Parameter estimation and consistency for discrete determinantal point processes

Dissertation submitted for the degree of
Master of Science in Interdisciplinary Mathematics

Johannes Müller

September 9, 2018

Supervised by

Professor Nikolaos Zygouras and Professor Theodoros Damoulas

University of Warwick
Department of Mathematics

Acknowledgements

First and foremost I would like to thank my supervisors Nikos and Theo for the suggestion of this interesting topic as well as their time, guidance and advice they gave me along the path of this dissertation. It was a pleasure to work with you in such a friendly and inspiring atmosphere.

I would like to thank the Evangelisches Studienwerk Villigst e.V. that gave me the opportunity to complete my MSc here at the University of Warwick through their financial and also their non-material support. Further I would like to thank Patrick Dondl who gave me good advice for the choice of the programme as well as helpful insights into the English university system.

I have to thank my favourite radio station egoFM for the creation of their electronical channel FLASH which kept me well entertained during the long days of write up. Further I want to thank those who did their best to distract me from my work and my dissertation, a special mention deserve all my visitors, the Pappenheim family and my mountaineering partners. Also, I want to thank my friends back home for always having a spot on their couches and an open ear for everything. A big thanks also to all the people I met during this year and who turned it into an amazing experienceand to Γωγώ for her crash course on Greek pronunciation.

I can't thank my family enough for supporting me in every possible way during the last year. A special Dankeschön goes to my grandma for her sweet and original letters.

Last but by no means least, I would like to thank the dreamteam for being such an eggcelent friendgroup and for all the pun we had during the last year!

Abstract

Determinantal point processes (DPPs) are a probabilistic model of diverse subsets that exhibits desirable computational properties in terms of its simulation, marginalisation and other operations. That is why they have recently been used in an increasing amount of real world applications like text summarisation, the selection of a diverse subset of pictures returned by an image search or the selection of human poses in a picture (cf. [26]). A crucial step in all of those applications is the estimation of different parameters associated with DPPs and this is the focus of this dissertation. We give an overview over two different types of point estimators, their benefits and hindrances and provide proofs for their consistency. The first one is based on the reconstruction of a symmetric matrix from its principal minors and allows to obtain an estimate for the marginal kernel based on the empirical measures. The second approach is the well established method of maximum likelihood estimation (MLE). We provide proofs for the consistency of those estimators including their regularised versions which can not be found in the literature so far. In practice, however, one has to maximise the log likelihood function which is not concave in the case of DPPs and therefore this is not possible in an efficient way. This motivates the Bayesian approach to the estimation of those parameters and we show how the posterior density can be approximated using different Markov chain Monte Carlo (MCMC) methods. Further we provide toy examples for some of the presented estimation procedures and use those to investigate how the prior or, equivalently, the regularisation of the MLE influences the estimation. In fact, we show that they are not benefitial at all, not even under the presence of noise unless one knows exactly how the noise affects the estimation.

Contents

Introduction

Before we introduce determinantal point processes (DPPs) mathematically we should give a short motivation for their study as well as an overview over the dissertation and its contributions. It is the goal to give a mostly self contained presentation to different approaches for the parameter estimation for DPPs that is accessible to any student familiar with the basic notions of linear algebra, analysis and probability theory. We prove most statements of this dissertation or give precise references if the statements are not assumed to be (mathematical) general knowledge.

Motivation

Determinantal point processes are point processes, i.e. random subsets that exhibit a diversifying, repulsive behaviour in the sense that the subset is likely to obtain only elements that are different in some way. They arose first as the distribution of the eigenvalues of random matrices in [40] and later on in theoretical physics as the positions of Fermions like positively charged α-particles that repell themselves (cf. [4]). Since then, they have appeared in the study of different random objects like non intersecting random walks and the descent positions in a random digit sequence (cf. [24] and [8]). The Wigner hypothesis states that the energy levels at which a neutron is scattered or reflected by heavy nuclei are distributed according to a DPP (cf. [58]). Furthermore, DPPs arise in number theory as it has been conjectured that the positions of the non trivial roots of the Riemannian zeta function are distributed according to a determinantal point process (cf. [9]). Hence, DPPs are fundamental to different theories and are therefore highly interesting objects and a rich mathematical theory has been developed for them (cf. [7], [23], [35]). In recent years DPPs have also been used to treat different real world phenomena and we will only present three of them shortly here.

(i) *Image search*: Assume we have given a set of 10^6 pictures that were returned by a search engine for a particular query. On the first page only a few, lets say 20 can be presented and in order to increase the probability that the user is satisfied with at least one picture it is favourable to include pictures that are not very similar in some notion. This can be modelled by a DPP since the goal is to select a diverse subset of pictures (cf. [28]).

(*ii*) *Text summarisation:* DPPs have also been used successfully for extractive summarisation of news articles. The task of extractive summarisation is to select a subset of the sentences in order to obtain a reasonable summary of the text. The reason for the use of DPPs – or any other diversifying point process – is that similar sentences should not be selected for the summary since they would be quite repetitive then and hence only one of the sentences should be included in the summary (cf. [29]).

(*iii*) *Pose selection:* One of the most impressive applications of DPPs has been found in the task of human pose extraction. The goal is – given an image with an unknown amount of persons – to schematically select their poses. A pose is associated with a quadrupel of rectangles which represent the head, torso and the two arms of a person. Since a picture consists of a finite number of pixels, the number of possible poses is also finite. It is possible to model how likely a certain pose is to be actually present in a given picture. If one would sample naively according to those probabilities one runs into the following problem: Similar poses usually have almost the same probability since they should describe the actual pose just about equally well. Hence, naive approaches are likely to select more than one of those poses for the same human. This is where the repellent structure of a

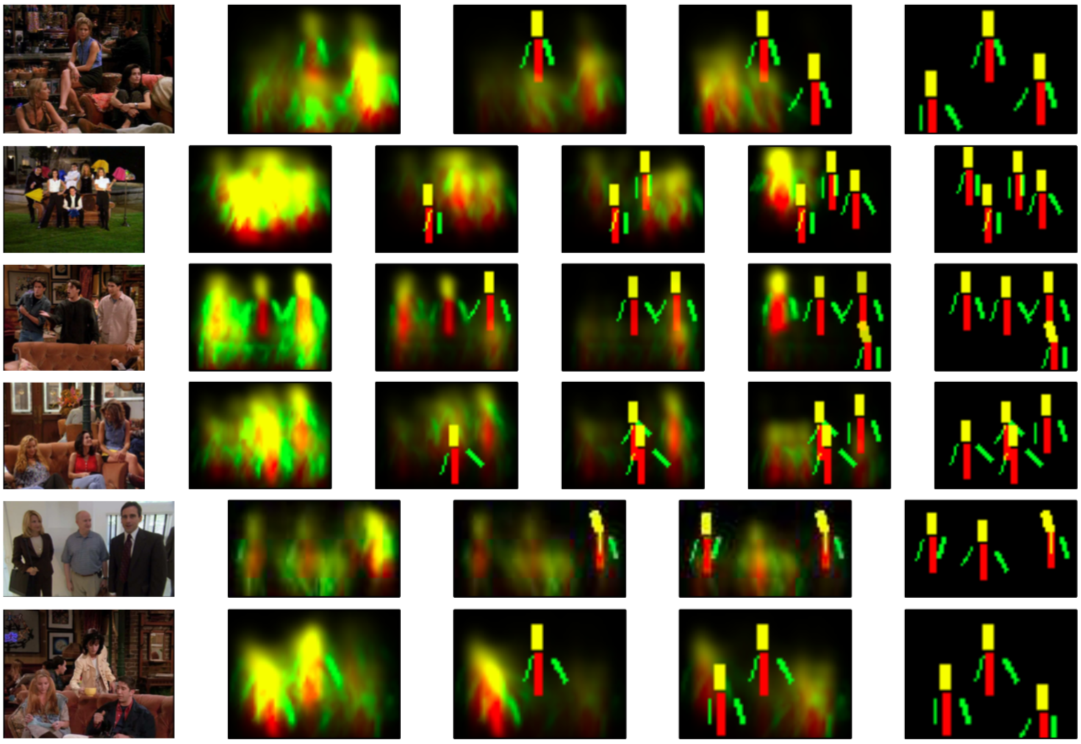


Figure 0.1: The successive selection of poses in a picture using a DPP based on a the quality of the poses which is depicted in the second column. Original graphic due to [27].

> DPP can help to make it unlikely to select similar poses which leads to the effect that in most cases only one pose is selected for one person in the picture. This approach has successfully been taken in [27] and made it possible to perform the pose selection without the knowledge of the number of persons present in the picture.

The procedure of the application of DPPs to those and further real world problems can roughly be divided into two parts. The first one consists of the selection of a suitable model for the given task and the second one of the estimation of different parameters of the DPP which is the focus of this dissertation.

Outline of the thesis

In the first chapter we introduce discrete determinantal point processes and present the fundamental concepts we will need. Further, we show that for a given marginal kernel a corresponding DPP exists and see how DPPs can be simulated and apply this to some toy examples. In the second chapter we will present two different ways to obtain an estimator for the marginal kernel or parametrisations of it. We will see that both strategies yield a consistent estimator and will discuss some of their benefits and hindrances. In the third chapter we will present the fundamentally different Bayesian approach to parameter estimation and apply it to the estimation of parameters of DPPs. In order to do this in practice we have to make use of Markov chain Monte Carlo (MCMC) methods and hence provide a minimalistic introduction to those. Finally we apply some of the presented estimation procedures to a toy example and will use this to investigate the effect of different regularisations. The appendix contains a collection of some statements used in the thesis, the R code that was used for the simulation of DPPs and also the parameter estimations that where performed.

Contributions

The dissertation is mainly based on the PhD thesis [26] and the research initiated by it, however, we provide a few novelties. We present a completely self contained presentation of the estimator of the marginal kernel that was first proposed in [60] and give a different, arguably easier proof for the consistency of this estimator. Furthermore, we will provide proofs for the consistency of the maximum likelihood estimators for different parametric models of DPPs that could not be found in the literature so far.[1] In the last chapter we give a short introduction to MCMC methods including a collection of its mathematical foundations that is shorter – and of course not as comprehensive – than in most text books. We hope that the given toy examples help the understanding of DPPs and the influence on the different parameters to its properties. The

[1] At least to the best knowledge of the author.

provision of the code may be helpful to some readers, although it should be mentioned that the algorithms and their implementations were chosen for brevity and clarity and therefore may not be computationally optimal.

Chapter I

Determinantal point processes: Basic notions and properties

In this chapter we provide an overview over the basic notions and results for discrete determinantal point processes. Those will be necessary to study the problem of parameter estimation later on. First we rigorously introduce the concept of discrete determinantal point processes and define the most important subclass that we will work with later. This is exactly the subclass where one can express the elementary probabilities of the point process nicely. This will be crucial later on if one wants to perform the parameter estimation based on elementary probabilities, like for example maximum likelihood estimation or Bayesian parameter estimation.

Then we will turn towards the question of existence and simulation of determinantal point processes. This will lead to an algorithm that samples from a DPP which we will use to simulate toy examples and also to generate the data sets we will perform parameter estimation for in the following chapters.

I.1 Definitions and properties

We begin by presenting the general frame we will work in. This means that we will keep the notation introduced here and will use those objects throughout the thesis without further explanation. We present all the important properties of determinantal point processes that we will need but omit some calculations that have been presented somewhere else and don't contribute to a better understanding of the topic. A much more in depth survey of properties of determinantal point processes including an extensive comparisons to several other point processes can be found in the report [30].

1.1 Setting. Let $\mathcal{Y}$ be a finite set, which we call the *ground set* and $N := |\mathcal{Y}|$ its cardinality. We call the elements of $\mathcal{Y}$ *items* and assume for the sake of easy notation $\mathcal{Y} = \{1, \dots, N\}$ unless specified differently. 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[1] $\mathbb{P}$ and thus refer to probability measures $\mathbb{P}$ on $2^{\mathcal{Y}}$ as point processes and will not distinguish between those objects. Further, $\mathbf{Y}$ will always denote a random subset distributed according to $\mathbb{P}$.

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

$$\mathbb{P}(A \subseteq \mathbf{Y}) = \det(K_A) \quad \text{for all } A \subseteq \mathcal{Y} \tag{1.1}$$

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

We note that all principal minors[2] of K are necessarily non negative and Sylvester's criterion implies that K is positive semi-definite.[3] Further it can be shown [4] that also the complement of a DPP is a DPP with marginal kernel $I - K$ where I is the identity matrix, i.e.

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

Thus, we can 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 (1.1) which we will see in the fourth section of this chapter.

1.3 Repulsive behaviour of DPPs. If we choose $A = \{i\}$ and $A = \{i, j\}$ for $i, j \in \mathcal{Y}$ in (1.1) we obtain the probabilities of the occurrence of the items i and j

$$\begin{aligned} \mathbb{P}(i \in \mathbf{Y}) &= K_{ii} \quad \text{and} \\ \mathbb{P}(i, j \in \mathbf{Y}) &= K_{ii}K_{jj} - K_{ij}^2 = \mathbb{P}(i \in \mathbf{Y}) \cdot \mathbb{P}(j \in \mathbf{Y}) - K_{ij}^2. \end{aligned} \tag{1.2}$$

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 and leads to a repulsive behaviour that can be seen in Figure 1.1. Note that Poisson point processes are exactly the DPPs without correlations of the points.

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 (1.2), the complement, which is nothing but the gaps that are left after eliminating the elements in $\mathbf{Y}$, tend to show a repulsive structure too.

[1] The law of a random variable is the push forward measure of the probability measure of the probability space the random variable is defined on.

[2] The *principal minors* of K are the determinants of the submatrices K_A for $A \subseteq \mathcal{Y}$.

[3] K is called *positive semi-definite* if $x^T Kx \geq 0$ for all $x \in \mathbb{R}^{\mathcal{Y}}$. The Sylvester criterion states that a matrix is positive semi-definite if and only if all principal minors are non negative.

[4] This follows from equation (2.3) in [7].

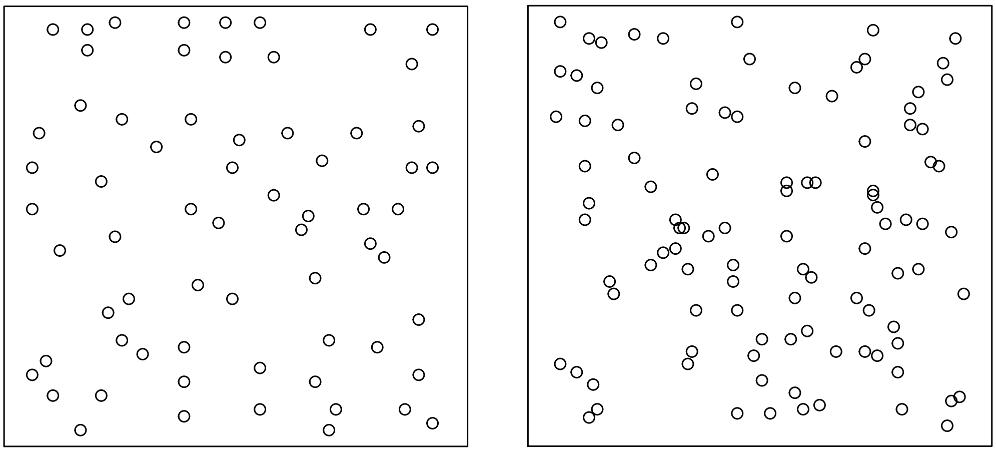


Figure 1.1: A DPP with negative correlations of close points on a 100×100 grid in the unit square on the left and a Poisson point process on the same grid and with the same expected cardinality on the right. The – in this case spatially – repellent structure of the DPP is clearly visible.

1.4 L-ensembles. Let us now introduce an important subclass of DPPs, namely the ones where not only the marginal but also the elementary probabilities can be expressed through a suitable kernel. This will be convenient for us later when we will need expressions for the elementary probability in order to take a maximum likelihood approach[5] to the estimation of certain parameters. If we have $K < I$, then we define the *elementary kernel*

$$L := K(I - K)^{-1} \tag{1.3}$$

which specifies the elementary probabilities since one can check [6]

$$\mathbb{P}(A = \mathbf{Y}) = \frac{\det(L_A)}{\det(I + L)} \quad \text{for all } A \subseteq \mathcal{Y}. \tag{1.4}$$

Conversely for any $L \geq 0$ a DPP can be defined via (1.2) and the corresponding marginal kernel is given by the inversion of (1.3)

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

and we have again $K < I$. We call DPPs which arise this way *L-ensembles*. We will see in 1.22 that the cardinality of a DPP is distributed like the sum of N Bernoulli experiments with expectation $(\lambda_n)_{n=1,\ldots,N}$ where λ_n are the eigenvalues of K. Being an L-ensemble is equivalent to $K < I$ which again is equivalent to $\lambda_n < 1$ for all $n = 1, \ldots, N$ and hence equivalent to

$$\mathbb{P}(\mathbf{Y} = \varnothing) = \mathbb{P}(|\mathbf{Y}| = 0) > 0.$$

[5] This will thoroughly be introduced in the next chapter.

[6] This is done in full detail in [26] and we will not repeat those arguments here.

The quality diversity decomposition

We note that any symmetric positive semi-definite matrix L can be written as a Gram matrix

$$L = B^T B$$

where $B \in \mathbb{R}^{D \times N}$ and 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 this as a product $B_i = q_i \cdot \phi_i$ 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* or *similarity kernel*. Since we will use this decomposition multiple times, we fix its properties.

1.5 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 semi-definite $N \times N$ matrices. The quality diversity parametrisation is a continuous and surjective mapping*

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

1.6 Remark. (*i*) In the case $D \geq N$ the quality diversity decomposition gives a parametrisation of the whole symmetric positive semi-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 $\phi \in \mathbb{S}^N_N$ 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 of Euclidean diversity features.

(*iv*) We call every preimage (q, ϕ) of L under Ψ *quality diversity decomposition* of L. Further, we call the tupel $\phi \in \mathbb{S}^N_D$ of normalised vectors the *diversity feature matrix*.

Not only will the quality diversity decomposition play a central role when it comes to the modelling of real world phenomena with DPPs. It will also provide some useful expressions like the following one for the elementary probabilies

$$\mathbb{P}(A = \mathbf{Y}) \propto \det\big((B^T B)_A\big) = \left(\prod_{i \in A} q_i^2 \right) \cdot \det(S_A) \quad \text{for all } A \subseteq \mathcal{Y}. \tag{1.5}$$

In order to get an intuitive understanding of the quality diversity decomposition we can think of $q_i \geq 0$ as a measure of how important or high in quality the item i 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 (1.5) since $\det\big((B^T B)_A\big)$ is the squared volume that is spanned by the columns $B_i, i \in A$, which is visualised in 1.2. This volume increases if the lengths of the edges which correspond to the quality increase and decrease when the similarity feature vectors point into more similar directions.

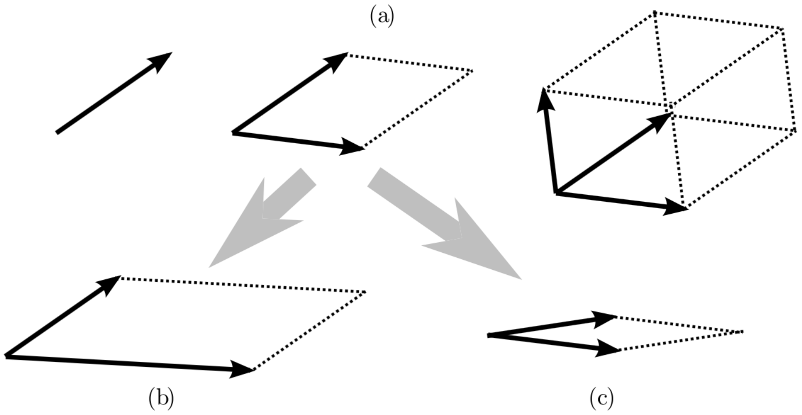


Figure 1.2: The first line (a) illustrates the volumes spanned by vectors, and in the second line it can be seen how this volume increases if the length – associated with the quality – increases (b) and decreases if they become more similar in direction which we interpret as two items becoming more similar (c). Original graphic from [30].

1.7 Modelling diversity over distance. Since we will use one approach for the 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

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

be mappings. Usually $d(i,r)$ will be some 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 $\mathcal{Y}$ 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 encodes 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 number of reference points bounds the rank of the kernel L and therefore also the largest subset that occurs with positive probability. Indeed we have $\mathrm{rk}(L) \le D$ and for $A \subseteq \mathcal{Y}$ with more than D elements $\det(L_A) = 0$ and therefore $\mathbb{P}(A) = 0$. In the last section of this chapter we will give examples where d is quite naturally a metric and will see how the choice of f is crucial for the strength of the repulsion.

Similar approaches for the modelling of the diversity feature vectors have been taken in [33] and [27] and further the method of reference points has been used in [3] to obtain bounds for the elementary probabilities of a DPP.

1.8 Comparison to other point processes. A wide variety of point processes has been studied and used in different applications and determinantal point processes are by far not the only diversifying point processes. For example every Poisson point process can be turned into a process with negative correlations by removing all points that lie within a certain distance of another point of the subset. Another well studied class of point processes are the so called *Gibbs* or *Markov point proccesses*. The elementary probabilities are given by

$$\mathbb{P}(A) \propto \exp(-F(A))$$

where $F\colon 2^{\mathcal{Y}} \to \mathbb{R}$ is called the *energy function* is interpreted as a measure of how unfavourable a subset $A \subseteq \mathcal{Y}$ is.

Although some of those classes of point processes posses nice theoretical properties they share one major drawback. In fact, a lot computations and also the simulation of those point processes can not be performed efficiently. For example, in the case of Gibbs point processes even the time needed for the computation of the normalisation constant

$$\sum_{A \subseteq \mathcal{Y}} \exp(-F(A))$$

grows exponentially with N since it is the sum over an exponentially large set. However, the special structure of the determinant itself leads to the explicit expression in (1.4) of the normalisation constant for DPPs and the computation time for the exact computation of it via Gauss elimination only grows like N^3 or even slower for numerical approximations (cf. [61] and [15]). A more in depth comparison of different point processes including their descriptive power can be found in [30].

The mode problem

One general motivation for modelling is the hope that one can make predictions based on 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 and possibly simplest one would be to sample from this model. This relies on the intuition that a realisation of a random variable should 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.

1.9 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*.

Our motivation for finding the mode of a random variable was to make better predictions for it. This hope is based on the believe 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. To see this we consider a random natural number with the following distribution

$$\mathbb{P}(\{n\}) := \begin{cases} 0.1 & \text{if } n = 20 \\ 0.09 & \text{if } n = 0, 1, \dots, 9 \\ 0 & \text{otherwise} \end{cases} .$$

Although 20 is the most likely elementary event, it is not a very typical outcome, since in 90% of the cases the random variable will have values in $\{0, 1, \dots, 9\}$ and hence is far away from the mode of the distribution. Similar examples can easily be constructed for continuous distributions.

The mode problem can often be solved explicitely or at least numerically if the density f is a smooth function defined on a subset of $\mathbb{R}^d$. However, in the case of DPPs we have to deal with the probability measure on a finite set and thus the mode problem is a discrete optimisation problem over the powerset $2^{\mathcal{Y}}$. The exponential size of the powerset turns this into a hard computational task and it has been shown that the time to compute the mode – or even an event with more than $\frac{8}{9}$ its probability – grows exponentially with the cardinality N of $\mathcal{Y}$ (cf. [30]). However, some algorithms for the approximation of the mode have been proposed for certain classes of DPPs or for the goal to find a subset of at least $\frac{1}{4}$ of the probability of the mode. For further reading we refer to [13] and [17].

I.2 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.

1.10 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 $\mathbb{P}(\cdot|X)$ on $\mathcal{Y}(X)$. We will usually assume that the single DPPs are L-ensembles and hence they are given by elementary kernels $L(X)$, 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 separate DPPs. The reason for this is that one wants to estimate the kernels $L(X)$ for every $X \in \mathcal{X}$. With a naive approach one would need to observe each of the DPPs $\mathbb{P}(\cdot|X)$ 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 estimate the mapping 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 and we will see one approach how this can be done in the next chapter.

In conclusion conditional DPPs are suitable for the extrapolation of parameter estimation from observed to similar DPPs.

