Fixed-Size Determinantal Point Processes Sampling For Species Phylogeny

Diala Wehbe, Nicolas Wicker, Baydaa Al-Ayoubi, Luc Moulinier

MathematicS

MathS In A.

In Action

Diala Wehbe, Nicolas Wicker, Baydaa Al-Ayoubi & Luc Moulinier

Fixed-Size Determinantal Point Processes Sampling For Species Phylogeny

Volume 10 (2021), p. 1-13.

<http://msia.centre-mersenne.org/item?id=MSIA_2021__10_1_1_0>

MathematicS In Action
Vol. 10, 1-13 (2021)

Fixed-Size Determinantal Point Processes Sampling For Species Phylogeny

Diala Wehbe [*]
Nicolas Wicker [**]
Baydaa Al-Ayoubi [***]
Luc Moulinier [†]

[*] Paul Painlevé Laboratory, University of Lille, 59650 Villeneuve D'Ascq, France and EDST, Lebanese University, Tripoli, Lebanon
E-mail address: diala.wehbe@net.usj.edu.lb
[**] Paul Painlevé Laboratory, University of Lille, 59650 Villeneuve D'Ascq, France
E-mail address: nicolas.wicker@univ-lille.fr
[***] Faculty of Sciences, Lebanese University, Rafic Hariri University Campus - Hadas, Lebanon
E-mail address: baydaa.ayoubi66@gmail.com
[†] ICube, CSTB (Complex Systems and Translational Bioinformatics), University of Strasbourg, 67085 Strasbourg, France
E-mail address: luc.moulinier@unistra.fr.

Abstract

Determinantal point processes (DPPs) are popular tools that supply useful information for repulsiveness. They provide coherent probabilistic models when negative correlations arise and also represent new algorithms for inference problems like sampling, marginalization and conditioning. Recently, DPPs have played an increasingly important role in machine learning and statistics, since they are used for diverse subset selection problems. In this paper we use k-DPP, a conditional DPP that models only sets of cardinality k, to sample a diverse subset of species from a large phylogenetic tree. The tree sampling task is important in many studies in modern bioinformatics. The results show a fast mixing sampler for k-DPP, for which a polynomial bound on the mixing time is given. This approach is applied to a real-world dataset of species, and we observe that leaves joined by a higher subtree are more likely to appear.

1. Introduction

A rooted phylogenetic tree is an oriented graph that depicts the evolutionary relationships amongst a set of species. The root of the tree represents the last common ancestor of all species of the set, the leaves of the tree correspond to the species, and an internal node represents the most recent common ancestor of all species descending from that node. Leaves and nodes are called taxons, and the subtree starting at a given node is called a clade. Currently, phylogenetic trees are generally built based on the complete genome or part of the DNA sequence of the species. As next-generation sequencing technologies become more and more efficient and cheap, the number of available DNA sequences or complete genomes has grown exponentially (NCBI `https://www.ncbi.nlm.nih.gov/genbank/statistics/`, GOLD, Genome OnLine Database, `https://gold.jgi.doe.gov/statistics`). As of August 2020, there are more than 240 000 complete sequenced genomes in public databases, without taking into account subspecies or strains (for example, there are 100 genomes of subspecies and strains of "Saccharomyces cerevisiae", the baker's yeast). Many studies in modern bioinformatics, including comparative genomics, multiple sequence alignment, etc., are based on a small subset of the partially or completely available sequenced species. The size of the subset is restricted (from hundreds to thousands of speciess)

Keywords: Determinantal point process, Kernel, Markov chain, Metropolis-Hasting, Mixing time, Phylogenetic tree.

either because these studies require human intervention or because of the computational time required to process the data. The choice of the species subset is crucial, as it should reflect the diversity within the tree they are taken from, and so minimize the redundancy resulting from the selection of very closely related species. The goal of the method presented here is to sample a phylogenetic tree in an efficient and relevant way.

Determinantal point processes (DPPs), which arise in random matrix theory ([16], [7]) and in quantum physics, were first identified by Machhi [15] who called them *fermion processes*. They are probabilistic models that capture negative correlation and give the likelihood of selecting a subset of items as the determinant of a kernel matrix. Kulesza and Taskar [13] provide a gentle introduction to DPPs and show that they offer appealing properties and practical algorithms, focusing on the extensions that are the most relevant to the machine learning community. For instance, DPPs can be used to select diverse sets of sentences to form document summaries, or to find multiple non-overlapping human poses in an image (see [13]).

