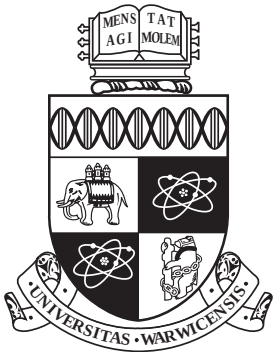
Parameter estimation for discrete determinantal structures

Dissertation submitted for the degree of
MASTER OF SCIENCE IN INTERDISCIPLINARY MATHEMATICS

Johannes Müller

June 27, 2018



UNIVERSITY OF WARWICK
DEPARTMENT OF MATHEMATICS

Acknowledgements

I would like to thank the dreamteam for being such an eggcelent friend-group and for all the pun we had during the last year.

Abstract

Determinantal point processes are random subsets that exhibit a diversifying behaviour in the sense that the randomly selected points tend to be not similar in some way. This repellent structure first arrose in theortical physics and pure mathematics, but they have recently been used to model a variety of many real world scenarios in a machine learning setup. We aim to give an overview over the main ideas of this approach which is easily accessible even without prior knowledge in the area of machine learning and sometimes omit technical calculations in order to keep the focus on the concepts.

Contents

Chapter I

Introduction and motivating examples

I.1 Motivation

I.2 Previous work

I.3 Aim and outline of the dissertation

Chapter II

Determinantal points processes: Basic notions and properties

II.1 Historical remarks

II.2 Definitions and properties

We begin by presenting general frame we will work in. This means that we will keep the notation introduced now and will use those objects throughout the thesis without further explanation. Further we will present all the important properties of determinantal point processes that we will need and postpone some calculations to the last section of this chapter. A much more ... survey of properties of determinantal point processes including extensive comparisons to several other point processes can be found in the monograph?? [Kulesza et al., 2012].

comment on continuous DPPs at one point!

2.1 Setting. Let in the following $\mathcal{Y}$ be a finite set, which we call the *ground set* and $N := |\mathcal{Y}|$ its cardinality. A *point process* on $\mathcal{Y}$ is a random subset of $\mathcal{Y}$, i.e. a random variable with values in the powerset $2^{\mathcal{Y}}$. We will identify this random variable with its law $\mathbb{P}$ and thus refer to probability measures $\mathbb{P}$ on $2^{\mathcal{Y}}$ as point processes and will not distinguish between those objects. Let further $\mathbf{Y}$ denote a random subset drawn according to $\mathbb{P}$.

2.2 Definition (Determinantal point process). We call $\mathbb{P}$ a *determinantal point process*, or in short a DPP, if we have

$$\mathbb{P}(A \subseteq \mathbf{Y}) = \det(K_A) \quad \text{for all } A \subseteq \mathcal{Y} \tag{2.1}$$

where K is a symmetric matrix indexed by the elements in $\mathcal{Y}$ and K_A denotes the submatrix of K to indexed by the elements of A. We call K the *marginal kernel* of the DPP. If the marginal kernel K is diagonal, then we call $\mathbb{P}$ a *Poisson point process*.

We note that all principle minors of K are non negative and thus K is surely non negative definite. Further it can be shown (cf. page 3 in [Borodin, 2009]) that also the complement of a DPP is a DPP with marginal kernel $I - K$ where I is the identity matrix, i.e.

cite or explain?

have a look at this and maybe explain it!

$$\mathbb{P}(A \subseteq \mathbf{Y}^c) = \det(I_A - K_A).$$

Thus we conclude $I - K \geq 0$ and obtain $0 \leq K \leq I$. This actually turns out to be sufficient for K to define a DPP through (2.1) (cf. [Kulesza et al., 2012]).

why? explain!

2.3 REPULSIVE BEHAVIOUR OF DPPS. We call the elements of $\mathcal{Y}$ *items* and by choosing $A = \{i\}$ and $A = \{i, j\}$ for $i, j \in \mathcal{Y}$ and using (2.1) we obtain the probabilities of their occurrence

$$\begin{aligned} \mathbb{P}(i \in \mathbf{Y}) &= K_{ii} \quad \text{and} \\ \mathbb{P}(i, j \in \mathbf{Y}) &= K_{ii}K_{jj} - K_{ij}^2 = \mathbb{P}(i \in \mathbf{Y}) \cdot \mathbb{P}(j \in \mathbf{Y}) - K_{ij}^2, \end{aligned} \tag{2.2}$$

Thus the appearances of the two items i and j are always negatively correlated. This negative correlation is exactly what causes the diversifying behaviour of determinantal point processes. In practice one usually models the negative correlations to be high between items that are similar in some notion. For example in a spatial setting being similar could mean being close together and in this case the selected items would tend to be not very close together. This is repulsive behaviour can be seen in Figure. Note that Poisson point processes are exactly the DPPs without correlations of the points.

insert picture!

In this light the fact that also $\mathbf{Y}^c$ exhibits negative correlations becomes less surprising. Since the set $\mathbf{Y}$ tends to spread out due to the repulsion in (2.2), the complement, which is nothing but the gaps that are left after eliminating the elements in $\mathbf{Y}$, tend to show a non clustering behaviour as well.

2.4 L-ENSEMBLES. Let us now introduce an important subclass of DPPs, namely the ones where not only the marginal probabilities can be expressed through a suitable kernel, but also the elementary probabilities. Because this will be convenient for us in the sequel, we will restrict ourselves to this case from now on. If we have even $K < I$, then we define the *elementary kernel*

really? what about sampling?

$$L := K(I - K)^{-1} \tag{2.3}$$

which specifies the elementary probabilities since one can check

$$\mathbb{P}(A = \mathbf{Y}) = \frac{\det(L_A)}{\det(I + L)} \quad \text{for all } A \subseteq \mathcal{Y}. \tag{2.4}$$

Conversely for any $L \geq 0$ a DPP can be defined via (2.2) and the corresponding marginal kernel is given by the inversion of (2.3)

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

We call DPPs which arise this way L *ensembles*.

discuss equivalence to DPPs with $\mathbb{P}(\emptyset) > 0$

The quality diversity decomposition

Note that any symmetric, positive semidefinite matrix L can be written as a Gram matrix

$$L = B^T B$$

where $B \in \mathbb{R}^{D \times N}$ whenever D is larger than the rank $\mathrm{rk}(L)$ of L. For example one could take the spectral decomposition $L = U^T C U$ of L and set $B := \sqrt{C}U$ and eventually drop some zero rows from $\sqrt{C}$. Let B_i denote the i-th column of B and write it as the product $q_i \cdot \phi_i$ where $q_i \geq 0$ and $\phi_i \in \mathbb{R}^D$ such that $\|\phi_i\| = 1$. This yields the representation

$$L_{ij} = q_i \phi_i^T \phi_j q_j =: q_i S_{ij} q_j$$

and we call q_i the *quality* of the item $i \in \mathcal{Y}$ and ϕ_i the *diversity feature vector* of i and S the *similarity matrix*. Since we will use this decomposition multiple times, we fix its properties.

2.5 Proposition (Quality diversity parametrisation). *Let $D \in \mathbb{N}$ and let $\mathbb{S}_D$ denote the sqhere in $\mathbb{R}^D$. Further let $\mathbb{R}^{N\times N}_{sym,+}$ be the set of symmetric positive semidefinite $N\times N$ matrices. The quality diversity parametrisation is a continuous and surjective mapping*

its not a bije tion!!!

$$\Psi\colon \mathbb{R}^N_+ \times \mathbb{S}^N_D \to \left\{L \in \mathbb{R}^{N\times N}_{sym,+} \mid \operatorname{rk}(L) \le D\right\}, \quad (q,\phi) \mapsto \left(q_i \phi_i^T \phi_j q_j\right)_{1\le i,j\le N}.$$

2.6 Remark. (*i*) In the case $D = N$ the quality diversity decomposition gives a parametrisation of the whole symmetric positive definite $N \times N$ matrices.

(*ii*) Note that this parametrisation is not unique, i.e. Ψ is not injective. For example the identity matrix I can be parametrised by any orthonormal system ϕ and $q = (1,\dots,1)^T$.

(*iii*) One can without any problems consider diversity features ϕ_i in an abstract Hilbert space $\mathcal{H}$. However we will not need this in the remainder and thus restrict ourselves to the easier case Euklidean diversity features.