1.11 Fixed size or k-DPPs. We have introduced DPPs as a model of random diverse subsets of a finite set. However, there are a lot of cases where the size of this subset is already known, like for example if the DPP models the position of football players on the field, we already know how many points we have to select, namely 11 – at least if no player was sent off or got injured.

The straight forward procedure to obtain a probability distribution over all subsets of a fixed size that still propagates diversity is to condition a DPP on the event that it has this exact size. If we conditioned on the event that the point process has $k \leq N$ elements, we call this new point process *k-DPP*. Luckily k-DPPs possess similarly attractive properties like normal DPPs, in the sense that there is an analytical form of the normalisation constant as well as an effective sampling algorithm (cf. [28]). Hence, k-DPPs allow to describe random diverse subsets of fixed size.

1.12 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

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

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

$$\begin{aligned} q(y) &= \prod_{\alpha \in F} q_\alpha(y_\alpha) \\ \phi(y) &= \sum_{\alpha \in F} \phi_\alpha(y_\alpha) \end{aligned} \tag{1.6}$$

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 $F = \left\{\{1, \dots, R\}\right\}$ – 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[7]

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

quality and diversity features instead of

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

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

Because we will neglect sDPPs in the following, we should quickly mention the reason why one could want to select the set with one and two elements as a factorisation. One could try to describe the trajectory of football players over a field through a sDPP and hence $\mathcal{M}$ would be a

[7] We write $f(x) = O(g(x))$ if $f(x) \leq Mg(x)$ for all $x \geq x_0$ and one $M > 0$.

discretisation of the field and $r = 1, \dots, R$ the different timesteps that one considers. Then the qualities q_α for $|\alpha| = 1$ are a measure of how favourable a position is for a player and the qualities q_α for $|\alpha| = 2$ can be seen as *transition qualities* that encode how good the transition from one position to another is. This gives the opportunity to dictate a certain regularity to the paths since a very big jump in position – the equivalent to a very irregular path – is very unlikely to occur in real life and can therefore be made unlikely by the assignment of a low transition quality. For more examples of the versatile applications of sDPPs we refer to [27].

I.3 Simulation and Existence of DPPs

One of the greatest challenges in the application of discrete point processes is that they are probability measures over 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 through 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 a computationally 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 analytical expressions for a lot of quantities like the normalisation constant in (1.4). In this section we will focus on the simulation of DPPs and see how the special properties of the determinant play a central role here as well. In the end we will give a short overview of further techniques that can improve the performance of this algorithm.

It should be mentioned that this section can be skipped if one is solely interested in the estimation of the parameters of DPPs.

I.3.1 Cauchy-Binet type identities

First we state a general form of the famous Cauchy-Binet identity and will then derive the version for matrices afterwards. Then we derive a result which can be seen as a formula for marginalisation for determinantal point processes and adapt ideas from [47] for this.

1.13 Proposition (Cauchy-Binet). *Let $(\mathcal{X}, \mu)$ be a σ finite measure space and let further $\phi_i, \psi_i \in L^2(\mu)$ be square integrable functions for $i = 1, \dots, n$. Then we have*

$$\frac{1}{n!} \int\limits_{\mathcal{X}^n} \det\left(\phi_i(x_j)\right)_{1\le i,j\le n} \det\left(\psi_i(x_j)\right)_{1\le i,j\le n} \mu(\mathrm{d}x_1) \cdot \ldots \cdot \mu(\mathrm{d}x_n)$$

$$= \det\left(\int_{\mathcal{X}} \phi_i(x)\psi_j(x)\mu(\mathrm{d}x)\right)_{1\le i,j\le n}.$$

Proof. We use the Leibniz formula to express the determinants in terms of permutations. This yields

$$
\begin{aligned}
&\int\limits_{\mathcal{X}^n} \det\left(\phi_i(x_j)\right)_{1\le i,j\le n} \det\left(\psi_i(x_j)\right)_{1\le i,j\le n} \mu(\mathrm{d}x_1)\cdot\ldots\cdot\mu(\mathrm{d}x_n)\\
&= \int\limits_{\mathcal{X}^n} \sum_{\sigma,\tau\in S_n} \operatorname{sgn}(\sigma)\operatorname{sgn}(\tau)\prod_{i=1}^{n}\phi_i(x_{\sigma(i)})\psi_i(x_{\tau(i)})\mu(\mathrm{d}x_1)\cdot\ldots\cdot\mu(\mathrm{d}x_n)\\
&= \int\limits_{\mathcal{X}^n} \sum_{\sigma,\tau\in S_n} \operatorname{sgn}(\sigma)\operatorname{sgn}(\tau)\prod_{i=1}^{n}\phi_i(x_{\sigma(i)})\psi_{\tau^{-1}(\sigma(i))}(x_{\sigma(i)})\mu(\mathrm{d}x_1)\cdot\ldots\cdot\mu(\mathrm{d}x_n)\\
&= \sum_{\sigma,\tau\in S_n} \operatorname{sgn}\left(\tau^{-1}\circ\sigma\right)\prod_{i=1}^{n}\int\limits_{\mathcal{X}}\phi_i(x)\psi_{\tau^{-1}(\sigma(i))}(x)\mu(\mathrm{d}x)\\
&= n!\cdot\sum_{\rho\in S_n}\prod_{i=1}^{n}\int\limits_{\mathcal{X}}\phi_i(x)\psi_{\rho(i)}(x)\mu(\mathrm{d}x)\\
&= n!\cdot\det\left(\int_{\mathcal{X}}\phi_i(x)\psi_j(x)\mu(\mathrm{d}x)\right)_{1\le i,j\le n}.
\end{aligned}
$$

In the calculation we have used that the sign function is a group homomorphism from the permutation group to $\{\pm 1\}$ and thus

$$\operatorname{sgn}\left(\tau^{-1}\circ\sigma\right) = \operatorname{sgn}(\tau)^{-1}\operatorname{sgn}(\sigma) = \operatorname{sgn}(\sigma)\operatorname{sgn}(\tau).$$

Further the second to last step is valid since for $\rho\in S_n$ exactly $n!$ pairs of permutations (σ,τ) satisfy $\tau^{-1}\circ\sigma=\rho$. □

Now we present a discrete analogon of the Cauchy-Binet identity which will be of great use later. We write $[n]$ for the set $\{1,\ldots,n\}$ where n is a natural number and A_{IJ} for the submatrix of A where the first index is in I and the second one in J. Further, we keep the notation $A_I = A_{II}$.

1.14 Proposition (Cauchy-Binet for matrices). *Let $m,n\in\mathbb{N}, m\le n$ be two natural numbers and $A\in\mathbb{R}^{m\times n}$, $B\in\mathbb{R}^{n\times m}$ two matrices. Then we have*

$$\det(AB) = \sum_{\substack{I\subseteq[n]\\|I|=m}} \det\left(A_{[m]I}\right)\det\left(B_{I[m]}\right).$$

Proof. The assertion follows from the general Cauchy-Binet identity by using the counting measure on $[n]$ since the right hand side is equal to

$$\frac{1}{m!}\sum_{i_1,\ldots,i_m\in[n]}\det\left(A_{ki_l}\right)_{1\le k,l\le m}\det\left(B_{i_k l}\right)_{1\le k,l\le m}$$

where we used that the determinants vanish if two indices i_k and i_l agree for $k\neq l$. □

1.15 Proposition (Marginalisation). *Let $(\mathcal{X}, \mu)$ be a a σ finite measure space and assume that $\{\phi_i\}_{i=1,\dots,n} \subseteq L^2(\mu)$ is an orthonormal set. Let $x_1, \dots, x_m \in \mathcal{X}$ for $m < n$, then we have*

$$\frac{1}{(n-m)!} \int\limits_{\mathcal{X}^m} \det\left(\phi_i(x_j)\right)^2_{1\le i,j\le n} \mu(\mathrm{d}x_{m+1}) \cdot \ldots \cdot \mu(\mathrm{d}x_n)$$

$$= \det\left(\sum_{k=1}^{n} \phi_k(x_i)\phi_k(x_j)\right)_{1\le i,j\le m} .$$

Proof. Just like in the proof of the Cauchy-Binet identity we begin by expressing the determinants through permutations and obtain

$$\int\limits_{\mathcal{X}^m} \det\left(\phi_i(x_j)\right)^2_{1\le i,j\le n} \mu(\mathrm{d}x_{m+1}) \cdot \ldots \cdot \mu(\mathrm{d}x_n)$$

$$= \sum_{\sigma,\tau\in S_n} \operatorname{sgn}(\sigma)\operatorname{sgn}(\tau) \int\limits_{\mathcal{X}^m} \prod_{k=1}^{n} \phi_{\sigma(k)}(x_k)\phi_{\tau(k)}(x_k)\mu(\mathrm{d}x_{m+1}) \cdot \ldots \cdot \mu(\mathrm{d}x_n).$$

The multiple integrals over the product can be evaluated individually and hence the term above is only non trivial – and identical to one in this case – if $\sigma(k) = \tau(k)$ for $k = m+1, \dots, n$ which we will denote by $\sigma \sim \tau$. Therefore, the expression is equal to

$$\sum_{\substack{\sigma,\tau\in S_n\\ \sigma\sim\tau}} \operatorname{sgn}(\sigma)\operatorname{sgn}(\tau) \prod_{k=1}^{m} \phi_{\sigma(k)}(x_k)\phi_{\tau(k)}(x_k)$$

$$= (n-m)! \cdot \sum_{\substack{I\subseteq[n]\\ |I|=m}} \det\left(\phi_{i_k}(x_l)\right)^2_{1\le k,l\le m}$$

$$= (n-m)! \cdot \det\left(\sum_{k=1,\dots,n} \phi_k(x_i)\phi_k(x_j)\right)_{1\le i,j\le m}$$

where we used Cauchy-Binet and the notation $I = \{i_1, \dots, i_m\}$. The combinatorial factor of $(n-m)!$ arises since for a fixed image $I = \sigma(\{1, \dots, m\})$, the two permutations σ and τ still have $(n-m)!$ possibilities for their values on $\{m+1, \dots, n\}$. □

Just like in the case of the Cauchy-Binet identity we will give a discrete version of the previous result in which the normalisation factor does not appear.

1.16 Proposition (Variation of Cauchy-Binet). *Let $m \le n$ be two natural numbers and $B \in \mathbb{R}^{m\times n}$ be a matrix such that the rows of B form an orthonormal system. Further, let $I \subseteq [n]$ with $|I| \le m$, then we have*

$$\det\left((B^T B)_I\right) = \sum_{\substack{I\subseteq J\subseteq[n]\\ |J|=m}} \det\left(B_{[m]J}\right)^2 .$$

Proof. Set $r := |I| \leq m$. If $r = m$, then the statement is trivial. Otherwise, let $I = \{i_1, \ldots, i_r\}$, then the right hand side is equal to

$$\frac{1}{(m-r)!} \cdot \sum_{i_{r+1},\ldots,i_m \in [n]} \det \left(B_{k i_l}\right)^2_{1 \leq k,l \leq r} \tag{1.7}$$

where we used that the determinant vanishes if two indices i_k and i_l agree for $k \neq l$. Now the previous result completes the proof.

□

I.3.2 Sampling and Existence

We roughly follow the approaches taken in [23] and [30] and will start with the result that every determinantal point process is the mixture of a smaller class of DPPs.

1.17 Theorem (Mixture representation of DPPs). *Let $\mathbb{P}$ be a DPP and*

$$K = \sum_{k=1}^{N} \lambda_k v_k v_k^T$$

be the spectral decomposition of its marginal kernel. Let now $\{\xi_k\}_{k=1,\ldots,N}$ be a collection of independent Bernoulli random variables with mean λ_k. Define now the random kernel

$$K_\xi = \sum_{k=1}^{N} \xi_k v_k v_k^T. \tag{1.8}$$

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

We will postpone the proof and first discuss its consequences which will be the existence of DPPs for a given marginal kernel as well as the construction of a sampling algorithm.

1.18 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 $K = \sum_{k=1}^m v_k v_k^T$ for some $m \leq N$. We call DPPs of this type *elementary* and note that this corresponds to the class of DPPs where the eigenvalues of the marginal kernel are contained in $\{0, 1\}$.

Now we study the existence and simulation of elementary DPPs and 1.17 will immediately generalise those results to DPPs without much effort.

1.19 Proposition (Existence of elementary DPPs). *Let $K = \sum_{k=1}^{m} v_k v_k^T$ for some orthonormal set $V = \{v_k\}_{k=1,\dots,m} \subseteq \mathbb{R}^{\mathcal{Y}}$. Further, define the measure on $2^{\mathcal{Y}}$ through*

$$\mathbb{P}(A) := \begin{cases} \det(K_A) & \text{if } |A| = m \\ 0 & \text{else} \end{cases}. \tag{1.9}$$

Then $\mathbb{P}$ is a DPP on $\mathcal{Y}$ with marginal kernel K. In particular elementary DPPs exist.

Proof. First we have to show that (1.9) defines a probability measure. For this let $B \in \mathbb{R}^{m \times N}$ be the matrix with rows v_k for $k = 1, \dots, m$. By definition we have $K = B^T B$ and hence

$$\sum_{\substack{A \subseteq \mathcal{Y} \\ |A| = m}} \det(K_A) = \sum_{\substack{A \subseteq \mathcal{Y} \\ |A| = m}} \det\left(B_{[m]A}\right)^2 = \det\left(BB^T\right) = \det\left(v_k^T v_l\right)_{1 \le k, l \le m} = 1$$

where we have used the Cauchy-Binet identity and the fact that V is orthonormal. It remains to check that all marginal probabilities satisfy

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

For $|A| \ge m$ this follows immediately, so let $A = \{i_1, \dots, i_r\}$ for $r < m$. Then we obtain the marginal probability of A through summation over the other $m - r$ points. Namely we have

$$\mathbb{P}(A \subseteq \mathbf{Y}) = \sum_{\substack{A \subseteq J \subseteq [n] \\ |J| = m}} \mathbb{P}(J \subseteq \mathbf{Y}) = \sum_{\substack{A \subseteq J \subseteq [n] \\ |J| = m}} \det\left(B_{[m]J}\right)^2 = \det\left((B^T B)_A\right) = \det(K_A)$$

where we used Proposition 1.16. □

1.20 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 $0 \le K \le I$.*

Now we can turn towards the simulation of elementary DPPs where we will make use of the previous result. In order to present the algorithms in a compact form we will usually present them in pseudocode. Here, the symbol ← stands for the assignment of a value, i.e. $x \leftarrow y$ means that the variable x should have the value y from now on.

1.21 Proposition (Sampling from elementary DPPs). *Let $K = \sum_{k=1}^{m} v_k v_k^T$ where $\{v_k\}_{k=1,\dots,m}$ is a set of orthonormal vectors. Then Algorithm 1 produces a random variable $\mathbf{Y}$ with values in $2^{\mathcal{Y}}$ which is an elementary DPP with marginal kernel K.*

Proof. We note that we only have to check that (1.9) holds and for this we fix $A \subseteq \mathcal{Y}$. The output $\mathbf{Y}$ has cardinality $|m|$ since no element can be selected twice in the while loop and the size of V decreases by exactly one in each iteration. Hence, it remains to show

$$\mathbb{P}(A = \mathbf{Y}) = \det(K_A)$$

Algorithm 1 Sampling from an elementary DPP

```
Input: Marginal kernel $K = \sum_{k=1}^{m} v_k v_k^T$ for $\{v_k\}_{k=1,\dots,m}$ orthonormal
$V \leftarrow \{v_k\}_{k=1,\dots,m}$
$Y \leftarrow \varnothing$
while $|V| > 0$ do
    $p_i \leftarrow Pe_i$ the projection of $e_i$ onto $\operatorname{span}(V)$ for $i \in \mathcal{Y}$
    Select $i \in \mathcal{Y}$ with probability $\frac{1}{|V|} \cdot \|p_i\|^2$
    $Y \leftarrow Y \cup \{i\}$
    $V \leftarrow V_\perp$ an orthonormal basis of the subspace of $V$ orthogonal to $p_i$
end while
return $Y$
```

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

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

Suppose now that we have selected $1, \dots, k-1$ in the first $k-1$ steps of the while loop. The probability to select k in the next iteration is

$$\frac{1}{|V_k|} \cdot \|P_{k-1} e_k\|^2 = \frac{1}{m-k} \cdot \|P_{k-1} b_k\|^2 .$$

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

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

Since P_{k-1} is the projection onto the subspace orthogonal to $b_1, \dots, b_{k-1}$, the product is equal to the squared m-dimensional surface measure of the parallelepiped spanned by $b_1, \dots, b_m$. It is well known from measure and integration theory that the squared surface is given by the determinant of the Gram matrix

$$\det \begin{pmatrix} b_1^T b_1 & \cdots & b_1^T b_m \\ \vdots & \ddots & \vdots \\ b_m^T b_1 & \cdots & b_m^T b_m \end{pmatrix} = \det\big((B^T B)_A\big)$$

where $B \in \mathbb{R}^{N \times N}$ is the matrix which columns are equal to b_k. Therefore, it remains to show $B^T B = K$. However, by definition B is the projection onto the span of $\{v_k\}_{k=1,\dots,m}$ and thus

$B = K$. Because K is symmetric like every projection, we have $B^T = B$ and hence we can conclude $B^T B = B^2 = B = K$ where we used that B is a projection. □

1.22 Corollary (Cardinality of DPPs). *Let $\mathbb{P}$ be a DPP with kernel*

$$K = \sum_{k=1}^{N} \lambda_k v_k v_k^T .$$

Then the cardinality of the DPP is distributed like the sum of the Bernoulli variables $\{\xi_k\}_{k=1,\dots,N}$ with expectations $\{\lambda_k\}_{k=1,\dots,m}$.

Proof. To prove this, we only have to convince ourselves that after the Bernoulli experiments the cardinality of a DPP with kernel (1.8) has size $m := \sum_{k=1}^{N} \xi_k$ almost surely. However, this is obvious from the construction of elementary DPPs in Proposition 1.19. □

Now we can apply Theorem 1.17 to extend the sampling algorithm to general DPPs.

Algorithm 2 Sampling from a DPP

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

1.23 Theorem (Sampling algorithm). *Let $K \in \mathbb{R}^{N \times N}$ be any symmetric and positive semi-definite matrix such that $K \leq I$. Then the distribution of the output $\mathbf{Y}$ of Algorithm 2 is a DPP with marginal kernel K.*

Proof. Theorem 1.17 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. Further, the sampling result for elementary DPPs yields that the output of the second part of the algorithm, namely the while loop, is distributed according to a DPP with marginal kernel $\sum_{v \in V} v v^T$.

□

1.24 Remark. Later on it will usually be more convenient to model or estimate the elementary kernel L instead of the marginal kernel K. Thus, we should explain how the sampling algorithm would work in this case. Since the two kernels are related by

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

their eigendecompositions are closely related. Namely, if v is an eigenvector of L with eigenvalue $\lambda \geq 0$, then v is also an eigenvector of K with eigenvalue $\frac{\lambda}{\lambda+1} > 0$. After this transformation of the eigendecomposition of L the sampling algorithm for K can be applied.

We close this paragraph with the proof of 1.17 given in [30].

Proof of Theorem 1.17. Let $A \subseteq \mathcal{Y}, m := |A|$ and set $W_k := (v_k v_k^T)_A$ and $W_J := \sum_{k \in J} W_k$. Then we have

$$\tilde{\mathbb{P}}(A \subseteq \mathbf{Y}) = \sum_{J \subseteq [N]} \det(W_J) \cdot \tilde{\mathbb{P}}\big(\xi_j = 1 \text{ if and only if } j \in J\big).$$

Let $\big((W_{k_1})_1 (W_{k_2})_2 \cdots (W_{k_m})_m\big)$ denote the $m \times m$ matrix with j-th row equal to the j-th row of W_{k_j}. Using the multilinearity of the determinant we obtain that the marginal probability above is equal to

$$\begin{aligned}
&\sum_{J \subseteq [N]} \sum_{k_1,\ldots,k_m \in J} \det\big((W_{k_1})_1 (W_{k_2})_2 \cdots (W_{k_m})_m\big) \cdot \tilde{\mathbb{P}}\big(\xi_j = 1 \text{ if and only if } j \in J\big) \\
&= \sum_{k_1,\ldots,k_m=1}^{N} \det\big((W_{k_1})_1 (W_{k_2})_2 \cdots (W_{k_m})_m\big) \sum_{J \supseteq \{k_1,\ldots,k_m\}} \tilde{\mathbb{P}}\big(\xi_j = 1 \text{ if and only if } j \in J\big) \\
&= \sum_{k_1,\ldots,k_m=1}^{N} \det\big((W_{k_1})_1 (W_{k_2})_2 \cdots (W_{k_m})_m\big) \cdot \tilde{\mathbb{P}}\big(\xi_{k_j} = 1 \text{ if and only if } j = 1,\ldots,m\big) \\
&= \sum_{k_1,\ldots,k_m=1}^{N} \det\big((\lambda_{k_1} W_{k_1})_1 (\lambda_{k_2} W_{k_2})_2 \cdots (\lambda_{k_m} W_{k_m})_m\big) \\
&= \det\left(\sum_{k=1}^{N} \lambda_k W_k\right) = \det(K_A).
\end{aligned}$$

This computation shows that $\tilde{\mathbb{P}}$ is a DPP with marginal kernel K. □

The dual representation

We will shortly discuss one method how the simulation of DPPs can be made more efficient. The step in the sampling algorithm that takes the longest in practice is the computation of the eigendecomposition of the matrix K or L. Hence, we will quickly show how this can be reduced to the computation of the eigendecomposition of a smaller matrix.

Consider the matrix $A = B^T B \in \mathbb{R}^{N \times N}_{\text{sym},+}$, $B \in \mathbb{R}^{D \times N}$ and set $C := BB^T \in \mathbb{R}^{D \times D}_{\text{sym},+}$. Then the spectral decomposition of A and C can be related in the following way.

(i) The eigenvalues of A and C agree. In fact, if $v \in \mathbb{R}^D$ is an eigenvector to the eigenvalue $\lambda \in \mathbb{R}$, then $B^T v$ is an eigenvector to the eigenvalue λ, since

$$AB^T v = B^T BB^T v = B^T C v = \lambda B^T v.$$

(ii) If v is a normed eigenvector to the eigenvalue $\lambda > 0$, then we have

$$\left\| B^T v \right\|^2 = (B^T v)^T (B^T v) = v^T BB^T v = v^T C^T v = \lambda$$

and hence $\frac{B^T v}{\sqrt{\lambda}}$ is a normed eigenvector to the eigenvalue $\lambda > 0$.

(iii) Finally, if $\{v_1, \ldots, v_m\}$ is an orthonormal set of eigenvectors to the non trivial eigenvalues $\{\lambda_1, \ldots, \lambda_m\}$ of C, then

$$\left\{ \frac{B^T v_k}{\sqrt{\lambda_k}} \;\middle|\; k = 1, \ldots, m \right\}$$

is an orthonormal set of eigenvectors to the non trivial eigenvalues of A.

Hence, if K or L are given as a gram matrix $B^T B$, it suffices to compute the eigendecomposition of $BB^T \in \mathbb{R}^{D \times D}_{\text{sym},+}$, which could be significantly faster if $D < N$. Since the sampling algorithm relies only on the eigendecomposition it can be performed based on the dual representation presented above. It should be mentioned that typically L will modelled as a gram matrix via the quality diversity decomposition and hence this dual representation will mostly be used for L. Further, it comes with even greater benefits here, since the normalisation constant

$$\det(L + I) = \prod_{k=1}^{N} (1 + \lambda_k) = \det\left(BB^T + I\right)$$

reduces to the computation of the determinant of a $D \times D$ matrix.

The dual representation can make the computations involved with DPPs efficient, but in some cases it might not be effective enough. Therefore, different techniques have been proposed in order to achieve faster computation times, like random projections. This approach relies on the result from [37] that points in an N-dimensional space can be randomly projected into a space of dimension $O(\log(N))$ in such a way, that the volume spanned by those points is almost preserved with a high probability. For a discussion of this approach we refer to [26].

I.4 Simulation of toy examples

We will present two examples and although – or maybe even because – they are very simple they show how the choice of different parameters in the modelling process affect the DPP.

