A Polynomial Time MCMC Method for Sampling from Continuous Determinantal Point Processes

Alireza Rezaei [1] **Shayan Oveis Gharan** [1]

Abstract

We study the Gibbs sampling algorithm for discrete and continuous k-determinantal point processes. 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.

1. Introduction

Determinantal Point Processes (DPPs) are a family of probability distributions that were first introduced in quantum physics to model particles with repulsive interactions (Macchi, 1975). They have been extensively studied by mathematicians, as they naturally appear in many contexts including non-intersecting random walks (Johansson, 2002), random spanning trees (Burton & Pemantle, 1993), eigenvalues of random matrices (Mehta & Gaudin, 1960; Ginibre, 1965), and zero-set of Gaussian analytic functions (Peres & Virág, 2005). Studying DPPs in machine learning was initiated by the work of (Kulesza et al., 2012) who exploited the repulsive behavior of them to model *diversity* in real world tasks. Following this insight, DPPs have found numerous applications in machine learning, including document summarization, building news story timelines, tweet generation (Kulesza et al., 2012; Chao et al., 2015; Yao et al., 2016; Gillenwater et al., 2012), video summarization, pose estimation, and image search (Kulesza et al., 2012; Kulesza & Taskar, 2010; Gong et al., 2014; Mirzasoleiman et al., 2017).

Formally, let $L \in \mathbb{R}^{n\times n}$ be a positive semi-definite (PSD) matrix. Given a set Ω of size n, matrix L defines a probability distribution over the set of subsets of Ω which is known as a *discrete* DPP, and is characterized as follows: Letting π denote the DPP, for every subset $S \subseteq \Omega$, we have

$$\pi(S) \propto \det(L_S) = \frac{\det(L_S)}{\sum_{T\subseteq[n]}\det(L_T)},$$

where L_S is the principal submatrix of L indexed by the corresponding elements of set S. In this setting, matrix L is called the *kernel* of the DPP and Ω is known as the domain, i.e. we say π is a DPP on domain $[n]$ defined by kernel L. For an integer $k \leq n$, a k-DPP defined by L is the restriction of π to subsets of Ω of size k. k-DPPs are useful when the size of the desired diverse set that we aim to select is known in advance.

DPPs and k-DPP have been also studied on continuous spaces. *Continuous* DPPs can be defined by extending the above definition by considering a subset $\Omega \subseteq \mathbb{R}^d$ as the domain and letting the kernel be a PSD integral operator $L : L^2(\Omega) \to L^2(\Omega)$ (with some additional constraints, see subsection 2.1 for more details). In the other words, the continuous DPP defined by L is a probability distribution on the finite subsets of Ω whose PDF function, denoted by f, is given by the following for any integer k and any $\{x_1, \ldots, x_k\} \subset \Omega$:

$$f(\{x_1,\ldots,x_k\}) \propto \det_L(x_1,\ldots,x_k),$$

where the $\det_L$ notation indicates the determinant of the k by k matrix formed by $L(a,b)$ values for $a,b \in \{x_1,\ldots,x_k\}$. Similar to the discrete case, the continuous k-DPPs are defined by restricting DPPs. These distributions have recently gained more attention in ML as in many applications of DPPs, the domain is naturally a continuous space; in particular, such applications arise in learning parameters of generative mixture models (Petralia et al., 2012; Kwok

[*]Equal contribution [1]Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA, USA. Correspondence to: Alireza Rezaei <arezaei@cs.washington.edu>, Shayan Oveis Gharan <shayan@cs.washington.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

& Adams, 2012; Hafiz Affandi et al., 2013), and tuning hyper-parameters of neural networks (Dodge et al., 2017); also, see (Lavancier et al., 2015) for their applications in statistics and (Biscio & Lavancier, 2016) for the connections to repulsive systems.

The wide range of applications of DPPs motivates designing efficient learning and inference primitives for them. In the discrete setting, efficient algorithms have been discovered for sampling (Hough et al., 2006; Li et al., 2015; Deshpande & Rademacher, 2010; Anari et al., 2016), marginalization (Borodin & Rains, 2005), conditioning (Kulesza et al., 2012), and many other inference tasks. On the other hand, in the continuous domain, despite previous efforts (Scardicchio et al., 2009; Lavancier et al., 2012; Hafiz Affandi et al., 2013; Hennig & Garnett, 2016), there has been much less success.

In this work we study sampling algorithms for continuous k-DPPs. Note that, devising such algorithms in full generality is not well-defined, since such an algorithm would depend on how the input kernel is represented. Therefore, the main question is that, in what settings the sampling can be done efficiently. We show that, given a conditional sampling oracle for a kernel L (defined in 1.1), one can simulate a *lazy* random scan Gibbs sampler to generate samples from the continuous k-DPP defined by L, efficiently. We also prove that for several kernels of interest, one can construct the conditional sampler efficiently.

