Exact Sampling of Determinantal Point Processes without Eigendecomposition

Claire Launay, Bruno Galerne, Agnès Desolneux

Exact Sampling of Determinantal Point Processes without Eigendecomposition

Claire Launay claire.launay@parisdescartes.fr
Laboratoire MAP5, CNRS
Université Paris Descartes, Sorbonne Paris Cité
Paris, 75006, FRANCE

Bruno Galerne bruno.galerne@univ-orleans.fr
Institut Denis Poisson
Université d'Orléans, Université de Tours, CNRS
Orléans, 45067, FRANCE

Agnès Desolneux agnes.desolneux@cmla.ens-cachan.fr
CMLA, ENS Cachan, CNRS
Université Paris Saclay
Cachan, 94230, FRANCE

Abstract

Determinantal point processes (DPPs) enable the modeling of repulsion: they provide diverse sets of points. This repulsion is encoded in a kernel K that can be seen as a matrix storing the similarity between points. The exact algorithm to sample DPPs uses the spectral decomposition of K, a computation that becomes costly when dealing with a high number of points. Here, we present an alternative exact algorithm in the discrete setting that avoids the eigenvalues and the eigenvectors computation. Instead, it relies on the Cholesky decomposition of the matrix K and a thinning procedure. It can be, for some applications specified below, faster than the original algorithm.

Keywords: Determinantal point processes, Exact Sampling, Thinning, Cholesky decomposition, General marginal

1. Introduction

Determinantal point processes (DPPs) are processes that capture negative correlations. The more similar two points are, the less likely they are to be sampled simultaneously. Then DPPs tend to create sets of diverse points. They naturally arise in random matrix theory (Ginibre, 1965) or in the modelling of a natural repulsive phenomenon like the repartition of trees in a forest (Lavancier et al., 2015). Ever since the work of Kulesza and Taskar (2012a), these processes have become more and more popular in machine learning, thanks to their ability to draw subsamples that account for the inner diversity of data sets. This property is useful for many applications, such as summarizing documents (Dupuy and Bach, 2018), improving a stochastic gradient descent by drawing diverse subsamples at each step (Zhang et al., 2017) or extracting a meaningful subset of a large data set to estimate a cost function or some parameters (Tremblay et al., 2018b; Bardenet et al., 2017; Amblard et al., 2018). Several issues are under study, as learning DPPs, for instance through maximum likelihood estimation (Kulesza and Taskar, 2012b; Brunel et al., 2017), or sampling these processes. Here we will focus on the sampling question and we will only deal with a discrete and finite

determinantal point process Y, defined by its kernel matrix K, a configuration particularly adapted to machine learning groundsets.

The main algorithm to sample DPPs is a spectral algorithm (Hough et al., 2006) : it uses the eigendecomposition of K to sample Y. It is exact and in general quite fast. Yet, the computation of the eigenvalues of K may be very costly when dealing with large-scale data. That is why numerous algorithms have been conceived to bypass this issue. Some authors tried to design a sampling algorithm adapted to specific DPPs. For instance, it is possible to speed the initial algorithm up by assuming that K has a bounded rank (Kulesza and Taskar, 2010; Gartrell et al., 2017). These authors use a dual representation of the kernel so that almost all the computations in the spectral algorithm are reduced. One can also deal with another class of DPPs associated to kernels K that can be decomposed in a sum of tractable matrices (Dupuy and Bach, 2018). In this case, the sampling is much faster and the authors study the inference on these classes of DPPs. At last, Propp and Wilson (1998) use Markov chains and the theory of coupling from the past to sample exactly particular DPPs : uniform spanning trees.

Another type of sampling algorithms is the class of approximate methods. Some authors approach the original DPP with a low rank matrix, either by random projections (Kulesza and Taskar, 2012a; Gillenwater et al., 2012) or thanks to the Nystrom approximation (Affandi et al., 2013). The Monte Carlo Markov Chain methods offer also nice approximate sampling algorithms for DPPs. It is possible to obtain satisfying convergence guarantees for particular DPPs; for instance, k-DPPs with fixed cardinal (Anari et al., 2016; Li et al., 2016a) or projection DPPs (Gautier et al., 2017). Li et al. (2016b) even proposed a polynomial-time sampling algorithm for general DPPs, thus correcting the initial work of Kang (2013). These algorithms are commonly used as they save significant time but the price to pay is the lack of precision of the result.

As one can see, except the initial spectral algorithm, no algorithm allows for the exact sampling of a generic DPP. The main contribution of this paper is to introduce such a general and exact algorithm that does not involve the kernel eigendecomposition. The proposed algorithm is a sequential thinning procedure that relies on two new results: (i) the explicit formulation of the marginals of any determinantal point process and (ii) the derivation of an adapted Bernoulli point process containing a given DPP.

The rest of the paper is organized as follows : in the next section, we present the general framework of determinantal point processes and the classic spectral algorithm. In Section 3, we provide an explicit formulation of the general marginals and pointwise conditional probabilities of any determinantal point process, from its kernel K. Thanks to these formulations, we first introduce a "naive", exact but slow, sequential algorithm that relies on the Cholesky decomposition of the kernel K. In Section 4, using the thinning theory, we accelerate the previous algorithm and introduce a new exact sampling algorithm for DPPs that we call the sequential thinning algorithm. Its computational complexity is compared with that of the two previous algorithms. In Section 5, we display the results of some experiments comparing these three sampling algorithms and we describe the conditions under which the sequential thinning algorithm is more efficient than the spectral algorithm. Finally, we discuss and conclude around this algorithm.

2. DPPs and their Usual Sampling Method : the Spectral Algorithm

In the next sections, we will use the following notations. Let us consider a discrete finite set $\mathcal{Y} = \{1, \dots, N\}$. For $M \in \mathbb{R}^{N \times N}$ a matrix, we will denote by $M_{A \times B}$, $\forall A, B \subset \mathcal{Y}$, the matrix $(M(i,j))_{(i,j) \in A \times B}$ and the short notation $M_A = M_{A \times A}$. Suppose that K is a Hermitian positive semi-definite matrix of size $N \times N$, indexed by the elements of $\mathcal{Y}$, so that any of its eigenvalues is in $[0, 1]$. A subset $Y \subset \mathcal{Y}$ is said to follow a DPP distribution of kernel K if,

$$\forall A \subset \mathcal{Y}, \mathbb{P}(A \subset Y) = \det(K_A).$$

The spectral algorithm is standard to draw a determinantal point process. It relies on the eigendecompostition of its kernel K. It was first introduced by Hough et al. (2006) and is also presented in a more detailed way by Scardicchio et al. (2009); Kulesza and Taskar (2012a) or Lavancier et al. (2015). It proceeds in 3 steps : the first step is the computation of the eigenvalues λ_j and the eigenvectors v^j of the matrix K. The second step consists in randomly selecting a set of active eigenvectors according to N Bernoulli variables of parameter λ_i, for $i = 1, \dots, N$. The third step is drawing sequentially the associated points using a Gram-Schmidt process.