Points on a line

We start by modelling a one dimensional DPP and simulating from it. More precisely we consider the ground set $\mathcal{Y} := \{1, \ldots, 100\}$. Further, we model the diversity feature vectors like in 1.7 using reference points and choose $\mathcal{R} := \mathcal{Y}$ as a reference set. Now let f to be a normal density with mean 0, i.e. we have

$$(\phi_i)_j \propto \exp\left(-\frac{(i-j)^2}{\sigma}\right) \quad \text{for } i, j \in \mathcal{Y}.$$

We will choose $\sigma = 20$ first and then $\sigma = 5$ to see how this parameter affects the repulsion of the DPP. Finally we set the qualities to be constant and scale them so that the expected cardinality of the DPP is approximately 15. Further, we define a Poisson point process with the same expected cardinality. This means the Poisson point process includes every point independently with probability $\frac{15}{100}$. A comparison of a sample from those three point processes is depicted in Figure 1.3 and the – in this case spatially – repulsive structure of the DPP is apparent.

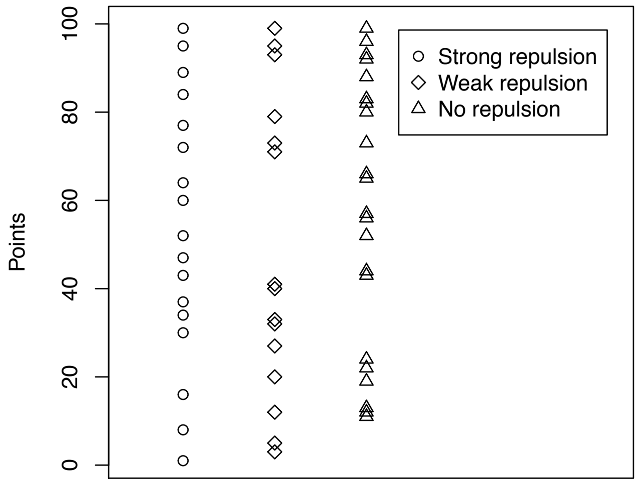


Figure 1.3: Two DPPs with a different strength of repulsion on the left and a Poisson point process on a (discretised) line with same expected cardinality. The spatial repulsion of the DPPs is clearly visible.

We shall quickly discuss what influence the choice of f has on the strength of the repulsion of the DPP. For this we let the parameter σ tend to infinity and note

$$\phi_i \xrightarrow{\sigma\to\infty} \frac{1}{\sqrt{N}} \cdot \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^N.$$

Hence the similarity $\phi_i^T \phi_j$ between item i and j will increase and therefore the negative correlation gets stronger as σ grows. Hence, the DPP will become increasingly repulsive if we increase the parameter σ which can be seen in Figure 1.3 where the first sample corresponds to the choice $\sigma = 20$, the second one to $\sigma = 5$. On a more formal level one can argue that if one models the diversity feature vectors this way, we center Gaussian densities at the reference points and then associate an item i with how likely it is under this Gaussian densities. If we increase the standard deviation σ of this density all items become increasingly similar in relation to the reference points. Similar considerations apply if f is not a normal density.

Binary sequences

It is well known that a $0-1$ sequence that is generated by a human and made to look random will typically differ strongly from a randomly generated $0-1$ sequence.[8] For example the total amount of changes between zeros and ones is typically significantly higher in the human pseudorandom sequence. Also the length of the longest chain of zeros or ones will likely be significantly shorter (cf. [56]). Hence, the position of the ones tend to repell themselves since a human will typically think that a long chain of successive ones will be atypical for a random sequence and thus one could model these positions through a DPP.

We will consider $0-1$ sequences of length 30 and therefore set $\mathcal{Y} := \{1, \dots, 30\}$ and define the DPP in the exact same way as above. This means, we choose f to be a normal density, but will choose the variance such that the repulsion is visible but not too strong and scale the qualities such that the expected cardinality is 15, since a human would probably aim to write down around 15 ones. In completely analogue fashion to the previous examples we define a Poisson point process with the same expected cardinality. This time we will represent the samples from the two point processes through a $0-1$ sequence where a 1 at the i-th position indicates that item i was in the sample. We obtain the two following samples:

$$101001011010001010010101010011$$
$$111111010100001101000110100 11$$

Although the first sequence might actually look more random at first, this is the one generated by the DPP and on second sight one realises that the positions of the ones are rather spread out, i.e. negatively correlated. Indeed in the first sequence the longest chain of zeros or ones is of length three, in the second one of length seven. The amount of changes between zero and one is 22 in the first sequence and only 14 in the second sequence.

Although the DPP presented above incorporates some of the properties one might expect from a $0-1$ sequence created by a human, this process will not be exactly determinantal. However,

[8] At least if the human is sufficiently unfamiliar with statistics.

it shall be noted that different $0-1$ sequences studied in probability theory exhibit an exact determinantal structure. For example Borodin-Deift-Fulman studied the sequence of descent positions in [8]. To obtain those sequences one first samples a sequence of $N+1$ independent digits $\{0,1,\dots,9\}$. Then one marks the positions in $\{1,\dots,N\}$ where the successor of the digit is strictly smaller than the current digit. The heuristical argument why those positions of descent repell themselves is that if k is not a point of descent, then the digit on the position $k+1$ is likely to be big and hence likely to be a point of descent.

Points in a square

This time we want to built a DPP in a two dimensional square $[0,1]^2$ or at least a discrete approximation of it. This might be used to model the positions of trees that repel themselves due to a competition for natural resources, positions of football players on the field or the positions people choose for a picnic in a park.

In order to do this we follow an approach similar to the case of the one dimensional DPPs. Hence, we set

$$\mathcal{Y} := 99^{-1}\{0,\dots,99\}^2$$

and obtain a 100×100 grid covering the unit square. Again, we choose $\mathcal{R} := \mathcal{Y}$ and f to be the normal density with mean 0 and variance $\sigma > 0$. Then we choose the similarity feature vectors to be

$$(\phi_i)_j \propto f(\|i-j\|) \quad \text{for } i,j\in\mathcal{Y}.$$

where $\|\cdot\|$ is the Euclidean norm. We propose constant qualities just like before and scale them and also the variance σ in such a way that we get a reasonable cardinality and also a notable repulsion of the DPP. The resulting samples are depicted in Figure 1.4.

It should be noted at this stage, that the simulation of the DPP can still be performed in relatively short time, even without making use the dual sampling or the random projections. In fact, one sample could be produced on a 1.8GHz Intel Core i5 with 8GB RAM[9] in about two minutes. This is actually quite astonishing given that the DPP is a discrete probability distribution over $2^{10^4}\approx 10^{3000}$ elements. This is roughly the estimated number of elementary particles in the universe to the power of 35.

Assume now that we have some reason to believe that the qualities of the individual points on the grid are not equal. Maybe there might be a road just around the park and hence people prefer to sit in the middle of the park and we can implement this into our model by letting the qualities of the items decrease depending on their distance to the centre $m := (0.5, 0.5)$ of the grid. More

[9] I used a MacBook Air from 2012.

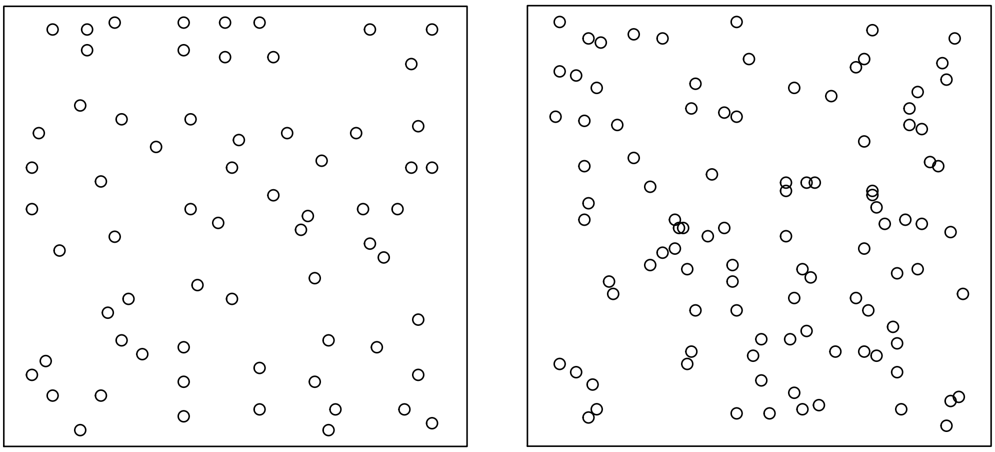


Figure 1.4: A DPP (right) and a Poisson point process (left) on a 100×100 grid in the unit square with the same expected cardinality. The – in this case spatially – repellent structure of the DPP is clearly visible.

precisely we choose

$$q_i := a \cdot \exp\big(-b\, \|i - m\|\big)$$

where $a, b > 0$ are scaled such that the decrease of quality is visible but not too strong and that a reasonable cardinality of the DPP is obtained. The results for this can be seen in Figure 1.5.

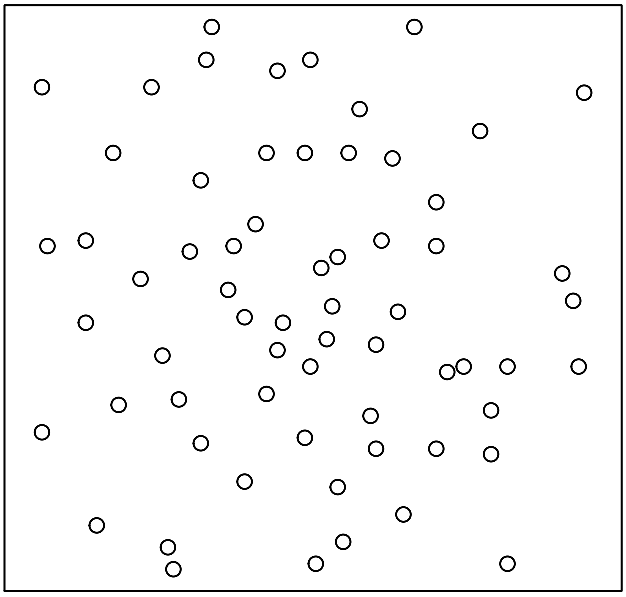


Figure 1.5: A DPP on a 100×100 grid in the unit square with decreasing qualities towards the edges.

Chapter II

Point estimators and parametric models

Parameter estimation is one of the central components of every theory of real world phenomena. In a nutshell one could split the process of the construction of a descriptive model into two parts. The first one being the selection of the model which is done by a scientist and the second one being the determination of the constants that belong to the model.

To make this more clear we will consider one of the most famous advances in the natural sciences, namely the law of universal gravitation that was discovered by Sir Isaac Newton and published in the *Philosophiæ Naturalis Principia Mathematica* (cf. [46]). More precisely, Newton discovered that the gravitational force acting between two massive objects is given by

$$F = G \cdot \frac{m_1 m_2}{r^2}$$

where m_1, m_2 are the masses of the two objects, r is the distance between the centers of masses and G is the gravitational constant. This constant can not be deduced from the theory itself and needs to be estimated based on some empirical data.

If we want to describe, simulate and predict the occurence of diverse subsets we can take a similar approach and impose the model of a determinantal point process. This will usually be an assumption that will not strictly hold, but will often lead to reasonable, sometimes even impressive results. We will not be concerned with how suitable this model selection is, although this is a highly interesting question. Leaving that aside we are left with the second step, namely the estimation of the parameters of the model, which are in the case of a DPP over a set of cardinality N exactly $N(N-1)/2$. Because of the rather large amount of parameters and also the complicated structure of the DPPs it will in practice only be possible to perform those estimations through the use of computational tools. The task of computer based parameter or density estimation is an important field in the discipline of *machine learning* and thus we will sometimes speak of the parameters being learned instead of estimated. Actually the interest of parameter estimation for DPPs arose from the machine learning community at the beginning of this decade. However, we will phrase things in a way that no prior knowledge in this field is required.

In this chapter we will be concerned in how we can make point estimates for either the marginal or the elementary kernels K and L. Point estimators are the most basic type of estimators and consist of the suggestion of one possible parameter, for example in the case of the gravitational constant

$$6.674 \cdot 10^{-11} \mathrm{N}\,\mathrm{kg}^{-2}\,\mathrm{m}^2.$$

This is in contrast to the Bayesian approach to parameter estimation that we will present in the next chapter where the philosophy is to estimate a distribution over all possible parameter sets that indicates how likely are to have caused the empirical data. We will discuss two essentially different methods of point estimators, the first one provides a way to reconstruct a marginal kernel for the empirical marginal distributions at least in the case where the empirical distribution is essentially a DPP. The other one being maximum likelihood estimation in different variations.

But before we can proceed we want to remind the reader of two desirable properties of point estimators. For this we will assume that we want to estimate the distribution of a random variable X from a parametric family of probability measures

$$\left\{ \mathbb{P}_\theta \mid \theta \in \Theta \right\}.$$

This means we want to estimate θ out of a possible set of parameters Θ such that X is distributed according to $\mathbb{P}_\theta$ which we will do based upon some data $x_1, \dots, x_n$. Further, we assume that those points are actually generated by $\mathbb{P}_{\theta_0}$ for one $\theta_0 \in \Theta$ and denote the estimator by $\hat{\theta}_n$. We call the estimator *unbiased* if we have

$$\mathbb{E}[\hat{\theta}_n] = \theta_0$$

and *consistent* if we have

$$\hat{\theta}_n \to \theta_0 \quad \text{in probability.}$$

It shall be noted that although those properties are beneficial, they are not crucial for an estimator to be reasonable. First they both assume that the data generating process, i.e. the random variables one wants to describe actually follows one of the laws $\mathbb{P}_{\theta_0}$ which will typically not be the case in real world examples. Further, the asymptotic property of consistency is rather of theoretical nature since in practice it is not possible to create large sets of empirical data and certainly not infinitely large ones.

II.1 Kernel reconstruction from the empirical measures

Now we will display the first way how one can estimate the marginal kernel K of a DPP based on some samples drawn from it.

2.1 Setting. Let $\mathcal{Y}$ be a finite set of cardinality N and let $K \in \mathbb{R}^{N\times N}_{\text{sym}}$ satisfy $0 \leq K \leq I$. Let further $\mathbf{Y}_1, \mathbf{Y}_2, \dots$ be independent and distributed according to a DPP with marginal kernel K.

In order to perform an approximate reconstruction of the marginal kernel we will consider the *empirical measures*

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

The interest in $\hat{\mathbb{P}}_n$ lies in the fact that they are quite natural estimates for the actual underlying distribution. In fact, they are unbiased estimators for $\mathbb{P}$ since for $A \subseteq \mathcal{Y}$ we have

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

Furthermore, by the strong law of large numbers they converge to $\mathbb{P}$ almost surely if the sequence $(\mathbf{Y}_k)_{k\in\mathbb{N}}$ of observations is independent. This can be seen by identifying the probability measures on $2^{\mathcal{Y}}$ with the probability simplex

$$\left\{ \mu \in \mathbb{R}^{2^{\mathcal{Y}}} \;\middle|\; \mu_A \in [0,1] \text{ for all } A \subseteq \mathcal{Y} \text{ and } \sum_{A\subseteq\mathcal{Y}} \mu_A = 1 \right\}$$

and using the strong law of large numbers in $\mathbb{R}^{2^{\mathcal{Y}}}$.

Therefore the empirical measures are reasonable approximations of the actual probability distribution. Assume now for one moment that the empirical measures $\hat{\mathbb{P}}_n$ are also determinantal point processes with marginal kernel $\hat{K}_n$, then $\hat{K}_n$ would be a quite intuitive estimate for the actual marginal kernel K. Thus, we are interested in the question whether we can reconstruct the marginal kernel of a DPP if we know the DPP itself. Since the marginal density of a DPP corresponds to the principal minors of the marginal kernel, we first investigate whether we can reconstruct a matrix from its principal minors. For the answer to this problem we follow the main ideas presented in [60] and [49] although we modify their arguments to make them shorter and hopefully more accessible.

2.2 The principal minor assignment problem. Let $K \in \mathbb{R}^{N\times N}$ be a symmetric matrix. We want to investigate whether K is uniquely specified by its principal minors

$$\varDelta_S := \det(K_S) \quad \text{where } S \subseteq \{1, \dots, N\}$$

and if so how it can be reconstructed from those. We call this the *symmetric principal minor assignment problem* (PMAP).

Before we present the general procedure we want to see how this would work in the case of a symmetric 3×3 matrix $K = (K_{ij})_{1\leq i,j\leq 3}$. First we note that we can regain the diagonal elements as the determinant of the 1×1 sub matrices

$$\det(K_{\{i\}}) = K_{ii} \quad \text{for } i = 1, 2, 3.$$

Further the squares of the off diagonal are determined by the 2×2 principal minors since

$$\det(K_{\{i,j\}}) = K_{ii}K_{jj} - K_{ij}^2 \quad \text{for } i, j = 1, 2, 3.$$

Therefore we only need to reconstruct the signs of the off diagonal entries. To do this, we consider the determinant of the matrix itself

$$\det(K) = K_{11}K_{22}K_{33} + 2K_{12}K_{13}K_{23} - K_{11}K_{23}^2 - K_{22}K_{13}^2 - K_{33}K_{12}^2. \tag{2.1}$$

Rearranging this yields

$$K_{12}K_{13}K_{23} = \frac{1}{2}\left(\det(K) + K_{11}K_{23}^2 + K_{22}K_{13}^2 + K_{33}K_{12}^2 - K_{11}K_{22}K_{33}\right).$$

Since we know all of the expressions on the right side, we can determine the sign of the product on the left side. Now we assign the signs of the off diagonal elements in such a way, that the above equation holds. More precisely if the product is negative, we assign a minus to one or all three elements, if it is positive, then we assign a minus to none or two elements. If the product is zero, every configuration of signs satisfy the desired property. It is now straight forward to check that this assignment actually leads to the desired principal minors.

II.1.1 Graph theoretical concepts

One major step in the general procedure will be a generalisation of the formula (2.1) for larger principal minors that will allow for the reconstruction of the signs. For this we will need the following graph theoretical concepts.

2.3 Notions from graph theory. Let $G = (V, E)$ be a finite graph, i.e. V is a finite set, called the *vertex set* and E consists of subsets of V with two elements, the *edges*. Sometimes we will be sloppy in notation and not distinguish between the graph and the edge set. We will need the following notions:

(*i*) *Degree:* For a vertex $v \in V$ the *degree* is the number of edges that contains v.

(*ii*) *Subgraph:* A graph $\tilde{G} = (\tilde{V}, \tilde{E})$ is called a *subgraph* of G if $\tilde{V} \subseteq V$ and $\tilde{E} \subseteq E$.

(*iii*) *Induced graph:* For a subset $S \subseteq V$ of vertices the *induced graph* $G(S) = (S, E(S))$ is formed of all edges $e \in E$ of G that are subsets of S.

(*iv*) *Path:* A *path* in G is a sequence $v_0v_1 \cdots v_k$ of vertices such that $\{v_{i-1}, v_i\} \in E$ for all $i = 1, \dots, k$.

(v) *Connected graph:* A graph is called *connected* if for every two vertices $v, w \in V$ there is a path from v to w.

(vi) *Cycle:* A *cycle* C is a connected subgraph such that every vertex has even degree in C.

(vii) *Cycle space:* Each cycle C can be identified with a vector $x = x(C) \in \mathbb{F}_2^E$ such that

$$x_e := \begin{cases} 1 & \text{if } e \in C \\ 0 & \text{if } e \notin C \end{cases}$$

indicates whether the edge $e \in E$ belongs to the cycle C. The *cycle space* $\mathcal{C}$ is the span of $\{x(C) \mid C \text{ is a cycle}\}$ in $\mathbb{F}_2^E$. Note that the sum of two cycles in the cycle space corresponds to the symmetric difference of the edges.

($viii$) *Simple cycle:* A cycle is called *simple* if every vertex of C has degree 2 in C.

(ix) *Chordless cycle:* A cycle C is called *chordless* if two vertices $v, w \in C$ form an edge in G if and only they form an edge in C. This is equivalent to the statement that C is an induced subgraph that is a cycle.

(x) *Cycle sparsity:* The cycle sparsity is the minimal number l such that a basis of the cycle space consisting of chordless simple cycles of length at most l exists. Such a basis is called *shortest maximal cycle basis* or short *SMCB*. If the cycle space is trivial we define the cycle sparsity to be 2.

(xi) *Pairings:* Let $S \subseteq V$ be a set of vertices. Then a *pairing* P of S is a subset of edges of $G(S)$ such that two different edges of P are disjoint. The vertices contained in the edges of P are denoted by $V(P)$ and the set of all pairings by $\mathcal{P}(S)$.

It is recommended to study the examples in Figure 2.1 and further ones in order to get more familiar with the definitions above. To see that the above definition of the cycle sparsity is well defined, we need to show that shortest maximal cycle bases exist. This might be well known to people familiar with graph theory, but we will present an elementary proof here. The first part of the statement, namely the existence of cycle basis consisting of simple cycles is known as Veblen's theorem and can be found in its original form in [64], however we will rather follow the approach in [6].

2.4 Proposition (Existence of SMCBs). *There always exists a basis $\{x(C_1), \dots, x(C_k)\}$ of the cycle space where $C_1, \dots, C_k$ are chordless simple cycles.*

Proof. First we prove that the set of simple cycles generates the whole cycle space which we can then improve to show that the simple chordless cycles already generate the cycle space. A shortest maximal cycle basis is then attained by successively dropping simple chordless cycles.

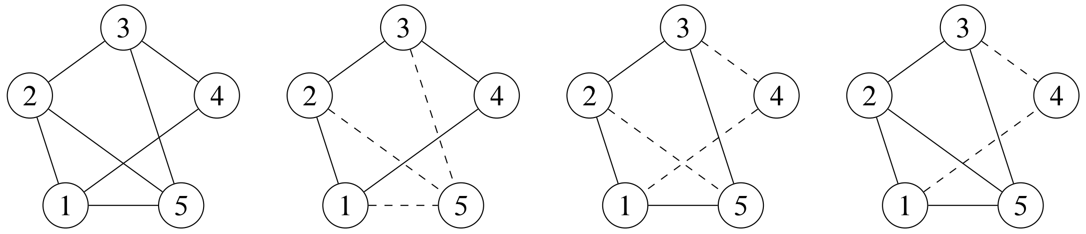


Figure 2.1: Some examples of graphs and cycles. The first sketch shows a graph and the three other ones subsgraphs of it where the edges not belonging to the subgraph are depicted dashed. The first one is a symple chordless cycle, the second one a simple but not chordless cycle and the last one is not a cycle at all.

We show that every cycle $x(C)$ can be written as the sum of simple cycles $x(C_1), \dots, x(C_k)$ where $C_i \subseteq C$ are disjoint. This is equivalent to the statement that the edges of every cycle are the disjoint union of the edges of simple cycles. To see that this is true, we fix a maximal non intersecting path $v_0 v_1 \cdots v_k$.[1] Since v_k has degree at least 2, there is an edge $\{v_k, v_{k+1}\}$ such that

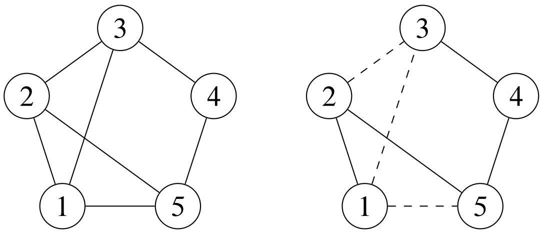


Figure 2.2: Illustration of the search for a simple cycle in a graph with degrees greater than two. Once a maximal non intersecting path like 12543 is selected, every continuation of the path – in this case 2 or 1 – is already present in the path and therefore induces a simple cycle.

$v_{k+1} \neq v_{k-1}$. Since the path is maximal, v_{k+1} has to agree with one vertex $v_i \in \{v_0, \dots, v_{k-2}\}$, because otherwise we could add v_{k+1} to the path which is a contradiction to the maximality. Now $v_i v_{i+1} \cdots v_k v_i$ corresponds to a simple cycle C_1 and $C_2 := C \setminus C_1$ is again a cycle. Thus, we can write C as the disjoint union $C = C_1 \cup C_2$ where C_1 is a simple cycle. By repeating this procedure we get the desired expression for C in terms of simple cycles.

