Learning Determinantal Point Processes in Sublinear Time

Christophe Dupuy, Francis Bach

Learning Determinantal Point Processes in Sublinear Time

Christophe Dupuy[1,2] and Francis Bach[1,3]

[1]INRIA - Sierra project - team
[2]Technicolor R&D France
[3]Département Informatique de l'École Normale Supérieure Paris

Abstract

We propose a new class of determinantal point processes (DPPs) which can be manipulated for inference and parameter learning in potentially sublinear time in the number of items. This class, based on a specific low-rank factorization of the marginal kernel, is particularly suited to a subclass of continuous DPPs and DPPs defined on exponentially many items. We apply this new class to modelling text documents as sampling a DPP of sentences, and propose a conditional maximum likelihood formulation to model topic proportions, which is made possible with no approximation for our class of DPPs. We present an application to document summarization with a DPP on 2^{500} items.

1 Introduction

Determinantal point processes (DPPs) show lots of promises for modelling diversity in combinatorial problems, e.g., in recommender systems or text processing [Kulesza and Taskar, 2011, Gillenwater et al., 2012a, 2014], with algorithms for sampling [Kang, 2013, Affandi et al., 2013, Li et al., 2016a,b] and likelihood computations based on linear algebra [Mariet and Sra, 2015, Gartrell et al., 2016, Kulesza and Taskar, 2012].

While most of these algorithms have polynomial-time complexity, determinantal point processes are too slow in practice for large number N of items to choose a subset from. Simplest algorithms have cubic running-time complexity and do not scale well to more than $N = 1000$. Some progress has been made recently to reach quadratic or linear time complexity in N when imposing low-rank constraints, for both learning and inference [Mariet and Sra, 2016, Gartrell et al., 2016].

This is not enough, in particular for applications in continuous DPPs where the base set is infinite, and for modelling documents as a subset of all possible sentences: the number of sentences, even taken with a bag-of-word assumption, scales exponentially with the vocabulary size. Our goal in this paper is to design a class of DPPs which can be manipulated (for inference and parameter learning) in potentially sublinear time in the number of items N.

In order to circumvent even linear-time complexity, we consider a novel class of DPPs which relies on a particular low-rank decomposition of the associated positive definite matrices. This corresponds to an embedding of the N potential items in a Euclidean space of dimension V. In order to allow efficient inference and learning, it turns out that a single operation on this embedding is needed, namely the computation of a second-order moment matrix, which would take time (at least) proportional to N if done naively, but may be available in closed form in several situations. This computational trick makes a striking parallel with positive definite kernel methods [Scholkopf and Smola, 2001, Shawe-Taylor and Cristianini, 2004], which use the "kernel trick" to work in very high dimension at the cost of computations in a smaller dimension.

In this paper we make the following contributions:

– We propose in Section 3 a new class of determinantal point processes (DPPs) which is based on a particular low-rank factorization of the marginal kernel. Through the availability of a particular second-

moment matrix, the complexity for inference and learning tasks is polynomial in the rank of the factorization and thus often sublinear in the total number of items (with exact likelihood computations).

– As shown in Section 4, these new DPPs are particularly suited to a subclass of continuous DPPs (infinite number of items), such as on $[0, 1]^m$, and DPPs defined on the V-dimensional hypercube, which has 2^V elements.

– We propose in Section 5 a model of documents as sampling a DPP of sentences, and propose a conditional maximum likelihood formulation to model topic proportions. We present an application to document summarization with a DPP on 2^{500} items.

2 Review of Determinantal Point Processes

In this work, for simplicity, we consider a very large *finite* set $\mathcal{X}$, with cardinality $|\mathcal{X}| = N$, following Kulesza and Taskar [2012]. In several places, we will consider an infinite set (see, e.g., Section 3.4) [Affandi et al., 2013, Lavancier et al., 2015].

A determinantal point process (DPP) on a set $\mathcal{X}$ is a probability measure on $2^{\mathcal{X}}$, the set of all subsets of $\mathcal{X}$. It can either be represented by a L-ensemble $L(x, y)$, for $x, y \in \mathcal{X}$ or by its marginal kernel $K(x, y)$, which we refer to as the "K-representation" and the "L-representation". In this paper, K and L will be $N \times N$ matrices, with elements $K(x, y)$ and $L(x, y)$ for $x, y \in \mathcal{X}$. Both L and K are potentially large matrices, as they are indexed by elements of $\mathcal{X}$.

A sample X drawn from a DPP on $\mathcal{X}$ is a subset of $\mathcal{X}$, $X \subseteq \mathcal{X}$. In the "K-representation" of a DPP, for any set $A \subset \mathcal{X}$, we have:

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

where K_A is the matrix of size $|A| \times |A|$ composed of pairwise evaluations of $K(x, y)$ for $x, y \in A$. If we denote by "$\preccurlyeq$" the positive semidefinite order on symmetric matrices (i.e., $A \preccurlyeq B \Leftrightarrow (B - A)$ is positive semidefinite), the constraint on K is $0 \preccurlyeq K \preccurlyeq I$ so that the DPP is a probability measure.

In the "L-representation", for any set $A \subset \mathcal{X}$, we have

$$\mathbb{P}(X = A) = \frac{\det L_A}{\det(I + L)}. \tag{1}$$

The constraint on L is $L \succcurlyeq 0$.

Given a DPP and its two representations L and K, we can go from L to K as $K = I - (I + L)^{-1}$ and vice-versa as $L = K(I - K)^{-1}$. The L-representation only exists when $K \prec I$, where "$\prec$" denotes the positive definite order of symmetric matrices (i.e., $A \prec B \Leftrightarrow (B - A)$ is positive definite).

Several tasks can be solved, e.g., marginalization, conditioning, etc., that are either easy in the L-representation or in the K-representation. For instance, (conditional) maximum likelihood when observing sets is easier in the L-representation, as the likelihood of an observed set $A \subseteq \mathcal{X}$ is directly obtained with L through Eq. (1). Conversely, the expected number of selected items, $\mathbb{E}[|X|]$ for a DPP defined by L, K is easily computed with K as $\mathbb{E}[|X|] = \operatorname{tr} K$ [Kulesza and Taskar, 2012].

The DPPs model aversion between items. For instance, if X is drawn from a DPP(K, L), the probability that items i and j are together included in X is

$$\mathbb{P}(\{i, j\} \subseteq X) = K_{ii}K_{jj} - (K_{ij})^2.$$

This probability then decreases with similarity K_{ij} between item i and item j. This key aversion property makes DPPs useful to document summarization (our application in Section 5) where we want to select sentences that covers the most the document while avoiding the selection of two similar sentences.

Approximate computations. In practice, the key difficulty is to deal with the cubic complexity in $|\mathcal{X}|$ of the main operations — determinant and computations of inverses. In their work, Kulesza and Taskar [2012] propose a low-rank model for the DPP matrix L, namely $L(x,y) = q(x)\langle\phi(x),\phi(y)\rangle q(y)$, where $q(x) \in \mathbb{R}^+$ corresponds to a "quality" measure of x and $\phi(x) \in \mathbb{R}^r$, $\|\phi(x)\| = 1$ corresponds to the "diversity" feature (or embedding) of x. In matrix notations, we have $L = \mathrm{Diag}(q)\Phi\Phi^\top \mathrm{Diag}(q)$. In particular, they show that most of the computations are based on the matrix $C = \Phi^\top \mathrm{Diag}(q^2)\Phi \in \mathbb{R}^{r\times r}$. As $\Phi \in \mathbb{R}^{N\times r}$, they achieve an overall complexity $O(Nr^2)$. In their application to document summarization, they only parameterize and learn the "quality" vector q, fixing the diversity features Φ.