Algorithm 1 The spectral sampling algorithm

1. Compute the orthonormal eigendecomposition (λ_j, v^j) of the matrix K.
2. Select the active frequencies: Draw a Bernoulli process $\mathbf{X} \in \{0,1\}^N$ with parameter $(\lambda_j)_j$.

   Denote by n the number of active frequencies, $\{\mathbf{X} = 1\} = \{j_1, \dots, j_n\}$. Define the matrix $V = \left(v^{j_1}\, v^{j_2}\, \cdots\, v^{j_n}\right) \in \mathbb{R}^{N \times n}$ and denote by $V_k \in \mathbb{R}^n$ the k-th line of V, for $k \in \mathcal{Y}$.
3. Return the sequence $Y = \{y_1, y_2, \dots, y_n\}$ sequentially drawn as shown:
   For $l = 1$ to n
   - Sample a point $y_l \in \mathcal{Y}$ from the discrete distribution,

$$p_k^l = \frac{1}{n - l + 1} \left(\|V_k\|^2 - \sum_{m=1}^{l-1} |\langle V_k, e_m \rangle|^2 \right), \forall k \in \mathcal{Y}.$$

   - If $l < n$, define $e_l = \frac{w_l}{\|w_l\|} \in \mathbb{R}^n$ where $w_l = V_{y_l} - \sum_{m=1}^{l-1} \langle V_{y_l}, e_m \rangle e_m$.

This algorithm is exact and relatively fast but it becomes heavy when the size of the groundset grows. For a groundset of size N and a sample of size n, the third step costs $O(Nn^3)$ because of the Gram-Schmidt orthonormalisation. Tremblay et al. (2018a) propose to speed it up thanks to optimized computations and they achieve the complexity $O(Nn^2)$ for this third step. Nevertheless, the eigendecomposition of the matrix K is the heaviest part of the algorithm, as it runs in time $O(N^3)$, and we will see in the numerical results that this first step represents in general more than 90% of the running time of the spectral algorithm. As nowadays the amount of data explodes, in practice the matrix K is very large so it seems

relevant to try to avoid this costly operation. We compare the time complexities of the different algorithms presented in this paper at the end of Section 4. In the next section, we show that any DPP can be exactly sampled by a sequential algorithm that does not require the eigendecomposition of K.

3. Sequential Sampling Algorithm

Our goal is to build a competitive algorithm to sample DPPs that does not involve the eigendecomposition of the matrix K. To do so, we first develop a "naive" sequential sampling algorithm and subsequently, we will accelerate it thanks to a thinning procedure, presented in Section 4.

3.1 Explicit General Marginal of a DPP

First, we need to explicit the marginals and the conditional probabilities of any DPP. When $I - K$ is invertible, a formulation of the explicit marginals already exists (Kulesza and Taskar, 2012a), it implies to deal with a L-ensemble L instead of the matrix K. However, this hypothesis is reductive : among others, it ignores the useful case of projection DPPs, when the eigenvalues of K are either 0 or 1. We show below that general marginals can easily be formulated from the associated kernel matrix K. For all $A \subset \mathcal{Y}$, we denote I^A the $N \times N$ matrix with 1 on its diagonal coefficients indexed by the elements of A, and 0 anywhere else. We also denote $|A|$ the cardinal of any subset $A \subset \mathcal{Y}$ and $\overline{A} \in \mathcal{Y}$ the complementary set of A in $\mathcal{Y}$.

Proposition 1 (Distribution of a DPP) *For any $A \subset \mathcal{Y}$, we have*

$$\mathbb{P}(Y = A) = (-1)^{|A|} \det(I^{\overline{A}} - K).$$

Proof We have that $\mathbb{P}(A \subset Y) = \sum_{B \supset A} \mathbb{P}(Y = B)$. Thanks to the Möbius inversion formula (see Appendix A), for all $A \subset \mathcal{Y}$,

$$\begin{aligned}\mathbb{P}(Y = A) &= \sum_{B \supset A} (-1)^{|B \setminus A|} \mathbb{P}(B \subset Y) = (-1)^{|A|} \sum_{B \supset A} (-1)^{|B|} \det(K_B) \\ &= (-1)^{|A|} \sum_{B \supset A} \det((-K)_B)\end{aligned}$$

Besides, Kulesza and Taskar (2012a) state in Theorem 2.1 that $\forall L \in \mathbb{R}^{N \times N}, \forall A \subset \mathcal{Y}$, $\sum_{A \subset B \subset \mathcal{Y}} \det(L_B) = \det(I^{\overline{A}} + L)$. Then we obtain $\mathbb{P}(Y = A) = (-1)^{|A|} \det(I^{\overline{A}} - K)$.

■

We have by definition $\mathbb{P}(A \subset Y) = \det(K_A)$ for all A, and as a consequence $\mathbb{P}(B \cap Y = \emptyset) = \det((I - K)_B)$ for all B. The next proposition gives for any DPP the expression of the general marginal $\mathbb{P}(A \subset Y, B \cap Y = \emptyset)$, for any A, B disjoint subsets of $\mathcal{Y}$, using K. In what follows, H^B denotes the symmetric positive semi-definite matrix

$$H^B = K + K_{\mathcal{Y} \times B}((I - K)_B)^{-1} K_{B \times \mathcal{Y}}.$$

Theorem 2 (General Marginal of a DPP) *Let $A, B \subset \mathcal{Y}$ be disjoint. If $\mathbb{P}(B \cap Y = \emptyset) = \det((I-K)_B) = 0$, then $\mathbb{P}(A \subset Y, B \cap Y = \emptyset) = 0$. Otherwise, the matrix $(I-K)_B$ is invertible and*

$$\mathbb{P}(A \subset Y, B \cap Y = \emptyset) = \det((I-K)_B)\det(H^B_A).$$

Proof Let $A, B \subset \mathcal{Y}$ disjoint such that $\mathbb{P}(B \cap Y = \emptyset) \neq 0$. Using the previous proposition,

$$\mathbb{P}(A \subset Y, B \cap Y = \emptyset) = \sum_{A \subset C \subset \overline{B}} \mathbb{P}(Y = C) = \sum_{A \subset C \subset \overline{B}} (-1)^{|C|}\det(I^{\overline{C}} - K).$$

For any C such that $A \subset C \subset \overline{B}$, one has $B \subset \overline{C}$. Hence, by reordering the matrix coefficients, and using the Schur's determinant formula,

$$\begin{aligned}\det(I^{\overline{C}} - K) &= \det\begin{pmatrix}(I^{\overline{C}} - K)_B & (I^{\overline{C}} - K)_{B\times\overline{B}} \\ (I^{\overline{C}} - K)_{\overline{B}\times B} & (I^{\overline{C}} - K)_{\overline{B}}\end{pmatrix} = \det\begin{pmatrix}(I-K)_B & -K_{B\times\overline{B}} \\ -K_{\overline{B}\times B} & (I^{\overline{C}} - K)_{\overline{B}}\end{pmatrix} \\ &= \det((I-K)_B)\det((I^{\overline{C}} - H^B)_{\overline{B}}).\end{aligned}$$