To prove that the simple chordless cycles generate the cycle space we have to prove that we can write every simple cycle $x(C)$ as a sum of simple chordless cycles $x(C_1), \dots, x(C_k)$. Let $\big\{\{v_0, v_1\}, \dots, \{v_k, v_0\}\big\}$ be the edge set of C and assume that C is not chordless like in Figure 2.3, otherwise the statement would be trivial. Thus there is are indices $i, j \in \{1, \dots, k\}$ such that $\{v_i, v_j\} \in E$ but $\{v_i, v_j\} \notin C$. Let now C_1 and C_2 be the two cycles associated with the

[1] Non intersecting should be intuitive and means that $v_i \neq v_j$ for $i \neq j$.

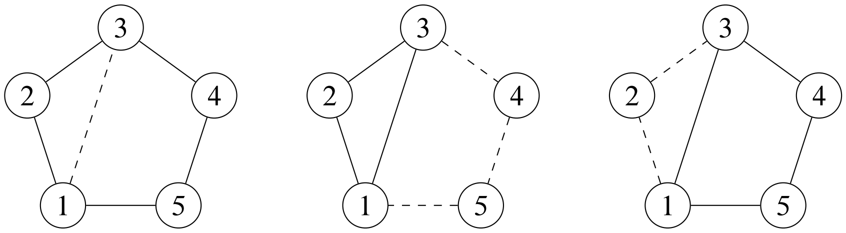


Figure 2.3: The simple cycle 123451 on the left is not chordless but the symmetric difference of the two simple chordless cycles 1231 and 13451 on the right.

paths

$$v_0 v_1 \cdots v_i v_j v_{j+1} \cdots v_k v_0 \quad \text{and } v_i v_{i+1} \cdots v_{j-1} v_j v_i.$$

Then we have $x(C) = x(C_1) + x(C_2)$. By iterating this procedure as long as the cycles are not chordless the desired decomposition can be achieved in finitely many steps.

□

II.1.2 The solution of the principal minor assignment problem

Now we have all the graph theoretical prerequisites to show how one can reconstruct a matrix with preassigned principal minors. However, the matrix that arises from this reconstruction is not unique and thus we need to identify matrices with the same principal minors with each other.

2.5 Definition (Determinantal equivalence). Two symmetric matrices $A, B \in \mathbb{R}^{N\times N}$ are called *determinantally equivalent* if they have the same principal minors and we write $A \sim B$.

2.6 Remark. It can be shown that two matrices $A, B \in \mathbb{R}^{N\times N}$ are determinantally equivalent if and only if there is a diagonal matrix D with diagonal entries ± 1 such that $A = DBD$. This is equivalent to the change of basis obtained by changing the signs of some basis elements. Since we will not need this result, we refer to Theorem 4.1 in [26] for a proof.

It is obvious that – given its principal minors – we can only hope to reconstruct a symmetric matrix up to determinantal equivalence. However, this would be satisfactory, because determinantally equivalent matrices are exactly those that give rise to the same DPP. On the other hand it is obvious that the solution of the principle minor assignment problem will be unique up to determinantal equivalence. Thus, the main work will be in showing that already the principal minors up to a certain size uniquely specify this equivalence class and further we see how the equivalence class can be constructed.

We notice – just like in the case of the 3 × 3 matrix – that the principal minors up to size two immediately determine the diagonal and the absolute values of the off diagonal of K since we have

$$K_{ii} = \Delta_{\{i\}} \quad \text{and } K_{ij}^2 = K_{ii}K_{jj} - \Delta_{\{i,j\}}.$$

Thus it only remains to regain the signs $\operatorname{sgn}(K_{ij})$ of the off diagonal entries. For this we use the following object.

2.7 The adjacency graph and sign function. The adjacency graph $G_K = (V_K, E_K)$ associated with K consists of the vertex set $\{1, \ldots, N\}$ and $\{i, j\}$ form an edge if and only if $K_{ij} \neq 0$. Further, we introduce *weights* on the edges. This means we consider a mapping $w \colon E_K \to \mathbb{R}$ and set

$$w_{ij} := w(\{i, j\}) := \operatorname{sgn}(K_{ij})$$

where we call w_{ij} the weight of the edge $\{i, j\}$. This graph together with the weights determines the signs of the off diagonal elements, and so we are interested in how we can reconstruct the weights from the principal minors. Finally we define the *sign* $\operatorname{sgn}(C)$ of a cycle $C = (S, E)$ to be

$$\operatorname{sgn}(C) := \prod_{e \in E} w_e.$$

It will become important later to consider this sign function on the cycle space and thus we note that this definition corresponds to

$$\operatorname{sgn}(x(C)) := \prod_{e \in E} w_e^{x(C)_e}.$$

Note that this is a group homomorphism from the cycle space $\mathcal{C}$ to $\{\pm 1\}$ and therefore it is uniquely determined by its values on a generator, for example on a shortest maximal cycle basis.

2.8 Proposition (Principal minors of simple chordless cycles). *Let $C = (S, E(S))$ be a simple and chordless cycle. Then the principal minor of K with respect to S is given by*

$$\Delta_S = \sum_{P \in \mathcal{P}(S)} (-1)^{|P|} \cdot \prod_{\{i,j\} \in P} K_{ij}^2 \cdot \prod_{i \notin V(P)} K_{ii} + 2 \cdot (-1)^{|S|+1} \cdot \prod_{\{i,j\} \in E(S)} K_{ij}. \tag{2.2}$$

Proof. Set $k := |S|$. Then by the Leibniz formula we have

$$\Delta_S = \sum_{\sigma \in S_k} \operatorname{sgn}(\sigma) \prod_{i \in S} K_{i\sigma(i)} \tag{2.3}$$

where S_k is the set of permutations of S. Note that the product is only non trivial if $\{i, \sigma(i)\} \in E(S)$ for all $i \in S$. Since C is a simple cycle, those permutations consist exactly of the pairing of S or the two shifts of the set S along the cycle in both directions. Those correspond exactly to the summands in (2.2).

To see this, we fix a permutation σ such that $\{i, \sigma(i)\}$ always forms an edge in $(S, E(S))$. We note that every vertex $i \in S$ has two possible images which are exactly the endpoint of its two edges, cf. Figure 2.4. Lets assume it is mapped to $j \in S$, then j has again two possible images under σ namely i and a second vertex $k \in \mathcal{Y}$. If $j \mapsto i$, no other vertex can be mapped to i or

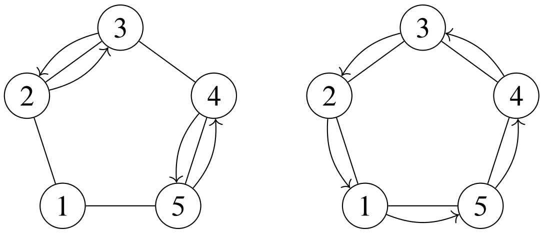


Figure 2.4: An easy example for the two kinds of permutations of a chordless simple cycle that maps vertices to neighbors.

j, however some other items can be swapped in the same way. The permutations of this form correspond exactly to the pairings of S and are represented in the first sum in (2.2). If however j is not mapped back to i but rather to its other neighbor k, then k can't get mapped back to j since σ is injective. Thus, it has to be mapped to its other neighbor $l \in \mathcal{Y}$. A repetition of this argument shows that this induces a cascade of mappings from vertices to their neighbors until i is reached again. Since the cycle is simple this path exhausts the entire cycle. The factor 2 is due to the fact that this shift of the indices can be done into either direction. □

2.9 Proposition (Sign determines principal minors). *The knowledge of all principal minors up to size two and the sign function*

$$\mathrm{sgn}\colon \mathcal{C} \to \{\pm 1\}$$

completely determines all principal minors of K.

Proof. Let $S \subseteq \{1, \dots, N\}$ be arbitrary. We will again work with the expression (2.3) of the principal minor Δ_S and fix one permutation σ. We can assume without loss of generality that $\{i, \sigma(i)\} \in E_K$ because the product is trivial otherwise. Since we know the absolute values of the off diagonal elements and the diagonal elements from the principal minors up to size two, it suffices to express the sign

$$\prod_{i \in S} \mathrm{sgn}(K_{i\sigma(i)}) \tag{2.4}$$

through the sign function on the cycle space. For this we write σ as the product of disjoint cycles[2]

$$\sigma = \sigma_1 \circ \dots \circ \sigma_m$$

where σ_k permutes D_k in cyclic fashion. The sign (2.4) is equal to the product of

$$\prod_{i \in D_k} \mathrm{sgn}(K_{i\sigma_k(i)}) \tag{2.5}$$

[2] A permutation $\sigma \in S_n$ is called a *cycle* if it maps some elements $i_1, \dots, i_k$ to each other in cyclic fashion, i.e. $i_l \mapsto i_{l+1}, i_k \mapsto i_1$ while fixing the other elements. Two cycles are called disjoint if the sets of elements that are not fixed are disjoint. Elementary considerations show that any permutation is the composition of disjoint cycles.

so it suffices to give expressions for those. Note that we could assume $\{i, \sigma_k(i)\} \in E_K$ and therefore $C_k = (D_k, E_k)$ with

$$E_k = \Big\{ \{i, \sigma_k(i)\} \mid i \in D_k \Big\}$$

is a cycle and thus (2.5) is equal to $\operatorname{sgn}(C_k)$. □

2.10 Theorem (Solution of the PMAP). *Let $K \in \mathbb{R}^{N \times N}$ be a symmetric matrix and l be the sparsity of its adjacency graph. Then the principal minors up to size l uniquely determine all principal minors of K and therefore the matrix K up to determinantal equivalence.*

Proof. In the light of the previous proposition it suffices to show that the sign function is uniquely specified by the principal minors up to size l. Recall that the sign function is determined by its values on a shortest maximal cycle basis, which consists by definition of simple chordless cycles of length at most l. However, under the knowledge of the diagonal elements and the absolute values of the off diagonal ones, the sign of those simple chordless cycle is uniquely determined by the principal minors up to size l using the equality (2.2) . □

2.11 Remark. One can even show that this result is optimal in the sense that if one only has access to the principal minors up to size $l - 1$, then the equivalence class is not uniquely determined. To see this, we note that the sign function is not uniquely specified through the principal minors up to size $l - 1$ and thus there is more than one extension of the sign function onto the shortest maximal cycle basis. The equation (2.2) shows that those different extensions give rise to different principal minors.

2.12 Construction of the equivalence class. We have shown that the determinantal equivalence class of a symmetric matrix is uniquely specified by its principal minors up to size l. Now we want to investigate how this equivalence class can be computed and we will see that we can reduce this task to the solution of a system of linear equations over the finite field $\mathbb{F}_2$.

Let us assume that we have knowledge of the principal minors Δ_S for every $S \subseteq \{1, \dots, N\}$ with size at most l and we want to construct a matrix $\tilde{K}$ that is determinantally equivalent to K. We have seen that we only need to reconstruct the signs of the off diagonal entries of K which is equivalent to reconstructing the edge weights w_{ij}. To do this, we fix a shortest maximal cycle basis $\{C_1, \dots, C_m\}$ with vertex sets $S_1, \dots, S_m$. Let us now rewrite (2.2) in the form

$$H_k := \Delta_{S_k} - \sum_{P \in \mathcal{P}(S_k)} (-1)^{|P|} \cdot \prod_{\{i,j\} \in P} K_{ij}^2 \cdot \prod_{i \notin V(P)} K_{ii} = 2 \cdot (-1)^{|S_k|+1} \operatorname{sgn}(C_k) \cdot \prod_{\{i,j\} \in C_k} \big| K_{ij} \big| .$$

Given the principal minors, we can determine the value on the right side and comparing the signs of both sides yields

$$(-1)^{|S_k|+1} \cdot \operatorname{sgn}(H_k) = \operatorname{sgn}(C_k) = \prod_{\{i,j\} \in C_k} w_{ij}$$

which we seek to solve for w. However, this multiplicative equation is hard to solve and therefore we use the canonical group isomorphism ϕ between $\{\pm 1\}$ and $\{0, 1\}$ to turn it into a linear equation.[3] Setting $x_{ij} := \phi(w_{ij})$ we get that the condition above is equivalent to

$$b_k := \phi(\operatorname{sgn}(H_k)) + |S_k| + 1 = \sum_{\{i,j\}\in C_k} x_{ij} = (Ax)_k \quad \text{in } \mathbb{F}_2$$

where A is the matrix with the rows $x(C_k)^T$. Now we can fix any such solution $x \in \mathbb{F}_2^E$ of

$$Ax = b \tag{2.6}$$

and we know that at least one exists, namely the one given by $x_{ij} = \phi(\operatorname{sgn}(K_{ij}))$. Let now $w_{ij} := \phi^{-1}(x_{ij})$, then it is straight forward to see that $\tilde{K}$ defined through

$$\tilde{K}_{ii} := \Delta_{\{i\}} \quad \text{and } \tilde{K}_{ij} = w_{ij} \cdot \sqrt{\tilde{K}_{ii}\tilde{K}_{jj} - \Delta_{\{i,j\}}}$$

is determinantally equivalent to K.

It shall be noted that there are algorithms with possibly better computational performance for the construction of the determinantal equivalence class. For some examples of efficient algorithms we refer to [60] and [49].

II.1.3 Definition of the estimator and consistency

So far we have seen that the principal minors determine a symmetric matrix up to determinantal equivalence. However, the empirical marginal densities do not in general need to be the principal minors of any symmetric matrix, in other words the empirical measures are not necessarily determinantal. Therefore, the definition of the estimator is still not straight forward and we will follow [60] for this and make the following assumption.

2.13 Assumption. Fix $\alpha > 0$ and assume from now on that

$$\min\Big\{ |K_{ij}| \;\Big|\; K_{ij} \neq 0 \Big\} \geq \alpha.$$

Note that such an α can always be found, however it is not a priori known and hence we have to postulate it.

2.14 Definition of the estimator. The straight forward estimators for the principal minors are

$$\hat{\Delta}_S := \hat{\mathbb{P}}_n(S \subseteq \mathbf{Y}) \quad \text{for } S \subseteq \{1, \dots, N\}.$$

[3] The isomorphism ϕ has the explicit structure $1 \mapsto 0, -1 \mapsto 1$.

The resulting estimates for the diagonal elements and the squares of the off diagonals are

$$\hat{K}_{ii} := \hat{\Delta}_{\{i\}} \quad \text{and } \hat{B}_{ij} := \hat{K}_{ii}\hat{K}_{jj} - \hat{\Delta}_{\{i,j\}}.$$

Next we will introduce an estimate $\hat{G} = (\hat{E}, \hat{V})$ for the adjacency graph and we choose the edge set $\hat{E}$ to consist of all sets $\{i, j\}$ such that $\hat{B}_{ij} \geq \frac{1}{2}\alpha^2$. We will see that this truncation leads to the almost surely convergence of the graph $\hat{G}$ towards G. In analogy to the previous paragraph 2.12 we define $\{\hat{C}_1, \dots, \hat{C}_{\hat{m}}\}$, $\hat{H}_1, \dots, \hat{H}_{\hat{m}}$, $\hat{A}$ and $\hat{b}$ exactly the same way. If there is a solution $\hat{x} \in \mathbb{F}_2^E$ to the linear equation

$$\hat{A}\hat{x} = \hat{b}, \tag{2.7}$$

then we estimate the signs to be $\hat{w}_{ij} := \phi^{-1}(\hat{x}_{ij})$ and define

$$\hat{K}_{ij} := \hat{w}_{ij}\sqrt{\hat{B}_{ij}}.$$

If there is no such solution $\hat{x}$ then we simply set the signs of the off diagonal elements to be positive, i.e. we define

$$\hat{K}_{ij} := \sqrt{\hat{B}_{ij}}.$$

This choice is completely arbitrary, but we will see in the consistency result below that the probability for this case tends to zero as the sample size increases.

In order to talk about consistency of the estimator that we constructed above, it is necessary to define a metric on the marginal kernels of DPPs. However, the usual operator norm is clearly not right for this job, since we already know that we can only hope to reconstruct the determinantal equivalence class but not the exact marginal kernel. Thus, we will work with the usual choice of pseudometric if one has to deal with equivalence classes.

2.15 Pseudometric on the marginal kernels. We define the distance between two marginal kernels $A, B \in \mathbb{R}^{N\times N}$ through

$$d(A, B) := \inf_{C\sim A} \|B - C\|_\infty$$

where $\|A\|_\infty := \max_{1\leq i,j\leq N} \left|A_{ij}\right|$ denotes the uniform norm on the space of matrices.

2.16 Theorem (Consistency). *Let K be the marginal kernel of a DPP that satisfies Assumption 2.13. Then we have for any $\varepsilon > 0$*

$$\mathbb{P}\left(d(\hat{K}, K) \leq \varepsilon\right) \to 1 \quad \textit{for } n \to \infty.$$

Proof. We will keep the notations from the paragraphs 2.12 and 2.14. We have already seen in the motivation of this section that the empirical measures converge almost surely which directly implies

$$\hat{K}_{ii} \to K_{ii} \quad \text{and } \hat{K}_{ij}^2 \to K_{ij}^2 \quad \text{almost surely.} \tag{2.8}$$

Since almost sure convergence implies convergence in probability we have

$$\mathbb{P}(\hat{G} = G_K) = \mathbb{P}\left(\hat{K}_{ij}^2 \geq \alpha^2/2 \text{ for } K_{ij} \neq 0\right) \to 1 \quad \text{for } n \to \infty.$$

In this case the two shortest cycle basis $\{\hat{C}_1, \dots, \hat{C}_{\hat{k}}\}$ and $\{C_1, \dots, C_k\}$ can be chosen the same and so $\hat{A}$ and A agree. Because of (2.8) we also have $\hat{H}_k \to H_k$ almost surely and thus $\hat{b}_k \to b_k$ almost surely for all k. This yields

$$\mathbb{P}\left(\hat{A} = A \text{ and } \hat{b} = b\right) \to 1 \quad \text{for } n \to \infty. \tag{2.9}$$

In this case the two linear quations (2.6) and (2.7) agree and therefore $\tilde{K} \in \mathbb{R}^{N\times N}$ defined through $\tilde{K}_{ij} := \hat{w}_{ij} \left|K_{ij}\right|$ is determinantally equivalent to K. Further, we know that for $\delta > 0$

$$\mathbb{P}\left(\left|\hat{K}_{ij}^2 - \tilde{K}_{ij}^2\right| < \delta \text{ for all } i, j\right) \to 1 \quad \text{for } n \to \infty. \tag{2.10}$$

If (2.9) and (2.10) hold then we have $\operatorname{sgn}(\hat{K}_{ij}) = \operatorname{sgn}(\tilde{K}_{ij})$ as well as $\left|\hat{K}_{ij}^2 - \tilde{K}_{ij}^2\right| < \delta$ and hence

$$d(\hat{K}, K) \leq \|\hat{K} - \tilde{K}\|_\infty < \varepsilon$$

for δ small. □

2.17 Remark (Speed of convergence). Although the result above states that the estimators $\hat{K}$ converges to K in probability, it doesn't give any information about the speed of convergence. This problem is addressed in [60], but it turns out that the convergence is very slow. For example for the very moderate case $\alpha = 0.4$ and $l = 3$ one already needs more than 10^6 samples to get some theoretical guarantees from their result. This is not due to careless estimates since they show that you need very high sample sizes in order to ensure that the estimator is close to the actual kernel with probability bigger than $\frac{2}{3}$. In practice such sample sizes can almost never be achieved.

2.18 Computation of the estimator. Although we have already seen how the estimator can be constructed theoretically, we will now touch on the implementation details for the actual computation of the estimator. In fact the only two non trivial steps in the definition of the estimator are the construction of a shortest maximal cycle basis and the solution of the linear equation (2.7) over the finite field $\mathbb{F}_2$. There have been various algorithms proposed for the computation of a shortest maximal cycle basis and we refer to the original work of Horton [22] and a recent improvement of his algorithm in [2]. Further, we note that the linear equation can be solved just like every linear equation over any field using Gaussian elimination.

II.2 Maximum likelihood estimation

The method of maximum likelihood estimation (MLE) 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 in order to present the general procedure we follow the corresponding section in [48].

Suppose we have given some candidates $f(x_1,\dots,x_n|\theta)$ for the joint density of some random variables $X_1,\dots,X_n$ with respect to some reference measure $\prod_{i=1}^n \mu(\mathrm{d}x_i)$ and we want to decide which parameter $\theta \in \Theta$ describes the realisations $x_1,\dots,x_n$, which we will also call data or observation, best. Hence, it is reasonable to pick θ under which the observations $x_1,\dots,x_n$ are 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

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

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.

2.19 Definition (Maximum likelihood estimator). Let Θ be a set, which we call the *parameter set* and let

$$\mathcal{F} = \Bigl\{f(\cdot|\theta)\colon \mathcal{X} \to [0,\infty) \bigm| \theta \in \Theta\Bigr\}$$

be a family of probability densities with respect to some measure μ on some measurable space $\mathcal{X}$. We call the function

$$\mathcal{L}\colon \Theta \to [-\infty, 0], \quad \mathcal{L}(\theta) := \sum_{i=1}^n \log(f(x_i \mid \theta))$$

the *log likelihood function* associated with the observations $x_1,\dots,x_n$ and its maximiser

$$\hat{\theta}_n := \operatorname*{arg\,max}_{\theta\in\Theta} \mathcal{L}(\theta) \tag{2.11}$$

the *maximum likelihood estimator* or short *MLE.*

Very short reminder on optimisation

Since the calculation of the MLE is a maximisation task, it is suitable to review some general properties of optimisation problems. For this let $U \subseteq \mathbb{R}^M$ and $f : U \to \mathbb{R}$ be a function. In practice the maximisation

$$\hat{x} := \arg\max_{x \in U} f(x)$$

will usually not be explicitly solvable and therefore one has to exploit numerical algorithms.

Those work particularly well if the function f is concave and possibly smooth and one powerful method is given by the so called gradient descent. To quickly explain its philosophy, we note that ∇f points into the direction of the steepest ascent of the function f and thus an intuitive approach the maximise f would be to follow the gradient, i.e. to take a solution γ of the gradient flow $\gamma' = \nabla f(\gamma)$ and work out its limit. However, if the function is not concave one can not even guarantee that the gradient flow reaches a local minimum, since one can construct examples where γ gets stuck in a critical point. However, in the concave case this suffices since critical points and global minima agree for concave functions. The gradient descent is an algorithm derived from this observation and is essentially a discretisations of the gradient flow meaning that it iteratively takes small steps into the direction of the gradient and thus lowers the value of the function. Some more sophisticated versions of gradient descent methods even consider higher order derivatives and use the information they provide over the geometry of the graph. Generally speaking those algorithms work extremely well even in high dimensions and thus their efficiency and stability have been studied broadly and we refer to the extensive monograph [10]. All together we note that concavity is an extremely favourable property for a function that shall be maximised, which will be the log likelihood function later on.

A second property which is important in the existence theory of maximisers is the *coercivity* of the function in the sense that

$$f(x) \to -\infty \quad \text{for } |x| \to \infty.$$

In fact every (upper semi-) continuous and coercive function defined on a closed set $U \subseteq \mathbb{R}^M$ attains its minimum. To see this one can fix $x_0 \in U$ and use the coercivity to obtain $f < f(x_0)$ outside of a compact set K and thus the supremum of f agrees with the supremum of f over K which it is attained. We will later introduce some abstract theory about the consistency of estimators and for this we will need this result in a more general setting. However, the version above is enough in the case of the maximum likelihood estimators for parameters of DPPs and therefore readers that are not familiar with elementary notions of topology are advised to neglect the following statement.

