Thèse de Doctorat en Cotutelle

Pour obtenir le grade de Docteur délivré par

L'Université de Lille
L'École Doctorale Sciences pour l'Ingénieur
ET
L'Université Libanaise
L'École Doctorale Sciences et Technologie

Discipline
Mathématiques appliquées

présentée et soutenue publiquement par

Wehbe Diala

Le 3 Avril 2019

SIMULATIONS AND APPLICATIONS OF LARGE-SCALE k-DETERMINENTAL POINT PROCESSES

Membre du Jury

Président du jury:	Pr. Gérard Biau	Sorbonne Université, France
Directeur de thèse:	Pr. Nicolas Wicker	Université de Lille, France
Directrice de thèse:	Pr. Baydaa AL Ayoubi	Université Libanaise, Liban
Rapporteurs:	Pr. Anne Philippe	Université de Nantes, France
	Pr. Laurent Decreusefond	Telecom ParisTech, France
Examinateur:	Dr. Rami El Haddad	Université Saint-Joseph, Liban

Contents

To my parents
who made me what I am today.
To my brothers and sisters
who I couldn't imagine my life without them ♡

Acknowledgment

First and Foremost praise is to ALLAH, the Almighty, who provided me with blessings and grace, help and peace.

I would like to express my deepest gratitude to my supervisor Nicolas Wicker who started by proposing this PhD project on Determinantal Point Processes. Without his supervision, permanent support, help and scientific guidance, this thesis would not have been completed.

I would also like to thank my supervisor Baydaa Al Ayoubi at Lebanese university, for her patience, good advices, motivation and support during these years.

I extend my sincerest thanks to professor Anne Philippe at university of Nantes and professor Laurent Decreusefond at Telecom ParisTech, for agreeing to be the reviewers of my work and to participate in the defense of this thesis.

I am grateful to professor Gérard Biau at Sorbonne university for his commitment to take a part in my jury committee. I am also very glad for having doctor Rami EL-Haddad in my jury committee. I take this opportunity to thank him also for his help, support and encouragement during my master's degree in the university of Saint-Joseph in Lebanon.

Many thanks to professor Luc Moulinier at Strasbourg university, for his work in order to improve the biological application.

A particular thank must also be recorded to professor Rémi Bardenet at Lille university, for the time he devoted to me in a part of my work. Thank you for your kind collaboration.

I warmly thank and appreciate all my family and friends. You have all encouraged and believed in me. You have also played a very important role in trying to reduce the stress and anxiety. You have continually encouraged me to overcome the difficulties of life and continue my path till the end. Special thanks also to all of my laboratory partners for the good time that we shared.

Finally but profoundly, I pay my heartily thanks to my beloved father for the love and support he gave to me and for the strength of character and the sense of responsibility he taught me.

Résumé

Avec la croissance exponentielle de la quantité de données, l'échantillonnage est une méthode pertinente pour étudier les populations. Parfois, nous avons besoin d'échantillonner un grand nombre d'objets d'une part pour exclure la possibilité d'un manque d'informations clés et d'autre part pour générer des résultats plus précis. Le problème réside dans le fait que l'échantillonnage d'un trop grand nombre d'individus peut constituer une perte de temps.

Dans cette thèse, notre objectif est de chercher à établir des ponts entre la statistique et le k-processus ponctuel déterminantal(k-DPP) qui est défini via un noyau. Nous proposons trois projets complémentaires pour l'échantillonnage de grands ensembles de données en nous basant sur les k-DPPs. Le but est de sélectionner des ensembles variés qui couvrent un ensemble d'objets beaucoup plus grand en temps polynomial. Cela peut être réalisé en construisant différentes chaînes de Markov où les k-DPPs sont les lois stationnaires.

Le premier projet consiste à appliquer les processus déterminantaux à la sélection d'espèces diverses dans un ensemble d'espèces décrites par un arbre phylogénétique. En définissant le noyau du k-DPP comme un noyau d'intersection, les résultats fournissent une borne polynomiale sur le temps de mélange qui dépend de la hauteur de l'arbre phylogénétique.

Le second projet vise à utiliser le k-DPP dans un problème d'échantillonnage de sommets sur un graphe connecté de grande taille. La pseudo-inverse de la matrice Laplacienne normalisée est choisie d'étudier la vitesse de convergence de la chaîne de Markov créée pour l'échantillonnage de la loi stationnaire k-DPP. Le temps de mélange résultant est borné sous certaines conditions sur les valeurs propres de la matrice Laplacienne.

Le troisième sujet porte sur l'utilisation des k-DPPs dans la planification d'expérience avec comme objets d'étude plus spécifiques les hypercubes latins d'ordre n et de dimension d. La clé est de trouver un noyau positif qui préserve le contrainte de ce plan c'est-à-dire qui préserve le fait que chaque point se trouve exactement une fois dans chaque hyperplan. Ensuite, en créant une nouvelle chaîne de Markov dont le n-DPP est sa loi stationnaire, nous déterminons le nombre d'étapes nécessaires pour construire un hypercube latin d'ordre n selon le n-DPP.

Abstract

With the exponentially growing amount of data, sampling remains the most relevant method to learn about populations. Sometimes, larger sample size is needed to generate more precise results and to exclude the possibility of missing key information. The problem lies in the fact that sampling large number may be a principal reason of wasting time.

In this thesis, our aim is to build bridges between applications of statistics and k-Determinantal Point Process(k-DPP) which is defined through a matrix kernel. We have proposed different applications for sampling large data sets basing on k-DPP, which is a conditional DPP that models only sets of cardinality k. The goal is to select diverse sets that cover a much greater set of objects in polynomial time. This can be achieved by constructing different Markov chains which have the k-DPPs as their stationary distribution.

The first application consists in sampling a subset of species in a phylogenetic tree by avoiding redundancy. By defining the k-DPP via an intersection kernel, the results provide a fast mixing sampler for k-DPP, for which a polynomial bound on the mixing time is presented and depends on the height of the phylogenetic tree.

The second application aims to clarify how k-DPPs offer a powerful approach to find a diverse subset of nodes in large connected graph which authorizes getting an outline of different types of information related to the ground set. A polynomial bound on the mixing time of the proposed Markov chain is given where the kernel used here is the Moore-Penrose pseudo-inverse of the normalized Laplacian matrix. The resulting mixing time is attained under certain conditions on the eigenvalues of the Laplacian matrix.

The third one purposes to use the fixed cardinality DPP in experimental designs as a tool to study a Latin Hypercube Sampling(LHS) of order n. The key is to propose a DPP kernel that establishes the negative correlations between the selected points and preserve the constraint of the design which is strictly confirmed by the occurrence of each point exactly once in each hyperplane. Then by creating a new Markov chain which has n-DPP as its stationary distribution, we determine the number of steps required to build a LHS with accordance to n-DPP.

Publications

- D.Wehbe, N. Wicker, B. AL-Ayoubi and L. Moulinier. *Fixed-Size Determinantal Point Processes Sampling For Species Phylogeny*, 2018. Submitted to MathematicS In Action Journal.
- D.Wehbe and N. Wicker. *Efficient approximate sampling from k-DPP for large graphs*, 2018. Submitted to Theory and Applications of Graphs (TAG).

List of Figures

List of Tables

Introduction

Several types of probability sampling techniques have been introduced. These methods are known as simple random sampling, systematic sampling, stratified sampling, cluster sampling and multi-stage sampling. Actually, these types are unreasonable for sampling from a large population mainly when the focus is on capturing diverse samples since it leads to lose some crucial information. Furthermore, it is very time consuming and expensive to sample a large data set based on these probability sampling techniques.

Determinantal point processes (DPPs) are introduced in 1960's as probabilistic models that capture negative correlation and give the likelihood of selecting a subset of items as the determinant of a kernel matrix. Recently, a strong relation between DPPs and machine learning appeared and a DPP sampling algorithm has been devised. The main idea behind this algorithm is to select a diverse subset of given items. In fact, limiting the amount of sample elements plays an increasingly important role in real-world applications. For this reason, the extension to k-DPP is proposed, which is a conditional DPP that models only sets of cardinality k. By assuming that the eigen-decomposition of the DPP kernel is available, the sampling time for the k-DPP algorithm is $\mathcal{O}(Nk^3)$ where N is the number of elements present in the population. Actually, it is inefficient and sometimes not possible to compute in practice the eigen-decomposition of a huge matrix.

Since our focus is oriented to diversity and saving time, the best solution is to find a faster method based on k-DPP. Many studies show that Markov chain techniques are very appealing in the context of generating random samples of a k-DPP due to their simplicity and efficiency. Hence, the technique which we will follow is to construct rapidly mixing Markov chains which have the k-DPPs as their stationary distribution. Three different applications for sampling from k-DPP are represented in this thesis.

The first application interests the biologists where a fast k-DPP sampling for species phylogeny is offered. Since a tree sampling method is needed in many studies in modern bioinformatics, we use k-DPP to sample a diverse subset of species from a large phylogenetic tree. The k-DPP is defined through a symmetric positive define matrix which plays the main role in expressing the degree of similarity between any two species by comparing the ancestors of a species with the ancestors of another one. The most important step in this sampling method is constructing a Markov chain whose stationary distribution is the k-DPP. The aim behind this approach is to suggest a new configuration by choosing two elements: one to be removed from the set of size k and another to be added. The technique used to

study the mixing time of the Markov chain is based on the the bound of the Poincaré constant and on the fact that any k-DPP is a homogeneous SR distribution. We show that the convergence speed of this chain to its stationary distribution is reached in a polynomial time that depends on the height of the phylogenetic tree. The experiments confirm the usefulness and efficiency of this approach by showing that certain subsets are more likely to be sampled than others.

As the number of graph-structured data increases quickly, the second application is proposed for solving the problem of sampling a diverse subset of nodes from a connected graph. Following the same reasoning as of the first application, choosing the Moore-Penrose pseudo-inverse of the normalized Laplacian matrix as the kernel will be the right tool utilized to generate an approximate sample from k-DPP. A polynomial bound on the mixing time for Markov chain sampling from a k-DPP is stated under certain conditions on the eigenvalues of the normalized Laplacian matrix.

The third application is about making connections between fixed-size Determinantal Point Processes and experimental designs. The main idea is to build a Latin Hypercube sampling design of order n and dimension d with accordance to n-DPPs. Since DPPs give higher probability to points that are negatively correlated then a Latin Hypercube design with more spread out points is likely to be selected. This is reached by proposing a special DPP kernel that preserves the Latin hypercubes properties and by constructing an appropriate Markov chain which has n-DPP as its stationary distribution. The bound of the Poincaré constant here is achieved by using canonical paths where the lengths of these paths are taken into consideration. This leads to prove an upper bound on the total variation mixing time of the Markov chain.

This thesis is composed of four chapters:

In **chapter 1**, a background about the methods and the results appearing in this thesis is provided. First, a simple introduction about Determinantal Point Process is stated by presenting an extension of this process with fixed cardinality k, denoted k-DPP and by illustrating an algorithm for sampling from k-DPP. Then, an overall background about Markov chain and mixing time is given. Finally, a Markov chain with a k-DPP as its stationary distribution is defined and a technique to efficiently generate random samples of a k-DPP is described.

In **chapter 2**, the focus is on the problem of selecting a diverse set of species in an enormous phylogenetic tree. A specific matrix called the intersection kernel is proposed to define the DPP kernel. The effort here is to find a way to offer a polynomial bound on the mixing time of the lazy Markov chain specified in the previous chapter for the k-DPP. The results were that the usage of k-DPP had an influence on stating that the leaves joined by a higher subtree are more likely to appear. This outcome is achieved by showing that in a tree of maximum height h, for $0 < h_2 < h_1 < h/2$, choosing k species joined by a subtree of height h_1 is much more probable than choosing k species joined by a subtree of height h_2. At last, this approach is applied to a real case on a large dataset of species.

In **chapter 3**, a clarification of how k-DPPs offer a powerful approach to modeling diversity is stated by finding interesting nodes in a connected graph. The convergence speed of the constructed Markov chain into k-DPP is studied where the Moore-Penrose pseudo-inverse of the normalized Laplacian matrix is chosen as the DPP kernel. Then a polynomial bound on the mixing time is presented under certain conditions on the eigenvalues of the Laplacian matrix.

In **chapter 4**, the attention fall on displaying the performance of DPP with fixed cardinality in experimental designs. This approach leads to generating a Latin hypercube sampling of order n and dimension d from DPPs with fixed cardinality n, denoted n-DPP. The first step taken is choosing a positive kernel that preserve the Latin Hypercube sampling properties and then the second one is by constructing a new Markov chain which has n-DPP as its stationary distribution. The usage of n-DPP strategy allows the selection of LHS with more spread out points.

Chapter 1

Preliminaries

In this chapter some notions, background, methods and results are presented. The Determinantal point process is introduced and an extension of this process is presented with fixed cardinality k, denoted k-DPP. To sample from k-DPP an algorithm is needed. Then, an overall background about Markov chain and mixing time is illustrated. Finally, a Markov chain with a k-DPP as its stationary distribution is defined and the technique to efficiently generate random samples of a k-DPP is described.

1.1 Determinantal Point Processes

Determinantal point processes (DPPs) are coherent probabilistic models in the presence of negative correlation which arise in random matrix theory ([Mehta and Gaudin(1960)], [Ginibre(1965)]). DPPs were identified by [Machhi(1975)] who called them *fermion processes* that they have the ability to model the position of fermions, where nearby particles repel each other. Recently, [Kulesza and Taskar(2012)] provide a gentle introduction to DPPs by presenting new algorithms for inference problems like conditioning, marginalization and sampling. They showed that DPPs play a progressively important role in machine learning, for example they can be used to compute the marginals of a Markov random field, select diverse sets of sentences to form document summaries and model non-overlapping human poses in images or video. [Kulesza and Taskar(2011)] propose an extension of DPPs that permits to model the content of a fixed number of items. Indeed, for large data set it would be thriftless to model the size. For an integer $0 \leq k \leq n$, a k-DPP is obtained simply by conditioning a standard DPP on sampling sets of fixed cardinality k.
Moreover, [Affandi et al.(2012)] introduce a Markov-DPP (M-DPP) to model diverse sequences of subsets and they show the performance of M-DPP for sequentially displaying articles that are relevant and diverse on any given day. In addition, [Gillenwater et al.(2012)] propose a new algorithm to solve the DPP MAP problem which is finding the most likely configuration. This approach is based on continuous techniques for sub-modular maximization.

For a discrete set $\mathcal{X}$, there are two equivalent ways to define a DPP: via a marginal kernel K that gives rise to marginal probability of selecting a random subset, or via an L-ensemble by using a symmetric and positive semidefinite matrix L which only determines the probability of observing subsets where its cardinality is fixed. The matrix K is another positive semidefinite matrix where its eigenvalues are bounded above by one. Let's figure it out:

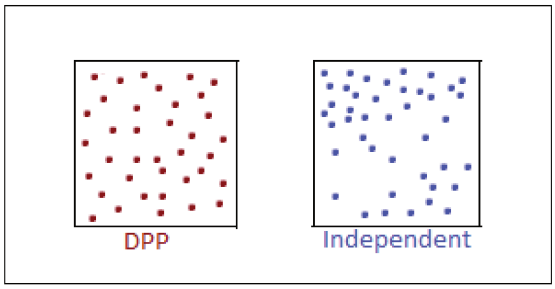

Figure 1.1: A set of points in the plane drawn from a DPP (left), and the same number of points sampled independently using a Poisson process (right).

A DPP, denoted $\mathcal{P}$, is a probability measure on the set $2^{\mathcal{X}}$ of all subset of $\mathcal{X}$ defined by the marginal kernel which is a semidefinite positive matrix $K \preceq I$ (i.e. all its eigenvalues are in the interval $[0, 1]$) indexed by the elements of $\mathcal{X}$ such that if $\mathbf{X}$ is a random subset drawn according to $\mathcal{P}$, we have,

$$\mathcal{P}(A \subseteq \mathbf{X}) = \det(K_A),$$

for every $A \subseteq \mathcal{X}$ and $K_A \equiv [K_{ij}]_{i,j \in A}$ indicates the restriction of K to the entries indexed by the elements of A.

Remark 1.1. *If $A = \{i\}$ is a singleton, then we have*

$$\mathcal{P}(i \in \mathbf{X}) = K_{ii}.$$

If $A = \{i, j\}$ is a two-element set, then we have

$$\begin{aligned} \mathcal{P}(i, j \in \mathbf{X}) &= K_{ii}K_{jj} - K_{ij}K_{ji} \\ &= \mathcal{P}(i \in \mathbf{X})\mathcal{P}(j \in \mathbf{X}) - K_{ij}^2. \end{aligned}$$

Here, K_{ii} presents the marginal probability of inclusion for element $i \in \mathcal{X}$ and K_{ij} gives the negative correlations between $(i, j) \in \mathcal{X}^2$ where $i \neq j$.

A DPP, denoted $\mathcal{P}_L$, can be defined via an L-ensemble by using a $n \times n$ positive semidefinite matrix L indexed by the elements of $\mathcal{X}$ such that if Y is a subset selected according to $\mathcal{P}_L$, we have

$$\mathcal{P}_L(Y) = \frac{\det(L_Y)}{\det(L + I)};$$

where I is the $n \times n$ identity matrix and $L_Y = (L_{ij})_{i,j \in Y}$. Note that the required normalization constant can be given in closed form for the reason that

$$\sum_{Y \subseteq \mathcal{X}} \det(L_Y) = \det(L + I).$$

The key point is that we can compute the marginal kernel K for the L-ensemble to get probabilities of item co-occurrence.

Theorem 1.1. *([Kulesza and Taskar(2012)]) An L-ensemble is a DPP, and its marginal kernel is*

$$K = L(L + I)^{-1}.$$

Conversely, L can be computed from a DPP with marginal kernel K as follows,

$$L = K(I - K)^{-1}.$$

In this thesis, we will work with the most relevant construction of k-DPPs which is based on L-ensemble.

A great effort has been made by DPP to model two distinct characteristics: the size of the set and its content. Actually, for huge data set it would be wasteful to model the size. For that reason, [Kulesza and Taskar(2011)] propose an extension of DPP called k-DPP, which is a conditional DPP that models only sets of cardinality k. [Kulesza and Taskar(2011)] present an example of k-DPPs on a real world image search problem where the goal is to show users diverse sets of images that correspond to their query. They showed that k-DPPs has better performance than Maximal Marginal Relevance (MMR) in generating diverse results set, where MMR is a technique that compute maximal diversity ranking in multi-document summarization and document retrieval ([Carbonell and Goldstein(1998)]). For more information about k-DPP and their applications we refer to recent studies by [Deshpande and Rademacher(2010)] and [Kulesza and Taskar(2012)].

The L-ensemble construction of a k-DPP, denoted $\mathcal{P}_L^k$, only gives positive probability to sets of cardinality k. $\mathcal{P}_L^k$ is given as follows:

$$\mathcal{P}_L^k(Y) = \frac{\det(L_Y)}{\sum_{|Y'|=k} \det(L_{Y'})}, \tag{1.1}$$

where $|\cdot|$ denote the cardinality of the set, $|Y| = k$ and L is a $k \times k$ positive semidefinite matrix.
A k-DPP can capture distributions whereas a standard DPP cannot. This refers to the normalization constant formula , where the sum in the k-DPP is on the same set of cardinality, while in the DPP, the sum is overall subsets of $\mathcal{X}$ of any cardinality.

1.1.1 Sampling from k-DPP

The extension to k-DPPs requires new algorithms to normalize, sample and marginalize based on recursions for the elementary symmetric polynomials.
The purpose of this section is to shed light on a sampling algorithm for k-DPPs, especially the one proposed by [Kulesza and Taskar(2012)]. Note that the eigen-decomposition of the kernel requires $\mathcal{O}(n^3)$ time and computing elementary symmetric polynomials can be obtained in $\mathcal{O}(nk)$ time. Further, by taking into consideration that k-DPP is a conditional DPP that models only sets of cardinality k, then sampling from k-DPP can be attained by sampling several times from the corresponding DPP until we get a sample of size k. This process can be reached in $\mathcal{O}(nk^3)$ times as shown in [Kulesza and Taskar(2012)]. Consequently, Algorithm 1 which arose from [Kulesza and Taskar(2012)] runs in $\mathcal{O}(nk^3)$ time and that is by supposing that the eigen-decomposition of the DPP kernel already exists.

Algorithm 1 Sampling from a k-DPP [[Kulesza and Taskar(2012)]]

Input: eigen-decomposition $\{(v_N, \lambda_N)\}_{N=1}^{n}$ of L, size k
$I \leftarrow \emptyset$
$e_0^N \leftarrow 1 \ \forall N \in \{0, 1, \cdots, n\}$
$e_0^m \leftarrow 0 \ \forall m \in \{1, \cdots, k\}$
for $m = 1, \cdots, k$ **do**
  for $N = 1, \cdots, n$ **do**
    $e_m^N \leftarrow e_m^{N-1} + \lambda_N e_{m-1}^{N-1}$
  end for
end for
$m \leftarrow k$
for $N = n, \ldots, 1$ **do**
  if $u \sim \mathcal{U}[0,1] < \lambda_N \frac{e_{m-1}^{N-1}}{e_m^N}$ **then**
    $I \leftarrow I \cup \{N\}$
    $m \leftarrow m - 1$
    if $m = 0$ **then**
      Break
    end if
  end if
end for
$V \leftarrow \{v_N\}_{N \in I}$
$Y \leftarrow \emptyset$
while $|V| > 0$ **do**
  Select y_i from Y with $Pr(y_i) = \frac{1}{|V|} \sum_{v \in V} (v^T e_i)^2$
  $Y \leftarrow Y \cup \{y_i\}$
  $V \leftarrow V_\perp$, an orthonormal basis for the subspace of V orthogonal to e_i
end while
Output: Y

1.2 Markov chain

This section is concerned to present some definitions, properties and consequences of Markov chains.

Let $S = \{1, \cdots, n\}$ be a finite set of states. Let $X_0, X_1, \cdots$ be a sequence of random variables with values in S.

Definition 1.1. *We say that $\left(X_n\right)_{n \geq 0}$ is a Markov chain on the finite state space S with transition probability $P = (p_{ij})_{i,j \in S}$ if it satisfies the Markov property:*

$$\begin{aligned} \mathcal{P}(X_n = s_n | X_{n-1} = s_{n-1}) &= \mathcal{P}(X_n = s_n | X_0 = s_0, \cdots, X_{n-1} = s_{n-1}) \\ &= p_{s_{n-1} s_n}(n-1) \end{aligned}$$

for any $s_0, \cdots s_n \in S$ and $n \in \mathbb{N}$. Then, $p_{s_{n-1} s_n}(n-1)$ is called transition probability from state s_{n-1} to state s_n in time $n-1$.

Further, if $\mathcal{P}(X_n = s_n|X_{n-1} = s_{n-1}) = p_{s_{n-1}s_n}$ is independent of n then the Markov chain is called time homogeneous. All Markov chains considered in this chapter are time homogeneous.

For all $n, m \geq 0$ and $s_i, s_j \in S$, the n-step transition probabilities of the chain are expressed as follows:

$$p^{(n)}_{s_i s_j} = \mathcal{P}(X_{m+n} = s_j|X_m = s_i).$$

Definitions 1.1.

- *A state s_j is reachable from s_i if for some $n \geq 0$, the n-step transition probability is strictly positive i.e. $p^{(n)}_{s_i s_j} > 0$.*
- *s_i and s_j communicate if s_j is reachable from s_i and s_i is reachable from j.*
- *A Markov chain is said to be irreducible if all states communicate with each other.*
- *A state s_i is recurrent if and only if $\sum_{n\geq 1} p^{(n)}_{s_i s_i} = \infty$.*
- *A recurrent state s_i is positive recurrent if $\mathbb{E}(T_{s_i}|X_0 = s_i) < \infty$ where T_{s_i} is the first return time to state s_i.*
- *A state s_i is periodic with period d if d is the smallest integer such that $p^{(n)}_{s_i s_i} = 0$ whenever n is not a multiple of d. If $d = 1$, then s_i is said to be aperiodic.*

Proposition 1.1. *Suppose we have two recurrent states s_i and s_j. If s_j is positive recurrent and if s_j communicate with s_i, then s_i is also a positive recurrent state. In particular, if the Markov chain is irreducible then all the states must be positive recurrent.*

If all states in an irreducible Markov chain are positive recurrent, then we say that the Markov chain is positive recurrent.

Proposition 1.2. *If s_i is periodic with period d and if s_i communicate with s_j, then s_j also is periodic with period d. In particular, if the Markov chain is irreducible then all the states must be periodic with period d.*

If all states in an irreducible Markov chain are periodic (respectively aperiodic), then we say that the Markov chain is periodic (respectively aperiodic).

