How to Sample Uniform Spanning Trees, DPPs, and NDPPs

Guanghao Ye[*] Lichen Zhang[†]

Abstract

In this note, we review a list of results that generate samples from non-symmetric determinantal point process (NDPP). Our main focus in an Markov Chain Monte Carlo approach that results in a simple algorithm for NDPP sampling. As a consequence, we also show such MCMC process gives fast algorithms for sampling symmetric DPP and uniform spanning trees from graphs.

[*] `ghye@mit.edu`. MIT.

[†] `lichenz@mit.edu`. MIT.

Contents

1 Introduction

Given a matrix $L \in \mathbb{R}^{n\times n}$, one is interested in generating a sample from the determinantal point process (DPP) distribution defined by L. More precisely, given a set $S \subseteq [n]$, the DPP distribution says the probability S is sampled is proportional $\det(L_S)$, where L_S is the submatrix of L indexed by L. Of course, not all square L can define a DPP, a necessary condition is that all principle minors of L are non-negative. In this note, we consider algorithms that generate samples from DPP. Our focus is on an Markov Chain Monte Carlo (MCMC) approach that has fast mixing time and implies a conceptually simple algorithm. We start by showing a seemingly unrelated combinatorial problem, namely, sample a spanning tree uniformly from a graph is closely related to DPP where L is PSD. Then, we extend our discussion for more general matrices in which L is might be non-symmetric and its corresponding DPP called non-symmetric determinantal process (NDPP). We study several different algorithms for sampling from NDPP.

The main focus of this note is an in-depth look into the MCMC process for sampling NDPP. We demonstrate its connection with a family of polynomials and their geometric structure. Finally, we briefly discuss the maximum a posteriori (MAP) inference for NDPP.

2 Sample a spanning tree

In this section, we show how to sample a spanning tree uniform at random and how it's connected to the sampling of DPP.

Algorithm 1 MCMC algorithm for sample spanning tree

procedure SAMPLINGTREESAMPLE($G = (V, E)$)
    Let T_0 be an arbitrary spanning tree of G
    for $i = 1, \ldots,$ **do**
        Remove an edge e from T_{i-1} uniformly at random
        Let $A \cup B$ be the two connected components such that $A \cup B = T_{i-1} \setminus \{e\}$
        Pick e' from $\mathrm{Cut}(A, B)$ uniformly at random
        $T_i \leftarrow A \cup B \cup \{e'\}$
    end for
end procedure

This is a random walk on a huge graph $G_{\mathcal{T}}$, where each node in $G_{\mathcal{T}}$ corresponding to a spanning tree in G, and two nodes in $G_{\mathcal{T}}$ are adjacent if and only they differ by exactly a single edges.

It's easy to see the stationary distribution of the Markov chain above is the uniform distribution among all spanning trees. The only problem that remains is the mixing time of this Markov chain.

We first show random spanning tree distribution is a special case of k-DPPs.

Lemma 2.1. *Given graph $G = (V, E)$, its uniform spanning tree distribution is a DPP restricted to sets of cardinality $n-1$. Moreover, its L-ensemble is given $L = BB^\top$, where $B \in \mathbb{R}^{|E|\times|V|}$ is the incidence matrix of G.*

Proof. It's suffices to show that for any $|S| = n - 1$, we have

$$\det(L_S) = \begin{cases} n & \text{if } S \text{ is a sampling tree,} \\ 0 & \text{otherwise.} \end{cases}$$

We note that if S contains a cycle C, then B_C is linear dependent, hence $\det(L_S) = 0$.

For the case S is a sampling tree, we note that

$$\det(L_S) = \sum_{F \in \binom{n}{n-1}} \det(B_{S,F})^2 = n,$$

where the first equation follows by Cauchy-Binet, and the second one follows by $\det(B_{S,F})$ is either -1 or 1. $\square$

This relationship between k-DPPs and uniform spanning tree hints us the algorithm for sampling k-DPPs.

Algorithm 2 MCMC-based approximate sampling for k-DPPs

```
procedure k-DPP-SAMPLE(L ∈ ℝ^{n×n}, k ∈ ℕ, t_iter ∈ ℕ)
    Select S_0 uniformly from ([n] choose k)
    for t = 1 → t_iter do
        A_t ← Size k − 1 subset of S_{t−1} uniformly at random
        Select a ∈ [n] with probability ∝ det(L_{A_t ∪ {a}})
        S_t ← A_t ∪ {a}
    end for
    return S_{t_iter}
end procedure
```

Theorem 2.2. *The random walks above for k-DPPs has mixing time $O(k \log(k/\epsilon))$.*

2.1 Real stable polynomial

To carry over the analysis, let us form one more layer of abstractions. Let $\mu : \binom{[n]}{k} \to \mathbb{R}_{\geq 0}$ denote a density function over all subsets of $[n]$ with size k. For k-NDPP, $\mu(S) = \det(L_S)$. We define the generating polynomials as follows.

Definition 2.3 (Generating polynomials)**.** *Given a density $\mu : \binom{[n]}{k} \to \mathbb{R}_{\geq 0}$, we associate it with a multivariate polynomial g_μ as*

$$g_\mu(z_1, \ldots, z_n) = \sum_S \mu(S) \prod_{i \in S} z_i.$$

Definition 2.4 (Real stable polynomial)**.** *A multivariate polynomial $p(z_1, \ldots, z_n)$ is real stable if all of the coefficients are real, and $p(z_1, \ldots, z_n) \neq 0$ whenever $\mathrm{Im}(z_i) > 0$ for all i,*

Given these two definitions, we are ready to state the main theorem:

Theorem 2.5 (Theorem 1 of [ALG+21])**.** *For any distribution $\mu : \binom{[n]}{k} \to \mathbb{R}_{\geq 0}$ with real stable generating polynomials, the mixing time of random walk above is $O(k \log(k/\epsilon))$.*

Now, we will show that the generating polynomial for any DPP is real stable. First, we need the following lemmas:

Lemma 2.6. *A multivariate polynomial $p(z_1, \ldots, z_n)$ is real stable if for every point $a \in \mathbb{R}^n_{>0}$ and $b \in \mathbb{R}^n$, the univarite polynomial $p(at + b)$ is not identically zero and is real rooted.*

Proof. We prove by contradiction. Suppose p is not real stable. Then, there exists $(z_1, \ldots, z_n)$ such that $\mathrm{Im}(z_i) > 0$ for all i that is a root of p. Let $a_i = \mathrm{Im}(z_i)$ and $b_i = \mathrm{Re}(z_i)$. Then, we note $p(at+b)$ is not identically zero and it must be real rooted. However, $t = \mathbf{i}$ is a root of $p(at+b)$. $\square$

Theorem 2.7. *The generating polynomial of any DPP is real stable.*

Proof. For the sake of simplicity, we show it using K-ensemble. It's easy to check that

$$g_\mu(z_1, \ldots, z_n) = \det(I - K + K \cdot \mathrm{diag}(z)).$$

Then, we note that

$$\begin{aligned}\det(I - K + K \cdot \mathrm{diag}(z)) &= \det(K) \cdot \det(K^{-1} - I + \mathrm{diag}(z)) \\ &= \det(K) \cdot \det(L^{-1} + \mathrm{diag}(z))\end{aligned}$$

Since $0 \prec K \prec I$, it suffices to show $p(z) = \det(L^{-1} + \mathrm{diag}(z))$ is real stable.