2.20 Proposition (Existence of maximisers). *Let $\mathcal{X}$ be a topological Hausdorff space and $f : \mathcal{X} \to [-\infty, \infty)$ be an* upper semicontinuous *function, i.e.*

$$L_f(\alpha) := \{x \in \mathcal{X} \mid f(x) \geq \alpha\}$$

is closed for all $\alpha \in \mathbb{R}$. Further, we will assume that f is coercive*, meaning that for any $\alpha \in \mathbb{R}$ the set $L_f(\alpha)$ is compact. Then f attains its maximum in at least one point, i.e. there is $\hat{x} \in \mathcal{X}$ such that*

$$f(\hat{x}) = \sup_{x \in \mathcal{X}} f(x).$$

Proof. Let without loss of generality f be not identical to $-\infty$ because otherwise the statement is trivial. Then we have

$$\alpha := \sup_{x \in \mathcal{X}} f(x) > -\infty.$$

If we choose (α_n) to be strictly increasing towards α, then we get $L_f(\alpha_{n+1}) \subseteq L_f(\alpha_n)$ for all $n \in \mathbb{N}$ and further none of the sets $L_f(\alpha_n)$ is empty. By the Cantor intersection theorem[4] we get that also the intersection is non empty, i.e there is

$$\hat{x} \in \bigcap_{n \in \mathbb{N}} L_f(\alpha_n).$$

This implies

$$f(\hat{x}) \geq \alpha_n \xrightarrow{n \to \infty} \alpha = \sup_{x \in \mathcal{X}} f(x).$$

□

II.2.1 Presentation of different models

Let in the following $(\mathbf{Y}_n)_{n \in \mathbb{N}}$ be a sequence of independent random variables distributed according to a DPP. In order to follow the MLE approach, we need to express the density of the point process with respect to some measure and we will do this by giving the elementary probabilities which are nothing but the densities with respect to the counting measure. Thus, we will assume that we are dealing with L-ensembles in this section. Since the observations $(\mathbf{Y}_n)_{n \in \mathbb{N}}$ are defined on some common probability space which we will denote by $(\Omega, \mathbb{P})$, we will write

$$f(A|\theta) = \frac{\det(L(\theta)_A)}{\det(L(\theta) + I)}$$

for the elementary probabilities of the DPP that arises from the parameter θ. We will now present the maximum likelihood estimators for different parametric classes.

[4] A precise formulation can be found in the appendix.

MLE of the elementary kernel L

The most intuitive parameter that one can estimate is the elementary kernel L itself since it parametrises the entire class of L-ensembles.

2.21 Maximum likelihood estimator for L. We consider the parameter space $\Theta = \mathbb{R}^{N\times N}_{\text{sym},+}$ of positive semi-definite symmetric matrices and the parametric family

$$\mathcal{F} = \left\{ f(\cdot|L) \;\middle|\; L \in \mathbb{R}^{N\times N}_{\text{sym},+} \right\}$$

where $f(A|L) \propto \det(L_A)$ is the elementary probability of DPP with elementary kernel L. We seek to find the MLE

$$\hat{L}_n := \operatorname*{arg\,max}_{L\in\mathbb{R}^{N\times N}_{\text{sym},+}} \mathcal{L}(L).$$

The log likelihood function is now given by

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

Using (1.4) we get the expression

$$\mathcal{L}(L) = \sum_{i=1}^{n} \log\big(\det(L_{\mathbf{Y}_i})\big) - n\log\big(\det(L+I)\big) \tag{2.12}$$

which is upper semi-continuous in L. Although the parametric family that arises from the elementary kernels L parametrises the whole class of L-ensembles and therefore gives a high variety of different associated L-ensembles, we will see that it makes the computation of the MLE more complex. Therefore, we introduce smaller classes of L-ensembles, which will decrease the flexibility of the model, but make computation more efficient.

MLE of the qualities

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

$$(q, \phi) \mapsto \Psi(q, \phi) = L \quad \text{where } L_{ij} = q_i \phi_i^T \phi_j q_j.$$

Now we fix a diversity feature matrix $\hat{\phi}$, that we will usually model according to some perceptions we might have and set $\hat{S}_{ij} := \phi_i^T \phi_j$. Now we aim to estimate the quality vector $q \in \mathbb{R}^N_+$ instead of the whole kernel L. This means that we optimise the likelihood function over a smaller set of kernels, namely the ones of the form $\Psi(q, \hat{\phi})$ 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

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

2.22 Maximum likelihood estimator for the quality. This time we work with the parameter set $\Theta = \mathbb{R}_+^N$ and the parametric family

$$\mathcal{F} = \left\{ f(\cdot|q) \mid q \in \mathbb{R}_+^N \right\}$$

where $f(A|q) \propto \det(\Psi(q,\hat{\phi})_A)$ is the elementary probability of DPP with elementary kernel $\Psi(q,\hat{\phi})$. We aim to find the MLE of the quality vector $q \in \mathbb{R}_+^N$, in other words we set

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

where we perceive the likelihood function as a function of q.

Using (1.5) we obtain the following expression for the single summands of the log likelihood function

$$\log\left(\prod_{j\in Y_i} q_j^2\right) + \log(\det(\hat{S}_{Y_i})) - \log\left(\sum_{A\subseteq\mathcal{Y}}\prod_{j\in A} q_j^2 \det(\hat{S}_A)\right) \tag{2.13}$$

and note that it is upper semicontinuous.

Log linear model for the qualities

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. Unfortunately we can tell from (2.13) that the log likelihood still isn't concave in q and in order to achieve this, we will introduce the following model for the qualities.

2.23 Log linear model for the qualities and MLE. 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

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

and will only consider quality vectors $q \in \mathbb{R}_+^N$ that have this form. To formulate the maximum likelihood estimator for θ we set $\Theta := \mathbb{R}^M$ and consider the parametric family

$$\mathcal{F} = \left\{ f(\cdot|\theta) \mid \theta \in \mathbb{R}^M \right\}$$

where $f(\cdot|\theta)$ is the density of the DPP with diversity feature matrix $\hat{\phi}$ and the according qualities $q_i = \exp\left(\frac{1}{2}\theta^T f_i\right)$. Further, we will consider the maximum likelihood estimator

$$\hat{\theta}_n := \arg\max_{\theta\in\mathbb{R}^M} \mathcal{L}(\theta)$$

where we regard $\mathcal{L}$ again as function of θ. We will assume that the feature vectors f_i span the whole space $\mathbb{R}^M$ because otherwise we can simple work with the projections of the parameter θ onto the span of the feature vectors and obtain an equivalent model.

2.24 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 is just a logarithmic transformation of the parameters and thus the maximal likelihood that can be achieved with this model does not change. In practice it will be of interest to work with rather low dimensional parameters θ, because if the ground set $\mathcal{Y}$ gets large, optimisation in $\mathbb{R}^N$ 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.

Again, we can express the log likelihood function explicitely and note that it is upper semi-continuous. In fact, it takes the form

$$2 \cdot \theta^T \sum_{i \in Y} f_i + \det(\hat{S}_Y) - \log\left(\sum_{A \subseteq \mathcal{Y}} \exp\left(2 \cdot \theta^T \sum_{i \in A} f_i\right) \det(\hat{S}_A)\right). \tag{2.14}$$

II.2.2 Coercivity and existence of the maximum likelihood estimators

A priori it is not clear that the maximum likelihood estimators exist and we will actually see that they do not exist in general. However not everything is lost since we will show that the probability that they exist tends to one for increasing sample size. We will also shortly discuss a second method how one can slightly adjust the concept of MLE to obtain the general existence of the estimator which we will do by a regularisation term.

MLE of the qualities

The MLE $\hat{q}_n$ does not exist for all realisations $(Y_n)_{n \in \mathbb{N}}$ of $(\mathbf{Y}_n)_{n \in \mathbb{N}}$. To see this, we suppose that we have only one sample $Y_1 = \mathcal{Y}$ which is the whole set. The higher the qualities of the items are, the more likely this observation gets and therefore the maximum of the log likelihood function – which is 0 in this case – is not obtained. This can also be made rigorous in the following computation. For this we assume that $\det(\hat{S}_{\mathcal{Y}}) > 0$ and that the qualities are constant, then the log likelihood function takes the form

$$\log\left(q^{2N} \det(\hat{S}_{\mathcal{Y}})\right) - \log\left(\sum_{A \subseteq \mathcal{Y}} q^{2|A|} \det(\hat{S}_A)\right) = \log\left(\frac{q^{2N} \det(\hat{S}_{\mathcal{Y}})}{\sum_{A \subseteq \mathcal{Y}} q^{2|A|} \det(\hat{S}_A)}\right) \xrightarrow{q \to \infty} 0.$$

However this maximum is never attained, since for every L-ensemble we have $\mathbb{P}_L(\varnothing) > 0$ and therefore

$$\mathcal{L}(q) = \log\left(\mathbb{P}_{\Psi(q,\hat{S})}(\mathcal{Y})\right) < 0 \quad \text{for every } q \in \mathbb{R}_+^N.$$

The thing that goes wrong in this case is, that under the observation of the whole set $\mathcal{Y}$ we would estimate a deterministic model that always selects the whole set, namely the DPP with marginal kernel I. Since all of the eigenvalues are 1 in this case, this DPP is not a L ensemble and therefore we can not describe it with the quality diversity decomposition. However if we assume that the data is actually generated by a L-ensemble, then such a scenario becomes unlikely as the sample size increases. We will fix this in the following result.

2.25 PROPOSITION (COERCIVITY AND EXISTENCE OF THE MLE). *Let* $\mathbf{Y}_1, \mathbf{Y}_2, \ldots$ *be a sequence of independent and identically distributed point processes that belong to the class of L-ensembles. Then we have*

$$\mathbb{P}\left(\hat{q}_n \in \mathbb{R}_+^N \textit{ exists}\right) \ge \mathbb{P}\big(\mathcal{L} \textit{ is coercive}\big) \xrightarrow{n\to\infty} 1.$$

Proof. The first inequality follows from Proposition 2.20 since the log likelihood function is upper semicontinuous. We will show that $\mathcal{L}$ is coercive if one of the observations is the emptyset. Then the claim follows from

$$\begin{aligned}\mathbb{P}\big(\mathcal{L} \text{ is coercive}\big) &\ge \mathbb{P}\left(\bigcup_{i=1}^n \{\mathbf{Y}_i = \varnothing\}\right) = 1 - \mathbb{P}\left(\bigcap_{i=1}^n \{\mathbf{Y}_i \neq \varnothing\}\right) \\ &= 1 - \mathbb{P}(\mathbf{Y}_1 \neq \varnothing)^n \xrightarrow{n\to\infty} 1\end{aligned}$$

since we have $\mathbb{P}(\mathbf{Y}_1 \neq \varnothing) < 1$ for every L-ensemble.

So let $Y_1, \ldots, Y_n$ be some observations with $Y_i = \varnothing$ for at least one $i \in \{1, \ldots, n\}$ and let $(q^k)_{k\in\mathbb{N}} \subseteq \mathbb{R}_+^N$ be a sequence such that $|q^k| \to \infty$. Note that it suffices to show that every subsequence of (q^k) contains a subsubsequence (q^l) such that

$$\mathcal{L}(q^l) \to -\infty \quad \text{for } l \to \infty.$$

Hence we fix a subsequence of (q^k) which we denote by (q^k) again in slightly abusive notation. Let (q^l) be a subsequence of (q^k) such that one coordinate diverges to infinity, i.e.

$$q_{j_0}^l \xrightarrow{l\to\infty} \infty \quad \text{for one } j_0 \in \{1, \ldots, N\}.$$

The i-th summand of $\mathcal{L}$ takes the form

$$-\log\left(\sum_{A\subseteq\mathcal{Y}} \prod_{j\in A} (q_j^l)^2 \det(\hat{S}_A)\right) \le -\log\left((q_{j_0}^l)^2\right) \xrightarrow{l\to\infty} -\infty$$

where we used $\hat{S}_{\{j_0\}} = 1$. Because the other summands are non positive this implies

$$\mathcal{L}(q^l) \xrightarrow{l\to\infty} -\infty$$

which we had to show. □

2.26 Remark. The proof above should be read in the following way. The statement $q^l_{j_0} \to \infty$ is equivalent to a model that would always select the item j_0. However, since we have observed the empty set, the observations would be impossible under this model and thus the log likelihood function takes the value $-\infty$ for this model.

2.27 Proposition (Positivity of the MLE). *Assume that $\mathbf{Y}_1, \mathbf{Y}_2, \dots$ is a sequence of independent and identically distributed point processes that are distributed according to a L-ensemble with strictly positive qualities. Then we have*

$$\mathbb{P}\left(\hat{q}_n \in \mathbb{R}^N_+ \text{ exists and } \hat{q}_n \in (0,\infty)^N\right) \xrightarrow{n\to\infty} 1.$$

Proof. We have already seen that the probability that the MLE exists tends to one, so we only have to show that the probability that the estimated qualities are strictly positive tends to one. The approach to prove this is exactly the same than in the proof of existence. Indeed we note that once j occurs in one of the observations $Y_1, \dots, Y_n$ we have $\mathcal{L}(q) = -\infty$ for every $q \in \mathbb{R}^N_+$ with $q_j = 0$. Therefore, we have $(\hat{q}_n)_j > 0$ if $j \in Y_i$ for at least one $j \in \{1, \dots, n\}$. Finally we note that the probability that j occurs in the i-th sample is strictly positive since we have

$$\mathbb{P}(j \in \mathbf{Y}_i) \geq \mathbb{P}(\{j\} = \mathbf{Y}_i) = q_j^2 > 0.$$

□

MLE of the elementary kernel

We can quite easily adapt the proof for the existence of MLEs of the qualities to the case of MLEs for the whole elementary kernel L.

2.28 Proposition (Coercivity and existence of MLE). *Let $\mathbf{Y}_1, \mathbf{Y}_2, \dots$ be asequence of independent and identically distributed point processes that fall in the class of L-ensembles. Then we have*

$$\mathbb{P}\left(\hat{L}_n \in \mathbb{R}^{N\times N}_{sym,+} \text{ exists}\right) \geq \mathbb{P}\big(\mathcal{L} \text{ is coercive}\big) \xrightarrow{n\to\infty} 1.$$

Proof. Again it suffices to show $\mathcal{L}(L) \to -\infty$ for $|L| \to \infty$ once we have observed the empty set once. To see this, we use the quality diversity parametrisation

$$\Psi\colon \mathbb{R}^N_+ \times \mathbb{S}^N_N \to \mathbb{R}^{N\times N}_{\text{sym},+}, \quad (q,\phi) \mapsto \left(q_i \phi_i^T \phi_j q_j\right)_{1\leq i,j\leq N}.$$

Note that since Ψ is continuous and $\mathbb{S}_N^N$ is bounded, $|\Psi(q,\phi)| \to \infty$ implies $|q| \to \infty$. The exact same calculations as in the previous proof show

$$\mathcal{L}(L) = \mathcal{L}(\Psi(q,\phi)) \to -\infty \quad \text{for } |L| \to \infty.$$

□

Coercivity for the log linear model

We proceed just like before.

2.29 Proposition (Coercivity and existence of MLE). *Assume that* $\mathbf{Y}_1, \mathbf{Y}_2, \dots$ *is a sequence of independent and identically distributed point processes that are distributed according to a* L*-ensemble with strictly positive qualities. Then we have*

$$\mathbb{P}\left(\hat{\theta}_n \in \mathbb{R}^M \textit{ exists}\right) \geq \mathbb{P}\Big(\mathcal{L} \textit{ is coercive as a function on } U\Big) \xrightarrow{n\to\infty} 1.$$

Proof. Again we show that $\mathcal{L}$ is coercive on $\mathbb{R}^M$ whenever we have observed the emptyset as well as every item at least once. Let now $(\theta^k)_{k\in\mathbb{N}} \subseteq U$ be a sequence such that $|\theta^k| \to \infty$. Then there is at least one index $i \in \{1,\dots,N\}$ and a subsequence $(\theta^l)_{l\in\mathbb{N}}$ such that

$$f_i^T\theta^l \to \infty \quad \text{or } f_i^T\theta^l \to -\infty \quad \text{for } l \to \infty$$

since otherwise all sequences $\left(f_i^T\theta^l\right)$ therefore also (θ^l) would be bounded. However, this is equivalent to

$$\exp(f_i^T\theta^l) \to \infty \quad \text{or } \exp(f_i^T\theta^l) \to 0 \quad \text{for } l \to \infty$$

and we have seen in the proof of 2.27 that the log likelihood function tends to $-\infty$ in this case. □

MLE with regularisation or MAP estimation

We have seen that the probability that the MLE exists tends to one if the sample size goes to infinity. However, it might not be possible in practice to obtain larger data sets and therefore we introduce a variation of maximum likelihood estimation which forces the existence of a maximiser. The idea is to add a coercive function to the log likelihood function such that the sum is coercive and to optimise this sum.

2.30 Definition (MAP estimation). Let the setting be the same as for the normal maximum likelihood estimation. Further we assume that we have given a function $R\colon \Theta \to [-\infty, 0]$ which we call the *regulariser*. The *regularised MLE*, *maximum a posteriori probability* or shortly *MAP estimator*[5] is the maximiser

$$\hat{\theta}_n := \operatorname*{arg\,max}_{\theta\in\Theta}\big(\mathcal{L}(\theta) + R(\theta)\big).$$

[5] The term maximum posteriori probability will only properly make sense once we introduce the Bayesian setting in the next chapter. We will see there that the regularised MLE is nothing but the mode of the posterior density.

2.31 Remark. (*i*) The regularised maximum likelihood approach is clearly an extension of the classical approach since one can simply set $R = 0$.

(*ii*) If the regulariser is upper semi-continuous and coercive, then the MAP estimator for DPPs always exists. In fact, $\mathcal{L} \le 0$ implies that $\mathcal{L} + R$ is coercive and also upper semi-continuous since the log likelihood functions are upper semi-continuous for all parametric models and 2.20 yields the assertion.

(*iii*) The regularised MLE and the MLE can agree but don't necessarily agree.

(*iv*) The regulariser can be used to encode any prior conceptions one has. For example one can use the regulariser to change the parameter set. Indeed, if we want to consider all matrices $\mathbb{R}^{N\times N}$ as a parameter space instead of only the symmetric positive semi-definite matrices, we can simply set

$$R(L) := \begin{cases} 0 & \text{if } L \text{ symmetric and positive semi-definite} \\ -\infty & \text{otherwise} \end{cases}$$

which is equivalent to the MLE on the smaller parameter space. The advantage of $\mathbb{R}^{N\times N}$ as a parameter space is that one can make use of pre-implemented optimisation algorithms, since they are usually defined over real vector spaces.

(*v*) We will see in the last chapter how different choices of the regulariser affect the estimator.

II.2.3 Consistency of the maximum likelihood estimators

We will now turns towards the question of consistency of the maximum likelihood estimators introduced earlier in this section. For this we will first give a formal proof of the consistency of the MLE and then present a rather general framework that will allow us to turn the formal proof into a rigorous one.

2.32 Formal proof of consistency. We will consider a general MLE like in (2.11) and we will assume that the observations (X_n) are independent and have density $f(x|\theta_0)$ with respect to the reference measure μ. By the law of large number we have

$$\frac{1}{n}\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log(f(X_i|\theta)) \xrightarrow{n\to\infty} \mathbb{E}\big[\log(f(X|\theta))\big]. \tag{2.15}$$

Hence the maximiser of the left hand side should be close to the maximiser of the right hand. Differentiating the right hand side yields

$$\begin{aligned} \partial_\theta \mathbb{E}\big[\log(f(X|\theta))\big] &= \mathbb{E}\left[\partial_\theta \log(f(X|\theta))\right] = \mathbb{E}\left[\frac{\partial_\theta f(X|\theta)}{f(X|\theta)}\right] \\ &= \int \frac{\partial_\theta f(x|\theta)}{f(x|\theta)} f(x|\theta_0)\mu(\mathrm{d}x). \end{aligned}$$

Evaluating this at $\theta = \theta_0$ gives

$$\int \partial_\theta f(x|\theta)\mu(\mathrm{d}x) = \partial_\theta \int f(x|\theta)\mu(\mathrm{d}x) = \partial_\theta(1) = 0.$$

Hence θ_0 is a critical point and under mild conditions the right hand side of (2.15) is concave and thus θ_0 is the unique maximiser. In conclusion the estimator $\hat{\theta}$ should be close to θ_0.

Although the rough structure of the rigorous proof is present in the argument above it is highly formal. For example we argue that if a sequence $(f_n)_{n\in\mathbb{N}}$ of functions converges towards f pointwise, then the maximisers $(x_n)_{n\in\mathbb{N}}$ should converge to the maximiser x of f. The major tool to make this rigorous will be to use some kind of uniform convergence. Namely, we have the following result where we will omit the proof since it is very easy and we give a similar but stronger version of it later.

2.33 Lemma (Swapping limit and maximisation). *Let $(f_n)_{n\in\mathbb{N}}$ be a sequence of real functions on a compact space with maximisers $(x_n)_{n\in\mathbb{N}}$ that are bounded from above and converge uniformly towards f. Further, assume that f is continuous and has a unique maximum in x_0. Then we have $x_n \to x_0$ for $n \to \infty$.*

Unfortunately the convergence in (2.15) does only hold uniformly on a compact set $K \subseteq \Theta$. To deal with this, we will argue that the maximisers (x_n) lie in this compact set K for large n. We will do this in a general setup in the next paragraph.

A general consistency result for extremal estimators

We provide a general consistency result for a rather broad class of estimators which is taken from [45] and slightly adapted to our needs. Although it would be possible to prove the consistency of the MLEs directly we present this general procedure since it clearly highlights the theoretical arguments and can therefore easily be adjusted to other cases.

2.34 Setting. Let in the following Θ be a topological Hausdorff space and $F_n \colon \Theta \to [-\infty, \infty)$ be a sequence of random functions with maximisers

$$\hat{\theta}_n := \operatorname*{arg\,max}_{\theta\in\Theta} F_n(\theta).$$

If no maximiser exists, we choose $\hat{\theta}_n \in \Theta$ arbitrary. Further, let $F \colon \Theta \to [-\infty, \infty)$ be a deterministic function with maximiser θ_0. The maximisers $\hat{\theta}_n$ are called *extremal estimators* since they are extremal points of the functions F_n.

We now investigate whether the extremal estimators converge to the maximiser θ_0.

2.35 Theorem (Consistency of extremal estimators). *Let the setting be as above and assume that the following conditions hold.*

(*i*) *Assume that there is* $\varepsilon_0 > 0$ *and a compact set* K_0 *containing* θ_0*, such that with probability tending to one*

$$F_n(\theta) \leq F(\theta_0) - \varepsilon_0 \quad \textit{for all } \theta \notin K_0. \tag{2.16}$$

(*ii*) *Let* F_n *converge to* F *uniformly on* K_0 *in probability, i.e. for any* $\varepsilon > 0$ *we have with probability tending to one*

$$\left|F_n(\theta) - F(\theta)\right| \leq \varepsilon \quad \textit{for all } \theta \in K_0. \tag{2.17}$$

(*iii*) *Let* F *have a unique maximum at* $\theta_0 \in \Theta$.

(*iv*) *Assume that* F *is upper semicontinuous in the sense that*

$$\left\{\theta \in \Theta \mid F(\theta) \geq \alpha\right\} \subseteq \Theta$$

is closed for all $\alpha \in \mathbb{R}$.

(*v*) *With probability tending to one* F_n *admits a maximiser.*

Then we have $\hat{\theta}_n \to \theta_0$ *in probability, i.e.*

$$\mathbb{P}\left(\hat{\theta}_n \in U\right) \xrightarrow{n\to\infty} 1$$

for any open subset $U \subseteq \Theta$ *containing* θ_0.

Proof. Note that it suffices to show $\hat{\theta}_n \in U$ whenever (2.16) and (2.17) hold and F_n admits a maximiser. From here on the proof is of purely analytic content.

Fix now an open set $U \subseteq \Theta$ that contains θ_0. Choosing $\varepsilon < \varepsilon_0$ in (*ii*) and using (*i*) yields

$$F_n(\theta_0) \geq F(\theta_0) - \varepsilon > F(\theta_0) - \varepsilon_0 \geq F_n(\theta) \quad \text{for all } \theta \notin K_0.$$

Hence the maximum of F_n is attained in K_0 and we have $\hat{\theta}_n \in K_0$. Thus if $K_0 \subseteq U$ we are done. If this is not the case F attains its maximum α on $K_0 \setminus U$ because F is upper semicontinuous and $K_0 \setminus U$ is compact. Further, (*iii*) implies $\alpha < F(\theta_0)$ and thus we have

$$K_0 \cap \left\{\theta \in \Theta \mid F(\theta) > \alpha\right\} \subseteq U.$$

So in order to show $\hat{\theta}_n \in U$, it suffices to show $F(\hat{\theta}_n) > \alpha$. However, (*ii*) implies

$$F(\hat{\theta}_n) \geq F_n(\hat{\theta}_n) - \varepsilon \geq F_n(\theta_0) - \varepsilon \geq F(\theta_0) - 2\varepsilon > \alpha$$

for ε small enough. □

2.36 Remark. The theorem above might seem artificially general at first, but one has a high interest in consistency results at least for extremal estimators in metric spaces. Those can be used to deduce the consistency for the estimation of a function in a function space rather than a finite dimensional parameter.