Definition 1.2. *A stationary distribution of a Markov chain is a probability distribution $\pi = (\pi_0, \cdots, \pi_n)$ such that*

$$\pi = P\pi.$$

Definition 1.3. *A Markov chain is said to be ergodic if it is irreducible, positive recurrent and aperiodic. Then, the Markov chain converges to the stationary distribution.*

Theorem 1.2. *A finite, irreducible, aperiodic Markov chain is an ergodic Markov chain.*

Theorem 1.3. *An ergodic Markov chain has a unique stationary distribution, which is a limiting distribution, i.e. for all $s_i, s_j \in S$ we have*

$$\lim_{n\to\infty} p^{(n)}_{s_i s_j} = \pi_{s_j}.$$

In the following sections, the Markov chains will be considered as finite Markov chains.

1.2.1 Metropolis chain

Suppose π is a probability distribution on S and the goal is to use an arbitrary transition probability q to construct a Markov chain with stationary distribution π ([Chib and Greenberg(1995)]). The Metropolized chain is achieved when at state x, a new state y is generated from the transition probability $q(x,\cdot)$. The proposal state y is accepted as a new state in the chain with probability

$$\alpha = \min\left\{\frac{\pi(y)q(y,x)}{\pi(x)q(x,y)}, 1\right\}$$

or rejected with probability $1-\alpha$, hence the chain remains at x.

Moreover, by using the fact that the transition probability q can be symmetric ($q(x,y) = q(y,x)$), thus the acceptance function α will be expressed as the ratio $\frac{\pi(y)}{\pi(x)}$.
Then, for a stationary distribution π and a symmetric transition probability q, the transition matrix P for the Metropolis chain is defined as:

$$P(x,y) = \begin{cases} q(x,y)\cdot\min\left\{\frac{\pi(y)}{\pi(x)}, 1\right\} & \text{if } y \neq x \\ 1 - \sum\limits_{z\neq x} q(x,z)\cdot\min\left\{\frac{\pi(z)}{\pi(x)}, 1\right\} & \text{otherwise.} \end{cases}$$

Hence, the Metropolis chain with its stationary distribution π is reversible.

1.2.2 Mixing time

To determine the mixing time of a finite Markov chain the definition of the total variation distance is needed.

Definition 1.4. *Given two probability measure on space S denoted μ and ν. The total variation distance between μ and ν is defined as follows:*

$$\begin{aligned} ||\mu-\nu||_{TV} &= \sup_{A\subset S} |\mu(A)-\nu(A)| \\ &= \frac{1}{2}\sum_{s\in S} |\mu(s)-\nu(s)|. \end{aligned}$$

Theorem 1.4. *(Convergence Theorem) Supposing that $\mathcal{M}$ is an irreducible and aperiodic Markov chain and P its transition matrix such as $\pi P = \pi$. Then, there exist constants $\alpha \in (0,1)$ and $C > 0$ such that*

$$d(t) := \max_{x\in S} ||P^t(x,\cdot) - \pi||_{TV} \leq C\alpha^t,$$

where $P^t(x,\cdot)$ is the distribution of the chain started at x at time t.

We say that $\mathcal{M}$ converges asymptotically to π. Thus, the total variation distance between them approaches to zero.

Definition 1.5. *For $\epsilon > 0$, the total variation mixing time of $\mathcal{M}$ and π is defined as follows:*

$$\tau_\epsilon := \inf\{t : d(t) \leq \epsilon\}.$$

We say that the distribution of $\mathcal{M}$ after τ_ϵ approaches the chain's stationary distribution π.

1.3 Efficient sampling for k-DPP

[Kannan and Vempala(2009)], [Deshpande and Rademacher(2010)] and [Kulesza and Taskar(2012)] have presented algorithms for sampling from k-DPP. Although, for a large dataset these algorithms are inefficient to use since eigen-decomposition of a possibly huge matrix is needed. Nevertheless, the Markov chain techniques are very attractive because of their plainness and effectiveness in generating k-DPP random samples. For instance, the Metropolis-Hastings algorithm was used by [Kang(2013)] who has considered a Monte Carlo Markov chain (MCMC) to sample from k-DPP. In his work the coupling argument is ill-defined, hence the proof of the rapid mixing time of the Markov chain is wrong; however, the sampling scheme is right. We note that [Borcea et al.(2009)] show that any DPP is a strong Rayleigh (SR) distribution, defined by strong negative dependence properties. As SR distributions lead themselves to an effective Markov chain sampling: [Anari et al.(2016)] used a lazy MCMC which was also described by [Kang(2013)]. [Anari et al.(2016)] showed that the natural MCMC algorithm can be mixed quickly in the support of a homogeneous SR distribution which is a distribution over sets of a fixed size k. We know that DPPs are considered to be special cases of SR distributions which are closed under truncation, so any k-DPP is a homogeneous SR distribution. The results of [Anari et al.(2016)] infer that the same Markov chain can be used to efficiently generate random samples of a k-DPP. This result does not comprise the general DPP; concerning DPP, [Li et al.(2016b)] give a polynomial mixing time for sampling Markov chains from a general DPP.

We mention below some definitions needed to introduce a SR Measure and the proposition that shows that any DPP is strongly Rayleigh.

Definitions 1.2.

- *A polynomial $p : \mathbb{C}^n \to \mathbb{C}$ is said to be stable if $p(z_1, \cdots, z_n) \neq 0$ whenever $\Im(z_1), \cdots, \Im(z_n)$ are strictly positive.*
- *A polynomial $p : \mathbb{C}^n \to \mathbb{C}$ is said to be real stable if it is stable and all of its coefficients are real.*

Supposing that $\mu : 2^{[n]} \to \mathbb{R}_+$ is a probability measure on the set $2^{[n]}$ of all subsets of $[n] = \{1, \cdots, n\}$ that satisfies $\sum_{T \subseteq [n]} \mu(T) = 1$.

Definitions 1.3.

- *The generating polynomial of μ is defined as*
  $p_\mu : \mathbb{C}^n \to \mathbb{C}$
  $$z \to p_\mu(z) = \int z^T d\mu(T) = \sum_{T \subseteq [n]} \mu(T) \prod_{i \in T} z_i,$$
  where $z^T := \prod_{i \in T} z_i$.
- *If for any $T \subseteq [n]$, the number of elements in T is equal to k, then p_μ is a homogeneous polynomial of degree k and μ is said to be k-homogeneous.*

Definitions 1.4.

- *μ is said to be a strongly Rayleigh distribution if its generating polynomial p_μ is real stable.*

- *μ is said to be k-homogeneous strongly Rayleigh distribution if its generating polynomial p_μ is a homogeneous polynomial of degree k and real stable.*

Proposition 1.3.1. *[Borcea et al.(2009)] Let μ be a determinantal measure on the set $2^{[n]}$ i.e. μ is defined such that there is a square matrix M of size n in order that for any subset $Y \subseteq [n]$ where $M_Y = (M_{ij})_{i,j\in Y}$, we have*

$$\mu(\{S \in 2^{[n]} : Y \subseteq S\}) = \det(M_Y).$$

If μ is a determinantal measure induced by a positive matrix with eigenvalues in $[0, 1]$, then μ is strongly Rayleigh.

1.3.1 Construct a Markov chain for sampling from $\mathcal{P}_L^k$

The main idea of [Kang(2013)] and [Anari et al.(2016)] is to construct a Markov chain $\mathcal{M}$ which has $\mathcal{P}_L^k$ as the stationary distribution. This method is generated by the Metropolis-Hastings algorithm. Suppose that q is a proposal transition matrix. The purpose behind this approach is to suggest a new configuration by choosing two elements u and v: u to be removed from the current set X of size $|X| = k$, and v to be added. Hence, for $X = Y \cup \{u\}$ and $v \in \{1, \cdots, n\}\backslash X$ the acceptance probability of removing u from X and replace it with v is computed as follows:

$$p = \min\left\{1, \frac{\det L_{Y\cup\{v\}} \cdot q\big(Y \cup \{v\}, Y \cup \{u\}\big)}{\det L_{Y\cup\{u\}} \cdot q\big(Y \cup \{u\}, Y \cup \{v\}\big)}\right\}.$$

It is easy to see that q is a symmetric transition matrix, thus the acceptance probability is expressed as

$$p = \min\left\{1, \frac{\det L_{Y\cup\{v\}}}{\det L_{Y\cup\{u\}}}\right\}. \tag{1.2}$$

The advantage of having a rapidly-mixing Markov chain as a mean of sampling from a DPP is that when a new element is introduced or removed from X, we may simply continue the current chain until it is mixed again to obtain a sample from the new distribution. For large X, a single iteration will become very costly. For this reason, [Kang(2013)] presents a linear-algebraic manipulation of the determinant ratio where the explicit computation of the determinant is unnecessary. Since the determinant is permutation-invariant with respect to the index set and due to its symmetry, $L_{Y\cup\{u\}}$ is represented as the following block matrix form:

$$L_{Y\cup\{u\}} = \begin{pmatrix} L_Y & b_u \\ b_u^T & c_u \end{pmatrix},$$

where $b_u = L(i, u)_{i\in Y} \in \mathbb{R}^{|Y|}$ and $c_u = L(u, u)$. With this, the determinant of $L_{Y\cup\{u\}}$ is expressed as

$$\det(L_{Y\cup\{u\}}) = \det(L_Y)(c_u - b_u^T L_Y^{-1} b_u).$$

Since q is a symmetric transition matrix, this allows us to formulate the acceptance probability as

$$p = \min\left\{1, \frac{\det(L_{Y\cup\{v\}})}{\det(L_{Y\cup\{u\}})}\right\} = \min\left\{1, \frac{c_v - b_v^T L_Y^{-1} b_v}{c_u - b_u^T L_Y^{-1} b_u}\right\}.$$

Furthermore, the transition probability matrix P of $\mathcal{M}$ is given as:

$$P(Y\cup\{u\}, Y\cup\{v\}) = \begin{cases} q\big(Y\cup\{u\}, Y\cup\{v\}\big)\cdot\frac{1}{2}\min\left\{1, \frac{\det L_{Y\cup\{v\}}}{\det L_{Y\cup\{u\}}}\right\} & \text{if } v\neq u \\ 1 - \sum\limits_{w\neq u} q\big(Y\cup\{u\}, Y\cup\{w\}\big)\cdot\frac{1}{2}\min\left\{1, \frac{\det L_{Y\cup\{w\}}}{\det L_{Y\cup\{u\}}}\right\} & \text{otherwise.} \end{cases} \tag{1.3}$$

As $P(X,X) \geq \frac{1}{2}$ for all $X \subseteq [n]$, then Markov chain described above is said to be a lazy chain. This will guarantee that all eigenvalues of P are positive.

The lazy MCMC $\mathcal{M}$ produced by Metropolis-Hastings algorithm is introduced to acquire a sample from k-DPP as follows:

Algorithm 2 Markov chain for sampling from $\mathcal{P}_L^k$ [[Anari et al.(2016)]]

Require: Itemset $S = \{1, \cdots, n\}$, similarity matrix $L \succ 0$
  Randomly initialize state $X \subseteq S$, s.t. $|X| = k$
  Sample $r \sim \text{Unif}(0,1)$
  if $r < \frac{1}{2}$ **then**
    $X \leftarrow X$
  else
    while Not mixed **do**
      Sample $u \in X$, and $v \in S \setminus X$ u.a.r.
      Letting $Y = X \setminus \{u\}$, set

$$p \leftarrow \min\left\{1, \frac{c_v - b_v^T L_Y^{-1} b_v}{c_u - b_u^T L_Y^{-1} b_u}\right\}.$$

      $X \leftarrow Y \cup \{v\}$ with prob. p
      $X \leftarrow X$ with prob. $1-p$
    end while
  end if
  return X

The essential idea of this algorithm is to obtain a rapidly-mixing Markov chain whose stationary distribution is the k-DPP $\mathcal{P}_L^k$.

1.3.2 The mixing time of $\mathcal{M}$

In this section, we will refer to a theorem illustrated by [Anari et al.(2016)] which study the convergence speed of the distribution of $\mathcal{M}$ to its stationary distribution $\mathcal{P}_L^k$.
Note that P_μ in the following theorem is the transition probability matrix of $\mathcal{M}$ defined in Equation 1.3.

Theorem 1.5 ([Anari et al.(2016)]). *For any k-DPP $\mu : 2^{[n]} \to \mathbb{R}_+$ and supp$\{\mu\}$ is the state space of the lazy MCMC $\mathcal{M}$ defined in Algorithm 2. Let $X = Y \cup \{u\} \in supp\{\mu\}$ and $X' = Y \cup \{v\}$ where $u \in X$ and $v \notin X$. For $\epsilon > 0$,*

$$\tau_X(\epsilon) \leq \frac{1}{C_\mu} \cdot \log\Big(\frac{1}{\epsilon \cdot \mu(X)}\Big),$$

where the Poincaré constant is defined as follows:

$$C_\mu = \min_{X,X' \in supp\{\mu\}} \max(P_\mu(X,X'), P_\mu(X',X))$$

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

The proof of the above theorem is established on the bounds of the second largest eigenvalue of the transition probability matrix P_μ and by using the fact that any k-DPP is a homogeneous SR distribution.

1.4 Canonical Paths Method

Let (Ω, P, π) be a lazy irreducible Markov chain and reversible with respect to the stationary distribution. Note that Ω is the state space with $|\Omega| = n$, P is the transition probability distribution and π is the stationary distribution.
The basic idea here is to view (Ω, P, π) as a weighted undirected connected Graph G on a vertex set Ω, where the states represent the nodes and the transitions represent the edges ([Sinclair(1992)]).
Let us begin by some background information:

Notation 1.1. *The inner product for $L^2(\pi)$ denoted $\langle \cdot, \cdot \rangle_\pi$ is given by*

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

Definitions 1.5. *Let $f, g : \Omega \to \mathbb{R}$.*

- *The Dirichlet form associated with P is defined as follows:*

$$\varepsilon_\pi(f,g) := \frac{1}{2} \sum_{x,y \in \Omega} [f(x) - f(y)][g(x) - g(y)] P(x,y) \pi(x).$$

  In particular,

$$\varepsilon_\pi(f,f) = \frac{1}{2} \sum_{x,y \in \Omega} [f(x) - f(y)]^2 P(x,y) \pi(x).$$

- *The Variance of the function f is defined by*

$$Var_\pi(f) := \sum_{x \in \Omega} (f(x) - \mathbb{E}_\pi f)^2 \pi(x).$$

Furthermore, P (because of reversibility) satisfies the detailed balance condition: for all $x, y \in \Omega$,

$$Q(x,y) \equiv \pi(x) P(x,y) = \pi(y) P(y,x). \tag{1.4}$$

It is well known that in case of irreducibility, P has real eigenvalues that satisfy:

$$1 = \beta_0 > \beta_1 \geq \cdots \geq \beta_{n-1} \geq -1.$$

Since the Markov chain is introduced as a lazy chain, then all the eigenvalues are positive and the chain is aperiodic. Thus P is ergodic and the rate of convergence to π is based on the bound of the Poincaré constant of the chain which is $1-\beta_1$ and noted λ_1.
The following theorem gives the Poincaré constant in terms of the Dirichlet form ([Horn and Johnson(1985)]):

Theorem 1.6. *Assuming that P is reversible with respect to the probability distribution π on Ω, then the Poincaré constant is given by:*

$$\lambda_1 := \inf_{\mathbb{E}_\pi f=0} \frac{\varepsilon_\pi(f,f)}{Var_\pi(f)}, \tag{1.5}$$

where $f : \Omega \to \mathbb{R}$ is non-constant function.

For $x, y \in \Omega$, an edge of weight Q linking x and y, denoted $e = (x, y)$, is defined if and only if $Q(x, y) > 0$ where Q is given in Equation (1.4). As G is a connected graph, then, for every ordered pair of states (x, y) there exist a path between them which is noted δ_{xy}. This path can be built according to a series of states $x_0, \cdots, x_{n-1}$ knowing that x is supposed as the starting state and y is the arriving state where the transition probability between (x_i, x_{i+1}) is strictly positive for all $i \in \{0, \cdots, n-1\}$.
The basic goal of constructing canonical paths between each pair of states in G is to present a new bound for the Poincaré constant by taking into consideration the lengths of these paths.

Considering a family of canonical paths $\Gamma = \{\delta_{xy}\}$, the maximum loading of an edge by paths in terms of the lengths of the paths is evaluated by:

$$C = \max_e \left\{ \frac{1}{Q(e)} \sum_{x,y:e\in\delta_{xy}} \pi(x)\pi(y)|\delta_{xy}| \right\},$$

where $|\delta_{xy}|$ denotes the length of the path δ_{xy}.

Proposition 1.3 ([Sinclair(1992)])**.** *For any choice of canonical paths, the Poincaré constant defined in Equation (1.5) of a reversible Markov chain satisfies*

$$\lambda_1 \geq \frac{1}{C}.$$

The above proposition offer a bound for the second largest eigenvalue of P which is related to the mixing time. Hence, the results provided by [Diaconis and Stroock(1991)] is needed since they gives a bound on the mixing time in terms of λ_1 and the stationary distribution π. The following theorem is proceeded from [[Diaconis and Stroock(1991)], Proposition 3].

Theorem 1.7. *Supposing that (Ω, P, π) is a lazy irreducible and reversible Markov chain with Poincaré constant λ_1. Then, the mixing time is upper bounded by*

$$\tau_x(\epsilon) \leq \frac{1}{\lambda_1} \log\left(\frac{1}{\epsilon \cdot \pi(x)}\right) \quad \forall x \in \Omega, \epsilon > 0.$$

Bibliography

[Affandi et al.(2012)] R. H. Affandi, A. Kulesza and E. B. Fox. Markov determinantal point processes. *In Proc. UAI*, 2012.

[Anari et al.(2016)] N. Anari, S.O. Gharan, and A. Rezaei. Monte Carlo Markov chain algorithms for sampling strongly Rayleigh distributions and determinantal point processes. *In COLT*, pages 103-115, 2016.

[Borcea et al.(2009)] J. Borcea, P. Branden, and T.M. Liggett. Negative dependence and the geometry of polynomials. *Journal of American Mathematical Society* , volume 22, pages 521-567, 2009.

[Carbonell and Goldstein(1998)] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. *In Proc. SIGIR*, pages 335-336, 1998.

[Chib and Greenberg(1995)] S. Chib, and E. Greenberg. Understanding the Metropolis-Hastings algorithm. *American Statistician* , volume 49, pages 327-335, 1995.

[Deshpande and Rademacher(2010)] A. Deshpande and L. Rademacher. Efficient volume sampling for row/column subset selection. *FOCS*, number 1-3-4, 329-338, 2010.

[Diaconis and Stroock(1991)] P. Diaconis and D. Stroock. Geometric bounds for eigenvalues of Markov chains. *The Annals of Applied Probability*, volume 1, pages 36-61, 1991.

[Horn and Johnson(1985)] R. A. Horn and C. R, Johnson. Matrix Analysis. *Cambridge University Press*, 1985.

[Ginibre(1965)] J. Ginibre. Statistical ensembles of complex, quaternion, and real matrices. *Journal of Mathematical Physics* , volume 6, pages 440-449, 1965.

[Gillenwater et al.(2012)] J. Gillenwater, A. Kulesza and B. Taskar. Near-optimal MAP inference for determinantal point processes. *In Advances in Neural Information Processing Systems* , pages 2735-2743, 2012.

[Kannan and Vempala(2009)] R. Kannan and S. Vempala. Spectral algorithms. *Foundations and Trends in Theoretical Computer Science*, volume 4, pages 157-288, 2009.

[Kang(2013)] B. Kang. Fast Determinantal Point Process Sampling with Application to Clustering. *NIPS*, pages 2319-2327, 2013.

[Kulesza and Taskar(2012)] A. Kulesza and B. Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, volume 5, number 2-3, pages 123-286, 2012.

[Kulesza and Taskar(2011)] A. Kulesza and B. Taskar. kDPPs: fixed size determinantal point processes. *In Proceedings of the 28th International Conference on Machine Learning* , pages 1193-1200, 2011.

[Li et al.(2016b)] C. Li, S. Jegelka, and S. Sra. Fast Mixing Markov Chains for Strongly Rayleigh Measures, DPPs, and Constrained Sampling. *In Neural Information Processing Systems* , pages 4188-4196, 2016b.

[Machhi(1975)] O. Macchi. The Coincidence approach to stochastic point processes. *Advances in Applied Probability* , volume 7, number 1, pages 83-122, 1975.

[Mehta and Gaudin(1960)] M. L. Mehta and M. Gaudin. On the density of eigenvalues of a random matrix. *Nuclear Physics* , volume 18, pages 420-427, 1960.

[Sinclair(1992)] A. Sinclair. Improved bounds for mixing rates of Markov chains and multicommodity flow. *Combinatorics, Probability and Computing* ***1***, pages 351-370, 1992.

Chapter 2

k-DPP Sampling For Species Phylogeny

In this chapter we are interested in sampling diverse subsets of species from a large phylogenetic tree by avoiding redundancy. The technique used is based on k-DPP since they are utilized for diverse subset selection problems. This chapter investigate a simple MCMC algorithm to generate samples of a k-DPP. The results provide a fast mixing sampler for k-DPP, for which polynomial bound on the mixing time is presented. This approach can be applied to a real-world datasets of species, and it shows that the leaves joined by a higher subtree are more likely to appear.

2.1 Introduction

A rooted phylogenetic tree is a diagram that depicts the lines of evolutionary relationships among species from a common ancestor. We will assume that the tips of the tree represent species, a node represents the most recent common ancestor of all species descending from that point in the tree and the root is a single point from which it has been supposed that all species descend. The branch lengths may represent the accumulation of evolutionary change. In our case the tips may not line up because the rate of evolutionary change is not constant across all branches.
The goal is to present a method to sample the tree of life of complete genomes, which consists in the taxonomic tree of living organisms (the leaves of the tree) having their genome completely sequenced. Many studies in modern bioinformatics (comparative genomics, phylogeny inference studies, multiple sequence alignment building, ...) are based on a subset of the available complete sequenced genomes. Indeed, next-generation sequencing technologies led to an exponential number of available genomes that cannot be manually handled, hence, a tree sampling method is needed. Redundancy can thus be avoided while sampling when the work is linked to subsets of species.

Since determinantal point processes are probabilistic models that capture negative correlation and give the likelihood of selecting a subset of items as the determinant of a kernel matrix, thus, the main idea of this chapter is to sample, according to k-DPP a diverse set of species in an enormous phylogenetic tree which contains millions of nodes. The goal is to

show the performance of k-DPP in selecting a subset of species to cover a much greater set of species in a polynomial time.

To define a k-DPP, a positive semi-definite *Kernel* is needed. This kernel plays the main role in expressing the degree of similarity between any two nodes $x, x' \in \mathcal{X}$ with fine distinctions in the degree to which x and x' are distant from each other in the tree. As it is known, the species are connected to each other through ancestors. That is why our choice in this chapter fell on a specific kernel that is the intersection kernel that compares the ancestors of a species with the ancestors of another one.

Different algorithms were proposed for sampling from k-DPP, but these algorithms commonly need the eigendecomposition of a matrix which is typically of size more than one million ([Pundir et al.(2017)]). As noted in Chapter 1, [Anari et al.(2016)] used a lazy Markov chain Monte Carlo (MCMC) described also by [Kang(2013)] and showed that the natural MCMC algorithm mixes rapidly in the support of a k-DPP. For this reason, in this chapter the technique which we use for the convergence speed is based on Theorem 1.5.

Initially, Section 2.2 in this chapter gathers some basic facts about kernels functions and our k-DPP kernel. Our main result is presented in Section 2.3 stating that if the tree is of maximum height h, for $0 < h_2 < h_1 < h/2$, choosing k species simultaneously joined by a subtree of height h_1 is much more probable than choosing k species simultaneously joined by a subtree of height h_2. Then in Section 2.4 we focus on the mixing time of the Markov chain specified in Algorithm 2 for the k-DPP. The resultant mixing time depends on the height of the phylogenetic tree. In Section 2.5, we apply our approach to a real case on a large dataset of species. Finally, Section 2.6 presents our conclusions.

2.2 DPP Kernel

To provide the similarity between nodes in data $\mathcal{X}$ a kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is needed. By considering the discrete case, the way to characterize positive semidefiniteness is that for all sets of real coefficients $\{f_x\}$, we have

$$\sum_{x \in \mathcal{X}} \sum_{x' \in \mathcal{X}} f_x f_{x'} K(x, x') \geq 0.$$

Then, for finite $\mathcal{X}$, the kernel can be uniquely represented by $|\mathcal{X}| \times |\mathcal{X}|$ matrix with rows and columns indexed by the elements of $\mathcal{X}$ and related to the kernel by $K_{xx'} = K(x, x')$ and it is called Gram matrix.