More recently, Gartrell et al. [2016] use a low rank factorization of L ($L = UU^\top$, with $U \in \mathbb{R}^{N\times r}$) and apply accelerated stochastic gradient ascent on the log-likelihood of observed sets for learning U. They achieve a linear complexity in N: $O(Nr^2)$. Mariet and Sra [2016] propose a Kronecker factorization of L: $L = L_1 \otimes L_2$ where $\otimes$ is the Kronecker product, $L_i \in \mathbb{R}^{N_i\times N_i}$ and $N = N_1N_2$. They use a fixed point mehtod (with the Picard iteration) to maximize the likelihood, that consists in alternatively updating L_1 and L_2 with a computational complexity $O(N^{3/2})$ if $N_1 \approx N_2 \approx \sqrt{N}$.

However, when the set $\mathcal{X}$ is very large (e.g., exponential) or infinite, even linear operations in $N = |\mathcal{X}|$ are intractable. In the next sections, we provide a representation of the matrices L and K together with an optimization scheme that makes the optimization of the likelihood tractable even when the set $\mathcal{X}$ is too large to perform linear operations in $N = |\mathcal{X}|$.

3 A Tractable Family of Kernels

We consider the family of matrices decomposed as a sum of the identity matrix plus a low-rank term, where the column-space of the low-rank term is fixed. We show that if K is in the family (with its additional constraint that $K \prec I$), so is L, and vice-versa.

3.1 Low-rank family

We consider a feature map $\phi : \mathcal{X} \to \mathbb{R}^V$, a probability mass function $p : \mathcal{X} \to \mathbb{R}_+$, a scalar $\sigma \in \mathbb{R}_+$ and a symmetric matrix $B \in \mathbb{R}^{V\times V}$. The kernel $K(x,y)$ is constrained to be of the form:

$$K(x,y) = \sigma 1_{x=y} + p(x)^{1/2}\phi(x)^\top B\phi(y)p(y)^{1/2}. \tag{2}$$

In matrix notation, this corresponds to

$$K = \sigma I + \mathrm{Diag}(p)^{1/2}\Phi B\Phi^\top \mathrm{Diag}(p)^{1/2},$$

with $\Phi \in \mathbb{R}^{|\mathcal{X}|\times V}$ and $B \in \mathbb{R}^{V\times V}$. With additional constraints detailed below, this defines a valid DPP, for which the L-representation can be easily derived in the form

$$L(x,y) = \alpha 1_{x=y} + p(x)^{1/2}\phi(x)^\top A\phi(y)p(y)^{1/2}, \tag{3}$$

with $\alpha \in \mathbb{R}_+$ and $A \in \mathbb{R}^{V\times V}$. The following proposition is a direct consequence of the Woodbury matrix identity:

Proposition 1 *The kernel K defined in Eq. (2) is a valid DPP if*
(a) $\sigma \in [0,1]$,
(b) $0 \preccurlyeq B \preccurlyeq (1-\sigma)(\Phi^\top \mathrm{Diag}(p)\Phi)^{-1}$.
It corresponds to the matrix L defined in Eq. (3) with $\sigma = \frac{\alpha}{\alpha+1}$ *and*

$$B = \frac{1}{(\alpha+1)^2}\big[A^{-1} + \frac{1}{\alpha+1}\Phi^\top \mathrm{Diag}(p)\Phi\big]^{-1}.$$

Moreover, we may use the matrix determinant lemma to obtain

$$\det(L+I) \quad = \quad \det[(\alpha+1)I]\det[A]\det\Big(A^{-1}+\frac{1}{\alpha+1}\Phi^\top \operatorname{Diag}(p)\Phi\Big),$$

which is expressed through the determinant of a $V \times V$ matrix (instead of $N \times N$).

We may also go from L to K as $\alpha = \frac{\sigma}{1-\sigma}$ and $A = \frac{1}{(1-\sigma)^2}\big[B^{-1} - \frac{1}{1-\sigma}\Phi^\top \operatorname{Diag}(p)\Phi\big]^{-1}$.

3.2 Tractability

From the identities above, we see that a sufficient condition for being able to perform the computation of L and its determinant is the availability of the matrix

$$\Sigma = \Phi^\top \operatorname{Diag}(p)\Phi = \sum_{x\in\mathcal{X}} p(x)\phi(x)\phi(x)^\top \in \mathbb{R}^{V\times V}.$$

In the general case, computing such an expectation would be (at least) linear time in N, but throughout the paper, we assume this is available in polynomial time in V (and not in N). See examples in Section 4.

Note that this resembles the kernel trick, as we are able to work implicitly in a Euclidean space of dimension N while paying a cost proportional to V. In our document modelling example, we will have $N = 2^V$, and hence we achieve sublinear time.

We can compute other statistics of the DPP when K, L belong to the presented family of matrices. For instance, the expected size of a set X drawn from the DPP represented by K, L is:

$$\mathbb{E}[|X|] = \operatorname{tr} K = \sigma|\mathcal{X}| + \operatorname{tr}(B\Sigma). \tag{4}$$

Given $\phi(x)$, the parameters are the distribution $p(x)$ on $\mathcal{X}$, $A \in \mathbb{R}^{V\times V}$ and $\alpha \in \mathbb{R}_+$. If $\mathcal{X}$ is very large, it is hard to learn $p(x)$ from observations and $p(x)$ is thus assumed fixed. We also have to assume that α is proportional to $1/|\mathcal{X}|$ or zero when $\mathcal{X}$ is infinite as the first term of $\mathbb{E}[|X|]$ in Eq. (4), $\frac{\alpha}{\alpha+1}|\mathcal{X}|$, must be finite.

3.3 Additional low-rank approximation

If V is large, we use a low-rank representation for A:

$$A = \gamma I + U \operatorname{Diag}(\theta) U^\top, \tag{5}$$

with $\gamma \in \mathbb{R}_+$, $\theta \in \mathbb{R}_+^r$ and $U \in \mathbb{R}^{V\times r}$. All the exact operations on L are then linear in V, i.e., as $O(Vr^2)$. See details in Appendix C. Moreover, the parameter θ can either be global or different for each observation, which gives flexibility to the model in the case where observations come from different but related DPPs on $\mathcal{X}$. For instance, in a corpus of documents, the distance between words (conveyed by U) may be different from one document to another (e.g., *field* and *goal* may be close in a sport context, not necessary in other contexts). This can be modelled through θ as topic proportions of a given document (see Section 5).

Note that the additional low-rank assumption (5) corresponds to an embedding $x \in \mathcal{X} \mapsto U^\top\phi(x) \in \mathbb{R}^r$, where $\phi(x)$ is fixed and U is learned.

3.4 Infinite $\mathcal{X}$

Although we avoid dealing rigorously with continuous-state DPPs in this paper [Affandi et al., 2013, 2014, Bardenet and Titsias, 2015], we note that when dealing with exponentially large finite sets $\mathcal{X}$ or infinite sets, we need to set $\sigma = \alpha = 0$ to avoid infinite (or too large) expectations for the numbers of sampled elements (which we use in experiments for $N = 2^V$).

Note moreover, that in this situation, we have the kernel $K(x,y) = p(x)^{1/2}p(y)^{1/2}\phi(x)^\top B\phi(y)$ and thus the rank of the matrix K is at most V, which implies that the number of sampled elements has to be less than V.