1.1. Results

First, we formally define the Gibbs sampler chain that we use for sampling from continuous k-DPPs [1]. Let π be the input k-DPP defined by a kernel $L : \Omega \times \Omega \to \mathbb{R}$ for some $\Omega \in \mathbb{R}^d$. If the current state of the chain is $\{x_1, \dots, x_k\} \subset \Omega$, it evolves as follows: It stays at the current state with probability half. Otherwise, A point $x_i \in \{x_1, \dots, x_k\}$ is chosen uniformly at random, and is replaced by $y \in \Omega$ sampled from the conditional distribution whose PDF, f, is defined by

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

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. Before stating the results, we need a few definitions. For two probability distributions μ, π which are defined on the same space, we use $d_{\mathrm{TV}}(\pi, \mu)$ to denote their total variation distance, that is

$$d_{\mathrm{TV}}(\pi, \mu) = \sup_A |\mu(A) - \pi(A)|$$

where A ranges over all events. Moreover, for a Markov chain started from a distribution μ_0 and for any $\epsilon > 0$, we let $\tau_{\mu_0}(\epsilon)$ denote the ϵ-mixing time of the chain defined by

$$\tau_{\mu_0}(\epsilon) = \min\{t \mid d_{\mathrm{TV}}(\mu_t, \pi) \leq \epsilon\},$$

where μ_t is the distribution of the chain after t steps. Moreover, an ϵ-approximate sample from a distribution π refers to a random sample from a distribution μ with $d_{\mathrm{TV}}(\mu, \pi) \leq \epsilon$.

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 1.1. *If we run the Gibbs sampler for a k-DPP π, starting from an arbitrary distribution μ_0, then 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 PDFs for π and μ_0.

Applications for Discrete k-DPPs. In this case, to find a proper starting state of the chain, we can use a the greedy algorithm for determinant maximization which returns a state (a subset of Ω of size k) S with $\pi(S) \geq \frac{1}{k!}$ (Çivril & Magdon-Ismail, 2009); 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 $O(n) \cdot \mathrm{poly}(k)$. This is not an improvement over the currently known running times for sequential algorithms. 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 1.2. *Given access to n^δ processors for some $\delta > 0$, an approximate sample of a k-DPP defined on domain of size n can be generated in time $O(n^{1-\delta}) \cdot poly(k)$.*

On the other hand, for continuous k-DPPs, to turn the above result into an efficient algorithm, finding a "good" starting distribution μ which makes the log variance term in the bound of Theorem 1.1 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 1.1. *For a kernel $L : \Omega \times \Omega \to \mathbb{R}$, and a finite subset $S \subset \Omega$, we define the (S)-conditional distribution of*

[1] The analogous chain can be defined for discrete k-DPPs

L to be a distribution on Ω, defined by the following PDF

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

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

It is straight-forward to see that taking a step of the Gibbs sampler of the k-DPP defined by L from the state $x_1, \ldots, x_k$ is equivalent to removing a point x_i, for some $1 \leq i \leq k$, and generating a sample from $\mathcal{D}_L(\{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k\})$. So 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 starting distribution which gives the following guarantee.

Theorem 1.3. *Consider a continuous k-DPP π defined by a kernel L. Given access to $\mathrm{CD}_L(i)$ oracles for all $0 \leq i \leq k-1$, we can find a distribution μ and simulate the Gibbs sampler started at μ to get ϵ-approximate samples of π in $O(k^5 \log \frac{k}{\epsilon})$ steps.*

Therefore, the task of sampling from a continuous k-DPP boils down to have algorithms to sample from conditional 1-DPPs ($\mathcal{D}_L(.)$ distributions), which is a much 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 (Lavancier et al., 2012), with uniform distribution on the domain as the proposal distribution. Analyzing the rejection sampler and combining that with Theorem 1.3, we get the following.

Theorem 1.4. *Let L be a kernel on a bounded domain Ω, and we have an oracle which can generate uniform samples from Ω. For any integer k and any $\epsilon > 0$, an ϵ-approximate sample from the k-DPP defined by L can be generated by*

$$O(k^5 \log \frac{1}{\epsilon}) \cdot \frac{M \cdot \mathrm{vol}(\Omega)}{\sum_{i=k}^{\infty} \lambda_i}$$

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

For some of the widely used kernels such as a Gaussian kernel defined by $L(x, y) = \exp((x-y)^\intercal \Sigma^{-1}(x-y))$, the $L(x, x)$ is a constant for all x and so $\mathrm{tr}(L) = \int_\Omega L(x, x) \propto \mathrm{vol}(\Omega)$ and the bound in the above theorem becomes proportional to $\frac{\mathrm{tr}(L)}{\sum_{i=k}^{\infty} \lambda_i}$. 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.

1.2. Comparison with Previous Work

In general, two families of algorithms have been considered to generate samples from discrete DPPs: spectral algorithms, and MCMC methods. Spectral algorithms are based on the work of (Hough et al., 2006) which given the eigen-decomposition of the kernel suggests a two-step sampling procedure: 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 (Lavancier et al., 2012; Hafiz Affandi et al., 2013), a heuristic is to find a finite rank approximation of the original kernel. (Hafiz Affandi et al., 2013) 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.