Different kernels have successfully been applied to capture the long-range relationships between pairs of points induced by the local structure of the graph. The Laplacian matrix has been enormously effective for graph isomorphism problems, biochemistry and design of statistical experiments and take an important place in analysis of random walks and electrical networks on graphs ([Doyle and Snell (1984)], [Cvetkovic et al.(1980)], and [Merris(1994)]). Sometimes it is also known by the Kirchhoff matrix or the information matrix. The Lapla-

cian matrix of the graph is defined as $L = D - A$, where the elements of adjacency matrix A of the graph are:

$$A_{ij} = \begin{cases} 1 & \text{if } i \sim j \\ 0 & \text{otherwise} \end{cases}$$

and $D = Diag(A_i)$, with $A_i = \sum_j A_{ij}$.

The concept of pseudoinverse generalizes the matrix inverse to matrix that are not full rank or not square. Thus, it is more advantageous to use the pseudoinverse of the Laplacian matrix of the graph denoted by $L^\dagger$ where it is defined as a kernel on a graph (or a tree) and can be interpreted as a similarity measure ([Lovász(1993)]). Therefore, $L^\dagger$ can be used to compute the average commute time (see [Gobel and Jagers(1974)]), it is defined as the average number of steps taken by a random walker for reaching node j, when starting from node i and coming back to node i. Another quantity of interest, is the square root of the average commute time which is a distance measure between any two nodes, called by the Euclidean Commute Time Distance (ECTD). The elements of $L^\dagger$ are the inner products of the node vectors in a Euclidean space preserving the ECTD.

Moreover, on this occasion, it is interesting to mention the work of [Kondor and Lafferty(2002)] who made reference to a class of exponential kernels on graphs as Diffusion kernels. Then, they showed how these kernels correspond to standard Gaussian kernels in a continuous limit. Thereby, by analogy with the exponentiation of real numbers, the matrix exponential of L:

$$e^{\beta L} = \lim_{s\to\infty} \big(I + \frac{\beta L}{s}\big)^s, \quad s, \beta \in \mathbb{N}$$

where the limit always exists and it is equivalent to

$$e^{\beta L} = I + \beta L + \frac{\beta^2}{2!}L^2 + \frac{\beta^3}{3!}L^3 + \cdots .$$

An important effect of defining kernels in such an exponential form is that any power of symmetric matrix is positive semidefinite.

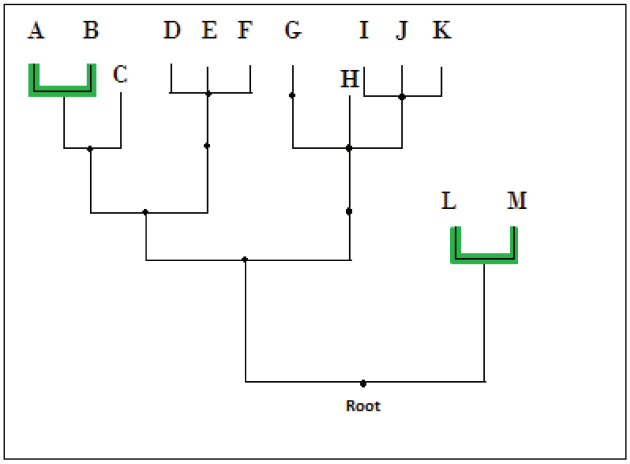

Figure 2.1: A subtree where the length of the longest path from the root to a leaf is 5.

In our case, we can notice that these two kernels are troublesome. Indeed, referring to the subtree in Figure 2.1, L and M are two close species that are not far from the root while A and B are also two close species but with long branches making them far from the root. If we suppose that the distance between A and B is similar to that between L and M, then we are going to have $K(A, B) \approx K(L, M)$ which is erroneous from a phylogenetical point of view.
When considering two species A and B, it is important to take into account that they are connected to each other through ancestors. Between their joining point and the root there is a common number of ancestors. For this reason, the most interesting kernel is the one that compares the set of the ancestors of species A with the set of the ancestors of species B, denoted by E_A and E_B respectively; the kernel is then defined as

$$K(A, B) = |E_A \cap E_B|, \tag{2.1}$$

where $|E_A \cap E_B|$ means the cardinality of the intersection of the two sets E_A and E_B. This function is a positive semi-definite kernel because it can be written as a dot product of binary vectors.

Thereby, the k-DPP kernel matrix L_X with $|X| = k$ is defined as $L_X = \mathbf{X}^T\mathbf{X}$ where the elements of the matrix $\mathbf{X}^T$ are: $\forall i \in \{1, \cdots, k\}$, $\forall j \in \{1, \cdots, N\}$ where N is the number of elements present in the tree,

$$\mathbf{X}^T_{ij} = \begin{cases} 1 & \text{if } j \text{ is an ancestor of } i \\ 0 & \text{otherwise.} \end{cases}$$

We illustrate this by an example where $k = 4$ and $N = 12$.

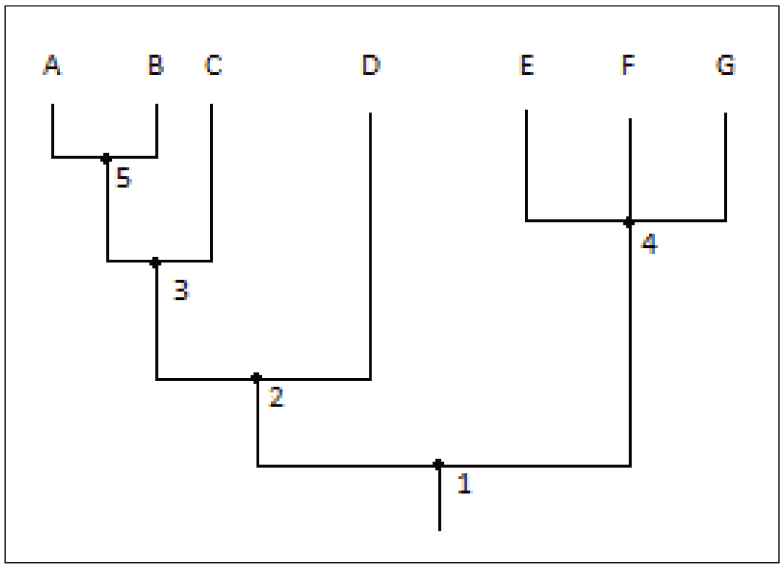

Figure 2.2: A subtree of height 5.

Let us take the subtree above in Figure 2.2, by choosing $X = \{A, C, D, G\}$, the matrix $\mathbf{X}^T$

is given as follows:

$$\begin{array}{c} \\ A \\ C \\ D \\ G \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 1 & 2 & 3 & 4 & 5 & A & B & C & D & E & F & G \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

and the kernel is equal to

$$L_X = \begin{pmatrix} 5 & 3 & 2 & 1 \\ 3 & 4 & 2 & 1 \\ 2 & 2 & 3 & 1 \\ 1 & 1 & 1 & 3 \end{pmatrix}.$$

Maybe it could be important to work with the normalized kernel, which can be defined by the following function:

$$\tilde{K}(A,B) = \frac{|E_A \cap E_B|}{\sqrt{|E_A||E_B|}}.$$

However with this normalization, we will face the same problem already suffered with the first two kernels. That is why our choice has been directed towards the intersection kernel (equation (2.1)).

2.3 The k-DPP selects diverse subsets

The k-DPP is ideal for selecting a diverse subset of given items; when selecting one item, the probability of simultaneously choosing a similar item is indeed low. In this section, we denote by h the height of the phylogenetic tree from the deepest leaf to the root and n the number of leaves (see figure below). To make things simple, we will consider a phylogenetic tree that is a perfect r-ary tree of height h, the number of nodes at depth d is then equal to r^d. For $0 < h_2 < h_1 < h/2$, Proposition 2.3.1 shows that choosing k species simultaneously joined by a subtree of height h_1 (that is, there are no leaves connected by a subtree of height $h_1 - 1$) is much more probable than choosing k species simultaneously joined by a subtree of height h_2 (that is, there are no leaves connected by a subtree of height $h_2 - 1$). This illustrates that k-DPP enables to achieve diversity in the most probable samples.

Proposition 2.3.1. *For a positive integer $r = k$ where $k \leq n$, let us consider a r-ary tree T of height h and let $0 < h_2 < h_1 < h/2$. Then choosing k leaves simultaneously joined by a subtree of height h_1 is $\left(\frac{h_1}{h_2}\right)^k \cdot \frac{1+k}{1+kh} \cdot \frac{k^{h-h_1}}{k^{h-h_2}} \cdot \left(\frac{k^{h_1}-1}{k^{h_2}-1}\right)^k$ times more probable than choosing k leaves simultaneously joined by a subtree of height h_2.*

Proof. Let h be the maximal height and let $0 < h_2 < h_1 < h/2$. Let A be a set containing k leaves joined all for the first time by a subtree of height h_1 and B a set containing k leaves joined by a subtree of height h_2.
Between any two leaves in the set A there are $h-h_1$ common ancestors. Thus, the intersection kernel L_A can be written as follows:

$$L_A = \begin{pmatrix} h & h-h_1 & \dots & h-h_1 \\ h-h_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & h-h_1 \\ h-h_1 & \dots & h-h_1 & h \end{pmatrix}.$$

Then, we can present the determinant of L_A as follows

$$\begin{aligned} \det L_A =& \begin{vmatrix} h & h-h_1 & \dots & h-h_1 \\ h-h_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & h-h_1 \\ h-h_1 & \dots & h-h_1 & h \end{vmatrix} \\ =& |(h-h_1)\mathbf{1}\mathbf{1}' + h_1 I_k| \\ =& h_1^k \Big| I_k + \frac{h-h_1}{h_1}\mathbf{1}\mathbf{1}' \Big| \\ =& h_1^k \Big(1 + \frac{h-h_1}{h_1}\mathbf{1}'\mathbf{1}\Big) \text{ by the matrix determinant lemma} \\ =& h_1^k \Big(1 + \Big(\frac{h}{h_1} - 1\Big)k\Big). \end{aligned}$$

Following this same reasoning for the set B that contains k leaves joined by a subtree of height h_2, between any two leaves there are $h - h_2$ common ancestors. Then, we have

$$\det L_B = h_2^k \Big(1 + \Big(\frac{h}{h_2} - 1\Big)k\Big).$$

Besides the number of subtrees of height h_1 (resp. h_2) with leaves being a subset of the leaves of T and the root a node at height h_1 (resp. h_2) in T is given by k^{h-h_1} (resp. k^{h-h_2}) and the number of leaves that can be chosen from a subtree of height h_1 (resp. h_2) is given by k^{h_1-1} (resp. k^{h_2-1}). Consequently, the ratio of choosing k leaves joined by a subtree of height h_1 to choosing k leaves joined by a subtree of height h_2 is expressed as:

$$\begin{aligned} \frac{\det L_A}{\det L_B} \cdot \frac{k^{h-h_1}}{k^{h-h_2}} \cdot \Big(\frac{k^{h_1-1}}{k^{h_2-1}}\Big)^k =& \left(\frac{h_1}{h_2}\right)^k \cdot \frac{1 + k\Big(\frac{h}{h_1} - 1\Big)}{1 + k\Big(\frac{h}{h_2} - 1\Big)} \cdot \frac{k^{h-h_1}}{k^{h-h_2}} \cdot \Big(\frac{k^{h_1-1}}{k^{h_2-1}}\Big)^k \\ \geq& \left(\frac{h_1}{h_2}\right)^k \cdot \frac{1+k}{1+k(h-1)} \cdot \frac{k^{h-h_1}}{k^{h-h_2}} \cdot \Big(\frac{k^{h_1-1}}{k^{h_2-1}}\Big)^k \quad \text{as } h_2 < h_1 < h/2 \\ \geq& \left(\frac{h_1}{h_2}\right)^k \cdot \frac{1+k}{1+kh} \cdot \frac{k^{h-h_1}}{k^{h-h_2}} \cdot \Big(\frac{k^{h_1-1}}{k^{h_2-1}}\Big)^k. \end{aligned}$$

Thus concluding the proof of the Proposition. □

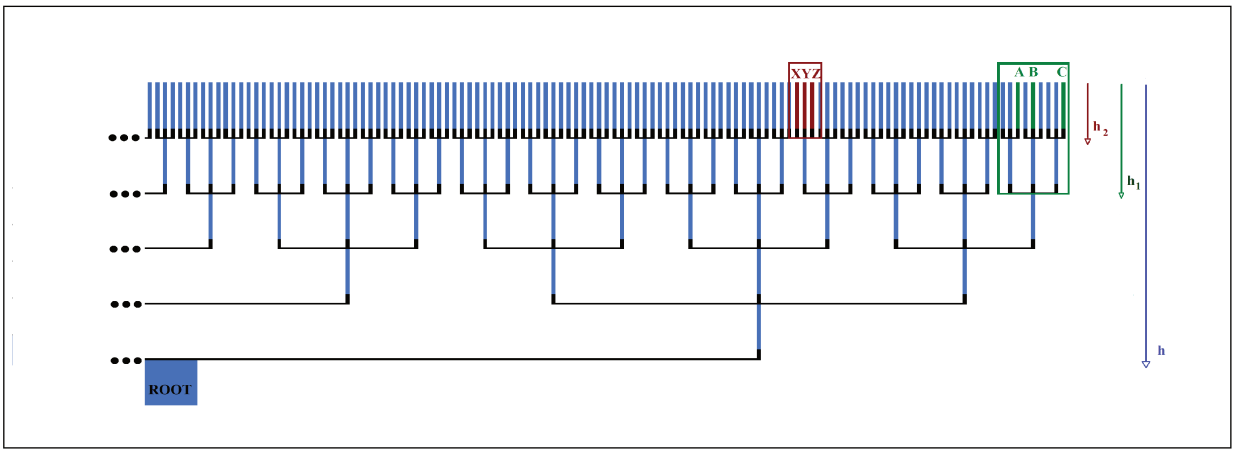

Figure 2.3: A part of a perfect 3-ary tree where the length of the longest path from the root to a leaf is $h = 5$. The species A, B and C are joined by a subtree of height $h_1 = 2$ and X, Y and Z are joined by a subtree of height $h_2 = 1$.

Now, we illustrate the results of this Proposition by considering the case of the perfect 3-ary tree represented in Figure 2.3 where $h = 5$, $h_1 = 2$ and $h_2 = 1$. Then according to Proposition 2.3.1 it is 18 times more likely to choose 3 leaves simultaneously joined by a subtree of height h_1 than from a subtree of height h_2.

2.4 Sampling via k-DPP

An L-ensemble is defined by $L = \mathbf{X}^T\mathbf{X}$, where $\mathbf{X}$ is $N \times n$ matrix with N the number of elements present in the tree and n the number of species. L is determined by $L(i,j) = k(x_i, x_j)$. It is inefficient to use Algorithm 1 proposed by [Kulesza and Taskar(2012)] since eigen-decomposition of L is needed. As mention in Chapter 1, DPPs are a special cases of strongly Rayleigh distributions which are closed under truncation, then any k-DPP is a homogeneous SR distributions. [Anari et al.(2016)] show that the natural MCMC algorithm mixes rapidly in the support of a homogeneous SR distributions. Accordingly, these last are distributions over all subsets of some fixed size k.
This method includes constructing a Markov chain with stationary distribution $\mathcal{P}_L^k$ generated by Metropolis-Hastings algorithm which proposes to select a row and column of L_X with $|X| = k$, and replace them with the row and column corresponding to the element to be added. By considering that q is the proposal transition matrix, therefore, the acceptance probability is the ratio of $\det(L_{X'}) \cdot q(X', X)$ to $\det(L_X) \cdot q(X, X')$, where $X = Y \cup \{u\}$, $X' = Y \cup \{v\}$ and u, v are the elements being removed and added, respectively.

2.4.1 Mixing Time

In this section, we will calculate the mixing time of the distribution of the Markov chain $\mathcal{M}$ described in Section 1.3 in Chapter 1 to the stationary distribution $\mathcal{P}_L^k$. The technique

which we use for the convergence speed is based on the study of [Anari et al.(2016)]. According to Theorem 1.5, for $\epsilon > 0$ the mixing time is bounded by

$$\tau_S(\epsilon) \leq \frac{1}{C_\mu} \cdot \log\Big(\frac{1}{\epsilon \cdot \mu(S)}\Big),$$

where $C_\mu = \min_{S,T \in \text{supp}\{\mu\}} \max(P_\mu(S,T), P_\mu(T,S))$, $\mu : 2^{[n]} \to \mathbb{R}_+$, $S \in \text{supp}\{\mu\}$ and $T = S\backslash\{u\} \cup \{v\}$ with $u \in S$ and $v \notin S$.
The following theorem gives the resulting mixing time of the Markov chain specified in Algorithm 2 for our k-DPP.

Theorem 2.1. *Let $\mathcal{P}_L^k$ be a k-DPP where L is the matrix defined by Equation (2.1). For any $\epsilon > 0$, the lazy Markov chain defined in Algorithm 2 generates ϵ-approximate sample of $\mathcal{P}_L^k$ in time*

$$\tau_\epsilon \leq 2k^2 n \cdot \log\left(\frac{n(h-1)}{\big(\epsilon \cdot \big(1+k(h-1)\big)\big)^{\frac{1}{k}}}\Big(1+\frac{k}{h-1}\Big)^{\frac{1}{k}}\right).$$

Proof. The proof is based on the main theorem of [Anari et al.(2016)] by considering μ as a k-DPP noted $\mathcal{P}_L^k$.
As a first step and according to Theorem 1.5, we need to lower bound

$$C_{\mathcal{P}_L^k} = \min_{X,X' \subseteq [n],\ |X|=|X'|=k} \max(P_{\mathcal{P}_L^k}(X,X'), P_{\mathcal{P}_L^k}(X',X)).$$

To do so, let us consider a set $X \subseteq [n]$ such that $|X| = k$ and choose an element $u \in X$ and $v \notin X$ uniformly and independently at random and let $Y = X \setminus \{u\}$. Following [Kang(2013)], the acceptance probability is lower bounded by the ratio of the determinants of two matrices as follows:

$$\frac{\det(L_{Y \cup \{v\}})}{\det(L_{Y \cup \{u\}})} = \frac{c_v - b_v^T L_Y^{-1} b_v}{c_u - b_u^T L_Y^{-1} b_u}, \tag{2.2}$$

where u and v are the elements being removed and added, respectively. Thus, the transition probability is given as follows:

$$P_{\mathcal{P}_L^k}\big(Y \cup \{u\}, Y \cup \{v\}\big) = q\big(Y \cup \{u\}, Y \cup \{v\}\big) \cdot \frac{1}{2} \min\left\{\frac{c_v - b_v^T L_Y^{-1} b_v}{c_u - b_u^T L_Y^{-1} b_u}, 1\right\}$$

where q is the proposal transition matrix.
Therefore, the lower bound of $C_{\mathcal{P}_L^k}$ is obtained by using the fact that q is a a symmetric transition matrix:

$$C_{\mathcal{P}_L^k} = \min_{Y,v,u} \max q\big(Y \cup \{u\}, Y \cup \{v\}\big)\left(\frac{1}{2}\min\left\{\frac{c_v - b_v^T L_Y^{-1} b_v}{c_u - b_u^T L_Y^{-1} b_u}, 1\right\}, \frac{1}{2}\min\left\{\frac{c_u - b_u^T L_Y^{-1} b_u}{c_v - b_v^T L_Y^{-1} b_v}, 1\right\}\right)$$
$$\geq \frac{1}{2kn}$$

which is the result of [Anari et al.(2016)].

Then, it remains the calculation of the lower bound of $\mathcal{P}_L^k$ to complete the proof. By using an intuitive geometric interpretation of determinants for a set X of cardinality k, we have

$$\det(L_X) = \text{Vol}(X)^2,$$

where L_X for the smallest volume is given by

$$L_X = \begin{pmatrix} h & h-1 & \dots & h-1 \\ h-1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & h-1 \\ h-1 & \dots & h-1 & h \end{pmatrix}.$$

Thereby,

$$\begin{aligned} \det L_X = & \begin{vmatrix} h & h-1 & \dots & h-1 \\ h-1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & h-1 \\ h-1 & \dots & h-1 & h \end{vmatrix} \\ =& |I_k + (h-1)\mathbf{1}\mathbf{1}'| \\ =& \big(1 + (h-1)\mathbf{1}'\mathbf{1}\big) \text{ by the matrix determinant lemma} \\ =& 1 + k(h-1). \end{aligned}$$

In addition, according to the definition of k-DPP, we should also calculate the normalization constant. Then, for any set X of cardinality k we want to choose $X' \subseteq [n]$ of size k maximizing $\det L_X$. Thus, by considering the intersection kernel we define $L_{X'}$ as follows:

$$L_{X'} = \begin{pmatrix} h & 1 & \dots & 1 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 1 & \dots & 1 & h \end{pmatrix}.$$

Then, we can present the determinant of $L_{X'}$ as follows

$$\begin{aligned} \det L_{X'} = & \begin{vmatrix} h & 1 & \dots & 1 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 1 & \dots & 1 & h \end{vmatrix} \\ =& |(h-1)I_k + \mathbf{1}\mathbf{1}'| \\ =& (h-1)^k \Big| I_k + \frac{1}{h-1}\mathbf{1}\mathbf{1}' \Big| \\ =& (h-1)^k \Big(1 + \frac{1}{h-1}\mathbf{1}'\mathbf{1}\Big) \text{ by the matrix determinant lemma.} \end{aligned}$$

Thus,

$$\det L_{X'} = (h-1)^k \Big(1 + \frac{k}{h-1}\Big).$$

Consequently,

$$\mathcal{P}_L^k(X) = \frac{\det(L_X)}{\sum_{|X'|=k} \det(L_{X'})} \geq \frac{1 + k(h-1)}{\begin{pmatrix} n \\ k \end{pmatrix} (h-1)^k \Big(1 + \frac{k}{h-1}\Big)} \geq \frac{n^{-k}\big(1 + k(h-1)\big)}{(h-1)^k \Big(1 + \frac{k}{h-1}\Big)}.$$

Now, by using Theorem 1.5 we can directly upper bound the mixing time in total variation

distance as follows:

$$\begin{aligned}\tau_X(\epsilon) \leq & \frac{1}{C_{\mathcal{P}_L^k}} \cdot \log\left(\frac{1}{\epsilon \cdot \mathcal{P}_L^k}\right) \\ \leq & 2kn \cdot \log\left(\frac{n^k(h-1)^k\left(1+\frac{k}{h-1}\right)}{\epsilon\left(1+k(h-1)\right)}\right) \\ \leq & 2k^2 n \cdot \log\left(\frac{n(h-1)}{\left(\epsilon \cdot \left(1+k(h-1)\right)\right)^{\frac{1}{k}}}\left(1+\frac{k}{h-1}\right)^{\frac{1}{k}}\right).\end{aligned}$$

This proves the result.

□

2.5 Experiments

As an evaluation result, we compare our method with a simple proportional method. Recursively, the number of samples to be shared at a given node is divided between its children proportionally to the number of descendants (leaves) attached to them. The two algorithms were applied to extract 200 sample from the "Eukaryota" (taxa ID 2759) sub-tree of the Tree of Life of complete genomes. The complete genomes and their taxonomy were taken from the UniProt (Universal Protein Resource) database [Pundir et al.(2017)] as of January 25 2018 and contains 3871 nodes including 1356 leaves. Since the resulting tree is huge, two subtrees were chosen and represented in Figures 2.4 and 2.5 where the yellow squares correspond to our method and the blue squares correspond to the proportional method.
A close inspection of the two sample sets leads to two conclusions. First, as Algorithm 2 is sensitive to the branching complexity, it favors divergent nodes as shown in the "Cranata" subtree. (The proportional method is indirectly sensitive to a sub-node complexity as a large number of descendants may be linked to a complex sub-node). Secondly, Algorithm 2 is more stable upon sampling repetition as it is guided by the tree structure where the proportional method use pure random choices without repulsion.

In practice, the algorithm is applied to a huge phylogenetic tree which contains 1827829 nodes from which 1464190 are species ([Pundir et al.(2017)]) making necessary the use of a MCMC algorithm.