Let $A = \mathrm{diag}(z))$ and $b = \mathrm{diag}(b)$, we have

$$\begin{aligned}p(at+b) &= \det(L^{-1} + At + B) \\ &= \det(A) \cdot \det(A^{-1/2} L^{-1} A^{-1/2} + \mathrm{diag}(b/a) + It).\end{aligned}$$

Then, we complete the proof by noting $\det(A^{-1/2} L^{-1} A^{-1/2} + \mathrm{diag}(b/a) + It)$ is the charasteric polynomial of matrix $M = A^{-1/2} L^{-1} A^{-1/2} + \mathrm{diag}(b/a)$, which is symmetry. $\square$

3 Efficient algorithms for NDPP

Let $L \in \mathbb{R}^{n \times n}$ be a matrix that is possibly non-symmetric. [GHD+21] shows that $L = VV^\top + B(D - D^\top)B^\top$, where $V, B \in \mathbb{R}^{n \times d}$ and $D \in \mathbb{R}^{d \times d}$. Here, $d \ll n$ should be viewed as a low rank decomposition of L into a symmetric part $VV^\top$ and skew-symmetric part $B(D - D^\top)B^\top$. Let $X = \begin{bmatrix} V & B \end{bmatrix} \in \mathbb{R}^{n \times 2d}$ and $W = \begin{bmatrix} I_d & 0 \\ 0 & D - D^\top \end{bmatrix} \in \mathbb{R}^{2d \times 2d}$, we can write L compactly as

$$L = XWX^\top.$$

We can write the K matrix as

$$\begin{aligned}K &= L(I+L)^{-1} \\ &= ((L+I) - I)(I+L)^{-1} \\ &= I - (I-L)^{-1} \\ &= I - (I - XWX^\top)^{-1} \\ &= I - I + XW(I + X^\top XW)^{-1} X^\top \\ &= X \underbrace{W(I + X^\top XW)^{-1}}_{\widetilde{W}} X^\top \\ &= X\widetilde{W}X^\top,\end{aligned}$$

where the second-to-last step is by matrix Woodbury identity. Thus, we can compactly write K as

$$K = X\widetilde{W}X^\top.$$

Let us introduce some tools.

Definition 3.1 (Youla decomposition). *Let $S \in \mathbb{R}^{2d\times 2d}$ be a skew-symmetric matrix and $A = BSB^\top \in \mathbb{R}^{n\times n}$ for $B \in \mathbb{R}^{n\times 2d}$. Then*

- *Eigenvalues of A are purely imaginary, in the form of $\mathbf{i}\lambda_1, -\mathbf{i}\lambda_1, \dots, \mathbf{i}\lambda_d, -\mathbf{i}\lambda_d$,*
- *Eigenvectors of A are in the form of $u_1 + \mathbf{i}v_1, u_1 - \mathbf{i}v_1, \dots, u_d + \mathbf{i}v_d, u_d - \mathbf{i}v_d$ where all u_i, v_i's are orthogonal to each other, and of dimension n.*
- *The Youla decomposition of A is*

$$A = \sum_{i=1}^{d} \begin{bmatrix} a_i - b_i & a_i + b_i \end{bmatrix} \begin{bmatrix} 0 & \lambda_i \\ -\lambda_i & 0 \end{bmatrix} \begin{bmatrix} a_i^\top - b_i^\top \\ a_i^\top + b_i^\top \end{bmatrix}$$

To simplify the notation, we will use u_i to denote $a_i - b_i$ and v_i for $a_i + b_i$.

There are two major ways for fast NDPP sampling, with one being Cholesky decomposition and the other being the up-down MCMC.

3.1 Cholesky decomposition approach [HGG+22]

The first algorithm is to leverage Cholesky decomposition. The idea is to iteratively construct the set based on conditional probability. Then, the K matrix will be updated appropriately based on the conditional probability. We give a sketch of [HGG+22] algorithm as follows.

Algorithm 3 Cholesky-based exact sampling

procedure CHOLESKYSAMPLE($L = XWX^\top \in \mathbb{R}^{n\times n}$)
  $\widetilde{W} \leftarrow W(I + X^\top XW)^{-1}$ ▷ $\mathcal{T}_{\mathrm{mat}}(d, n, d)$
  ▷ $K = X\widetilde{W}X^\top$, but we won't explicitly form K
  $Y \leftarrow \emptyset$
  for $i = 1 \to n$ **do**
    $p_i \leftarrow x_i^\top \widetilde{W} x_i$
    $u \leftarrow \mathrm{Unif}(0, 1)$
    if $u \le p_i$ **then**
      $Y \leftarrow Y \cup \{i\}$
    else
      $p_i \leftarrow p_i - 1$
    end if
    $\widetilde{W} \leftarrow \widetilde{W} - \frac{\widetilde{W} x_i x_i^\top \widetilde{W}}{p_i}$ ▷ $O(d^2)$
  end for
  return Y
end procedure

Note that to generate one sample from NDPP, this algorithm runs in time $O(nd^2)$.

3.2 Rejection sampling approach [HGG+22]

To further improve the running time, a rejection sampling approach is proposed. The idea is to construct a proposal distribution that is a DPP, then utilize efficient DPP sampling algorithm. Finally, correct the sampling probability via rejection sampling.

Recall that by Youla decomposition, we can write $B(D - D^\top)B^\top$ as

$$B(D - D^\top)B^\top = \sum_{i=1}^{d} \lambda_i (v_i u_i^\top - u_i v_i^\top)$$

Let $\widetilde{X} = \begin{bmatrix} V & v_1 & u_1 & \dots & v_d & u_d \end{bmatrix}$ and $\widetilde{W}_{\text{NDPP}} = \text{Diag}(I_d, \begin{bmatrix} 0 & \lambda_1 \\ -\lambda_1 & 0 \end{bmatrix}, \dots, \begin{bmatrix} 0 & \lambda_d \\ -\lambda_d & 0 \end{bmatrix})$ then $L = \widetilde{X}\widetilde{W}_{\text{NDPP}}\widetilde{X}^\top$. To design the desired L matrix for DPP, we set $\widetilde{W}_{\text{DPP}} = \text{Diag}(I_d, \lambda_1, \lambda_1, \dots, \lambda_d, \lambda_d)$ and consider $\widetilde{L} = \widetilde{X}\widetilde{W}_{\text{DPP}}\widetilde{X}^\top$. Theorem 1 of [HGG+22] shows that for any subset $Y \subseteq [n]$, $\det(L_Y) \leq \det(\widetilde{L}_Y)$ and for size-d subset Y, $\det(L_Y) = \det(\widetilde{L}_Y)$.

Before doing so, we introduce a tree-based sampling algorithm for DPP.