We can sample from the very large DPP as soon as we can sample from a distribution on $\mathcal{X}$ with density proportional to $p(x)\phi(x)^\top A\phi(x)$. Indeed, one can sample from a DPP by first selecting the eigenvectors of L, each with probability $\lambda_i/(\lambda_i+1)$—where the λ_is are the eigenvalues of L—and then projecting the canonical basis vectors—one per item—on this subset of eigenvectors. The density for selecting the first item is proportional to the squared norm of the latter projection (see Algorithm 1 of Kulesza and Taskar [2012] for more details). Given our fomula for L, all the required densities can be expressed as being proportional to $p(x)\phi(x)^\top A\phi(x)$. In our simulations, we use instead a discretized scheme.

3.5 Learning parameters with maximum likelihood

In this section, we present how to learn the parameters of the model, corresponding to the matrix A.

We have access to the likelihood through observations. We denote by $X_1, \dots, X_M$, with $X_i \subseteq \mathcal{X}$, the observations drawn from a density $\mu(X)$. Each set X_i is a set of elements $X_i = \{x_1^i, \dots, x_{|X_i|}^i\}$, with $x_j^i \in \mathcal{X}$. We denote by $\ell(X|L)$ the log-likelihood of a set X given a DPP matrix L. Our goal is to maximize the expected log-likelihood under μ, i.e., $\mathbb{E}_{\mu(X)}[\ell(X|L)]$. As we only have access to μ through observations, we maximize an estimation of $\mathbb{E}_{\mu(X)}[\ell(X|L)]$, i.e., $\mathcal{L}(L) = \frac{1}{M}\sum_{i=1}^M \ell(X_i|L)$. As the log-likelihood of a set X is $\ell(X|L) = \log\det L_X - \log\det(L+I)$, our objective function becomes:

$$\mathcal{L}(L) = \frac{1}{M}\sum_{i=1}^{M} \left(\log\det L_{X_i} - \log\det(I+L)\right). \tag{6}$$

In the following, we assume p fixed and we only learn A in its form (5).

In practice we minimize a penalized objective, that is, for our parameterization of A in Eq. (5),

$$F(L) = -\mathcal{L}(L) + \lambda\mathcal{R}(U,\theta),$$

where $\mathcal{L}$ is the log-likelihood of a train set of observations [Eq. (6)] and $\mathcal{R}$ is a penalty function. We choose the penalty $\mathcal{R}(U,\theta) = \|\theta\|_1 + \|U\|_{1,2}^2$ where $\|.\|_1$ is the ℓ^1 norm and $\|U\|_{1,2} = \sum_{i=1}^r \|u_i\|_2$, where $\|.\|_2$ is the ℓ^2 norm and u_i is the i^{th} column of U. The group sparsity norm $\|.\|_{1,2}^2$ allows to set columns of U to zero and thus learn the number of columns.

This is a non non-convex problem made non smooth by the group norm. Following Lewis and Overton [2013], we use BFGS to reach a local optimum of our objective function.

4 Examples

In this section, we review our three main motivating examples: (a) orthonormal basis based expansions applicable to continuous space DPPs; (b) standard orthonomal embedding with $\mathcal{X} = \{1, \dots, N\}$ and (c) exponential set $\mathcal{X} = \{0,1\}^V$ for applications to document modelling based on sentences in Section 5.

4.1 Orthonormal basis based expansions

We consider a fixed probability distribution $p(x)$ on $\mathcal{X}$ and an orthonormal basis of the Euclidean space of square integrable (with respect to p) functions on $\mathcal{X}$. We consider $\phi(x)_i$ as the value at x of the i-th basis function. Note that this extends to any $\mathcal{X}$, even not finite by going to Hilbert spaces.

We consider $A = \mathrm{Diag}(a)$ and $B = \mathrm{Diag}(b)$ two diagonal matrices in $\mathbb{R}^{V\times V}$. Since $\phi(x)$ is an orthonormal basis, we have:

$$\Sigma = \sum_{x\in\mathcal{X}} p(x)\phi(x)\phi(x)^\top = I,$$

with a similar result for any subsampling of $\phi(x)$ (that is keeping a subset of the basis vectors).

For example, for $\mathcal{X} = [0,1]$, $p(x)$ the uniform distribution, $\alpha = 0$ and $\phi(x)$ the cosine/sine basis, we obtain the matrix $L(x,y) = \phi(x)^\top \operatorname{Diag}(a)\phi(y)$ which is a 1-periodic function of $x - y$, and we can thus model any of these functions. This extends to $\mathcal{X} = [0,1]^m$ by tensor products, and hyperspheres by using spherical harmonics [Atkinson and Han, 2012].

Truncated Fourier basis. In practice, we consider the truncated Fourier orthonormal basis of $\mathbb{R}^V$ with $V = 2d+1$, that is, $\phi_1(x) = 1$, $\phi_{2i}(x) = \sqrt{2}\cos(2\pi i x)$ and $\phi_{2i+1} = \sqrt{2}\sin(2\pi i x)$, for $i \in \{1,\dots,d\}$ and $x \in \mathcal{X}$. If $A = \operatorname{Diag}(a)$ is diagonal, then $L(x,y) = \phi(x)^\top \operatorname{Diag}(a)\phi(y)$ is a 1-periodic function of $x - y$, with only the first d frequencies, which allows us to learn covariance functions which are invariant by translation in the cube. We could also use the K-representation $K(x,y) = \phi(x)^\top B\phi(y)$, with $B = \operatorname{Diag}(b)$ diagonal in $[0,1]$, but the log-likelihood maximization is easier in the L-representation.

We use this truncated basis to optimize the log-likelihood $\mathcal{L}(L)$ on finite observations [Eq. (6)] , i.e., $X_i \subseteq \mathcal{X}$ and $|X_i| < \infty$. In particular, the normalization constant is computed efficiently with this representation of L as we have $\det(L+I) = \prod_{i=1}^V (a_i + 1)$.

Non-parametric estimation of the stationary function. We may learn any 1-periodic function of $x - y$ for $L(x,y)$ or $K(x,y)$ and we do so by choosing the truncated Fourier basis of size V, we could also use positive definite kernel techniques to perform non-parametric estimation.

Running time complexity. For general continuous ground set $\mathcal{X} = [0,1]^m$, with $m \geq 1$, the running time complexity is still controlled by $V = (2d+1)^m$, d corresponding to the number of selected frequencies in each dimension of the Fourier basis (with $a \in \mathbb{R}^V$). The value of d may be adjusted to fit the complexity in $O(V\kappa^3)$ or $O(d^m\kappa^3)$, where κ is the size of the biggest observation (i.e., the largest cardinality of all observed sets).

4.2 Standard orthonormal embeddings

In this section, we consider DPPs on the set $\mathcal{X} = \{1,\dots,V\}$ (i.e., $N = V$). We choose the standard orthonormal embedding, that is $\Phi = I$ which gives $L = \alpha I + U\operatorname{Diag}(\theta)U^\top$, taking $\gamma = 0$. For this particular model, the complete embedding $\Phi U = U$ is learned and the distribution $p(x)$ is included in U. This is only possible when V is small. This model is suited to item selection, where groups of items are observed (e.g., shopping baskets) and we want to learn underlying embeddings of these items (through parameter U). Again, the size of the catalog V may be very large. Note that unlike existing methods leveraging low-rank representations of DPPs [Mariet and Sra, 2016, Gartrell et al., 2016], the parameter θ in our representation can be different for each observation, which makes our model more flexible.