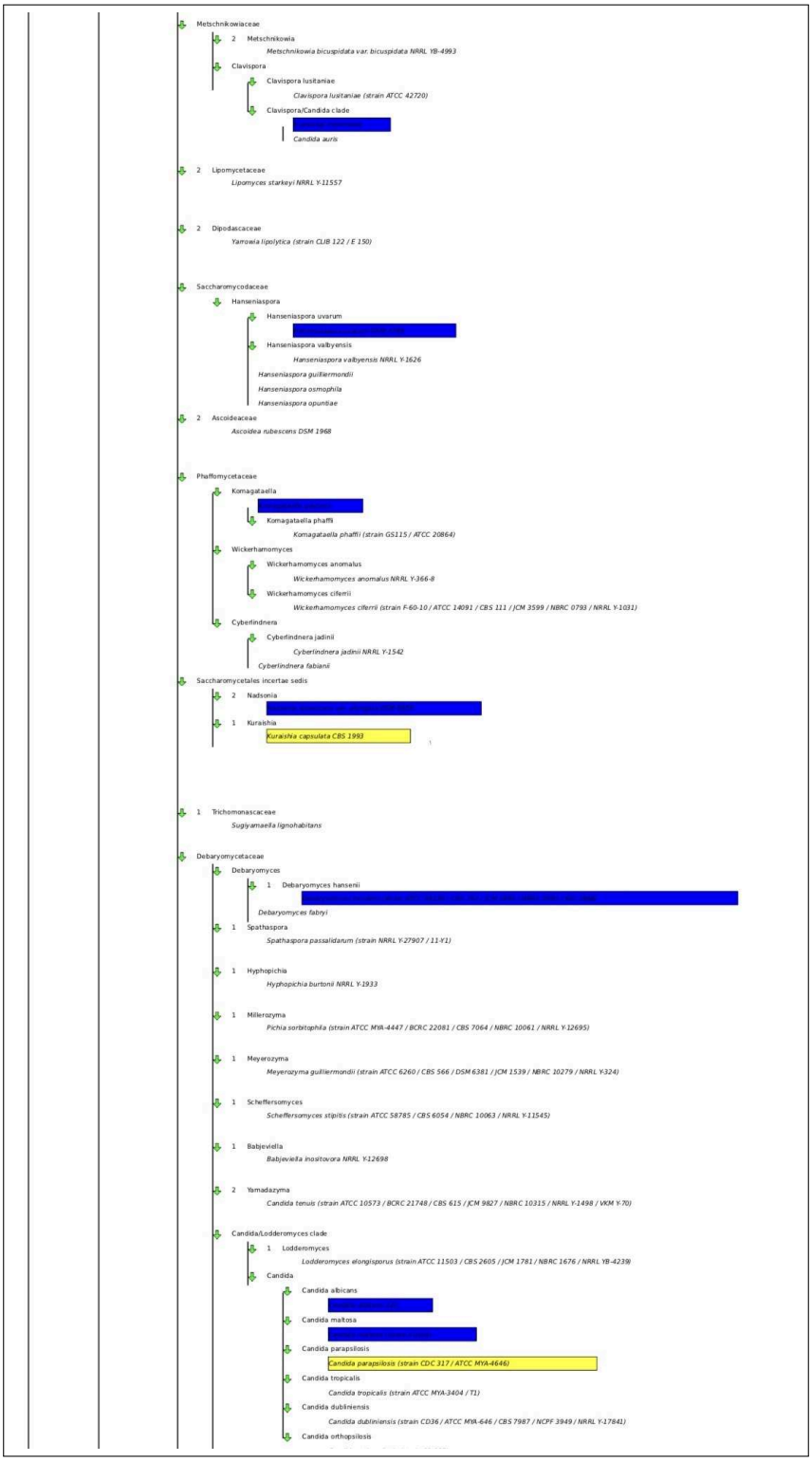

Figure 2.4: A subtree of a tree contains 3871 nodes including 1356 leaves where the two yellow squares correspond to the method presented in this chapter and the seven blue ones correspond to the proportional method

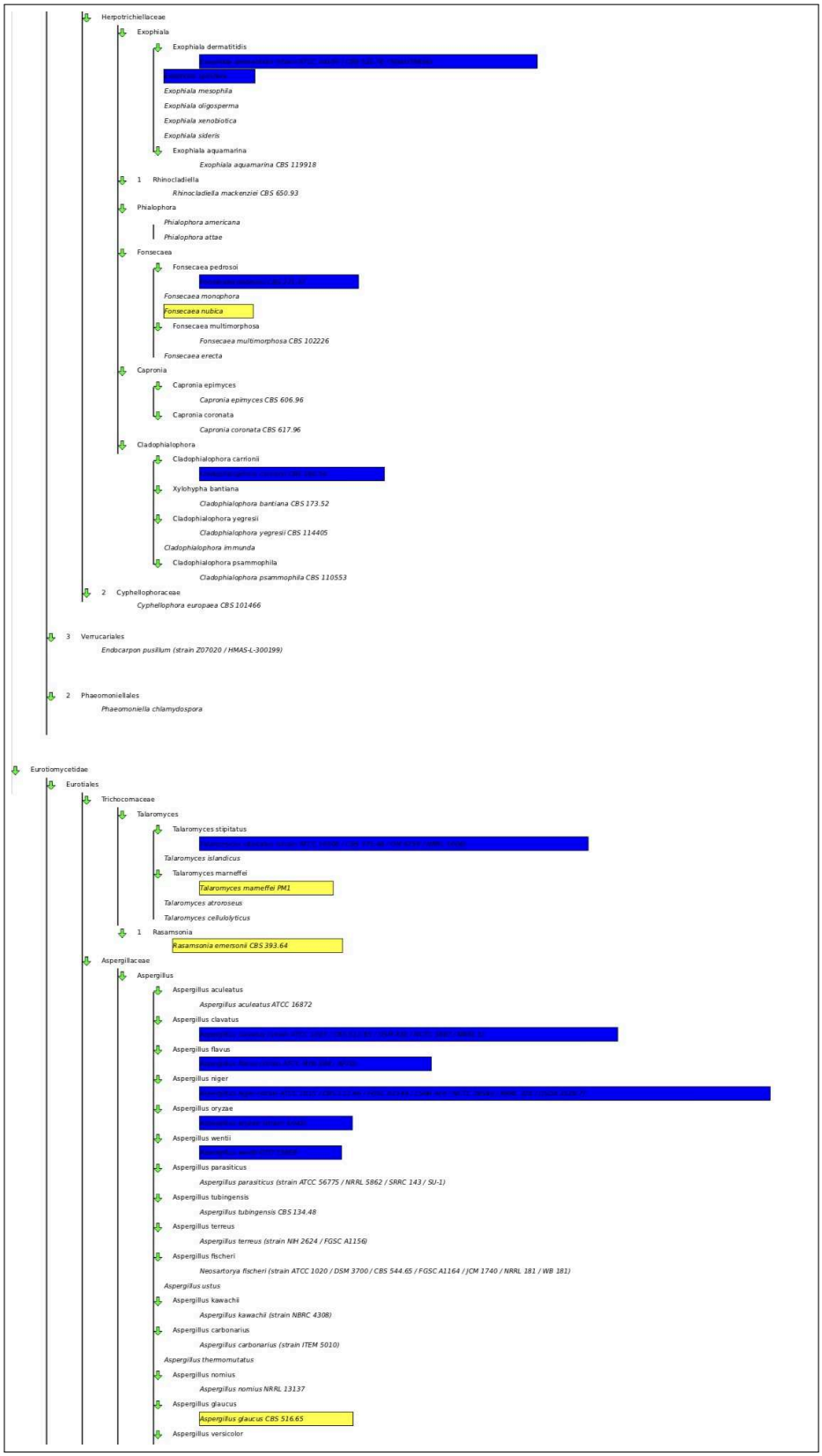

Figure 2.5: A subtree of a tree contains 3871 nodes including 1356 leaves where the four yellow squares correspond to the method presented in this chapter and the ten blue ones correspond to the proportional method

2.6 Conclusion

In this chapter, the main focus was on a particular case treated by the article of [Anari et al.(2016)]. They used a greedy algorithm of [Çivril and Magdon-Ismail(2009)] to generate a set $X \subseteq [n]$ such that $\mathcal{P}_L^k : 2^{[n]} \to \mathbb{R}_+$ is bounded away from zero. They found a set X such that $\det(L_X) \geq n^{-k}$. However, in Theorem 2.1 and according to the kernel chosen (equation (2.1)) we managed directly to bound $\det(L_X)$ without using the algorithm of [Çivril and Magdon-Ismail(2009)] allowing to gain substantive time at the initialization step and enhancing also the convergence speed bound.

The experiments attest the effectiveness and efficiency of this approach by showing that diverse subsets of species of size k selected from more than one million species are more likely to be sampled than less diverse ones and that is achieved in a polynomial time with respect to k and the number of species n.

Bibliography

[Anari et al.(2016)] N. Anari, S.O. Gharan, and A. Rezaei. Monte Carlo Markov chain algorithms for sampling strongly Rayleigh distributions and determinantal point processes. *In COLT*, pages 103-115, 2016.

[Çivril and Magdon-Ismail(2009)] A. Çivril and M. Magdon-Ismail. On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoretical Computer Science*, pages 4801-4811, 2009.

[Cvetkovic et al.(1980)] D.M. Cvetkovic, M. Doob and H.Sachs. Spectra of Graphs. *Academic Press, New York*, 1980.

[Doyle and Snell (1984)] P.G. Doyle and J.L. Snell. Random Walks and Electric Networks. *Mathematical Association of America*, 1984.

[Gobel and Jagers(1974)] F. Gobel and A. Jagers. Random walks on graphs. *Stochastic Processes and their Applications*, volume 2, pages 311-336, 1974.

[Kang(2013)] B. Kang. Fast Determinantal Point Process Sampling with Application to Clustering. *NIPS*, pages 2319-2327, 2013.

[Kondor and Lafferty(2002)] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. *In Proceedings of the* 19^{th} *International Conference On Machine Learning* , pages 315-322, 2002.

[Kulesza and Taskar(2012)] A. Kulesza and B. Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, volume 5, number 2-3, pages 123-286, 2012.

[Kulesza and Taskar(2011)] A. Kulesza and B. Taskar. kDPPs: fixed size determinantal point processes. *In Proceedings of the 28th International Conference on Machine Learning* , pages 1193-1200, 2011.

[Lovász(1993)] L. Lovász. Random walks on graphs: A survey. *Combinatorics, Paul Erdős is Eighty* , Volume 2, pages 1-46, 1993.

[Merris(1994)] R. Merris. Laplacian Matrices of Graphs: A Survey. *Linear Algebra and its applications* , volume 197-198, pages 143-176, 1994.

[Pundir et al.(2017)] S. Pundir, M. J. Martin, and C. O'Donovan. UniProt Protein Knowledgebase. *Methods Mol. Biol.* , volume 1558, pages 41-55, 2017.

Chapter 3

Efficient Approximate k-DPP Sampling For Large Graphs

Recently, the amount of graph-structured data available has been increasing rapidly. While the graph might contain similar nodes, our goal is to find a diverse subset of nodes which authorizes getting an outline of different types of information related to the ground set. The aim of this chapter is to sample a set of nodes from a large graph with accordance to k-DPPs. A polynomial bound on the mixing time for Markov chain sampling from a k-DPP is given under certain conditions on the eigenvalues of the Laplacian matrix.

3.1 Introduction

A graph is a set of objects (also called nodes) that are linked to each other by edges. For example, cities are nodes and highways are edges, humans are nodes and relationships between them are edges, likewise a video game can be considered as a graph where the states represent the nodes and the actions represent the edges that lead from one state to the next. Actually, we use graph applications daily. The navigation, PageRank in the Internet, genome assembly, computer chips, and game strategies can be the most useful applications of graphs.

Many applications of localizing the source of diffusion in a network, like locating the person who started a rumor in a social network, finding the computer that initiated the spreading of a computer virus in a network are based on a subset of nodes ([Zejnilović et al.(2013)]). Moreover, the focus of studies in modern bioinformatics (comparative genomics, multiple sequence alignment building...) need to select intersecting nodes and sets of nodes in graphs or networks.

Four flexible and generic methods called by the graph mining methods were presented by [Langohr(2014)] to find interesting nodes in graph. The first one incrementally selects one interesting node after another to produce a ranked list of relevant and non-redundant nodes. The second one is an iterative method that iteratively improve the overall interestingness of a set of nodes. The aim of the third method is to assign nodes to clusters and choose

a medoid node as a representative for each cluster. These three methods are based on the graph topology and a similarity or distance function for nodes. Lastly, the fourth method finds subsets of nodes characteristic for a class, such as a given node attribute value.
However, [Mavroforakis et al.(2015)] have addressed the problem of finding central nodes in a graph with respect to a set of query nodes. They studied a new notion of graph centrality based on absorbing random walks and developed efficient algorithms based on spectral clustering and on personalized PageRank. All these methods can, in principle, be applied to undirected weighted graph.

Here, we clarify how k-DPPs offer a powerful approach to modeling diversity by finding interesting nodes in connected graphs where the goal is to select a diverse set of relevant nodes according to user's query.
As the study of eigenvalues of graphs is needed to bound the determinant of the kernel and the presence of a Laplacian matrix is of hight importance according to the spectral graph theory. For this reason, the most interesting kernel is the Laplacian matrix. However, the advantage of normalizing the Laplacian is that its eigenvalues became of normalized form as well. Moreover, in our case the more the node is connected, the less is its probability to be chosen. That is why our choice in this chapter fell on the Moore-Penrose pseudo-inverse of the normalized Laplacian matrix.

Based on Theorem 1.5 in Chapter 1, choosing the Moore-Penrose pseudo-inverse of the normalized Laplacian matrix as the kernel will be the right tool utilized to generate an approximate sample from k-DPP. However, its outcome is established by a subset of nodes that abstract the large given set of nodes.

First, Section 3.2 gathers some basic facts about Laplacian matrix properties and Section 3.3, we define our k-DPP kernel and the Markov Chain to efficiently generate random samples of a k-DPP. Our main result is presented in Section 3.3.2 where we realize the characterization of Laplacian eigenvalues and we focus on the mixing time of the Markov chain for the k-DPP. Finally, Section 3.4 presents the conclusions.

3.2 Background

3.2.1 On graph Laplacian

We are given an undirected unweighted graph $G = (V, E)$ consists of a collection of nodes called vertices numbered from 1 to $n = |V|$, and connected by links called Edges E where $|E|$ is the number of edges in the graph. The degree of a vertex is the number of graph edges that are attached to it. In a finite graph, the degree sum formula says that the sum of the degrees of all vertices is twice the number of edges. A graph is called connected if for every pair of vertices there exists a path between them.
Several kernels have effectively been used to capture the long-range similarity between pairs of nodes induced by the local structure of the graph. The Laplacian matrix has been highly effective for graph isomorphism problems, design of statistical experiments and mod-

eling networks of resistors and occupies an important place in the study of electrical networks and analysis of random walks ([Cvetkovic et al.(1980)], [Doyle and Snell (1984)], and [Merris(1994)]). The combinatorial Laplacian matrix of the graph is defined as

$$L_G = D - A,$$

where the elements of the adjacency matrix A of the graph are:

$$A_{ij} = \begin{cases} 1 & \text{if } i \sim j \\ 0 & \text{otherwise} \end{cases}$$

and $D = \text{Diag}(d_1, \cdots, d_n)$, with $d_i = \sum_j A_{ij}$.
On occasion, it is also known as the Kirchhoff matrix or the information matrix. Besides, often the normalized Laplacian is used as it offers a simple probabilistic interpretation. The normalized Laplacian of the graph is given by:

$$\mathcal{L}_G := D^{-1/2} L_G D^{-1/2} = I - D^{-1/2} A D^{-1/2}.$$

For a graph G we can see that,

$$\mathcal{L}_{ij} = \begin{cases} 1 & \text{if } i = j \text{ and } d_j \neq 0 \\ -\frac{1}{\sqrt{d_i d_j}} & \text{if } i \sim j \\ 0 & \text{otherwise} \end{cases}$$

$\mathcal{L}_G$ is a symmetric, positive semidefinite matrix, and its eigenvalues $\lambda_1, \cdots, \lambda_n$ satisfy

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \leq 2.$$

The normalized Laplacian eigenvalues can be used to afford effective information about a graph. The number of connected components can be obtained from the multiplicity of the eigenvalue 0. If the graph G is connected then 0 is a simple eigenvalue of $\mathcal{L}_G$.

3.2.1.1 On the second smallest normalized Laplacian eigenvalues

Different types of studies were presented by [Li et al.(2008)], [Li et al.(2011)] and [Li et al.(2014)] on the second smallest normalized Laplacian eigenvalue which is, in turn, equal to the inverse of the largest eigenvalue of the pseudoinverse of the normalized Laplacian matrix. Since $\mathcal{L}_G$ is symmetric, its eigenvalues are all real and nonnegative. We recall some properties of the eigenvalues and eigenfunctions of $\mathcal{L}_G$ by using the variational characterization of those eigenvalues in terms of the Rayleigh quotient of $\mathcal{L}_G$.
Let G be a graph, $G = (V, E)$. Let $g \neq 0$ be a vector which can be viewed as a function that assigns a real value $g(v)$ to each vertex v of G. Then,

$$\frac{g^T \mathcal{L}_G g}{g^T g} = \frac{f^T D^{1/2} \mathcal{L}_G D^{1/2} f}{(D^{1/2} f)^T D^{1/2} f} = \frac{f^T L f}{f^T f} = \frac{\sum_{uv \in E(G)} (f(u) - f(v))^2}{\sum_{v \in V(G)} d(v)(f(v))^2}, \tag{3.1}$$

where $g = D^{1/2} f$.
It is easy to deduce that 0 is an eigenvalue of $\mathcal{L}_G$ and $D^{1/2} e$ is an eigenfunction corresponding to 0 where e denote the constant function which takes the value 1 on each vertex. Thus, by using equation (3.1) we can obtain the following formula corresponding to the second smallest normalized Laplacian eigenvalue of a graph G:

$$\lambda_2 = \inf_{f \perp D^{1/2}e} \frac{f^T L f}{f^T f} = \inf_{f \perp D^{1/2}e} \frac{\sum_{uv \in E(G)} (f(u) - f(v))^2}{\sum_{v \in V(G)} d(v)(f(v))^2}.$$

Moreover, λ_2 is closely related to the discrete Cheeger constant. For a subset S of V whose complement is $\overline{S} = V - S$, we define

$$h_G(S) = \frac{\sum\limits_{i \in S, j \in \overline{S}} A_{ij}}{\min\big(\sum\limits_{i \in S} d_i, \sum\limits_{j \in \overline{S}} d_j\big)},$$

where A_{ij} are the elements of the adjacency matrix.
The *Cheeger constant* h_G of a graph G is defined to be

$$h_G = \min_S h_G(S).$$

In the following theorem, the Cheeger inequality is introduced, which provides both upper and lower bounds of the second-smallest eigenvalue of the normalized Laplacian matrix.

Theorem 3.1. *(Cheeger Inequality)[Cheeger (1970)] If G is connected, then*

$$\frac{h_G^2}{2} \leq \lambda_2 \leq 2h_G,$$

where h_G is the Cheeger constant of G.

3.2.1.2 The Moore-Penrose pseudo-inverse of the normalized Laplacian matrix

The concept of pseudo-inverse generalizes the matrix inverse to matrices that are not full rank or not square. In fact, when the number of paths that link two nodes raises, the entries of the pseudo-inverse of the Laplacian matrix raise as well. This matrix represents the similarity between any pair of nodes. Then, it is motivating to use the pseudo-inverse of the normalized Laplacian matrix of the graph as a Gram matrix. This is a convenient way of defining a kernel on a graph ([Lovász(1993)]). The Moore-Penrose pseudo-inverse of the normalized Laplacian matrix $\mathcal{L}_G$ will be denoted $\mathcal{L}_G^\dagger$.
Some of the important properties of $\mathcal{L}_G^\dagger$ are :

- $\mathcal{L}_G^\dagger$ is symmetric, positive and semidefinite.
- If $(u_i,\ \lambda_i \neq 0)$ are (eigenvectors, eigenvalues) of $\mathcal{L}_G$, then $(u_i,\ \lambda_i^{-1} \neq 0)$ are the corresponding (eigenvectors, eigenvalues) of $\mathcal{L}_G^\dagger$.
- If $(u_i,\ \lambda_i = 0)$ are (eigenvectors, eigenvalues) of $\mathcal{L}_G$, then they are also (eigenvectors, eigenvalues) of $\mathcal{L}_G^\dagger$.

Interestingly, the pseudo-inverse of the Laplacian matrix can be used to figure the average commute time (see [Gobel and Jagers (1974)]), which is defined as the average number of steps taken by a random walker for reaching node j and coming back to node i when starting from node i. Let V_G denote the volume of the graph, the average commute time can be computed as follows:

$$n(i,j) = V_G\Big(\big(\mathcal{L}_G^\dagger\big)_{ii} + \big(\mathcal{L}_G^\dagger\big)_{jj} - 2\big(\mathcal{L}_G^\dagger\big)_{ij}\Big)$$

with $V_G = \sum_{l=1}^{n} d_{ll}$.
One more amount of interest, is the square root of the average commute time which is a distance measure between any two nodes, called the Euclidean Commute Time Distance (ECTD).

The following interlacing eigenvalues lemma is well-known in matrix analysis. It will be applied in this chapter on $\mathcal{L}_G^\dagger$. First, let us denote by $\lambda_k(A)$ the k-th smallest eigenvalue of A.

Lemma 3.1 ([Horn and Johnson(1985)])**.** *Let M be a real symmetric matrix and M_r denote any r-by-r principal submatrix of M. For any integer k such that $1 \leq k \leq r$ we have*

$$\lambda_k(M) \leq \lambda_k(M_r) \leq \lambda_{k+n-r}(M).$$

3.3 Sampling nodes via k-DPP

3.3.1 DPP Kernel

k-DPP is typical for selecting a diverse subset of given items; when selecting one item, the probability of simultaneously pick a similar item must be low. In such case, the more the node is connected, the less is its probability to be chosen. Hence, the Moore-Penrose pseudo-inverse of the normalized Laplacian matrix is an excellent kernel to choose. Now, we can define a k-DPP via this kernel:

Let $G = (V, E)$ be a finite connected graph and $n = |V(G)|$. The kernel we are using is given $\forall i, j \in \{1, \cdots, k\}$ by:

$$L(i, j) = \left(\mathcal{L}_G^\dagger\right)_{ij}. \tag{3.2}$$

The purpose here is to study the convergence speed of the Markov chain $\mathcal{M}$ described in Algorithm 2 which have $\mathcal{P}_L^k$ as stationary distribution where L is given in Equation 3.2. The method which we use for the convergence speed is based on the results of [Anari et al.(2016)].

3.3.2 Convergence Theorem

As the acceptance probability (equation (1.2)) is lower bounded by the ratio of the determinants of two matrices, this require us to bound the spectrum of L which is a submatrix of $\mathcal{L}_G^\dagger$. Since in a connected graph 0 is a single eigenvalue of $\mathcal{L}_G$, then it is a single eigenvalue of $\mathcal{L}_G^\dagger$. This gives a hint that under some conditions the eigenvalues of L_X can be lower bounded by a strictly positive constant. This is achieved by using the fact that $X \subset V$ as usually $k < n$. The following proposition will guarantee that all eigenvalues of L_X are strictly positive.

Proposition 3.3.1. *Let G be a connected graph. The submatrix obtained from the Moore-Penrose pseudo-inverse of the normalized Laplacian matrix $\mathcal{L}_G^\dagger$ by deleting its i-th row and*

i-th column will be denoted by $\underline{\mathcal{L}}_G^{\dagger}$. Then, all eigenvalues of $\underline{\mathcal{L}}_G^{\dagger}$ are strictly positive,

$$0 < \lambda_1(\underline{\mathcal{L}}_G^{\dagger}) \leq \cdots \leq \lambda_{n-1}(\underline{\mathcal{L}}_G^{\dagger}).$$

Proof. Let $\underline{\mathcal{L}}_G$ be the submatrix obtained from the normalized Laplacian matrix $\mathcal{L}_G$ by deleting its i-th row and i-th column where $\mathcal{L}_G = \mathcal{W}\mathcal{D}\mathcal{W}'$.
As G is a connected graph, for all non-zero $a \in \mathbb{R}^{n-1}$ we have

$$a'\underline{\mathcal{L}}_G a = a'\underline{\mathcal{W}}\mathcal{D}\underline{\mathcal{W}}'a > 0.$$

Set $V' = a'\underline{\mathcal{W}}$, therefore

$$\begin{aligned} a'\underline{\mathcal{L}}_G a =& V'\mathcal{D}V = \sum_{i=1}^{n} V_i^2 \mathcal{D}_{ii} = \sum_{i=2}^{n} V_i^2 \mathcal{D}_{ii} \quad (\text{as } \lambda_1 = 0) \\ \Rightarrow & \sum_{i=2}^{n} V_i^2 \lambda_i > 0 \end{aligned}$$

Thereby, with regard to the pseudo-inverse of the matrix and as G is a connected graph we have,

$$\sum_{i=2}^{n} V_i^2 \frac{1}{\lambda_i} > 0 \Rightarrow a'\underline{\mathcal{W}}\mathcal{D}^{\dagger}\underline{\mathcal{W}}'a > 0 \Rightarrow a'\underline{\mathcal{L}}_G^{\dagger} a > 0.$$

Thus, all eigenvalues of $\underline{\mathcal{L}}_G^{\dagger}$ are strictly positive. □

As the eigenvalues of $\mathcal{L}_G^{\dagger}$ are the inverse of those of $\mathcal{L}_G$, thus, to bound the largest eigenvalue of $\underline{\mathcal{L}}_G^{\dagger}$, the lower bound of the smallest eigenvalue of $\underline{\mathcal{L}}_G$ is needed. The main idea in the following lemma is to find a lower bound of $\lambda_1(\underline{\mathcal{L}}_G)$.