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, (Hafiz Affandi et al., 2013) 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 particular, it is shown in (Anari et al., 2016) that the natural Metropolis-Hastings algorithm for k-DPPs gives an efficient sampling method running in time $O(n)\mathrm{poly}(k)$, where n is the size of the domain, and it has been extended to an algorithm for sampling from discrete DPPs in time $\tilde{O}(n^3)$ (Li et al., 2015; 2016). However, to best of our knowledge, such an MCMC algorithm with a provable guarantee is not known for the continuous setting; in an attempt, (Hafiz Affandi et al., 2013) 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 (Hennig & Garnett, 2016) 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 [2].

2. Background

Let $\mathbb{R}^d$ denote the d-dimensional euclidean space. For a compact body $\Omega \subset \mathbb{R}^d$, we use $\mathrm{vol}(\Omega)$ to denote its volume (with respect to the standard Lebesgue measure). For a $(d-1)$-dimensional body in $\mathbb{R}^d$ like an sphere, we use vol to refer to the surface area. An integrable function $f : \Omega \to R$ belongs to $L^2(\Omega)$, if $\int_\Omega |f(x)|^2 dx < \infty$. Also throughout the paper, whenever we have a distribution it is defined with respect to the standard Lebesgue measure on $\mathbb{R}^d$, or the corresponding product measures for $\mathbb{R}^d \times \mathbb{R}^d \cdots \times \mathbb{R}^d$.

2.1. Continuous k-DPPs

Let $\Omega \subset \mathbb{R}^d$ be a closed set. A k-DPP on domain Ω is defined by a continuous function $L : \Omega \times \Omega$ (a.k.a. *kernel*) with the following properties: It is symmetric which means $L(x, y) = L(y, x)$ for any $x, y \in \Omega$. Its corresponding Hilbert-Schmidt operator defined by $T_L(f)(x) = \int_\Omega L(x, y)f(y)$ for any $f \in \ell^2(\Omega)$, is a PSD operator, and finally $\int_\Omega \int_\Omega |L(x, x)| < \infty$.

We only work with kernels with the aforementioned properties. Such a kernel generalizes finite PSD matrices in many aspects. In particular, the classical Mercer's theorem implies that there is an orthonormal basis of eigen-functions $\{e_j\}_{i\in\mathbb{N}}$ of $L^2(\Omega)$ with non-negative eigenvalues $\{\lambda_i\}_{i\in\mathbb{N}}$ such that for any $x, y \in \Omega$, $L(x, y) = \sum_{i=0}^{\infty} \lambda_i e_i(x)e_i(y)$. It is also follows from the continuity of L that for any finite set of points $x_1, \ldots, x_k \in \Omega$, the $k \times k$ matrix $\{L(x_i, x_j)\}_{1\le i,j\le k}$ is a PSD matrix. We use the notation $\det_L(x_1, \ldots, x_k)$ to denote its determinant. Whenever, the kernel is clear from the context, we may drop the subscript.

If π is a probability distribution, we use f_π to refer to the corresponding probability density function (PDF). Continuous k-DPPs can be defined formally as follows.

Definition 2.1 (Continuous k-DPP)**.** *The continuous k-DPP defined by kernel L on domain Ω is a probability distribution π on subsets of Ω of size k with PDF function* $f_\pi(x_1, \ldots, x_k) \propto \det_L(x_1, \ldots, x_k)$

We refer interested readers to (Hough et al., 2006) for more on continuous DPPs.

2.2. Markov Chains and Mixing Time

A Markov chain on a measurable state space can be defined similarly to that of on a finite space. In this section we give a high level overview, and provide more details is provided in the supplementary. Let $\mathcal{M}(\mathcal{S}, P)$ be a Markov chain where the Lebesgue measurable set $\mathcal{S} \subset \mathbb{R}^d$ is the state space and $P : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^+$ is the transition probability kernel. So if μ is the current distribution of the chain, the distribution of the chain after one step is a probability distribution $P(\mu, .)$ on $\mathcal{S}$ which is defined by

$$\forall \text{measurable } B \subset \mathcal{S}, P(\mu, B) = \int_{\mathcal{S}} \int_B f_\mu(x)P(x, y)dxdy.$$

π is a stationary measure of the chain if $P(\pi, .) = \pi(.)$. Then $\mathcal{M}$ is denoted by $(\mathcal{S}, P, \pi)$. The chain is reversible if for any pair of states $x, y \in \mathcal{S}$, $\pi(dx)P(x, dy) = \pi(dy)P(y, dx)$. It is easy to verify that the random scan Gibbs sampler for a k-DPP is reversible and its stationary measure is the corresponding k-DPP distribution. A chain is also called *lazy* if at each step it stays at the current state with probability at least half.

To bound the mixing time of the our Gibbs sampler and prove , we analyze its *Poincaré* constant (a.k.a. spectral gap).

Theorem 2.1 ((Kontoyiannis & Meyn, 2012))**.** *For any reversible, lazy, π-irreducible Markov chain $\mathcal{M}$, if the spectral gap $\lambda > 0$, then starting the chain from any distribution μ (which is absolutely continuous with respect to π), after t steps we have*

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

In the above, $\mathrm{var}_\pi(f_\mu/f_\pi) := \int_{\mathcal{S}} (\frac{f_\mu(x)}{f_\pi(x)})^2 f_\pi(x)dx$.