4.3 $\mathcal{X} = \{0,1\}^V$

In this section, we consider DPPs on the set $\mathcal{X} = \{0,1\}^V$. For large values of V, direct operations on matrices L, K may be impossible as $\mathcal{X}$ is exponential, $|\mathcal{X}| = 2^V$. In particular, we consider the model where $\phi(x) = x$, i.e., $\Phi \in \mathbb{R}^{2^V \times V}$ (in other words we simply embed $\{0,1\}^V$ in $\mathbb{R}^V$).

As mentioned above, the tractability of the DPP(L,K) on $\mathcal{X} = \{0,1\}^V$ depends on the expectation $\Sigma = \sum_{x\in\mathcal{X}} p(x)xx^\top$. For particular distributions $p(x)$, Σ is computed in closed form. For instance, if $p(x)$ corresponds to V independent Bernouillis, i.e., $p(x) = \prod_{i=1}^V \pi_i^{x_i}(1-\pi_i)^{1-x_i}$, the expectation quantity is $\Sigma = \operatorname{Diag}(\pi(1-\pi)) + \pi\pi^\top$. If the independent Bernoullis are exchangeable, i.e., all π's are equal, we have $\Sigma = \pi(1-\pi)I + \pi^2 11^\top$.

The tractability of our model is extended to the case $\mathcal{X} = \mathbb{N}^V$ with Poisson variables. Indeed, if $p(x) = \prod_{i=1}^V \frac{e^{-\lambda_i}\lambda_i^{x_i}}{x_i!}$ for $x \in \mathbb{N}^V$, the expectation of $xx^\top$ over $\mathcal{X}$ is $\Sigma = \operatorname{Diag}(\lambda) + \lambda\lambda^\top$.

Given these examples, we assume in the rest of the paper that:

$$\Sigma = \sum_{x \in \mathcal{X}} p(x) x x^\top = \mathrm{Diag}(\nu) + \mu\mu^\top.$$

The complexity of operations on the matrix L with this structure is $O(Vr^2)$ — instead of $O(2^V r^2)$ if working directly with L; see Appendix C for details. If we use the factorization of Σ above:

$$\begin{aligned} \mathrm{tr}\, K &= \frac{\alpha}{\alpha+1} 2^V \\ &\quad + \frac{1}{(\alpha+1)^2} \mathrm{tr}\left[\left(\mathrm{Diag}(\nu) + \mu\mu^\top \right) \right. \\ &\quad \left. \times \Big(\frac{1}{\alpha+1} (\mathrm{Diag}(\nu) + \mu\mu^\top) + A^{-1} \Big)^{-1} \right]. \end{aligned}$$

This identity suggests that we replace α by $\alpha 2^{-V}$ in order to select a finite set $X \subseteq \mathcal{X}$. For large values of V, $\alpha = 0$ is the key choice to avoid infinite number of selected items.

5 DPP for document summarization

We apply our DPP model to document summarization. Each document X is represented by its sentences, $X = (x_1, \ldots, x_{|X|})$ with $x_i \in \mathcal{X} = \{0,1\}^V$. The variable V represents the size of the vocabulary, i.e., the number of possible words. A sentence is then represented by the set of words it contains, ignoring their exact count and the order of the words. We want to extract the summary of each document as a subset of observed sentences. We use the structure described in Section 3.1 to build a generative model of documents. Let $K \in \mathbb{R}^{2^V \times 2^V}$ be the marginal kernel of a DPP on the possible sentences $\mathcal{X}$. We consider that the summary $Y \subseteq \mathcal{X}$ of document X is generated from the DPP(K, L) as follows:

1. Draw sentences $X = (x_1, \ldots, x_{|X|})$ from DPP represented by L,
2. Draw summary $Y \subseteq X$ from DPP represented by L_X.

In practice, we observe a set of documents and we want to infer the word embeddings U and the topic proportions θ for each document. In the following we consider that α and γ are fixed. We also denote by $\mathcal{L}(U, \theta) \equiv \mathcal{L}(L)$ the log-likelihood of observations [Eq. (6)] for simplicity as our DPP matrix L is encoded by U and θ.

The intuition behind this generative model is that the sentences of a document cover a particular topic (the topic proportions are conveyed by the variable θ) and it is very unlikely to find sentences that have the same meaning in the same document. In this sense, we want to model aversion between sentences of a document.

Parameter learning. As explained in Section 4.3, we assume $\Sigma = \mathrm{Diag}(\nu) + \mu\mu^\top$. The log-likelihood of an observed document X is $\ell(X|L) = \log\det L_X - \log\det(L+I)$. The computation of the second term, $\log\det(L+I)$, is untractable to compute in reasonable time for any L when $V \geq 20$, since $L \in \mathbb{R}^{2^V \times 2^V}$. We can still compute this value exactly for structured L coming from our model with complexity $O(Vr^2)$ (see Appendix C for details).

We infer the parameters U and θ by optimizing our objective function $F(U, \theta) = -\mathcal{L}(U, \theta) + \lambda \mathcal{R}(U, \theta)$ with respect to U and θ alternatively. In practice, we perform 100 iterations of L-BFGS for the function $U \mapsto F(U, \theta)$ and 100 iterations of L-BFGS for each function $\theta_i \mapsto F(U, \theta)$, for $i = 1, \ldots, M$. The optimization in U can also be done with stochastic gradient descent (SGD) [Bottou, 1998], using a mini-batch D_t of observations at iteration t: $U \leftarrow U - \rho_t G_{D_t}(U)$, with $G_{D_t}(U)$ the unbiased gradient $G_{D_t}(U) = -\frac{1}{|D_t|} \sum_{i \in D_t} \nabla_U \ell(X_i | L(U, \theta_i) + \lambda \nabla_U \mathcal{R}(U, \theta)$. We choose L-BFGS over SGD for settings simplicity. In particular, the choice of the stepsize ρ and the mini-batch size $|D_t|$ is not straightforward.

6 Experiments

6.1 Datasets

We run experiments on synthetic datasets generated from the different types of DPPs described above. For all the datasets, we generate the observations using the sampling method described by Kulesza and Taskar [2012] (Algorithm 1 page 16) and perform the evaluation for 10 different datasets. This method draws exact samples from a DPP matrix L and its eigendecomposition (which requires N to be less than 1000). For the evaluation figures, the mean and the variance over the 10 datasets are respectively displayed as a line and a shaded area around the mean.