Lemma 3.2. *Let $I_G = \frac{d_{\min}}{m}$ where $d_{\min} = \min_i d_i$ with d_i is the degree of node i and m is the number of edges. Let $B_n = 2\sqrt{\frac{d_n}{2m}\left(1 + \frac{1}{d_{\min}}\right)}$ and $C_n = -2\lambda_2(\mathcal{L}_G) + \frac{2}{\sqrt{d_{\min}}} + 2\frac{d_n}{m}$.*
For all $n \in \mathbb{N}^*$ and $d_{\max} = \max_i d_i$, if $\frac{B_n}{C_n} \longrightarrow 0$ and $\lambda_2(\mathcal{L}_G) - \frac{1}{\sqrt{d_{\min}}} > \frac{d_{\max}}{m}$, then I_G is a lower bound of $\lambda_1(\underline{\mathcal{L}}_G)$.

Proof. Let us figure out the first eigenvalue of $\underline{\mathcal{L}}_G$ as

$$\begin{aligned} \lambda_1(\underline{\mathcal{L}}_G) =& \lambda_1 \begin{pmatrix} \underline{\mathcal{L}}_G & 0 \\ 0 & 1 \end{pmatrix} \\ =& \lambda_1 \left[\mathcal{L}_G + \begin{pmatrix} 0 & \dots & 0 & l_1 \\ \vdots & & \vdots & \vdots \\ 0 & \dots & 0 & l_{n-1} \\ l_1 & \dots & l_{n-1} & 0 \end{pmatrix} \right], \end{aligned}$$

where for all $i \in \{1, \cdots, n-1\}$, $l_i = \frac{\delta_{in}}{\sqrt{d_i d_n}}$ with $d_1, \dots, d_n$ are the degrees of the vertices and $\delta_{in} = \begin{cases} 1 & \text{if } i \sim n \\ 0 & \text{otherwise} \end{cases}$.

The matrix $\begin{pmatrix} 0 & \dots & 0 & l_1 \\ \vdots & & \vdots & \vdots \\ 0 & \dots & 0 & l_{n-1} \\ l_1 & \dots & l_{n-1} & 0 \end{pmatrix}$ has only 0 and $\mu \in \{\mu_1, \mu_2\}$ as eigenvalues.

In order to find μ, the following system of linear equations is used:

$$\begin{cases} \mu x_1 &= & l_1 x_n \\ \vdots & & \vdots \\ \mu x_{n-1} &= & l_{n-1} x_n \\ \mu x_n &= & \sum_{i=1}^{n-1} l_i x_i \end{cases}$$

thereby $\mu x_n = \sum_{i=1}^{n-1} l_i^2 \frac{x_n}{\mu}$ hence $\mu = \pm\sqrt{\sum_{i=1}^{n-1} l_i^2}$. Next, eigenvectors associated with

$$\mu_1 = \sqrt{\sum_{i=1}^{n-1} l_i^2} \text{ and } \mu_2 = -\sqrt{\sum_{i=1}^{n-1} l_i^2}$$

are respectively

$$u_1' = \Big(l_1, \dots, l_{n-1}, \sqrt{\sum_{i=1}^{n-1} l_i^2}\Big) \text{ and } u_2' = \Big(l_1, \dots, l_{n-1}, -\sqrt{\sum_{i=1}^{n-1} l_i^2}\Big).$$

Now we can proceed with bounding of $\lambda_1(\underline{\mathcal{L}}_G)$. We note $a = \sum_{i=1}^{n} a_i V_i$ with V_i is the eigenvector of $\mathcal{L}_G$ associated with $\lambda_i(\mathcal{L}_G)$ and forming an orthonormal basis. For that, for all $i \in \{1, \cdots, n\}$ we have $|a_i| \leq 1$ and $\sum_{i=1}^{n} a_i^2 = 1$, without loss of generality, we suppose that $a_1 \geq 0$ and minimize $a' \underline{\mathcal{L}}_G a$ to bound $\lambda_1\big(\underline{\mathcal{L}}_G\big)$.
This yields,

$$\begin{aligned} a' \begin{pmatrix} \underline{\mathcal{L}}_G & 0 \\ 0 & 1 \end{pmatrix} a &= \left(\sum_{i=1}^{n} a_i V_i\right)' \left(\mathcal{L}_G + \mu_1 \frac{u_1 u_1'}{||u_1||^2} + \mu_2 \frac{u_2 u_2'}{||u_2||^2}\right) \left(\sum_{i=1}^{n} a_i V_i\right) \\ &\geq \left(\sum_{i=2}^{n} a_i V_i\right)' \left(\mathcal{L}_G + \mu_1 \left(\frac{u_1 u_1'}{||u_1||^2} - \frac{u_2 u_2'}{||u_2||^2}\right)\right) \left(\sum_{i=2}^{n} a_i V_i\right) + \frac{\mu_1}{||u_1||^2} a_1^2 \big((V_1' u_1)^2 - (V_1' u_2)^2\big) + \\ &\quad 2(a_1 V_1') \left[\mu_1 \frac{u_1 u_1'}{||u_1||^2} - \mu_1 \frac{u_2 u_2'}{||u_2||^2}\right] \left(\sum_{i=2}^{n} a_i V_i\right) \\ &\geq \big(\lambda_2(\mathcal{L}_G) - \mu_1\big) \left|\left| \sum_{i=2}^{n} a_i V_i \right|\right|^2 + \frac{1}{2\mu_1} a_1^2 \left[\left(\sum_{i=1}^{n} V_{1i} u_{1i}\right)^2 - \left(\sum_{i=1}^{n} V_{1i} u_{2i}\right)^2\right] + \\ &\quad 2 \frac{a_1 V_1'}{2\mu_1} \big[u_1 u_1' - u_2 u_2'\big] \left(\sum_{i=2}^{n} a_i V_i\right). \end{aligned}$$

Recalling that the matrix $u_1 u_1' - u_2 u_2'$ is as follows:

$$u_1 u_1' - u_2 u_2' = 2\mu_1 \begin{pmatrix} 0 & \dots & 0 & l_1 \\ \vdots & & \vdots & \vdots \\ 0 & \dots & 0 & l_{n-1} \\ l_1 & \dots & l_{n-1} & 0 \end{pmatrix}.$$

Then, by introducing

$$w' := V_1'(u_1u_1' - u_2u_2') = 2\mu_1\left(V_{1n}l_1, \cdots, V_{1n}l_{n-1}, \sum_{i=1}^{n-1} l_iV_{1i}\right)$$

and by using the fact that $\mathcal{L}_G$ has an orthonormal basis of eigenvectors, we get

$$\begin{aligned}
a'\underline{\mathcal{L}}_G a &\geq \big(\lambda_2(\mathcal{L}_G) - \mu_1\big)\sum_{i=2}^{n} a_i^2 + \frac{a_1^2}{2\mu_1}\left[\left(\sum_{i=1}^{n} V_{1i}u_{1i} - \sum_{i=1}^{n} V_{1i}u_{2i}\right)\left(\sum_{i=1}^{n} V_{1i}u_{1i} + \sum_{i=1}^{n} V_{1i}u_{2i}\right)\right] + \\
&\quad \frac{a_1}{\mu_1}2\mu_1\left(V_{1n}l_1, \cdots, V_{1n}l_{n-1}, \sum_{i=1}^{n-1} l_iV_{1i}\right)\left(\sum_{i=2}^{n} a_iV_i\right) \\
&\geq \big(\lambda_2(\mathcal{L}_G) - \mu_1\big)\big(1 - a_1^2\big) + \frac{a_1^2}{2\mu_1}\left[\left(\sum_{i=1}^{n} V_{1i}(u_{1i} - u_{2i})\right)\left(\sum_{i=1}^{n} V_{1i}(u_{1i} + u_{2i})\right)\right] + \frac{a_1}{\mu_1}\left[w'\left(\sum_{i=2}^{n} a_iV_i\right)\right] \\
&\geq \big(\lambda_2(\mathcal{L}_G) - \mu_1\big)\big(1 - a_1^2\big) + \frac{a_1^2}{2\mu_1}\left[\left(V_{1n}\left(2\sqrt{\sum_{i=1}^{n-1} l_i^2}\right)\right)\left(\sum_{i=1}^{n-1} V_{1i}(2l_i)\right)\right] - \frac{a_1}{\mu_1}\left|\left|w'\left(\sum_{i=2}^{n} a_iV_i\right)\right|\right| \\
&\geq \big(\lambda_2(\mathcal{L}_G) - \mu_1\big)\big(1 - a_1^2\big) + 2a_1^2V_{1n}\left(\sum_{i=1}^{n-1} V_{1i}l_i\right) - \frac{a_1}{\mu_1}||w||\left|\left|\sum_{i=2}^{n} a_iV_i\right|\right|.
\end{aligned}$$

Or, $D^{\frac{1}{2}}e$ is an eigenvector of $\mathcal{L}_G$ corresponding to the eigenvalue 0 as mentioned in Section 3.2.1.1.

Then, $V_1 = \frac{1}{\sqrt{2m}}\begin{pmatrix} \sqrt{d_1} \\ \vdots \\ \sqrt{d_n} \end{pmatrix}$, with $\sum_{i=1}^{n} d_i = 2m$ where m is the number of edges.

Consequently,

$$\begin{aligned}
a'\underline{\mathcal{L}}_G a &\geq \big(\lambda_2(\mathcal{L}_G) - \mu_1\big)\big(1 - a_1^2\big) + 2a_1^2\frac{\sqrt{d_n}}{2m}\left(\sum_{i=1}^{n-1}\sqrt{d_i}\frac{\delta_{in}}{\sqrt{d_id_n}}\right) - \frac{|a_1|}{\mu_1}||w||\sqrt{1 - a_1^2} \\
&\geq \big(\lambda_2(\mathcal{L}_G) - \mu_1\big)\big(1 - a_1^2\big) + \frac{a_1^2}{m}\sum_{i=1}^{n-1}\delta_{in} - \frac{|a_1|}{\mu_1}||w||\sqrt{1 - a_1^2} \\
&\geq \big(\lambda_2(\mathcal{L}_G) - \mu_1\big)\big(1 - a_1^2\big) + \frac{d_n}{m}a_1^2 - \frac{|a_1|}{\mu_1}||w||\sqrt{1 - a_1^2}.
\end{aligned}$$

Although, we have

$$\begin{aligned}
||w||^2 &\leq \left|\left|2\mu_1\left(V_{1n}l_1, \cdots, V_{1n}l_{n-1}, \sum_{i=1}^{n-1} l_iV_{1i}\right)\right|\right|^2 \\
&\leq 4\mu_1^2\left[(V_{1n}l_1)^2 + \cdots + (V_{1n}l_{n-1})^2 + \left(\sum_{i=1}^{n-1} l_iV_{1i}\right)^2\right] \\
&\leq 4\mu_1^2\left[\frac{d_n}{2m}\left(\sum_{i=1}^{n-1}\frac{\delta_{in}}{d_id_n}\right) + \left(\sum_{i=1}^{n-1}\frac{\delta_{in}}{\sqrt{d_id_n}}\frac{\sqrt{d_i}}{\sqrt{2m}}\right)^2\right]
\end{aligned}$$

Then,

$$||w||^2 \leq 4\mu_1^2\left[\frac{1}{2md_{\min}}\sum_{i=1}^{n-1}\delta_{in} + \left(\frac{1}{\sqrt{2m}\sqrt{d_n}}\sum_{i=1}^{n-1}\delta_{in}\right)^2\right]$$
$$\leq 4\mu_1^2\left[\frac{d_n}{2md_{\min}} + \frac{d_n}{2m}\right],$$

and

$$\mu_1 = \sqrt{\sum_{i=1}^{n-1} l_i^2} = \sqrt{\sum_{i=1}^{n-1}\frac{\delta_{in}^2}{d_i d_n}} \leq \frac{1}{\sqrt{d_{\min}}}.$$

This implies that

$$a'\underline{\mathcal{L}}_G a \geq \left(\lambda_2(\mathcal{L}_G) - \frac{1}{\sqrt{d_{\min}}}\right)\left(1-a_1^2\right) + \frac{d_n}{m}a_1^2 - 2a_1\sqrt{\frac{d_n}{2m}\left(1+\frac{1}{d_{\min}}\right)}\sqrt{1-a_1^2}. \tag{3.3}$$

Let us introduce the following notations:

$$A = \lambda_2(\mathcal{L}_G) - \frac{1}{\sqrt{d_{\min}}},$$
$$B_n = 2\sqrt{\frac{d_n}{2m}\left(1+\frac{1}{d_{\min}}\right)},$$
$$f(a_1) = A\left(1-a_1^2\right) + \frac{d_n}{m}a_1^2 - B_n a_1\sqrt{1-a_1^2}. \tag{3.4}$$

Then, the derivative of the function f is given by

$$f'(a_1) = \Big(-2A + 2\frac{d_n}{m}\Big)a_1 - B_n\left(\sqrt{1-a_1^2} - \frac{a_1^2}{\sqrt{1-a_1^2}}\right)$$
$$= C_n a_1 - B_n\left(\frac{1-2a_1^2}{\sqrt{1-a_1^2}}\right).$$

Therefore, $f'(a_1) = 0$ if

$$C_n^2 a_1^2 = B_n^2\frac{(1-2a_1^2)^2}{1-a_1^2}$$
$$\Longrightarrow C_n^2 a_1^2 - C_n^2 a_1^4 = B_n^2 + 4B_n^2 a_1^4 - 4B_n^2 a_1^2 \tag{3.5}$$

Considering that $x = a_1^2$, equation (3.5) yields:

$$(4B_n^2 + C_n^2)x^2 - (4B_n^2 + C_n^2)x + B_n^2 = 0.$$

To find the roots of this equation we need the value of the discriminant

$$\Delta = (4B_n^2 + C_n^2)^2 - 4B_n^2(4B_n^2 + C_n^2)$$
$$= (4B_n^2 + C_n^2)^2\left(1 - \frac{4B_n^2}{4B_n^2 + C_n^2}\right),$$

then,

$$x = \frac{(4B_n^2 + C_n^2) \pm (4B_n^2 + C_n^2)\sqrt{1 - \frac{4B_n^2}{4B_n^2+C_n^2}}}{2(4B_n^2 + C_n^2)}$$
$$= \frac{1}{2} \pm \frac{1}{2}\sqrt{1 - \frac{4B_n^2}{4B_n^2 + C_n^2}} \ \in [0; 1].$$

Moreover, according to the hypothesis we have $f(1) < f(0)$ where

$$f(0) = \lambda_2(\mathcal{L}_G) - \frac{1}{\sqrt{d_{\min}}},$$
$$f(1) = \frac{d_n}{m}.$$

As $f'(x) \underset{x\to 1}{\longrightarrow} +\infty$ and $f(1) < f(0)$, then the minimum of the above function is attained at $x = \frac{1}{2} + \frac{1}{2}\sqrt{1 - \frac{4B_n^2}{4B_n^2+C_n^2}}$.

Since $x = a_1^2$ and $a_1 \geq 0$, then the function $f(a_1)$ has a relative minimum at $a_1 = \sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{1 - \frac{4B_n^2}{4B_n^2+C_n^2}}}$.
Therefore, according to equation (3.3), we get

$$a'\underline{\mathcal{L}}_G a \geq f\left(\sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{1 - \frac{4B_n^2}{4B_n^2 + C_n^2}}}\right).$$

By using the fact that $\frac{B_n}{c_n} \longrightarrow 0$ for all $n > 0$, thus we have $\sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{1 - \frac{4B_n^2}{4B_n^2+C_n^2}}} \longrightarrow 1$.
Consequently, the first eigenvalue of $\underline{\mathcal{L}}_G$ is lower bounded by $\frac{d_{\min}}{m}$. □

To prove the main convergence theorem stated below, we need to recall the bound of resulting mixing time of Theorem 1.5 proposed by [Anari et al.(2016)].
For $\epsilon > 0$,

$$\tau_X(\epsilon) \leq \frac{1}{C_\mu} \cdot \log\Big(\frac{1}{\epsilon \cdot \mu(X)}\Big),$$

with C_μ is the Poincaré constant which is at least $\frac{1}{2kn}$ by construction, $\mu : 2^{[n]} \to \mathbb{R}_+$ and $X \in \text{supp}\{\mu\}$ where $\text{supp}\{\mu\}$ is the state space of the lazy MCMC $\mathcal{M}$ defined in Algorithm 2.
The results of the following theorem provide a fast mixing sampler for our k-DPP, for which polynomial bounds on the mixing time are presented.

Theorem 3.2. *Let $\mathcal{P}_L^k : 2^{[n]} \to \mathbb{R}^+$ where L is the matrix defined by equation (3.2). For any $\epsilon > 0$, if $\lambda_2(\mathcal{L}_G) - \frac{1}{\sqrt{d_{\min}}} > \frac{d_{\max}}{m}$ then the lazy Markov chain defined in Algorithm 2 generates ϵ-approximate sample of $\mathcal{P}_L^k$ in time*

$$\tau_\epsilon \leq 2k^2 n \cdot \log\left[\frac{2n}{I_G \epsilon^{\frac{1}{k}}}\right],$$

where I_G is the lower bound of $\lambda_1(\underline{\mathcal{L}}_G)$.

Proof. The proof is based on the main theorem of [Anari et al.(2016)] by considering μ as a k-DPP noted $\mathcal{P}_L^k$ and the lazy MCMC $\mathcal{M}$ described in Algorithm 2
As a first step and according to Theorem 1.5, we need to lower bound

$$C_{\mathcal{P}_L^k} = \min_{X,X' \subseteq \{1,\cdots,n\},\ |X|=|X'|=k} \max(P_{\mathcal{P}_L^k}(X,X'), P_{\mathcal{P}_L^k}(X',X)).$$

The main idea behind the work of [Anari et al.(2016)] is to propose a new configuration by selecting two elements $x \in X$ where $X \subseteq \{1,\cdots,n\}$, $|X| = k$ and $y \in \{1,\cdots,n\}\backslash X$ in such a way that: x to be removed from the current set, and y to be added. The acceptance of this move is according to the probability defined by the ratio of the proposed determinant to the current determinant. Hence, for $X = Y \cup \{x\}$ the transition probability of removing x and replacing it with y is expressed as

$$P_{\mathcal{P}_L^k}\big(Y \cup \{x\}, Y \cup \{y\}\big) = q\big(Y \cup \{x\}, Y \cup \{y\}\big) \cdot \frac{1}{2} \min\left\{ \frac{\det\left(L_{Y\cup\{y\}}\right)}{\det\left(L_{Y\cup\{x\}}\right)}, 1 \right\}$$

with q is the proposal transition matrix.
Since q is a symmetric transition matrix, thus we have

$$C_{\mathcal{P}_L^k} = \min_{Y,y,x} \max q\big(Y \cup \{x\}, Y \cup \{y\}\big) \left(\frac{1}{2} \min\left\{ \frac{\det\left(L_{Y\cup\{y\}}\right)}{\det\left(L_{Y\cup\{x\}}\right)}, 1 \right\}, \frac{1}{2} \min\left\{ \frac{\det\left(L_{Y\cup\{x\}}\right)}{\det\left(L_{Y\cup\{y\}}\right)}, 1 \right\} \right)$$
$$\geq \frac{1}{2kn}.$$

Then, it remains the calculation of the lower bound of $\mathcal{P}_L^k$ to apply the theorem of [Anari et al.(2016)].
To do so, let us consider $\lambda_2(\mathcal{L}_G) - \frac{1}{\sqrt{d_{\min}}} > \frac{d_{\max}}{m}$ so $\lambda_1(\underline{\mathcal{L}}_G)$ is lower bounded by $I_G > 0$. Therefore, by applying Lemma 3.1, the eigenvalues of L are as follows:

$$0 < \lambda_i(\underline{\mathcal{L}}_G^\dagger) \leq \lambda_i(L) \leq \lambda_{n-1-k+i}(\underline{\mathcal{L}}_G^\dagger). \tag{3.6}$$

By using equation (3.6), the bounds of the determinant of L_X are represented as:

$$\det\left(L_{Y\cup\{x\}}\right) = \det\left(\mathcal{L}_{Y\cup\{x\}}^\dagger\right) \geq \lambda_1(\underline{\mathcal{L}}_G^\dagger) \times \cdots \times \lambda_k(\underline{\mathcal{L}}_G^\dagger) \geq \left(\lambda_1(\underline{\mathcal{L}}_G^\dagger)\right)^k$$

$$\det\left(L_{Y\cup\{x\}}\right) = \det\left(\mathcal{L}_{Y\cup\{x\}}^\dagger\right) \leq \lambda_{n-1-k+1}(\underline{\mathcal{L}}_G^\dagger) \times \cdots \times \lambda_{n-1-k+k}(\underline{\mathcal{L}}_G^\dagger) \leq \left(\lambda_{n-1}(\underline{\mathcal{L}}_G^\dagger)\right)^k. \tag{3.7}$$

Where, the eigenvalues of $\mathcal{L}_G$ are:

$$0 \leq \lambda_1(\mathcal{L}_G) \leq \cdots \leq \lambda_n(\mathcal{L}_G) \leq 2,$$

then by deleting its i-th row and i-th column and by applying Lemma 3.1, the eigenvalues of $\underline{\mathcal{L}}_G$ stand as follows:

$$0 < \lambda_1(\underline{\mathcal{L}}_G) \leq \lambda_2(\mathcal{L}_G) \cdots \leq \lambda_{n-1}(\underline{\mathcal{L}}_G) \leq \lambda_n(\mathcal{L}_G) \leq 2. \tag{3.8}$$

Thus, equation (3.8) allows us to formulate the eigenvalues of $\underline{\mathcal{L}}_G^\dagger$ as follows

$$\frac{1}{2} \leq \frac{1}{\lambda_n(\mathcal{L}_G)} \leq \frac{1}{\lambda_{n-1}(\underline{\mathcal{L}}_G)} = \lambda_1(\underline{\mathcal{L}}_G^\dagger) \leq \cdots \leq \frac{1}{\lambda_1(\underline{\mathcal{L}}_G)} = \lambda_{n-1}(\underline{\mathcal{L}}_G^\dagger).$$

Hence, for any set X of cardinality k, we have

$$\det(L_X) \geq \left(\lambda_1(\underline{\mathcal{L}}_G^\dagger)\right)^k \geq \left(\frac{1}{2}\right)^k.$$

Moreover, according to the definition of the k-DPP (equation (1.1)), we should also calculate the normalization constant. Then, for any set X of cardinality k we want to choose $X' \subseteq V$ of size k maximizing $\det L_X$. Thus, according to Lemma 3.2 we can present the determinant

of $L_{X'}$ as follows:

$$\begin{aligned}\det L_{X'} &\leq \left(\lambda_{n-1}(\underline{\mathcal{L}}_G^{\dagger})\right)^k \\ &\leq \left(\frac{1}{\lambda_1(\underline{\mathcal{L}}_G)}\right)^k \\ &\leq \left(\frac{1}{I_G}\right)^k\end{aligned}$$

Consequently,

$$\mathcal{P}_L^k(X) = \frac{\det(L_X)}{\sum_{|X'|=k}\det(L_{X'})} \geq \frac{\left(\frac{1}{2}\right)^k}{\begin{pmatrix} n \\ k \end{pmatrix}\left(\frac{1}{I_G}\right)^k} \geq \frac{I_G^k}{n^k 2^k} \geq \left(\frac{I_G}{2n}\right)^k.$$

Now, by using Theorem 1.5 we can directly upper bound the mixing time in total variation distance as follows:

$$\begin{aligned}\tau_X(\epsilon) \leq & \frac{1}{C_{\mathcal{P}_L^k}} \cdot \log\left(\frac{1}{\epsilon \cdot \mathcal{P}_L^k}\right) \\ \leq & 2kn \cdot \log\left[\left(\frac{2n}{I_G}\right)^k \frac{1}{\epsilon}\right] \\ \leq & 2k^2 n \cdot \log\left[\frac{2n}{I_G \epsilon^{\frac{1}{k}}}\right].\end{aligned}$$

This proves the result. □

3.4 Conclusion

This chapter presents a method for solving the problem of sampling a diverse subset of nodes in a connected graph. This is reached by designing a MCMC algorithm whose stationary distribution is the k-DPP. The proof used for bounding the convergence speed is based on the main theorem of [Anari et al.(2016)] applied to the special case of k-DPP on graphs where the kernel is the Moore-Penrose pseudo-inverse of the normalized Laplacian matrix. The results provide a fast mixing sampler for k-DPP, for which polynomial bounds on the mixing time are presented if for all $n \geq 0$,

$$\frac{B_n}{C_n} \longrightarrow 0 \text{ and } \lambda_2(\mathcal{L}_G) - \frac{1}{\sqrt{d_{\min}}} > \frac{d_{\max}}{m},$$

with

$$B_n = 2\sqrt{\frac{d_n}{2m}\left(1 + \frac{1}{d_{\min}}\right)} \text{ and } \; C_n = -2\lambda_2(\mathcal{L}_G) + \frac{2}{\sqrt{d_{\min}}} + 2\frac{d_n}{m},$$

where $d_{\min} = \min_i d_i$, $d_{\max} = \max_i d_i$ and m is the number of edges of the graph G.

