SETS FOR DETERMINANT MAXIMIZATION: FROM A PRACTICAL PERSPECTIVE

In the previous chapter, we introduced the notion of spectral spanners and used them to construct composable core-sets for the determinant maximization problem. Although the guarantees of the resulting method are theoretically almost optimal, and the algorithm can be simply implemented and run in polynomial time, it might not be yet efficient in practice for very large datasets. In this chapter we study construction of composable core-sets for the determinant maximization problem from a more practical perspective. In particular, we study two popular heuristics for the problem:

1. The *Greedy* algorithm that has been previously used in similar contexts for the submodular maximization problem. We provide a theoretical approximation guarantee of $O(C^{k^2})$ in the context of composable core-sets.

2. We propose a *Local Search* based algorithm that while being still practical, achieves a nearly optimal approximation bound of $O(k)^{2k}$.

Further, we implement these two proposed methods and compare them with the optimal method via spectral spanners to show the effectiveness of these methods on standard datasets.

7.1 Introduction

Given a set of vectors $P \subset \mathbb{R}^d$ and an integer $1 \leq k \leq d$, the goal of the determinant maximization problem is to find a subset $S = \{v_1, \ldots, v_k\}$ of P such that the determinant of the Gram matrix of the vectors in S is maximized. As pointed out before, in the language of DPPs this problem corresponds to the problem of finding the most diverse subset in a set of given items; in this context the problem has found several applications in machine learning over the last few years [72, 93, 52, 119]. Many of these applications need to handle large amounts of data and consequently the problem has been considered in massive data models of computation [93, 117, 103, 96, 94, 91, 14]. One strong such model that we consider, is *composable core-set* [61] which is an efficient summary of a data set with the composability property: union of summaries of multiple data sets should provably result in a good summary for the union of the data sets (see definition 6.10 for the formal definition). If designed for a task, composable core-sets will further imply efficient streaming and distributed algorithms for the same task. This has lead to recent interest in composable core-sets model since its introduction [91, 10, 60].

An almost optimal approximate composable core-set. In the previous chapter, we designed composable core-sets of size $O(k \log k)$ with approximation guarantee of $\tilde{O}(k)^k$ for the determinant maximization problem. Moreover, we showed that the best approximation one can achieve is $\Omega(k^{k-o(k)})$ for any polynomial size core-sets, proving that their algorithm is almost optimal. However, its complexity makes it less appealing in practice. First of all, the algorithm requires an explicit representation of the point set, which is not present for many DPP applications; a common case is that the DPP kernel is given by an oracle which returns the inner product between the points; in this setting, the algorithm needs to construct the associated gram matrix, and use SVD decomposition to recover the point set, making the time and memory quadratic in the size of the point-set. Secondly, even in the point set setting, the algorithm is not efficient for large inputs as it requires solving $O(kn)$ many linear programs, where n is size of the point set.

In this chapter, we focus on two simple to implement algorithms which are typically exploited in practice, namely the Greedy and the Local-Search algorithms. We study these algorithms from theoretical and experimental points of view for the composable core-set problem with respect to the determinant maximization objective, and we compare their performance with the algorithm of the previous chapter, which we refer to as the Spanner-based algorithm.

7.1.1 Our Contributions

Greedy algorithm. The greedy algorithm for determinant maximization has been described in algorithm 4.1. The algorithm proceeds in k iterations and at each iteration it picks the point that maximizes the volume of the parallelepiped formed by the set of points picked so far. [31] has studied the approximation of the greedy algorithm in the standard setting. In the context of submodular maximization over large data sets, variants of this algorithm have been studied [94]. One can view the greedy algorithm as a heuristic for constructing a core-set of size k. To the best of our knowledge, the previous analysis of this algorithm does not provide any multiplicative approximation guarantee in the context of composable core-sets.[1]

Our first result shows the first multiplicative approximation factor for composable core-sets on the determinant maximization objective achieved by the Greedy algorithm.

Theorem 7.1. *Given a set of points $P \subset \mathbb{R}^d$, the Greedy algorithm achieves a $O(C^{k^2})$-composable coreset of size k for the determinant maximization problem, where C is a constant.*

The Local Search algorithm. Our main contribution is to propose to use the Local Search algorithm for constructing a composable core-set for the task of determinant maximization. The algorithm starts with the solution of the Greedy algorithm and at each iteration, swaps in a point that is not in the core-set with a point that is already in the

[1] For more details, see related work.

core-set, as long as this operation increases the volume of the set of picked points. While still being simple, as we show, this algorithm achieves a near-optimal approximation guarantee.

Theorem 7.2. *Given a set of points $P \subset \mathbb{R}^d$, the Local Search algorithm achieves an $O(k)^{2k}$-composable coreset of size k for the determinant maximization problem.*

Directional height. Both of our theoretical results use a modular framework: In Section 7.3, we introduce a new geometric notion defined for a point set called *directional height*, which is closely related to the width of a point set defined in [4]. We show that core-sets for preserving the directional height of a point set in fact provide core-sets for the determinant maximization problem. Finally, we show that running either the Greedy (Section 7.5) or Local Search (Section 7.4) algorithms on a point set obtain composable core-sets for its directional height. We believe that this new notion might find applications elsewhere.

Experimental results. Finally, we implement all three algorithms and compare their performances on two real data sets: MNIST[78] data set and GENES data set, previously used in [15, 83] in the context of DPPs. Our empirical results show that in more than 87% percent of the cases, the solution reported by the Local Search algorithm improves over the Greedy algorithm. The average improvement varies from 1% to up to 23% depending on the data set and the settings of other parameters such as k. We further show that although the Local Search algorithm picks fewer points than the tight approximation algorithm of [60] (k vs. upto $O(k \log k)$), its performance is better and it runs faster.

7.1.2 Related Work

In a broader scope, determinant maximization is an instance of the (non-monotone) submodular maximization where the logarithm of the determinant is the submodular objective function. There is a long line of work on distributed submodular optimization and

its variants [30, 12, 96, 73]. In particular, there has been several efforts to design composable core-sets in various settings of the problem [96, 91, 14]; In [96], authors study the problem for monotone functions, and show the greedy algorithm offers a $min(m,k)$-composable core-set for the problem where m is the number of parts. On the other hand, [61] shows that it is impossible to go beyond an approximation factor of $\Omega(\frac{\sqrt{k}}{\log k})$ with polynomial size core-sets. Moreover, [91, 14] consider a variant of the problem where the data is randomly distributed, and show the greedy algorithm achieves constant factor "randomized" composable core-sets for both monotone and non-monotone functions. However, one can notice that these results can not be directly compared to the current work, as a multiplicative approximation for determinant converts to an additive guarantee for the corresponding submodular function.

As discussed before, the determinant is one way to measure the diversity of a set of items. Diversity maximization with respect to other measures has been also extensively studied in the literature, [55, 51, 22, 16]. More recently, the problem has received more attention in distributed models of computation, and for several diversity measures constant factor approximation algorithms have been devised [120, 61, 26]. However, these notions are typically defined by considering only the pairwise dissimilarities between the items; for example, the summation of the dissimilarities over all pairs of items in a set can define its diversity.

One can also go further, and study the problem under additional constraints, such as matroid and knapsack constraints. This has been an active line of research in the past few years, and several centralized and distributed algorithms have been designed in this context for submodular optimization [92, 80, 81, 29] and in particular determinant maximization [45, 100].

7.2 *Preliminaries*

Throughout the chapter, we fix d as the dimension of the ambient space and $k(k \leq d)$ as the size parameter of the determinant maximization problem. We call a subset of $\mathbb{R}^d$ a point set, and use the term point or vector to refer to an element of a point set. For a set of points $S \subset \mathbb{R}^d$ and a point $p \in \mathbb{R}^d$, we write $S + p$ to denote the set $S \cup \{p\}$, and for a point $s \in S$, we write $S - p$ to denote the set $S \setminus \{s\}$.

Let S be a point set of size k. We use $\text{VOL}(S)$ to denote the k-dimensional volume of the parallelepiped spanned by vectors in S. Also, let M_S denote a $k \times d$ matrix where each row represents a point of S. Then, the following relates volume to the determinant

$$\det(M_S M_S^\intercal) = \text{VOL}^2(S).$$

So the determinant maximization problem can also be phrased as *volume maximization*. We use the former, but because of the geometric nature of the arguments, sometimes we switch to the volume notation. For any point set P, we use MAXDET_k to denote the optimal of determinant maximization for P, i.e. $\text{MAXDET}_k(P) = \max_S \det(M_S M_S^\intercal)$, where S ranges over all subsets of size k. MAXVOL_k is also defined similarly.

For a point set P, we use $\langle P \rangle$ to refer to the linear subspace spanned by the vectors in P. We also denote the set of all k-dimensional linear subspaces by $\mathcal{H}_k$. For a point p and a subspace $\mathcal{H}$, we use $\text{dist}(p, \mathcal{H})$ to show the Euclidean distance of p from $\mathcal{H}$.

Greedy algorithm for volume maximization. As pointed out before, a widely used algorithm for determinant maximization in the offline setting is a greedy algorithm which given a point set P and a parameter k as the input does the following: start with an empty set $\mathcal{C}$. For k iterations, add $\text{argmax}_{p \in P} \text{dist}(p, \langle \mathcal{C} \rangle)$ to $\mathcal{C}$. The result would be a subset of size k which has the following guarantee.

Theorem 7.3 ([31]). *Let P be a point set and $\mathcal{C}$ be the output of the greedy algorithm on P. Then* $\text{VOL}(\mathcal{C}) \geq \frac{\text{MAXVOL}_k(P)}{k!}$.