For an integer $0 \leq k \leq n$, conditioning on sampling sets of fixed cardinality k, one obtains a k-DPP. The simplest example is described by Kulesza and Taskar [12] on a real world image search problem, where the goal is to show users diverse sets of images that correspond to their query. For more information about k-DPP and their applications, we refer to recent studies by Deshpande and Rademacher [5] and Kulesza and Taskar [13].

The main idea here is to sample, according to k-DPP, a diverse set of species in a very large phylogenetic tree containing millions of nodes. 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 and x' with fine distinctions in the degree to which x and x' are distant from each other in the tree. As mentioned above, the leaves are connected to each other through ancestors. Therefore, in this paper we chose to use a specific kernel, namely the intersection kernel that compares the ancestors of one leaf with the ancestors of another one.

Different algorithms have been presented for sampling from k-DPP, but these algorithms commonly need the eigen-decomposition of a matrix which typically has a size of more than one million ([18]). Thus, they are inefficient in time and memory (see Mehta and Gaudin [10], Deshpande and Rademacher [5], and Kulesza and Taskar [12]). In contrast, Markov chain techniques are very appealing in the context of generating random samples of a k-DPP, due to their simplicity and efficiency. For example, Kang [9] considered a Markov Chain Monte Carlo (MCMC) generated by the Metropolis-Hastings algorithm to sample. In his work, the coupling argument is not well defined, hence the proof of the rapid mixing time of the Markov chain is wrong, but the sampling scheme is correct. Borcea et al. [2] show that any DPP is a Strongly Rayleigh (SR) distribution. Such distributions are defined by strong negative dependence properties. Since SR distributions are amenable to efficient Markov chain sampling, Anari et al. [1] used a lazy MCMC, described also by Kang [9], and showed that the natural MCMC algorithm mixes rapidly in the support of a homogeneous SR distribution. Since DPPs are considered special cases of SR distributions that are closed under truncation, then any k-DPP is a homogeneous SR distribution. Their result implies that the same Markov Chain can be used to efficiently generate random samples of a k-DPP.

This paper aims to show the performance of k-DPP in selecting a subset of species to represent a much larger set of species in a polynomial time and illustrate the convergence speed result of Anari et al. [1] by making it explicit in this particular case.

The remainder of this paper is organized as follows. Section 2 presents some basic facts about DPPs, kernel functions and our k-DPP kernel. Our main result is presented in Section 3. It states that if a tree is 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. Then, we focus on the mixing time of the Markov chain specified in Algorithm 1 for the k-DPP. The resulting mixing time depends on the height of the phylogenetic tree. In

Section 4, we apply our approach to a real case involving a large dataset of species. Finally, Section 5 presents our conclusions.

2. Background

2.1. k-Determinantal Point Processes

For a discrete set $\mathcal{X}$, a DPP is defined via an L-ensemble, by using a symmetric and positive semidefinite matrix L that directly defines the probability of observing each subset of the set $\mathcal{X}$. In this paper, we will work with the most relevant construction of k-DPPs based on L-ensembles.

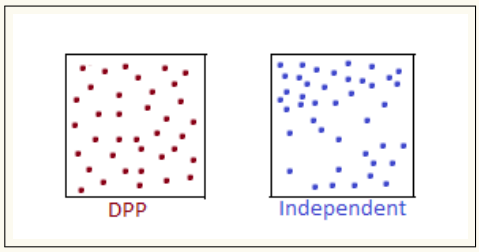

FIGURE 2.1. A set of points in the plane drawn from a DPP (left), and the same number of points sampled independently using a Poisson process (right).

For a discrete set $\mathcal{X} = \{1, 2, \ldots, n\}$ of n items, a DPP, denoted $\mathcal{P}_L$, is a probability distribution on the set $2^{[n]}$ of all subsets of $[n] = \{1, 2, \ldots, n\}$ defined by a $n \times n$ positive semidefinite matrix L indexed by the elements of $\mathcal{X}$ such that if X is a random subset drawn according to $\mathcal{P}_L$, we have

$$\mathcal{P}_L(X) = \frac{\det(L_X)}{\det(L + I)},$$

where I is the $n \times n$ identity matrix and L_X is the principal submatrix of L indexed by the elements of X ($L_X = (L_{ij})_{i,j \in X}$). Note that $\mathcal{P}_L$ is suitably normalized in view of the identity

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

DPP can model two distinct characteristics: the size of the set and its content. Kulesza and Taskar [13] propose a solution to this problem. They introduced the k-DPP, a conditional DPP that models only sets of cardinality k.