Continuous set $[0,1]^m$. We describe in Section 4.1 a method to learn from subsets drawn from a DPP on a continuous set $\mathcal{X}$. As sampling from continuous DPPs is not straightforward and approximate [Affandi et al., 2013], we consider a discretization of the set $[0,1]^2$ into the discrete set $\{0, 1/N, \dots, (N-1)/N\}^2$. Note that this discretization only affects the sampling scheme. We generate a dataset from the ground set $\mathcal{X} = \{0, 1/N, \dots, (N-1)/N\}^2$ with the DPP represented by $L(x,y) = \phi(x)^\top \operatorname{Diag}(a)\phi(y)$, with embedding $\phi(x) \in \mathbb{R}^{N^2}$ the discrete Fourier basis of $(\mathbb{R}^N)^2$, i.e., for $(i,j) \in \{1,\dots,N\}^2$, $\phi(x)_{i,j} = \psi(x_1)_i \psi(x_2)_j$ with $\psi(z)_1 = 1$, $\psi(z)_{2k} = \sqrt{2}\cos(2\pi k z)$ and $\psi(z)_{2k+1} = \sqrt{2}\sin(2\pi k z)$, for $k = 1, \dots, (N-1)/2$. With notations of Section 4.1 we have $V = N^2$. For $(i,j) \in \{0,\dots,N-1\}^2$, we set $a_{(i,j)} = C_i C_j \tilde{a}_i \tilde{a}_j$, with $C_0 = 1$, $\tilde{a}_0 = 1$ and $C_i = 1/\sqrt{2}$, $\tilde{a}_i = 1/i^\beta$ for $i \geq 1$. We choose $N = 33$ (i.e., $V = N^2 = 1069$) and $\beta = 2$ for the experiments. We present in Figure 1 two samples: a sample drawn from the DPP described above and a set of points that are i.i.d. samples from the uniform distribution on $\mathcal{X}$. We observe aversion between points of the DPP sample that are distributed more uniformly than points of the i.i.d. sample.

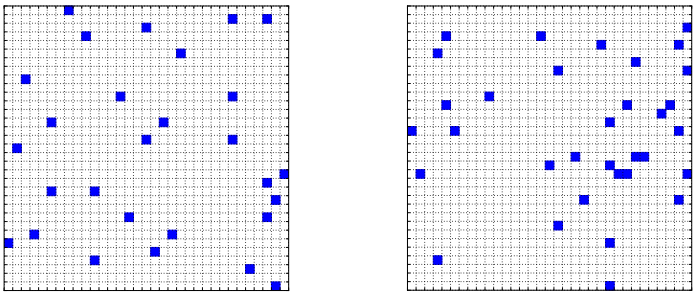
Items set. We generate observations from the ground set $\mathcal{X} = \{1, \dots, V\}$, which corresponds to the matrix $L = \alpha I + U \operatorname{Diag}(\theta) U^\top$. For these observations, we set $V = 100$, $r = 5$, $\alpha = 10^{-5}$. For each dataset, we generate U and θ randomly with different seeds accross the datasets.

Exponential set. We generate observations from the ground set $\mathcal{X} = \{0,1\}^V$ with $\phi(x) = x$. In this case, we set $V = 10$, $r = 2$, $\alpha = 10^{-5}$, $\gamma = 1/V$. As we need the eigendecomposition of $L \in \mathbb{R}^{2^V \times 2^V}$, we could not generate exact samples with higher orders of magnitude for V. However, we can still optimize the likelihood for ground sets with large values of V and we run experiments on real document datasets, where the size of the vocabulary is $V = 500$ (i.e., $|\mathcal{X}| = 2^{500} \approx 10^{150}$).

For both ground sets $\mathcal{X} = \{1, \dots V\}$ and $\mathcal{X} = \{0,1\}^N$, we consider two types of datasets: one dataset where all the observations are generated with the same DPP matrix L and another dataset where observations are generated with a different matrix $L(\theta^i)$ for each observation. For the second type of dataset, the embedding U is common to all the observations while the variable θ^i differs from one observation to another.

Real dataset. We consider a dataset of 100,000 restaurant reviews and minimize the objective function $F(U, \theta)$ mentioned above. We first remove the stopwords using the NLTK toolbox [Bird et al., 2009]. Among the remaining words, we only keep the $V = 500$ most frequent words of the dataset. After filtering, the average number of sentences per review is 10.5 and each sentence contains on average 4.5 words. We use the proposed DPP structure to (1) learn word embedding U from observations and (2) extract a summary for each review using the model of Section 5. Given a document X, the inferred parameters U and $\theta(X)$ and the corresponding DPP matrix L, we extract the l sentences summarizing the document X by solving the following maximization:

$$Y^* \in \arg \max_{Y \subseteq X,\, |Y| = l} \frac{\det(L_Y)}{\det(L_X + I)}.$$



(a) DPP. (b) I.i.d.

Figure 1: Comparison of points drawn from a DPP (left) independently from uniform distribution (right).

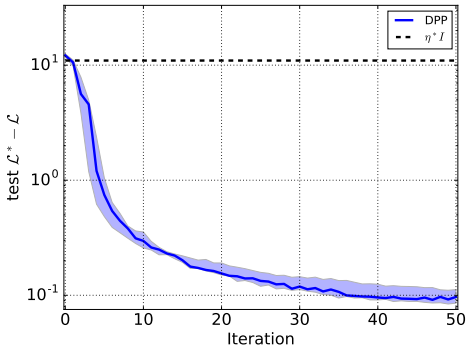


Figure 2: Continuous set $[0, 1]^2$. Distance in log-likelihood ($\mathcal{L}^* - \mathcal{L}$).

In practice we use the greedy MAP algorithm [Gillenwater et al., 2012b] to extract the summary $\widehat{Y}$ of document X, as an approximation of the MAP Y^* (with the usual submodular maximization approximation guarantee [Krause and Golovin, 2012]).

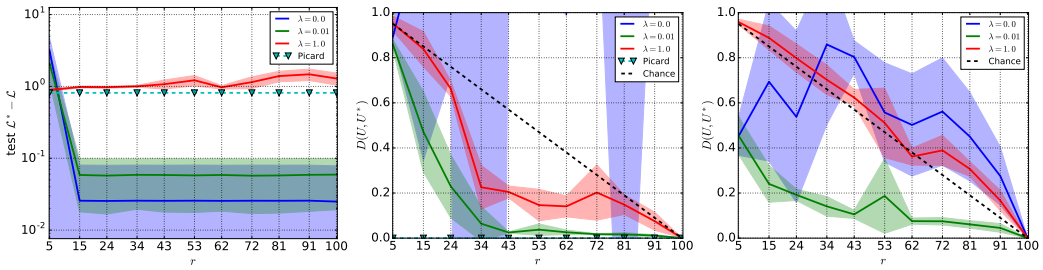
6.2 Evaluation

We evaluate our optimization scheme with two metrics. First, we compare the log-likelihood on the test set obtained with the inferred model $\mathcal{L}$ to the test log-likelihood with the model that generated the data $\mathcal{L}^*$. We use this metric when the data is generated with a single set of parameters over the dataset (i.e., the same DPP matrix L is used to generate all the observations) as in such case the difference of test log-likelihood between two models ($\mathcal{L}^* - \mathcal{L}$) is an estimation of the Kullback-Leibler divergence between the two models.

We also consider a distance between the inferred embedding U and the embedding that generated the data U^*. As the performance is invariant to any permutation of column in the matrix U (together with indices of θ) and to a scaling factor — both (U, θ) and $(\frac{1}{\sqrt{\gamma}}U, \gamma\theta)$ correspond to the same DPP matrix L — we consider the following distance that compares the linear space produced with $U \in \mathbb{R}^{V \times r}$ and $U^* \in \mathbb{R}^{V \times r^*}$:

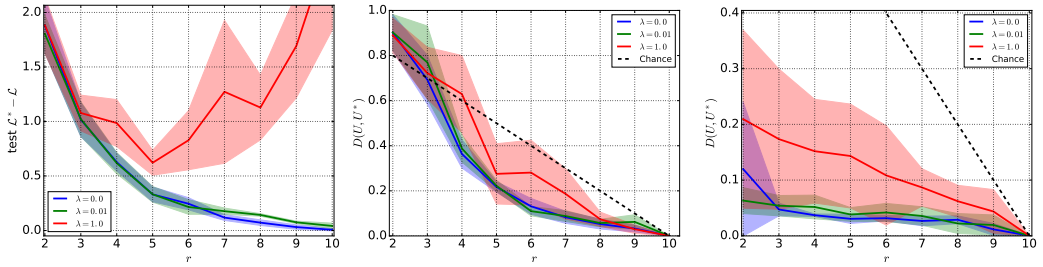
$$D(U, U^*) = \frac{\|U(U^\top U)^{-1}U^\top U^* - U^*\|_F}{\|U^*\|_F},$$

where $\|.\|_F$ is the Frobenius norm. This distance is invariant to scaling and rotation and is equal to zero when U and U^* span the same space in $\mathbb{R}^V$. In particular, if we generate randomly the r columns of $Z \in \mathbb{R}^{V \times r}$, the expectation of the distance to U^* is $\mathbb{E}_Z[D(Z, U^*)] = 1 - \frac{r}{V}$. We display this quantity as "chance" in the following.