7.3 k-Directional Height

As pointed out in the introduction, we introduce a new geometric notion called directional height, and reduce the task of finding composable core-sets for determinant maximization to finding core-sets for this new notion.

Definition 7.4 (k-Directional Height)**.** Let $P \subset \mathbb{R}^d$ be a point set and $\mathcal{H} \in \mathcal{H}_{k-1}$ be a $(k-1)$-dimensional subspace. We define the k-directional height of P with respect to $\mathcal{H}$, denoted by $h(P, \mathcal{H})$, to be the distance of the farthest point in P from $\mathcal{H}$, i.e., $h(P, \mathcal{H}) = \max_{p \in P} \text{dist}(p, \mathcal{H})$.

The notion is an instance of an *extent measure* defined in [4]. It is also related to the notion of *directional width* of a point set previously used in [4], which for a direction vector $v \in \mathbb{R}^d$ is defined to be $\max_{p \in P} \langle v \cdot p \rangle - \min_{p \in P} \langle v \cdot p \rangle$.

Next, we define core-sets with respect to this notion. It is essentially a subset of the point set that approximately preserves the k-directional height of the point set with respect to any subspace in $\mathcal{H}_k$.

Definition 7.5 (α-Approximate Core-set for the k-Directional Height)**.** Given a point set P, a subset $C \subseteq P$ is a α-approximate core-set for the k-directional height of P, if for any $\mathcal{H} \in \mathcal{H}_{k-1}$, we have $h(C, \mathcal{H}) \geq h(P, \mathcal{H})/\alpha$.

We also say a mapping $c(.)$ which maps any point set in $\mathbb{R}^d$ to one of its subsets, is an α-approximate core-set for the k-directional height problem, if the above relation holds for any point set P and $c(P)$.

The above notion of core-sets for k-directional height is similar to the notion of ϵ-kernels defined in [4] for the directional width of a point set.

We connect it to composable core-sets for determinant maximization by the following lemma.

Lemma 7.6. *Let $P_1, \ldots, P_m \in \mathbb{R}^d$ be an arbitrary collection of point sets, and for any i, let $c(P_i)$ be an α-approximate core-set for the k-directional height for P_i. Then*

$$\text{MAXDET}_k(\bigcup_{i=1}^{m} P_i) \leq \alpha^{2k} \cdot \text{MAXDET}_k(\bigcup_{i=1}^{m} c(P_i)).$$

Proof. Let $W \subset \bigcup_{i=1}^{m} P_i$ be any subset of size k, and also let $w \in W \setminus \bigcup_{i=1}^{m} c(P_i)$. We claim that there is a point q in the union of the core-sets such that $\alpha \cdot \text{VOL}(W - w + q) \geq \text{VOL}(W)$. Note that showing this claim is enough to prove the lemma. Since, one can start from the optimum solution which achieve the largest volume on $\bigcup_{i=1}^{m} P_i$, and for at most k iterations, replace a point outside $\bigcup_{i=1}^{m} c(P_i)$ by a point inside, while decreasing the volume by a factor of at most α.

So it remains to prove the claim. Let $W = \{w_1, \ldots, w_k\}$, and let $H = \langle w_2, \ldots, w_k \rangle \in \mathcal{H}_{k-1}$ be the plane spanned by $w_2, \ldots, w_k$. By definition, $\text{VOL}(W) = \text{dist}(w_1, H) \cdot \text{VOL}(W - w_1)$. On the other hand, suppose that $w_1 \in P_i$. Then by our assumption, there exists $q \in c(P_i)$ so that $\text{dist}(q, H) \geq \frac{\text{dist}(w_1, H)}{\alpha}$. Replacing w_1 with q, we get

$$\begin{aligned} \text{VOL}(W - w_1 + q) &= \text{dist}(q, H) \cdot \text{VOL}(W - w_1) \\ &\geq \frac{\text{dist}(w_1, H) \cdot \text{VOL}(W - w_1)}{\alpha} = \frac{\text{VOL}(W)}{\alpha} \end{aligned}$$

which completes the proof.

□

Corollary 7.7. *Any mapping which is an α-approximate core-set for k-directional height, is an α^{2k}-composable core-set for the determinant maximization.*

We employ the above corollary to analyze both greedy and local search algorithms in Sections 7.4 and 7.5.

7.4 *The Local Search Algorithm*

In this section, we describe and analyze the local search algorithm and prove theorem 7.2. The algorithm is described in algorithm 7.9.

To prove theorem 7.2, we follow a two steps strategy. We first analyze the algorithm for individual point sets, and show that the output is a $(2k)$-approximate core-set for the k-directional height problem, as follows.

Lemma 7.8. *Let P be a set of points and $c(P)$ be the result of running the local search algorithm on P. Then, for any $\mathcal{H} \in \mathcal{H}_{k-1}$,*

$$h(c(P), \mathcal{H}) \geq \frac{h(P, \mathcal{H})}{2k(1+\epsilon)}.$$

Next, we apply corollary 7.7, which implies that local search gives $(2k(1+\epsilon))^{2k}$-composable core-sets for the determinant maximization. Clearly this completes the proof of the theorem by setting ϵ to a constant.

So proving theorem 7.2 boils down to showing lemma 7.8. Before, getting into that, we analyze the running time, and present some remarks about the implementation of the algorithm.

Running time. Let $\mathcal{C}_0$ be the output of the greedy. By theorem 7.3 $\frac{\text{VOL}(\mathcal{C}_0)}{\text{MAXVOL}_k(P)} \geq \frac{1}{k!}$. The algorithm starts with $\mathcal{C}_0$ and by definition, in any iteration increases the volume by a factor of at least $1+\epsilon$, hence the total number of iterations is $O(\frac{k \log k}{\log(1+\epsilon)})$. Finally, each iteration can be naively executed by iterating over all points in P, forming the corresponding $k \times k$ matrix, and computing the determinant in total time $O(|P| \cdot kd \cdot k^3|P|)$.

We also remark that unlike the algorithm in [60], our method can also be executed without any changes and additional complexity, when the point set P is not explicitly given in the input; instead, it is presented by an oracle that given two points of P returns their

inner product. One can note that in this case the algorithm can be simulated by querying this oracle for at most $O(|P|k)$ times.

Algorithm 7.9 Local Search Algorithm

Input: A point set $P \subset \mathbb{R}^d$, integer k, and $\epsilon > 0$.

Output: A set $\mathcal{C} \subset P$ of size k.

1: Initialize $\mathcal{C} = \emptyset$.
2: **for** $i = 1$ **to** k **do**
3: Add $\text{argmax}_{p \in P \setminus \mathcal{C}} \text{VOL}(\mathcal{C} + p)$ to $\mathcal{C}$.
4: **end for**
5: **repeat**
6: If there are points $q \in P \setminus C$ and $p \in \mathcal{C}$ such that

$$\text{VOL}(\mathcal{C} + q - p) \geq (1 + \epsilon)\text{VOL}(\mathcal{C})$$

replace p with q.
7: **until** No such pair exists.

return $\mathcal{C}$.

7.4.1 *Proof of lemma 7.8*

With no loss of generality, suppose that $\epsilon = 0$, the proof automatically extends to $\epsilon \neq 0$. Therefore, $c(P)$ has the following property: for any $v \in P \setminus c(P)$ and $u \in c(P)$, $\text{VOL}(c(P)) \geq \text{VOL}(c(P) - u + v)$. Fix $\mathcal{H} \in \mathcal{H}_{k-1}$, and let $p = \text{argmax}_{p \in P} \text{dist}(p, \mathcal{H})$. Our goal is to show there exists $q \in c(P)$ so that $\text{dist}(q, \mathcal{H}) \geq \frac{\text{dist}(p, \mathcal{H})}{2k}$.

Let $\mathcal{G} = \langle c(P) \rangle$ be the k-dimensional linear subspace spanned by the set of points in the core-set, and let $p_{\mathcal{G}}$ be the projection of p onto this subspace. We proceed with proof by contradiction. Set $\text{dist}(p, \mathcal{H}) = 2x$, and suppose to the contrary that for any $q \in c(P)$, $\text{dist}(q, \mathcal{H}) < \frac{x}{k}$. With this assumption, we prove the two following lemmas.

Lemma 7.9. $\text{dist}(p, p_{\mathcal{G}}) < x$.

Lemma 7.10. $\text{dist}(p_{\mathcal{G}}, \mathcal{H}) < x$.

One can note that, combining the above two lemmas and applying the triangle inequality implies $\text{dist}(p, \mathcal{H}) \leq \text{dist}(p, p_{\mathcal{G}}) + \text{dist}(p_{\mathcal{G}}, \mathcal{H}) < 2x$, which contradicts the assumption $\text{dist}(p, \mathcal{H}) = 2x$ and completes the proof.

Therefore, it only remains to prove the above lemmas. Let us first fix some notation. Let $c(P) = \{q_1, \ldots, q_k\}$ and for any i, let $\mathcal{G}_{\bar{i}}$ denote the $(k-1)$-dimensional subspace spanned by points in $c(P) \setminus \{q_i\}$.