(*iv*) We call every preimage (q, S) of L under Ψ *quality diversity decomposition* of L.

The quality diversity decomposition will provide some useful expressions. For example the elementary probabilities take the form

$$\mathbb{P}(A = \mathbf{Y}) \propto \det\big((B^T B)_A\big) = \left(\prod_{i\in A} q_i^2\right) \cdot \det(S_A) \quad \text{for all } A \subseteq \mathcal{Y}. \tag{2.5}$$

An intuitive understanding of the quality diversity decomposition will play a central role in the modelling process of real world phenomena through DPPs. To get this we can think of $q_i \ge 0$ as a measure of how important or high in quality the item is and the diversity feature vector $\phi_i \in \mathbb{R}^D$ can be thought of as some kind of state vector that consists of internal quantities that describe the item i in some way. Further we interpret the scalar product $\phi_i^T \phi_j \in [0,1]$ as a measure of similarity between the items i and j which justifies the name similarity matrix for S. Note that if i and j are perfectly similar or antisimilar, i.e. $\phi_i^T \phi_j = \pm 1$, then they can not occur at the same time, since

$$\mathbb{P}(i, j \in \mathbf{Y}) = \det\begin{pmatrix} 1 & \pm 1 \\ \pm 1 & 1 \end{pmatrix} = 0.$$

If we identify i with the vector $B_i = q_i \phi_i \in \mathbb{R}^D$, we can obtain a geometric interpretation of (2.5) since $\det\big((B^T B)_A\big)$ is the volume that is spanned by the columns $B_i, i \in A$, which is visualised in II.1. This volume increases if the lengths of the edges that correspond to the quality increase and decrease when the similarity feature vectors point into more similar directions.

2.7 Modelling diversity over distance. Since we will use one form of diversity features multiple times, we will now give a short general formulation of it. Let $\mathcal{R} = \{r_1,\dots,r_D\}$ be a finite set which we will call the *reference set* and its elements the *reference points*. Further let

$$d : \mathcal{Y} \times \mathcal{R} \to \mathbb{R}_+, \quad f : \mathbb{R}_+ \to \mathbb{R}$$

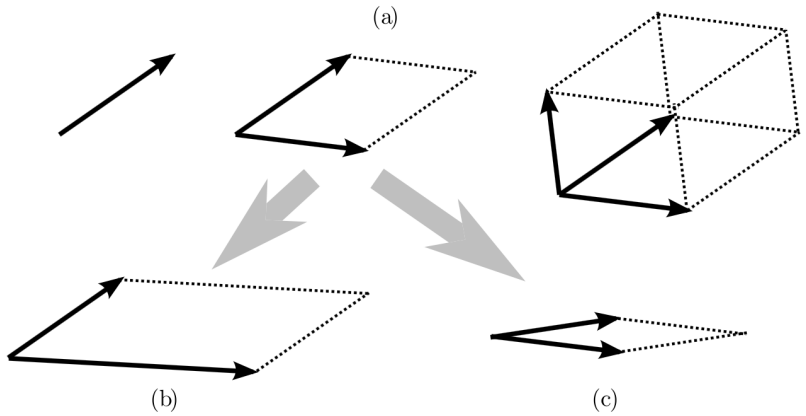


Figure II.1.: Taken from [Kulesza et al., 2012]; the first line (a) illustrates the volumes spanned by vectors, and in the second line it can be seen how this volume increases if the length – associated with the quality – increases (b) and decreases if they become more similar in direction which we interpret as two items becoming more similar (c)

mappings. Usually $d(i,r)$ will be interpreted as a measure of distance between an item $i \in \mathcal{Y}$ and a reference point $r \in \mathcal{R}$ and will typically be given by a metric on a larger space that contains both $\mathcal{Y}$ and $\mathcal{R}$. One can now model $\phi_i \in \mathbb{R}^{\mathcal{R}}$ via

$$(\phi_i)_r \propto f\big(d(i,r)\big) \quad \text{for } r \in \mathcal{R}$$

The function f will typically be decreasing and thus $(\phi_i)_r$ can be seen as a measure of how similar item i is to the reference point $r \in \mathcal{R}$. Thus the diversity feature vector ϕ_i stores how similar the item i is to all reference points and the scalar product $\phi_i^T \phi_j$ will be close to one, if the items i and j have approximately the same degrees of similarity to the reference points. It shall be noted that the choice of the D, the number of reference points bounds the rank of the kernel L and therefore of the largest subset that occurs with positive probability. Indeed we have $\mathrm{rk}(L) \leq D$ and for $A \subseteq \mathcal{Y}$ with more than D elements $\det(L_A) = 0$ and therefore $\mathbb{P}(A) = 0$. In the fourth chapter we will see that there is a natural choice for the mapping d in most cases, at least in the ones where $\mathcal{Y}$ consists of points in a metric space. On the other hand the choice of f is crucial since it determines the structure and strength of the repulsion.

2.8 Transitivity of repulsion. One last property of DPPs that we shall mention is the fact that the negative correlations of the DPP posses a transient property in the sense, that if i and j and j and k are similar, then i and k are also similar. This is due to the fact

$$\|\phi_i - \phi_j\|^2 = \|\phi_i\|^2 + \|\phi_j\|^2 - 2\phi_i^T \phi_j = 2(1 - \phi_i^T \phi_j)$$

and thus

$$\sqrt{1 - \phi_i^T \phi_k} = \frac{1}{2}\|\phi_i - \phi_k\| \leq \frac{1}{2}\big(\|\phi_i - \phi_j\| + \|\phi_j - \phi_k\|\big) = \sqrt{1 - \phi_i^T \phi_j} + \sqrt{1 - \phi_j^T \phi_k}.$$

reformulate that part!

2.9 Comparison to other point processes.

II.3 Variations of DPPs

In this section we will present some useful variations of determinantal point processes. They serve different purposes and we will shortly explain their individual benefits.

2.10 Conditional DPPs. A *conditional DPP* is a collection of DPPs indexed by $X \in \mathcal{X}$, where X is called the *input* of the conditional DPP. Thus for every $X \in \mathcal{X}$ we get a finite set $\mathcal{Y}(X)$ and a determinantal point process $\mathbb{P}(\cdot \mid X)$ on $\mathcal{Y}(X)$ which is given by the elementary kernel $L(X)$, i.e.

$$\mathbb{P}(A|X) \propto \det\big(L_A(X)\big) \quad \text{for all } A \subseteq \mathcal{Y}(X).$$

Further we denote the quality and diversity features of the conditional DPP by $q_i(X)$ and $\phi_i(X)$ respectively.

It is not immediately clear why one would want to model a family of DPPs as a conditional DPP rather than as seperate DPPs. The reason for this is that one wants to estimate the kernels $L(X)$ for every $X \in \mathcal{X}$. However if we would do this naively we would need to observe each of the DPPs $\mathbb{P}(\cdot \mid X)$ individually which is often not possible. Thus one hopes to not only memorise the kernels $L(X)$ for every single input $X \in \mathcal{X}$ but rather to learn the mapping L that assigns every input X its elementary kernel $L(X)$. If one achieved this task, one would be able to simulate and predict a DPP that one has not observed so far just by the knowledge about some DPPs that belong to the same conditional DPP. Of course this can only work if we assume some regularity or a certain structure of the function L which we will do in the third chapter where we put those consideration into a precise framework.

2.11 Fixed size or k-DPPs.

Say something about number of parameters

2.12 Structured DPPs. We call a DPP *structured DPP* or short sDPP if the ground set is the cartesian product of some other set $\mathcal{M}$, which we will call the *set of parts*, i.e. if we have

$$\mathcal{Y} = \mathcal{M}^R = \{y_i = (y_i^r)_{r=1,\dots,R} \mid i = 1, \dots, N\}$$

where R is a natural number, $M = |\mathcal{M}|$ and $N = M^R$. The quality diversity decomposition of L take the form

$$L_{ij} = q(y_i)\phi(y_i)^T\phi(y_j)q(y_j)$$

and since $N = M^R$ is typically very big, it is impractical to define or store the quality and diversity features for every item $y_i \in \mathcal{Y}$. To deal with this problem we will assume that they admit factorisations and are thus a combination of only a few qualities and diversities.