A k-DPP is a distribution over all subsets $X \subseteq \mathcal{X}$ with cardinality k. The L-ensemble construction of a k-DPP, denoted $\mathcal{P}_L^k$, gives the following probability to k-sets:

$$\mathcal{P}_L^k(X) = \frac{\det(L_X)}{\sum_{|X'|=k} \det(L_{X'})}, \tag{2.1}$$

where $|\cdot|$ denotes the cardinality of the set, $|X| = k$ and L any positive semidefinite matrix. Hereafter, we show how to choose the kernel.

2.2. DPP Kernel

To provide the similarity between nodes in the data $\mathcal{X}$, a positive semidefinite kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is needed. Positive semidefiniteness means 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 a $|\mathcal{X}| \times |\mathcal{X}|$ matrix with rows and columns indexed by the elements of $\mathcal{X}$, and related to the kernel by $K_{xx'} = K(x, x')$. This matrix is called the Gram matrix of the kernel.

Different kernels have successfully been applied to capture the long-range relationships between pairs of points induced by the local structure of a graph. The Laplacian matrix has often been effective for graph isomorphism problems in biochemistry and design of statistical experiments, and plays an important role in the analysis of random walks and electrical networks on graphs ([6], [4], and [17]). It is also known as the Kirchhoff matrix or the information matrix. The Laplacian matrix of the graph is defined as $L = D - A$, 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 $D = \mathrm{Diag}(A_i)$, with $A_i = \sum_j A_{ij}$. Then, the pseudoinverse $L^\dagger$ of L is often used as a kernel on a graph (or a tree) and can be interpreted as a similarity measure ([14]). In addition, $L^\dagger$ can be used to compute the average commute time (see Gobel and Jagers [8]), which is the average number of steps taken by a random walker to reach node j and come back to node i when starting from node i.

Let us mention the work of Kondor and Lafferty [11] on diffusion kernels, a class of exponential kernels on graphs. They showed how these kernels correspond to standard Gaussian kernels in a continuous limit. The kernel is given by the matrix exponential of L:

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

where the limit always exists and 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 the symmetric matrix is positive semidefinite.

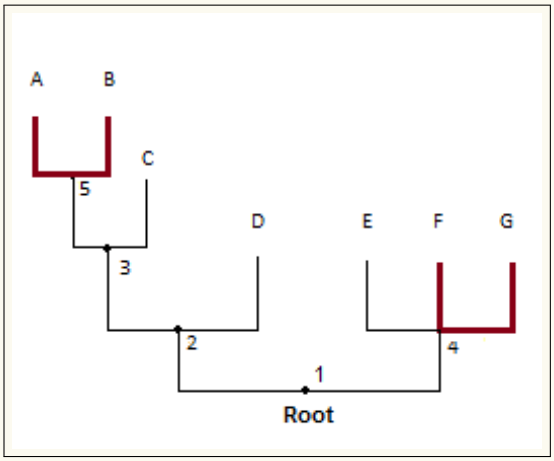

Figure 2.2. A subtree of height 5.

In our case, we note that these two kernels are troublesome. Indeed, referring to the subtree in Figure 2.2, F and G are two close species that are not far from the root. A and B are also two close species, but with longer branches making them more distant from the root. With the

two kernels we will have $K(A,B) \approx K(F,G)$, which is incorrect from a phylogenetical point of view.

When considering two species A and B, it is important to take into account the fact 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, a more interesting kernel is one that compares the set of ancestors of species A with the set of ancestors of species B, denoted by E_A and E_B respectively; hence the kernel is here defined as

$$K(A,B) = |E_A \cap E_B|, \tag{2.2}$$

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 semidefinite 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 $L_X = X^T X$ where the elements of the matrix X^T are: $\forall i \in \{1,\dots,k\}$, $\forall j \in \{1,\dots,n\}$ where n is the number of nodes present in the tree,

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

We illustrate this by an example where $k = 4$ and $n = 12$. Let us take the subtree above in Figure 2.2, by choosing $X = \{A, C, D, G\}$, the matrix 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

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

One might also consider the normalized kernel $\widetilde{K}$ defined as follows:

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

However with this normalization, we will face the same problem as with the first two kernels. Consequently, our choice has been directed towards the intersection kernel (equation (2.2)).