Thus, $\mathbb{P}(A \subset Y, B \cap Y = \emptyset) = \det((I-K)_B) \sum_{A \subset C \subset \overline{B}} (-1)^{|C|}\det((I^{\overline{C}} - H^B)_{\overline{B}})$.
According to Kulesza and Taskar (2012a), for all $A \subset \overline{B}$,

$$\sum_{A \subset C \subset \overline{B}} \det(-H^B_C) = \det((I^{\overline{A}} - H^B)_{\overline{B}}).$$

Then, Möbius inversion formula ensures that, $\forall A \subset \overline{B}$,

$$\sum_{A \subset C \subset \overline{B}} (-1)^{|C\setminus A|}\det((I^{\overline{C}} - H^B)_{\overline{B}}) = \det(-H^B_A) = (-1)^{|A|}\det(H^B_A).$$

Hence, $\mathbb{P}(A \subset Y, B \cap Y = \emptyset) = \det((I-K)_B)\det(H^B_A)$. ■

Thanks to this formula, we can explicitly formulate the pointwise conditional probabilities of any DPP.

Corollary 3 (Pointwise conditional probabilities of a DPP) *Let $A, B \subset \mathcal{Y}$ be two disjoint sets such that $\mathbb{P}(A \subset Y,\ B \cap Y = \emptyset) \neq 0$, and let $k \notin A \cup B$. Then,*

$$\mathbb{P}(\{k\} \subset Y | A \subset Y,\ B \cap Y = \emptyset) = \frac{\det(H^B_{A\cup\{k\}})}{\det(H^B_A)} = H^B(k,k) - H^B_{\{k\}\times A}(H^B_A)^{-1}H^B_{A\times\{k\}}. \quad (1)$$

This is a straightforward application of the previous expression and the Schur determinant formula. Now, we have all the necessary expressions for the sequential sampling of a DPP.

3.2 Sequential Sampling Algorithm of a DPP

This sequential sampling algorithm simply consists in using Formula (1) and updating at each step the pointwise conditional probability, knowing the previous selected points. It is presented in Algorithm 2. We recall that this sequential algorithm is a first step to develop a competitive sampling algorithm for DPPs : with this method, one doesn't need eigendecomposition anymore. The second step (Section 4) will be to reduce its computational cost.

Algorithm 2 Sequential sampling of a DPP with kernel K

- Initialization: $A \leftarrow \emptyset$, $B \leftarrow \emptyset$.
- For $k = 1$ to N:

1. Compute $H^B_{A\cup\{k\}} = K_{A\cup\{k\}} + K_{A\cup\{k\}\times B}((I-K)_B)^{-1}K_{B\times A\cup\{k\}}$.
2. Compute the probability p_k given by

$$p_k = \mathbb{P}\left(\{k\} \subset Y | A \subset Y,\ B \cap Y = \emptyset\right) = H^B(k,k) - H^B_{\{k\}\times A}(H^B_A)^{-1}H^B_{A\times\{k\}}.$$

3. With probability p_k, k is included, $A \leftarrow A \cup \{k\}$, otherwise $B \leftarrow B \cup \{k\}$.

- Return A.

The main operations of Algorithm 2 involve solving linear systems related to $(I-K)_B^{-1}$. Fortunately, here we can use the Cholesky factorization, which alleviates the computational cost. Suppose that L^B is the Cholesky factorization of $(I-K)_B$, that is, L^B is a lower triangular matrix such that $(I-K)_B = L^B(L^B)^*$ (where $(L^B)^*$ is the conjugate transpose of L^B). Then, denoting $J^B = (L^B)^{-1}K_{B\times A\cup\{k\}}$, one simply has $H^B_{A\cup\{k\}} = K_{A\cup\{k\}} + (J^B)^*J^B$.

Besides, at each iteration where B grows, the Cholesky decomposition $L^{B\cup\{k\}}$ of $(I-K)_{B\cup\{k\}}$ can be computed from L^B using standard Cholesky update operations, involving the resolution of only one linear system of size $|B|$. See Appendix B for the details of a typical Cholesky decomposition update.

In comparison with the spectral sampling algorithm of Hough et al. (2006), one requires computations for each site of $\mathcal{Y}$, and not just one for each sampled point of Y. We will see at the end of Section 4 and in the experiments that it is not competitive.

4. Sequential Thinning Algorithm

In this section, we show that we can significantly decrease the number of steps and the running time of Algorithm 2 : we propose to first sample a point process X containing Y, the desired DPP, and then make a sequential selection of the points of X to obtain Y. This procedure can be called a sequential thinning.

4.1 General Framework of Sequential Thinning

We first describe a general sufficient condition for which a target point process Y - it will be a determinantal point process in our case - can be obtained as a sequential thinning of a point process X. This is a discrete adaptation of the thinning procedure on the continuous line of Rolski and Szekli (1991). To do this, we will consider a coupling (X, Z) such that $Z \subset X$ will be a random selection of the points of X and that will have the same distribution as Y. From this point onward, we identify the set X with the vector of size N with 1 in the place of the elements of X and 0 elsewhere, and we use the notations $X_{1:k}$ to denote the vector $(X_1, \dots, X_k)$ and $0_{1:k}$ to denote the null vector of size k. We want to define the random vector $(X_1, Z_1, X_2, Z_2, \dots, X_N, Z_N) \in \mathbb{R}^{2N}$ with the following conditional distributions for X_k and Z_k:

$$\begin{cases} \mathbb{P}(X_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) = \mathbb{P}(X_k = 1 | X_{1:k-1} = x_{1:k-1}) \\ \\ \mathbb{P}(Z_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k} = x_{1:k}) = \mathbb{1}_{\{x_k = 1\}} \dfrac{\mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1})}{\mathbb{P}(X_k = 1 | X_{1:k-1} = x_{1:k-1})}. \end{cases} \tag{2}$$

Proposition 4 (Sequential thinning) *Assume that X, Y, Z are discrete point processes on $\mathcal{Y}$ that satisfy for all $k \in \{1, \dots, N\}$, and all z, $x \in \{0,1\}^N$,*

$$\begin{gathered} \mathbb{P}(Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) > 0 \\ \textit{implies} \\ \mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1}) \leq \mathbb{P}(X_k = 1 | X_{1:k-1} = x_{1:k-1}). \end{gathered} \tag{3}$$

Then, it is possible to choose (X, Z) in such a way that (2) is satisfied. In that case, we have that Z is a thinning of X, that is $Z \subset X$, and Z has the same distribution as Y.