More precisely we call $F \subseteq 2^{\{1,\dots,R\}}$ a *set of factorisations* and for a *factor* $\alpha \in F$, y_α denotes the subtupel of $y \in \mathcal{Y}$ that is indexed by α. Further we will work with the decompositions

$$\begin{aligned} q(y) &= \prod_{\alpha \in F} q_\alpha(y_\alpha) \\ \phi(y) &= \sum_{\alpha \in F} \phi_\alpha(y_\alpha) \end{aligned} \tag{2.6}$$

for a suitable set of factorisations F and qualities and diversities q_α and ϕ_α for $\alpha \in F$. Note that so far this is neither a restriction of generality – we could simply choose $F = \{\{1, \dots, R\}\}$ – nor a simplification – in that case we have the exact same number of qualities and diversities.

However we are interested in the case where F consists only of small subsets of $\{1, \dots, R\}$. For example suppose that F is the set of all subsets with one or two elements, then we only have

$$R \cdot M + \binom{R}{2} \cdot M^2 = O(R^2 M^2)$$

quality and diversity features instead of

$$M^R = O(M^R).$$

This reduction of variables will make modelling, storing and estimating them feasible again in a lot of cases where naive approaches are foredoomed because of their shear size.

2.13 Sequential DPPs.

2.14 Kronecker DPPs.

II.4 The magic properties of DPPs

One of the main difficulties that arrises in the theory of discrete point processes is that they are probability measures on an exponentially large set, namely the powerset $2^{\mathcal{Y}}$ which has cardinality 2^N. Determinantal point processes have the benefit that they describe this distribution over the matrix K which consists of only N^2 parameters. This reduction of the number of parameters plays a central role in making a lot of operations possible in an efficient way. However it is not only the relatively small amount of parameters that lead to this, but also the structure of the determinant itself that leads to closed expressions for a lot of quantities like the normalisation constant in (2.4). In this section we will focus on the efficient simulation of DPPs and give a short overview of further techniques that can improve the performance of this algorithm.

We roughly follow the approach taken in and will start by showing that every determinantal point process can be seen as a mixture of a smaller class of determinantal point processes.

cite paper

2.15 Theorem (Mixture representation of DPPs). *Let $\mathbb{P}$ be a DPP and $K = \sum_{i \in \mathcal{Y}} \lambda_i v_i v_i^T$ be the spectral decomposition of its marginal kernel. Let now $\{B_i\}_{i \in \mathcal{Y}}$ be a collection of independent Bernoulli random variables with mean λ_i. Define now the random kernel*

$$K_B = \sum_{i \in \mathcal{Y}} B_i v_i v_i^T. \tag{2.7}$$

Finally define a second point process $\tilde{\mathbb{P}}$ on $\mathcal{Y}$ that is obtain by first drawing the Bernoulli variables B_i and then the DPP according to K_B. Then we have $\tilde{\mathbb{P}} = \mathbb{P}$ and thus $\tilde{\mathbb{P}}$ is also a DPP with marginal kernel K.

Before we proof this result we discuss a few consequences that show how crucial it is.

2.16 Remark. Since it is fairly easy to simulate Bernoulli experiments, it remains to know how we can sample from DPPs with marginal kernels of the form $K = \sum_{i \in I} v_i v_i^T$ for some index set $I \subseteq \mathcal{Y}$. We call DPPs of this type *elementary* and note that this corresponds to the class of DPPs were the eigenvalues of the marginal kernel are contained in $\{0, 1\}$.

2.17 Corollary (Cardinality of DPPs). *Let $\mathbb{P}$ be a DPP with kernel $K = \sum_{i \in \mathcal{Y}} \lambda_i v_i v_i^T$. Then the cardinality of the DPP is distributed like the sum of the Bernoulli variables $\{B_i\}_{i \in \mathcal{Y}}$ from theorem 2.15.*

Proof. To proof this, we only have to convince ourselves that after the Bernoulli experiments the cardinality of a DPP with kernel (2.7) has size $n := \sum_{i \in \mathcal{Y}} B_i$ almost surely. Since K_B has rank at most n, the cardinality is almost surely smaller than n. On the other hand we have

$$\mathbb{E}\big[|\mathbf{Y}|\big] = \sum_{i \in \mathcal{Y}} \mathbb{P}(i \in \mathbf{Y}) = \sum_{i \in \mathcal{Y}} (K_B)_{ii} = \operatorname{Tr}(K_B) = n. \tag{2.8}$$

In the last step we used that the trace of a symmetric matrix is the sum over its eigenvalues, which are B_i in our case. This computation lets us conclude $|\mathbf{Y}| = n$ almost surely. □

We will use Theorem 2.15 to prove that the following algorithm samples from a DPP. This will also show the existence of DPPs to a given marginal kernel since it gives an explicit construction.

Algorithm 1 Sampling from a DPP

Input: Eigendecomposition $\{v_n, \lambda_n\}$ of K

$J \leftarrow \varnothing$
for $i = 1, \dots, N$ **do**
  $J \leftarrow J \cup \{i\}$ with probability λ_i
end for
$V \leftarrow \{v_k\}_{k \in J}$
$Y \leftarrow \varnothing$
while $|V| > 0$ **do**
  $b_i \leftarrow P e_i$ the projection of e_i onto $\operatorname{span}(V)$ for $i \in \mathcal{Y}$
  Select $i \in \mathcal{Y}$ with probability $\frac{1}{|V|} \cdot \|b_i\|^2$
  $Y \leftarrow Y \cup \{i\}$
  $V \leftarrow V_\perp$ an orthonormal basis of the subspace of V perpendicular to b_i
end while
return Y

2.18 Theorem (Sampling algorithm). *Let $K \in \mathbb{R}^{N \times N}$ be any symmetric and positive definite matrix such that $K \leq I$. Then the distribution of the output Y of the above algorithm is a DPP with marginal kernel K.*

Proof. Theorem 2.15 states that an arbitrary DPP is the mixture of elementary DPPs and the for loop in the algorithm represents exactly this mixing with the respective weights. Thus we only have to show that the output of the second part of the algorithm, namely the while loop, is distributed according to a DPP with marginal kernel $K^V := \sum_{v \in V} v v^T$.

To see this let $\mathbf{Y}$ denote the output and assume that k eigenvectors where selected in the first part of the algorithm and fix now A. We seek to prove

$$\mathbb{P}(A \subseteq \mathbf{Y}) = \det(K^V_A).$$

Obviously the marginal kernel K^V has rank k and the output $\mathbf{Y}$ has exactly k elements. This is due to the fact that no element can be selected twice in the while loop and the size of V decreases by exactly one in each iteration. Thus for $|A| > k$ both sides are equal to zero and further for $|A| < k$ we get

what do we get?

Thus we only have to consider the case that A has k elements and have to show

$$\mathbb{P}(A = \mathbf{Y}) = \det(K^V_A).$$

Let for the sake of convenience $A = \{1, \dots, k\}$ and $\mathcal{Y} = \{1, \dots, N\}$. Note that it suffices to show that the while loop selects $1, \dots, k$ in this exact order with probability $\frac{1}{k!} \det(K_A^V)$.

Let V_i denote the orthonormal set V in the i-th step of the while loop and let P_{i-1} be the projection onto $\operatorname{span}(V_i)$ and set $b_i := P_0 e_i$ for $i = 1, \dots, k$. We note that if $1, \dots, i-1$ were selected in the first steps, then P_{i-1} is exactly the projection to the subspace of $\operatorname{span}(V_{i-1})$ that is orthogonal to $b_1, \dots, b_{i-1}$. Since the spaces $\operatorname{span}(V_i)$ are decreasing we have $P_i P_j = P_i$ for $i \geq j$ and thus $P_{i-1} e_i = P_{i-1} P_0 e_i = P_{i-1} b_i$. Suppose now that we have selected $1, \dots, i-1$ in the first $i-1$ steps of the while loop. The probability to select i in the next iteration is

$$\frac{1}{|V_i|} \cdot \| P_{i-1} e_i \|^2 = \frac{1}{k-i} \cdot \| P_{i-1} b_i \|^2 .$$

Thus the probability to sample $1, \dots, k$ in this order is equal to

$$\frac{1}{k!} \cdot \|b_1\| \cdot \ldots \cdot \| P_{k-1} b_k \|^2 .$$

Since P_{i-1} is the projection onto the subspace orthogonal to $b_1, \dots, b_i$, the product is equal to the squared k-dimensional surface measure of the parallel epiped spanned by $b_1, \dots, b_k$. This surface can be parametrised by the linear mapping

$$\Phi \colon [0,1]^k \to \mathbb{R}^N, \quad e_i \mapsto b_i$$

and the Jacobian of this is given by $\det(B^T B)$ where the columns of B are $b_1, \dots, b_k$. To see that this is equal to $\det(K_A^V)$ we note that $K^V = BB^T$ and set

$$C := \begin{pmatrix} I_k & 0 \\ 0 & 0 \end{pmatrix}$$

where I_k is the $k \times k$ identity matrix. Then we have

$$\det(K_A^V) = \det(C^T B B^T C) = \det(B^T C C^T B) \ = \det(B^T B)$$

since $CC^T = I$.