3. Sampling via k-DPP

3.1. 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). For simplification purposes, 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 3.1 shows that choosing k species simultaneously joined by a subtree of height h_1 (i.e., 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 (i.e., there are no leaves connected by a subtree of height $h_2 - 1$). This demonstrates that k-DPP allows to achieve diversity in the most probable samples.

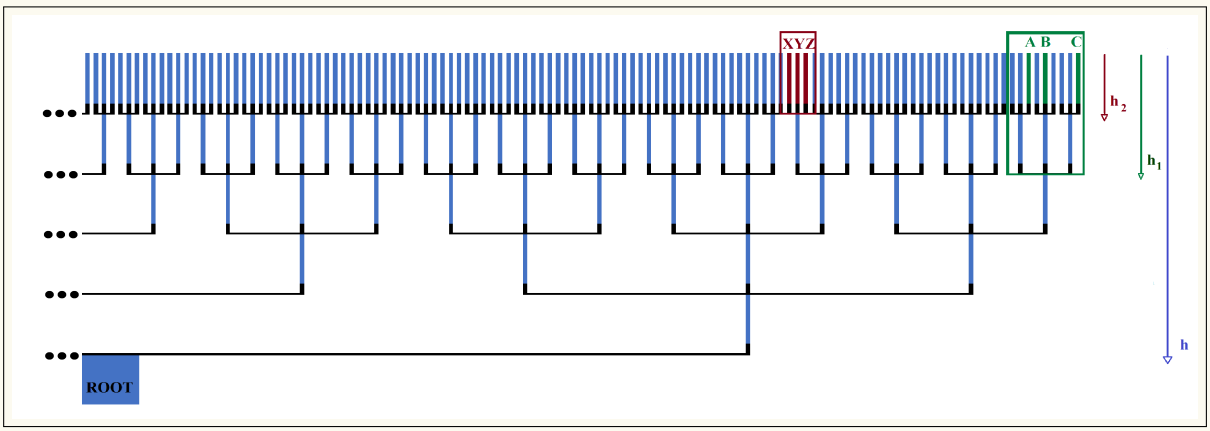

Figure 3.1. A part of a perfect 3-ary tree where the length of the longest path from the root to a leaf is $h = 5$. The species A, B and C are joined by a subtree of height $h_1 = 2$ and X, Y and Z are joined by a subtree of height $h_2 = 1$.

Proposition 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 A be a set containing k leaves all joined initially 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 is given by:

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

Then, the determinant of L_A can be rewritten as follows:

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

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

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

The number of subtrees of height h_1 (resp. h_2), whose leaves are a subset of the leaves of T and roots are nodes at height h_1 (resp. h_2) in T, is given by k^{h-h_1} (resp. k^{h-h_2}). 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 probability 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:

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

This concludes the proof. □

Now, we illustrate the results of this Proposition by considering the case of the perfect 3-ary tree represented in Figure 3.1 where $h=5, h_1=2$ and $h_2=1$. Then, according to Proposition 3.1 it is 18 times more likely to choose 3 leaves simultaneously joined by a subtree of height 2 than by a subtree of height 1.

3.2. **Convergence Theorem**

In this section, we will study the convergence of the distribution of a Markov chain $\mathcal{M}$ to the stationary distribution $\mathcal{P}_L^k$. More precisely, the bound found by Anari et al. [1] will be made explicit in the particular case we are dealing with. As a preliminary to the proof of the main convergence theorem stated below, we need the following section.

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

For large datasets, it is inefficient to use Kulesza and Taskar's k-DPP sampling algorithm [12], since eigen-decomposition of L is needed. To solve this problem, Kang [9] and Anari et al. [1] present an original method based on the Metropolis-Hastings algorithm. This method proposes to define a Markov chain $\mathcal{M}$ with $[n]$ as the state space such that at each step we stay at state $X\subseteq[n]$ with $|X|=k$ with probability $1/2$ and we propose to move to a new state according to the transition probability q with probability $1/2$. If the current state is X, a new proposal state is drawn as follows: two elements $u\in X$ and $v\notin X$ are chosen randomly where u is to be removed from the current set X of size k, and v is to be added. Hence, the main idea behind this method is to create a new configuration by selecting a row and column of L_X and replace them with the row and column corresponding to the element to be added.