Intuitively, these conditions are satisfied if the second smallest eigenvalue of the normalized Laplacian $\lambda_2(\mathcal{L}_G)$ is large compared to $\frac{d_{\max}}{m}$. Finally, the resulting mixing time depends on whether the graph bottleneck is large or narrow. The larger the better for the convergence.

Bibliography

[Anari et al.(2016)] N. Anari, S.O. Gharan, and A. Rezaei. Monte Carlo Markov chain algorithms for sampling strongly Rayleigh distributions and determinantal point processes. *In COLT*, pages 103-115, 2016.

[Cheeger (1970)] J. Cheeger. A lower bound for the smallest eigenvalue of the Laplacian. *Problems in Analysis, (R. C. Gunning, ed.) Princeton Univ. Press*, pages 195-199, 1970.

[Cvetkovic et al.(1980)] D.M. Cvetkovic, M. Doob and H.Sachs. Spectra of Graphs. *Academic Press, New York*, 1980.

[Doyle and Snell (1984)] P.G. Doyle and J.L. Snell. Random Walks and Electric Networks. *The Mathematical Association of America* 1984.

[Gobel and Jagers (1974)] F. Gobel and A. Jagers. Random walks on graphs. *Stochastic Processes and their Applications*, volume 2, pages 311-336, 1974.

[Horn and Johnson(1985)] R.A. Horn and C.R. Johnson. Matrix Analysis. *Cambridge University Press*, 1985.

[Langohr(2014)] L. Langohr. Methods for finding interesting nodes in weighted graphs, PhD thesis, University of Helsinki, 2014.

[Li et al.(2008)] H.H. Li, J.S. Li, Y.-Z. Fan. The effect on the second smallest eigenvalue of the normalized Laplacian of a graph by grafting edges. *Linear Multilinear Algebra*, volume 56, pages 627-638, 2008.

[Li et al.(2011)] H.H. Li, J.S. Li. A note on the normalized Laplacian spectra. *Taiwanese J. Math,* volume 15, pages 129-139, 2011.

[Li et al.(2014)] J. Li, J.-M. Guo, W.C. Shiu, A. Chang. An edge-separating theorem on the second smallest normalized Laplacian eigenvalue of a graph and its applications. *Discrete Appl.Math,* volume 171, pages 104-115, 2014.

[Lovász(1993)] L. Lovász. Random walks on graphs: A survey. *Combinatorics, Paul Erdős is Eighty* , volume 2, pages 1-46, 1993.

[Mavroforakis et al.(2015)] C. Mavroforakis , M. Mathioudakis , A. Gionis. Absorbing Random-Walk Centrality: Theory and Algorithms. *Proceedings of the 2015 IEEE International Conference on Data Mining (ICDM)*, volume 14-17, pages 901-906, 2015.

[Merris(1994)] R. Merris. Laplacian Matrices of Graphs: A Survey. *Linear Algebra and its applications* , volume 197-198, pages 143-176, 1994.

[Zejnilović et al.(2013)] S. Zejnilović, J. Gomes, B. Sinopoli. Network observability and localization of the source of diffusion based on a subset of nodes. *51st Annual Allerton Conference on Communication Control, and Computing*, pages 847-852, 2013.

Chapter 4

Connections Between LHS And Fixed-Size Determinantal Point Processes

This chapter aims at using the fixed cardinality determinantal point processes in experimental designs as a tool to study Latin Hypercube Sampling(LHS). A LHS of order n and dimension d is a technique used to generate a sample points of size n from input variables $(x_1, x_2 \cdots, x_d)$ for a simulation study. The basic idea behind this approach is to construct a typical Latin Hypercube design where the points are unlikely to co-occur and the correlation between them is mostly non-positive. The first goal here is to determine the negative correlations between the selected points via a DPP kernel and that is achieved only by respecting the constraints of each design.
The second goal is to study the convergence speed of Markov chains constructed from a 2-dimensional Latin Hypercube sampling and d-dimensional Latin Hypercube sampling which have the n-DPP as their stationary distribution. Therefore, by considering two cases $d = 2$ and $d > 2$, we managed to present the necessary time needed to build a d-dimensional Latin Hypercube sampling with accordance to n-DPP.

4.1 Introduction

Various types of space filling criterion were introduced for designing computer experiments. For example, the Maximin and Minimax ([Johnson et al.(1990)]) were considered as distance criterion because they tend to spread the design points in terms of distance between points, the Minimum Spanning Tree ([Dussert et al.(1986)]), Maximum Entropy Designs ([Schwery and Wynn(1987)]). Another design of space filling is the Latin Hypercube design which is characterized by the fact that all the values of each input are different and this will lead to obtain important information about the output. While different approaches regarding the challenging design problems were proposed, our main interest in this work lies on d-dimensional Latin hypercubes designs.

Latin squares of order n are $n \times n$ matrices whose cells are filled with n objects with the restriction that each object appears only once in each row or column. They were studied by [Euler(1782)] and the term *Latin* refers to the usage of Latin characteristic as objects. A Latin Hypercube sampling of order n and dimension d is an $n \times d$ matrix where the first column lists the elements $\{1, \cdots, n\}$ in the natural order $1, \cdots, n$ and the remaining columns are constructed by permuting the elements of $\{1, \cdots, n\}$ such that each integer appears only once in each column. This means that there cannot be two sample points in the same hyperplane. This method is a type of stratified Monte-Carlo Sampling and may be viewed as a d-dimensional extension of Latin square sampling ([Raj(1968)]).

[McKay et al.(1979)] first considered Latin hypercube sampling as a design for deterministic simulation models. They showed the efficient performance of LHS in estimating the distribution of a monotonic function. Then an extension of this work was expanded by [Iman and Conover(1980)] by comparing the variability of estimates obtained from LHS and random sampling method. There has been a growing interest to develop the LHS strategy for constructing optimal designs. For example, [Ye (1998)] introduced the orthogonal column Latin Hypercubes. Moreover, [Ye et al.(2000)] presented a new design which is an optimal symmetric Latin Hypercubes and they managed to construct a new algorithm according to this design. They showed the effectiveness of this algorithm by comparing it with existing algorithm proposed by [Park(1994)] and [Morris and Mitchelle(1995)]. Further, [Petelet et al.(2010)] suggested a novel technique called constrained Latin hypercube sampling (cLHS), which gives the possibility to construct a LHS such that the samples are constrained by some inequality relations. It is based on the fact that permuting two values of an input variable in a LHS does not break the LHS structure of the sample. [Sheikholeslami and Razavi(2017)] offered a novel sampling strategy which is a sequential version of LHS, called Progressive Latin Hypercube Sampling (PLHS). They proved that PLHS offered better performance comparing to LHS in preserving the distributional properties like space-filling and one-dimensional projection properties. Although that these methods involved new properties which improved the sampling strategy, they are still impractical when it comes to high dimensional input spaces and to large number of sample points.

The aim of this chapter is to display the capability of k-DPP with $k = n$ to generate a Latin Hypercube sampling of order n and dimension d with sample points located as far as possible from each other. The idea is inspired from several works where k-DPP is proposed as a solution to ensure the diversity in the location of sample points. For instance, the k-DPP was stated by [Casquilho et al.(2018)] to obtain spatially balanced designs for environmental monitoring networks where these last have become increasingly important to supervise the environmental processes which affect human health and nature. They used k-DPP to enforce diversity in the sampling locations, thus a good spatial coverage of the region of interest could be obtained. Moreover, [Wang et al.(2018)] presented a sampling method based on k-DPP to solve the combinatorial optimization problem of sub-determinant maximization. They provided a k-DPP approach for finding optimal designs of spatial monitoring network. Finally, [Pratola et al.(2018)] offered a novel approach by sampling from fixed cardinality Determi-

nantal Projection Point Process, called k-DPPP, to construct entropy optimal designs for Gaussian Processes. The benefit that arises from their work is the ability of performing an optimal design in the presence of these two cases, the first one when the dimensionality of the input space is high and the second one is when the number of the points to cover the input space is large. This method requires to compute the k largest eigenvalues and their corresponding eigenvectors which cost at least $\mathcal{O}(kn^2+k^2n)$. Therefore, this technique would still be too time-consuming when applied to large matrices.

Additionally, this chapter displays the performance of fixed-size Determinantal Point Processes in LHS in terms of space filling in higher dimensions. This is achieved through the utilization of MCMC methods, more precisely by constructing Markov chains which have n-DPPs as their stationary distributions. The novelty lies in introducing a positive kernel that preserve the constraint of LHS which is strictly confirmed by the occurrence of each point only once in each hyperplane. Our choice fell on a new kernel which is defined in terms of d distances. The advantage of this special kernel is that it can be easily manipulated to study the convergence speed of the Markov chains constructed from d-dimensional LHS to their stationary distribution n-DPPs. Moreover, the kernel chosen has an influence on selecting the design points that are far from each other. This can be described by considering any two points in the 2-dimensional space: if the first coordinate, without loss of generality, is close to the first coordinate of the other point then this assures that both points must be somehow far according to the second coordinates.
It is shown that generating LHS from n-DPP requires an exponential bound on the mixing time of the Markov chain. The proof is based on canonical paths where their lengths play a major role in improving the lower bound on the Poincaré constant. Hence, the outcome is established by a subset of points that ensure a good coverage of the Latin hypercube design.

Initially, in Section 4.2 an overview of LHS is given. Section 4.3 includes only the case where the cardinality of the input space is equal to two. The work started by choosing a positive kernel that preserves the Latin Hypercube sampling properties and then we illustrate our choice by comparing it with the Gaussian kernel. Moreover, we support our choice by taking two different configurations of 2-dimensional LHS and show that choosing points, which ensure a good coverage are more probable than choosing clusters of points. In Section 4.4 we present the generalization of the kernel chosen to d dimensions and then we enforce our approach to a d-dimensional LHS. Finally, Section 4.5 presents our conclusions.

4.2 Latin Hypercube Sampling (LHS)

To describe the standard approach to LHS of order n and dimension d, we begin by writing the vector of input variables as $(x_1, x_2, \cdots, x_d)$ and assuming for the time being that the variables are mutually independent. Then divide each input range into $M = n$ intervals of equal length, numbered from 1 to n and draw a random value on each interval for each variable. Thus, a Latin Hypercube of order n and dimension d can be represented as a d-dimensional array of n^d cells. However, there are $(n!)^{d-1}$ possible combinations of LHS.

Let us denote a LHS of size n and dimension d by LHS$(x_1, \ldots, x_d)$ where for all $j \in \{1, \cdots .d\}$, $x_j = (x_{1j}, \ldots, x_{nj})$ with $x_{ij} \in \{1, \ldots, n\}$. To construct a LHS, we suppose that the first column $x_1 = (x_{11}, \cdots, x_{n1})$ lists the elements $\{1, \cdots, n\}$ in the natural order $1, \cdots, n$. Then, the remaining columns x_j, $j \in \{2, \cdots, d\}$, are constructed by rearranging the elements of x_1 such that each integer in $\{1, \cdots, n\}$ appears only once in each column j.

For $d = 2$, LHS places this integers $1, \cdots, n$ such that each integer appears only once in each row and each column of the design. For example, in Figure 4.1, a Latin Hypercube of $n = 4$ divisions with $d = 2$ is presented as:

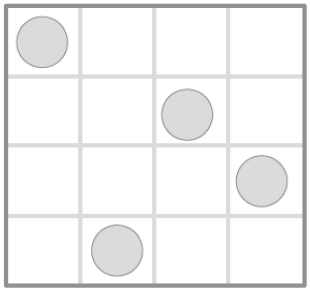

Figure 4.1: A 2-dimensional example of LHS.

4.3 2-dimensional Latin Hypercube sampling

4.3.1 DPP kernel

The overall idea here is to propose a kernel function that define a n-DPP. For all $i, j \in \{1, \cdots, n\}$ with $x_i = (x_{i1}, x_{i2}) \in \{1, \cdots, n\}^2$, the kernel is expressed as follows:

$$L(x_i, x_j) = e^{-d_1(x_i, x_j) \cdot d_2(x_i, x_j)} \tag{4.1}$$

where $d_1(x_i, x_j) = |x_{i1} - x_{j1}|$ and $d_2(x_i, x_j) = |x_{i2} - x_{j2}|$.

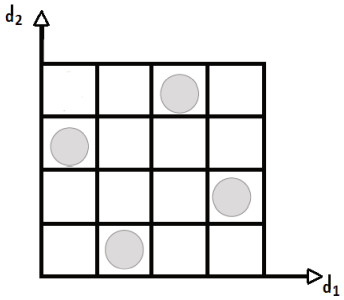

Figure 4.2: A 2-dimensional example of LHS with 4 sample points where the horizontal axes correspond to the distance d_1 and the vertical to the distance d_2.

For example, the kernel that corresponds to the LHS in Figure 4.2, is given as follows:

$$L = \begin{pmatrix} 1 & e^{-2} & e^{-2} & e^{-3} \\ e^{-2} & 1 & e^{-3} & e^{-2} \\ e^{-2} & e^{-3} & 1 & e^{-2} \\ e^{-3} & e^{-2} & e^{-2} & 1 \end{pmatrix}.$$

Because the Gaussian Kernel is used extensively in computer experiments and statistical, hence it is interesting to compare it with the given kernel.
The following results represent the comparison of the determinants of Kernel L defined in equation (4.1) in different configurations of 2-dimensional LHS with the determinant of Gaussian kernel which is represented by the following formula:

$$G(x_i, x_j) = \exp\left\{ - \frac{||x_i - x_j||^2}{2\sigma^2} \right\}, \text{ where } \sigma = 1.$$

However, the figure below illustrates the LHS chosen for $n = 4$ and the next following tables indicates the determinants of the matrices L and G corresponding to these configurations.

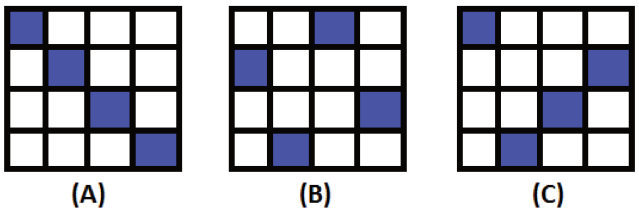

Figure 4.3: Optimal samples with respect to Latin Hypercube properties.

$n = 4$	$\det(L)$	$\det(G)$
(A)	0.6215	0.6416
(B)	0.9289	0.9733
(C)	0.7301	0.9633

Table 4.1: The determinants of the matrices L and G corresponding to the Latin Hypercubes (A), (B) and (C) for $n = 4$.

The Kernel L gives different determinant values in the configurations (B) and (C) and the kernel G gives approximately same values for these two configurations. Therefore, this results shows that the usage of the kernel L is more favorable. Moreover, the advantage of L lies in the fact that the highest determinant value is presented in sample (B) where the points afford a good sample coverage. However, it remains to prove that L is a positive kernel.

Proposition 4.1. *The $n \times n$ matrix L is a positive semi-definite only if L is filled with n points such that each point occurs only once in each row and in each column.*

Proof. To prove that L is a positive semi-definite matrix, two cases are taken into consideration to guarantee the positivity of the kernel for any configuration of 2-dimensional LHS. They are shown in Figure 4.4.

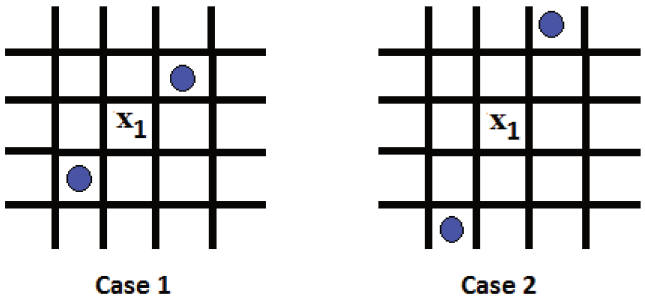

Figure 4.4: Illustrative examples of different configurations of Latin Hypercube where x_1 is in the middle.

Let L_1 be the matrix corresponding to case 1 and L_2 be the matrix corresponding to case 2. By proving that all the eigenvalues of L_1 and L_2 are non-negative, then L is positive semi-definite matrix. Thus, we will use Gershgorin's theorem to bound the spectrum of L. So, every eigenvalue of L satisfies

$$|\lambda(L) - 1| \leq \sum_{x_i \neq x_j} L(x_i, x_j).$$

Therefore, by considering case 1 we have:

$$\begin{aligned}
\lambda_{\min}(L_1) &\geq 1 - \sum_{x_i \neq x_1} L_1(x_i, x_1) \\
&\geq 1 - \sum_{x_i \neq x_1} e^{-d_1(x_i,x_1)\cdot d_2(x_i,x_1)} \\
&\geq 1 - 2e^{-1} - 2\sum_{i=2}^{\infty} e^{-2i} \\
&\geq 1 - 2e^{-1} - 2\frac{e^{-4}}{1 - e^{-2}} \qquad (4.2) \\
&\geq 0.22. \qquad (4.3)
\end{aligned}$$

By applying the same formula for the case 2 we get,

$$\begin{aligned}
\lambda_{\min}(L_2) &\geq 1 - \sum_{x_i \neq x_1} e^{-d_1(x_i,x_1)\cdot d_2(x_i,x_1)} \\
&\geq 1 - 2e^{-2} - 2e^{-2} - 2\sum_{i=3}^{\infty} e^{-i} \\
&\geq 1 - 4e^{-2} - 2\frac{e^{-3}}{1 - e^{-1}} \qquad (4.4) \\
&\geq 0.3.
\end{aligned}$$

Therefore, for any Latin Hypercube sampling all the eigenvalues of the matrix L are lower bounded by 0.22. Hence, L is a positive semi-definite kernel when there is only one point in each row and each column. □

4.3.2 More likely samples chosen

The main idea regarding all the following calculations is to show that choosing points that provide a good coverage of the input space is more probable than choosing clusters of points or diagonal points. This heuristic is based on the fact that each point should be far from others as much as possible, which is precisely what DPP aims at.

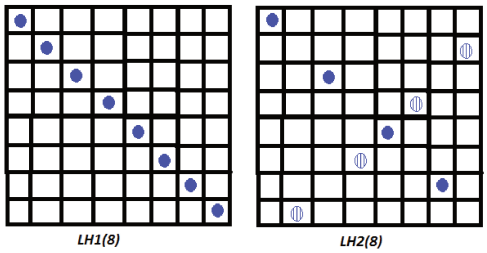

Figure 4.5: Example of two different Latin Hypercube for $n = 8$

In Figure 4.5, $LH1(8)$ contains points that are aligned on the same diagonal whereas in $LH2(8)$ the points are divided into two equal sets: the first set represented by the blue points is forming a diagonal line and the second set represented by the blue hatched points is forming another diagonal line.
Hence, our objective here is to show that $\mathcal{P}_L^8(LH2(8)) > \mathcal{P}_L^8(LH1(8))$.

Proposition 4.2. *Let $\mathcal{P}_L$ be a k-DPP with kernel L defined in equation(4.1). For n divisions we have*

$$\frac{\mathcal{P}_L^n\big(LH1(n)\big)}{\mathcal{P}_L^n\big(LH2(n)\big)} \leq Cq^n,$$

where $C = \frac{(1-e^{-2})(e^2-2-e^{-6}+e^{-8})^4(\vartheta_3(0,e^{-1})+1)^2}{4(1-e^{-4})^2(e^2-2)^4} \simeq 1.72$ *and* $q = \frac{2(e^2-2)}{(\vartheta_3(0,e^{-1})+1)(e^2-2-e^{-6}+e^{-8})} \simeq 0.721\cdot$

Proof. By considering the diagonal case for n divisions we have,

$$L_Y := M_n = \begin{pmatrix} 1 & e^{-1^2} & \dots & e^{-(n-1)^2} \\ e^{-1^2} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & e^{-1^2} \\ e^{-(n-1)^2} & \dots & e^{-1^2} & 1 \end{pmatrix} = \begin{pmatrix} M_{n-1} & v_n \\ v_n^T & 1 \end{pmatrix},$$

where $v_n^T = [e^{-(n-1)^2}, \cdots, e^{-1}]$.
By using the matrix determinant lemma we can formulate the determinant of the Toeplitz matrix M_n as follows

$$\begin{aligned} \det M_n &= \begin{vmatrix} M_{n-1} & v_n \\ v_n^T & 1 \end{vmatrix} \\ &= \det M_{n-1} \times \big(1 - v_n^T M_{n-1}^{-1} v_n\big) \\ &\leq \det M_{n-1} \times \left(1 - \frac{||v_n||^2}{\lambda_{\max}(M_{n-1})}\right) \qquad \text{as } \lambda_{\min}(M_{n-1}) > 0\cdot \end{aligned} \tag{4.5}$$

Now, we use Gershgorin's theorem to bound the spectrum of M_{n-1}. So, every eigenvalue of M_{n-1} satisfies

$$|\lambda(M_{n-1}) - 1| \leq \sum_{x \neq y} (M_{n-1})_{xy}.$$

Therefore,

$$\begin{aligned}\lambda_{\max}(M_{n-1}) &\leq 1 + e^{-1} + e^{-2^2} + \cdots + e^{-(n-2)^2} \\ &\leq \sum_{n=0}^{\infty} e^{-n^2} \\ &\leq \frac{1}{2}\big(\vartheta_3(0, e^{-1}) + 1\big),\end{aligned}$$

where $\vartheta_3(z,q) = \sum_{n=-\infty}^{+\infty} q^{n^2} e^{2niz}$ is a Jacobi theta function.

The same reasoning can also be applied to $||v_n||^2$,

$$\begin{aligned}||v_n||^2 &= (e^{-1})^2 + \cdots + \left(e^{-(n-1)^2}\right)^2 \\ &\leq e^{-1} + \cdots + e^{-(n-1)^2} \\ &\leq \sum_{n=0}^{\infty} e^{-n^2} - 1 \\ &\leq \frac{1}{2}\big(\vartheta_3(0, e^{-1}) + 1\big) - 1.\end{aligned}$$

Thus, according to equation (4.5) we have

$$\det M_n \leq \det M_{n-1} \times \left[1 - \frac{\frac{1}{2}\big(\vartheta_3(0, e^{-1}) + 1\big) - 1}{\frac{1}{2}\big(\vartheta_3(0, e^{-1}) + 1\big)}\right] \leq \det M_{n-1} \times \left[\frac{2}{\vartheta_3(0, e^{-1}) + 1}\right].$$

Since $\det M_2 = 1 - e^{-2}$, we have that

$$\det M_n \leq (1 - e^{-2}) \left[\frac{2}{\vartheta_3(0, e^{-1}) + 1}\right]^{n-2}. \tag{4.6}$$

Regarding $LH2(n)$, the points are divided into two groups: the first $n/2$ points are localized on the left diagonal line while the other $n/2$ hatched points are on the right diagonal line. Therefore, for $n = 8$ the matrix corresponding to $LH2(8)$ is as follows:

$$M_8' = \begin{pmatrix} 1 & e^{-2^2} & e^{-4^2} & e^{-6^2} & e^{-7} & e^{-15} & e^{-15} & e^{-7} \\ e^{-2^2} & 1 & e^{-2^2} & e^{-4^2} & e^{-5} & e^{-3} & e^{-3} & e^{-5} \\ e^{-4^2} & e^{-2^2} & 1 & e^{-2^2} & e^{-9} & e^{-1} & e^{-1} & e^{-9} \\ e^{-6^2} & e^{-4^2} & e^{-2^2} & 1 & e^{-5} & e^{-3} & e^{-3} & e^{-5} \\ e^{-7} & e^{-5} & e^{-9} & e^{-5} & 1 & e^{-2^2} & e^{-4^2} & e^{-6^2} \\ e^{-15} & e^{-3} & e^{-1} & e^{-3} & e^{-2^2} & 1 & e^{-2^2} & e^{-4^2} \\ e^{-15} & e^{-3} & e^{-1} & e^{-3} & e^{-4^2} & e^{-2^2} & 1 & e^{-2^2} \\ e^{-7} & e^{-5} & e^{-9} & e^{-5} & e^{-6^2} & e^{-4^2} & e^{-2^2} & 1 \end{pmatrix} = \begin{pmatrix} A_4 & B_4 \\ C_4 & D_4 \end{pmatrix} = \begin{pmatrix} A_4 & B_4 \\ B_4^T & A_4 \end{pmatrix},$$

Since the points are symmetric in $LH2(8)$ we can remark that some columns of B_4 are identical so its determinant is equal to zero.