□

2.19 COROLLARY (EXISTENCE OF DPPS). *Let K be a symmetric $N \times N$ matrix. Then K is the marginal kernel of a DPP if and only if $0 \leq K \leq I$.*

Proof of Theorem 2.15. • □

•

2.20 SAMPLING FROM ELEMENTARY DPPS.

2.21 SAMPLING FROM DPPS.

Possible improvements

2.22 DUAL SAMPLING.

2.23 DIMENSION REDUCTION.

II.5 The mode problem

One general motivation for modelling is the hope that predictions can be made from the selected model. If the model is of stochastic nature, like in our case, and if one wants to predict its outcome, there are a few possible approaches. The first one would be to sample from this model. This relies on the intuition that a realisation of our random variable will be a rather typical example for the random event. Going one step further one could try to find the most likely outcome of the random variable, which is known as the mode problem.

2.24 The mode problem. Let X be a random variable with values in some space $\mathcal{X}$ and let f be the density of the distribution of X with respect to some reference measure. Then the *mode* is the maximiser

$$\hat{x} = \arg\max_{x \in \mathcal{X}} f(x)$$

of the density if it exists. The search for the mode is called the *mode problem*.

look for better word

Our motivation for finding the mode of a random variable was to make better predictions for it. This is justified by the assumption that the mode should be a typical realisation of the random variable. However this is not generally the case and therefore one should be cautious with this intuition. Consider for example the mixture of two independent Gaussian random variables

$$0.1 \cdot X + 0.9 \cdot Y$$

check whether this gives the desired effect and plot the density!

where X is centered with variance 10 and Y has mean 5 with variance 1, the densities are shown in Figure .... It is clear that mode is 0 in this example, but it is not a very typical outcome of the random variable, since the majority of events is centered around 10.

cite

do this!

The mode problem is rather well behaved if the density f is a smooth function defined on a subset of $\mathbb{R}^d$, but in the case of DPPs we have to deal with the probability measure on a finite set. Thus this turns into a discrete optimisation problem over the exponentially large powerset $2^{\mathcal{Y}}$. This is in general very hard to solve and it has been shown in that it is NP hard to do so or even approximate it upto a factor of $\frac{8}{9}$. However there were still different strategies proposed and we will present some of them including their main ideas.

II.6 Calculations

(*i*) Complement of DPPs

(*ii*) Explain why every suitably definite matrix is a marginal kernel

(*iii*) Expression of elementary probabilities

Chapter III

Learning setups

III.1 What does learning mean and why is it interesting?

III.2 Reconstruction of the marginal kernel using principle minors

In this section we want to see how we can estimate the marginal kernel from an increasing number of observations $\mathbf{Y}_1, \dots, \mathbf{Y}_n \subseteq \mathcal{Y}$ that are distributed according to $\mathbb{P}$. For this we will sketch the procedure in [Urschel et al., 2017]. Let $\hat{\mathbb{P}}_n$ be the *empirical measure*

$$\hat{\mathbb{P}}_n := \frac{1}{n} \sum_{i=1}^{n} \delta_{\mathbf{Y}_i}.$$

The interest in those lies in the fact that they quite natural estimates for the actual underlying distribution. More precisely they are *unbiased estimators* for $\mathbb{P}$, i.e. they agree in expectation with $\mathbb{P}$. This can be seen by evaluating it at $A \subseteq \mathcal{Y}$

$$\mathbb{E}_{\mathbb{P}}\big[\hat{\mathbb{P}}_n(A)\big] = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\mathbb{P}}\big[\delta_{\mathbf{Y}_i}(A)\big] = \mathbb{P}(A).$$

And even stronger by the strong law of large numbers they converge to $\mathbb{P}$ almost surely if the sequence $(Y_k)_{k \in \mathbb{N}}$ of observations is independent. Therefore we can consistently estimate all principle minors of K, since

$$\hat{\mathbb{P}}_n(A \subseteq \mathbf{Y}) \xrightarrow{n \to \infty} \mathbb{P}(A \subseteq \mathbf{Y}) = \det(K_A) \quad \text{almost surely.}$$

cite, explain in more detail

explain consistency

Thus the question naturally arrises whether we can reconstruct the kernel K from the knowledge of all of its principle minors, which we will address in the next section.

Principle minor assignment problem

This is known as the *principle minor assignment problem* and has been studied extensively (cf. [Griffin and Tsatsomeros, 2006] and [Urschel et al., 2017]) and an computationally efficient algorithm has been proposed for the problem in [Rising et al., 2015]. It is in fact possible to retain the matrix from its principle minors up to an equivalence relation which identifies matrices with each other, that have the same principle minors. Obviously this is sufficent for the task of learning

a DPP, because those matrices are exactly those who give rise to the same point process. To see roughly how this reconstruction works we note that the diagonal is given by

$$K_{ii} = \det(K_{\{i\}})$$

and the absolute value of the off diagonal can be obtained through

$$K_{ij}^2 = K_{ii}K_{ii} - \det\left(K_{\{i,j\}}\right).$$

see whether this proof can be done in a simplified way without considering the sparsity l

The reconstruction of the signs of the entries K_{ij} turns out to be the main difficulty, but this can be done analysing the cycles of the adjacency graph G_K corresponding to K. The adjacency graph has $\mathcal{Y}$ as its vertex set and the set of edges consists of the pairs $\{i, j\}$ such that $K_{ij} \neq 0$. The reconstruction now relies on the analysis of the cycles of this graph and it has been shown, that one only needs to know all the principle minors up to the order of the cycle sparsity of G_K (cf. [Urschel et al., 2017]). Following this method it is possible to compute estimators $\hat{K}_n$ of K in polynomial time and give a bound on the speed of convergence in some suitable metric.

state and explain the result

III.3 Maximum likelihood estimation using optimisation techniques

is this estimator unbiased? well $\hat{\mathbb{P}}_n$ is unbiased

The method of maximum likelihood estimation is a very well established procedure to estimate parameters. The philosophy of MLE is that one selects the parameter under which the given data would be the most likely to be observed and to motivate this in more detail we roughly follow the corresponding section in [Rice, 2006].

For example we might consider a sequence random variables $X_1, \dots, X_n$ with a joint density $f(x_1, \dots, x_n, \theta)$ with respect to some reference measure $\prod_{i=1}^n \mu(\mathrm{d}x_i)$. Now we want to estimate the parameter θ based on a sample $x_1, \dots, x_n$ of our random variables. Then one reasonable guess for θ would be the one under which the observation of those observations $x_1, \dots, x_n$ is the most likely. In other words we want to find the parameter θ that maximises the density $f(x_1, \dots, x_n, \theta)$. If additionally the random variables are indepent and identically distributed, their joint density factorises and thus we obtain

$$f(x_1, \dots, x_n, \theta) = \prod_{i=1}^n f(x_i, \theta)$$

where $f(x, \theta)$ is the density with respect to μ of the X_i. In practice it is often easier to maximise the logarithm of the density

$$\mathcal{L}(\theta) = \log\big(f(x_1, \dots, x_n, \theta)\big) = \sum_{i=1}^n \log\big(f(x_i, \theta)\big)$$

since this transforms the product over functions into a sum. However this is clearly equivalent to maximising the density since the logarithm is strictly monotone. We call the function $\mathcal{L}$ the *log likelihood function* and we denote its domain which is just the set of all parameters we wish to consider by Θ. Further we call its maximiser

$$\hat{\theta}_n := \arg\max_{\theta \in \Theta} \mathcal{L}(\theta)$$

the *maximum likelihood estimater* or short MLE.