(a) Distance in log-likelihood. (b) Distance between U and U^*. (c) Distance between U and U^*.

Figure 3: Performance for ground set $\mathcal{X} = \{1, \dots, V\}$ as a function of r. (a,b) Same θ for all the observations; (c) A different θ for each observation.



(a) Distance in log-likelihood. (b) Distance between U and U^*. (c) Distance between U and U^*.

Figure 4: Performance for ground set $\mathcal{X} = \{0, 1\}^V$ as a function of r. (a,b) Same θ for all the observations; (c) A different θ for each observation.

Continuous set $[0, 1]^2$. We compare our inference method to the best diagonal DPP $L_{\eta^*} = \eta^* I$, where $\eta^* \in \mathbb{R}$ maximizes the log-likelihood.

Items set, $\mathcal{X} = \{1, \dots, V\}$. We compare our inference method to the Picard iteration on full matrices proposed by Mariet and Sra [2015]. As they only consider the scenario where all the observations are drawn from the same DPP, we only compare to our method in that case.

6.3 Results

Continuous set $[0, 1]^2$. We present the difference in log-likelihood between the inferred model and the model that generates the data as a function of the iterations in Figure 2. The comparison between the resulting kernel and the kernel that generates the data is presented in Appendix A. We observe that our model performs significantly better than the $\eta^* I$ kernel and converges to the the true log-likelihood.

Items set & exponential set. We present the difference in log-likelihood and the distance of embeddings U between the inferred model and the model that generates the data as a function of the rank r of the representation in Figure 3 for the gound set $\mathcal{X} = \{1, \dots, V\}$ and in Figure 4 for the ground set $\mathcal{X} = \{0, 1\}^V$. We observe that the penalization may deteriorate the performance in terms of log-likelihood but significantly improves the quality of the recovered parameters. In practice, as our penalization $\mathcal{R}$ induces sparsity we recover sparse θ when $r > r^*$. For both ground sets, the parameter U^* that generated the data is recovered for $r^* \leq r < V$.

For the items set $\mathcal{X} = \{1, \dots, V\}$, while the datasets are generated with $r^* = 5$, we observe that the parameter U^* is only recovered when we optimize with $r \geq 30$. We also observe that our method performs

Table 1: Examples of reviews with extracted summaries (of size $l = 5$ sentences) colored in blue.

Review 1
Ate here once each for dinner and Sunday brunch. **[Dinner was great.] [We got a good booth seat and had some tasty food.]** I ordered just an entree since I wasn't too hungry. The guys ordered appetizers and salad and I couldn't resist trying some. The risotto with rabbit meatballs was so good. **[Corn soup, good.] [And my duck breast, also good.]** I was happy. **[The sides were good too.]** Potatoes and asparagus. Came back for Mother's Day brunch. Â Excellent booth table at the window, so we could watch our valeted car. Pretty good service. Good food. No complaints.
Review 2
This will be my 19 month old's first bar. :D I came here with a good friend and my little guy. We shared the double pork chop and the Mac n Cheese. **[The double pork chop was delicious.....] [Huge portions and beautifully prepared vegetables.] [What a wonderful selection of butternut squash, spinach, cauliflower and mashed potato.]** We were very impressed with the chop, meat was tender and full of flavor. **[The mac n cheese, was okay.]** I would definitely go back for the pork chop... might want to try the fried mushrooms too. **[Place surprisingly was pretty kid friendly.]** The bathroom actually had a bench I could change my little guy!

better than the Picard iteration of Mariet and Sra [2015] in terms of log-likelihood. The Picard iteration updates the full matrix L and there is no tradeoff between the rank and the closeness of spanned subspaces, conveyed by $D(U, U^*)$.

For the exponential set $\mathcal{X} = \{0, 1\}^V$, $r^* = 2$ and the parameter U^* is recovered for $r \geq 6$.

Real dataset. Summaries with $l = 5$ sentences of two reviews are presented in Table 1. The corresponding embeddings U are presented in Appendix D. We observe that our method is able to extract sentences that describes the opinion of the user on the restaurant. In particular, the sentences extracted with our method convey commitment of the user to aspects (food, service,...) while other sentences of the reviews only describe the context of the meal.

7 Conclusion

In this paper, we proposed a new class of determinantal point processes that can be run on a huge number of items because of a specific low-rank decomposition. This allowed parameter learning for continuous DPPs and new applications such as document modelling and summarization.

We apply our model on exponential set $\mathcal{X} = \{0, 1\}^V$ to model documents, it would be interesting to apply our inference to the infinite ground set $\mathcal{X} = \mathbb{N}^V$ as suggested in the paper. We would also like to study the inference in continous exponential set $\mathcal{X} = \mathbb{R}^V$ using our low-rank decomposition.

While we focused primarily on DPPs to model diversity, it would also be interesting to consider other approaches based on submodularity [Djolonga and Krause, 2014, Djolonga et al., 2016] and study the tractability of these models for exponantially large numbers of items.

Acknowledgements

We would like to thank Patrick Perez for helpful discussions related to this work.

References

R. H. Affandi, E. Fox, and B. Taskar. Approximate inference in continuous determinantal processes. In *Adv. NIPS*, 2013.

R. H. Affandi, E. Fox, R. Adams, and B. Taskar. Learning the parameters of determinantal point process kernels. In *Proc. ICML*, 2014.

K. Atkinson and W. Han. *Spherical Harmonics and Approximations on the Unit Sphere: an Introduction*, volume 2044. Springer, 2012.

R. Bardenet and M. Titsias. Inference for determinantal point processes without spectral knowledge. In *Adv. NIPS*, 2015.

S. Bird, E. Klein, and E. Loper. *Natural language processing with Python*. O'Reilly Media, Inc., 2009.

L. Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17:9, 1998.

J. Djolonga and A. Krause. From MAP to marginals: Variational inference in bayesian submodular models. In *Adv. NIPS*, 2014.

J. Djolonga, S. Tschiatschek, and A. Krause. Variational inference in mixed probabilistic submodular models. In *Adv. NIPS*, 2016.

M. Gartrell, U. Paquet, and N. Koenigstein. Low-rank factorization of determinantal point processes for recommendation. *arXiv:1602.05436*, 2016.

J. Gillenwater, A. Kulesza, and B. Taskar. Discovering diverse and salient threads in document collections. In *Proc. EMNLP*, 2012a.

J. Gillenwater, A. Kulesza, and B. Taskar. Near-optimal map inference for determinantal point processes. In *Adv. NIPS*, 2012b.

J. A. Gillenwater, A. Kulesza, E. Fox, and B. Taskar. Expectation-maximization for learning determinantal point processes. In *Adv. NIPS*, 2014.