Instead of directly bounding the spectral gap, we bound the *conductance* of the chain and apply the well-known Cheeger's inequality. For any subset B, the *conductance* of a set B is defined by $\phi(B) := \frac{Q(B,\overline{B})}{\pi(B)}$ where $Q(B, \overline{B})$ is known as the *ergodic flow* leaving B and is equal to $Q(B, \overline{B}) = \int_B \int_{\overline{B}} f_\pi(x)P(x, y)dxdy$. The conductance of the chain is then defined by $\phi(\mathcal{M}) = \min_{0<\pi(B)\le\frac{1}{2}} \phi(B)$.

Theorem 2.2 ((Lawler & Sokal, 1988))**.** *For a chain $\mathcal{M}$ defined on a general state space with spectral gap λ we have* $\frac{\phi(\mathcal{M})^2}{2} \le \lambda \le 2\phi(\mathcal{M})$.

The analogues of the above results also hold on a discrete domain, and we can use them to convert a lower bound on the conductance to an upper bound bound on the mixing time for a discrete k-DPP Gibbs sampler.

3. Gibbs Sampling for Discrete k-DPP

Let $\mathcal{M} = (\mathcal{S}, P, \pi)$ be the Gibbs sampler for a discrete k-DPP π defined on domain $[n]$, that is the state space is

[2] 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 1.1)

$\mathcal{S} = \binom{[n]}{k}$ and P denotes the transition probability matrix. Recall that the conductance is defined by $\Phi(\mathcal{M}) = \min_{S \subset \mathcal{S}: \pi(S) \leq \frac{1}{2}} \frac{Q(S,\overline{S})}{\pi(S)}$, where for $x, y \in \Omega$, $Q(x,y) = \pi(x)P(x,y)$ and $Q(S,\overline{S}) = \sum_{x \in S, y \notin S} Q(x,y)$. We prove the following.

Theorem 3.1. *Let $\mathcal{M}$ be the Gibbs sampler chain for an arbitrary discrete k-DPP, then we have $\phi(\mathcal{M}) \gtrsim \frac{1}{k^2}$.*

The proof of the above theorem follows an inductive approach similar to (Mihail, 1992) and (Anari et al., 2016) which uses the properties of k-DPPs, and in particular their negative correlation. Let S be a sample generated of π. Negative correlation says that for any arbitrary pair of elements $x, y \in \Omega$, we have

$$\mathbb{P}(x \in S | y \in S) \leq \mathbb{P}(x \in S).$$

Here, due the page limit we only include a sketch of the proof, and delegate the formal proof to the supplement.

Combining this theorem with the discrete versions of Theorem 2.1 and Theorem 2.2 establishes our main result for discrete k-DPPs and proves Theorem 1.1 for discrete k-DPPs.

Proof Sketch. WLOG, suppose that $\Omega = [n]$ for some integer n. Let $\mathcal{S} = \mathcal{S}_n \cup \mathcal{S}_{\overline{n}}$ where $\mathcal{S}_n$ and $\mathcal{S}_{\overline{n}}$ denote the subset of states which contain element n, do not contain n, respectively. Also define π_n and $\pi_{\overline{n}}$ to be conditioning of π to $\mathcal{S}_n$ and $\mathcal{S}_{\overline{n}}$, i.e. for any $x \in \mathcal{S}_n$, $\pi_n(x) = \frac{\pi(x)}{\pi(\mathcal{S}_n)}$ and for any $y \in \mathcal{S}_{\overline{n}}$, $\pi_{\overline{n}}(y) = \frac{\pi(y)}{\pi(\mathcal{S}_{\overline{n}})}$. It follows that $\pi_n, \pi_{\overline{n}}$ can be identified with a $(k-1)$-DPP, k-DPP supported on $\mathcal{S}_n, \mathcal{S}_{\overline{n}}$

Now, fix a subset $S \subset \mathcal{S}$ with $\pi(S) \leq \frac{1}{2}$. We need to show $Q(S,\overline{S}) \geq \frac{\pi(S)}{Ck^2}$. Letting $S_n = S \cap \mathcal{S}_n$ and $S_{\overline{n}} = S \cap \mathcal{S}_{\overline{n}}$, we have

$$\begin{aligned} Q(S,\overline{S}) = & Q(S_n, \mathcal{S}_n \setminus S_n) + Q(S_{\overline{n}}, \mathcal{S}_{\overline{n}} \setminus S_{\overline{n}}) \\ & + Q(S_n, \mathcal{S}_{\overline{n}} \setminus S_{\overline{n}}) + Q(S_{\overline{n}}, \mathcal{S}_n \setminus S_n). \end{aligned} \tag{1}$$

To bound the contribution of the edges of $(S,\overline{S})$ with both endpoints in $\mathcal{S}_n$ or $\mathcal{S}_{\overline{n}}$ we use induction; More precisely, we induct on $k+n$ and consider the implication of Theorem 3.1 for π_n and $\pi_{\overline{n}}$. Then, we can use the induction hypothesis to bound $(S_n, \mathcal{S}_n \setminus S_n)$ and $Q(S_{\overline{n}}, \mathcal{S}_{\overline{n}} \setminus S_{\overline{n}})$. [3]. In order to bound the edges across the cut, we use the negative correlation property for k-DPPs and in particular the following consequence of that appeared in (Anari et al., 2016).