Proof of lemma 7.9. For $1 \leq i \leq k$, let q'_i be the projection of q_i onto $\mathcal{H}$. We prove that there exists an index $j \leq k$ such that we can write $q'_j = \sum_{i \neq j} \alpha_i q'_i$ where every $\alpha_i \leq 1$. Let r be the rank, i.e., maximum number of independent points of $\mathcal{C}' = \{q'_i | i \leq k\}$ and clearly as $\mathcal{H}$ has dimension $k-1$, we have $r \leq k-1$. Take a subset $S \subset \mathcal{C}'$ of r independent points that have the maximum volume and let q'_j be a point in $\mathcal{C}' \setminus S$ and note that this point should exist as there are k points in the core-set. Thus we can write $q'_j = \sum_{i:q'_i \in S} \alpha_i q'_i$. With an idea similar to the one presented in [31], we can prove that the following claim holds.

Claim 7.11. *For any i such that $q'_i \in S$, we have $|\alpha_i| \leq 1$.*

Proof. We prove that if the claim is not ture, then $S \setminus \{q'_i\} \cup \{q'_j\}$ has a larger volume than S which contradicts the choice of S. Let $\mathcal{F}$ be the linear subspace passing through $S \setminus \{q'_i\}$. It is easy to see that $\frac{\text{VOL}(S)}{\text{VOL}(S \setminus \{q'_i\} \cup \{q'_j\})} = \frac{\text{dist}(q'_i, \mathcal{F})}{\text{dist}(q'_j, \mathcal{F})}$. This means that $\text{dist}(q'_i, \mathcal{F}) \geq \text{dist}(q'_j, \mathcal{F})$. However, if $|\alpha_i| > 1$ then since q'_i is the only point in S which in not in $\mathcal{F}$, then $\text{dist}(q'_j, \mathcal{F}) \geq \text{dist}(q'_i, \mathcal{F})$ which is a contradiction. □

Finally, for any $q'_i \notin S$, set the corresponding coefficient $\alpha_i = 0$. So we get that $q'_j =$

$\sum_{i \neq j} \alpha_i q'_i$ where every $|\alpha_i| \leq 1$.

Now take the point $q = \sum_{i \neq j} \alpha_i q_i$. Note that, q'_j is in fact the projection of q onto $\mathcal{H}$. Therefore, using triangle inequality, we have

$$\begin{aligned} \operatorname{dist}(q'_j, q) = \operatorname{dist}(q, \mathcal{H}) &\leq \sum_{i \neq j} |\alpha_i| \operatorname{dist}(q_i, \mathcal{H}) \\ &\leq (k-1)x/k. \end{aligned} \tag{7.1}$$

Then we get that

$$\begin{aligned} &\operatorname{dist}(p, p_{\mathcal{G}}) = \operatorname{dist}(p, \mathcal{G}) \\ &\leq \operatorname{dist}(p, \mathcal{G}_{\bar{j}}) \quad \text{as } \mathcal{G}_{\bar{j}} \subset \mathcal{G} \\ &\leq \operatorname{dist}(q_j, \mathcal{G}_{\bar{j}}) \quad \text{by the local search property} \\ &\leq \operatorname{dist}(q_j, q) \quad \text{as } q \in \mathcal{G}_{\bar{j}} \\ &\leq \operatorname{dist}(q_j, q'_j) + \operatorname{dist}(q'_j, q) \quad \text{by triangle inequality} \\ &< x/k + (k-1)x/k \quad \text{by our assumption and Equation 7.1} \\ &= x \end{aligned}$$

□

Proof of lemma 7.10. Again we prove that we can write $p_{\mathcal{G}} = \sum_{i=1}^{k} \alpha_i q_i$ where all $|\alpha_i| \leq 1$. We assume that the set of points q_i are linearly independent, otherwise the points in P have rank less than k and thus the volume is 0. Therefore, we can write $p_{\mathcal{G}} = \sum_{i=1}^{k} \alpha_i q_i$. Note that for any i, we have

$$\begin{aligned} \operatorname{dist}(p_{\mathcal{G}}, \mathcal{G}_{\bar{i}}) &\leq \operatorname{dist}(p, \mathcal{G}_{\bar{i}}) \\ &\leq \operatorname{dist}(q_i, \mathcal{G}_{\bar{i}}) \quad \text{by the local search property} \end{aligned}$$

where the first inequality follows since $\mathcal{G}_{\bar{i}}$ is a subspace of $\mathcal{G}$ and $p_{\mathcal{G}}$ is the projection of p onto $\mathcal{G}$. Again, similar to the proof of Claim 7.11, this means that $|\alpha_i| \leq 1$. Therefore, using triangle inequality

$$\begin{aligned} \operatorname{dist}(p_{\mathcal{G}}, \mathcal{H}) = \operatorname{dist}(\sum_{i=1}^{k} \alpha_i q_i, \mathcal{H}) &\leq \sum_{i=1}^{k} |\alpha_i| \operatorname{dist}(q_i, \mathcal{H}) \\ &< k \times x/k = x \end{aligned}$$

□

7.5 *The Greedy Algorithm*

In this section we analyze the performance of the greedy algorithm (seesection 7.2) as a composable core-set function for the determinant maximization and prove theorem 7.1. Our proof plan is similar to the to the analysis of the local search. We analyze the guarantee of the greedy as a core-set mapping for k-directional height, and combining that with corollary 7.7 we achieve the result. We prove the following.

Lemma 7.12. *Let P be an arbitrary point set and $c(P)$ denote the output of running greedy on P. Then, $c(P)$ is a $(2k) \cdot 3^k$-approximate core-set for the k-directional height of P, i.e. for any $\mathcal{H} \in \mathcal{H}_{k-1}$ we have*

$$h(c(P), \mathcal{H}) \geq \frac{1}{2k \cdot 3^k} \cdot h(P, \mathcal{H})$$

So the greedy is a $(2k \cdot 3^k)$-approximate core-set for k-directional height problem. Combining with corollary 7.7, we conclude it is also a $(2k \cdot 3^k)^{2k}$ composable core-set for the determinant maximization which proves theorem 7.1.

7.5.1 *Proof of lemma 7.12*

The proof is similar to the proof of lemma 7.8. Let $\mathcal{G} = \langle c(P) \rangle$ be the k-dimensional subspace spanned by the output of greedy. Also for a point $p \in P$, define $p_{\mathcal{G}}$ to be its projection onto $\mathcal{G}$. Fix $\mathcal{H} \in \mathcal{H}_{k-1}$, let $h(c(P), \mathcal{H}) = \frac{x}{k}$ for some number x, which in particular implies that for any $q \in c(P)$, $\text{dist}(q, \mathcal{H}) \leq \frac{x}{k}$. Then, our goal is to prove $h(P, \mathcal{H}) \leq 2 \cdot 3^k \cdot x$. We show that by proving the following two lemmas.

Lemma 7.13. *For any $p \in P$, $\text{dist}(p_{\mathcal{G}}, \mathcal{H}) \leq 2^{k-1}x$.*

Lemma 7.14. *For any $p \in P$, $\text{dist}(p, p_{\mathcal{G}}) \leq 3^k x$.*

Clearly, combining them with triangle inequality, we get that for any $p \in P$, $\text{dist}(p, \mathcal{H}) < 2 \cdot 3^k x$, which implies $h(P, \mathcal{H}) \leq 2 \cdot 3^k \cdot x$ and completes the proof. So it remains to prove the lemmas.

Let the output of the greedy $c(P)$ be $q_1, \ldots, q_k$ with this order, i.e. q_1 is the first vector selected by the algorithm.

Proof of lemma 7.13. Recall that $q_1, \ldots, q_k$ is the output of greedy. For any $p \in P$ and for any $1 \leq t \leq k$, let $\mathcal{G}_t = \langle q_1, \ldots, q_t \rangle$ and define p^t to be the projection of p onto $\mathcal{G}_t$. We prove the lemma by showing the following claim.

Claim 7.15. *For any $p \in P$ and any $1 \leq t \leq k$, we can write $p^t = \sum_{i=1}^{t} \alpha_i q_i$ so that for each i, $|\alpha_i| \leq 2^{t-1}$.*

Let us first show how the above claim implies the lemma. It follows that we can write $p_{\mathcal{G}} = p^k = \sum_{i=1}^{k} \alpha_i q_i$ where all $|\alpha_i| \leq 2^{k-1}$. Now since for each $i \leq k$, $\text{dist}(q_i, \mathcal{H}) \leq x/k$ by assumption, we have that $\text{dist}(p_{\mathcal{G}}, \mathcal{H}) \leq \sum \alpha_i \, \text{dist}(q_i, \mathcal{H}) \leq 2^{k-1} x$. Therefore, it suffices to prove the claim.

Proof of claim 7.15. We use induction on t. To prove the base case of induction, i.e., $t = 1$, note that q_1 is the vector with largest norm in P. Thus we have that $\|p^1\| \leq \|q_1\|$ and therefore we can write $p^1 = \alpha_1 q_1$ where $|\alpha_1| \leq 1$. Now, lets assume that the hypothesis holds for the first t points; that is, the projection of any point p onto $\mathcal{G}_t$ can be written as $\sum_{j \leq t} \alpha_j q_j$ where $|\alpha_j|$'s are at most 2^{t-1}.