B. Kang. Fast determinantal point process sampling with application to clustering. In *Adv. NIPS*, 2013.

A. Krause and D. Golovin. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, 3(19):8, 2012.

A. Kulesza and B. Taskar. k-DPPs: Fixed-size determinantal point processes. In *Proc. ICML*, 2011.

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

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.

A. Lewis and M. Overton. Nonsmooth optimization via quasi-newton methods. *Mathematical Programming*, 141(1-2):135–163, 2013.

C. Li, S. Jegelka, and S. Sra. Efficient sampling for k-determinantal point processes. In *Proc. AISTATS*, 2016a.

C. Li, S. Jegelka, and S. Sra. Fast DPP sampling for Nystrom with application to kernel methods. In *Proc. ICML*, 2016b.

Z. Mariet and S. Sra. Fixed-point algorithms for learning determinantal point processes. In *Proc. ICML*, 2015.

Z. Mariet and S. Sra. Kronecker determinantal point processes. *arXiv:1605.08374*, 2016.

B. Scholkopf and A. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

A Continuous set $[0, 1]^2$

In this section, we present a comparison between the true marginal kernel (that generates the data) K^* and the inferred marginal kernel K_t. More precisely, $\mathcal{X} = [0, 1]^2$ and we compute the induced distance from the center point $q = (\frac{1}{2}, \frac{1}{2})$ to any point $x \in \mathcal{X}$, i.e., $K(x, q)$. We show in Figure 5 a comparison between the true distance $K^*(x, q)$ and the inferred distance $K_t(x, q)$ after $t = 100$ iterations.

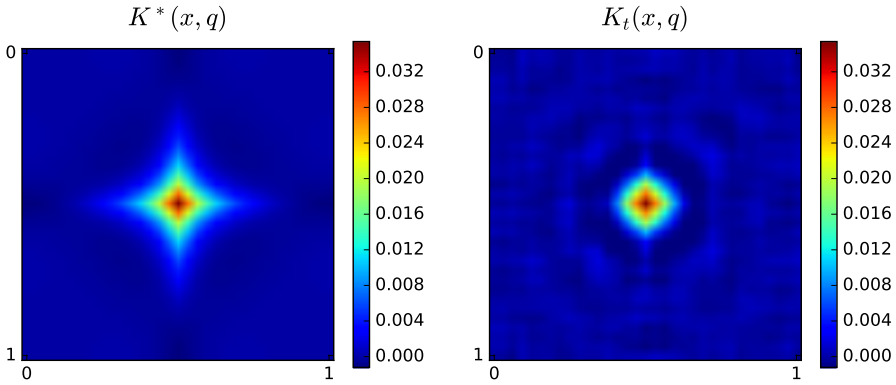


Figure 5: Comparison of K^* and K_t.

B Picard iteration

We apply the Picard iteration of Mariet and Sra [2015] on the synthetic "items" datasets (i.e., observations are generated from $L = \alpha I + U \operatorname{Diag}(\theta) U^\top$) with $N = 100$ items. We present the evolution of the objective function through the iterations with the Picard iteration in Figure 6. We observe a similar evolution than presented in the original paper [Mariet and Sra, 2015]. This however led in Figure 3 to a lower likelihood than L-BFGS on U.

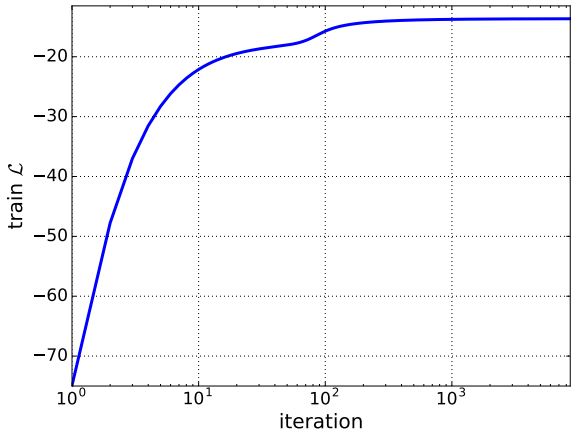


Figure 6: Picard iteration [Mariet and Sra, 2015]. Evolution of the objective function (train log-likelihood) as a function of the iterations.

C Summary as a subsample – Parameter learning

We assume that $\sum_{x\in\mathcal{X}} p(x)\phi(x)\phi(x)^\top = \mathrm{Diag}(\nu)+\mu\mu^\top$. The log-likelihood of an observed document X is expressed as $\ell(X|L) = \log\det L_X - \log\det(L+I)$. The computation of the second term, $\log\det(L+I)$, is untractable to compute in reasonable time for any L when $V \geq 20$, since $L \in \mathbb{R}^{2^V\times 2^V}$. We can still compute this value for structured L. When $L = \alpha I + \mathrm{Diag}(p)^{1/2}\Phi A\Phi^\top \mathrm{Diag}(p)^{1/2}$, we have, using the matrix determinant lemma and Woodbury identity:

$$\det(L+I) = \det[(\alpha+1)I]\det A \det\left(A^{-1} + \frac{1}{\alpha+1}\Phi^\top \mathrm{Diag}(p)\Phi\right).$$

We then have, if $\rho = \frac{1}{\alpha+1}$:

$$\begin{aligned}
\log\det\left(A^{-1} + \rho\,\mathrm{Diag}(\nu)+\rho\mu\mu^\top\right) &= \log\left[1+\rho\mu^\top\left(A^{-1}+\rho\,\mathrm{Diag}(\nu)\right)^{-1}\mu\right] + \log\det(A^{-1}+\rho\,\mathrm{Diag}(\nu)) \\
&\quad \text{(matrix determinant lemma)} \\
&= \log\left[1+\mu^\top\left(\mathrm{Diag}(1/\nu) - \mathrm{Diag}(1/\nu)(\rho A + \mathrm{Diag}(1/\nu))^{-1}\mathrm{Diag}(1/\nu)\right)\mu\right] \\
&\quad + \log\det(A^{-1}+\rho\,\mathrm{Diag}(\nu)) \\
&\quad \text{(Woodbury identity)}.
\end{aligned}$$

If we consider $A = \gamma I + U\,\mathrm{Diag}(\theta)U^\top$, we have:

$$\begin{aligned}
(\rho A + \mathrm{Diag}(1/\nu))^{-1} &= \left[(\rho\gamma I + \rho U\,\mathrm{Diag}(\theta)U^\top + \mathrm{Diag}(1/\nu)\right]^{-1} \\
&= \mathrm{Diag}\left(\frac{\nu}{1+\nu\rho\gamma}\right) \\
&\quad - \mathrm{Diag}\left(\frac{\nu}{1+\nu\rho\gamma}\right) U \left(\mathrm{Diag}(1/\rho\theta) + U^\top\,\mathrm{Diag}\left(\frac{\nu}{1+\nu\rho\gamma}\right)U\right)^{-1} U^T\,\mathrm{Diag}\left(\frac{\nu}{1+\nu\rho\gamma}\right),
\end{aligned}$$