Proof Let us first discuss the definition of the coupling (X, Z). Thanks to the conditions (3), the ratios defining the conditional probabilities of Equation (2) are ensured to be between 0 and 1 (if the conditional events have non zero probabilities). Hence the conditional probabilities permits to construct sequentially the distribution of the random vector $(X_1, Z_1, X_2, Z_2, \dots, X_N, Z_N)$ of length $2N$, and thus the coupling is well-defined. Besides, as Equation 2 is satisfied, $Z_k = 1$ only if $X_k = 1$, so one has $Z \subset X$.

Let us now show that Z has the same distribution as Y. By complementarity of the events $\{Z_k = 0\}$ and $\{Z_k = 1\}$, it is enough to show that for all $k \in \{1, \dots, N\}$, and $z_1, \dots, z_{k-1}$ such that $\mathbb{P}(Z_{1:k-1} = z_{1:k-1}) > 0$,

$$\mathbb{P}(Z_k = 1 | Z_{1:k-1} = z_{1:k-1}) = \mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1}). \tag{4}$$

Let $k \in \{1, \dots, N\}$, $(z_{1:k-1}, x_{1:k-1}) \in \{0,1\}^{2(k-1)}$, such that $\mathbb{P}(Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) > 0$. Since $Z \subset X$, $\{Z_k = 1\} = \{Z_k = 1, X_k = 1\}$. Suppose first that $\mathbb{P}(X_k = 1 | X_1 = x_1, \dots, X_{k-1} = x_{k-1}) \neq 0$. Then

$$\begin{aligned} &\mathbb{P}(Z_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) \\ &= \mathbb{P}(Z_k = 1, X_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) \\ &= \begin{array}{l} \mathbb{P}(Z_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}, X_k = 1) \\ \times \mathbb{P}(X_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) \end{array} \\ &= \mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1}), \text{ by Equations (2).} \end{aligned}$$

If $\mathbb{P}(X_k = 1|X_{1:k-1} = x_{1:k-1}) = 0$, then $\mathbb{P}(Z_k = 1|Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) = 0$ and thanks to (3), $\mathbb{P}(Y_k = 1|Y_{1:k} = z_{1:k}) = 0$. Hence the identity

$$\mathbb{P}(Z_k = 1|Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) = \mathbb{P}(Y_k = 1|Y_{1:k-1} = z_{1:k-1})$$

is always valid. Since the values $x_1, \dots, x_{k-1}$ do not influence this conditional probability, one can conclude that given $(Z_1, \dots, Z_{k-1})$, Z_k is independent of $X_1, \dots, X_{k-1}$, and thus (4) is true. ■

The characterization of the thinning defined here allows both extreme cases: there can be no pre-selection of points by X, meaning that $X = \mathcal{Y}$ and that the DPP Y is sampled by Algorithm 2, or there can be no thinning at all, meaning that the final process Y can be equal to the dominating process X. Regarding sampling acceleration, a good dominating process X must be sampled quickly and with a cardinal as close as possible to $|Y|$.

4.2 Sequential Thinning Algorithm for DPPs

In this section, we use the sequential thinning approach, where Y is a DPP of kernel K on the groundset $\mathcal{Y}$, and X is a Bernoulli point process (BPP). BPPs are the fastest and easiest point processes to sample. X is a Bernoulli process if the components of the vector $(X_1, \dots, X_N)$ are independent. Its distribution is determined by the probability of occurrence of each point k, that we denote by $q_k = \mathbb{P}(X_k = 1)$. Thanks to the independence property, the conditions (3) simplifies to

$$\mathbb{P}(Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) > 0 \text{ implies } \mathbb{P}(Y_k = 1|Y_{1:k-1} = z_{1:k-1}) \leq q_k.$$

The second inequality does not depend on x, hence it must be valid as soon as there exists a vector x such that $\mathbb{P}(Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) > 0$, that is, as soon as $\mathbb{P}(Z_{1:k-1} = z_{1:k-1}) > 0$. Since we want Z to have the same distribution as Y, we finally obtain the conditions

$$\forall y \in \{0,1\}^N, \quad \mathbb{P}(Y_{1:k-1} = y_{1:k-1}) > 0 \text{ implies } \mathbb{P}(Y_k = 1|Y_{1:k-1} = y_{1:k-1}) \leq q_k.$$

Ideally, we want the q_k to be as small as possible to ensure that the cardinal of X is as small as possible. So we look for the optimal values q_k^*, that is,

$$q_k^* = \max_{\substack{(y_{1:k-1}) \in \{0,1\}^{k-1} \text{ s.t.} \\ \mathbb{P}(Y_{1:k-1} = y_{1:k-1}) > 0}} \mathbb{P}(Y_k = 1|Y_{1:k-1} = y_{1:k-1}).$$

A priori, computing q_k^* would raise combinatorial issues. However, thanks to the repulsive nature of DPPs, we have the following proposition.

Proposition 5 *Let $A, B \subset \mathcal{Y}$ be two disjoint sets such that $\mathbb{P}(A \subset Y,\ B \cap Y = \emptyset) \neq 0$, and let $k \neq l \in \overline{A \cup B}$. If $\mathbb{P}(A \cup \{l\} \subset Y,\ B \cap Y = \emptyset) > 0$, then*

$$\mathbb{P}(\{k\} \subset Y|A \cup \{l\} \subset Y,\ B \cap Y = \emptyset) \leq \mathbb{P}(\{k\} \subset Y|A \subset Y,\ B \cap Y = \emptyset).$$

If $\mathbb{P}(A \subset Y,\ (B \cup \{l\}) \cap Y = \emptyset) > 0$, *then*

$$\mathbb{P}(\{k\} \subset Y | A \subset Y,\ (B \cup \{l\}) \cap Y = \emptyset) \geq \mathbb{P}(\{k\} \subset Y | A \subset Y,\ B \cap Y = \emptyset).$$

Consequently, for all $k \in \mathcal{Y}$, *if* $y_{1:k-1} \leq z_{1:k-1}$ *(where* $\leq$ *stands for the inclusion partial order) are two states for* $Y_{1:k-1}$, *then*

$$\mathbb{P}(Y_k = 1 | Y_{1:k-1} = y_{1:k-1}) \geq \mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1}).$$

In particular, $\forall k \in \{1, \ldots, N\}$, *if* $\mathbb{P}(Y_{1:k-1} = 0_{1:k-1}) > 0$ *then*

$$q_k^* = \mathbb{P}(Y_k = 1 | Y_{1:k-1} = 0_{1:k-1}) = K(k,k) + K_{k \times \{1:k-1\}}((I-K)_{\{1:k-1\}})^{-1} K_{\{1:k-1\} \times k}.$$