Therefore, for $X=Y\cup\{u\}$, the proposal state is $X'=Y\cup\{v\}$, which is accepted with probability

$$p=\min\left\{1,\frac{\det L_{X'}\cdot q(X',X)}{\det L_X\cdot q(X,X')}\right\}.$$

By taking into consideration the fact that q is a symmetric transition probability, the transition probability matrix $P_{\mathcal{P}_L^k}$ of $\mathcal{M}$ is given by:

$$P_{\mathcal{P}_L^k}(X,X')=\begin{cases}q(Y\cup\{u\},Y\cup\{v\})\cdot\frac{1}{2}\min\left\{1,\frac{\det L_{Y\cup\{v\}}}{\det L_{Y\cup\{u\}}}\right\} & \text{if } u\neq v\\ 1-\sum_{w\neq u}q(Y\cup\{u\},Y\cup\{w\})\cdot\frac{1}{2}\min\left\{1,\frac{\det L_{Y\cup\{w\}}}{\det L_{Y\cup\{u\}}}\right\} & \text{otherwise.}\end{cases}$$

As $P_{\mathcal{P}_L^k}(X,X)\geq\frac{1}{2}$ for all X, then the Markov chain described above is said to be a lazy chain.

For large X, a single iteration may become very costly. Observing that $L_{Y\cup\{u\}}$ can be represented as the following block matrix:

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

where $b_u = L(i,u)_{i\in Y} \in \mathbb{R}^{|Y|}$ and $c_u = L(u,u)$, Kang [9] rewrites the determinant of $L_{Y\cup\{u\}}$ as

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

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

$$p = \min\left\{1, \frac{\det(L_{X'})}{\det(L_X)}\right\} = \min\left\{1, \frac{c_v - b_v^T L_Y^{-1} b_v}{c_u - b_u^T L_Y^{-1} b_u}\right\}.$$

Thus, the lazy Markov chain $\mathcal{M}$ produced by the Metropolis-Hastings algorithm is defined to obtain a sample from k-DPP. This procedure is outlined in the following algorithm.

Algorithm 1 Markov chain for sampling from $\mathcal{P}_L^k$ [[9], [1]]

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

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

      $X \leftarrow Y \cup \{v\}$ 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 which has $\mathcal{P}_L^k$ as its stationary distribution.

3.2.2. *Notations and Preliminaries*

Definition 3.2. The total variation distance between two probability distributions ν and μ on a finite space Ω is:

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

If Y is a random variable that samples according to μ and $\|\mu - \nu\|_{TV} \leq \epsilon$, then we say Y is an ϵ-approximate sample of ν.

Definition 3.3. (Mixing Time). Let $\mathcal{C}$ be a Markov chain on the state space Ω, started at $x \in \Omega$ with transition probability matrix P and stationary distribution π. For $\epsilon > 0$, the total variation mixing time of the Markov chain $\mathcal{C}$ is defined as follows:

$$\tau_x(\epsilon) := \min\{t : \|P^t(x,\cdot) - \pi\|_{TV} \leq \epsilon\},$$

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

Now, we will use the theorem of Anari et al. [1], which gives the convergence speed of the distribution of the lazy MCMC described in Algorithm 1 to its stationary distribution. In what follows, $P_{\mathcal{P}_L^k}$ denotes the transition probability matrix of $\mathcal{M}$.

Theorem 3.4 ([1])**.** *For any k-DPP, $\mathcal{P}_L^k$ defined on all subsets $X \subset \mathcal{X}$ of size k, $\epsilon > 0$ and starting point X, the Markov chain described in Algorithm 1 verifies:*

$$\tau_X(\epsilon) \leq \frac{1}{C_{\mathcal{P}_L^k}} \cdot \log\left(\frac{1}{\epsilon \cdot \mathcal{P}_L^k(X)}\right),$$

where

$$C_{\mathcal{P}_L^k} = \min_{X,X' \in S} \max\left(P_{\mathcal{P}_L^k}(X, X'), P_{\mathcal{P}_L^k}(X', X)\right)$$

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

Theorem 3.4 is a very generic theorem, the linear factors k and n are appealing but the term under the log can make the bound quite large if $\mathcal{P}_L^k(X)$ is small enough. Therefore, the next theorem provides a more accurate estimate by giving a lower bound to $\mathcal{P}_L^k(X)$.