Lemma 3.2 (Proposition 1 of [HGG+22])**.** *There exists a data structure for sampling DPP with the following guarantee:*

- Preprocess*: the data structure takes in the set of eigenvectors and eigenvalues of L and preprocesses them in $O(nd^2)$ time.*
- SampleDPP*: the data structure outputs a sample obeying the DPP distribution defined by L, in $O(d + k^3 \log n + k^4)$ time, where k is the size of the output set.*

Algorithm 4 Reject-based exact sampling

---

procedure Preprocess($V \in \mathbb{R}^{n\times d}, B \in \mathbb{R}^{n\times d}, D \in \mathbb{R}^{d\times d}$)
        $\triangleright\ L = VV^\top + B(D - D^\top)B^\top$
 $\{(\lambda_i, v_i, u_i)\}_{i=1}^{d} \leftarrow$ YoulaDecompose(B, D)
 $\widetilde{W}_{\text{DPP}} \leftarrow \text{Diag}(I_d, \lambda_1, \lambda_1, \dots, \lambda_d, \lambda_d)$
 $\widetilde{X} \leftarrow \begin{bmatrix} V & v_1 & u_1 & \dots & v_d & u_d \end{bmatrix}$
 $\{(\sigma_i, z_i)\}_{i=1}^{2d} \leftarrow$ EigenDecompose$(\widetilde{X}\widetilde{W}_{\text{DPP}}^{1/2})$
 DPP $\mathcal{T}$
 $\mathcal{T} \leftarrow$ Preprocess$(\{(\sigma_i, z_i)\}_{i=1}^{2d})$
 return $\mathcal{T}$
end procedure

procedure RejectSample($V \in \mathbb{R}^{n\times d}, B \in \mathbb{R}^{n\times d}, D \in \mathbb{R}^{d\times d}, \mathcal{T}, \{(\sigma_i, z_i)_{i=1}^{2d}\}$)
 while true **do**
  $Y \leftarrow$ SampleDPP$(\mathcal{T})$
  $\widetilde{L}_Y \leftarrow V_{Y,*}V_{Y,*}^\top + B_{Y,*}(D - D^\top)B_{Y,*}^\top$
  $L_Y \leftarrow Z_{Y,*}\Sigma Z_{Y,*}^\top$
  $p \leftarrow \frac{\det(L_Y)}{\det(\widetilde{L}_Y)}$
  $u \leftarrow \text{Unif}(0, 1)$
  if $u \leq p$ **then**
   break
  end if
 end while
 return Y
end procedure

For rejection sampling, it is important to bound the expected number of rejections. [HGG+22] shows that if $V \perp B$, then the expected number of rejections $(1+\alpha)^d$ where $\alpha = \frac{1}{d}\sum_{i=1}^{d} \frac{2\lambda_j}{\lambda_j^2+1}$. The preprocessing time is $O(nd^2)$ and the per-sample (expected) time is

$$O((1+\alpha)^d \cdot (d + k^3 \log n + k^4)).$$

Note that for dimensional-d kernel L, we will always have $k = d$.

3.3 MCMC approach [HGDK22]

Note that prior algorithm has exponential dependence on d, which is less satisfactory. To address this issue, [HGDK22] develops a fast MCMC sampling approach based on that of [AASV21]. The algorithm starts with a uniform random size k set S_0, and repeatedly the algorithm selects a uniform size $k-2$ set A_t, then consider all possible size k supersets of A_t, and update S_{t+1} as the two new elements of the superset. To put it into code, we have

Algorithm 5 Down-up MCMC-based approximate sampling

```
procedure MCMCSample(L ∈ R^{n×n}, k ∈ N, t_iter ∈ N)
    Select S_0 uniformly from ([n] choose k)
    for t = 1 → t_iter do
        A_t ← Size k − 2 subset of S_{t−1} uniformly at random
        Select a, b ∈ [n] with probability ∝ det(L_{A_t ∪ {a,b}})
        S_t ← A_t ∪ {a, b}
    end for
    return S_{t_iter}
end procedure
```

[AASV21] shows that the Markov chain mixes well (ϵ-close to the NDPP distribution in TV distance) in roughly k^2 steps:

Lemma 3.3 (Theorem 11 of [AASV21]). *For any $\epsilon > 0$, a sample S obtained by Algorithm 5 with*

$$t_{\text{iter}} = O(k^2 \cdot \log(\frac{1}{\epsilon \cdot \Pr[S_0]}))$$

and randomly chosen subset $S_0 \in \binom{[n]}{k}$, the TV distance to the target k-NDPP distribution is at most ϵ.

Note that Algorithm 5 is far from efficient: a naive implementation will require compute $k \times k$ determinant for $O(n^2)$ possible a, b, so per iteration might cost $O(n^2 k^\omega)$ time. We note that Algorithm 5 does not exploit the low rank structure of L.

For sampling $\propto \det(L_{A\cup\{a,b\}})$, one can show that $\det(L_{A\cup\{a,b\}}) \propto \det(L^A_{\{a,b\}})$ where L^A is the conditional NDPP kernel on A defined as

$$\begin{aligned} L^A &= L - L_{*,A} L_A^{-1} L_{A,*} \\ &= X W^A X^\top \end{aligned}$$

for

$$W^A = W - W X_{A,*}^\top (X_{A,*} W X_{A,*}^\top)^{-1} X_{A,*} W. \tag{1}$$

$X_{A,*}$ is an $(k-2)\times d$ matrix, therefore, the time to form W^A is $\mathcal{T}_{\mathrm{mat}}(d,k,d)+k^\omega+d^\omega$, independent of n.

Then, instead of performing Youla decomposition on $XW^AX^\top$, they do so on a skew-symmetric matrix depends solely on W^A, which means all running time are $\mathrm{poly}(d)$.

Algorithm 6 Size 2 reject sampling

procedure $\textsc{Size2RejectSample}(X \in \mathbb{R}^{n\times 2d}, W\in\mathbb{R}^{2d\times 2d}, A\subseteq[n])$
  $W^A \leftarrow W - WX_{A,*}^\top(X_{A,*}WX_{A,*}^\top)^{-1}X_{A,*}W$
  $\{(\lambda_i,v_i,u_i)\}_{i=1}^d \leftarrow \textsc{YoulaDecompose}(\frac{W^A-(W^A)^\top}{2})$
  $\widehat{W}^A \leftarrow \frac{W^A+(W^A)^\top}{2}+\sum_{i=1}^d\lambda_i(v_iv_i^\top+u_iu_i^\top)$
  while true **do**
    $\{a,b\}\leftarrow \textsc{SampleDPP}(X\widehat{W}^AX^\top)$
    $u\leftarrow \mathrm{Unif}(0,1)$
    $L^A_{\{a,b\}}\leftarrow X_{\{a,b\},*}W^AX^\top_{\{a,b\},*}$
    $\widehat{L}^A_{\{a,b\}}\leftarrow X_{\{a,b\},*}\widehat{W}^AX^\top_{\{a,b\},*}$
    $p\leftarrow \frac{\det(L^A_{\{a,b\}})}{\det(\widehat{L}^A_{\{a,b\}})}$
    if $u\le p$ **then**
      break
    end if
  end while
  return $\{a,b\}$
end procedure

For sampling the size-2 DPP, we can use the tree-based DPP sampler. It remains to argue the expected number of rejections. We define the natural condition number as follows:

Definition 3.4. *Given $X\in\mathbb{R}^{n\times d}, W\in\mathbb{R}^{d\times d}$ with $W+W^\top\succeq 0$ and $A\in\binom{[n]}{k-2}$ for $k\ge 2$. Let W^A as in Eq. (1), we define κ_A as*

$$\kappa_A := \frac{\lambda_{\max}(W^A-(W^A)^\top)}{\min_{Y\binom{[n]\setminus A}{2}}\lambda_{\min}([X(W^A+(W^A)^\top)X^\top]_Y)}$$

and

$$\kappa := \max_{A\subseteq[n],|A|\le d-2}\kappa_A.$$

Then, the expected number of rejections is at most $(1+\sigma^2_{\max}(X)\kappa)^2$.