Proof Recall that by Proposition 3, $P(\{k\} \subset Y | A \subset Y,\ B \cap Y = \emptyset) = H^B(k,k) - H^B_{\{k\} \times A}(H^B_A)^{-1} H^B_{A \times \{k\}}$. Let $l \notin A \cup B \cup \{k\}$. Consider L^B the Cholesky decomposition of the matrix H^B obtained with the following ordering the coefficients: A, l, the remaining coefficients of $\mathcal{Y} \setminus (A \cup \{l\})$. Then, the restriction L^B_A is the Cholesky decomposition (of the reordered) H^B_A and thus

$$H^B_{\{k\} \times A}(H^B_A)^{-1} H^B_{A \times \{k\}} = H^B_{\{k\} \times A}(L^B_A (L^B_A)^*)^{-1} H^B_{A \times \{k\}} = \|(L^B_A)^{-1} H^B_{A \times \{k\}}\|_2^2.$$

Similarly,

$$H^B_{\{k\} \times A \cup \{l\}}(H^B_{A \cup \{l\}})^{-1} H^B_{A \cup \{l\} \times \{k\}} = \|(L^B_{A \cup \{l\}})^{-1} H^B_{A \cup \{l\} \times \{k\}}\|_2^2.$$

Now remark that solving the triangular system with $b = (L^B_{A \cup \{l\}})^{-1} H^B_{A \cup \{l\} \times \{k\}}$ amounts solving the triangular system with $(L^B_A)^{-1} H^B_{A \times \{k\}}$ and an additional line at the bottom. Hence, one has $\|b\|_2^2 \geq \|(L^B_A)^{-1} H^B_{A \times \{k\}}\|_2^2$. Consequently, provided that $\mathbb{P}(A \cup \{l\} \subset Y,\ B \cap Y = \emptyset) > 0$,

$$\mathbb{P}(\{k\} \subset Y | A \cup \{l\} \subset Y,\ B \cap Y = \emptyset) \leq \mathbb{P}(\{k\} \subset Y | A \subset Y,\ B \cap Y = \emptyset).$$

The second inequality is obtained by complementarity in applying the above inequality to the DPP $\overline{Y}$ with $B \cup \{l\} \subset \overline{Y}$ and $A \cap \overline{Y} = \emptyset$. ■

As a consequence, an admissible choice for the distribution of the Bernoulli process is

$$q_k = \begin{cases} \mathbb{P}(Y_k = 1 | Y_{1:k-1} = 0_{1:k-1}) & \text{if } \mathbb{P}(Y_{1:k-1} = 0_{1:k-1}) > 0, \\ 1 & \text{otherwise.} \end{cases} \tag{5}$$

Remark that if for some index k, $\mathbb{P}(Y_{1:k-1} = 0_{1:k-1}) > 0$ is not satisfied, then for all the subsequent indexes $l \geq k$, $q_l = 1$, that is the Bernoulli process becomes degenerate and contains all the points after k. In the remaining of this section, X will denote a Bernoulli process with probabilities (q_k) given by (5).

As discussed in the previous section, in addition to being easily simulated, one would like the cardinal of X to be close to the one of Y, the final sample. The next proposition shows that this is verified if all the eigenvalues of K are strictly less than 1.

Proposition 6 ($|X|$ is proportional to $|Y|$) *Suppose that $P(Y = \emptyset) = \det(I - K) > 0$ and denote by $\lambda_{\max}(K) \in [0, 1)$ the maximal eigenvalue of K. Then,*

$$\mathbb{E}(|X|) \leq \left(1 + \frac{\lambda_{\max}(K)}{2\,(1 - \lambda_{\max}(K))}\right) \mathbb{E}(|Y|). \tag{6}$$

Proof By Proposition 3, $q_k = K(k,k) + K_{\{k\}\times\{1:k-1\}}((I-K)_{\{1:k-1\}})^{-1} K_{\{1:k-1\}\times\{k\}}$. Since

$$\|((I-K)_{\{1:k-1\}})^{-1}\|_{\mathcal{M}_{k-1}(\mathbb{C})} = \tfrac{1}{1-\lambda_{\max}(K_{\{1:k-1\}})}$$

and $\lambda_{\max}(K_{\{1:k-1\}}) \leq \lambda_{\max}(K)$ one has,

$$K_{\{k\}\times\{1:k-1\}}((I-K)_{\{1:k-1\}})^{-1} K_{\{1:k-1\}\times\{k\}} \leq \tfrac{1}{1-\lambda_{\max}(K)} \|K_{\{1:k-1\}\times\{k\}}\|_2^2.$$

Summing all these inequalities gives $\mathbb{E}(|X|) \leq \operatorname{Tr}(K) + \frac{1}{1-\lambda_{\max}(K)} \sum_{k=1}^{N} \|K_{\{1:k-1\}\times\{k\}}\|_2^2$. The last term is the Frobenius norm of the upper triangular part of K, hence in can be bounded by $\frac{1}{2}\|K\|_F^2 = \frac{1}{2}\sum_{j=1}^{N} \lambda_j(K)^2$. Since $\lambda_j(K)^2 \leq \lambda_j(K)\lambda_{\max}(K)$, $\sum_{j=1}^{N} \lambda_j(K)^2 \leq \lambda_{\max}(K) \operatorname{Tr}(K) = \lambda_{\max}(K)\mathbb{E}(|Y|)$. ■

We can now introduce the whole sampling algorithm that we call sequential thinning algorithm (Algorithm 3). It presents the different steps of our sequential thinning algorithm to sample a DPP of kernel K. The first step is a preprocess that must be done only once for a given matrix K. Step 2 is trivial and fast. The critical point is to sequentially compute the conditional probabilities $p_k = \mathbb{P}(\{k\} \subset Y | A \subset Y,\ B \cap Y = \emptyset)$ for each point of X. Recall that in Algorithm 2 we use a Cholesky decomposition of the matrix $(I-K)_B$ which is updated by adding a line each time a point is added in B. Here, the inverse of the matrix $(I-K)_B$ is only needed when visiting a point $k \in X$, so one updates the Cholesky decomposition by block, where the new block corresponds to all indices added to B in one iteration (see Appendix B).