III.3.1 Kernel estimation

y formulate
sk

Assume again that we have a set of observations $Y_1, \dots, Y_n \subseteq \mathcal{Y}$ drawn independently and according to the DPP $\mathbb{P}$. This time we want to find the maximum likelihood estimator for the elementary kernel and in order to do this we need to be able to express the density of the DPP which is nothing but the values of the elementary probabilities. Thus we will assume that we are dealing with L-ensembles in this section. We start by formulating the goal of this section.

3.1 Maximum likelihood estimator for L. We seek to find the MLE for the elementary kernel L in the set $\mathbb{R}^{N\times N}_{\text{sym},+}$ of all symmetric and positive semidefinite $N \times N$ matrices. The log likelihood function is now given by

$$\mathcal{L}\colon \mathbb{R}^{N\times N}_{\text{sym},+} \to [-\infty, 0], \qquad L \mapsto \log\left(\prod_{i=1}^{n} \mathbb{P}_L(Y_i)\right).$$

Using (2.4) we get the expression

$$\mathcal{L}(L) = \sum_{i=1}^{n} \log\big(\det(L_{Y_i})\big) - n\log\big(\det(L+I)\big) \tag{3.1}$$

We note that $\mathcal{L}$ is smooth and that its gradient can be expressed explicitly, at least on the domain $\{\mathcal{L} > -\infty\}$. This is due to the fact that the determinants of the submatrices are polynomials in the entries of L and the composition of those with the smooth function $\log\colon (0,\infty) \to \mathbb{R}$ stays smooth. This property allows the use of gradient methods but they face the problem that the loss function is non concave and thus those algorithms will generally not converge to a global maximiser. To see that the log linear likelihood function is not concave, we may consider the span $\{qI \mid q \in \mathbb{R}\}$ of the identity matrix. On this subspace $\mathcal{L}$ takes the form

$$\mathcal{L}(qI) = \sum_{i=1}^{n} \log(q^{|Y_i|}) - n\log((1+q)^N) = \sum_{i=1}^{n} |Y_i|\log(q) - nN\log(1+q)$$

which is not concave in general.

This obviously causes substantial computational problems in the calculation of the MLE let alone it exists. In fact it is NP hard to maximise a general non concave function and it is also conjectured to be NP hard to maximise the log likelihood function $\mathcal{L}$ in the case of L-ensembles. However there are still efficient maximising techniques for such functions that will eventually converge to local maximiser and that also work in very high dimensional spaces and thus this approach was taken by . Nevertheless we will not present this approach here, but rather favour a maximisation technique that is based on a fixed point iteration and was proposed in .

explain this term
cite
cite
cite

Fixed point iteration based maximisation

read, understand and summarise the paper

III.3.2 Learning the quality

Let again $\{Y_t\}_{t=1,\dots,n}$ be a set of independent observations drawn according to a L-ensemble $\mathbb{P}$. Unlike earlier we will not try to estimate the whole kernel L but only the qualities q_i of the

items $i \in \mathcal{Y}$. More precisely we recall that we can parametrise the positive definite symmetric matrices L using the quality diversity parametrisation

$$(q, S) \mapsto \Psi(q, S) = L \quad \text{where } L_{ij} = q_i S_{ij} q_j .$$

Now we fix a similarity kernel $\hat{S}$, that we will usually model according to some perceptions we might have, and will only try to estimate the quality $q \in \mathbb{R}_+^N$. This means that we optimise the likelihood function over a smaller set of kernels, namely the ones of the form $\Psi(q, \hat{S})$ for $q \in \mathbb{R}_+^N$. Obviously the maximal likelihood that can be achieved using this more restrictive model decreases since we consider less positive definite matrices and we have

$$\max_{q \in \mathbb{R}_+^N} \mathcal{L}(\Psi(q, \hat{S})) \leq \max_{L \in \mathbb{R}_{\text{sym},+}^{N \times N}} \mathcal{L}(L).$$

Although we can only expect a worse descriptive power of the observation, the hope is that the task of estimating only the qualities $q \in \mathbb{R}_+^N$ is more feasible which actually turn out to be true in certain cases. But before we investigate this, we clearly state our goal.

3.2 Maximum likelihood estimator for the quality. We aim to find the MLE of the quality vector $q \in \mathbb{R}_+^N$, in other words we are interested in the existence and the computability of the quantity

$$\hat{q}_n := \arg\max_{q \in \mathbb{R}_+^N} \mathcal{L}(\Psi(q, \hat{S}))$$

where the likelihood is still given by (3.1).

The motivation for restricting our ambitions of estimation to the qualities q_i rather than the whole elementary kernel $L \in \mathbb{R}_{\text{sym},+}^{N \times N}$ was to obtain a more tractable optimisation problem. In order to see whether we succeeded in that regard, we note that each summand in the log likelihood function takes the following form under the quality diversity parametrisation

$$\log\left(\prod_{j \in Y_i} q_j^2\right) + \log(\det(\hat{S}_{Y_i})) - \log\left(\sum_{A \subseteq \mathcal{Y}} \prod_{j \in A} q_j^2 \det(\hat{S}_A)\right). \tag{3.2}$$

does it makes sense?

Unfortunately this still isn't concave in q and in order to achieve this, we will have to make following assumption and keep them in throughout this section.

3.3 Log linear model for the qualities. From now on we will fix vectors $f_i \in \mathbb{R}^M$ for $i \in \mathcal{Y}$ and call them *feature vectors*. Further we set

$$q_i = \exp\left(\frac{1}{2}\theta^T f_i\right) \quad \text{for } \theta \in \mathbb{R}^M$$

and will only consider quality vectors $q \in \mathbb{R}_+^N$ that have this form.

3.4 Remark. It shall be noted that although this log linear model seems to be a harsh restriction, it isn't a restriction at all, at least theoretically. If we take $M = N$ and choose f_i to be the unit vectors in $\mathbb{R}^N$, then this just a logarithmic transformation of the parameters and thus the maximal likelihood that can be achieved with this model does not change. In practice however it will be of interest to work with rather low dimensional parameters θ, because if the ground set $\mathcal{Y}$ gets large, optimisation in $\mathcal{R}^N$ can be inefficient. In this case of course the maximal likelihood under the optimal parameter may decrease. However the approximation of the optimal parameter might become possible again which justifies this sacrifice.

Under the assumption of a log linear model for the qualities the individual terms of the log likelihood function take the form

$$\theta^T f_Y(X) + \det(S_Y(X)) - \log\left(\sum_{A\subseteq\mathcal{Y}(X)} \exp\left(\theta^T f_A(X)\right)\det(S_A(X))\right). \tag{3.3}$$

The first two terms are affine linear in θ and thus concave. To see that the last expression is also concave, it is convenient to to introduce the notion of log concavity and give a fundamental result.

3.5 Definition (Log concavity). We call a function f *log concave*, *log convex* or *log (affine) linear* if $\log(f)$ has the respective property.

3.6 Proposition (Additivity of log concavity). *The sum of log concave functions is again log concave.*

Proof. □

Give of cite proof.

As an immediate consequence we obtain that the expression in (3.3) is log concave which we will fix in a separate statement.

3.7 Corollary (Concavity of the likelihood function). *Under the log linear model for the qualities, the log likelihood function is concave in the log linearity parameter $\theta \in \mathbb{R}^M$.*

Before we discuss the actual process of maximisation of the log likelihood function we will be concered with the existence of maximisers and the consistency of the resulting estimators.

explain the term

3.8 Existence of maximisers. It is in general not true that the maximum likelihood estimator $\hat{\theta}_n$ exists for arbitrary observations $\{Y_i\}_{i=1,\dots,n}$. In fact, suppose we have fixed our similarity matrix $\hat{S}$ to be the identity, maybe we expect the items to be uncorrelated and only want to estimate their qualities. Further we believe that all items are equally likely and thus we set $f_i = 1$ for all $i \in \mathcal{Y}$. Assume now that we have only one observation Y_1 which is the whole set $\mathcal{Y}$ itself. The higher the quality of the items, the more likely this observation would be and thus the likelihood function does not posses a maximiser since it assymptotically approaches 1 if θ goes to infinity.