Then with respect to the same space-filling designs in $LH2(8)$ we can present a Latin Hypercube for n divisions. The kernel matrix is represented as the following block matrix

form:

$$L_{Y'} := M'_n = \begin{pmatrix} A_{\frac{n}{2}} & B_{\frac{n}{2}} \\ B^T_{\frac{n}{2}} & A_{\frac{n}{2}} \end{pmatrix} = \begin{pmatrix} A_{\frac{n}{2}} & \mathbf{0} \\ \mathbf{0} & A_{\frac{n}{2}} \end{pmatrix} + \begin{pmatrix} \mathbf{0} & B_{\frac{n}{2}} \\ B^T_{\frac{n}{2}} & \mathbf{0} \end{pmatrix}.$$

With this, the determinant of M'_n is expressed as follows

$$\det(M'_n) = \left| \begin{pmatrix} A_{\frac{n}{2}} & \mathbf{0} \\ \mathbf{0} & A_{\frac{n}{2}} \end{pmatrix} + \begin{pmatrix} \mathbf{0} & B_{\frac{n}{2}} \\ B^T_{\frac{n}{2}} & \mathbf{0} \end{pmatrix} \right|.$$

Then, by the Minkowski determinant theorem we have the following inequality:

$$\det(M'_n) \geq \begin{vmatrix} A_{\frac{n}{2}} & \mathbf{0} \\ \mathbf{0} & A_{\frac{n}{2}} \end{vmatrix} + \begin{vmatrix} \mathbf{0} & B_{\frac{n}{2}} \\ B^T_{\frac{n}{2}} & \mathbf{0} \end{vmatrix} \geq \big(\det(A_{\frac{n}{2}})\big)^2 - \big(\det(B_{\frac{n}{2}})\big)^2$$

as $\begin{pmatrix} A_{\frac{n}{2}} & \mathbf{0} \\ \mathbf{0} & A_{\frac{n}{2}} \end{pmatrix}$ and $\begin{pmatrix} \mathbf{0} & B_{\frac{n}{2}} \\ B^T_{\frac{n}{2}} & \mathbf{0} \end{pmatrix}$ are non-negative matrices.

Thus, this allows us to bound the determinant of M'_n as

$$\det(M'_n) \geq \big(\det(A_{\frac{n}{2}})\big)^2. \tag{4.7}$$

Let us start with the determinant of the Toeplitz matrix $A_{\frac{n}{2}}$. The matrix $A_{\frac{n}{2}}$ can be presented as follows:

$$A_{\frac{n}{2}} = \begin{pmatrix} 1 & e^{-2^2} & \dots & e^{-(\frac{n}{2}-1)^2} \\ e^{-2^2} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & e^{-2^2} \\ e^{-(\frac{n}{2}-1)^2} & \dots & e^{-2^2} & 1 \end{pmatrix} = \begin{pmatrix} A_{\frac{n}{2}-1} & u_{\frac{n}{2}} \\ u^T_{\frac{n}{2}} & 1 \end{pmatrix},$$

where $u^T_{\frac{n}{2}} = [e^{-(\frac{n}{2}-2)^2}, \cdots, e^{-2^2}]$.

By using the matrix determinant lemma we can present the determinant of $A_{\frac{n}{2}}$ as follows

$$\begin{aligned} \det A_{\frac{n}{2}} &= \begin{vmatrix} A_{\frac{n}{2}-1} & u_{\frac{n}{2}} \\ u^T_{\frac{n}{2}} & 1 \end{vmatrix} \\ &= \det A_{\frac{n}{2}-1} \times \big(1 - u^T_{\frac{n}{2}} A^{-1}_{\frac{n}{2}-1} u_{\frac{n}{2}}\big) \\ &\geq \det A_{\frac{n}{2}-1} \times \Big(1 - \frac{||u_{\frac{n}{2}}||^2}{\lambda_{\min}(A_{\frac{n}{2}-1})}\Big). \end{aligned} \tag{4.8}$$

Thus by applying Gershgorin's theorem every eigenvalue of $A_{\frac{n}{2}-1}$ satisfies

$$|\lambda(A_{\frac{n}{2}-1}) - 1| \leq \sum_{x \neq y} (A_{\frac{n}{2}-1})_{xy}.$$

Therefore,

$$\begin{aligned} \lambda_{\min}(A_{\frac{n}{2}-1}) &\geq 1 - \Big(e^{-2^2} + e^{-4^2} + \cdots + e^{-(\frac{n}{2}-2)^2}\Big) \\ &\geq 1 - \Big(e^{-2} + e^{-4} + \cdots + e^{-(\frac{n}{2}-2)}\Big) \\ &\geq 1 - \sum_{i=1}^{n/4-1} e^{-2i} \end{aligned}$$

Thus,

$$\begin{aligned}\lambda_{\min}(A_{\frac{n}{2}-1}) &\geq 1-\sum_{i=1}^{n/4-1}\left(e^{-2}\right)^i\\ &\geq 1-\Big(\sum_{i=0}^{\infty}\left(e^{-2}\right)^i-1\Big)\\ &\geq 1-\Big(\frac{1}{1-e^{-2}}-1\Big) \text{ because } \sum_{i=0}^{\infty}\left(e^{-2}\right)^i \text{ is a geometric series}\\ &\geq \frac{e^2-2}{e^2-1}.\end{aligned}$$

Moreover,

$$||u_{\frac{n}{2}}||^2=\left(e^{-2^2}\right)^2+\cdots+\left(e^{(-\frac{n}{2}-2)^2}\right)^2\geq\left(e^{-4}\right)^2\geq e^{-8}.$$

Then, according to equation (4.8) we have

$$\det A_{\frac{n}{2}}\geq\det A_{\frac{n}{2}-1}\times\left[1-\frac{e^{-8}(e^2-1)}{e^2-2}\right].$$

Since $\det A_2=1-e^{-4}$, we have that

$$\det A_{\frac{n}{2}}\geq(1-e^{-4})\left[1-\frac{e^{-8}(e^2-1)}{e^2-2}\right]^{\frac{n}{2}-2}.$$

Thus equation (4.7) yields:

$$\det(M_n')\geq(1-e^{-4})^2\left[1-\frac{e^{-8}(e^2-1)}{e^2-2}\right]^{n-4}.$$

Consequently, the ratio of the determinants of two matrices is upper bounded by

$$\begin{aligned}\frac{\mathcal{P}_L^n\big(LH1(n)\big)}{\mathcal{P}_L^n\big(LH2(n)\big)}&\leq\frac{\det(M_n)}{\det(M_n')}\\ &\leq\frac{(1-e^{-2})\left[\frac{2}{\vartheta_3(0,e^{-1})+1}\right]^{n-2}}{(1-e^{-4})^2\left[1-\frac{e^{-8}(e^2-1)}{e^2-2}\right]^{n-4}}\\ &\leq\frac{(1-e^{-2})(\vartheta_3(0,e^{-1})+1)^2(e^2-2-e^{-6}+e^{-8})^4}{4(1-e^{-4})^2(e^2-2)^4}\left[\frac{2(e^2-2)}{(\vartheta_3(0,e^{-1})+1)(e^2-2-e^{-6}+e^{-8})}\right]^n.\end{aligned}$$

□

4.3.3 Construct a Markov chain from a 2-dimensional LHS

For a natural number n and $d=2$, let S_n denote the set of permutations of the set $[n]=\{1,\cdots,n\}$. Each permutation $x\in S_n$ leads to a different LHS, denoted $(x_1,\cdots,x_n)$. We define a Markov chain $\mathcal{M}_2$ with state space S_n such that at each step we stay at state x with probability $1/2$ and we propose to move to a new state according to the proposal distribution

q with probability 1/2. If the current state is x, a new proposal state is drawn as follows: we choose an element $i \in \{1, \cdots, n\}$ uniformly at random and moved x_i to the top or inversely we take the top element and insert it at one of the n positions in the set. Both movements are selected with an equal probability.
We construct the Metropolis chain with stationary distribution $\pi = \mathcal{P}_L^n$ as follows:

- at a state x we choose a new LHS y with probability $P(x, y)$ and we propose to move to y.
- we accept this proposal with probability $\min\left\{1, \frac{\mathcal{P}_L^n(y)q(y,x)}{\mathcal{P}_L^n(x)q(x,y)}\right\}$ and we reject and stay at x with the remaining probability.

By taking into consideration that q is a symmetric proposal distribution, the transition probability matrix P of this Markov chain is given as:

$$P(x,y) = \begin{cases} q(x,y) \cdot \frac{1}{2} \min\left\{1, \frac{\mathcal{P}_L^n(y)}{\mathcal{P}_L^n(x)}\right\} & \text{if random-to-top or top-to-random} \\ 1 - \sum\limits_{z \neq x} q(x,z) \cdot \frac{1}{2} \min\left\{1, \frac{\mathcal{P}_L^n(z)}{\mathcal{P}_L^n(x)}\right\} & \text{otherwise.} \end{cases}$$

This implies that all states communicate with each other. As $P(x, x) \geq \frac{1}{2}$ for all $x \subseteq [n]$, then Markov chain described above is said to be a lazy chain.
The Metropolis chain $\mathcal{M}_2$ clearly describes an irreducible aperiodic Markov chain, so it converges to its stationary distribution $\mathcal{P}_L^n$.

4.3.4 Rapid mixing of $\mathcal{M}_2$ via canonical path

The following lemma studies the convergence speed of the distribution of the Markov chain $\mathcal{M}_2$ to the stationary distribution $\mathcal{P}_L^n$, for which a bound on the mixing time is presented.

Lemma 4.1. *The Markov chain $\mathcal{M}_2$ with the state space S_n and stationary distribution $\mathcal{P}_L^n$ has a mixing time*

$$\tau_\epsilon \leq 4n^3(\log n + 1)\left[\frac{177}{22}\right]^{4n} \cdot \log\left[\frac{177}{22}\left(\frac{n!}{\epsilon}\right)^{\frac{1}{n}}\right].$$

Proof. According to [Sinclair(1992)], the proof is based on Poincaré inequality and canonical path method. Thus, we need to lower bound

$$C = \max_e \left\{ \frac{1}{Q(e)} \sum_{x,y:e \in \delta_{xy}} \mathcal{P}_L^n(x)\mathcal{P}_L^n(y)|\delta_{xy}| \right\}.$$

First of all, we will lower bound $\mathcal{P}_L^n(x)$ according to case 1 in Figure 4.4. By considering this case for n divisions and by using equation (4.3), the lower bound of $\lambda_{\min}(L)$ is given as

$$\lambda_{\min}(L) \geq 0.22$$

and the upper bound of $\lambda_{\max}(L)$ can be represented as

$$\begin{aligned}\lambda_{\max}(L_1) &\leq 1 + \sum_{x_i \neq x_1} L_1(x_i, x_1) \\ &\leq 1 + \sum_{x_i \neq x_1} e^{-d_1(x_i,x_1)\cdot d_2(x_i,x_1)} \\ &\leq 1 + 2e^{-1} + 2\sum_{i=2}^{\infty} e^{-2i} \\ &\leq 1 + 2e^{-1} + 2\frac{e^{-4}}{1-e^{-2}} \\ &\leq 1.77.\end{aligned}$$

Thus, all the eigenvalues of the kernel L are upper bounded by 1.77. Consequently,

$$\begin{aligned}\mathcal{P}_L^n(x) &= \frac{\det L_x}{\sum_{|x'|=n} \det L_{x'}} \\ &\leq \frac{1}{n!}\left[\frac{\lambda_{\max}(L)}{\lambda_{\min}(L)}\right]^n. \end{aligned} \tag{4.9}$$

Now, we need to establish paths δ_{xy} between two permutations x and y using edges $e = (x, y)$. A canonical path between permutations $x, y \in S_n$ where $x = (x_1, \cdots, x_n)$ and $y = (y_1, \cdots, y_n)$ can be described as follows: when an element in x is elevated and moved to the top, it will be represented as y_n in y vector. Then, the second element chosen from x and moved to the top will be y_{n-1} and so on. However, the last element chosen from x will be represented as y_1 in y. Let us create a schema to illustrate this idea. For $i \in \{1, \cdots, n\}$, the path between x and y can be expressed as

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \to \begin{pmatrix} y_n \\ x^*_{1,1} \\ x^*_{1,2} \\ \vdots \\ x^*_{1,n-1} \end{pmatrix} \to \begin{pmatrix} y_{n-1} \\ y_n \\ x^*_{2,1} \\ \vdots \\ x^*_{2,n-2} \end{pmatrix} \to \cdots \to \begin{pmatrix} y_{n-i+1} \\ \vdots \\ y_n \\ x^*_{i,1} \\ \vdots \\ x^*_{i,n-i} \end{pmatrix} \to \cdots \to \begin{pmatrix} y_2 \\ y_3 \\ \vdots \\ y_n \\ x^*_{n-1,1} \end{pmatrix} \to \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

where $x^*_{l,1}, \cdots, x^*_{l,j}$ with $l, j \in \{1, \cdots, n-1\}$ are the remaining elements in x which has not been chosen and elevated to the top yet.

For successive values $p = 0, \cdots, n-1$, we choose an element of x uniformly at random and move it to the top, then δ_{xy} is the reunion of $\delta_0, \cdots, \delta_{n-1}$:

- Path δ_0: we choose an element of x and move it to the top. Then by fixing this possibility it remains $(n-1)!$ permutations for the next path.

  $\vdots$

- Path δ_p: p elements are chosen and moved to the top. Then by fixing these $p!$ possibilities it remains $(n-(p+1))!$ permutations for the next path.

Thus, there are $\sum_{p=0}^{n-1} \begin{pmatrix} n \\ p \end{pmatrix} (n-p-1)!$ path δ_{xy} uses the edge e. Therefore, the number of

paths that use e is

$$\begin{aligned}
\#\{x,y : e \in \delta_{xy}\} &= \sum_{p=0}^{n-1} \left(\begin{array}{c} n \\ p \end{array} \right) (n-p-1)! \\
&= n! \sum_{p=0}^{n-1} \frac{(n-p-1)!}{p!(n-p)!} \\
&= n! \sum_{p=0}^{n-1} \frac{1}{p!(n-p)} \\
&\leq n! \sum_{p=0}^{n-1} \frac{1}{(n-p)} \\
&\leq n! \sum_{h=1}^{n} \frac{1}{h} \\
&\leq n!(\log n + 1).
\end{aligned}$$

Moreover, the maximal length of a path δ_{xy} is clearly no more than n.
Thereafter, it remains the calculation of the lower bound of $Q(e) = \mathcal{P}_L^n(x) \cdot P(x,y)$. By using the upper and lower bounds of $\det(L)$, the lower bound of $Q(e)$ can be represented as follows:

$$\begin{aligned}
\mathcal{P}_L^n(x) \times P(x,y) &\geq \frac{1}{n!} \left[\frac{\lambda_{\min}(L)}{\lambda_{\max}(L)} \right]^n \times \frac{1}{2} \min \left\{ \frac{\mathcal{P}_L^n(y)}{\mathcal{P}_L^n(x)}, 1 \right\} \cdot q(x,y) \\
&\geq \frac{1}{2n!} \left[\frac{\lambda_{\min}(L)}{\lambda_{\max}(L)} \right]^n \times \min \left\{ \frac{\mathcal{P}_L^n(y)}{\mathcal{P}_L^n(x)}, 1 \right\} \cdot \frac{1}{2n} \\
&\geq \frac{1}{4nn!} \left[\frac{\lambda_{\min}(L)}{\lambda_{\max}(L)} \right]^n \times \min \left\{ \left(\frac{\lambda_{\min}(L)}{\lambda_{\max}(L)} \right)^n, 1 \right\} \\
&\geq \frac{1}{4nn!} \left[\frac{\lambda_{\min}(L)}{\lambda_{\max}(L)} \right]^n \times \left[\frac{\lambda_{\min}(L)}{\lambda_{\max}(L)} \right]^n \\
&\geq \frac{1}{4nn!} \left[\frac{\lambda_{\min}(L)}{\lambda_{\max}(L)} \right]^{2n}. \qquad (4.10)
\end{aligned}$$

Hence, the lower bound of C is obtained by using equations (4.9) and (4.10):

$$\begin{aligned}
C &= \max_e \left\{ \frac{1}{Q(e)} \sum_{x,y:e\in\delta_{xy}} \mathcal{P}_L^n(x) \mathcal{P}_L^n(y) |\delta_{xy}| \right\} \\
&\leq 4nn! \left[\frac{\lambda_{\max}(L)}{\lambda_{\min}(L)} \right]^{2n} \cdot \frac{1}{n!} \frac{1}{n!} \left[\frac{\lambda_{\max}(L)}{\lambda_{\min}(L)} \right]^{2n} \cdot n \cdot n! (\log n + 1) \\
&\leq 4n^2 (\log n + 1) \left[\frac{\lambda_{\max}(L)}{\lambda_{\min}(L)} \right]^{4n} \\
&\leq 4n^2 (\log n + 1) \left[\frac{177}{22} \right]^{4n}.
\end{aligned}$$

Consequently, according to Theorem 1.7 in Chapter 1 the mixing time of the Markov chain

$\mathcal{M}_2$ is represented as

$$\begin{aligned}\tau_\epsilon &\leq C \cdot \log\left(\frac{1}{\epsilon \cdot \mathcal{P}_L^n(x)}\right)\\ &\leq 4n^2(\log n + 1)\Big[\frac{177}{22}\Big]^{4n} \cdot \log\left[\frac{n!}{\epsilon}\left(\frac{\lambda_{\max}(L)}{\lambda_{\min}(L)}\right)^n\right]\\ &\leq 4n^3(\log n + 1)\Big[\frac{177}{22}\Big]^{4n} \cdot \log\left[\frac{177}{22}\Big(\frac{n!}{\epsilon}\Big)^{\frac{1}{n}}\right].\end{aligned}$$

□

4.4 d-dimensional Latin Hypercube Sampling

Let $d > 2$ be the dimension of the input space and n the number of sample points. According to Section 4.2, a LHS of size n and dimension d is denoted by LHS$(x_1, \ldots, x_d)$ where for all $j \in \{1, \cdots, d\}$, $x_j = (x_{1j}, \ldots, x_{nj})$ with $x_{ij} \in \{1, \ldots, n\}$.
Hence by assuming that the first column x_1 lists the elements $\{1, \cdots, n\}$ in the natural order $1, \cdots, n$, a $n \times d$ matrix is obtained such as each of the $d-1$ columns, x_j, $j \in \{2, \cdots, d\}$ is obtained by permuting $\{1, \cdots, n\}$ meaning that there cannot be two points in the same hyperplane.

In the figure bellow, an examples of 3-dimensional LHS of size 10 is given where each input range is divided into 10 intervals of equal length, numbered from 1 to 10 and draw a random sample on each interval for each variable.

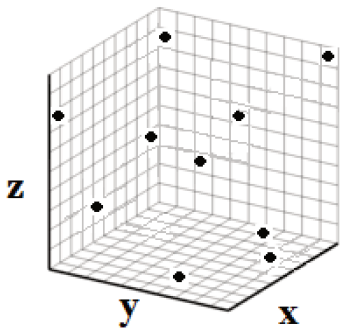

Figure 4.6: Optimal design of a 3-dimensional LHS.

	d_1	d_2	d_3
x_1	1	3	8
x_2	2	6	7
x_3	3	5	2
x_4	4	1	3
x_5	5	2	9
x_6	6	4	1
x_7	7	8	6
x_8	8	7	10
x_9	9	10	4
x_{10}	10	9	5

Table 4.2: A 3-dimensional LHS with $n = 10$.

4.4.1 Generalisation of the DPP Kernel

For all $i \in \{1, \cdots, n\}$, x_i is described as a set of coordinates $x_{i1}, x_{i2}, \cdots, x_{id} \in \{1, \cdots, n\}$, we define a kernel in terms of d distances as follows:

$$K(x_i, x_j) = \frac{\sum_{l<m} e^{-d_l(x_i,x_j)d_m(x_i,x_j)}}{d(d-1)/2} \qquad \forall i, j \in \{1, \cdots, n\},\ \forall l, m \in \{1, \cdots, d\} \qquad (4.11)$$

where $d_l(x_i, x_j) = |x_{il} - x_{jl}|$ and $d_m(x_i, x_j) = |x_{im} - x_{jm}|$.

As it is demonstrated in Proposition 4.1, L is a positive kernel for $d = 2$, the aim of the following proposition is to prove that the kernel K defined in equation (4.11) is also a positive kernel for $d > 2$.

Proposition 4.3. *The $n \times n$ matrix K is a positive semi-definite matrix only if K is filled with n points such that each point occurs only once in each hyperplane.*

Proof. Projecting the set of n points from a d-dimensional space into a 2-dimensional space while considering the fact that each point occurs only once in each hyperplane will be the right tool to be used to prove that K is a positive semi-definite matrix. This projection allows the occurrence of each point once in each row and each column since the constraint is not to have two sample points on the same hyperplane. Thus, the proof of positivity of K is based on the results of Proposition 4.1 where two cases were taken into consideration to guarantee the positivity of the kernel L for any configuration of 2-dimensional LHS.
By following the same reasoning of Proposition 4.1, proving that K is positive semi-definite matrix will required non-negative eigenvalues. According to Gershgorin's theorem, every eigenvalue of K satisfies

$$|\lambda(K) - 1| \leq \sum_{x_i \neq x_j} K(x_i, x_j)$$

for all $i, j \in \{1, \cdots, n\}$.
Therefore, by projecting the set of n points in d-dimensional space into a 2-dimensional space

while preserving the constraints leads to:

$$\begin{aligned}\lambda_{\min}(K) &\geq 1 - \sum_{x_i \neq x_1} K(x_i, x_1) \\ &\geq 1 - \sum_{x_i \neq x_1} \frac{\sum\limits_{l<m} e^{-d_l(x_i,x_1)\cdot d_m(x_i,x_1)}}{d(d-1)/2} \\ &\geq 1 - \frac{2}{d(d-1)} \sum_{l<m} \sum_{x_i \neq x_1} e^{-d_l(x_i,x_1)\cdot d_m(x_i,x_1)}.\end{aligned}$$

According to equations (4.2) and (4.4), we have

$$\begin{aligned}\lambda_{\min}(K) &\geq 1 - \frac{2}{d(d-1)} \sum_{l<m} \max\Big\{2e^{-1} + 2\frac{e^{-4}}{1-e^{-2}},\ 4e^{-2} + 2\frac{e^{-3}}{1-e^{-1}}\Big\} \\ &\geq 1 - \frac{2}{d(d-1)} \frac{d(d-1)}{2} \max\Big\{2e^{-1} + 2\frac{e^{-4}}{1-e^{-2}},\ 4e^{-2} + 2\frac{e^{-3}}{1-e^{-1}}\Big\} \\ &\geq 0.22. \end{aligned} \tag{4.12}$$

Therefore, for $d > 2$ all the eigenvalues of the matrix K are lower bounded by 0.22. Hence, K is a positive semi-definite kernel when there is only one point in each hyperplane. □

According to the results of Proposition 4.3, K is positive semi-definite for any d-dimensional LHS since each point in LHS occurs only once in each axis-aligned hyperplane containing it.

4.4.2 Construct a Markov chain from a d-dimensional LHS

The main idea behind this section is to introduce a special Markov chain constructed from d-dimensional LHS which has n-DPP as its stationary distribution. The challenge here is to study the convergence of this chain to the n-DPP with kernel K. As k-DPPs designate higher probability to points that are negatively correlated then a LHD with more spread out points is likely to be selected.