4.3 Computational Complexity

Recall that the size of the groundset $\mathcal{Y}$ is N and the size of the final sample is $|Y| = n$. Both algorithms introduced in this paper have running complexities of order $O(N^3)$, as the spectral algorithm. Yet, if we get into the details, the most expensive task in the spectral algorithm is the computation of the eigenvalues and the eigenvectors of the kernel K. As this matrix is Hermitian, the common routine to do so is the reduction of K to some tridiagonal matrix to which the QR decomposition is applied. When N is large, the total number of operations is approximately $\frac{4}{3}N^3$ (Trefethen and Bau, 1997). In Algorithms 2 and 3, one of the most expensive operations is the Cholesky decomposition of several matrices. We recall that the Cholesky decomposition of a matrix of size $N \times N$ costs approximately $\frac{1}{3}N^3$ computations, when N is large (Mayers and Süli, 2003). Concerning the sequential algorithm 2, at each iteration k, the number of operations needed is of order $|B|^2|A| + |B||A|^2 + |A|^3$, where $|A|$ is the number of selected points at step k so it's lower than n, and $|B|$ the number

Algorithm 3 sampling of a DPP by sequential thinning of an adapted Bernoulli process

1. Compute sequentially the probabilities $\mathbb{P}(X_k = 1) = q_k$ of the Bernoulli process X:

- Compute the Cholesky decomposition L of the matrix $I - K$.
- For $k = 1$ to N:
  - If $q_{k-1} < 1$ (with the convention $q_0 = 0$),

$$q_k = K(k,k) + \|L^{-1}_{\{1,\dots,k-1\}} K_{\{1,\dots,k-1\}\times\{k\}}\|_2^2$$

  - Else, $q_k = 1$.

2. Draw the Bernoulli process X. Let $m = |X|$ and $k_1 < k_2 < \cdots < k_m$ be the points of X.

3. Apply the sequential thinning to the points of X:

- Attempt to add sequentially each point of X to Y:
  Initialize $A \leftarrow \emptyset$ and $B \leftarrow \{1, \dots, k_1 - 1\}$
  For $j = 1$ to m
  - If $j > 1$, $B \leftarrow B \cup \{k_{j-1} + 1, \dots, k_j - 1\}$
  - Compute the conditional probability $p_{k_j} = \mathbb{P}(\{k_j\} \subset Y | A \subset Y,\ B \cap Y = \emptyset)$ (see Formula 1).
  - Add k_j to A with probability $\frac{p_{k_j}}{q_{k_j}}$ or to B otherwise.
- Return A.

of not-selected points, bounded by k. Then, when N tends to infinity, the total number of operations in Algorithm 2 is lower than $\frac{n}{3}N^3 + \frac{n^2}{2}N^2 + n^3N$ or $O(nN^3)$, as in general $n \ll N$. Concerning Algorithm 3, the sequential thinning from X, coming from Algorithm 2, costs $O(n|X|^3)$. Recall that $|X|$ is propositional to $|Y| = n$ when the eigenvalues of K are smaller than 1 (see Equation 6) so this step costs $O(n^4)$. Then, the Cholesky decomposition of $I - K$ is the most expensive operation in Algorithm 3 as it costs approximately $\frac{1}{3}N^3$. In this case, the overall running complexity of the sequential thinning algorithm is of order $\frac{1}{3}N^3$, which is 4 times less than the spectral algorithm. When some eigenvalues of K are equal to 1, Equation 6 doesn't stand anymore so, in that case, the running complexity of Algorithm 3 is only bounded by $O(nN^3)$.

We will retrieve this experimentally as, depending on the application or on the kernel K, this Algorithm 3 is able to speed up the sampling of DPPs.

5. Experiments

For the following experiments, we ran the algorithms on a laptop HP Intel(R) Core(TM) i7-6600U CPU and the software Matlab. First, let us compare the sequential thinning Algorithm 3 presented here with the two main sampling algorithms: the classic spectral

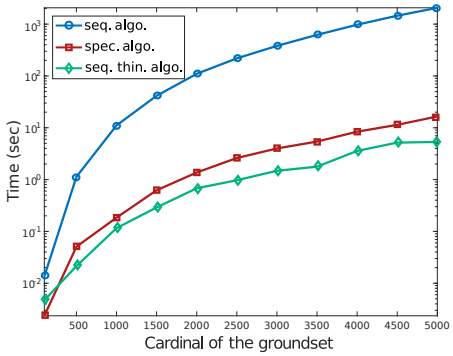

Figure 1: Running times of the 3 studied algorithms in function of the size of the groundset.

Algorithm 1 and the "naive" sequential Algorithm 2. Figure 1 presents the running times of the three algorithms as a function of the total number of points of the groundset. Here, we have chosen a common DPP kernel, a discrete adaptation of the Ginibre kernel. The expected cardinal $\mathbb{E}(|Y|)$ is constant, equal to 20. As foreseen, the sequential algorithm (Algorithm 2) is far slower than the two others. Whatever the chosen kernel and the expected cardinal of the DPP, this algorithm is not competitive. Note that the sequential thinning algorithm uses this sequential method after sampling the particular Bernoulli process. But we will see that this first dominating step can be very efficient and lead to a relatively fast algorithm.

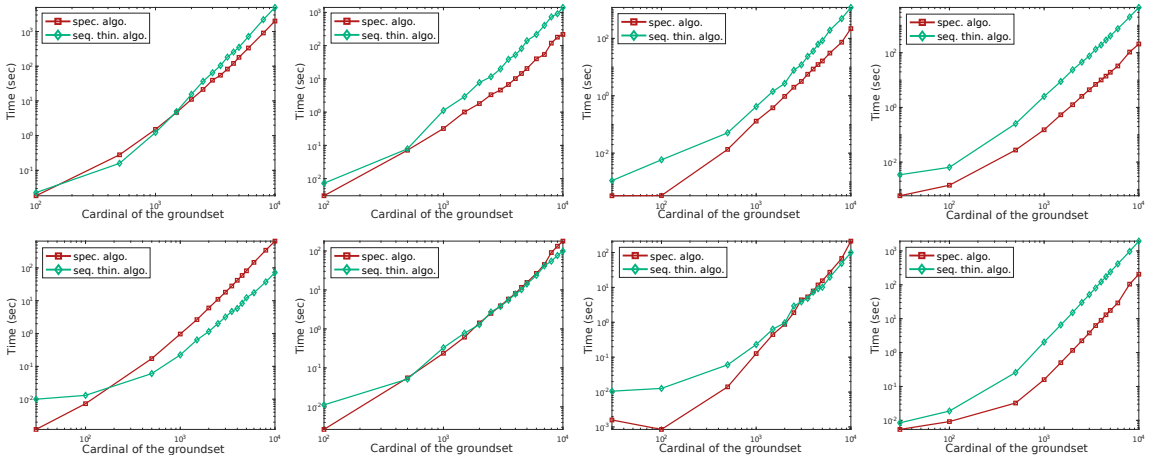

Figure 2: Running times in log-scale of the spectral and the sequential thinning algorithms as a function of the size of the groundset $|\mathcal{Y}|$, using "classic" DPP kernels. From left to right: a random kernel, a Ginibre kernel, a kernel based on the similarity between patches of an image and a projection kernel. On the first row, the expectation of the number of points is set to 4% of the $|\mathcal{Y}|$ and on the second row, $\mathbb{E}(|Y|)$ is constant, equal to 20.