Comparison to learning the kernel L

III.3.3 Learning kernels of conditional DPPs

$$q_i(X) = g(f_i(X)), \quad \phi_i(X) = G(f_i(X))$$

where $f_i(X) \in \mathcal{Z}$ is being modelled and $g\colon \mathcal{Z} \to [0,\infty)$ and $G\colon \mathcal{Z} \to \mathbb{R}^D$ will be learned based on the observations. We will assume that $\mathcal{Z}$ is a subset of a vector space and therefore we call $f_i(X)$ the *feature vector*. If it is possible to estimate the quality and diversity as above, we would be able to sample from every DPP $\mathbb{P}(\cdot \mid X)$ and even from those that we haven't observed so far – just by the knowledge about DPPs with a similar structure.

Let us again illustrate this procedure in the example of the human point selection and we will restrict ourselves to learn the function g that determines the quality function, we might have a reason to be absolutely sure that we have modelled the diversity features $\phi_i(X)$ perfectly, so there is no need to learn, i.e. optimise them any further. However we are not convinced any more that

humans really do not prefer some points over others – maybe we have the feeling that they lean more towards the points located in the center of the square. Therefore it is natural to assume that the quality, which is nothing but the popularity of a point, depends on the distance to the centre point of the square $m = (1/2, 1/2)$, i.e.

$$q_i(n) = g(\|i - m\|) = g(f_i(n))$$

where we want to learn g with respect to some loss function over a given family $\mathcal{F}$ of functions.

To put this back into the general setting we note that $g \in \mathcal{F}$ gives rise to a different conditional DPP which we will denote by $\mathbb{P}_g(\cdot \mid X)$. Just like in the case of simple DPPs we will work with the negative of the log likelihood function

$$\mathcal{L}(g) := -\log\left(\prod_{t=1}^{T} \mathbb{P}_g(Y_t \mid X_t)\right)$$

and seek a minimiser of the loss function $\mathcal{L}$. Thus we obtain an optimisation problem over a family of functions and in practice it is convenient to restrict ourselves to a parametric family

$$\mathcal{F} = \left\{g_\theta \mid \theta \in U \subseteq \mathbb{R}^M\right\}.$$

In this case we write $\mathbb{P}_\theta(\cdot \mid X)$ for the conditional DPP that is induced by g_θ and the kernels become functions of θ and thus we write $L(\theta; X)$ and $K(\theta; X)$ for the kernel associated with the parameter θ. In analogue fashion we denote the loss function by

$$\mathcal{L}(g_\theta) = \mathcal{L}(\theta) = -\sum_{t=1}^{T} \log\left(\mathbb{P}_\theta(Y_t \mid X_t)\right).$$

We want to see how the log likelihood approach naturally leads to a log linear model in θ for the quality features if one wants to obtain a convex loss function. Of course the motivation for a convex loss function is given by the nice properties of convex optimisation tasks described earlier. In order to see in which cases the loss function is convex, we use (**??**) to obtain

$$\begin{aligned} -\log\left(\mathbb{P}_\theta(Y_t \mid X_t)\right) &= -\log(\det(L_Y(\theta; X))) + \log\big(\det(L(\theta; X_t) + I)\big) \\ &= -2 \cdot \sum_{i \in Y_t} \log\left(g_\theta(f_i(X_t))\right) - \log\left(\det\left(S_{Y_t}(X_t)\right)\right) \\ &\quad + \log\left(\sum_{A \subseteq \mathcal{Y}(X_t)} \left(\prod_{i \in A} g_\theta(f_i(X_t))^2\right) \det(S_A(X_t))\right). \end{aligned} \tag{3.4}$$

This expression is well defined in $[0, \infty]$ if we adapt the common convention $\det(S_\emptyset(X)) = 1$. In order to give some criteria for the convexity and coercivity of the loss function, we say that a function f *log concave*, *log convex* or *logarithmically (affine) linear* if $\log(f)$ has the respective property.

3.9 Proposition (Coercivity and convexity of the loss function). (*i*) *The rate function is coercive for all possible training sets if and only if*

$$\mathbb{P}_\theta(Y \mid X) \xrightarrow{|\theta| \to \infty} 0 \quad \textit{for all } Y \subseteq \mathcal{Y}(X) \textit{ and } X \in \mathcal{X}. \tag{3.5}$$

(*ii*) *The rate function is convex for all possible training sets if* $g_\theta(f_i(X_t))$ *is log concave in* θ *for all* $i \in \mathcal{Y}(X)$, $X \in \mathcal{X}$ *and if*

$$\prod_{i \in B} g_\theta^2(f_i(X_t))$$

is log convex in θ *for all* $B \subseteq \mathcal{Y}(X))$ *and* $X \in \mathcal{X}$.

(*iii*) *The conditions in* (*ii*) *are satisfied if and only if* $g_\theta(f_i(X))$ *is logarithmically affine linear in* θ *for every* $i \in \mathcal{Y}(X)$ *and* $X \in \mathcal{X}$.

Proof. (*i*) It is clear that under (3.5) we have

$$\exp\big(-\mathcal{L}(\theta)\big) = \prod_{t=1}^{T} \mathbb{P}_\theta(Y_t \mid X_t) \xrightarrow{|\theta| \to \infty} 0$$

for every possible training set and thus $\mathcal{L}$ is coercive. If on the other hand $\mathcal{L}$ is coercive for every training set we could also choose (Y, X) arbitrary as our training set and immediately obtain (3.5).

(*ii*) This condition for the convexity of the loss function can be directly derived from the fact that linear combination of log convex functions are log convex and formula (3.4).

(*iii*) If $g_\theta(f_i(X))$ is logarithmically affine linear, then it is also log convex and

$$\log\left(\prod_{i \in B} g_\theta^2\left(f_i(X)\right)\right) = 2\sum_{i \in B} \log\left(g_\theta(f_i(X))\right)$$

is convex. On the other side if (*ii*) holds, then all functions $\log\big(g_\theta(f_i(X))\big)$ are concave and $\sum_{i \in B} \log\big(g_\theta(f_i(X))\big)$ is convex and thus $\log\big(g_\theta(f_i(X))\big)$ has to be affine linear.

□

The result above shows that logarithmically affine linear models are the natural fit for the parametric family $\mathcal{F}$ that we want to optimise over. However they can be easily transformed into log linear models through a simple parameter shift if we assume $f_i(X) \neq 0$ and thus we can assume without loss of generality that the functions g_θ have the form

$$g_\theta(f_i(X)) = \exp\left(\frac{1}{2}\theta^T f_i(X)\right) \quad \text{for all } i \in \mathcal{Y}(X) \text{ and } X \in \mathcal{X}.$$

This structure can be used to derive some explicit expression for this case. Of course this log linear model is only well defined if the feature space $\mathcal{Z}$ is a subset of $\mathbb{R}^M$ which we will assume from now on. We note that this is no restriction if we assume a log linear model, because otherwise we could just replace the feature functions f_i by the log linearity constants $\hat{f}_i(X) \in \mathbb{R}^M$. First we can apply the explicit structure to the elementary probabilities and get

$$\mathbb{P}_\theta(A \mid X) \propto \exp\Big(\theta^T f_A(X)\Big) \det(S_A(X))$$

where $f_A(X) := \sum_{i \in A} f_i(X)$. Using this we get that the single summands of the loss function are equal to

$$-\theta^T f_Y(X) - \det(S_Y(X)) + \log\left(\sum_{A \subseteq \mathcal{Y}(X)} \exp\Big(\theta^T f_A(X)\Big) \det(S_A(X))\right) \tag{3.6}$$

Since a lot of numerical optimisation algorithms depend on the gradient of the function, it is worth noting that an explicit expression for the gradient of the loss function $\mathcal{L}$ can be derived from this formula, since differentiating (3.6) with respect to θ gives

$$\begin{aligned} -f_Y(X) + \frac{\sum_{A\subseteq\mathcal{Y}(X)} f_A(X) L_A(\theta; X)}{\sum_{A\subseteq\mathcal{Y}(X)} L_A(\theta; X)} &= -f_Y(X) + \sum_{A\subseteq\mathcal{Y}(X)} f_A(X)\mathbb{P}_\theta(A \mid X) \\ &= -f_Y(X) + \sum_{i\in\mathcal{Y}(X)} f_i(X) \sum_{i\in A\subseteq\mathcal{Y}(X)} \mathbb{P}_\theta(A \mid X) \\ &= -f_Y(X) + \sum_{i\in\mathcal{Y}(X)} f_i(X)\mathbb{P}_\theta(i \in \mathbf{Y} \mid X) \\ &= -f_Y(X) + \sum_{i\in\mathcal{Y}(X)} f_i(X) K_{ii}(\theta; X). \end{aligned} \tag{3.7}$$

The later expression of this gradient has the advantage that it can be efficiently computed in contrary to the evaluation of the exponentially large sum in the first line.