Now, note that by the definition of the greedy algorithm, q_{t+1} is the point with farthest distance from $\mathcal{G}_t$. Therefore, for any point $p \in P \setminus \{q_1, \cdots, q_{t+1}\}$, we know that $\text{dist}(p, \mathcal{G}_t) \leq \text{dist}(q_{t+1}, \mathcal{G}_t)$, and thus, $\text{dist}(p^{t+1}, \mathcal{G}_t) \leq \text{dist}(q_{t+1}, \mathcal{G}_t)$. Therefore we can write

$$p^{t+1} = \alpha_{t+1} q_{t+1} - \alpha_{t+1} q_{t+1}^t + p^t \quad \text{where } |\alpha_{t+1}| \leq 1.$$

By the hypothesis, we can write $p^t = \sum_{j \le t} \beta_j q_j$, and $q^t_{t+1} = \sum_{j \le t} \gamma_j q_j$, where $|\beta_j| \le 2^{t-1}$, and $|\gamma_j| \le 2^{t-1}$. Since $|\alpha_{t+1}| \le 1$, we can write

$$p^{t+1} = \alpha_{t+1} q_{t+1} + \sum_{j \le t} (\beta_j - \alpha_{t+1}\gamma_j) q_j = \sum_{j \le t+1} \alpha_j q_j$$

where $|\alpha_j| \le 2^t$. This completes the proof of the claim and the lemma. □

□

Proof of lemma 7.14. First, note that for any t, we have $\text{dist}(q_{t+1}, \mathcal{G}_t) \ge \text{dist}(p, \mathcal{G}_{k-1})$. This is because the greedy algorithm has chosen q_k over p in its k-th round which means that $\text{dist}(p, \mathcal{G}_{k-1}) \le \text{dist}(q_k, \mathcal{G}_{k-1})$, and by definition of the greedy algorithm for any $i < j$ we have $\text{dist}(q_{i+1}, \mathcal{G}_i) \ge \text{dist}(q_{j+1}, \mathcal{G}_j)$. So it is enough to prove

$$\exists\, 1 \le t \le k-1 \text{ s.t. } \text{dist}(q_{t+1}, \mathcal{G}_t) \le 3^k x \tag{7.2}$$

For $1 \le i \le k$, let q'_i be the projection of q_i onto $\mathcal{H}$. Recall that, we are assuming that for any i, $\text{dist}(q_i, q'_i) < x/k$. To prove (7.2), we use proof by contradiction, so suppose that for all t, $\text{dist}(q_{t+1}, \mathcal{G}_t) > 3^k x$. We also define $\mathcal{G}'_t$ to be the projection of $\mathcal{G}_t$ on $\mathcal{H}$, i.e., $\mathcal{G}'_t = \langle q'_1, \ldots, q'_t \rangle$. Given these assumptions, we prove the following claim.

Claim 7.16. *For any $1 \le t \le k-1$, we can write $\Pi(\mathcal{G}'_t)(q'_{t+1}) = \sum_{i \le t} \alpha_i q'_i$ where $|\alpha_i| \le 3^t$, where for a point q and a subspace $\mathcal{A}$, $\Pi(\mathcal{A})(q)$ denotes projection of q onto $\mathcal{A}$.*

Intuitively, this is similar to claim 7.15. However, instead of looking at the execution of the algorithm on the points $q_1, \cdots, q_k$, we look at the execution of the algorithm on the projected points $q'_1, \cdots, q'_k$. Since all of these k points are relatively close to the hyperplane $\mathcal{H}$, the distances are not distorted by much and therefore, we can get approximately the same bounds. The formal proof is presented at the end of the section.

To finish the proof of the lemma, let us show how it follows from the claim. First, note that $q'_1, \ldots, q'_k$ are k points in the $(k-1)$-dimensional space $\mathcal{H}$, so for some t, q'_{t+1} should lie inside $\mathcal{G}'_t$ and we have $\Pi(\mathcal{G}'_t)(q'_{t+1}) = q'_{t+1}$. Fix such t. Define the point $q_\alpha = \sum_{i \le t} \alpha_i q_i$ where $|\alpha_i| \le 3^k$ are taken from the above claim which means $q'_{t+1} = \sum_{i \le t} \alpha_i q'_i$. Note that

by definition $q'_{t+1} = \Pi(\mathcal{H})(q_\alpha)$. Therefore,

$$\begin{aligned} \operatorname{dist}(q'_{t+1}, q_\alpha) &= \operatorname{dist}(q_\alpha, \mathcal{H}) && (7.3)\\ &\leq \sum_{i \leq t} \alpha_i \operatorname{dist}(q_i, \mathcal{H}) \leq 3^k t \cdot x/k. && (7.4) \end{aligned}$$

Then we get that

$$\begin{aligned} \operatorname{dist}(q_{t+1}, \mathcal{G}_t) &\leq \operatorname{dist}(q_{t+1}, q_\alpha) \quad \text{as } q_\alpha \in \mathcal{G}_t \\ &\leq \operatorname{dist}(q_{t+1}, q'_{t+1}) + \operatorname{dist}(q'_{t+1}, q_\alpha) \\ &\leq x/k + 3^k t \cdot x/k \leq 3^k x \end{aligned}$$

where the second inequality holds because of triangle inequality and the last one from (7.3) and the fact that $t \leq k-1$. This contradicts our assumption that $\operatorname{dist}(q_{t+1}, \mathcal{G}_t) > 3^k x$, and proves the lemma. □

Proof of claim 7.16.

We prove the claim by induction on t, and show that for any j s.t. $j > t$, the point $\Pi(\mathcal{G}'_t)(q'_j)$ can be written as the sum $\sum_{i \leq t} \alpha_i q'_i$ such that $|\alpha_i| \leq 3^t$.

Base Case. First, we prove the base case of induction, i.e., $t = 1$. Recall that by our assumption, $\|q_1\| > 3^k x$, and thus by triangle inequality, we have that $\|q'_1\| \geq \|q_1\| - x/k \geq 3^k x - x/k \geq 2x$. Therefore, since q_1 is the vector with largest norm in P, using triangle inequality again, we have that for any $j > 1$,

$$\|q'_j\| \leq \|q_j\| \leq \|q_1\| \leq \|q'_1\| + x/k \leq (1 + \frac{1}{2k})\|q'_1\|$$

Therefore we can write $\Pi(\mathcal{G}'_1)(q'_j) = \alpha_1 q'_1$ where $|\alpha_1| \leq 2$.

Inductive step. Now, lets assume that the hypothesis holds for $\mathcal{G}'_t$. In particular this means that we can write $\Pi(\mathcal{G}'_t)(q'_{t+1}) = \sum_{i \leq t} \beta_i q'_i$ where $|\beta_i| \leq 3^t$, and that for a given $j > t+1$, we can write $\Pi(\mathcal{G}'_t)(q'_j) = \sum_{i \leq t} \gamma_i q'_i$ where $|\gamma_i|$'s are at most 3^t. Now let

$\ell = \text{dist}(q'_{t+1}, \mathcal{G}'_t)$. By triangle inequality, we get that

$$\begin{aligned}
\text{dist}(q_{t+1}, \mathcal{G}_t) &\leq \text{dist}(q_{t+1}, q'_{t+1}) && (7.5)\\
&+ \text{dist}(q'_{t+1}, \Pi(\mathcal{G}'_t)(q'_{t+1}) + && (7.6)\\
&\text{dist}(\Pi(\mathcal{G}'_t)(q'_{t+1}), \mathcal{G}_t)\\
&\leq x/k + \ell + \text{dist}(\sum_{i\leq t} \beta_i q'_i, \sum_{i\leq t} \beta_i q_i)\\
&\leq x/k + \ell + \sum_{i\leq t} |\beta_i|\, x/k\\
&\leq \ell + 3^t x. && (7.7)
\end{aligned}$$

Now we consider two case. If $\ell \leq 3^t x$ then using the above

$$\text{dist}(q_{t+1}, \mathcal{G}_t) \leq 2 \cdot 3^t x \leq 3^k x,$$

which contradicts our assumption of $\text{dist}(q_{t+1}, \mathcal{G}_t) > 3^k x$. Otherwise,

$$\begin{aligned}
\text{dist}(\Pi(\mathcal{G}'_{t+1})(q'_j), \mathcal{G}'_t) &\leq \text{dist}(q'_j, \mathcal{G}'_t) \leq \text{dist}(q_j, \mathcal{G}_t)\\
&\leq \text{dist}(q_{t+1}, \mathcal{G}_t) \leq 2\ell,
\end{aligned}$$

where the last inequality follows from Equation 7.5 . Therefore, we can write $\Pi(\mathcal{G}'_{t+1})(q'_j) = \alpha_{t+1} q'_{t+1} - \alpha_{t+1}\Pi(\mathcal{G}'_t)(q'_{t+1}) + \Pi(\mathcal{G}_t)(q'_j)$ where $\alpha_{t+1} \leq 2$.

By the hypothesis, we can write $\Pi(\mathcal{G}'_t)(q'_j) = \sum_{i\leq t} \gamma_i q'_i$, where $|\gamma_i| \leq 3^t$. Since $|\alpha_{t+1}| \leq 2$, we can write

$$\begin{aligned}
\Pi(\mathcal{G}'_{t+1})(q'_j) &= \alpha_{t+1} q'_{t+1} + \sum_{i\leq t} (\gamma_i - \alpha_{t+1}\beta_i) q'_i\\
&= \sum_{i\leq t+1} \alpha_i q'_i \quad \text{where } |\alpha_i| \leq 3^{t+1}.
\end{aligned}$$

This completes the proof of the claim.

□

7.6 Experiments

In this section, we evaluate the effectiveness of our proposed Local Search algorithm empirically on real data sets. We implement the following three algorithms.

- The Greedy algorithm of Section section 7.5 (GD).
- The Local Search algorithm of Section section 7.4 with accuracy parameter $\epsilon = 10^{-5}$ (LS).
- The LP-based algorithm of the previous chapter which has almost tight approximation guarantee theoretically (LP). Note that this algorithm might pick up to $O(k \log k)$ points in the core-set.

Data sets. We use two data sets that were also used in [83] in the context of approximating DPPs over large data sets.