From now on, we restrict the comparison to the spectral and the sequential thinning algorithms (Algorithms 1 and 3). We present in Figure 2 the running times of these algorithms as a function of the size of $|\mathcal{Y}|$ in several situations. The first row shows the running times when the expectation of the number of sampled point $\mathbb{E}(|Y|)$ is equal to 4% of the size of $\mathcal{Y}$: it increases as the total number of points increases. In this case, we can see that whatever the chosen kernel, the spectral algorithm is faster as the complexity of sequential part of Algorithm 3 depends on the size $|X|$ that also grows since $X \subset Y$. On the second row, as $|\mathcal{Y}|$ grows, $\mathbb{E}(|Y|)$ is fixed to 20. Except for the right-hand-side kernel, we are in the configuration where $|X|$ stays proportional to $|Y|$, then the Bernoulli step of Algorithm 3 is very efficient and this sequential thinning algorithm becomes competitive with the spectral algorithm. For these general kernels, we observe that the sequential thinning algorithm can be faster than the spectral algorithm, when the expected cardinal of the sample is small compared to the size of the groundset. The question is : up to which expected cardinal is Algorithm 3 faster ?

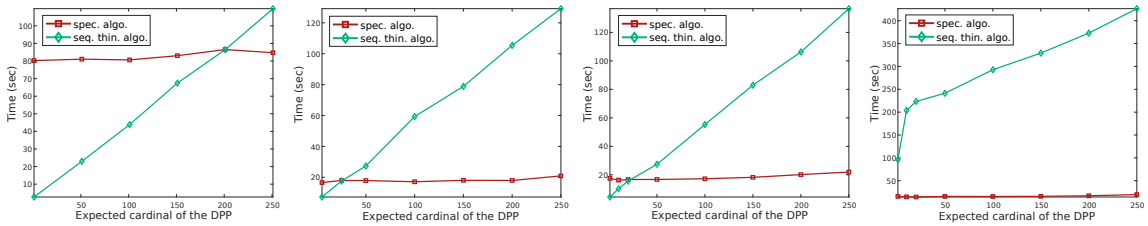

Figure 3: Running times of the spectral and sequential thinning algorithms in function of the expectation of the cardinal of the process. From left to right, using a random kernel, the Ginibre kernel, a kernel based on the similarity between patches of an image and a projection kernel. The size of the groundset is fixed to 5000 in all examples.

Figure 3 displays the running times of both algorithms in function of the expected cardinal of the sample when the size of the groundset is constant, equal to 5000 points. Notice that, concerning the three left-hand-side general kernels with no eigenvalue equal to 1, the sequential thinning algorithm is faster under a certain expected number of points -which depends on the kernel. For instance, when the kernel is randomly defined and the range of desired points to sample is below 200, it is relevant to use this algorithm. To conclude, when the eigenvalues of the kernel are below 1, Algorithm 3 seems relevant for large data sets but small samples. This case is quite common, for instance to summarize a text, to work only with representative points in clusters or to denoise an image with a patch-based method.

The projection kernel (when the eigenvalues of K are either 0 or 1) is, as expected, a complicated case. Figure 2 (bottom, right) shows that our algorithm is not competitive when using this kernel. Indeed, the cardinal of the dominating Bernoulli process X can be very large. In this case, the bound in Equation 6 isn't valid (and even tends to infinity) as $\lambda_{\max} = 1$, and we can quickly reach the degenerated case when, after some index k, all the Bernoulli probabilities $q_l, l \geq k$, are equal to 1. Then the second part of the sequential thinning algorithm -the sequential sampling part- is done on a larger set which significantly increases the running time of our algorithm. Figure 3 confirms this observation as in that

Algorithms	Steps	Expected cardinal	
		4% of $\lvert\mathcal{Y}\rvert$	Constant (20)
Sequential	Matrix inversion	$85,31\%$	$85,38\%$
	Cholesky computation	$13,37\%$	$12,82\ \%$
Spectral	Eigendecomposition	$91,49\%$	$98,82\%$
	Sequential sampling	$8,15\%$	$0,783\%$
Sequential thinning	Preprocess to define q	$5,10\%$	$18,40\%$
	Sequential sampling	$94,86\%$	$81,55\%$

Table 1: Mean of the detailed running times of the sequential, spectral and sequential thinning algorithms with $\lvert\mathcal{Y}\rvert \in [100, 10000]$ and a Ginibre kernel.

configuration, the sequential thinning algorithm is never the fastest.

Table 1 presents the individual weight of the main steps of the three algorithms. Concerning the sequential algorithm, logically, the matrix inversion is the heaviest part taking 85.31% of the global running time. These proportions remain the same when the number of datapoints N grows. The main operation of the spectral algorithm is by far the eigendecomposition of the matrix K, counting for at least 90% of the global running time, when the expectation of the number of points to sample evolves with the size of $\mathcal{Y}$. Finally, the sequential sampling is the heaviest step of the sequential thinning algorithm. We have already mentioned that the thinning is very fast and that it produces a point process with a cardinal as close as possible to the final DPP. When the expected cardinal is low, the number of selected points by the thinning process is low too, so the sequential sampling part remains bounded (81.55% when the expected cardinal $\mathbb{E}(|Y|)$ is constant). On the contrary, when $\mathbb{E}(|Y|)$ grows, the number of points selected by the dominated process rises as well so the running time of this step is growing (with a mean of 94.86%). As seen before, the global running time of the sequential thinning algorithm really depends on how good the domination is.

6. Discussion

In this paper, we proposed a new sampling algorithm adapted to general determinantal point processes, which doesn't use the spectral decomposition of the kernel and which is exact. It proceeds in two phases. The first one samples a Bernoulli process whose distribution is adapted to the targeted DPP. It is a fast and efficient step that reduces the initial number of points of the groundset. We know that if $(I - K)$ is invertible, the expectation of the cardinal of the Bernoulli process is proportional to the expectation of the cardinal of the DPP. Moreover, even if this first sampling procedure may become degenerate (as soon as there is k such that $\mathbb{P}(X_k = 1) = 1$, all the subsequent points are selected), one can always change the visiting order to decrease the expected cardinal of the Bernoulli process. In practice, even when $(I-K)$ isn't invertible, the cardinal of the Bernoulli process remains low in comparison with the cardinal of the groundset. The second phase is a sequential sampling from the points selected in the first step. This phase is made possible thanks to the explicit formulations of the general marginals and the pointwise conditional probabilities of any DPP from its kernel K. Using updated Cholesky decompositions to compute the conditional

probabilities, we fastened the sampling, even if its running time increases significantly with the size of its starting set of points.

In terms of running times, we have detailed the cases for which this algorithm is competitive with the spectral algorithm, in particular when the size of the groundset is high and the expected cardinal of the DPP is modest. This framework is common in machine learning applications. Indeed, DPPs are an interesting solution to subsample a data set, initialize a segmentation algorithm or summarize an image, examples where the number of datapoints needs to be significantly reduced.

Acknowledgments

This work was supported by grants from Région Ile-de-France.