Obviously the loss function is not coercive in general, since for $f_i(X) = 0$ the probability $\mathbb{P}_\theta(\{i\} \mid X)$ is constant in θ. However it is not straight forward whether it becomes coercive under the assumption $f_i(X) > 0$ entrywise for every $i \in \mathcal{Y}(X)$ and $X \in \mathcal{X}$ and this could be investigated further.

III.3.4 Estimating the mixture coefficients of k-DPPs

III.4 A Bayesian approach to the kernel estimation

Chapter IV

Toy examples and experiments

IV.1 Minimal example?

IV.2 Points on the line

The first example we present is a selection of points on a (discretised) line. More precisely we will assume that we have 100 points on a line that are equally spaced and we aim to model a spacial repulsion between the selected points. For this we will use the method 2.7 of reference points the diversity features. In this case we will use the set $\mathcal{Y}$ itself as reference set and use a

4.1 Setup of the example. Let $\mathcal{Y} := \{1, \dots, 100\}$ and for $i \in \mathcal{Y}$. Then we will let $\phi_i \in \mathbb{R}^{100}$ be given up to scaling by

$$(\phi_i)_j \propto f\left(\frac{|i-j|}{???}\right)$$

where f is the density of the standard nomal distribution. Further we choose the qualities to be constant and so that the expected cardinatlity is 10.

check

4.2 Remark. (*i*) describe scaling including choice of cardinality

(*ii*) describe rank of the kernel?

(*iii*) describe choice of ‚repulsiveness', plot density around a point; make comment to kernel methods? comment on the qualitative properties of f and why they are suitable here

To make the difference to an uncorrelated point pattern more apparent we also defined a Poisson process, i.e. a DPP without correlations between the points with the same expected cardinality. The sampling results are compared in Figure IV.1.

make comment on zeta function!

4.3 Representation as binary sequence.

comment on problems and findings?

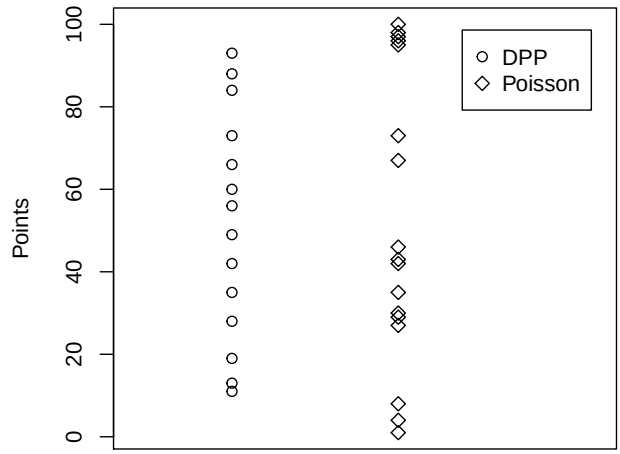


Figure IV.1.: Comparison of a DPP with negative correlations on the left and no correlations, i.e. a Poisson point process on the right.

IV.3 Points in the square

IV.4 Toy example for quality learning

Chapter V

Summary and conclusion

Chapter A

Generated code

All my coding was done in R and I will provide the code for sampling, my examples and also the learning algorithm of my toy example here. During my coding I mostly followesd Google's R Style Guide (https://google.github.io/styleguide/Rguide.xml).

A.1 Sampling algorithm

```
# Implementation of the sampling algorithm as a function

SamplingDPP <- function (lambda, eigenvectors) {
  # First part of the algorithm, doing the selection of the eigenvectors
  N = length(lambda)
  J <- runif(N) <= lambda/(1 + lambda)
  k <- sum(J)
  V <- matrix(eigenvectors[, J], nrow=N)
  Y <- rep(0, k)

  # Second part of the algorithm, the big while loop
  while (k > 0) {
    # Calculating the weights and selecting an item i according to them
    wghts <- k^(-1) * rowSums(V^2)
    i <- sample(N, 1, prob=wghts)
    Y[k] <- i
    if (k == 1) break

    # Projecting e_i onto the span of V
    help <- V %*% V[i, ]
    help <- sum(help^2)^(-1/2) * help

    # Projecting the elements of V onto the subspace orthogonal to help
    V <- V - help %*% t(t(V) %*% help)

    # Orthonormalize V and set near zero entries to zero
    V[abs(V) < 10^(-9)] <- 0
    j <- 1
    while(j <= k) {
      help2 <- rep(0, N)
      m <- 1
        while (m <= j - 1) {
        help2 <- help2 + sum(V[, j] * V[, m]) * V[, m]
```

```
      m <- m + 1
    }
    V[, j] <- V[, j] - help2
    if (sum(V[, j]^2) > 0) {
      V[, j] <- sum(V[, j]^2)^(-1/2) * V[, j]
    }
    j <- j + 1
  }
  V[abs(V) < 10^(-9)] <- 0

  # Selecting a linear independent set in V
  k <- k - 1
  q <- qr(V)
  V <- matrix(V[, q$pivot[seq(k)]], ncol=k)
 }
 return(Y)
}
```

A.2 Points on the line

```
# NEEDS: sampling algorithm

# In this example we sample points on a (discrete) line according to a DPP
# We model L directly and via the quality-diversity decomposition. We plot and
# compare the patterns to uncorrelated points i.e. to a Poisson point process.

# Minimal example ____________________________________________________________
n <- 3
L <- matrix(c(2,1,0,1,2,0,0,0,2), nrow=n)

# Points on a line ___________________________________________________________
n <- 100
L <- rep(0, n^2)
for (i in 1:n) {
  for (j in 1:n) {
    L[(i - 1) * n + j] <- dnorm((i-j) * n^(-1/4))
  }
}
L <- matrix(L, nrow=n)

# Modelling phi and q ________________________________________________________
# Points on the line.
m <- 99  # 29
n <- m + 1
q <- rep(10, n)  # 0-1 sequences: rep(10^2, n)
phi <- rep(0, n^2)
for (i in 1:n) {
  for (j in 1:n) {
    phi[(i - 1) * n + j] <- dnorm((i - j) / 10)  # 0-1 sequences: devide by 2
  }
}
phi <- matrix(phi, ncol=n)

# Log linear quality for the points on the line ______________________________
m <- 99
n <- m + 1
```

```
q <- rep(0, n)
for (i in 1:n) {
  q[i] <- 10^2 * sqrt(m) * exp(-0.2 * abs(i - 50.5))
}
phi <- rep(0, n^2)
for (i in 1:n) {
  for (j in 1:n) {
    phi[(i - 1) * n + j] <- dnorm(2 * (i - j) / sqrt(m))
  }
}
phi <- matrix(phi, ncol=n)

# General part, define L ____________________________________________________
for (i in 1:n) {
  phi[, i] <- sum(phi[, i]^2)^(-1/2) * phi[, i]
}
S <- t(phi) %*% phi
time <- proc.time()
L <- t(q * S) * q
proc.time() - time

# Compute the eigendecomposition, set near zero eigenvalues to zero and
# set up poisson point process with same expected cardinality _______________
time <- proc.time()
edc <- eigen(L)
lambda <- edc$values
lambda[lambda < 10^(-9)] <- 0
mean <- sum(lambda / (1 + lambda))
eigenvectors <- edc$vectors
lambda2 <- rep(mean / n / (1 - mean / n), n)
eigenvectors2 <- diag(rep(1, n))
proc.time() - time

# Sample and plot things ____________________________________________________
# Minimal example

# 0-1 sequences
x <- sort(SamplingDPP(lambda, eigenvectors))
as.integer(1:n %in% x)
y <- sort(SamplingDPP(lambda2, eigenvectors2))
as.integer(1:n %in% y)

# Sample from both point processes and plot the points on the line
pointsDPP <- SamplingDPP(lambda, eigenvectors)
pointsPoisson <- SamplingDPP(lambda2, eigenvectors2)
plot(rep(1, length(pointsDPP)), pointsDPP,
     ylim=c(1, n), xlim=c(.4, 3.2), xaxt='n', ylab="Points", xlab="")
points(rep(2, length(pointsPoisson)), pointsPoisson, pch=5)
legend("topright", inset=.05, legend=c("DPP", "Poisson"), pch=c(1, 5))

# Remove all objects apart from functions
rm(list = setdiff(ls(), lsf.str()))
```