- MNIST [78]: contains a set of 60000 images of hand-written digits, where each image is of size 28 by 28.
- GENES [15]: contains a set of 10000 genes, where each entry is a feature vector of a gene. The features correspond to shortest path distances of 330 different hubs in the BioGRID gene interaction network. This data set was initially used to identify a diverse set of genes to predict a tumor. Here, we slightly modify it and remove genes that have an unknown value at any coordinate which gives us a data set of size ~ 8000.

Moreover, we apply an RBF kernel on both of these data sets using $\sigma = 6$ for MNIST and $\sigma = 10$ for GENES. These are the same values used in the work of [83].

7.6.1 Experiment setup.

We partition the data sets uniformly at random into multiple data sets $P_1, \cdots, P_m$. We use $m = 10$ for the smaller GENES data set, and for the larger MNIST data set we use $m = 50$ and also we use $m = 10$ (equal to the number of digits in the data set). Moreover, since the partitions are random, we repeat every experiment 10 times and take the average in our reported results.

We then use a *core-set construction algorithm* ALG_c to compute core-sets of size k, i.e., $S_1 = \text{ALG}_c(P_1, k), \cdots, S_m = \text{ALG}_c(P_m, k)$, for $\text{ALG}_c \in \{\text{GD, LS, LP}\}$. Recall that GD, LS and LP correspond to the Greedy, Local Search and LP-based algorithm of [60] respectively.

Finally, we take the union of these core-sets $U_{\text{ALG}_c} = S_1 \cup \cdots \cup S_m$ and compute the solutions for U_{ALG_c}. Since computing the optimal solution can take exponential time ($\sim n^k$), we will instead use an *aggregation algorithm* ALG_a (either GD, LS or LP). We will use the notation $\text{ALG}_a/\text{ALG}_c$ to refer to the constructed set of k points, returned by $\text{ALG}_a(U_{\text{ALG}_c}, k)$. For example, GD/LS refers to the set of k points returned by the Greedy algorithm on the union of the core-sets, where each core-set is produced using the Local Search algorithm.

Finally, we vary the value of k from 3 to 20.

7.6.2 Results

Local Search vs. Greedy as offline algorithms.

Our first set of experiments simply compares the quality of Greedy and Local Search as centralized algorithms on whole data sets. We perform this experiment to measure the improvement of Local Search over Greedy in the offline setting. On average over all values of k, Local Search improves over Greedy by 13% for GENES data set and 5% for MNIST data set. Figure 7.1 shows the improvement ratio of the determinant of the solution returned by the Local Search algorithm over the determinant of the solution returned by the Greedy algorithm. On average over all values of k, Local Search improves over Greedy by 13% for GENES data set and 5% for MNIST data set. Figure 7.2 shows the ratio of the time it takes to run the Local Search and Greedy algorithms as a function of k for both data sets. On average, it takes about 6.5 times more to run the Local Search algorithm.

Intuitively, this improvement upper bounds the improvement one can expect in the core-

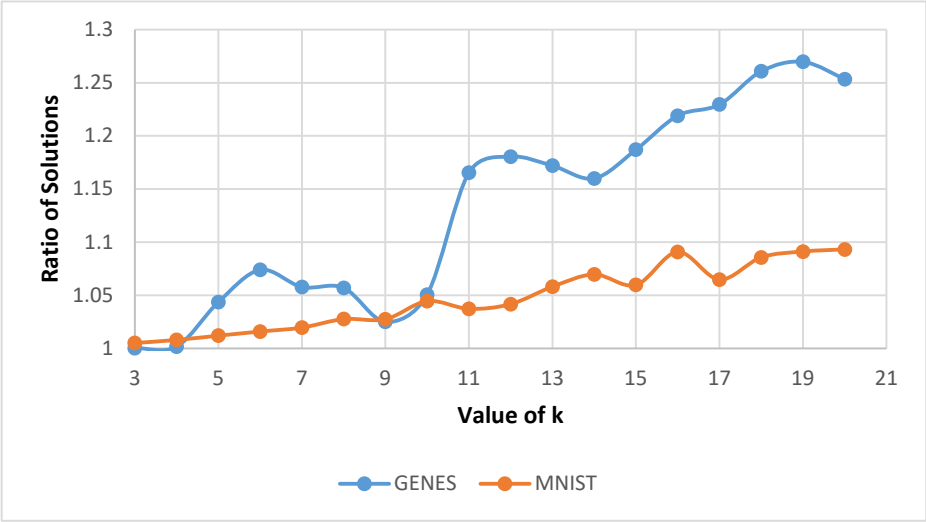


Figure 7.1: Average improvement of Local Search over Greedy as a function of k.

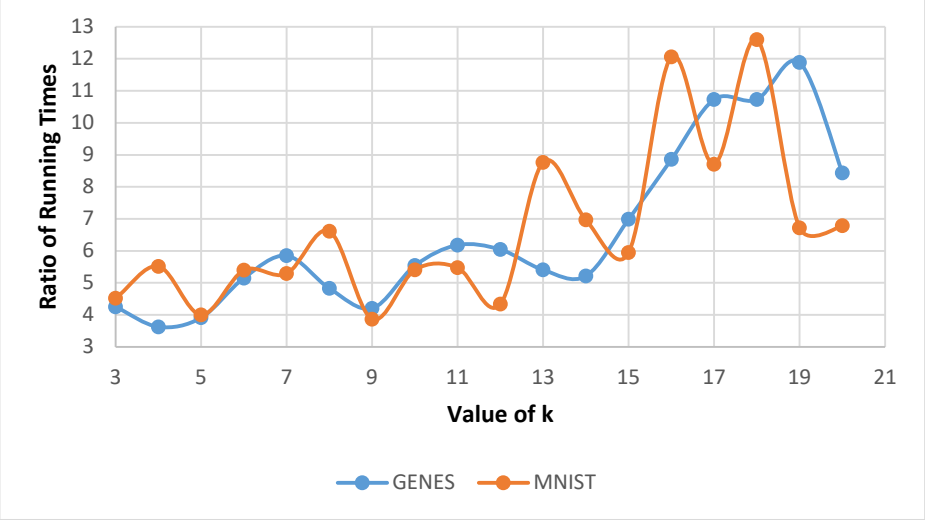


Figure 7.2: Average ratio of the run time of Local Search over Greedy as a function of k.

set setting.

Local Search vs. Greedy as core-sets.

In our second experiment, we use Greedy algorithm for aggregation, i.e., $\text{ALG}_a = GD$, and compare GD/LS with GD/GD. Figure 7.3 shows the improvement of local search over greedy as a core-set construction algorithm. The graph is drawn as a function of k, and for each k, the improvement ratio is an average over all 10 runs, and shown for all data sets (including GENES, MNIST with partition number $m = 10$, and MNIST with $m = 50$).

On average this improvement is 9.6%, 2.5% and 1.9% for GENES, MNIST10 and MNIST50 respectively. Moreover, in 87% of all 180 runs of this experiment, Local Search performed better than Greedy, and for some instances, this improvement was up to 58%. Finally, this improvement comes at a cost of increased running time. Figure 7.4 shows average ratio of the time to construct core-sets using Local Search vs. Greedy.

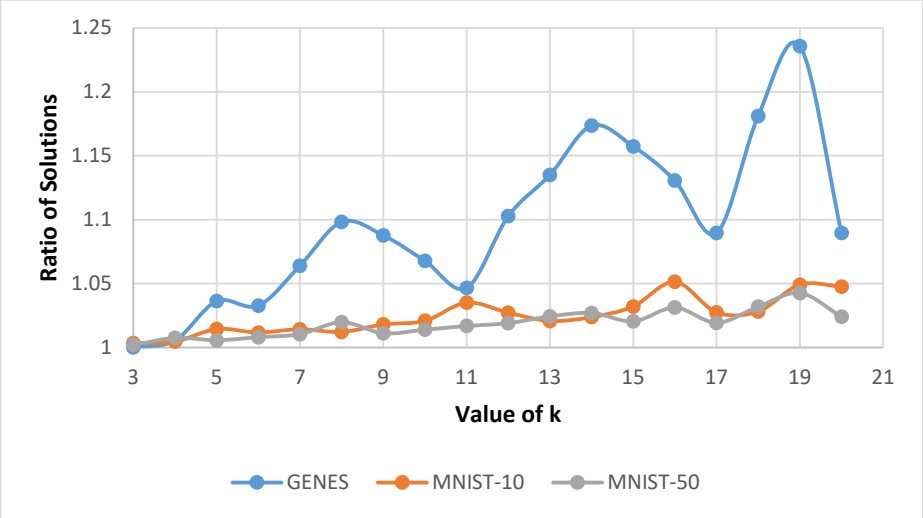


Figure 7.3: Average improvement of Local Search core-set over Greedy core-set as a function of k.

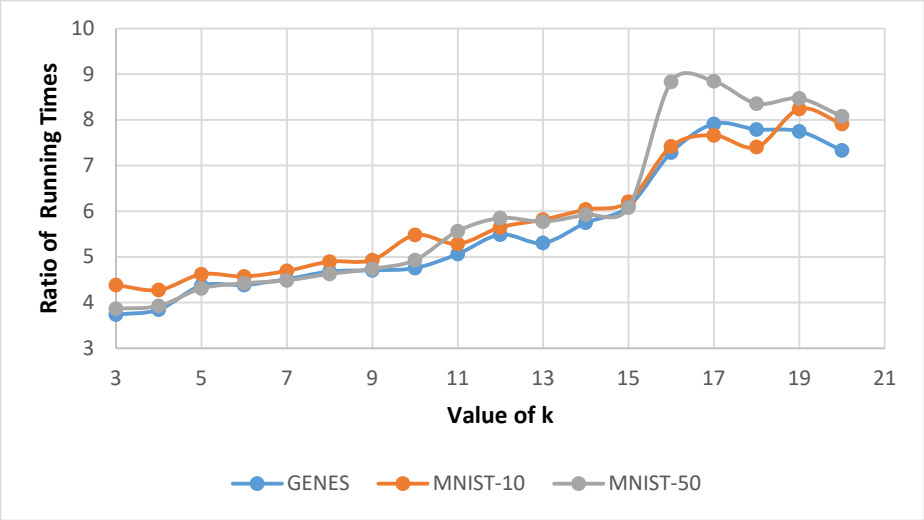


Figure 7.4: Average ratio of the run time of Local Search over Greedy as a function of k.