If we want to apply the previous result to the case of maximum likelihood estimation we need to set

$$F_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} \log(f(X_i|\theta)).$$

Note that the factor $\frac{1}{n}$ does not change the maximum. However, (2.15) already gives the almost surely pointwise limit of those functions and if condition (*ii*) of the previous statement should hold, we have to define

$$F(\theta) := \mathbb{E}\big[\log(f(X|\theta))\big].$$

The quantity F is – up to the sign – known as the *entropy* and plays an important role in many different fields, for example statistical mechanics, applied statistics and information theory. For further reading we refer to [39], [36], [65] and [18].

Information inequality and locally uniform convergence

The second and third requirement of the previous result can be proved in a general setting and without quantitative assumption and we adapt an argument from [45] to fit our needs. In order to do this we will work with the following assumptions.

2.37 Setting. Let in the following Θ be a set and let

$$\mathcal{F} = \Big\{ f(\cdot|\theta)\colon \mathcal{X} \to [0,\infty) \;\Big|\; \theta \in \Theta \Big\}$$

be a family of probability densities on some measurable space $\mathcal{X}$ with respect to some measure μ. Further, fix $\theta_0 \in \Theta$ and let X be distributed according to $f(\cdot|\theta_0)\mathrm{d}\mu$, hence we have

$$\mathbb{E}\big[h(X)\big] = \int h(x) f(x|\theta_0)\mu(\mathrm{d}x).$$

Let $(X_n)_{n\in\mathbb{N}}$ be a sequence of independent random variables distributed according to $f(\cdot|\theta_0)\mathrm{d}\mu$. Finally define

$$F_n(\theta) := \frac{1}{n}\sum_{i=1}^{n} \log(f(X_i|\theta)) \quad \text{and } F(\theta) := \mathbb{E}\big[\log(f(X|\theta))\big].$$

2.38 Proposition (Information inequality). *Let the setting be as above and assume that the parameter $\theta_0 \in \Theta$ is* identifiable*, i.e. we have $f(\cdot|\theta) \neq f(\cdot|\theta_0)$ whenever $\theta \neq \theta_0$. Let further*

$$\sup_{x\in\mathcal{X},\theta\in\Theta} f(x|\theta) < \infty \quad \textit{and } F(\theta_0) > -\infty.$$

Then the entropy

$$F(\theta) = \mathbb{E}\big[\log(f(X|\theta))\big]$$

has a unique maximum in θ_0.

Proof. Let $\theta \neq \theta_0$, then we either have $F(\theta) = -\infty < F(\theta_0)$ or

$$F(\theta) = \mathbb{E}\big[\log(f(X|\theta))\big] > -\infty. \tag{2.18}$$

In this case we want to exploit the strict Jensen inequality (cf. [34]) that yields for any positive random variable Y with finite expectation that is not constant

$$\mathbb{E}\big[\log(Y)\big] < \log(\mathbb{E}[Y]).$$

We set $Y := \frac{f(X|\theta)}{f(X|\theta_0)}$. This is positive $f(\cdot|\theta_0)\mathrm{d}\mu$ almost everywhere because otherwise (2.18) could not hold. Since θ_0 is identifiable, the random variable Y is not constant and we will see in the following computation that the expectation is finite. Now we obtain

$$\begin{aligned} F(\theta) - F(\theta_0) &= \mathbb{E}\big[\log(f(X|\theta))\big] - \mathbb{E}\big[\log(f(X|\theta_0))\big] = \mathbb{E}\left[\log\left(\frac{f(X|\theta)}{f(X|\theta_0)}\right)\right] \\ &< \log\left(\mathbb{E}\left[\frac{f(X|\theta)}{f(X|\theta_0)}\right]\right) = \log\left(\int f(x|\theta)\mu(\mathrm{d}x)\right) = 0. \end{aligned}$$

□

Next we take care of the second requirement of the consistency result. Namely we will show that the functions F_n associated with the MLE almost surely converge to F locally uniformly under fairly mild conditions. For this we modify the proof of a more general convergence result in [59].

2.39 Lemma (Locally uniform convergence). *Let the setting be as above, but let Θ be a metric space and let $K \subseteq \Theta$ be compact such that the following conditions hold.*

(*i*) *Let*

$$\mathbb{E}\left[\sup_{\theta \in K} \big|\log(f(X|\theta))\big|\right] < \infty.$$

(*ii*) *For every $\theta \in K$ we have $\log(f(\cdot, \gamma)) \to \log(f(\cdot|\theta))$ almost surely with respect to $f(\cdot|\theta_0)\mathrm{d}\mu$ for $\gamma \to \theta$.*

Then we almost surely have $F_n \to F$ uniformly on K, i.e. almost surely

$$\sup_{\theta \in K} \big|F_n(\theta) - F(\theta)\big| \xrightarrow{n\to\infty} 0.$$

Proof. Fix $\varepsilon > 0$ and define for $x \in \mathcal{X}$ and $\rho > 0$

$$u(x, \theta, \rho) := \sup_{d(\gamma,\theta)\leq\rho} \big| \log(f(x|\gamma)) - \log(f(x|\theta))\big| \xrightarrow{\rho\to 0} 0$$

almost surely for θ fixed where we used condition (ii).

This in combination with (i) and the dominated convergence theorem implies that the convergence also holds in expectation and therefore we have

$$\mathbb{E}\big[u(X, \theta, \rho)\big] \leq \varepsilon \quad \text{for } \rho \leq \delta(\theta).$$

The open balls $B_{\delta(\theta)}(\theta)$ with center θ and radius $\delta(\theta)$ cover the compact set K and hence we can select a finite subcover

$$K \subseteq \bigcup_{k=1}^{m} B_{\delta(\theta_k)}(\theta_k).$$

Further we set

$$\mu_k := \mathbb{E}\big[u(X, \theta_k, \delta(\theta_k))\big] \leq \varepsilon.$$

Let $\theta \in K$ and choose k such that $\theta \in B_{\delta(\theta_k)}(\theta_k)$, then we can conclude

$$\begin{aligned} |F_n(\theta) - F(\theta)| &\leq \frac{1}{n}\sum_{i=1}^{n} \big| \log(f(X_i|\theta)) - \log(f(X_i|\theta_k))\big| \\ &\quad + \left|\frac{1}{n}\sum_{i=1}^{n} \log(f(X_i|\theta_k)) - F(\theta_k)\right| + \big|F(\theta_k) - F(\theta)\big| \\ &\leq \left(\frac{1}{n}\sum_{i=1}^{n} u(X_i, \theta_k, \delta(\theta_k)) - \mu_k\right) + \mu_k + 2\varepsilon \\ &\leq 4\varepsilon \end{aligned}$$

almost surely for $n \geq N(\varepsilon)$ where we used the strong law of large numbers twice and

$$\big|F(\theta_k) - F(\theta)\big| \leq \mathbb{E}\left[u(X, \theta_k, \delta(\theta_k))\right] \leq \varepsilon.$$

□

Consistency of the MLEs for the quality and elementary kernel

In this part we will – for the first time – make use of the specific structure of the model. Since we have already taken care of the conditions (ii)-(v) of the general consistency result, we dedicate ourselves to the first requirement of Theorem 2.35. For this we keep the setting of the previous section although we now consider the case that

$$\mathcal{F} = \left\{ f(\cdot|\theta) \colon 2^{\mathcal{Y}} \to [0, \infty) \mid \theta \in \Theta \right\}$$

is one of the parametric families for the L-ensembles introduced in II.2.1. Further, we denote a realisation of a DPP by $\mathbf{Y}$ like earlier.

2.40 LEMMA (CONTROL OUTSIDE OF A COMPACT SET). *The requirement (i) from Theorem 2.35 is satisfied for the three kinds of parametric families for the kernel estimation. Further, the compact set K_0 can be chosen as follows. Let $\mathcal{A}$ be the family of subsets $A \subseteq \mathcal{Y}$ with positive probability $f(A|\theta_0) > 0$ and choose $c(A) > 0$ such that*

$$-c(A) < \frac{2 \cdot F(\theta_0)}{f(A|\theta_0)}.$$

Then we set

$$K_0 := \left\{\theta \in \Theta \;\middle|\; \log(f(A|\theta)) \geq -c(A) \textit{ for all } A \in \mathcal{A}\right\}.$$

Proof. At first we note that $F(\theta_0) > -\infty$. Let now

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

be the empirical measure. We have by the law of large numbers

$$\mathbb{P}\left(\hat{\mathbb{P}}_n(A) \geq \frac{f(A|\theta_0)}{2}\right) \xrightarrow{n\to\infty} 1$$

and so we can assume $\hat{\mathbb{P}}_n(A) \geq \frac{f(A|\theta_0)}{2}$, since we are only interested in proving a statement with probability tending to one. For $A \in \mathcal{A}$ we note that

$$K_A := \left\{\theta \in \Theta \;\middle|\; \log(f(A|\theta)) \geq -c(A)\right\}$$

is closed since $f(A|\theta)$ is upper semicontinuous. Further, K_A is compact for $A = \varnothing \in \mathcal{A}$ since $\log(f(\varnothing|\theta))$ is coercive in θ which has been shown in the coercivity proofs earlier in this chapter. Further, it contains θ_0 as

$$\log(f(A|\theta_0)) \geq 2 \cdot \log(f(A|\theta_0)) > -c(A)$$

because $f(A|\theta) \leq 1$ and

$$\frac{F(\theta_0)}{f(A|\theta_0)} = \frac{\sum_{B\subseteq\mathcal{Y}} f(B|\theta_0)\log(f(B|\theta_0))}{f(A|\theta_0)} \leq \log(f(A|\theta_0)).$$

Now

$$K_0 = \bigcap_{A\in\mathcal{A}} K_A$$

is compact because $K_\varnothing$ is compact. Fix $\theta \notin K_0$, lets say $\theta \notin K_A$, then we get

$$\begin{aligned} F_n(\theta) &= \int \log(f(x|\theta))\hat{\mathbb{P}}_n(\mathrm{d}x) = \sum_{B\in\mathcal{A}} \hat{\mathbb{P}}_n(B) \cdot \log(f(B|\theta)) \leq \hat{\mathbb{P}}_n(A) \cdot \log(f(A|\theta)) \\ &< -\frac{f(A|\theta_0)}{2} \cdot c(A) < F(\theta_0). \end{aligned}$$

□

Now we have all the auxiliary results to prove the desired consistency result.

2.41 Theorem (Consistency). (*i*) *The maximum likelihood estimator $\hat{L}_n$ for the elementary kernel is consistent. Namely if the observations $(\mathbf{Y}_n)$ follow the law of a L-ensemble with elementary kernel L_0, then we have*

$$\mathbb{P}\left(d(\hat{L}_n, L_0) \leq \varepsilon\right) \xrightarrow{n\to\infty} 1 \quad \textit{for all } \varepsilon > 0.$$

(*ii*) *The maximum likelihood estimator $\hat{q}_n$ for the quality vector is consistent. Namely if the observations $(\mathbf{Y}_n)$ follow the law of a L-ensemble with kernel $\Psi(q_0, \hat{\phi})$, then we have*

$$\mathbb{P}\Big(\big\|\hat{q}_n - q_0\big\| \leq \varepsilon\Big) \xrightarrow{n\to\infty} 1 \quad \textit{for all } \varepsilon > 0.$$

(*iii*) *Suppose that the observations $(\mathbf{Y}_n)$ follow the law of a L-ensemble with kernel $\Psi(p_0, \hat{S})$ where $(p_0)_i = \exp(\theta_0^T f_i)$. Then we have*

$$\mathbb{P}\left(\big\|\hat{\theta}_n - \theta_0\big\| \leq \varepsilon\right) \xrightarrow{n\to\infty} 1 \quad \textit{for all } \varepsilon > 0.$$

Proof. We will only sketch the main parts of the proof of the second statement, since all other arguments will be analogue and therefore redundant.

Obviously we want to exploit the machinery we have introduced and thus we will check the requirements of Theorem 2.35. First we note that (*v*) holds because of the section of the existence of the maximum likelihood estimators.

We can express the entropy function

$$F(q) = \mathbb{E}\big[\log(f(\mathbf{Y}|q))\big] = \sum_{A\subseteq\mathcal{Y}} \log(f(A|q))f(A|q_0) \tag{2.19}$$

where the elementary probabilities are given by

$$f(A|q) = \frac{\prod_{i\in A} q_i^2 \det(\hat{S}_A)}{\sum_{B\subseteq\mathcal{Y}} \prod_{i\in B} q_i^2 \det(\hat{S}_B)} \tag{2.20}$$

which is continuous in q. Hence, the entropy function F is upper semicontinuous and thus condition (*iv*) holds.

To check that (*iii*) holds, we will use the information inequality 2.38. First we note that because of

$$f(\{i\}\,|q) \propto q_i^2$$

the parameter q_0 is identifiable and further we have

$$\sup_{A\subseteq\mathcal{Y}, q\in\mathbb{R}_+^N} f(A|q) \leq 1$$

since the densities are elementary probabilities. Finally $F(q_0) > -\infty$ is clear from (2.19) and hence the third requirement is satisfied.

Since the previous lemma already takes care of condition (*i*) it suffices to show the second condition for which we will use 2.39. Hence, it remains to check the two conditions of this lemma, but the second one – the continuity condition – obviously holds as can be seen from (2.20). To see that the first one also holds, we note that for $A \subseteq \mathcal{Y}$ with $f(A|q_0) > 0$ and $q \in K_0$ we have

$$0 \geq \log(f(A|q)) \geq -c(A) > -\infty.$$

Hence the random variable

$$\sup_{q \in K_0} \big|\log(f(\mathbf{Y}|q))\big|$$

is almost surely finite with respect to $f(\cdot|q_0)\mathrm{d}\mu$ and since the probability space $2^{\mathcal{Y}}$ is finite, the second condition holds. □

2.42 Remark. Obviously in the proof of the consistency of the whole elementary kernel L one runs into the problem of unidentifiability. This is why one has to identify the parameters with each other that give rise to the same elementary probabilities which is just the determinantal equivalence. Once this is done, the proof follows analogously.

Consistency of regularised MLE

One can not make a general statement about the consistency of regularised MLE. Since it is straight forward to construct examples where the regularised MLE is not consistent. For example if the regulariser R is equal to $-\infty$ on a neighborhood of θ_0, then none of the regularised estimators will lie in this neighborhood and hence the estimations will not converge towards θ_0.

Nevertheless, this scenario can be avoided with one property of the regulariser. If we recall that the proof of the consistency of the MLE relied on the observation

$$\frac{1}{n} \cdot \mathcal{L}_n(\theta) \xrightarrow{n\to\infty} \mathbb{E}\big[\log(f(X|\theta))\big].$$

and an application of the consistency result for extremal estimators. In the case of the regularised MLE, we no longer maximise the functions $\mathcal{L}_n$ but the sum $\mathcal{L}_n + R$ and in order to obtain a consistent estimator we need to ensure that the convergence

$$\frac{1}{n} \cdot \mathcal{L}_n(\theta) + \frac{1}{n} \cdot R(\theta) \xrightarrow{n\to\infty} \mathbb{E}\big[\log(f(X|\theta))\big]$$

holds with the same uniformity which we will do now.

2.43 Setting. We work with one of the three parametric models for DPPs presented above and denote the parameter space by Θ and the elementary kernel arising from $\theta \in \Theta$ by $L(\theta)$. Further we will assume that we have a regulariser $R\colon \Theta \to [-\infty, 0]$ that is upper semicontinuous.

2.44 Theorem (Consistency of MAP estimation). *Assume that the sequence of observations $(\mathbf{Y}_n)_{n\in\mathbb{N}}$ are independent and distributed according to a L-ensemble with elementary kernel $L(\theta_0)$. Let further $K \subseteq \Theta$ be a compact set containing θ_0 such that R is bounded on K. Then the maximum a posteriori estimator arising from the regulariser R is consistent.*

Proof. We only have to check how the regulariser R influences the requirements (i)-(v) of Theorem 2.35. In fact it doesn't have any influence on the assumptions (iii) and (iv). Further (i) stays valid since the regulariser is non positive and (v) since it is non positive and upper semicontinuous. Hence it remains to validate (ii). For this we reduce the compact set K_0 to $K \cap K_0$ which is compact again. On this set the contribution $\frac{1}{n} \cdot R$ goes uniformly to zero which yields the assertion. □

2.45 Remark. The consistency result above is stated for a rather broad class of regularisers. For instance, all quadratic regulariser – which correspond exactly to Gaussian priors as we will see later – fall into this class. Further it covers every regulariser that is equal to $-\infty$ on a set of impossible parameters. This can be used to exclude non symmetric or non positive semidefinite matrices from the estimation process if one wants to work with the easier parameter set $\mathbb{R}^{N\times N}$ instead of $\mathbb{R}^{N\times N}_{\text{sym},+}$.

II.2.4 Approximation of the MLE

Having discussed the theoretical properties and guarantees on convergence of the MLE we will now turn towards the question of computability. In particular we will see that the MLE for the whole kernel can not be computed in an efficient way which justifies the use of smaller parametric models like the log linear model.

Likelihood maximisation for the elementary kernel L

We recall that the log likelihood function for the elementary kernel is given by

$$\mathcal{L}(L) = \sum_{i=1}^{n} \log\left(\det(L_{\mathbf{Y}_i})\right) - n\log\left(\det(L+I)\right). \tag{2.21}$$

This is a smooth function since the determinants of the submatrices are polynomials in the entries of L and the composition of those with the smooth function $\log\colon (0,\infty) \to \mathbb{R}$ is smooth. This property makes it possible to use gradient methods for the maximisation of $\mathcal{L}$, but they face the problem that $\mathcal{L}$ 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 $\left\{qI \mid q \in \mathbb{R}\right\}$ 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. Unfortunately there are no algorithms that can approximate the global maximum of a non concave function [63] and it also has been conjectured in [26] that no such algorithm exists for the log likelihood of the elementary kernel. Nevertheless one can still use optimisation techniques to obtain local maximisers of the log likelihood and indeed [38] proposes a fixed point iteration to do this.

In fact, the non concavity can be seen without any computations at all. If we have a maximiser L of the log likelihood function and take another matrix $\tilde{L}$ that is determinantally equivalent, then obviously the log likelihood of the two kernels agree. Hence, we expect a discrete set of maximisers which is only possible if $\mathcal{L}$ is not concave. However it is not straight forward whether there are some critical points that are different to the global maximisers. If this is not the case, common optimisation techniques can be exploit since they will converge to a critical point.

Similar arguments show that the log likelihood function for the qualities is non concave and therefore hard to maximise.

Computation for the log linear model

The motivation to introduce the log linear model was to obtain a log likelihood function that is easier to maximise in practice. We will now see that this is indeed the case and recall that the individual terms of the log likelihood are given by

$$2 \cdot \theta^T \sum_{i \in Y} f_i + \det(\hat{S}_Y) - \log\left(\sum_{A \subseteq \mathcal{Y}} \exp\left(2 \cdot \theta^T \sum_{i \in A} f_i\right) \det(\hat{S}_A)\right). \tag{2.22}$$

The first two terms are linear in θ and constant and thus concave. To see that the last expression is also concave we introduce the notion of log concavity and give a general result.

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

2.47 Proposition (Additivity of log concavity). *The sum of log convex functions is again log convex.*

Proof. Let f and g be log convex and thus $F := \log(f)$ and $G := \log(g)$ are convex. We will consider the function

$$H : \mathbb{R}^2 \to \mathbb{R}, \quad (x, y) \mapsto \log(e^x + e^y)$$

which is increasing both coordinates. Further, we note that $\log(f + g) = H(F, G)$ and hence it suffices to show that H is convex which we do by noting that the Hessian matrix

$$D^2 H(x, y) = (e^x + e^y)^2 \cdot \begin{pmatrix} e^x e^y & -e^x e^y \\ -e^x e^y & e^x e^y \end{pmatrix}$$

is positive semi-definite by Sylvester's criterion. □

The summands inside the logarithm in (2.22) are log convex, in fact even log affine linear since their logarithm is equal to

$$2 \cdot \theta^T \sum_{i \in A} f_i + \log(\det(\hat{S}_A)).$$

Hence we obtain as an immediate consequence that the whole expression (2.22) and therefore the log likelihood function is concave which we will fix in a separate statement.

2.48 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 $\theta \in \mathbb{R}^M$.*

This result together with the coercivity – which holds with probability tending to one – ensures that the MLE for the log linearity constant of the quality can be computed efficiently. Further, such optimisation algorithms are pre-implemented in most major programming languages and software environments like Mathematica, MATLAB and R. For theoretical guarantees and algorithmic details of those methods we refer once again to [10].

Although being easily available, those methods have the drawback that they need to approximate the gradient of the log likelihood function. However, the gradient can be expressed analytically by differentiating (2.22) and we obtain

$$\begin{aligned}
&2 \cdot \sum_{i \in Y} f_i - 2 \cdot \frac{\sum_{A \subseteq \mathcal{Y}} \exp\left(2 \cdot \theta^T \sum_{i \in A} f_i\right) \sum_{i \in A} f_i \det(\hat{S}_A)}{\sum_{A \subseteq \mathcal{Y}} \exp\left(2 \cdot \theta^T \sum_{i \in A} f_i\right) \det(\hat{S}_A)} \\
&= 2 \cdot \sum_{i \in Y} f_i - 2 \cdot \sum_{A \subseteq \mathcal{Y}} f(A|\theta) \sum_{i \in A} f_i \\
&= 2 \cdot \sum_{i \in Y} f_i - 2 \cdot \sum_{i \in \mathcal{Y}} f_i \sum_{A \supseteq \{i\}} f(A|\theta) \\
&= 2 \cdot \sum_{i \in Y} f_i - 2 \cdot \sum_{i \in \mathcal{Y}} f_i K(\theta)_{ii}.
\end{aligned}$$

where $K(\theta)$ is the marginal kernel arising from the parameter θ. This expression can be used in a direct implementation of the gradient method for the approximation of the MLE for the log linearity constant.

II.2.5 Further learning approaches

We will quickly touch on two approaches of parameter estimation that have proven to work very well for certain real world examples. Further, we propose a different parametric model that would allow the simultaneous estimation of the qualities and the similarity kernel, but it remains to be seen whether this circumvents the problems associated with the maximum likelihood estimation of the whole kernel L.

Learning for conditional DPPs

The estimation of the log linearity constant of conditional DPPs has been used in [29] to obtain extractive summaries of news articles. In fact the procedure is analogue to the one presented for the estimation of the log linearity constant of normal DPPs, apart from the fact that one has to model a family of feature vectors

$$f_i(X) \in \mathbb{R}^M \quad \text{for } i \in \mathcal{Y}(X), X \in \mathcal{X}.$$

We will quickly discuss how this could be done in the case of the DPP on a two dimensional grid. Maybe we do not want to restrict ourselves to one grid size and hence consider the conditional DPP

$$\mathcal{X} = \mathbb{N}, \quad \mathcal{Y}(n) = n^{-1}\{0, \dots, n\}^2 \quad \text{for } n \in \mathbb{N}.$$

Now we can model the similarity feature vectors $\phi_i(n)$ analogously to the case where we only considered one grid and impose the following log linear model for the qualities

$$q_i(n) = \exp\left(\theta^T f_i(n)\right) \quad \text{for } i \in \mathcal{Y}(n).$$

The diversity feaure vectors are given just like earlier by

$$f_i(n) = \begin{pmatrix} \|i - m\| \\ 1 \end{pmatrix} \quad \text{for } i \in \mathcal{Y}(n)$$

where m is again the centre of the unit square. Now the estimation can be carried out just like in the case of an ordinary DPP and for the same reason this will be consistent.

However, it shall be noted that the modelling of the diversity feature vectors can be far more complicated in more complex real world applications.