4 Sample from NDPP via MCMC

In this section, we review, in details, how [AASV21] generates samples from NDPP using Markov Chain Monte Carlo (MCMC) approach. The algorithm is strikingly simple as shown in Algorithm 5. It starts with uniformly random set of size k, then performs the following random walk: examine all 2-neighbors of the current set S, meaning that all subsets $T\subseteq[n]$ with $|T|=k$ and $|S\cap T|=k-2$, then update S to T with probability proportional to $\det(L_T)$. The mixing of this random walk is rather fast: it generates a sample whose TV distance to the target k-NDPP distribution is at most ϵ in $O(k^2\cdot\log(\frac{1}{\epsilon\cdot\Pr[S_0]}))$ steps.

4.1 Sector stable polynomials

The generating polynomial has nonnegative coefficients, therefore, there is no root $(z_1, \ldots, z_n) \in \mathbb{R}^n_{>0}$, i.e., strictly positive root. This motivates the definition of sector stability, meaning that g_μ not only does not have positive root, but root in a sector of the complex plane centered around $\mathbb{R}_{>0}$.

Definition 4.1 (Sector stability). *For an open sector $\Gamma \subseteq \mathbb{C}$ centered around the real positive axis in the complex plane, we say a polynomial $g(z_1, \ldots, z_n)$ is sector stable if*

$$z_1, \ldots, z_n \in \Gamma \Rightarrow g(z_1, \ldots, z_n) \neq 0.$$

The key result proved in [AASV21] is that if g_μ is sector stable with aperture $\Omega(1)$, then the down-up random walk $k \leftrightarrow \ell$ with $\ell = k - O(1)$ has relaxation time $k^{O(1)}$, meaning that the mixing is rapid. For the exposition of this note, we recall two results from [AASV21].

Theorem 4.2 (Theorem 4 of [AASV21]). *Suppose the density μ has a generating polynomial g_μ that is sector stable w.r.t a section Γ of aperture $\Omega(1)$. Then for an appropriate value $\ell = k - O(1)$, the $k \leftrightarrow \ell$ has relaxation time $k^{O(1)}$.*

For time-reversible Markov chains with positive eigenvalues, the relaxation time is the inverse spectral gap, which implies a mixing time bound.

Corollary 4.3 (Corollary 5 of [AASV21]). *Suppose μ has a generating polynomial that is sector stable for a sector of constant aperture and let $\ell = k - O(1)$ be the value given by Theorem 4.2. If $k \leftrightarrow \ell$ down-up random walk starts from set S_0, then it takes*

$$t_{\text{iter}} = O(k^2 \cdot \log(\frac{1}{\epsilon \cdot \Pr[S_0]}))$$

to generate a sample whose TV distance is at most ϵ from the target Markov chain.

4.2 Spectral independence

How to prove that mixing of the Markov chain of such down-up random walks? The idea is to interpret the target distribution μ as a weighted hypergraph. It is then natural to establish a notion of high-dimensional expander which has fast mixing rate of natural random walks. Thus, the problem of designing fast mixing Markov chains becomes the sufficient conditions for the hypergraphs defined by μ with good expansions. More concretely, we need to define the correlation matrix:

Definition 4.4 (Correlation matrix). *For a distribution μ over subsets S of the ground set $[n]$, define the correlation matrix $\Phi \in \mathbb{R}^{n \times n}$ such that*

$$\Phi_{i,j} = \Pr_{S \sim \mu}[j \in S \mid i \in S] - \Pr_{S \sim \mu}[j \in S].$$

If i and j are independent then $\Phi_{i,j} = 0$. On the other hand, if i and j are dependent then $\Phi_{i,j}$ has large value. The key of spectral independence framework is to show that the spectral norm of Φ is bounded by $O(1)$. To prove the constant upper bound on the spectral norm, we need one more ingredient on the conditioned distribution.

Definition 4.5 (Conditioned distribution). *For a distribution μ over subsets S of the ground set $[n]$ and $T \subseteq [n]$, define μ_T to be the distribution of $S \sim \mu$ conditioning on $T \subseteq S$.*

We review a list of techniques on bounding $\|\Phi\|$.

Trickle down. One natural idea is to start with some simple distributions, or in the case of NDPP, start with sets that contains certain singletons. More specifically, if one can bound $\|\Phi\|$ for $\mu_{\{1\}}, \mu_{\{2\}}, \ldots, \mu_{\{n\}}$, then this implies a bound on $\|\Phi\|$ for μ under some mild conditions. This conditional approach resembles what people typically do for NDPP, i.e., starting from a set (say of size $k-2$) and inductively bound $\|\Phi\|$. Unfortunately, this framework only works for matriod and matriod-related ones, as the induction is rather brittle.

Negative correlation. Negative correlation is a property for distributions such that the correlation matrix has negative entries except on the diagonal. Distributions such as the uniform distribution on spanning trees, DPP and balanced matriods all have such property. When negative correlation exists, $\|\Phi\|$ can be bounded by the ℓ_1 norm of the rows of Φ as $O(1)$.

Correlation decay. For the special case that μ is defined on an underlying graph, suppose there exists a measurement $\sigma : V \to \mathbb{R}$, then informally, correlation decay says that if u and v are "far away" vertices, then $\sigma(u)$ and $\sigma(v)$ are almost independent of each other. This is one of the most important tools for bounding the max eigenvalue of Φ, as it requires no assumptions such as bounded degree of the graph.

Unfortunately, all these techniques are not enough for bounding $\|\Phi\|$ for our applications. Instead, we turn our attention to the roots of the partition function in the complex plane. Before moving on, we record a result from [AASV21], which states that sector stable polynomials have small spectral norm on correlation matrix.