Local Search vs. Greedy - identical algorithms.

We also consider the setting where the core-set construction algorithm is the same as the aggregation algorithm. This mimics the approach of [96], who proposed to use the greedy algorithm on each machine to achieve a small solution; then each machine sends this solution to a single machine that further runs the greedy algorithm on the union of these solutions and reports the result.

In this paper show that if instead of Greedy, we use Local Search in *both steps*, the solution will improve significantly. Using our notation, here we are comparing LS/LS vs. GD/GD. Figure 7.5 shows the improvement as a function of k, taken average over all 10 runs.

On average the improvement is 23%, 5.5% and 6.0% for GENES, MNIST10 and MNIST50 respectively. Moreover, in only 1 out of 180 runs the Greedy perfomed better than Local Search. The improvement could go as high as 67.7%.

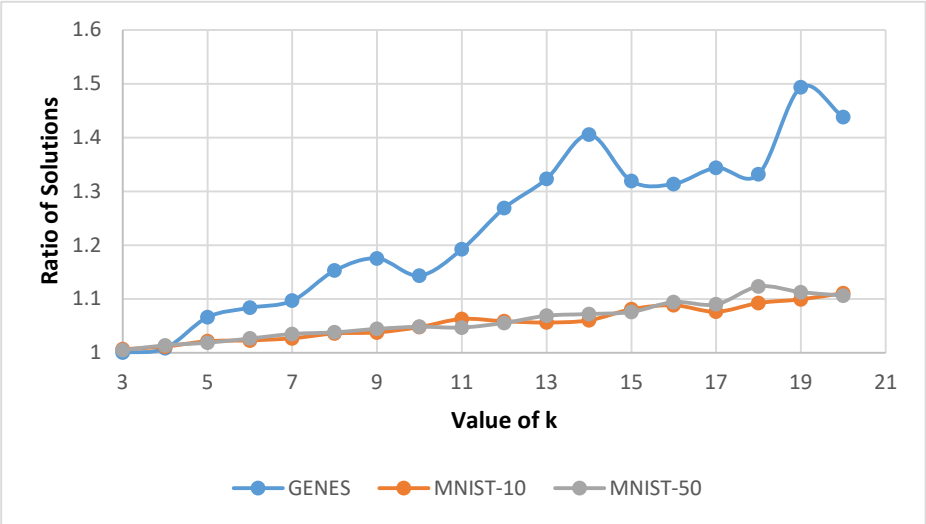


Figure 7.5: Average improvement of Local Search over Greedy as a function of k, in the identical algorithms setting.

Comparing Local Search vs. the LP-based algorithm.

Here we compare the performance of the Local Search algorithm and LP as algorithms for constructing core-sets, i.e. we compare GD/LS with GD/LP. Our experiments show that the proposed local search algorithm performs better: while picking fewer points in the core-set, in most cases local search finds a better solution and runs faster.

Figure 7.6 shows how much Local Search improves over the LP-based algorithm. On average this improvement is 7.3%, 1.8% and 1.4% for GENES, MNIST10 and MNIST50 respectively. Moreover, in 78% of all runs, Local Search performed better than Lp-based algorithm, and this improvement can go upto 63%. Figure 7.7 shows the average ratio of the time to construct core-sets using the LP-based algorithm vs. Local Search. As it is clear from the graphs, our proposed Local Search algorithm performs better than even the LP-based algorithm which has almost tight approximation guarantees: while picking fewer points in the core-set, in most cases it finds a better solution and runs faster.

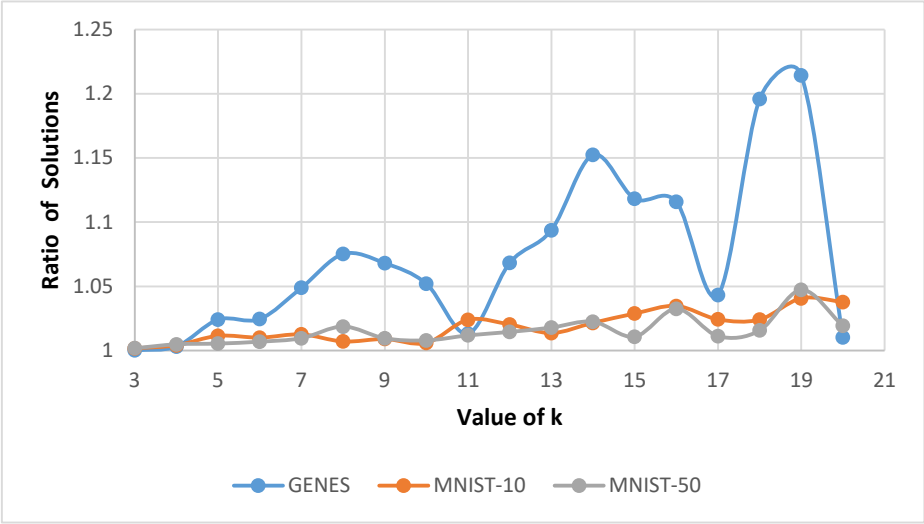


Figure 7.6: Average improvement of Local Search over LP-based algorithm for constructing core-sets as a function of k.

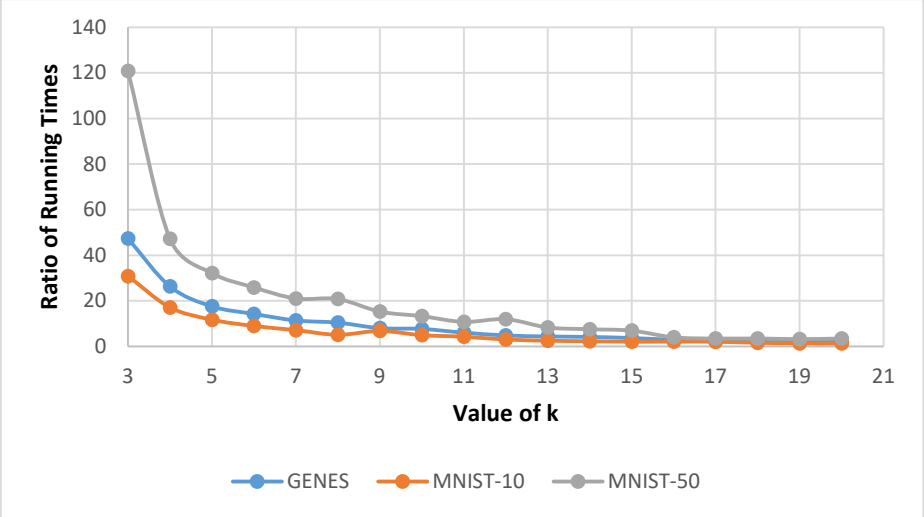


Figure 7.7: Average ratio of the run time of the optimal algorithm over local search as a function of k.

BIBLIOGRAPHY

[1] Sofiane Abbar, Sihem Amer-Yahia, Piotr Indyk, Sepideh Mahabadi, and Kasturi R Varadarajan. Diverse near neighbor problem. In *Proceedings of the twenty-ninth annual symposium on Computational geometry*, pages 207–214. ACM, 2013.

[2] Apoorv Agarwal, Anna Choromanska, and Krzysztof Choromanski. Notes on using determinantal point processes for clustering with applications to text clustering. *arXiv preprint arXiv:1410.6975*, 2014.

[3] Pankaj K Agarwal, Sariel Har-Peled, and Kasturi R Varadarajan. Approximating extent measures of points. *Journal of the ACM (JACM)*, 51(4):606–635, 2004.

[4] Pankaj K Agarwal, Sariel Har-Peled, and Kasturi R Varadarajan. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30, 2005.

[5] Zeyuan Allen-Zhu, Yuanzhi Li, Aarti Singh, and Yining Wang. Near-optimal discrete optimization for experimental design: A regret minimization approach. *arXiv preprint arXiv:1711.05174*, 2017.

[6] Nima Anari, Shayan Oveis Gharan, and Alireza Rezaei. Monte carlo markov chain algorithms for sampling strongly rayleigh distributions and determinantal point processes. In *Conference on Learning Theory*, pages 103–115, 2016.

[7] Nima Anari and Shayan Oveis Gharan. Effective-Resistance-Reducing Flows and Asymmetric TSP. In *FOCS*, pages 20–39, 2015.

[8] Nima Anari and Shayan Oveis Gharan. A generalization of permanent inequalities and applications in counting and optimization. In *STOC*, pages 384–396, 2017.

[9] Sanjeev Arora, Michelangelo Grigni, David R Karger, Philip N Klein, and Andrzej Woloszyn. A polynomial-time approximation scheme for weighted planar graph tsp. In *SODA*, volume 98, pages 33–41, 1998.

[10] Sepehr Assadi and Sanjeev Khanna. Randomized composable coresets for matching and vertex cover. *arXiv preprint arXiv:1705.08242*, 2017.