Estimating the mixture coefficients of k-DPPs

For this approach we first fix symmetric positive semi-definite matrices $L_1, \dots, L_M$. We assume now that the point process we are trying to describe is the mixture of the DPPs $\mathbb{P}_{L_m}$ with elementary kernel L_m and aim to estimate the mixing coefficients of

$$\mathbb{P}_\theta = \sum_{m=1}^{M} \theta_m \mathbb{P}_{L_m} \quad \text{where } \theta_m \in [0, 1] \text{ and } \sum_{m=1}^{M} \theta_m = 1.$$

This approach has been taken to create a diverse selection of pictures returned by an image search (cf. [28]). In this case one wants to fix the number of returned images up front, hence it is reasonable to work with k-DPPs instead of DPPs. The mixture coefficients were estimated based on a data set $\{(Y_t^+, Y_t^-)\}_{t=1,\dots,n}$ where Y_t^+ was chosen by a human to be more diverse than Y_t^-. Now θ was optimised such that

$$\mathbb{P}_\theta(Y_t^+) > \mathbb{P}_\theta(Y_t^-)$$

for as many $t \in \{1, \dots, n\}$ as possible.[6]

Learning the repulsiveness of a DPP

The estimation of the qualities, or the according log linearity constant have the major drawback that a significant part of the DPP – the repulsive structure – has to be modelled completely. We have seen so far that it is in practice not possible to estimate the whole repellent structure, namely the similarity kernel as this would lead to a hard optimisation problem. However, we will propose a parametrisation of the similarity kernel by only one parameter, that might have better computational properties.

For this we follow 1.7 to model the similarity over the distance to some reference points with respect to a Gaussian kernel just like in the toy example presented so far. This means we choose

$$(\phi_i)_r \propto \exp\left(-\frac{d(i,r)^2}{\sigma}\right) \quad \text{for } r \in \mathcal{R}, i \in \mathcal{Y}.$$

We have seen in the example of the DPP on a line that the parameter σ has a direct influence on the strength of the repulsions of the DPP. The estimation of not only the log linearity constant θ but also the *repulsiveness parameter* σ could now result in a significantly increased accuracy of the resulting model.

It is not immediately clear what properties the log likelihood has in this case, but it would be – a rather pleasant – surprise if it was coercive. Nevertheless, it might have nice properties, like a unique critical point that could allow the use of standard optimisation techniques. A different approach would be to optimise the two parameters θ and σ in an adaptive scheme, i.e. one after another and repeat this iteratively. It remains to be seen whether such approaches work theoretically and whether they give any improved results in practice.

[6] For people familiar with binary decision problems it should be mentioned that this was done using the logistic loss.

Chapter III

Bayesian parameter estimation and Markov chain Monte Carlo methods

So far we have seen two different estimation techniques for the parameters of DPPs. Although we proved that they provide reasonable estimators in the sense that they are consistent, they have some drawbacks. For example the MLEs for the different parameters do not exist in general, let alone that they are impossible to compute in practice. Further all of the estimators presented so far are point estimators, i.e. they return a single value for the desired parameter. Obviously this does not allow to capture any uncertainties that the estimation of the parameter has. Those are some reasons to consider the Bayesian approach of parameter estimation where the goal is to give a distribution – called the posterior – of the parameter that should be estimated instead of a single value. This can also help to overcome some – maybe even all of the problems mentioned above.

At first we will present the general concept of Bayesian parameter estimation and will then turn towards the question of computability of the posterior distribution. For this we will follow the approach of [1] and turn towards the popular Markov chain Monte Carlo (MCMC) methods. We quickly explain their philosophy and how they can be used to approximate the posterior distribution of the parameter one wishes to estimate.

III.1 Bayesian approach to parameter estimation

For the introduction of the general Bayesian setup we pursue like in [48]. Just like in the case of maximum likelihood estimation we want to estimate a parameter $\theta \in \Theta$ based on some realisations $x = (x_1, \dots, x_n)$ of random variables $X = (X_1, \dots, X_n)$. This time, however, we are not interested in returning a single value θ because this would be a vast simplification of the stochastic nature of the estimator. We rather want to obtain a probability distribution over whole parameter space Θ that indicates how likely the parameters are to have caused the observed data. In order to present the procedure we will introduce the frame we will work in.

3.1 Setting. Let Θ be a measurable space and ν be a measure on Θ. Let $f_\Theta \colon \Theta \to [0,\infty]$ be a probability density with respect to ν, i.e.

$$\int\limits_\Theta f_\Theta(\theta)\nu(\mathrm{d}\theta) = 1$$

which we will call the *prior* distribution of the parameter θ.[1] Further let

$$\mathcal{F} = \left\{ f_{X|\Theta}(\cdot|\theta) \mid \theta \in \Theta \right\}$$

by a family of probability densities with respect to $\mu^n := \prod_{i=1}^n \mu(\mathrm{d}x_i)$.

Usually the prior distribution will encode some perceptions or prior knowledge we might have of the parameter. For example if we are trying to estimate a physical constant that we know has to be positive, then it is reasonable to select a prior that has its whole mass on the positive real line. However, there is no clear set of rules how one can select a suitable prior to a given problem.

The density $f_{X|\Theta}(x|\theta)$ describes how likely the observations are under the parameter θ and we want to find an expression of how likely the parameter θ is under the observations x. In order to obtain this, we will work with the joint density

$$f_{X,\Theta}(x,\theta) = f_{X|\Theta}(x|\theta) f_\Theta(\theta) \quad \text{with respect to } \mu^n \times \nu$$

and condition this onto x. This yields

$$f_{\Theta|X}(\theta|x) = \frac{f_{X,\Theta}(x,\theta)}{\int_\Theta f_{X,\Theta}(x,\theta)\nu(\mathrm{d}\theta)} = \frac{f_{X|\Theta}(x|\theta) f_\Theta(\theta)}{\int_\Theta f_{X,\Theta}(x,\theta)\nu(\mathrm{d}\theta)}. \tag{3.1}$$

3.2 Definition (Posterior distribution). The density $f_{\Theta|X}$ is called the *posterior distribution* of the parameter θ given the data x. Further we call the normalisation constant

$$f(x|\mathcal{F}) := \int_\Theta f_{X,\Theta}(x,\theta)\nu(\mathrm{d}\theta)$$

the total probability of the data x under the model $\mathcal{F}$.

First we will convince ourselves that the approach of calculating a posterior distribution is a generalisation of the MLE in a lot of cases.

[1] The requirement of f being a probability density can easily be loosened. In fact if it has finite integral it is obvious that the normalisation cancels in the definition (3.1) of the posterior and even if it has infinite integral, (3.1) might still give a probability density.

3.3 Comparison to MLE. Maybe one feels slightly uncomfortable with the need to choose a prior distribution and it turns out that this is in fact a difficult step that has to be taken with a certain amount of care. However, we could pretend for one moment to be completely ignorant in the sense that we do not know anything about the parameter and hence we don't feel in the position to propose a reasonable prior. Then we could simply choose the uniform distribution as a prior – given it exists[2] – and would obtain

$$f_{\Theta|X}(\theta|x) \propto f_{X|\Theta}(x|\theta).$$

Hence we can regain the MLE from our posterior distribution since it is just the mode, i.e. the maximiser of the posterior density. This relation to the MLE can be seen in Figure 3.1. Hence, the Bayesian approach is a more general tool than MLE and allows to capture the randomness of the parameter θ. This is desirable since we have seen that the mode is not always a very typical outcome of a random variable.

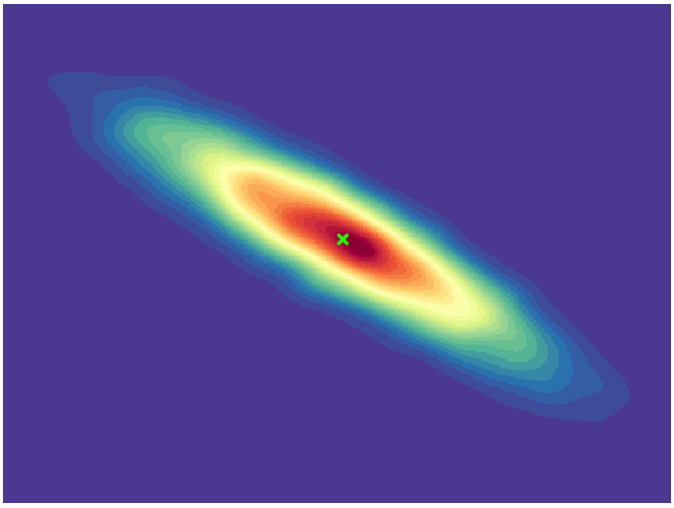


Figure 3.1: Approximated posterior density of the two dimensional log linearity constant of a two dimensional DPP with a uniform distribution as a prior. The MLE estimator is marked green and is at the mode of the distribution.

A further advantage over the MLE is that it might be possible to computationally approximate the posterior density, but not the MLE. This is typically the case if the log likelihood function is not concave, like in the setting of the MLE of the whole elementary kernel L. In fact the only hard step in the calculation of the posterior (3.1) is the computation of the normalisation constant

$$\int_{\Theta} f_{X,\Theta}(x,\theta)\nu(\mathrm{d}\theta).$$

[2] Even it doesn't one can still define the prior density to be constant and hope that the posterior is a probability density.

This can often not be performed efficiently but the Markov chain Monte Carlo methods introduced later will yield an approximation of the posterior without the need to compute the normalisation constant.

3.4 Regularisation through the prior. The prior density is very closely related to the regulariser introduced in the section about maximum likelihood estimation. In fact the mode of the posterior $f_{\Theta|X}$ is nothing else but the maximiser of

$$\log(f_{\Theta|X}) = \mathcal{L} + R$$

where $R = \log(f_\Theta)$ and hence nothing else but a regularised MLE. In fact if R is a regularisation and $\exp(R)$ is integrable with respect to μ, then one can choose $f_\Theta \propto \exp(R)$ as a prior density. Hence the proposition of a prior and a regulariser are equivalent in a wide variety of cases, but again, the posterior density encodes much more information than just the location of its mode which is the regularised MLE.

3.5 Bayesian approach without prior. We have seen that the prior is nothing else but a regularisation of the likelihood and since MLE can be carried out without regularisation it is natural to ask whether the Bayesian approach works without a prior. We have seen that if there is a uniform distribution on the parameter space, the unregularised MLE corresponds to the uniform distribution as a prior. So the question is what changes if we propose $f_\Theta = 1$ as a prior if there is no uniform distribution, or more generally what happens if the prior f_Θ has infinite integral.

In this case, the unnormalised posterior distribution

$$\mathfrak{f}(\theta) = f_{X|\Theta}(x|\theta) f_\Theta(\theta) \tag{3.2}$$

might not have finite integral and can therefore not be normalised. Hence the posterior can not be seen as a probability density, but the generalised form (3.2) still exists. If we use the constant prior $f_\Theta = 1$, the generalised posterior $\mathfrak{f}(\theta)$ is just the observation probability (or density if $\mathcal{X}$ is continuous) of the data x under the parameter θ.

Expression of the posterior for DPPs

Now we will express the posterior in the case of DPPs under the following conditions.

3.6 Setting. Let (Θ, ν) be a measure space and $L(\theta) \in \mathbb{R}^{N\times N}_{\text{sym},+}$ be an elementary kernel for every $\theta \in \Theta$. Further we assume that we have independent realisations $Y_1, \dots, Y_n$ of a L-ensemble.

Typically the parametrisations $\theta \mapsto L(\theta)$ will be one of the three parametric models in III.2.1, i.e. θ will either be the whole kernel itself, the quality vector or the log linearity constant of the qualities and $L(\theta)$ the associated elementary kernel.

The independence relation leads to a factorisation of the density and we obtain the following expression for the posterior density

$$f_{\Theta|\mathbf{Y}^n}(\theta|Y_1,\dots,Y_n) \propto f_\Theta(\theta)\prod_{i=1}^n f_{\mathbf{Y}|\Theta}(Y_i|\theta) = f_\Theta(\theta)\prod_{i=1}^n \frac{\det(L(\theta)_{Y_i})}{\det(L(\theta)+I)}. \tag{3.3}$$

Unfortunately the normalisation constant

$$\int_\Theta f_{\Theta|\mathbf{Y}^n}(\theta|Y_1,\dots,Y_n)\nu(\mathrm{d}\theta) = \int_\Theta f_\Theta(\theta)\prod_{i=1}^n \frac{\det(L(\theta)_{Y_i})}{\det(L(\theta)+I)}\nu(\mathrm{d}\theta) \tag{3.4}$$

can neither be computed analytically nor numerically in an efficient way since the evaluation of this density involves the computation of the determinant of a $N \times N$ matrix. This problem can be solved through the powerful methods of Markov chain Monte Carlo simulation that allow to approximate a distribution with only the knowledge of its unnormalised density.

Model selection using the Bayes factor

In this paragraph we will quickly touch on how the Bayesian approach can be used to compare two different models, i.e. two different parametric families $\mathcal{F}_1$ and $\mathcal{F}_2$ including two different priors f_{Θ_1} and f_{Θ_2}. For this we will work in the following setup.

3.7 Setting. Let Θ_1, Θ_2 be measurable spaces and ν_i measures on Θ_i for $i = 1, 2$. Let $f_{\Theta_i}: \Theta_i \to [0,\infty]$ be probability densities with respect to ν_i, i.e.

$$\int_{\Theta_i} f_{\Theta_i}(\theta)\nu_i(\mathrm{d}\theta) = 1 \quad \text{for } i = 1, 2.$$

Further let

$$\mathcal{F}_i = \left\{ f_{X|\Theta_i}(\cdot|\theta) \mid \theta \in \Theta_i \right\}$$

by a family of probability densities with respect to $\mu^n := \prod_{i=1}^n \mu(\mathrm{d}x_i)$.

The goal is now two compare which model $\mathcal{F}_i$ in combination with the corresponding prior describes the phenomenon better given some data x. For this we follow [25] and introduce the *Bayes factor* of the two models given the data x through

$$K := K(\mathcal{F}_1, \mathcal{F}_2|x) := \frac{f(x|\mathcal{F}_1)}{f(x|\mathcal{F}_2)} = \frac{\int_{\Theta_1} f_{X|\Theta_1}(x|\theta)f_{\Theta_1}(\theta)\nu_1(\mathrm{d}\theta)}{\int_{\Theta_2} f_{X|\Theta_2}(x|\theta)f_{\Theta_2}(\theta)\nu_2(\mathrm{d}\theta)}.$$

This is nothing but the ratio of the total probabilities of the data x under the respective models. If this ratio is big, the model $\mathcal{F}_1$ including its prior can be seen as a better description of the data compared to the second model. There is no clear definition on when the ratio can be seen as big enough to say this, but the following guidelines in Table III.1 were proposed in [25].

Value for K	Interpretation
$1 - 3.2$	Only worth a bare mention
$3.2 - 10$	Substantial
$10 - 100$	Strong
> 100	Decisive

Table III.1: Interpretation of how strongly different Bayes factors imply that one model a better description of the data than the other one.

III.2 Markov chain Monte Carlo methods

The method of Markov chain Monte Carlo (MCMC) simulation arose almost as early as the Monte Carlo[3] simulations itself and since then a rich theory has been established and a broad range of applications have been found. However, we can only give a short overview over the basic principles and refer to [42] for an introduction of Markov chain theory and to [50] for a survey on (Markov chain) Monte Carlo methods.

Our motivation for the study of MCMC methods was to obtain an approximation of a distribution π under the knowledge of its unnormalised density. In the nutshell the idea is to construct an ergodic Markov chain $(X_n)_{n\in\mathbb{N}}$ with stationary distribution π, i.e. such that one has

$$\hat{\mathbb{P}}_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{X_n} \xrightarrow{n\to\infty} \pi$$

almost surely in the weak sense. This Markov chain can then be simulated using Monte Carlo methods and the associated empirical measures $\hat{\mathbb{P}}_n$ will be approximations of π. However, to explain this in more detail we need to recapture some notions of Markov chains.

III.2.1 Reminder on Markov chains

We will provide an extremely short presentation of only those results that we will use to explain the core of MCMC methods. However, this will not contain any proofs and hence it can not replace the study of the already mentioned text books.

Let in the following $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ be a measurable space. This will typically be a topological space with its Borel algebra later.

3.8 Definition (Markov chain). (*i*) A *transition kernel* is a function

$$K\colon \mathcal{X} \times \mathcal{B}(\mathcal{X}) \to [0, 1]$$

[3] A legend has it that the name Monte Carlo was given to the work of von Neumann and Ulam by a colleague referring to Ulam's uncle who lost a significant amount of money gambling in the Monte Carlo casino in Monaco.

such that

a) $K(x,\cdot)$ is a probability measure for every $x \in \mathcal{X}$ and

b) $K(\cdot, A)$ is measurable for every $A \in \mathcal{B}(\mathcal{X})$.

(*ii*) A *Markov chain* with values in $\mathcal{X}$ and transition kernel K is a collection $(X_n)_{n\in\mathbb{N}}$ of $\mathcal{X}$ valued random variables such that

$$\mathbb{P}\big(X_0 \in A_0, \dots, X_n \in A_n\big) = \int_{A_0} \gamma(\mathrm{d}x_0) \int_{A_1} K(x_0, \mathrm{d}x_1) \cdots \int_{A_n} K(x_{n-1}, \mathrm{d}x_n) \tag{3.5}$$

for all $A_1, \dots, A_n \in \mathcal{B}(\mathcal{X})$ where γ denotes the distribution of X_0.

We will call γ the *initial* or *starting distribution* of the Markov chain and will denote the distribution of this Markov chain by $\mathbb{P}_\gamma$ and the expectation with respect to it by $\mathbb{E}_\gamma[\cdot]$. Further an easy application of Kolmogorov's consistency theorem implies that there is a measure $\mathbb{P}_\gamma$ on the *path space* $\mathcal{X}^\mathbb{N}$ that satisfies (3.5) which shows the existence of a Markov chain given a transition kernel K and initial distribution γ (cf. [32]). If the initial distribution is deterministic, i.e. $\gamma = \delta_x$ for one $x \in \mathcal{X}$, then we also write $\mathbb{P}_x$ for the distribution of the Markov chain. We close this paragraph by introducing the notation

$$K^n(x, A) := \mathbb{P}_x(X_n \in A)$$

which is consistent with (3.5) for $n = 1$.

Irreducibility, recurrence and existence of stationary distributions

From now on we will fix a reference measure μ on $\mathcal{X}$.

3.9 Definition (Irreducibility and recurrence). (*i*) We say a Markov chain is μ *irreducible* if for every $A \in \mathcal{B}(\mathcal{X})$ with $\mu(A) > 0$ there is an index $n \in \mathbb{N}$ such that

$$\mathbb{P}_x(X_n \in A) = K^n(x, A) > 0 \quad \text{for all } x \in \mathcal{X}.$$

(*ii*) A Markov chain $(X_n)_{n\in\mathbb{N}}$ is called *recurrent* if

a) there is a measure μ on $\mathcal{B}(\mathcal{X})$ such that (X_n) is μ-irreducible and

b) for every $A \in \mathcal{B}(\mathcal{X})$ with $\mu(A) > 0$ the expected number of visits of A is infinite, i.e.

$$\mathbb{E}_x\Big[\Big|\big\{n \in \mathbb{N} \mid X_n \in A\big\}\Big|\Big] = \infty \quad \text{for every } x \in A.$$

(*iii*) A Markov chain is called *Harris recurrent* if it is recurrent and the number of visits is almost surely infinite, i.e. for any $A \in \mathcal{B}(\mathcal{X})$ with $\mu(A) > 0$ we have

$$\mathbb{P}_x\Big(\Big|\big\{n \in \mathbb{N} \mid X_n \in A\big\}\Big| = \infty\Big) = 1 \quad \text{for every } x \in A.$$

3.10 Definition (Stationary distribution). Let π be a measure on $\mathcal{B}(\mathcal{X})$. We call π an *invariant* or *stationary distribution* of a Markov chain with kernel K, if X_{n+1} is distributed according to π whenever X_n is distributed according to π. This is equivalent to

$$\pi(A) = \int_{\mathcal{X}} K(x, A)\pi(\mathrm{d}x) \quad \text{for all } A \in \mathcal{B}(\mathcal{X}).$$

3.11 Theorem (Existence of stationary distributions). *If $(X_n)_{n\in\mathbb{N}}$ is a recurrent Markov chain, there exists an invariant σ-finite measure which is unique up to a multiplicative factor.*

Convergence to the stationary distribution and ergodicity

We will not introduce the notion of periodic and aperiodic Markov chains here, because it would distract us from our actual goal. However, we still present the following result that only holds for aperiodic Markov chains and refer to [42] for further information. The reason why we present the theorem is that it explains how one can approximately sample from the stationary distribution of a Markov chain, namely it says that the distribution of X_n converges to the invariant distribution.

3.12 Theorem (Convergence to stationary distribution). *Let $(X_n)_{n\in\mathbb{N}}$ be a Harris recurrent and aperiodic Markov chain with stationary distribution π. Let further γ_n be the distribution of X_n, then we have*

$$\|\gamma_n - \pi\|_{TV} \xrightarrow{n\to\infty} 0$$

non increasing. Here $\|\cdot\|_{TV}$ denotes the total variation of a measure

$$\|\mu\|_{TV} := \sup_{\mathcal{E}} \sum_{E\in\mathcal{E}} |\mu(E)|$$

where the supremum is taken over all finite families of disjoint measurable sets.

3.13 Theorem (Ergodic theorem). *Let $(X_n)_{n\in\mathbb{N}}$ be a Harris recurrent Markov chain with stationary probability distribution π, then $(X_n)_{n\in\mathbb{N}}$ is* ergodic*. This means that if*

$$\hat{\mathbb{P}}_n := \frac{1}{n}\sum_{i=1}^{n} \delta_{X_i}$$

is the empirical measure, we almost surely have

$$\int_{\mathcal{X}} f(x)\hat{\mathbb{P}}_n(\mathrm{d}x) \xrightarrow{n\to\infty} \int_{\mathcal{X}} f(x)\pi(\mathrm{d}x) \tag{3.6}$$

for every π integrable function f.

In the particular case that $\mathcal{X}$ is a topological space and $\mathcal{B}(\mathcal{X})$ is the Borel algebra and if π is a probability measure, we obtain the almost sure weak convergence of $\hat{\mathbb{P}}_n$ towards π. This means that the convergence in (3.6) almost surely holds for all continuous and bounded functions f. Hence, $\hat{\mathbb{P}}_n$ are approximations of the invariant distribution in the sense of weak convergence, which is metrisable for example by the Lévy-Prokhorov or the bounded dual Lipschitz metric (cf. [14]).

Idea of Markov chain Monte Carlo methods

The motivation of the study of Markov chain Monte Carlo methods was to approximate the posterior distribution (3.3). The idea is now to construct and then simulate a Markov chain $(X_n)_{n\in\mathbb{N}}$ such that the empirical measures $\hat{\mathbb{P}}_n$ converge to the posterior.

3.14 Definition (MCMC methods). A *Markov chain Monte Carlo* (*MCMC*) method for the simulation of a distribution π is any method that produces an ergodic Markov chain $(X_n)_{n\in\mathbb{N}}$ with stationary distribution π.

In order to achieve this we only have to construct a suitable Markov chain and check the requirements of the ergodic theorem. This means we want to construct a Harris recurrent Markov chain with invariant distribution π and we want to do this without having to compute the normalisation constant (3.4). We will now present two of the most common methods to do this which are the Metropolis-Hastings random walk and the method of slice sampling.

III.2.2 Metropolis-Hastings random walk

The Metropolis-Hastings random walk is arguably the most commonly used MCMC method and certainly one of the oldest. It was actually proposed in the early 1950s from researchers of the American nuclear programme in Los Alamos (cf. [41]). First we will touch on the theoretical aspects of this method and follow the presentation in [50].

3.15 Setting. Let Θ be a measurable space, μ a measure on that space and $f : \mathcal{X} \to [0,\infty]$ a function with finite positive integral

$$Z := \int\limits_{\mathcal{X}} f(x)\mu(\mathrm{d}x) \in (0,\infty).$$

Our goal is to find a Harris recurrent Markov chain with invariant distribution

$$\pi(A) := \frac{1}{Z}\int\limits_{A} f(x)\mu(\mathrm{d}x).$$

Let further

$$\Big\{ f(\cdot|x) \;\Big|\; x \in \mathcal{X} \Big\}$$

be a family of probability distributions, which we call the *proposal distributions*.