For a natural number n and $d > 2$, let S_n denote the set of permutations of the set $[n] = \{1, \cdots, n\}$. Each $d-1$ permutations leads to a different LHS. We define a Markov chain $\mathcal{M}_d$ with state space S_n such that at each step we stay at state x with probability $1/2$ and we propose to move to a new state according to the proposal distribution q with probability $1/2$. Following the same reasoning as for $d = 2$, to propose a new state two movements with an equal probability are selected. If the current state is x, a new proposal state is drawn as follows: firstly we choose an element $j \in \{2, \cdots, d\}$ uniformly at random, secondly we choose an element $i \in \{2, \cdots, n\}$ uniformly at random and we move x_{ij} to the top of the j^{th} column or inversely we take the top element of x_j and insert it at one of the $n-1$ positions in the set $\{2, \cdots, n\}$.
For example, by taking a 3-dimensional LHS with $n = 5$ and by choosing $j = 2$ and $i = 4$, the transition of moving x_{42} to the top is given as

$$\begin{bmatrix} 1 & 5 & 4 \\ 2 & 4 & 1 \\ 3 & 1 & 5 \\ 4 & 2 & 3 \\ 5 & 3 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & 4 \\ 2 & 5 & 1 \\ 3 & 4 & 5 \\ 4 & 1 & 3 \\ 5 & 3 & 2 \end{bmatrix}$$

We construct the Metropolis chain with stationary distribution $\pi = \mathcal{P}_K^n$ as follows:

- at a state $x = \text{LHS}(x_1, \cdots, x_d)$ we choose a new state $y = \text{LHS}(y_1, \cdots, y_d)$ with probability $P(x, y)$ and we propose to move to y.
- we accept this proposal with probability $\min\left\{1, \frac{\mathcal{P}_K^n(y)q(y,x)}{\mathcal{P}_K^n(x)q(x,y)}\right\}$ where q is the proposal distribution and we reject and stay at x with the remaining probability.

By using the fact that the proposal distribution is symmetric, then the transition probability matrix P of $\mathcal{M}_d$ is given as:

$$P(x,y) = \begin{cases} q(x,y) \cdot \frac{1}{2} \min\left\{1, \frac{\mathcal{P}_K^n(y)}{\mathcal{P}_K^n(x)}\right\} & \text{if random-to-top or top-to-random} \\ 1 - \sum\limits_{z \neq x} q(x,z) \cdot \frac{1}{2} \min\left\{1, \frac{\mathcal{P}_K^n(z)}{\mathcal{P}_K^n(x)}\right\} & \text{otherwise.} \end{cases}$$

As $P(x,x) \geq \frac{1}{2}$ for all LHS x, then Markov chain described above is said to be a lazy chain.

4.4.3 Rapid mixing of $\mathcal{M}_d$ via canonical path

The following theorem study the convergence speed of the distribution of the Markov chain $\mathcal{M}_d$ to the stationary distribution $\mathcal{P}_K^n$, for which a bound on the mixing time is offered.

Theorem 4.1. *The Markov chain $\mathcal{M}_d$ with the state space S_n and stationary distribution $\mathcal{P}_K^n$ has a mixing time*

$$\tau_\epsilon \leq 4(d-1)^3 n^3 (\log n + 1)\left[\frac{177}{22}\right]^{4n} \cdot \log\left[\left[\frac{177}{22}\right]^{\frac{1}{d-1}} \left(\frac{n!}{\epsilon^{\frac{1}{d-1}}}\right)^{\frac{1}{n}}\right].$$

Proof. Based on the reasoning of the proof of Lemma 4.1, we need to lower bound

$$C = \max_e \left\{ \frac{1}{Q(e)} \sum_{x,y: e \in \delta_{xy}} \mathcal{P}_K^n(x)\mathcal{P}_K^n(y)|\delta_{xy}| \right\}.$$

Firstly, let us calculate the lower bound of $\mathcal{P}_K^n(x)$ where $x = \text{LHS}(x_1, \cdots, x_d)$. According to equation (4.12) we have,

$$\lambda_{\min}(K) \geq 0.22$$

and according to Gershgorin's theorem we have

$$\lambda_{\max}(K) \leq 1 + \sum_{x_i \neq x_1} K(x_i, x_1).$$

The same reasoning that allowed us to get the bound of $\lambda_{\min}(K)$ in Proposition 4.3 will be applied to get the bound of $\lambda_{\max}(K)$, thus

$$\begin{aligned}\lambda_{\max}(K) &\leq 1+\sum_{x_i\neq x_1} K(x_i,x_1)\\ &\leq 1+\sum_{x_i\neq x_1}\frac{\sum\limits_{l<m} e^{-d_l(x_i,x_1)\cdot d_m(x_i,x_1)}}{d(d-1)/2}\\ &\leq 1+\frac{2}{d(d-1)}\sum_{l<m}\sum_{x_i\neq x_1} e^{-d_l(x_i,x_1)\cdot d_m(x_i,x_1)}.\end{aligned}$$

According to equations (4.2) and (4.4), we have

$$\begin{aligned}\lambda_{\max}(K) &\leq 1+\frac{2}{d(d-1)}\sum_{l<m}\max\Big\{2e^{-1}+2\frac{e^{-4}}{1-e^{-2}},\ 4e^{-2}+2\frac{e^{-3}}{1-e^{-1}}\Big\}\\ &\leq 1+\frac{2}{d(d-1)}\frac{d(d-1)}{2}\max\Big\{2e^{-1}+2\frac{e^{-4}}{1-e^{-2}},\ 4e^{-2}+2\frac{e^{-3}}{1-e^{-1}}\Big\}\\ &\leq 1.77.\end{aligned}$$

Then, all the eigenvalues of the kernel K are upper bounded by 1.77 and lower bounded by 0.22. Thus,

$$\begin{aligned}\mathcal{P}_K^n(x) &= \frac{\det K_x}{\sum_{|x'|=n}\det K_{x'}}\\ &\leq \frac{1}{(n!)^{d-1}}\left[\frac{\lambda_{\max}(K)}{\lambda_{\min}(K)}\right]^n.\end{aligned}\tag{4.13}$$

Secondly, we need to found paths δ_{xy} between $x=\mathrm{LHS}(x_1,\cdots,x_d)$ and $y=\mathrm{LHS}(y_1,\cdots,y_d)$. Since the natural order $1,\cdots,n$ is usually assumed for the first column, a canonical path between x and y can be described as follows: firstly, we select a column by choosing an element $j\in\{2,\cdots,d\}$. Secondly, an element in the column x_j is elevated and moved to the top, it will be represented as y_{nj} in LHS y. Then, the second element chosen from x_j and moved to the top will be y_{n-1j} and so on. However, the last element chosen from x_j will be represented as y_{1j} in LHS y. These steps will be repeated until the $d-1$ columns are chosen. Let us create a diagram to illustrate this idea. By choosing an element $j\in\{2,\cdots,d\}$, the first steps in the pathway between x and y can be expressed as

$$\begin{bmatrix} x_{11} & & x_{1j} & & x_{1d}\\ x_{21} & & x_{2j} & & x_{2d}\\ \vdots & \cdots & \vdots & \cdots & \vdots\\ x_{n1} & & x_{nj} & & x_{nd}\end{bmatrix}\rightarrow\begin{bmatrix} x_{11} & & y_{nj} & & x_{1d}\\ x_{21} & & x^*_{1j} & & x_{2d}\\ \vdots & \cdots & \vdots & \cdots & \vdots\\ x_{n1} & & x^*_{n-1j} & & x_{nd}\end{bmatrix}\rightarrow$$

$$\begin{bmatrix} x_{11} & & y_{n-1j} & & x_{1d}\\ x_{21} & & y_{nj} & & x_{2d}\\ x_{31} & & x^*_{1j} & & x_{3d}\\ \vdots & \cdots & \vdots & \cdots & \vdots\\ x_{n1} & & x^*_{n-2j} & & x_{nd}\end{bmatrix}\rightarrow\cdots\rightarrow\begin{bmatrix} x_{11} & & y_{n-i+1j} & & x_{1d}\\ \vdots & \cdots & \vdots & \cdots & \vdots\\ x_{i1} & & y_{nj} & & x_{id}\\ x_{i+11} & & x^*_{1j} & & x_{i+1d}\\ \vdots & \cdots & \vdots & \cdots & \vdots\\ x_{n1} & & x^*_{n-ij} & & x_{nd}\end{bmatrix}\rightarrow$$

$$\begin{bmatrix} x_{11} & & y_{2j} & & x_{1d} \\ x_{21} & & y_{3j} & & x_{2d} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ x_{n-11} & & y_{nj} & & x_{n-1d} \\ x_{n1} & & x^*_{1j} & & x_{nd} \end{bmatrix} \rightarrow \begin{bmatrix} x_{11} & & y_{1j} & & x_{1d} \\ x_{21} & & y_{2j} & & x_{2d} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ x_{n-11} & & y_{n-1j} & & x_{n-1d} \\ x_{n1} & & y_{nj} & & x_{nd} \end{bmatrix}$$

where $x^*_{1j}, \cdots, x^*_{ij}$ with $i \in \{1, \cdots, n-1\}$ are the remaining elements in x_j which has not been chosen and elevated to the top yet.
Thus, a canonical path between x and y is created by completing all the steps described above for all x_j, $j \in \{2, \cdots, d\}$.

For a given edge $e \in \delta_{xy}$, only one column, let us say j, is modified and among it already p elements of y_j have already been selected. Thus, there are $\sum_{p=0}^{n-1} \begin{pmatrix} n \\ p \end{pmatrix} (n-p-1)! \cdot (n!)^{d-2}$ path δ_{xy} uses the edge e.
Hence, the number of paths that use e is given as

$$\begin{aligned} \#\{x, y : e \in \delta_{xy}\} &= \sum_{p=0}^{n-1} \begin{pmatrix} n \\ p \end{pmatrix} (n-p-1)! \cdot (n!)^{d-2} \\ &= n!(n!)^{d-2} \sum_{p=0}^{n-1} \frac{(n-p-1)!}{p!(n-p)!} \\ &= (n!)^{d-1} \sum_{p=0}^{n-1} \frac{1}{p!(n-p)} \\ &\leq (n!)^{d-1} \sum_{p=0}^{n-1} \frac{1}{(n-p)} \\ &\leq (n!)^{d-1} (\log n + 1). \end{aligned}$$

Therefore, the maximal length of a path δ_{xy} is clearly no more than $(d-1)n$.

Lastly, it remains the calculation of the lower bound of $Q(e) = \mathcal{P}^n_K(x) \cdot P(x, y)$. By using the fact that $\det(K)$ is upper bounded by $(\lambda_{\max}(K))^n$ and lower bounded by $(\lambda_{\min}(K))^n$, the lower bound of $Q(e)$ can be expressed as follows:

$$\begin{aligned} \mathcal{P}^n_K(x) \times P(x, y) &\geq \frac{1}{(n!)^{d-1}} \left[\frac{\lambda_{\min}(K)}{\lambda_{\max}(K)}\right]^n \times \frac{1}{2} \min\left\{\frac{\mathcal{P}^n_K(y)}{\mathcal{P}^n_K(x)}, 1\right\} \cdot q(x, y) \\ &\geq \frac{1}{2(n!)^{d-1}} \left[\frac{\lambda_{\min}(K)}{\lambda_{\max}(K)}\right]^n \times \min\left\{\frac{\mathcal{P}^n_K(y)}{\mathcal{P}^n_K(x)}, 1\right\} \cdot \frac{1}{2(d-1)n} \\ &\geq \frac{1}{4(d-1)n(n!)^{d-1}} \left[\frac{\lambda_{\min}(K)}{\lambda_{\max}(K)}\right]^n \times \min\left\{\left(\frac{\lambda_{\min}(K)}{\lambda_{\max}(K)}\right)^n, 1\right\} \\ &\geq \frac{1}{4(d-1)n(n!)^{d-1}} \left[\frac{\lambda_{\min}(K)}{\lambda_{\max}(K)}\right]^n \times \left[\frac{\lambda_{\min}(K)}{\lambda_{\max}(K)}\right]^n \end{aligned}$$

Then,

$$\mathcal{P}_K^n(x) \times P(x,y) \geq \frac{1}{4(d-1)n(n!)^{d-1}} \Big[\frac{\lambda_{\min}(K)}{\lambda_{\max}(K)}\Big]^{2n}. \tag{4.14}$$

Hence, by using equations (4.13) and (4.14) we are able to present the lower bound of C as follows

$$\begin{aligned}
C &= \max_e \left\{ \frac{1}{Q(e)} \sum_{x,y:e\in\delta_{xy}} \mathcal{P}_K^n(x)\mathcal{P}_K^n(y)|\delta_{xy}| \right\} \\
&\leq 4(d-1)n(n!)^{d-1} \left[\frac{\lambda_{\max}(K)}{\lambda_{\min}(K)}\right]^{2n} \cdot \frac{1}{(n!)^{d-1}} \frac{1}{(n!)^{d-1}} \left[\frac{\lambda_{\max}(K)}{\lambda_{\min}(K)}\right]^{2n} \cdot (d-1)n \cdot (n!)^{d-1}(\log n + 1) \\
&\leq 4(d-1)^2 n^2 (\log n + 1) \left[\frac{\lambda_{\max}(K)}{\lambda_{\min}(K)}\right]^{4n} \\
&\leq 4(d-1)^2 n^2 (\log n + 1) \Big[\frac{177}{22}\Big]^{4n}.
\end{aligned}$$

Consequently, according to Theorem 1.7 in Chapter 1 the amount of time needed for the Markov chain $\mathcal{M}_d$ to reach the stationary distribution $\mathcal{P}_K^n$ is

$$\begin{aligned}
\tau_\epsilon &\leq C \cdot \log\Big(\frac{1}{\epsilon \cdot \mathcal{P}_K^n(x)}\Big) \\
&\leq 4(d-1)^2 n^2 (\log n + 1) \Big[\frac{177}{22}\Big]^{4n} \cdot \log\left[\frac{(n!)^{d-1}}{\epsilon}\left(\frac{\lambda_{\max}(K)}{\lambda_{\min}(K)}\right)^n\right] \\
&\leq 4(d-1)^3 n^3 (\log n + 1) \Big[\frac{177}{22}\Big]^{4n} \cdot \log\left[\Big[\frac{177}{22}\Big]^{\frac{1}{d-1}} \Big(\frac{n!}{\epsilon^{\frac{1}{d-1}}}\Big)^{\frac{1}{n}}\right].
\end{aligned}$$

□

4.5 Conclusion

In this chapter, the connection between fixed cardinality determinantal point process and Latin Hypercube sampling is illustrated. The usage of n-DPP strategy allows LHS to spread design points as far as possible.

Two cases were taken into account, the first one by considering 2-dimensional LHS and the second one by considering d-dimensional LHS where $d > 2$.

In the first case, a DPP kernel was proposed to sample a set of points from n-DPP that fill the input space when LHS properties are satisfied. This is achieved by designing a Markov chain where n-DPP was defined as stationary distribution. An exponential bound on the mixing time for the Markov chain sampling from a n-DPP is obtained. The proof used for bounding the convergence speed was based on the inequality of Poincaré and canonical path method. Moreover, a calculation of the ratio of two different configurations' probabilities confirms that samples, which abide by the Latin Hypercube properties and satisfy adequate coverage of the output space, are more likely to be chosen.

The second case which corresponds to d-dimensional LHS, a generalization of the DPP kernel and a construction of a new Markov chain with stationary distribution is n-DPP took place.

Furthermore, the mixing time of this Markov chain is attained by multiplying the exponential bound previously obtained for $d = 2$ by $(d-1)^3$.
As generating LHS from n-DPP requires exponential bounds on the mixing time, thus, the constructed Markov chains are not rapidly mixing. Hence, the ideas described in this chapter can be developed by improving the bound on the mixing time and therefore the design points may be sampled according to n-DPP with a polynomial time.

Bibliography

[Casquilho et al.(2018)] C. M. Casquilho-Resende, N. D. Le, J. V. Zidek and Y. Wang. Design of monitoring networks using k-determinantal point processes. *Environmetrics*, Volume 29:e2483, 2018.

[Dussert et al.(1986)] C. Dussert, G. Rasigni, M. Rasigni and J. Palmari. Minimal spanning tree: A new approach for studying order and disorder. *Physical Review B*, Volume 34(5), pages 3528-3531, 1986.

[Euler(1782)] L. Euler. Recherches sur une nouvelle espèces de quarrés magique. *Verh. Zeeuwsch Genootsch, Wetensch*, Vlissengen 9, pages 85-239, 1782.

[Iman and Conover(1980)] R. L. Iman and W.J. Conover. Small sample sensitivity analysis techniques for computer models, with an application to risk assessment. *Communications in Statistics Part A-Theory and Methods*, Volume 9, pages 1749-1842, 1980.

[Johnson et al.(1990)] M. Johnson, L. Moore, and D. Ylvisaker. Minimax and maximin distance design. *Journal of Statistical Planning and Inference*, Volume 26, pages 131-148, 1990.

[McKay et al.(1979)] M. D. McKay, R. J. Beckman and W. J. Conover. Journal Article A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *American Statistical Association and American Society for Quality*, volume 21, pages 239-245, 1979.

[Morris and Mitchelle(1995)] M.D. Morris, T.J. Mitchell. Exploratory Designs for Computational Experiments. *Journal of Statistical Planning and Inference*, Volume 43, pages 381-402, 1995.

[Park(1994)] J.S. Park. Optimal Latin-hypercube designs for computer experiments. *Journal of Statistical Planning and Inference*, Volume 39, pages 95-111, 1994.

[Petelet et al.(2010)] M. Petelet, B. Iooss, O. Asserin and A. Loredo. Latin hypercube sampling with inequality constraints. *Advances in Statistical Analysis*, Volume 94, pages 325-339, 2010.

[Pratola et al.(2018)] M. T. Pratola, C. D. Lin and P. F. Craigmile. Optimal Design Emulators: A Point Process Approach. *arXiv:1804.02089v1 [stat.ME]*, 2018.

[Raj(1968)] D. Raj. Sampling Theory. *New York: McGraw-Hill*, 1968.

[Schwery and Wynn(1987)] M. Schwery and H. Wynn. Maximum entropy sampling. *Journal of Applied Statistics*, Volume 14, pages 165-170, 1987.

[Sheikholeslami and Razavi(2017)] R. Sheikholeslami and S. Razavi. Progressive Latin Hypercube Sampling: An Efficient Approach for Robust Sampling-based Analysis of Environmental Models. *Environmental Modelling & Software*, Volume 93, pages 109-126, 2017.

[Sinclair(1992)] A. Sinclair. Improved bounds for mixing rates of Markov chains and multicommodity flow. *Combinatorics, Probability and Computing* ***1***, pages 351-370, 1992.

[Wang et al.(2018)] Y. Wang, N. D. Le and J. V. Zidek. Stochastic Approximation Algorithms in Combinatorial Optimization. *arxiv:1709.00151v2 [stat.CO]*, 2018.

[Ye (1998)] Q.K. Ye. Orthogonal column Latin hypercubes and their application in computer experiments. *Journal of the American Statistical Association*, Volume 93, pages 1430-1439, 1998.

[Ye et al.(2000)] Q.K. Ye, W. Li and A. Sudjianto. Algorithmic construction of optimal symmetric Latin hypercube designs. *Journal of Statistical Planning and Inference*, Volume 90, pages 145-159, 2000.

Conclusion

This thesis aims to cover various applications of sampling from large-scale k-DPPs. Specifically, the first application of k-DPP sampling covers species on phylogenetic trees, the second one covers nodes from large graphs and the third one covers building experimental designs.

For saving time, the technique used to generate diverse samples of k-DPPs was based on Markov chains. Actually, the basic idea was to construct Markov chains which have the k-DPPs as their stationary distributions. Then, the bounds on the rate of convergence of the chains are illustrated with accordance to the bounds on the Poincaré constant.

The same Markov chain was proposed in chapters 2 and 3 to sample species from a phylogenetic tree and nodes from a large connected graph. In chapter 2, the DPP kernel used was the intersection kernel which can be written as a dot product of binary vectors. We have proven that in a polynomial time it returns an approximate sample of the k-DPP. Thus, a diverse subset of species in a phylogenetic tree can be obtained rapidly for biological studies. As a test to a real-world datasets of species, this approach was applied to extract 200 sample from the "Eukaryota" (taxa ID 2759) sub-tree of the Tree of Life of complete genomes. By comparing this method to the simple proportional method, the results show that our approach:

- is sensitive to the branching complexity,
- it favors divergent nodes,
- it is more stable upon sampling repetition where the proportional method uses pure random choices.

In practice, the algorithm is applied to a huge phylogenetic tree which contains 1827829 nodes from which 1464190 are species making necessary the use of the MCMC methodology. Another result was presented in this chapter to reach the goal we are looking for which is diversity. It states that if the tree is of maximum height h, for $0 < h_2 < h_1 < h/2$, choosing k species simultaneously joined by a subtree of height h_1 is more probable than choosing k species simultaneously joined by a subtree of height h_2.

Furthermore, in chapter 3 we have introduced a suitable kernel to sample a subset of nodes in a connected graph by avoiding redundancy. This kernel was the Moore-Penrose pseudoinverse of the normalized Laplacian matrix. The same reasoning of application one was applied to this special case of k-DPP on graphs. A polynomial bond on the mixing time

was supplied under certain condition on the second smallest eigenvalue of the normalized Laplacian. It is shown that it depends on whether the graph bottleneck is large or narrow. The larger the better for the convergence.

Chapter 4 is devoted to display the performance of fixed-size Determinantal Point Processes in experimental designs. First, our focus intended to introduce a positive kernel that respects the constraint needed to build a LHS design of order n and dimension d which is strictly confirmed by the occurrence of each point exactly once in each hyperplane. By taking into account the chosen kernel, an important part was presented to show that selecting points that provide a good coverage of the input space is more probable than selecting clusters of points or points that lie on a diagonal line. This heuristic is due to the fact that the design points should be far from each other, which is precisely what DPP prefers. Second, a specific Markov chain was proposed for LHS designs where n-DPP was defined as its stationary distribution. A bound on the mixing time for the Markov chain sampling from n-DPP was obtained basing on canonical paths where the lengths of these paths are taken into consideration.

Perspectives

The results provided in this thesis are motivating for future research:

- Perspective on chapter 2

  Since a large phylogenetic tree might contain similar species, we aimed in chapter 2 to sample diverse subsets of species which allowed obtaining an overview of different kinds of information related to the ground set. The DPP kernel chosen was the intersection kernel that, by comparing the sets of the species' ancestors, it gives the number of common ancestors. It is shown that the most probable set to be selected was the one which contains species that does not have a large number of common ancestors. Since this study covered the diversity of species in terms of ancestors, it will be interesting to extend it and focus also on the distance that separate them. This research allows to quantitatively compare diversities using functional diversity. Furthermore, our method can be developed to be a solution of a problem that draws the attention of the biologists regarding sampling the nodes (not necessary species) through a tree with respect to the diversity concept for example sample diverse nodes from the respiration tree.

- Perspective on chapter 3

  Many applications in biology and computer science require to sample diverse sets of nodes in graph. This goal is achieved in chapter 3 by sampling diverse subsets of nodes from a large connected graph with accordance to k-DPP which originally gives weights to distinct objects.
  On the other side, a topic of interest would be to sample nodes with another characteristics. Many applications especially in physics and video games need to sample sets of similar nodes. But on graphs, it is not a trivial issue to sample clusters of nodes according to a distribution. One way to overcome this difficulty is to use the Permanental point processes (PPP). The power of PPP lies in that the probability of choosing a particular set of items is proportional to the permanent of a matrix that defines the similarity of those items.

  Therefore, the perspective that would be considered is to design a rapidly mixing Markov chain which has q as a stationary distribution proportional to the Permanent. Although the idea seems very acceptable, the problem that might be encountered in this future work is how to compute the permanents as it is known that computing it is time-consuming.

- Perspective on chapter 4

  In chapter 4, we have displayed the capability of k-DPP with $k = n$ to generate a Latin Hypercube sampling of order n and dimension d. A DPP kernel was proposed to sample a set of points from n-DPP that fill the input space only when each sample point occurs only once in each hyperplane. This was achieved by designing a Markov chain where n-DPP was defined as the stationary distribution. An exponential bound on the mixing time for the Markov chain sampling from a n-DPP is obtained so further research is needed to ameliorate the bound of the convergence speed. While the proof used for bounding the mixing time was based on the inequality of Poincaré and canonical path method, the improvement can be made by spreading flow on the path between pair of states among some collection of canonical paths. In addition, further studies will be needed in order to prove that, in general, n-DPP produces better d-dimensional LHS than randomly generated LHS. This can be achieved by doing some simulations showing that n-DPP will outperform the random sample in terms of various criteria. Further, it is interesting to redo the work in Section 4 by considering the Gaussian kernel as the DPP kernel and then compare the results generated by our DPP kernel with those from the Gaussian kernel.