Lemma 3.2. *For any subset $A \subseteq \mathcal{S}_n$,*

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

In the above is the set of neighbors of A in $\mathcal{S}_{\overline{n}}$, i.e. $N_{\overline{n}}(A) = \{y \in \mathcal{S}_{\overline{n}} \mid \exists x \in A : P(x,y) > 0\}$.

Algorithm 1 Choosing a starting state

```
Input: $\mathrm{CD}_L(i)$ oracles of $L$ for $0 \leq i \leq k-1$.
Let $S = \emptyset$.
for $i$ from $0$ to $k-1$ do
    Use the $\mathrm{CD}_L(i)$ oracle to generate a sample $x_i$ and add $x_i$ to $S$.
end for
Return $S$.
```

4. Gibbs Sampling for Continuous k-DPP

In this section we analyze the mixing time our Gibbs sampler for continuous k-DPPs and prove Theorem 1.3. The first step is to show that the analogous bound of Theorem 3.1 on conductance also holds for the Gibbs sampler for a continuous k-DPP. Throughout the section, we fix π to denote the k-DPP defined by a kernel $L : \Omega \times \Omega \to \mathbb{R}$, where the domain Ω is a closed subset of $\mathbb{R}^d$.

Theorem 4.1. *Let $\mathcal{M}$ be the Gibbs sampler for π, then $\phi(\mathcal{M}) \gtrsim \frac{1}{k^2}$.*

The main idea of the proof is to approximate the given continuous k-DPP by a sequence of discrete k-DPPs which are obtained by discretizating the underlying domain, and apply Theorem 3.1 on these approximated discrete kernels. The key points that we use in the argument are: First, the bound in Theorem 3.1 is independent of the size of the domain, so we can afford to consider discrete k-DPPs with arbitrarily large number of points, while maintaining the $\frac{1}{k^2}$ bound on the conductance. Secondly, continuity of L implies that the discrete k-DPPs in this sequence approach the given continuous k-DPP distribution in the limit. Therefore, a limiting argument would finish the proof of Theorem 4.1. The details of the proof are provided in the supplement.

Having this bound on the conductance, we can apply Theorem 2.2 to get $\lambda_{\mathcal{M}} \gtrsim \frac{1}{k^4}$, where $\lambda_{\mathcal{M}}$ is the spectral gap of $\mathcal{M}$. Now, we can use Theorem 2.1 to deduce Theorem 1.1. Next, we propose an algorithm that given access to conditional samplers of L, finds the proper starting distributions, and thus we conclude Theorem 1.3.

4.1. Finding a Warm Start

In this subsection, we are assuming we have access to $\mathrm{CD}_L(i)$ oracles of L for any $0 \leq i \leq k-1$. To find a starting state, we use Algorithm 1 which is the continuous version of a greedy algorithm analyzed at (Deshpande & Varadarajan, 2007) for approximate volume sampling. Algorithm 1 is a randomized algorithm which returns a single state of $\mathcal{M}$, i.e. a subset of k-points of the domain. We

[3] Note that the Gibbs chains defined for π_n and $\pi_{\overline{n}}$ induce different transition probabilities than the restriction of the Gibbs sampler for π on $\mathcal{S}_n$ and $\mathcal{S}_{\overline{n}}$. So the induction hypothesis cannot be applied directly to bound these two terms.

Algorithm 2 An Algorithm for Conditional Sampling

```
**Input:** A set of $k$ points $x_1, \dots, x_k \in \Omega$.
**Output:** A sample from $\mathcal{D}_L(\{x_1, \dots, x_k\})$.
Let $M$ be a number such that $M > \sup_{z \in C} L(z, z)$.
**repeat**
    Draw a uniform sample $x$ from $\Omega$ and a uniform number $u$ from $[0, 1]$.
    If $u \leq \frac{\det_L(x_1,\dots,x_k,x)}{M \cdot \det_L(x_1,\dots,x_k)}$, accept and return $x$.
**until** A sample is accepted.
```

can prove the following guarantee for the distribution of the output of the algorithm.

Lemma 4.2. *Let μ_0 be the probability distribution of the output of Algorithm 1. Also let f_{μ_0} and f_π denote the PDF for μ_0 and π. Then for any $\{x_1, \dots, x_k\} \subset \Omega$,*

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

The proof essentially follows from a similar argument in (Deshpande & Varadarajan, 2007). We provide it in the supplement. We use this lemma to bound $\mathrm{var}_\pi(\frac{f_{\mu_0}}{f_\pi})$ which appears in our bound for the mixing time. In particular, the lemma implies

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

Combining this bound with Theorem 1.1 yields the proof of Theorem 1.3.