Theorem 4.6 (Theorem 17 of [AASV21])**.** *Let $\mu : \binom{[n]}{k} \to \mathbb{R}_{\geq 0}$ be a density whose generating polynomial is sector stable with respect to a sector of aperture $\Omega(1)$. Then the ℓ_1 norm of any row in the correlation matrix Φ is bounded by $O(1)$, i.e.,*

$$\forall i : \sum_j |\Pr_{S\sim\mu}[j \in S \mid i \in S] - \Pr_{S\sim\mu}[j \in S]| \leq O(1).$$

4.3 Fractional log-concavity

Given a polynomial $g_\mu(z_1, \ldots, z_n)$ viewed as a function from $\mathbb{R}^n_{\geq 0} \to \mathbb{R}$, we say it's log-concave if $\log g_\mu$ is concave over its input from $\mathbb{R}^n_{\geq 0}$. Log-concave polynomials is a proper superset of the real-stable polynomials, another family of polynomials that lead to breakthrough in resolving the Kadison-Singer conjecture [MSS15]. For sector stable polynomials, the proper generalization is the fractional log-concave polynomials.

Definition 4.7 (Fractional log-concave polynomials)**.** *We call the polynomial $g_\mu(z_1, \ldots, z_n)$ with parameter $\alpha \in [0, 1]$ if $\log g_\mu(z_1^\alpha, \ldots, z_n^\alpha)$ is concave as a function over $\mathbb{R}^n_{\geq 0}$.*

One might wonder, given sector stability, why we still want to investigate its generalization, the fractional log-concave polynomials. The main reason is, fractional log-concave is both necessary and sufficient for spectral independence.

Theorem 4.8 (Proposition 20 of [AASV21])**.** *Let $\mu : \binom{[n]}{k} \to \mathbb{R}_{\geq 0}$ be a density, then we have that $\|\Phi\| \leq O(1)$ if and only if the polynomial g_μ is fractional log-concave around the vector $\mathbf{1}_n$ for a parameter $\alpha > \Omega(1)$.*

This implies that any sector stable polynomial is also fractional log-concave around the vector $\mathbf{1}_n$. One advantage of this formulation is that we only require fractional log-concave around $\mathbf{1}_n$, a very local condition that is extremely easy to verify. This, however, does not imply a spectral norm bound on conditional density μ_T, to achieve so, one requires fractional log-concavity of g_μ at all points in $\mathbb{R}^n_{\geq 0}$. We briefly sketch out a proof for this fact.

First, we note that sector stability is preserved under the real scaling: $(z_1, \ldots, z_n) \to (\lambda_1 z_1, \ldots, \lambda_n z_n)$, where λ_i's are positive reals. This is simply because sectors in complex planes are preserved under real scaling. We can use this trick to map any point in $\mathbb{R}^n_{\geq 0}$ to $\mathbf{1}_n$.

Now, let us examine the generating polynomial g_{μ_T}. We claim that

$$g_{\mu_T} \propto \lim_{\lambda \to \infty} g_\mu(\underbrace{\lambda z_1, \lambda z_2,}_{\text{elements in } T} \ldots, z_n)/\lambda^{|T|}.$$

Intuitively, one can view it as penalizing all elements that are not in T. Recall that scaling the variables and polynomials and taking limits all preserve fractional log-concavity. Thus, we have the following corollary:

Corollary 4.9. *If $\mu : \binom{[n]}{k} \to \mathbb{R}_{\geq 0}$ has fractional log-concave generating polynomial with parameter $\alpha = \Omega(1)$, or a sector stable polynomial with a sector of aperture $\Omega(1)$, then for all conditioned distributions μ_T, the correlation matrix has maximum eigenvalue $O(1)$.*

4.4 Generating polynomial of NDPP

With all machinery in hand, we only need to show that the generating polynomial of NDPP is fractional log-concave or sector stable with a sector of aperture $\Omega(1)$. The sector we will be working with is the following:

Definition 4.10. *Let Γ_α denote the open sector of aperture $\alpha\pi$ as follows:*

$$\Gamma_\alpha = \{\exp(x + y\mathbf{i}) : x \in \mathbb{R}, y \in (-\alpha\pi/2, \alpha\pi/2)\}.$$

We define a notion of half plane stability.

Definition 4.11. *Consider an open half-plane $H_\theta = \{e^{\theta \mathbf{i}} z : \operatorname{im}(z) > 0\} \subseteq \mathbb{C}$. A polynomial $g(z_1, \ldots, z_n) \in \mathbb{C}[z_1, \ldots, z_n]$ is H_θ stable if g does not have roots in H_θ^n.*

We say g is Hurwitz stable if it is $H_{\pi/2}$ stable. It is real stable if it is H_0 stable with real coefficients.

For NDPP, it is natural to associate it with the characteristic polynomial.

Theorem 4.12 (Theorem 38 of [AASV21]). *Let $A_1, \ldots, A_n$ be (complex) PSD matrices and B be (complex) Hermitian matrix, all matrices are of size $m \times m$.*

- *$f(z_1, \ldots, z_n) = \det(z_1 A_1 + \ldots + z_n A_n + B)$ is either real stable or zero.*
- *If B is also PSD then $f(z_1, \ldots, z_n)$ has non-negative coefficients.*

We prove that a specific polynomial associated with the NDPP process has non-negative coefficients, is either Hurwitz stable or zero.

Lemma 4.13 (Lemma 39 of [AASV21]). *Consider $L \in \mathbb{R}^{n \times n}$ such that $L + L^\top \succeq 0$. Let $f(z_1, \ldots, z_n) = \sum_{S \subseteq [n]} \det(L_{S,S}) z^{[n] \setminus S}$. Then f has non-negative coefficients, is either Hurwitz stable or zero.*

Proof. We first prove the coefficients are non-negative. This is because L is a P_0 matrix, therefore any principle minor is non-negative.

Let $A = \frac{L+L^\top}{2}$ and $B = \frac{L-L^\top}{2}$. Clearly, A is PSD, we argue that the matrix $\mathbf{i}B$ is Hermitian. Note that B is skew-symmetric and

$$\begin{aligned}(\mathbf{i}B)^* &= -\mathbf{i} \cdot (-B^\top) \\ &= \mathbf{i}B^\top.\end{aligned}$$

Set $g(z_1, \dots, z_n, z_{n+1}) = \det(z_1\mathrm{Diag}(e_1) + \dots + z_n\mathrm{Diag}(e_n) + z_{n+1}A + \mathbf{i}B)$, apply Theorem 4.12, it is either zero or real stable.

Let $w_j = \mathbf{i}^{-1}z_j$, $Z = \sum_{i=1}^n z_i\mathrm{Diag}(e_i) = \mathrm{Diag}(\begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix})$ and $W = \mathrm{Diag}(\begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix})$, consider

$$\begin{aligned}g(z_1, \dots, z_n, \mathbf{i}) &= \det(Z + \mathbf{i}(A+B)) \\ &= \det(\mathbf{i}W + \mathbf{i}L) \\ &= \mathbf{i}^n \det(W + L) \\ &= \mathbf{i}^n \sum_{S \subseteq [n]} w^{[n]\setminus S} \det(L_{S,S}) \\ &= \mathbf{i}^n f(w_1, \dots, w_n).\end{aligned}$$

If g is zero, so is f. Otherwise, fix any $w_1, \dots, w_n$ in the right half plane $H_{\pi/2}$, we claim that $z_j = \mathbf{i}w_j$ are in H_0, this can be seen via Euler's formula. Finally, g is real stable, which means that f is Hurwitz stable. □

Remark 4.14. *The right half plane $H_{\pi/2}$ is equivalent to open sector Γ_1.*

Note that the polynomial f we prove here is not the generating polynomial of NDPP. We will show that sector stable is preserved via duality.