3.2.3. *Mixing Time*

The following theorem studies the convergence speed of the distribution of the Markov chain $\mathcal{M}$ to the stationary distribution $\mathcal{P}_L^k$, for which a polynomial bound on the mixing time is offered.

Theorem 3.5. *Let $\mathcal{P}_L^k$ be a k-DPP where L is the matrix defined by Equation* (2.2)*. For any $\epsilon > 0$, the lazy Markov chain defined in Algorithm 1 on a tree where all points have height h generates an ϵ-approximate sample of $\mathcal{P}_L^k$ in time*

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

Proof. As a first step and according to Theorem 3.4, we need to find a lower bound for

$$C_{\mathcal{P}_L^k} = \min_{X,X' \in \mathrm{supp}\{\mathcal{P}_L^k\},\ |X|=|X'|=k} \max\left(P_{\mathcal{P}_L^k}(X, X'), P_{\mathcal{P}_L^k}(X', X)\right).$$

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 [9], the acceptance probability is lower bounded by the ratio of the determinants of two matrices as follows:

$$\frac{\det(L_{Y \cup \{v\}})}{\det(L_{Y \cup \{u\}})} = \frac{c_v - b_v^T L_Y^{-1} b_v}{c_u - b_u^T L_Y^{-1} b_u}, \tag{3.1}$$

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

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

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

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

which is the result of Anari et al. [1].

To complete the proof, $\mathcal{P}_L^k$ needs to be bounded from below. By property of Gram matrices determinants,

$$\det(L_X) = d(u, X \setminus \{u\})^2 \det(L_{X\setminus\{u\}})$$

where $d(u, X \setminus \{u\})$ represents the distance between u and its projector on the span of vectors $X \setminus \{u\}$. This distance is minimized by having all coordinates equal except one, that is when the leaves have only common ancestors. Thereby, we obtain as lower bound for $\mathcal{P}_L^k$:

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

In addition, according to equation (2.1), we should also calculate the normalization constant. Then, for any set X of cardinality k we look for $X' \subseteq [n]$ 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 compute the determinant of $L_{X'}$ as follows:

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

Consequently,

$$\mathcal{P}_L^k(X) = \frac{\det(L_X)}{\sum_{|X'|=k} \det(L_{X'})} \geq \frac{1 + k(h-1)}{\binom{n}{k} (h-1)^k \left(1 + \frac{k}{h-1}\right)} \geq \frac{n^{-k}\left(1 + k(h-1)\right)}{(h-1)^k \left(1 + \frac{k}{h-1}\right)}.$$

Now, by using Theorem 3.4, we can directly upper bound the mixing time in total variation distance as follows:

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

This proves the result. □

The obtained speed is thus mainly quadratic in k and linear in n. Usually, parameter k (number of leaves/species been sampled) is defined by the user. Note that k can range from tens to several thousands of species and its value mainly depends on the type of the undertaken study. If the sampled set of species is dedicated to Multiple Sequence Alignment (MSA) creation, the range of k is usually between 50 and 500. Indeed, MSAs have generally to be manually examined and curated and more than 500 sequences are barely manageable for human inspection. In the case of phylogenetic or comparative genomic studies, k can be up to several thousands of species as these studies are mainly computational, although the large number of species may then affect the computation times.

4. **Experiments**

In order to evaluate our method, we compare it with an alternative straightforward way of sampling within a tree. In this method, called henceforth "proportional method", species are selected proportional to the number of children weighted by their descendants in each subbranch, starting from the root. In other words, at a given node, the k samples are shared between the N children proportionally to the respective number of leaves attached to each of them. If the total number of leaves descending from the root node is L and l_i the number of leaves attached to children i, then the number of samples to be taken from node i is simply $k_i = \frac{l_i}{L}k$ with the necessary roundings. The process is repeated for each node until reaching the leaves.

The two algorithms were used to sample $k = 200$ species from the "Eukaryota" (taxa ID 2759) branch of the "Tree of Life", a phylogenetic tree of all completely sequenced species. The list of complete genomes and their taxonomy were taken from the UniProt (Universal Protein Resource, https://www.uniprot.org) database (Pundir et al. [18]). The tree contains all clades up to the species level, subspecies and strains are discarded from the set of taxa. As of January 25 2018, the tree contains 3871 nodes including 1356 leaves (species). After applying the two methods, the resulting sets of sampled species were boxed on the tree as shown in Figure 1 as supplementary material. The yellow boxed squares correspond to our method, the blue boxed squares correspond to the proportional method.