5. A Simple Conditional Sampler

Theorem 1.3 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 2. Let Ω be the domain of the k-DPP. The algorithm assumes that Ω is bounded and we have an oracle to generate uniform samples from Ω.

Correctness of the algorithm. We want to show that Algorithm 2 generate a sample of $\mathcal{D}_L(\{x_1, \dots, x_k\})$. Let Φ denote the distribution of the output and f_ϕ be its PDF. It suffices to show that for any $z \in \Omega$, $f_\phi(z) \propto \det_L(x_1, \dots, x_k, z)$. By the definition of the algorithm, it is enough to verify $\frac{\det_L(x_1,\dots,x_k,z)}{M \cdot \det_L(x_1,\dots,x_k)} \leq 1$ which follows from $\frac{\det_L(x_1,\dots,x_k,z)}{\det_L(x_1,\dots,x_k)} \leq L(z,z)$ and $M > L(z,z)$. The former holds, since if we write the PSD matrix given by restricting L to $x_1, \dots, x_k, z$ 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 $\frac{\det_L(x_1,\dots,x_k,z)}{\det_L(x_1,\dots,x_k)}$ is equal to the squared of distance of that vector from the plane spanned by vectors corresponding to $x_1, x_2, \dots, x_k$. Therefore, it remains to analyze the running time,

5.1. Analyzing the Running Time

Let T be a random variable which indicates the expected number of uniform samples generated from Ω 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 $\lambda_0 \geq \lambda_1 \geq \dots$ be eigenvalues of L. The following relates $\mathbb{E}[T]$ to the eigenvalues.

Lemma 5.1. *For any set of points $x_1, \dots, x_k$ as the input of Algorithm 2, we have*

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

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

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

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 $\mathbb{E}_{y \sim \mu} \frac{\det_L(\boldsymbol{x}+y)}{\det_L(\boldsymbol{x})} \geq \frac{\sum_{i=k}^{\infty} \lambda_i(L)}{\mathrm{vol}(\Omega)}$ to prove the lemma. By Mercer theorem, for any $x \in \Omega$, there exists a function (feature map) $f_x : \Omega \to \mathbb{R}$ such that for any $y \in \Omega$, $L(x,y) = \langle f_x, f_y \rangle$. Now, for any $y \in \Omega$, define $\mathcal{E}(y) = \Pi_{\langle f_{x_1}, \dots, f_{x_k} \rangle^\perp}(f_y)$, be the projection of f_y onto the space orthogonal to functions corresponding to $x_1, \dots, x_k$. Then, by definition $\frac{\det_L(\boldsymbol{x}+y)}{\det_L(\boldsymbol{x})} = \|\mathcal{E}(y)\|^2$, where recall that $\boldsymbol{x} = \{x_1, \dots, x_k\}$. It implies

$$\mathbb{E}_{y \sim \mu} \frac{\det_L(\boldsymbol{x}+y)}{\det_L(\boldsymbol{x})} = \mathbb{E}_{y \sim \mu} \|\mathcal{E}(y)\|^2 = \frac{\mathrm{tr}(\mathcal{E})}{\mathrm{vol}(\Omega)} \tag{2}$$

for the kernel $\mathcal{E} : \Omega \times \Omega \to \mathbb{R}$ 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 $\mathrm{CD}_L(.)$ oracles for L and combining that with Theorem 1.3 immediately implies Theorem 1.4. Next, we analyze the bound of Lemma 5.1 more precisely for special kernels defined on a sphere, and show it gives an efficient sampling algorithm for k-DPPs defined by spherical Gaussian.

5.2. Complexity of Algorithm 2 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 (Minh et al., 2006).

Theorem 5.2 ((Minh et al., 2006)). *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, (Minh et al., 2006) gives explicit formula for spherical Gaussians. The following lemma states its implication for bounding the complexity of Algorithm 2. The proof is provided in the supplement. Note that generating a uniform sample from a d-dimensional sphere can be done in $O(d)$ time, so combining the lemma with Theorem 1.4 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 2, we prove the following.

Lemma 5.3. *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 2, 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 $\mathbb{E}[T] = O(1)$.*

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

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 $\mathbb{R}^d$.

Algorithm 3 Gibbs Sampler for Continuous k-DPPs

```
Input: A kernel $L : \Omega \times \Omega \to \mathbb{R}$ along with an oracle which generates uniform samples form $\Omega$.
Let $S = \emptyset$.
Let $M$ be a number such that $M > \sup_{z \in \Omega} L(z,z)^4$ .
for $i$ from 0 to $k-1$ do
  repeat
    Draw a uniform sample $x$ from $\Omega$ 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 $S = S \cup x$.
  until A sample is accepted
end for
Let $\tau = \tilde{O}(k^5 \log \frac{1}{\epsilon})$.
for $\tau$ iterations do
  Let $S = \{x_1, \ldots, x_k\}$ and pick an uniform random integer $0 \leq i \leq k-1$. Set $S = S - x_i$
  repeat
    Draw a uniform sample $x$ from $\Omega$ 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 $S = S \cup x$.
  until A sample is accepted
end for
Return $S$.
```

Simulation Setup: For a fixed kernel, we use the rejection sampler described in Algorithm 2 as the conditional sampler of the kernel. To do the sampling from the continuous k-DPP defined by the kernel, we first run Algorithm 1 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 3. 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 (Brooks & Gelman, 1998). 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 1 and Figure 2. 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 1, 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 2, 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 2, 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.