Lemma 4.15 (Proposition 54 of [AASV21])**.** *Given an α-sector stable polynomial g, the duality mapping $g \mapsto g^*$ preserves α-sector stable, where $g(z_1, \dots, z_n) = \sum_{S\subseteq[n]} c_S z^S$ and $g^*(z_1, \dots, z_n) = \sum_{S\subseteq[n]} c_S z^{[n]\setminus S}$.*

Proof. We can write $g^*(z_1, \dots, z_n)$ as

$$\begin{aligned}g^*(z_1, \dots, z_n) &= \sum_{S\subseteq[n]} c_S z^{[n]\setminus S} \\ &= z_1 \dots z_n g(z_1^{-1}, \dots, z_n^{-1}) \\ &\neq 0\end{aligned}$$

for all $z_1, \dots, z_n \in \Gamma_\alpha$. This is because g is Γ_α and $z_1^{-1}, \dots, z_n^{-1}$ are also in Γ_α. □

We need the following lemma.

Lemma 4.16 (Corollary 62 of [AASV21])**.** *Suppose $g(z_1, \dots, z_n) \in \mathbb{R}[z_1, \dots, z_n]$ is Γ_1-stable, then g_k is either zero or $\Gamma_{1/2}$-stable.*

Together with the duality transform and Hurwitz-stability of f, we have $\Gamma_{1/2}$-stable for the generating polynomial of k-NDPP.

Corollary 4.17 (Corollary 63 of [AASV21]). *Let $L \in \mathbb{R}^{n\times n}$ with $L + L^\top \succeq 0$, then*

$$f_k(z_1, \dots, z_n) = \sum_{S \in \binom{[n]}{k}} \det(L_{S,S}) z^{[n]\setminus S}$$

and its dual

$$f_k^*(z_1, \dots, z_n) = \sum_{S \in \binom{[n]}{k}} \det(L_{S,S}) z^{S}$$

are either zero or $\Gamma_{1/2}$-stable with non-negative coefficients.

With $\Gamma_{1/2}$-sector stability, we have already obtained a bound on the spectral norm of Φ. In the following discussions, we connect sector stability to fractional log-concavity.

We introduce a result on the max ℓ_1 row norm of Φ when the polynomial is Γ_α-stable.

Theorem 4.18 (Theorem 51 of [AASV21]). *Let $f \in \mathbb{R}_{\geq 0}[z_1, \dots, z_n]$ be a multi-affine polynomial that is Γ_α-stable for $\alpha \leq 1$. Let $\mu : 2^{[n]} \to \mathbb{R}_{\geq 0}$ be the distribution generated by f, then*

$$\sum_j |\Phi_{i,j}| \leq 2/\alpha.$$

We are in the position of connecting sector stability and fractional log-concavity.

Lemma 4.19 (Lemma 69 of [AASV21]). *For $\alpha \in [0, 1/2]$, if polynomial $f \in \mathbb{R}_{\geq 0}[z_1, \dots, z_n]$ is $\Gamma_{2\alpha}$-sector stable, then f is α-fractional log-concave.*

Proof. We shall prove via the negative semi-definiteness of the Hessian of of $\log f$. Let μ be the associated density to f. First, we show that it is enough to consider the fractional log-concavity at $\mathbf{1}_n$. For any $v \in \mathbb{R}^n_{>0}$, set $f^v(z_i) = f(\{v^\alpha z_i\})$, note f^v is sector stable as f is over non-negative reals.

Computing the Hessian, we have

$$\nabla^2 \log f(\{z_i^\alpha\})|_v = D_v(\nabla^2 \log f^v(\{z_i^\alpha\})|_{\mathbf{1}_n}) D_v,$$

where $D_v = \mathrm{Diag}(\begin{bmatrix} v_1^{-1} \\ \vdots \\ v_n^{-1} \end{bmatrix})$. Thus, it suffices to consider f at $\mathbf{1}_n$. Let $H = \nabla^2 \log f(\{z_i^\alpha\})|_{\mathbf{1}_n}$, then

$$H_{i,j} = \begin{cases} \alpha(\alpha-1)\Pr_{S\sim\mu}[i \in S] - \alpha^2(\Pr_{S\sim\mu}[i \in S])^2 & \text{if } i = j, \\ \alpha^2(\Pr_{S\sim\mu}[i,j \in S] - \Pr_{S\sim\mu}[i \in S]\Pr_{S\sim\mu}[j \in S]) & \text{otherwise.} \end{cases}$$

By Theorem 4.18, we know that $\|\Phi\| \leq 2/(2\alpha) = 1/\alpha$, therefore

$$\begin{aligned} \frac{1}{\alpha} I &\succeq \Phi \\ &= \frac{1}{\alpha^2} \cdot \mathrm{Diag}(\begin{bmatrix} \Pr[1 \in S] \\ \Pr[2 \in S] \\ \vdots \\ \Pr[n \in S] \end{bmatrix}) \cdot H + \frac{1}{\alpha} I, \end{aligned}$$

this means that $H \preceq 0$ and $\log f(\{z_i^\alpha\})$ is concave. □

A very interesting consequence of fractional log-concavity is that, it gives a way to estimate the support size of the distribution. For the case of NDPP, this gives an estimation on how many principle minors are nonzero.

Definition 4.20. *For any distribution μ over a finite set Ω, we define the entropy of μ as*

$$H(\mu) = \sum_{\omega \in \Omega} \mu(\omega) \log \frac{1}{\mu(\omega)}.$$

For any element $i \in [n]$, we let $\mu(i) := \Pr_{S \sim \mu}[i \in S]$ denote the marginal of element μ.

Fact 4.21. *For any finite discrete distribution μ, we have*

$$\sum_i H(\mu(i)) \geq H(\mu).$$

The above fact is called the subaditivity of entropy, one can prove it using conditional entropy and monotonicity.

The next lemma provides a lower bound on $H(\mu)$ for fractional log-concave generating polynomials.

Lemma 4.22 (Lemma 73 of [AASV21])**.** *For any α-fractional log-concave distribution $\mu : 2^{[n]} \to \mathbb{R}_{\geq 0}$, with marginal probabilities $\mu(1), \ldots, \mu(n)$, we have*

$$H(\mu) \geq \alpha \sum_i \mu(i) \log(\frac{1}{\mu(i)}).$$

Proof. Let g_μ be the generating polynomial, define polynomial f as

$$f(z_1, \ldots, z_n) = \log g_\mu(\frac{z_1^\alpha}{\mu_1^\alpha}, \ldots, \frac{z_n^\alpha}{\mu_n^\alpha}),$$

since μ is α-fractional log concave, we know that $\log g_\mu$ is concave, and scaling the argument preserves the concavity. Thus, f is concave.