A close inspection of the two sets shows that Algorithm 1 favors nodes with high complexity compared to the proportional method. For example, under the "Craniata" taxon, part of the "Vertebrata" branch, Algorithm 1 selects 22 species whereas the proportional method selects only 18. Under the "Aves" class, our method selects 11 species where the proportional one selects only 8, leading to a poorer diversity. This could be explained by the fact that our method better fits the tree topology.

Under the "Primates" node, both methods select two species: our algorithm selects "Otolemur garnetti" and "Homo sapiens", two species that are evolutionarily distant from each other. In contrast, the proportional method selects "Pan troglodytes" and "Gorilla gorilla", which are much more similar to each other and do not reflect the diversity of the "Primates" branch. Our method makes a choice that optimizes the distance between the selected taxa. As a last example, the whole "Amphibia" branch is completely absent from the proportional method sample set. Our method selects at least one species extending the diversity at the "Tetrapoda" level.

As a robustness test, the DPP method has been applied to a huge phylogenetic tree (the whole "Tree of life" limited at the species clade), which contains 1363190 nodes of which 1232884 are species ([18]). The tree has a height of 37. The number of iterations is set to 1000000 and $k = 1000$. The computation takes 106 minutes on an Intel Xeon E5-2640 processor 2.5 GHz.

5. Conclusion

In this paper, the main focus was on an application of k-DPP to sampling of leaves in phylogenetic trees. The convergence speed of the general case has been treated by the article of Anari et al. [1]. They used a greedy algorithm of Çivril and Magdon-Ismail [3] to generate a set $X \subseteq [n]$ such that $\mathcal{P}_L^k(X)$ is bounded away from zero. They found a set X such that $\det(L_X) \geq n^{-k}$. However, in Theorem 3.5 and according to the kernel chosen (equation (2.2)), we managed directly to bound $\det(L_X)$ without using the algorithm of Çivril and Magdon-Ismail [3] allowing us to gain substantial time at the initialization step and also making the convergence speed bound more explicit.

The experiments demonstrate 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 this is achieved in a polynomial time with respect to k and the number of leaves n.

Acknowledgements

We are grateful to Julie D. Thompson for careful reading of the manuscript and English language correction.

References

[1] N. Anari, S. O. Gharan, and A. Rezaei. Monte Carlo Markov chain algorithms for sampling strongly Rayleigh distributions and determinantal point processes. *In COLT*, 2016.

[2] J. Borcea, P. Branden, and T. M. Liggett. Negative dependence and the geometry of polynomials. *J. Am. Math. Soc.*, 22:521–567, 2009.

[3] A. Çivril and M. Magdon-Ismail. On selecting a maximum volume sub-matrix of a matrix and related problems. *Theor. Comput. Sci.*, pages 4801–4811, 2009.

[4] D. M. Cvetkovic, M. Doob, and H. Sachs. *Spectra of Graphs*. Academic Press Inc., 1980.

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

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

[7] J. Ginibre. Statistical ensembles of complex, quaternion,and real matrices. *J. Math. Phys.*, 6:440–449, 1965.

[8] F. Gobel and A. Jagers. Random walks on graphs. *Stochastic Processes Appl.*, 2:311–336, 1974.

[9] B. Kang. Fast determinantal point process sampling with application to clustering. *NIPS*, pages 2319–2327, 2013.

[10] R. Kannan and S. Vempala. Spectral algorithms. *Foundations and Trends in Theoretical Computer Science*, 4:157–288, 2009.

[11] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the 19th International Conference On Machine Learning*, pages 315–322. Omnipress, 2002.

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

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

[14] L. Lovász. Random walks on graphs: A survey. *Combinatorics*, 2:1–46, 1993.

[15] O. Macchi. The Coincidence approach to stochastic point processes. *Adv. Appl. Probab.*, 7(1):83–122, 1975.

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

[17] R. Merris. Laplacian matrices of graphs: A survey. *Linear Algebra Appl.*, 197-198:143–176, 1994.

[18] S. Pundir, M. J. Martin, and C. O'Donovan. UniProt Protein Knowledgebase. *Methods Mol. Biol*, 1558:41–55, 2017.