A.3 Points in the square

```
# NEEDS: sampling algorithm
```

```
# In this example we sample points on a two dimensional grid according to a DPP
# We model L directly and via the quality-diversity decomposition including
# different dimensions D for the feature vectors phi. We plot and compare the
# patterns to uncorrelated points i.e. to a Poisson point process.

# Define the coordinates of a point ____________________________________________
CoordinatesNew <- function(i, n) {
  y1 <- floor((i - 1) / (n + 1))
  x1 <- i - 1 - (n + 1) * y1
  return (t(matrix(c(x1, y1)/n, nrow=length(i))))
}
DistanceNew <- function (i, j, n, d) {
  return (sqrt(colSums((CoordinatesNew(i, n) - CoordinatesNew(j, d))^2)))
}

# Direct modelling of L ________________________________________________________
m <- 19
n <- (m + 1)^2
L <- rep(0, n^2)
for (i in 1:n) {
  for (j in 1:n) {
    L[(i - 1) * n + j] = n^2 * dnorm(Distance(i, j, m))
  }
}
L <- matrix(L, nrow=n)

# Modelling phi and q __________________________________________________________
# Points in the square.
m <- 19
n <- (m + 1)^2
q <- rep(sqrt(m), n)
x <- ceiling(1:n^2 / n)
y <- rep(1:n, n)
time <- proc.time()
phi <- dnorm(sqrt(m) *matrix(DistanceNew(x, y, m, m), n))
proc.time() - time

# Quality diversity decomposition with small D _________________________________
d <- 25
q <- rep(10^5 * sqrt(m), n)
x <- ceiling(1:(n*d) / d)
y <- rep(1:d, n)
time <- proc.time()
phi <- dnorm(2 * sqrt(m) * matrix(DistanceNew(x, y, m, sqrt(d) - 1), ncol=n))
proc.time() - time

# Log linear quality for the points in the square ______________________________
m <- 39
n <- (m + 1)^2
q <- exp(-6 * DistanceNew(rep(5, n), 1:n, 2, m) + log(sqrt(m)))
x <- ceiling(1:n^2 / n)
y <- rep(1:n, n)
time <- proc.time()
phi <- dnorm(2 * sqrt(m) * matrix(DistanceNew(x, y, m, m), n))
proc.time() - time
```

```
# General part, defining L ____________________________________________________
# d <- length(phi) / n
for (i in 1:n) {
  phi[, i] <- sum(phi[, i]^2)^(-1/2) * phi[, i]
}
S <- t(phi) %*% phi
# B <- t(phi) * q
time <- proc.time()
L <- t(t(q * S) * q)  # B %*% t(B)
proc.time() - time

# Compute the eigendecomposition, set near zero eigenvalues to zero and
# set up poisson point process with same expected cardinality _________________
time <- proc.time()
edc <- eigen(L)
lambda <- edc$values
lambda[abs(lambda) < 10^(-9)] <- 0
mean <- sum(lambda / (1 + lambda))
eigenvectors <- edc$vectors
lambda2 <- rep(mean / n / (1 - mean / n), n)
eigenvectors2 <- diag(rep(1, n))
proc.time() - time

# Sample from both point processes and plot the points in the square __________
# par(mfrow = c(1,1))
time <- proc.time()
dataDPP <- sort(SamplingDPP(lambda, eigenvectors))
pointsDPP <- t(CoordinatesNew(dataDPP, m))
plot(pointsDPP, xlim=0:1, ylim=0:1, xlab="", ylab="", xaxt='n', yaxt='n', asp=1)
proc.time() - time
dataPoisson <- sort(SamplingDPP(lambda2, eigenvectors2))
pointsPoisson <- t(CoordinatesNew(dataPoisson, m))
plot(pointsPoisson, xlim=0:1, ylim=0:1, xlab="", ylab="",
                                        xaxt='n', yaxt='n', asp=1)

# Remove all objects apart from functions
rm(list = setdiff(ls(), lsf.str()))
```

A.4 Toy learning example

```
# NEEDS: Sampling algorithm, declaration of the points in the square
# TODO: Maybe do the gradient descent directly over the representation
# od the gradient

# With this toy example we aim to perform the first learning of paramters
# associated to a kernel of a DPP. More precisely we will generate our own
# data of points on a two dimensional grid with a log linear quality model
# and aim to estimate the log linearity parameter.

# Generation of data
time <- proc.time()
T <- 30
data <- rep(list(0), T)
for (i in 1:T) {
  data[[i]] <- sort(SamplingDPP(lambda, eigenvectors))
}
proc.time() - time
```

```
# Define the quality q, L, the feature sum and the loss in dependency of the
# parameter theta
Quality <- function(theta) {
  return(exp(theta[1] * DistanceNew(rep(5, n), 1:n, 2, m) + theta[2]))
}
LFunction <- function(theta) {
  return(t(t(Quality(theta) * S) * Quality(theta)))
}
Feature <- function(A) {
  # return(sum(DistanceNew(rep(5, length(A)), A, 2, m)))
  return(c(sum(DistanceNew(rep(5, length(A)), A, 2, m)), length(A)))
}
Loss <- function(theta) {
  T <- length(data)
  # Sum this over all data entries
  x <- 0
  for (i in 1:T) {
    A <- data[[i]]
    x <- x + 2 * sum(theta * Feature(A)) + log(det(matrix(S[A, A], length(A))))
  }
  return(- x + T * log(det(diag(rep(1, n)) + LFunction(theta))))
}

# Parameter estimations
time <- proc.time()
sol <- nlm(Loss, c(-3, 0))
proc.time() - time
sol$estimate

# Remove all objects apart from functions
rm(list = setdiff(ls(), lsf.str()))
```

Bibliography

[Affandi et al., 2014] Affandi, R. H., Fox, E., Adams, R., and Taskar, B. (2014). Learning the parameters of determinantal point process kernels. In *International Conference on Machine Learning*, pages 1224–1232.

[Benard and Macchi, 1973] Benard, C. and Macchi, O. (1973). Detection and"emission"processes of quantum particles in a"chaotic state". *Journal of Mathematical Physics*, 14(2):155–167.

[Borodin, 2009] Borodin, A. (2009). Determinantal point processes. *arXiv preprint arXiv:0911.1153*.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

[Griffin and Tsatsomeros, 2006] Griffin, K. and Tsatsomeros, M. J. (2006). Principal minors, part ii: The principal minor assignment problem. *Linear Algebra and its applications*, 419(1):125–171.

[Higham, 1990] Higham, N. J. (1990). Exploiting fast matrix multiplication within the level 3 blas. *ACM Transactions on Mathematical Software (TOMS)*, 16(4):352–368.

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

[Magen and Zouzias, 2008] Magen, A. and Zouzias, A. (2008). Near optimal dimensionality reductions that preserve volumes. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 523–534. Springer.

[Rice, 2006] Rice, J. (2006). *Mathematical statistics and data analysis*. Nelson Education.

[Rising et al., 2015] Rising, J., Kulesza, A., and Taskar, B. (2015). An efficient algorithm for the symmetric principal minor assignment problem. *Linear Algebra and its Applications*, 473:126–144.

[Samuel, 1959] Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229.

[Urschel et al., 2017] Urschel, J., Brunel, V.-E., Moitra, A., and Rigollet, P. (2017). Learning determinantal point processes with moments and cycles. *arXiv preprint arXiv:1703.00539*.