Let X be a random variable that indicates a set S is chosen according to μ, i.e., $\Pr[X = \mathbf{1}_S] = \mu(S)$. Note that

$$\begin{aligned} f(\mathbf{1}_S) = & \log(\sum_{T \subseteq S} \mu(T) \prod_{i \in T} \frac{1}{\mu(i)^\alpha}) \\ \geq & \log(\mu(S) \prod_{i \in S} \frac{1}{\mu(i)^\alpha}) \\ = & \log(\mu(S)) + \alpha \sum_{i \in S} \log \frac{1}{\mu(i)}, \end{aligned}$$

where the second step is by the monotonicity of log. Thus,

$$\begin{aligned} \mathbb{E}[f(X)] = & \sum_S \mu(S) f(\mathbf{1}_S) \\ \geq & \sum_S (\mu(S) \log \mu(S) + \alpha \mu(S) \sum_{i \in S} \log \frac{1}{\mu(i)}) \end{aligned}$$

$$
\begin{aligned}
&= -H(\mu) + \alpha \sum_S \mu(S) \sum_{i\in S} \log \frac{1}{\mu(i)} \\
&\geq -H(\mu) + \alpha \sum_{i\in[n]} \mu(i) \log \frac{1}{\mu(i)},
\end{aligned}
$$

where the last step is by the second term is a sum of non-negatives.

Note that

$$
\begin{aligned}
f(\mathbb{E}[X]) &= f(\mu_1, \ldots, \mu_n) \\
&= \log g_\mu(\frac{\mu_1^\alpha}{\mu_1^\alpha}, \ldots, \frac{\mu_n^\alpha}{\mu_n^\alpha}) \\
&= \log g_\mu(\mathbf{1}_n) \\
&= 0,
\end{aligned}
$$

where the last step is by $g_\mu(\mathbf{1}_n) = \sum_S \mu(S) \cdot 1 = 1$. By Jensen's inequality, we have

$$
\begin{aligned}
\mathbb{E}[f(X)] &\leq f(\mathbb{E}[X]) \\
&= 0,
\end{aligned}
$$

therefore,

$$
H(\mu) \geq \alpha \sum_{i\in[n]} \mu(i) \log \frac{1}{\mu(i)}.
$$

□

Given μ, we define the dual density $\mu^* : 2^{[n]} \to \mathbb{R}_{\geq 0}$ as $\mu^*(S) = \mu([n] \setminus S)$.

Corollary 4.23. *If μ and its dual μ^* are α-fractionally log-concave then $\sum_i H(\mu(i))$ is a $\frac{\alpha}{2}$-approximation to $H(\mu)$.*

In particular, if μ is $\Gamma_{2\alpha}$-sector stable, then μ and its dual μ^* are α-fractionally log-concave. Therefore, $\sum_i H(\mu(i))$ is a $\frac{\alpha}{2}$-approximation to $H(\mu)$.

Proof. For any density μ, we have $H(\mu) \leq \sum_i H(\mu(i))$. The goal is to prove $H(\mu) \geq \frac{\alpha}{2} \sum_i H(\mu(i))$. By Lemma 4.22, we have

$$
\begin{aligned}
H(\mu) &\geq \alpha \sum_i \mu(i) \log \frac{1}{\mu(i)}, \\
H(\mu^*) &\geq \alpha \sum_i (1-\mu(i)) \log \frac{1}{1-\mu(i)}.
\end{aligned}
$$

Since μ and μ^* are dual, we have $H(\mu) = H(\mu^*)$ as the sum is just rearranged in reverse order. Therefore,

$$
\begin{aligned}
2H(\mu) &= H(\mu) + H(\mu^*) \\
&\geq \alpha \sum_i (\mu(i) \log \frac{1}{\mu(i)} + (1-\mu(i)) \log \frac{1}{1-\mu(i)}) \\
&= \alpha \sum_i H(\mu(i)).
\end{aligned}
$$

□

Let F_μ denote the support of distribution μ. Using entropy, we can estimate $\log|F_\mu|$. Given a set F, let $\mathrm{conv}(F)$ denote the convex hull of F.

Lemma 4.24. *Consider $F \subseteq \binom{[n]}{k}$. Suppose there exists an α-fractional log-concave polynomial g with $\mathrm{supp}(g) = F$. For any $(p_1, \ldots, p_n) \in \mathrm{conv}(F)$, there exists v with $\mathrm{supp}(v) \subseteq F$ such that $\sum_S v(S) z^S$ is α-fractional log-concave and $v(i) = p_i$ for all $i \in [n]$.*

As a consequence, $\max\{\sum_i p_i \log \frac{1}{p_i} : p \in \mathrm{conv}(F)\} = \max\{\sum_i \mu(i) \log \frac{1}{\mu(i)} : \mu \in V\}$, where V is the set of α-fractional log-concave polynomials μ such that $\mathrm{supp}(\mu) \subseteq F$.

We are now in the position to present an approach to estimate $\log|F_\mu|$.

Lemma 4.25 (Lemma 75 of [AASV21])**.** *Consider $F \subseteq \binom{[n]}{k}$. Let $F^* := \{[n] \setminus S : S \in F\}$. Let*

$$\beta := \max\{\sum_i p_i \log \frac{1}{p_i} : p \in \mathrm{conv}(F)\},$$
$$\beta^* := \max\{\sum_i q_i \log \frac{1}{q_i} : q \in \mathrm{conv}(F^*)\}.$$

Suppose there are two α-fractional log-concave polynomials with $\mathrm{supp}(g) = F$ and $\mathrm{supp}(h) = F^*$. Then, $\beta + \beta^*$ is $\alpha/2$-approximation to $\log|F|$, i.e., $(\beta + \beta^*) \geq \log|F| \geq \frac{\alpha}{2}(\beta + \beta^*)$.

Particularly, if there exists some g that is $\Gamma_{2\alpha}$-sector stable with $\mathrm{supp}(g) = F$, then $\beta + \beta^*$ is an $\alpha/2$-approximation to $\log|F|$.

Proof. Let v and v^* be uniform distribution over F and F^*. For a set F', let $V_{F'}$ be the set of all α-fractional log-concave polynomials with support being a subset of F'. Since V_F is non-empty as $\mathrm{supp}(g) = F$, by Lemma 4.24, we can re-characterize β and β^* as

$$\beta = \max_{\mu \in V_F}\{\sum_i \mu(i) \log \frac{1}{\mu(i)}\},$$
$$\beta^* = \max_{\mu \in V_{F^*}}\{\sum_i \mu(i) \log \frac{1}{\mu(i)}\}.$$

Let $S \in \{F, F^*\}$, we define a unified notation

$$\mu_S^{\arg\max} := \arg\max_{\mu \in V_S} \sum_i \mu(i) \log \frac{1}{\mu(i)}.$$

Since v is the uniform distribution over F, we have

$$\begin{aligned}
H(v) &= \sum_{i \in F} \frac{1}{|F|} \log|F| \\
&= \log|F| \\
&\leq \sum_i H(v(i)) \\
&= \sum_i (v_i \log \frac{1}{v(i)} + (1 - v_i) \log \frac{1}{1 - v(i)}) \\
&\leq \beta + \beta^*,
\end{aligned}$$

this is because $v \in \mathrm{conv}(F)$ and $\mathbf{1}_n - v \in \mathrm{conv}(F^*)$.