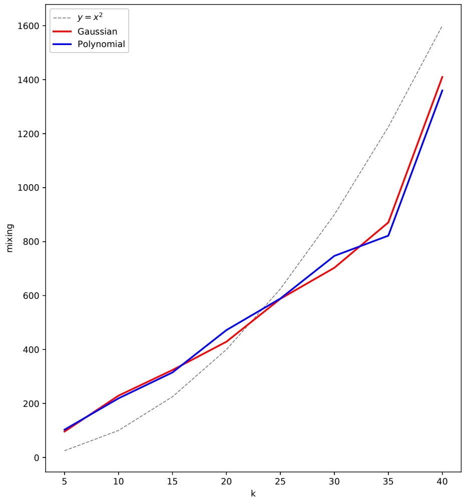

Figure 1. Empirical mixing time for different values of k while dimension and other parameters are fixed ($d = 40$, $\sigma = 1$ and $b = 5$)

.

7. Conclusion and Future Directions

We studied a Gibbs sampling scheme for sampling from continuous and discrete k-DPPs, and proved that the mixing time is only a function k. In the discrete case, this naturally leads to sublinear time algorithms for sampling in distributed models of computation. It is an interesting open question to extend these ideas to get efficient distributed algorithms for sampling from DPPs, as well.

On the other hand for continuous k-DPP, in order to get an efficient algorithm and simulate the chain, we need to generate samples from 1-conditional distributions associated to the kernel. To do that, we analyzed a simple rejection sampler, and show that when the spectrum of the kernel is not concentrated on the k largest eigenvalues, it is efficient. However, it remains an open question to design algorithms to efficiently simulate the chain for larger classes of kernels.

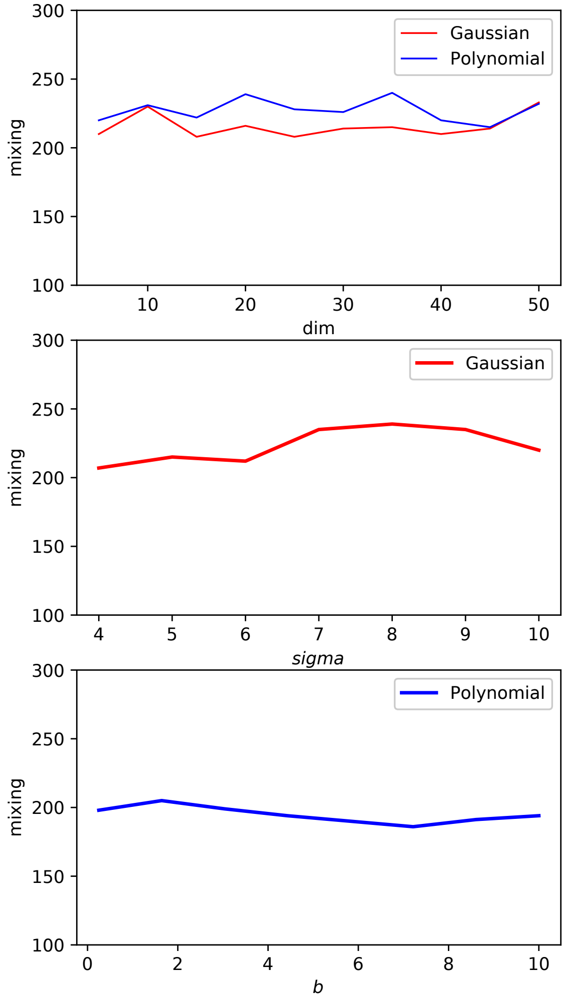

Figure 2. Plots of the empirical mixing time for a fixed k and varying σ (middle plot), b (bottom plot), and d (top plot).

Acknowledgments

The authors are supported by the NSF grant CCF-1552097 and ONR-YIP grant N00014-17-1-2429. Part of this work was done while the authors were visiting the Simons Institute for the Theory of Computing.

References

Anari, N., Gharan, S. O., and Rezaei, A. Monte carlo markov chain algorithms for sampling strongly rayleigh distributions and determinantal point processes. In *Conference on Learning Theory*, pp. 103–115, 2016.

Biscio, C. A. N. and Lavancier, F. Quantifying repulsiveness of determinantal point processes. *Bernoulli*, 22(4):2001–2028, 11 2016. URL `https://doi.org/10.3150/15-BEJ718`.

Borodin, A. and Rains, E. M. Eynard–mehta theorem, schur process, and their pfaffian analogs. *Journal of statistical physics*, 121(3):291–317, 2005.

Brooks, S. P. and Gelman, A. General methods for monitoring convergence of iterative simulations. *Journal of computational and graphical statistics*, 7(4):434–455, 1998.