Appendix A. Möbius inversion formula

Proposition 7 (Möbius inversion formula) *Let V be a finite subset and f and g be two functions defined on the set $\mathcal{P}(V)$ of subsets of V. Then,*

$$\forall A \subset V, \quad f(A) = \sum_{B \subset A} (-1)^{|A \setminus B|} g(B) \quad \Longleftrightarrow \quad \forall A \subset V, \quad g(A) = \sum_{B \subset A} f(B),$$

and

$$\forall A \subset V, \quad f(A) = \sum_{B \supset A} (-1)^{|B \setminus A|} g(B) \quad \Longleftrightarrow \quad \forall A \subset V, \quad g(A) = \sum_{B \supset A} f(B).$$

Proof The first equivalence is proved e.g. in Mumford and Desolneux (2010). The second equivalence corresponds to the first applied to $\widetilde{f}(A) = f(\overline{A})$ and $\widetilde{g}(A) = g(\overline{A})$. You will find more details on this matter in the book of Rota (1964). ■

Appendix B. Cholesky Decomposition Update

To be efficient, the sequential algorithm relies on Cholesky decompositions that are updated step by step to save computations. Let M be a symmetric semi-definite matrix of the form $M = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$ where A and C are square matrices. We suppose that the Cholesky decomposition L_A of the matrix A has already been computed and we want to compute the Cholesky decomposition L_M of M. Then, set

$$V = L_A^{-1} B \quad \text{and} \quad X = C - V^T V = C - B^T A^{-1} B$$

the Schur complement of the block A of the matrix M. Denote by L_X the Cholesky decomposition of X. Then, the Cholesky decomposition of M is given by

$$L_M = \begin{pmatrix} L_A & 0 \\ V^T & L_X \end{pmatrix}.$$

Indeed,

$$L_M L_M^T = \begin{pmatrix} L_A & 0 \\ V^T & L_X \end{pmatrix} \begin{pmatrix} L_A^T & V \\ 0 & L_X^T \end{pmatrix} = \begin{pmatrix} L_A L_A^T & L_A V \\ V^T L_A^T & V^T V + L_X L_X^T \end{pmatrix} = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}.$$

References

R. H. Affandi, A. Kulesza, E. B. Fox, and B. Taskar. Nystrom approximation for large-scale determinantal processes. In *AISTATS*, volume 31 of *JMLR Workshop and Conference Proceedings*, pages 85–98, 2013.

P-O. Amblard, S. Barthelme, and N. Tremblay. Subsampling with k determinantal point processes for estimating statistics in large data sets. In *2018 IEEE workshop on Statistical Signal Processing (SSP 2018)*, Freiburg, Germany, June 2018.

N. Anari, S. Ov. Gharan, and A. Rezaei. Monte carlo markov chain algorithms for sampling strongly rayleigh distributions and determinantal point processes. In *COLT*, volume 49 of *JMLR Workshop and Conference Proceedings*, pages 103–115, 2016.

R. Bardenet, F. Lavancier, X. Mary, and A. Vasseur. On a few statistical applications of determinantal point processes. *ESAIM: Procs*, 60:180–202, 2017. doi: 10.1051/proc/201760180.

V. Brunel, A. Moitra, P. Rigollet, and J. Urschel. Rates of estimation for determinantal point processes. In *COLT*, volume 65 of *Proceedings of Machine Learning Research*, pages 343–345. PMLR, 2017.

C. Dupuy and F. Bach. Learning determinantal point processes in sublinear time. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, pages 244–257, 2018.

M. Gartrell, U. Paquet, and N. Koenigstein. Low-rank factorization of determinantal point processes. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 1912–1918, 2017.

G. Gautier, R. Bardenet, and M. Valko. Zonotope hit-and-run for efficient sampling from projection DPPs. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1223–1232. PMLR, Aug. 2017.

J. Gillenwater, A. Kulesza, and B. Taskar. Discovering diverse and salient threads in document collections. In *EMNLP-CoNLL*, pages 710–720. ACL, 2012.

J. Ginibre. Statistical ensembles of complex: Quaternion, and real matrices. *Journal of Mathematical Physics*, Vol: 6, Mar 1965. doi: 10.1063/1.1704292.

J. B. Hough, M. Krishnapur, Y. Peres, and B. Virág. Determinantal processes and independence. *Probability Surveys*, pages 206–229, 2006.

B. Kang. Fast determinantal point process sampling with application to clustering. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2319–2327. Curran Associates, Inc., 2013.

A. Kulesza and B. Taskar. Structured determinantal point processes. In *NIPS*, pages 1171–1179. Curran Associates, Inc., 2010.

A. Kulesza and B. Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2-3):123–286, 2012a. doi: 10.1561/2200000044.

A. Kulesza and B. Taskar. Learning determinantal point processes. *CoRR*, abs/1202.3738, 2012b.

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

C. Li, S. Jegelka, and S. Sra. Efficient sampling for k-determinantal point processes. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51, pages 1328–1337, Cadiz, Spain, 09–11 May 2016a. PMLR.

C. Li, S. Sra, and S. Jegelka. Fast mixing markov chains for strongly rayleigh measures, dpps, and constrained sampling. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4188–4196. Curran Associates, Inc., 2016b.

D. Mayers and E. Süli. *An introduction to numerical analysis*. Cambridge Univ. Press, Cambridge, 2003.

D. Mumford and A. Desolneux. *Pattern Theory: The Stochastic Analysis of Real-World Signals*. Ak Peters Series. Taylor & Francis, 2010. ISBN 9781568815794.

J. G. Propp and D. B. Wilson. How to get a perfectly random sample from a generic markov chain and generate a random spanning tree of a directed graph. *J. Algorithms*, 27(2): 170–217, 1998.

T. Rolski and R. Szekli. Stochastic ordering and thinning of point processes. *Stochastic Processes and their Applications*, 37(2):299–312, 1991. ISSN 0304-4149. doi: 10.1016/0304-4149(91)90049-I.

G-C. Rota. On the foundations of combinatorial theory i. theory of möbius functions. *Z. Wahrscheinlichkeitstheorie und verw*, 2:340–368, 1964.

A. Scardicchio, C. E. Zachary, and S. Torquato. Statistical properties of determinantal point processes in high dimensional euclidean spaces. *Phys. Rev. E*, 79(4), 2009. doi: 10.1103/PhysRevE.79.041108.

L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics, June 1997. ISBN 0898713617.

N. Tremblay, S. Barthelmé, and P.-O. Amblard. Optimized algorithms to sample determinantal point processes. *CoRR*, abs/1802.08471, 2018a.

N. Tremblay, S. Barthelmé, and P.-O. Amblard. Determinantal point processes for coresets. *CoRR*, abs/1803.08700, 2018b.

C. Zhang, H. Kjellström, and S. Mandt. Balanced mini-batch sampling for SGD using determinantal point processes. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*, Aug. 2017.