[11] Haim Avron and Christos Boutsidis. Faster subset selection for matrices and applications. *SIAM Journal on Matrix Analysis and Applications*, 34(4):1464–1499, 2013.

[12] Ashwinkumar Badanidiyuru, Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Streaming submodular maximization: Massive data summarization on the fly. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 671–680. ACM, 2014.

[13] Keith Ball et al. An elementary introduction to modern convex geometry. *Flavors of geometry*, 31:1–58, 1997.

[14] Rafael Barbosa, Alina Ene, Huy Nguyen, and Justin Ward. The power of randomization: Distributed submodular maximization on massive datasets. In *International Conference on Machine Learning*, pages 1236–1244, 2015.

[15] Nematollah Kayhan Batmanghelich, Gerald Quon, Alex Kulesza, Manolis Kellis, Polina Golland, and Luke Bornn. Diversifying sparsity using variational determinantal point processes. *arXiv preprint arXiv:1411.6307*, 2014.

[16] Aditya Bhaskara, Mehrdad Ghadiri, Vahab Mirrokni, and Ola Svensson. Linear relaxations for finding diverse elements in metric spaces. In *Advances in Neural Information Processing Systems*, pages 4098–4106, 2016.

[17] Christophe Ange Napoléon Biscio and Frdŕic Lavancier. Quantifying repulsiveness of determinantal point processes. *Bernoulli*, 22(4):2001–2028, 11 2016.

[18] J. Borcea and P. Brädén. Multivariate Pólya-Schur classification problems in the Weyl algebra. *Proceedings of the London Mathematical Society*, 101(3):73–104, 2010.

[19] Julius Borcea and Petter Brändén. Applications of stable polynomials to mixed determinants: Johnson's conjectures, unimodality, and symmetrized Fischer products. *Duke Math. Journal*, 143(2):205–223, 2008.

[20] Julius Borcea, Petter Branden, and Thomas M. Liggett. Negative dependence and the geometry of polynomials. *Journal of American Mathematical Society*, 22:521–567, 2009.

[21] Alexei Borodin and Eric M Rains. Eynard–mehta theorem, schur process, and their pfaffian analogs. *Journal of statistical physics*, 121(3):291–317, 2005.

[22] Allan Borodin, Hyun Chul Lee, and Yuli Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on Principles of Database Systems*, pages 155–166. ACM, 2012.

[23] Christos Boutsidis, Michael W Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In *SODA*, pages 968–977, 2009.

[24] Petter Brändén. Polynomials with the half-plane property and matroid theory. *Advances in Mathematics*, 216(1):302–320, 2007.

[25] Stephen P Brooks and Andrew Gelman. General methods for monitoring convergence of iterative simulations. *Journal of computational and graphical statistics*, 7(4):434–455, 1998.

[26] Matteo Ceccarello, Andrea Pietracaprina, Geppino Pucci, and Eli Upfal. Mapreduce and streaming algorithms for diversity maximization in metric spaces of bounded doubling dimension. *Proceedings of the VLDB Endowment*, 10(5):469–480, 2017.

[27] Wei-Lun Chao, Boqing Gong, Kristen Grauman, and Fei Sha. Large-margin determinantal point processes. In *UAI*, pages 191–200, 2015.

[28] Kamalika Chaudhuri, Sham M Kakade, Praneeth Netrapalli, and Sujay Sanghavi. Convergence rates of active learning for maximum likelihood estimation. In *Advances in Neural Information Processing Systems*, pages 1090–1098, 2015.

[29] Chandra Chekuri, Shalmoli Gupta, and Kent Quanrud. Streaming algorithms for submodular function maximization. In *International Colloquium on Automata, Languages, and Programming*, pages 318–330. Springer, 2015.

[30] Flavio Chierichetti, Ravi Kumar, and Andrew Tomkins. Max-cover in map-reduce. In *Proceedings of the 19th international conference on World wide web*, pages 231–240. ACM, 2010.

[31] Ali Çivril and Malik Magdon-Ismail. On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoretical Computer Science*, 410(47-49):4801–4811, 2009.

[32] Ali Civril and Malik Magdon-Ismail. Exponential inapproximability of selecting a maximum volume sub-matrix. *Algorithmica*, 65(1):159–176, 2013.

[33] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of the johnson-lindenstrauss lemma. *International Computer Science Institute, Technical Report*, pages 99–006, 1999.

[34] FR De Hoog and RMM Mattheij. Subset selection for matrices. *Linear Algebra and its Applications*, 422(2-3):349–359, 2007.

[35] Amit Deshpande and Luis Rademacher. Efficient volume sampling for row/column subset selection. In *FOCS*, pages 329–338. IEEE, 2010.

[36] Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. In *SODA*, pages 1117–1126, 2006.

[37] Amit Deshpande and Kasturi Varadarajan. Sampling-based dimension reduction for subspace approximation. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 641–650. ACM, 2007.

[38] Yash Deshpande and Andrea Montanari. Linear bandits in high dimension and recommendation systems. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 1750–1754. IEEE, 2012.

[39] Persi Diaconis and David Freedman. On markov chains with continuous state space. Technical report, Technical Report, 1997.

[40] Persi Diaconis and Laurent Saloff-Coste. Comparison theorems for reversible markov chains. *The Annals of Applied Probability*, pages 696–730, 1993.

[41] Persi Diaconis and Daniel Stroock. Geometric bounds for eigenvalues of markov chains. *The Annals of Applied Probability*, pages 36–61, 1991.

[42] Jesse Dodge, Kevin Jamieson, and Noah A Smith. Open loop hyperparameter optimization and determinantal point processes. *arXiv preprint arXiv:1706.01566*, 2017.

[43] Yevgeniy Dodis and Sanjeev Khanna. Design networks with bounded pairwise distance. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 750–759. ACM, 1999.

[44] Javad B. Ebrahimi, Damian Straszak, and Nisheeth K. Vishnoi. Subdeterminant maximization via nonconvex relaxations and anti-concentration. In *FOCS*, pages 1020–1031, 2017.

[45] Javad B Ebrahimi, Damian Straszak, and Nisheeth K Vishnoi. Subdeterminant maximization via nonconvex relaxations and anti-concentration. In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, pages 1020–1031. Ieee, 2017.

[46] Michael Elkin and David Peleg. $(1+\varepsilon,\beta)$-spanner constructions for general graphs. *SIAM Journal on Computing*, 33(3):608–631, 2004.

[47] Tomás Feder and Milena Mihail. Balanced matroids. In *Proceedings of the twenty-fourth annual ACM symposium on Theory of Computing*, pages 26–38, New York, NY, USA, 1992. ACM.

[48] James Allen Fill. Eigenvalue bounds on convergence to stationarity for nonreversible markov chains, with an application to the exclusion process. *The annals of applied probability*, pages 62–87, 1991.

[49] Paul Francis, Sugih Jamin, Cheng Jin, Yixin Jin, Danny Raz, Yuval Shavitt, and Lixia Zhang. Idmaps: A global internet host distance estimation service. *IEEE/ACM Transactions On Networking*, 9(5):525–540, 2001.

[50] Jean Ginibre. Statistical ensembles of complex, quaternion, and real matrices. *Journal of Mathematical Physics*, 6(3):440–449, 1965.

[51] Sreenivas Gollapudi and Aneesh Sharma. An axiomatic approach for result diversification. In *Proceedings of the 18th international conference on World wide web*, pages 381–390. ACM, 2009.

[52] Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. Diverse sequential subset selection for supervised video summarization. In *Advances in Neural Information Processing Systems*, pages 2069–2077, 2014.

[53] Osman Güler. Hyperbolic polynomials and interior point methods for convex programming. *Mathematics of Operations Research*, 22(2):350–377, 1997.

[54] Raja Hafiz Affandi, Emily B Fox, and Ben Taskar. Approximate inference in continuous determinantal point processes. *arXiv preprint arXiv:1311.2971*, 2013.

[55] Refael Hassin, Shlomi Rubinstein, and Arie Tamir. Approximation algorithms for maximum dispersion. *Operations research letters*, 21(3):133–137, 1997.

[56] Philipp Hennig and Roman Garnett. Exact sampling from determinantal point processes. *arXiv preprint arXiv:1609.06840*, 2016.

[57] J Ben Hough, Manjunath Krishnapur, Yuval Peres, Bálint Virág, et al. Determinantal processes and independence. *Probability surveys*, 3:206–229, 2006.

[58] J.B. Hough, M. Krishnapur, Y. Peres, and B. Virág. Determinantal processes and independence. *Probability Surveys*, (3):206–229, 2006.

[59] Ruitong Huang, Tor Lattimore, András György, and Csaba Szepesvári. Following the leader and fast rates in linear prediction: Curved constraint sets and other

regularities. In *Advances in Neural Information Processing Systems*, pages 4970–4978, 2016.

[60] Piotr Indyk, Sepideh Mahabadi, Shayan Oveis Gharan, and Alireza Rezaei. Composable core-sets for determinant maximization problems via spectral spanners. *arXiv preprint arXiv:1807.11648*, 2018.

[61] Piotr Indyk, Sepideh Mahabadi, Mohammad Mahdian, and Vahab S Mirrokni. Composable core-sets for diversity and coverage maximization. In *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 100–108. ACM, 2014.

[62] Mark Jerrum and Jung Bae Son. Spectral gap and log-sobolev constant for balanced matroids. In *FOCS*, pages 721–729, 2002.