Burton, R. and Pemantle, R. Local characteristics, entropy and limit theorems for spanning trees and domino tilings via transfer-impedances. *The Annals of Probability*, pp. 1329–1371, 1993.

Chao, W.-L., Gong, B., Grauman, K., and Sha, F. Large-margin determinantal point processes. In *UAI*, pp. 191–200, 2015.

Çivril, A. and Magdon-Ismail, M. On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoretical Computer Science*, 410(47-49):4801–4811, 2009.

Deshpande, A. and Rademacher, L. Efficient volume sampling for row/column subset selection. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pp. 329–338. IEEE, 2010.

Deshpande, A. and Varadarajan, K. Sampling-based dimension reduction for subspace approximation. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pp. 641–650. ACM, 2007.

Dodge, J., Jamieson, K., and Smith, N. A. Open loop hyperparameter optimization and determinantal point processes. *arXiv preprint arXiv:1706.01566*, 2017.

Gillenwater, J., Kulesza, A., and Taskar, B. Discovering diverse and salient threads in document collections. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 710–720. Association for Computational Linguistics, 2012.

Ginibre, J. Statistical ensembles of complex, quaternion, and real matrices. *Journal of Mathematical Physics*, 6(3):440–449, 1965. URL `https://doi.org/10.1063/1.1704292`.

Gong, B., Chao, W.-L., Grauman, K., and Sha, F. Diverse sequential subset selection for supervised video summarization. In *Advances in Neural Information Processing Systems*, pp. 2069–2077, 2014.

Hafiz Affandi, R., Fox, E. B., and Taskar, B. Approximate inference in continuous determinantal point processes. *arXiv preprint arXiv:1311.2971*, 2013.

Hennig, P. and Garnett, R. Exact sampling from determinantal point processes. *arXiv preprint arXiv:1609.06840*, 2016.

Hough, J. B., Krishnapur, M., Peres, Y., Virág, B., et al. Determinantal processes and independence. *Probability surveys*, 3:206–229, 2006.

Johansson, K. Non-intersecting paths, random tilings and random matrices. *Probability theory and related fields*, 123(2):225–280, 2002.

Kontoyiannis, I. and Meyn, S. P. Geometric ergodicity and the spectral gap of non-reversible markov chains. *Probability Theory and Related Fields*, pp. 1–13, 2012.

Kulesza, A. and Taskar, B. Structured determinantal point processes. In *Advances in neural information processing systems*, pp. 1171–1179, 2010.

Kulesza, A., Taskar, B., et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

Kwok, J. T. and Adams, R. P. Priors for diversity in generative latent variable models. In *Advances in Neural Information Processing Systems*, pp. 2996–3004, 2012.

Lavancier, F., Møller, J., and Rubak, E. H. Statistical aspects of determinantal point processes. Technical report, Department of Mathematical Sciences, Aalborg University, 2012.

Lavancier, F., Møller, J., and Rubak, E. Determinantal point process models and statistical inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(4):853–877, 2015. ISSN 1467-9868. URL `http://dx.doi.org/10.1111/rssb.12096`.

Lawler, G. F. and Sokal, A. D. Bounds on the l^2 spectrum for markov chains and markov processes: a generalization of cheeger's inequality. *Transactions of the American mathematical society*, 309(2):557–580, 1988.

Li, C., Jegelka, S., and Sra, S. Efficient sampling for k-determinantal point processes. *arXiv preprint arXiv:1509.01618*, 2015.

Li, C., Jegelka, S., and Sra, S. Fast sampling for strongly rayleigh measures with application to determinantal point processes. *arXiv preprint arXiv:1607.03559*, 2016.

Macchi, O. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7(1):83–122, 1975.

Mehta, M. L. and Gaudin, M. On the density of eigenvalues of a random matrix. *Nuclear Physics*, 18:420–427, 1960.

Mihail, M. On the expansion of combinatorial polytopes. In *International Symposium on Mathematical Foundations of Computer Science*, pp. 37–49. Springer, 1992.

Minh, H. Q., Niyogi, P., and Yao, Y. Mercer's theorem, feature maps, and smoothing. In *International Conference on Computational Learning Theory*, pp. 154–168. Springer, 2006.

Mirzasoleiman, B., Jegelka, S., and Krause, A. Streaming non-monotone submodular maximization: Personalized video summarization on the fly. *arXiv preprint arXiv:1706.03583*, 2017.

Peres, Y. and Virág, B. Zeros of the iid gaussian power series: a conformally invariant determinantal process. *Acta Mathematica*, 194(1):1–35, 2005.

Petralia, F., Rao, V., and Dunson, D. B. Repulsive mixtures. In *Advances in Neural Information Processing Systems*, pp. 1889–1897, 2012.

Scardicchio, A., Zachary, C. E., and Torquato, S. Statistical properties of determinantal point processes in high-dimensional euclidean spaces. *Physical Review E*, 79(4): 041108, 2009.

Yao, J.-g., Fan, F., Zhao, W. X., Wan, X., Chang, E. Y., and Xiao, J. Tweet timeline generation with determinantal point processes. In *AAAI*, pp. 3080–3086, 2016.