On the other hand, the uniform distribution over discrete set maximizes the entropy, therefore

$$\begin{aligned}\log|F| &= H(v)\\ &\geq H(\mu_F^{\arg\max})\\ &\geq \alpha\sum_i \mu_F^{\arg\max}(i)\log\frac{1}{\mu_F^{\arg\max}(i)}\\ &= \alpha\beta,\end{aligned}$$

where the third step is by Lemma 4.22. Similarly, we have $\log|F^*| \geq \alpha\beta^*$. Finally, note $|F| = |F^*|$. Thus, we have $\log|F| \geq \frac{\alpha}{2}(\beta+\beta^*)$. $\square$

Both β and β^* can be computed efficiently using a convex program. As a corollary, we obtain an efficient approach to estimate the support size of NDPP.

Corollary 4.26. *Let $L \in \mathbb{R}^{n\times n}$ such that $L + L^\top \succeq 0$. Let V^L denote the family of sets $S \subseteq [n]$ such that $\det(L_S) \neq 0$. For $k \leq n$, let V_k^L be the family of sets $S \in \binom{[n]}{k}$ with $\det(L_S) \neq 0$. Then, we can efficiently compute an $\frac{1}{8}$ multiplicative approximation to $\log|V^L|$ and $\log|V_k^L|$.*

5 MAP inference for NDPP

In this section, we study the MAP inference for NDPP. Given a matrix $L \in \mathbb{R}^{n\times n}$ such that $L + L^\top \succeq 0$ and $L = XWX^\top$ for $X \in \mathbb{R}^{n\times d}, W \in \mathbb{R}^{d\times d}$, the MAP inference asks us to find a subset $S \subseteq [n]$ such that $\det(L_S)$ is maximized. This problem is also known as volume maximization. We are particularly interested in the following paper [AV22], in which a local search algorithm is proposed. For k-NDPP, it is well-known that $k^{O(k)}$-approximation is optimal. Following is a template of the algorithm.

Algorithm 7 Determinant maximization for NDPP

procedure GREEDY($L \in \mathbb{R}^{n\times n}, k \in \mathbb{N}$)
  $S \leftarrow \emptyset$
  while $|S| < k$ **do**
    Pick $i \notin S$ that maximizes $\mu(S \cup \{i\})$, where $\mu(T) = \sum_{S\in\binom{[n]}{k}, S\supseteq T}\det(L_S)$
    $S \leftarrow S \cup \{i\}$
  end while
  return S
end procedure

procedure LOCALSEARCH($L \in \mathbb{R}^{n\times n}, k \in \mathbb{N}, c \in (0,1)$)
  $S \leftarrow$ GREEDY(L, k)
  while $\det(L_S) < c \cdot \det(L_T)$ for some $T \in N_2(S)$ **do**
    $S \leftarrow \arg\max\{\det(L_T) : T \in N_2(S)\}$
  end while
  return S
end procedure

We use $N_r(S)$ to denote the r-neighborhood set of S, defined as $N_r(S) = \{T \in \binom{[n]}{k} : |T \cap S| = k - r\}$.

To implement the greedy process, we collect some useful facts.

Fact 5.1. *For any $Y, D \subseteq [n]$, we have*

$$\begin{aligned}\det(L_{Y\cup D}) =&\ \det(L_Y)\det(L_D - L_{D,Y}L_Y^{-1}L_{Y,D}) \\ =&\ \det(L_Y)\det(L_D - X_D W(X_Y^\top L_Y^{-1} X_Y) W X_D^\top).\end{aligned}$$

Given $\det(L_Y)$ *and* L_Y^{-1}, $\det(L_{Y\cup D})$ *can be computed in* $O(\mathcal{T}_{\mathrm{mat}}(|D|, d, d) + \mathcal{T}_{\mathrm{mat}}(|Y|, |Y|, d) + |D|^\omega)$ *time.*

Given $L = XWX^\top$, [AV22] shows that, set $D_S = X_S^\top(X_S W X_S^\top)^{-1} X_S$, then $\mu(S)$ can be computed as the eigenvalues of $L^S := X_{\overline{S}}(W - WD_S W)X_{\overline{S}}^\top$. As S starts with the emptyset, so forming L^S is expensive. Alternatively, they form a "shift" matrix $F^S := (W - WD_S W)X_{\overline{S}}^\top X_{\overline{S}}$ which has the same characteristic polynomial as L^S and nonzero eigenvalues. (If time permits, add a proof. Idea is characteristic polynomial of special traces.)

Lemma 5.2. *The* Greedy *procedure runs in* $O(kn^2 d^{\omega-1})$ *time.*

Next, let us analyze the runtime of LocalSearch. It starts with a greedy initialization, then per iteration, we try all elements in $N_2(S)$, which is $\binom{k}{2} \cdot \binom{n-k}{2} = O(n^2k^2)$. To utilize the fact, we first fix a subset of size $k-2$, then iterate through all possible $O(n^2)$ candidates. For each candidate, the fact only takes $O(d^2 + dk^{\omega-1})$ time, so the overall running time is $O(n^2d^2k^2 + n^2dk^{\omega+1})$.

Lemma 5.3. *The* LocalSearch *procedure runs in* $O(n^2d^2k^2 + n^2dk^{\omega+1})$ *time.*

References

[AASV21] Yeganeh Alimohammadi, Nima Anari, Kirankumar Shiragur, and Thuy-Duong Vuong. Fractionally log-concave and sector-stable polynomials: Counting planar matchings and more. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, 2021.

[ALG+21] Nima Anari, Kuikui Liu, Shayan Oveis Gharan, Cynthia Vinzant, and Thuy-Duong Vuong. Log-concave polynomials IV: approximate exchange, tight mixing times, and near-optimal sampling of forests. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 408–420. ACM, 2021.

[AV22] Nima Anari and Thuy-Duong Vuong. From sampling to optimization on discrete domains with applications to determinant maximization. In *Proceedings of Thirty Fifth Conference on Learning Theory*, Proceedings of Machine Learning Research. PMLR, 2022.

[GHD+21] Mike Gartrell, Insu Han, Elvis Dohmatob, Jennifer Gillenwater, and Victor-Emmanuel Brunel. Scalable learning and map inference for nonsymmetric determinantal point processes. In *International Conference on Learning Representations (ICLR)*, 2021.

[HGDK22] Insu Han, Mike Gartrell, Elvis Dohmatob, and Amin Karbasi. Scalable mcmc sampling for nonsymmetric determinantal point processes. In *International Conference on Machine Learning (ICML)*, 2022.

[HGG+22] Insu Han, Mike Gartrell, Jennifer Gillenwater, Elvis Dohmatob, and Amin Karbasi. Scalable sampling for nonsymmetric determinantal point processes. In *International Conference on Learning Representations (ICLR)*, 2022.

[MSS15] Adam W Marcus, Daniel A Spielman, and Nikhil Srivastava. Interlacing families ii: Mixed characteristic polynomials and the Kadison—Singer problem. *Annals of Mathematics*, pages 327–350, 2015.