[63] Mark Jerrum, Jung-Bae Son, Prasad Tetali, and Eric Vigoda. Elementary bounds on poincaré and log-sobolev constants for decomposable markov chains. *Annals of Applied Probability*, pages 1741–1765, 2004.

[64] Siddharth Joshi and Stephen Boyd. Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, 57(2):451–462, 2009.

[65] Byungkon Kang. Fast determinantal point process sampling with application to clustering. In *NIPS*, pages 2319–2327, 2013.

[66] R. Kannan and S. Vempala. Spectral algorithms. *Foundations and Trends in Theoretical Computer Science*, 4:157–288, 2009.

[67] Ioannis Kontoyiannis and Sean P Meyn. Geometric ergodicity and the spectral gap of non-reversible markov chains. *Probability Theory and Related Fields*, pages 1–13, 2012.

[68] Turgay Korkmaz and Marwan Krunz. Source-oriented topology aggregation with multiple qos parameters in hierarchical networks. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 10(4):295–325, 2000.

[69] Alex Kulesza and Ben Taskar. Structured determinantal point processes. In *Advances in neural information processing systems*, pages 1171–1179, 2010.

[70] Alex Kulesza and Ben Taskar. k-dpps: Fixed-size determinantal point processes. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1193–1200, 2011.

[71] Alex Kulesza and Ben Taskar. Learning determinantal point processes. 2011.

[72] Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

[73] Ravi Kumar, Benjamin Moseley, Sergei Vassilvitskii, and Andrea Vattani. Fast greedy algorithms in mapreduce and streaming. *ACM Transactions on Parallel Computing (TOPC)*, 2(3):14, 2015.

[74] James T Kwok and Ryan P Adams. Priors for diversity in generative latent variable models. In *Advances in Neural Information Processing Systems*, pages 2996–3004, 2012.

[75] Frédéric Lavancier, Jesper Møller, and Ege Rubak. Determinantal point process models and statistical inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(4):853–877, 2015.

[76] Frédéric Lavancier, Jesper Møller, and Ege Holger Rubak. Statistical aspects of determinantal point processes. Technical report, Department of Mathematical Sciences, Aalborg University, 2012.

[77] Gregory F Lawler and Alan D Sokal. Bounds on the l^2 spectrum for markov chains and markov processes: a generalization of cheegers inequality. *Transactions of the American mathematical society*, 309(2):557–580, 1988.

[78] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[79] Donghoon Lee, Geonho Cha, Ming-Hsuan Yang, and Songhwai Oh. Individualness and determinantal point processes for pedestrian detection. In *European Conference on Computer Vision*, pages 330–346. Springer, 2016.

[80] Jon Lee, Vahab S Mirrokni, Viswanath Nagarajan, and Maxim Sviridenko. Non-monotone submodular maximization under matroid and knapsack constraints. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 323–332. ACM, 2009.

[81] Jon Lee, Maxim Sviridenko, and Jan Vondrák. Submodular maximization over multiple matroids via generalized exchange properties. *Mathematics of Operations Research*, 35(4):795–806, 2010.

[82] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2006.

[83] Chengtao Li, Stefanie Jegelka, and Suvrit Sra. Efficient sampling for k-determinantal point processes. 2015.

[84] László Lovász and Miklós Simonovits. Random walks in a convex body and an improved volume algorithm. *Random structures & algorithms*, 4(4):359–412, 1993.

[85] Russell Lyons. Determinantal probability measures. *Publications Mathématiques de l'IHÉS*, 98:167–212, 2003.

[86] Russell Lyons, Jeffrey E Steif, et al. Stationary determinantal processes: phase multiplicity, bernoullicity, entropy, and domination. *Duke Mathematical Journal*, 120(3):515–575, 2003.

[87] Odile Macchi. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7(1):83–122, 1975.

[88] Madan Lal Mehta and Michel Gaudin. On the density of eigenvalues of a random matrix. *Nuclear Physics*, 18:420–427, 1960.

[89] Milena Mihail. On the expansion of combinatorial polytopes. In *International Symposium on Mathematical Foundations of Computer Science*, pages 37–49. Springer, 1992.

[90] Ha Quang Minh, Partha Niyogi, and Yuan Yao. Mercers theorem, feature maps, and smoothing. In *International Conference on Computational Learning Theory*, pages 154–168. Springer, 2006.

[91] Vahab Mirrokni and Morteza Zadimoghaddam. Randomized composable coresets for distributed submodular maximization. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 153–162. ACM, 2015.

[92] Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, and Amin Karbasi. Fast constrained submodular maximization: Personalized data summarization. In *ICML*, pages 1358–1367, 2016.

[93] Baharan Mirzasoleiman, Stefanie Jegelka, and Andreas Krause. Streaming non-monotone submodular maximization: Personalized video summarization on the fly. *arXiv preprint arXiv:1706.03583*, 2017.

[94] Baharan Mirzasoleiman, Amin Karbasi, Ashwinkumar Badanidiyuru, and Andreas Krause. Distributed submodular cover: Succinctly summarizing massive data. In *Advances in Neural Information Processing Systems*, pages 2881–2889, 2015.

[95] Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization: Identifying representative elements in massive data. In *Advances in Neural Information Processing Systems*, pages 2049–2057, 2013.

[96] Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization. *The Journal of Machine Learning Research*, 17(1):8330–8373, 2016.

[97] Ravi Montenegro and Prasad Tetali. Mathematical aspects of mixing times in Markov chains. *Found. Trends Theor. Comput. Sci.*, 1(3):237–354, May 2006.

[98] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functionsi. *Mathematical programming*, 14(1):265–294, 1978.

[99] Aleksandar Nikolov. Randomized rounding for the largest simplex problem. In *STOC*, pages 861–870, 2015.

[100] Aleksandar Nikolov and Mohit Singh. Maximizing determinants under partition constraints. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 192–201, 2016.

[101] Aleksandar Nikolov, Mohit Singh, and Uthaipon Tao Tantipongpipat. Proportional volume sampling and approximation algorithms for a-optimal design. *arXiv preprint arXiv:1802.08318*, 2018.

[102] Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. A Randomized Rounding Approach to the Traveling Salesman Problem. In *FOCS*, pages 550–559, 2011.

[103] Xinghao Pan, Stefanie Jegelka, Joseph E Gonzalez, Joseph K Bradley, and Michael I Jordan. Parallel double greedy submodular maximization. In *Advances in Neural Information Processing Systems*, pages 118–126, 2014.

[104] David Peleg. Distributed computing. *SIAM Monographs on discrete mathematics and applications*, 5, 2000.

[105] Robin Pemantle and Yuval Peres. Concentration of Lipschitz Functionals of Determinantal and Other Strong Rayleigh Measures. *Combinatorics, Probability and Computing*, 23:140–160, 1 2014.

[106] Yuval Peres and Bálint Virág. Zeros of the iid gaussian power series: a conformally invariant determinantal process. *Acta Mathematica*, 194(1):1–35, 2005.

[107] Francesca Petralia, Vinayak Rao, and David B Dunson. Repulsive mixtures. In *Advances in neural information processing systems*, pages 1889–1897, 2012.

[108] Friedrich Pukelsheim. *Optimal design of experiments*, volume 50. siam, 1993.

[109] Patrick Rebeschini and Amin Karbasi. Fast mixing for discrete point processes. In *COLT*, pages 1480–1500, 2015.

[110] Mark Rudelson. Random vectors in the isotropic position. *Journal of Functional Analysis*, 164(1):60–72, 1999.

[111] Antonello Scardicchio, Chase E Zachary, and Salvatore Torquato. Statistical properties of determinantal point processes in high-dimensional euclidean spaces. *Physical Review E*, 79(4):041108, 2009.

[112] Tomoyuki Shirai and Yoichiro Takahashi. Random point fields associated with certain fredholm determinants i: fermion, poisson and boson point processes. *Journal of Functional Analysis*, 205(2):414–463, 2003.

[113] Mohit Singh and Weijun Xie. Approximation algorithms for d-optimal design. 2018.

[114] Alexander Soshnikov. Determinantal random point fields. *Russian Mathematical Surveys*, 55(5):923, 2000.

[115] Damian Straszak and Nisheeth K. Vishnoi. Real stable polynomials and matroids: optimization and counting. In *STOC*, pages 370–383, 2017.

[116] Terence Tao. 254a, notes 3a: Eigenvalues and sums of hermitian matrices, 2010. https://terrytao.wordpress.com/2010/01/12/254a-notes-3a-eigenvalues-and-sums-of-hermitian-matrices.

[117] Kai Wei, Rishabh Iyer, and Jeff Bilmes. Fast multi-stage submodular maximization. In *International conference on machine learning*, pages 1494–1502, 2014.

[118] Baoyuan Wu, Fan Jia, Wei Liu, and Bernard Ghanem. Diverse image annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2559–2567, 2017.

[119] Jin-ge Yao, Feifan Fan, Wayne Xin Zhao, Xiaojun Wan, Edward Y Chang, and Jianguo Xiao. Tweet timeline generation with determinantal point processes. In *AAAI*, pages 3080–3086, 2016.

[120] Sepehr Abbasi Zadeh, Mehrdad Ghadiri, Vahab S Mirrokni, and Morteza Zadimoghaddam. Scalable feature selection via distributed diversity maximization. In *AAAI*, pages 2876–2883, 2017.

[121] Cheng Zhang, Hedvig Kjellström, and Stephan Mandt. Determinantal point processes for mini-batch diversification. In *33rd Conference on Uncertainty in Artificial*

Intelligence, UAI 2017, Sydney, Australia, 11 August 2017 through 15 August 2017. AUAI Press Corvallis, 2017.