$$\begin{aligned}
\log\det(A^{-1}+\rho\,\mathrm{Diag}(\nu)) &= \log\det\left[\mathrm{Diag}(\frac{1}{\gamma}+\rho\nu) - \frac{1}{\gamma}U\left(\mathrm{Diag}(\gamma/\theta)+U^\top U\right)^{-1}U^\top\right] \\
&\quad \text{(Woodbury identity on } A^{-1}) \\
&= \log\det\left[\left(\mathrm{Diag}(\gamma/\theta)+U^\top U\right) - \frac{1}{\gamma}U^\top\,\mathrm{Diag}\left(\frac{\gamma}{1+\nu\gamma\rho}\right)U\right] \\
&\quad - \log\det(\mathrm{Diag}(\gamma/\theta)+U^\top U) + \log\det(\frac{I}{\gamma}+\rho\,\mathrm{Diag}(\nu)) \\
&= \log\det\left[\mathrm{Diag}(\gamma/\theta)+U^\top\,\mathrm{Diag}\left(\frac{\nu\gamma\rho}{1+\nu\gamma\rho}\right)U\right] \\
&\quad - \log\det(\mathrm{Diag}(\gamma/\theta)+U^\top U) + \log\det(\frac{I}{\gamma}+\rho\,\mathrm{Diag}(\nu)),
\end{aligned}$$

$$\log\det(A) = \log\det(\mathrm{Diag}(1/\theta)+\frac{1}{\gamma}U^\top U) + \sum_k \log\theta_k + V\log\gamma$$

In the end, the computation of $\log\det(L+I)$ only needs matrix products of size V and inversions of size r.

D Results on real datasets

D.1 Columns of U

We present five embeddings (i.e., columns of $U \in \mathbb{R}^{V \times r}$) out of $r = 10$ learned on a restaurant reviews dataset with our DPP structure in Table 2 below. We display the 20 words with the highest absolute values for each column of U. We observe that our embeddings extract qualitative words (e.g., *good*, *great*, *friendly*). Even if the embeddings are not as consistent as topics extracted with topic models (e.g., LDA), we can distinguish different aspects of restaurants with the embeddings. For instance, words with positive values in embedding 1 are related to the food (e.g., *cream*, *love*, *crispy*, *tomato*); words with positive values in embedding 2 are associated to the service aspect (with *service*, *friendly*, *staff*, *attentive*). Moreover, they already lead to good summaries.

Table 2: Five embeddings (columns of U) inferred with $r = 10$ on restaurant reviews dataset.

Embed. 1	$U_{w,1}$	**Embed. 2**	$U_{w,2}$	**Embed. 3**	$U_{w,3}$	**Embed. 4**	$U_{w,4}$	**Embed. 5**	$U_{w,5}$
love	0.19	service	0.74	great	0.99	place	0.75	back	0.49
could	0.11	friendly	0.36	food	0.8	great	0.41	come	0.37
large	0.11	nice	0.33	service	0.4	good	0.35	try	0.34
cream	0.1	good	0.24	star	0.32	really	0.28	definitely	0.3
crispy	0.1	pretty	0.24	worth	0.26	love	0.21	get	0.28
tomato	0.09	staff	0.24	place	0.26	nice	0.16	would	0.26
meat	0.09	price	0.15	price	0.25	service	0.16	wait	0.23
ice	0.08	experience	0.14	back	0.21	atmosphere	0.14	dinner	0.15
sauce	0.08	well	0.14	wait	0.2	get	0.14	friend	0.15
mouth	0.08	attentive	0.13	definitely	0.18	friendly	0.13	recommend	0.14
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
back	-0.48	would	-0.26	also	-0.27	could	-0.21	small	-0.17
sushi	-0.51	think	-0.27	tasty	-0.28	dinner	-0.21	atmosphere	-0.19
place	-0.51	try	-0.28	fresh	-0.31	menu	-0.25	love	-0.21
pretty	-0.53	restaurant	-0.28	salad	-0.32	restaurant	-0.27	restaurant	-0.22
really	-0.58	one	-0.3	delicious	-0.36	well	-0.28	everything	-0.24
come	-0.64	amazing	-0.33	really	-0.4	come	-0.37	say	-0.24
great	-0.73	like	-0.57	nice	-0.43	eat	-0.38	nice	-0.25
service	-0.79	get	-0.64	like	-0.44	food	-0.39	delicious	-0.26
food	-1.08	love	-0.73	chicken	-0.45	time	-0.4	great	-0.34
good	-2.08	place	-1.05	order	-0.59	price	-0.4	food	-0.53

D.2 Rows of U

From the embeddings U, we can also compute similarity between words using the rows of U. We use the cosine similarity, i.e., for words $v, w \in \{1, \ldots, V\}$:

$$\text{Cos}(v, w) = \frac{\langle U_v, U_w \rangle}{\|U_v\|_2 \|U_w\|_2},$$

where $U_v \in \mathbb{R}^r$ is the v^{th} row of U. We present ten examples of words with their closest words for cosine similarity in Table 3. We observe that our word embeddings also capture context from the sentences. For instance, the closest words to *food* are mostly adjective applicable to food (e.g., *solid*, *average*, *decent*, *expensive*). We observe the same characteristic for the words of the top row in Table 3. For adjectives of the bottom row in Table 3 (i.e., *good*, *tender*, *tasty* and *dry*), the closest words are either synonnyms/antonyms or nouns that may have the characteristic conveyed by the corresponding adjective. For instance, among the closest words to *tender*, the words *juicy* and *flavorful* have similar meaning than *tender*, *hard* is an antonym while *gnocchi*, *shrimp*, *sausage* may be characterized as *tender*. Finally, the closest words to *time* are mostly words that convey temporal meaning (e.g., *late*, *day*, *open*, *saturday*)

Table 3: Ten examples of cosine similarity between words (i.e., between rows of U) with $r = 10$ on restaurant reviews dataset.

food	Cos	service	Cos	decor	Cos	atmosphere	Cos	meal	Cos
solid	0.97	slow	0.93	unique	1.0	cool	0.94	cheap	0.97
delivery	0.91	friendly	0.91	vibe	0.95	unique	0.88	drink	0.97
average	0.9	fast	0.9	warm	0.87	view	0.85	sunday	0.96
indian	0.9	quick	0.88	date	0.87	wonderful	0.85	though	0.96
decent	0.88	delivery	0.87	atmosphere	0.85	decor	0.85	sushi	0.94
overall	0.86	extremely	0.85	damn	0.82	fun	0.82	city	0.93
expensive	0.85	staff	0.83	cool	0.81	vibe	0.82	overall	0.92
quality	0.83	experience	0.8	beach	0.79	date	0.81	visit	0.91
italian	0.83	average	0.77	broth	0.76	kind	0.79	well	0.91
sunday	0.82	good	0.75	run	0.73	pancake	0.78	bad	0.91
good	**Cos**	**tender**	**Cos**	**tasty**	**Cos**	**dry**	**Cos**	**time**	**Cos**
location	0.98	juicy	0.96	awesome	0.99	light	0.93	late	0.97
look	0.96	hard	0.95	fresh	0.97	inside	0.93	day	0.97
hit	0.93	flavorful	0.93	delicious	0.96	ingredient	0.92	open	0.97
bad	0.9	light	0.93	people	0.95	salty	0.91	first	0.96
ever	0.9	gnocchi	0.89	course	0.92	potato	0.9	saturday	0.93
quick	0.87	shrimp	0.89	beer	0.92	sausage	0.89	far	0.92
okay	0.87	sausage	0.89	fill	0.92	meat	0.89	visit	0.91
pretty	0.86	real	0.88	fish	0.91	put	0.88	last	0.9
sure	0.85	water	0.88	nice	0.9	tender	0.85	though	0.89
city	0.85	main	0.87	server	0.9	kinda	0.85	price	0.88