3.16 The MH random walk. Given the first states $X_0 = x_0, \dots, X_n = x_n$ of the Markov, we define X_{n+1} as follows. Let Y be distributed according to $f(\cdot|x_n)\mathrm{d}\mu$ and take one realisation y of Y. Then set

$$X_{n+1} := \begin{cases} y & \text{with probability } \rho(x_n, y) \\ x_n & \text{with probability } 1 - \rho(x_n, y) \end{cases}$$

where

$$\rho(x, y) := \min\left\{ \frac{f(y) f(x|y)}{f(x) f(y|x)}, 1 \right\}. \tag{3.7}$$

and $\frac{a}{0} := \infty$. The first step of the random walk, namely the sampling of y is called the *proposal step* and the second one the *accept-reject step*. In conclusion a single step of the MH random walk can be expressed in the following way.

Algorithm 3 A single step of the MH random walk

```
Input: Current state $x_n$ of the MH random walk
$y \sim f(\cdot|x_n)\mathrm{d}\mu$
$a \sim \mathcal{U}([0,1])$
if $a \le \rho(x_n, y)$ then
    $x_{n+1} \leftarrow y$
else
    $x_{n+1} \leftarrow x_n$
end if
return $x_{n+1}$
```

To see that the definition above indeed yields a Markov chain we convince ourselves that the transition kernel is given by

$$K(x, A) = \int_A \rho(x, y) f(y|x) \mu(\mathrm{d}y) + (1 - m(x))\delta_x(A)$$

where δ_x is the Dirac measure in x and

$$m(x) = \int_{\mathcal{X}} \rho(x, y) f(y|x) \mu(\mathrm{d}y) \in [0, 1]$$

is the *acceptance probability* of the chain at state x.

3.17 Proposition (Stationary distribution). *The probability measure π is a stationary distribution of the MH random walk.*

Proof. We have

$$\int_{\mathcal{X}} K(x,A)\pi(\mathrm{d}x) = \frac{1}{Z}\int_{\mathcal{X}}\left(\int_A \rho(x,y)f(y|x)\mu(\mathrm{d}y) + (1-m(x))\delta_x(A)\right) f(x)\mu(\mathrm{d}x) \quad (3.8)$$

We note that

$$\rho(x,y)f(y|x)f(x) = \rho(y,x)f(x|y)f(y).$$

Furthermore we can compute

$$\begin{aligned}\int_{\mathcal{X}} m(x)\delta_x(A)f(x)\mu(\mathrm{d}x) &= \int_A\int_{\mathcal{X}} \rho(x,y)f(y|x)\mu(\mathrm{d}y)f(x)\mu(\mathrm{d}x)\\ &= \int_{\mathcal{X}}\int_A \rho(x,y)f(y|x)f(x)\mu(\mathrm{d}x)\mu(\mathrm{d}y)\\ &= \int_{\mathcal{X}}\int_A \rho(y,x)f(x|y)\mu(\mathrm{d}x)f(y)\mu(\mathrm{d}y)\end{aligned}$$

where we used Fubini-Tonelli theorem[4] in the second to last step. We note that two of the terms in (3.8) cancel out and we obtain

$$\int_{\mathcal{X}} K(x,A)\pi(\mathrm{d}x) = \frac{1}{Z}\int_{\mathcal{X}} \delta_x(A)f(x)\mu(\mathrm{d}x) = \pi(A).$$

□

Now we aim to prove that the MH random walk is Harris recurrent because then the ergodic theorem yields that the empirical measures associated with the Markov chain will actually converge to π. Obviously this is not trues for all proposal families in general the case, for example we could consider the case where the proposal distribution $f(\cdot|x)$ is just the Dirac measure in x.[5] Then the MH random walk would never leave its initial position which will typically be a deterministic point. Hence, the empirical measures are only the Dirac measure in the starting point and will not converge towards π.

The first step towards Harris recurrence is to show irreducibility and this will already give us some hints to what families of proposal are sensible.

3.18 Proposition (Irreducibility). *Assume that the proposal family is strictly positive, i.e.*

$$f(y|x) > 0 \quad \textit{for all } x, y \in \mathcal{X}.$$

Then the MH random walk is π irreducible.

[4] The Fubini-Tonelli theorem states that the order of integration with respect to two σ-additive measures can be swapped, if the integrated function is non negative.

[5] Obviously this is slightly formal, because the Dirac measure can typically not be expressed through a density. However, rigorous examples can be constructed similarly.

Proof. For any measurable set $A \subseteq \mathcal{X}$ with positive measure $\pi(A) > 0$ we have

$$K(x, A) \geq \int\limits_A \rho(x, y) f(y|x) \mu(\mathrm{d}y) > 0.$$

To see this, we can assume that this would not hold and then the integrant has to be zero μ almost surely. Since $f(y|x)$ is strictly positive this would imply $\rho(x, y) = 0$ and hence $f(y) = 0$ for μ almost all $y \in A$. However, this is a contradiction to

$$\pi(A) = \int\limits_A f(y) \mu(\mathrm{d}y) > 0.$$

□

Now we can formulate the ergodicity for π irreducible MH random walks.

3.19 Theorem (Ergodicity of the MH random walk). *If the MH random walk is π irreducible, then it is also Harris recurrent and hence ergodic.*

Proof. We refer to Lemma 7.3 in [50] for the proof of Harris recurrency, the ergodicity then follows from the ergodic theorem. □

Implementation of the MH random walk

So far we have presented the theoretical foundations of the MH random walk and now we want to touch on a few aspect of the simulation of this Markov chain. For this part we shall point the reader towards the example based introductions [51] and [53] to the implementation of the MH random walk which also provide coding examples. We have seen that the empirical measures associated with the MH random walk converge to π under fairly mild assumptions, meaning for a wide class of proposal distributions. Nevertheless it is mostly the choice of the proposal that determines the speed of this convergence. In order to shortly demonstrate this effect, we consider the case $\mathcal{X} = \mathbb{R}^d$ and that the reference measure μ is the Lebesgue measure.

3.20 Choosing a proposal family. Usually one chooses the proposal such that the expectation of $f(\cdot|x)$ is x. The most common choice of a proposals is a family of normal distributions $f(\cdot|x)$ with expectation x and covariance $\Sigma \in \mathbb{R}^{d \times d}$. This also has the effect that the acceptance ratio takes the easier form

$$\rho(x, y) = \min\left\{\frac{f(y)}{f(x)}, 1\right\}.$$

Also since the densities are strictly positive we ensure that the resulting Markov chain is π irreducible.

3.21 Acceptance rate, autocorrelation and effective sample size. Once we have agreed to stick to normal densities for the proposal distributions, we still have the freedom to choose the covariance $\Sigma \in \mathbb{R}^{d \times d}$. This determines how far the proposed new values will be away from the current state of the Markov chain. The motivation for an aggressive proposal distribution, i.e. for a high variance would be that this would enable the Markov chain to take bigger steps and hence explore the space $\mathcal{X}$ faster. Also the chain would be more likely to jump between possibly isolated areas of high density. However, this could also lead to a high rejection rate[6] if the proposed values are often so far away from the current state of the Markov chain that they are lie within an area of low density. In this case the Markov chain will only ‚visit' few distinct points in the space $\mathcal{X}$ which is also very unfavourable. In fact the findings in [54] suggests that an acceptance rate around 25% is desirable in dimension $d \geq 3$ and around 50% for dimension $d = 1, 2$. The connection between the proposal distribution and the acceptance rate is also elaborated in the upcoming example.

The *autocorrelation function* (acf) of a sequence of data points $x_0, \ldots, x_n$ captures the estimated correlation between the observations. More precisely $\mathrm{acf}(k)$ gives the empirical correlation[7] of $(x_0, x_1, \ldots, x_{n-k})$ and $(x_k, x_{k+1}, \ldots, x_n)$. In the case that the data points are generated by a MH random walk, the autocorrelation function determines the correlation of the Markov chain at time l with the Markov chain at time $l + k$. Hence, if $\mathrm{acf}(k) < \varepsilon_0$ where $\varepsilon_0 > 0$ is fixed in advance, one can perceive $x_0, x_k, x_{2k}, \ldots$ as an independent sequence of samples from π – or more precisely an only weakly correlated one. The *effective sample size* is the length m of this new almost uncorrelated sequence $x_0, x_k, x_{2k}, \ldots, x_{mk}$. Obviously the effective sample size strongly depends on the choice of ε_0 that incorporates how much correlation one is willing to accept.

We should quickly touch on how the proposal affects the autocorrelation function and hence the effective sample size. Assume we have a very aggressive proposal distribution. Then we will typically have a high rejection rate and hence $x_l = x_{l+k}$ a lot of times meaning that the autocorrelation function will be high and therefore the effective sample size is rather low. On the other hand if the proposal is too conservative the MH random walk will only take very small steps and hence x_{l+k} will still be close to x_l. Therefore, the autocorrelation will be high and the effective sample size low again. This effect of the proposal can be seen in Figure 3.2.

3.22 Example (One dimensional MH). We follow an examples for a one dimensional MH random walk given in [51] and we want to approximate the probability distribution with unnor-

[6] The term should be rather intuitive; the rejection rate is the relative amount of rejections that occurred in the MH random walk and analogously the acceptance rate is its counterpart.

[7] This is the correlation of the two empirical measures associated with $(x_0, \ldots, x_{n-k})$ and $(x_k, \ldots, x_n)$.

malised density

$$f(x) := \sin(x)^2 \cdot \sin(2x)^2 \cdot \exp\left(-\frac{x^2}{2}\right).$$

The goal of this example is to see how different proposal distributions lead to different acceptance rates, a different exploration of the state space $\mathcal{X} = \mathbb{R}$ and different effective sample sizes. In order to achieve this, we run $2 \cdot 10^4$ samples of the MH random walk with starting point $x_0 = 1$ and three different values $\alpha = 0.01, 3, 100$ for the variance of the proposal distributions. Then we plot a histogram including the actual density and the autocorrelation function for all different values. The acceptance rates where approximately 88% for $\alpha = 0.01$, 34% for $\alpha = 3$ and 9% for $\alpha = 100$. The effective sample sizes for the three different chains different values for α are

$$\frac{2 \cdot 10^4}{50} = 4 \cdot 10^2, \quad \frac{2 \cdot 10^4}{8} = 2.5 \cdot 10^3 \quad \text{and } \frac{2 \cdot 10^4}{30} \approx 7 \cdot 10^2$$

for $\alpha = 0.01, 3, 100$ respectively.

This simulation illustrates the problem of too aggressive – $\alpha = 100$ – and too conservative – $\alpha = 0.01$ – proposal distributions and shows how this effects the acceptance rate and the effective sample size.

3.23 Tuning the proposal. In order to obtain a higher acceptance rate without simply choosing the variance of the proposal distribution small one can *tune* or *adapt* the proposal distribution. This means one adjusts the proposal distribution after a while, lets say after the first 10^3 samples in such a way that one replaces the original covariance matrix Σ by the empirical covariance of the first 10^3 samples. Then one forgets about all the samples so far – they are usually called the *burn in period* – and starts a new MH random walk. It is essential to drop the first samples since otherwise the Markov property would break as all further samples now rely on the coveriance of the first burn in period and hence on those points. The reason why this increases the acceptance rate is, that the proposal now only is aggressive in those directions where the mass of the density is widely spread. For a further discussion we refer to [55].

3.24 The Gelman-Rubin diagnostic. So far we have seen guidelines as what properties of the MCMC simulation can be seen as favourable or not. However those comments can not replace quantitative measures on the convergence of the simulated Markov chains and one of them is the Gelman-Rubin diagnostics which is also called the $\hat{R}$ value. We will not be able to rigorously introduce this quantity, but will make a few comments since we will use it later and refer to [52] for a thorough introduction to convergence diagnostics for MCMC methods and to [16] and [11] for the orinigal work by Gelman, Rubin and Brooks. In a nutshell the $\hat{R}$ value is an estimate of how much longer a MCMC simulation would have to run to be a good approximation of the stationary distribution. It is generally accepted that a $\hat{R}$ value of at most 1.05 can be taken as a sign of convergence although this might be misleading in certain cases, cf. [11].

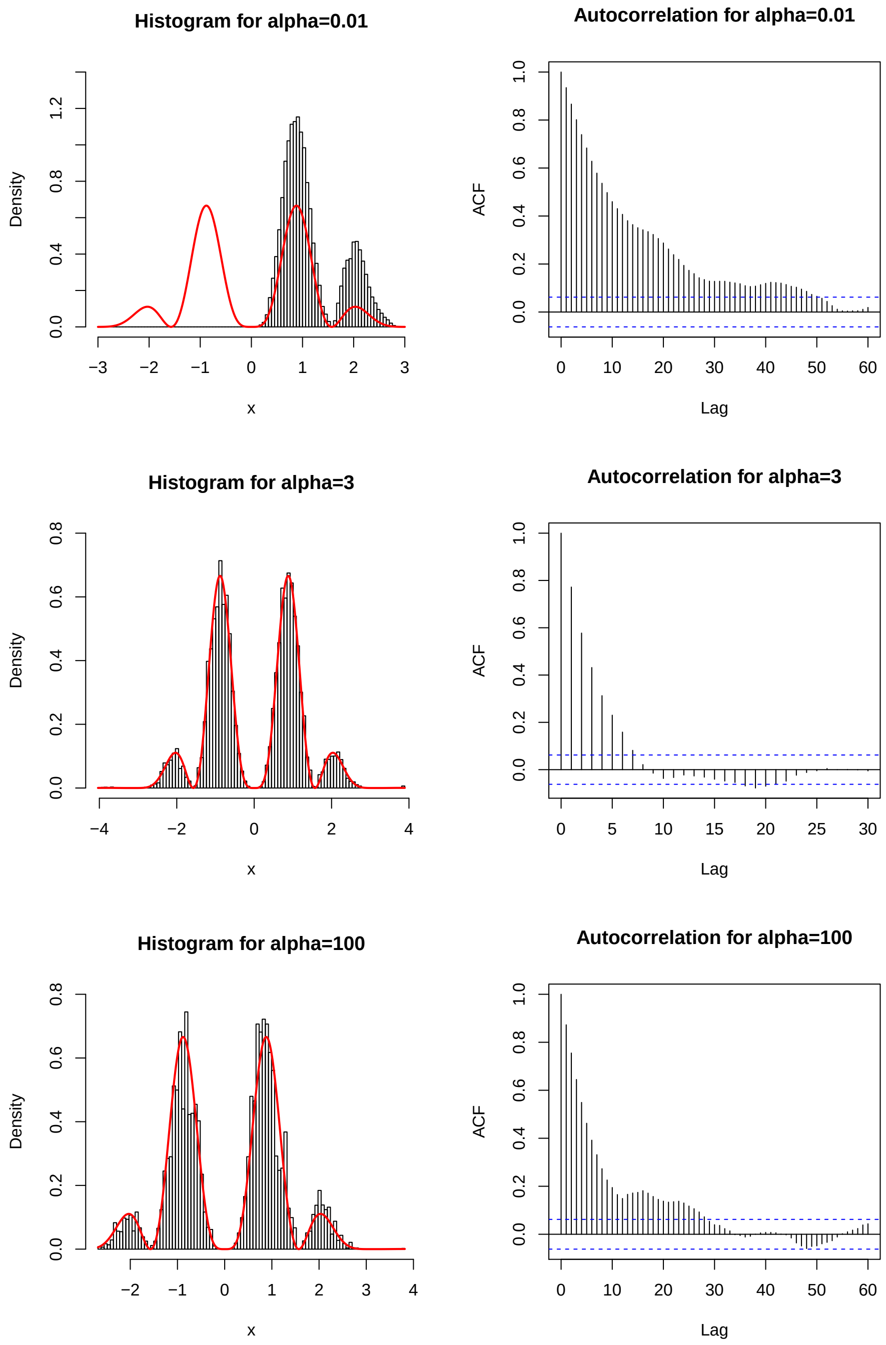


Figure 3.2: Histograms and autocorrelation functions of for three different variances α of the Gaussian proposal distributions. It is apparent that the histogram for $\alpha = 3$ fits the actual density the best and also that the autocorrelation decays the quickest for this parameter. Note that for $\alpha = 0.01$ the MH random walk only explored some area of high density. The actual density is obtained by numerical integration.

Although we don't introduce the statistics itself, we shall present the requirements to compute it. The procedure one has to take is the following:

(i) Find the possibly multiple modes of the distribution that should be approximated. This can be done either by exploiting optimisation algorithms or running short MCMC simulations, which we will do later in our toy example.

(ii) Run m MCMC simulations of length n starting at random points with variance greater than the estimated variance of the target distribution π. This variance is typically estimated through a first, shorter MCMC simulation which can also be used to tune the proposal.

Now the $\hat{R}$ value can be computed from the entirety of those m chains of length n and we will rely on a pre-implemented tool in R to do this.

III.2.3 Slice sampling

Slice sampling is a different MCMC method and quite similar to the MH random walk. Nevertheless it has the benefit that one does not have to define or tune a family of proposal distributions and that the constructed Markov chain is always irreducible. However, we will see that at least when one wants to simulate the slice sampling one runs into similar problems of having to choose a parameter that influences the auto correlation function and hence the speed of convergence of the method. We begin by fixing our frame we will work in.

3.25 Setting. Let $\mathcal{X}$ be a measurable space, μ a measure on that space and $f : \mathcal{X} \to [0, \infty]$ a function with finite integral

$$Z := \int\limits_{\mathcal{X}} f(x)\mu(\mathrm{d}x) \in (0, \infty).$$

In particular there is $\hat{x} \in \mathcal{X}$ such that $f(\hat{x}) > 0$. Our goal is to find an ergodic Markov chain with invariant distribution

$$\pi(A) := \frac{1}{Z} \int\limits_{A} f(x)\mu(\mathrm{d}x).$$

Further we will assume – after an eventual modification of f on a μ Null set – that

$$f \le \|f\|_{L^\infty(\mu)} = \inf \left\{ \alpha \in \mathbb{R} \;\middle|\; f \le \alpha \text{ almost surely with respect to } \mu \right\} \in [0, \infty].$$

3.26 The slice sampling method. Assume we have already given the first n samples $x_1, \ldots, x_n$ of the Markov chain. If we have $f(x_n) = 0$, then we set $x_{n+1} := \hat{x}$. Otherwise we sample y according to the uniform distribution on $[0, f(x_n)]$ and define the *slice*

$$S := S(y) := \left\{ x \in \mathcal{X} \mid f(x) \ge y \right\}.$$

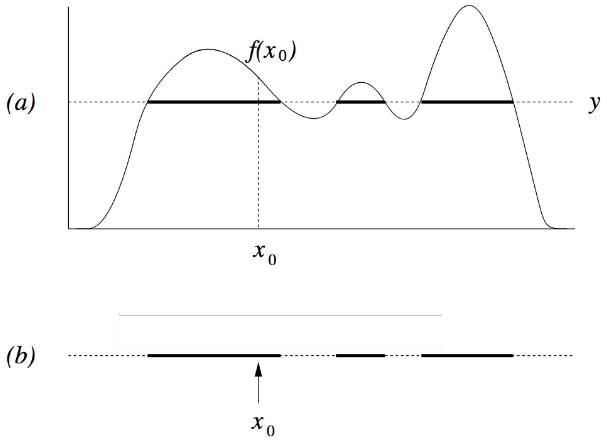


Figure 3.3: Schematic sketch of the selection of a slice: (a) first y is sampled uniformly in $[0, f(x_0)]$ and (b) the slice is selected. Original graphic from [44].

Note that because $y < f(x_n) \leq \|f\|_{L^\infty(\mu)}$ holds almost surely, we have $\mu(S) > 0$ as well as

$$\mu(S) \leq y^{-1} \int\limits_S f(x)\mu(\mathrm{d}x) < \infty$$

where we used Markov's inequality as well as $y > 0$ almost surely. Now draw x_{n+1} according to the uniform distribution[8] on S. Note that if $f(x_n) > 0$, then $f(x_{n+1}) \geq y > 0$ almost surely, hence $f(x_n) = 0$ can only hold for $n = 0$. Further the reason why we have to treat the case $f(x_n) = 0$ individually is, that there typically is no uniform distribution on the slice $S(0) = \mathcal{X}$. The pseudo code for the construction of the resulting Markov chain is presented in Algorithm 4.

Algorithm 4 A single slice sampling step

```
Input: Current state $x_n$ of the Markov chain
if then $f(x_n) = 0$
    $x_{n+1} \leftarrow \hat{x}$
else
    $y \sim \mathcal{U}([0, f(x_n)])$
    $S \leftarrow \{x \in \mathcal{X} \mid f(x) \geq y\}$
    $x_{n+1} \sim \mathcal{U}(S)$
end if
return $x_{n+1}$
```

[8] Of course we mean the uniform distribution with respect to μ that gives weight $\mu(S)^{-1} \cdot \mu(A)$ to a set $A \subseteq S$.

If we compare the Markov chain to the MH random walk, we notice that in the slice sampling we first create a random threshold y and then sample uniformly from all points that satisfy this threshold. This is just the other way round than in the MH random walk where we first make a proposal for the next state of the Markov chain and then decide whether we will accept it or not.

Just like in the case of the MH random walk we can explicitly give the transition kernel and use this expression then to check that π is a stationary distribution. The kernel of the Markov chain that arises from the slice sampling iteration is given by

$$\begin{aligned} K(x, A) &= \int_{\mathbb{R}} \frac{\mathbb{1}_{[0,f(x)]}(y)}{f(x)} \cdot \frac{\mu(A \cap S(y))}{\mu(S(y))} \lambda(\mathrm{d}y) \\ &= \int_{\mathbb{R}} \frac{\mathbb{1}_{[0,f(x)]}(y)}{f(x)} \cdot \frac{1}{Z(y)} \int_{A} \mathbb{1}_{[y,\infty)}(f(z))\mu(\mathrm{d}z)\lambda(\mathrm{d}y) \end{aligned}$$

where λ is the Lebesgue measure on $\mathbb{R}$, $\mathbb{1}$ is the indicator function and $Z(y)$ is the normalisation constant

$$Z(y) := \int_{\mathcal{X}} \mathbb{1}_{[y,\infty)}(f(z))\mu(\mathrm{d}z) = \mu(S(y)) \in (0, \infty).$$

Obviously the expression above only holds if $f(x) > 0$ and in the case $f(x) = 0$ we have

$$K(x, A) = \delta_{\hat{x}}(A).$$

3.27 Proposition (Invariant distribution). *The probability distribution π is a stationary distribution of the Markov chain associated with the slice sampling method.*

Proof. For any $A \subseteq \mathcal{X}$ we can compute

$$\begin{aligned} \int_{\mathcal{X}} K(x, A)\pi(\mathrm{d}x) &= \frac{1}{Z} \int_{\mathcal{X}} \int_{\mathbb{R}} \frac{\mathbb{1}_{[0,f(x)]}(y)}{f(x)} \cdot \frac{1}{Z(y)} \int_{A} \mathbb{1}_{[y,\infty)}(f(z))\mu(\mathrm{d}z)\lambda(\mathrm{d}y) f(x)\mu(\mathrm{d}x) \\ &= \frac{1}{Z} \int_{A} \int_{\mathbb{R}} \frac{1}{Z(y)} \int_{\mathcal{X}} \mathbb{1}_{[y,\infty)}(f(x))\mu(\mathrm{d}x)\mathbb{1}_{[0,f(z)]}(y)\lambda(\mathrm{d}y)\mu(\mathrm{d}z) \\ &= \frac{1}{Z} \int_{A} f(z)\mu(\mathrm{d}z) = \pi(A) \end{aligned}$$

where we again used Fubini's theorem for non negative functions. □

3.28 Proposition (Irreducibility). *The Markov chain that arises from the slice sampling algorithm is π irreducible.*

Proof. Fix $A \subseteq \mathcal{X}$ with positive probability $\pi(A) > 0$ and $x \in \mathcal{X}$. If we have $f(x) > 0$, then we have $\mu(A \cap S(y)) > 0$ for one $y \in (0, f(x))$. We obtain

$$K(x, A) \geq \int_{\mathbb{R}} \frac{\mathbb{1}_{[0,y]}(z)}{f(x)} \cdot \frac{\mu(A \cap S(z))}{\mu(S(z))} > 0.$$

If however $f(x) = 0$, then we get

$$K^2(x, A) = K(\hat{x}, A) > 0.$$

□

3.29 Theorem (Ergodicity). *If f is bounded, the Markov chain induced by the slice sampling method is ergodic.*

Proof. See Theorem 6 in [43]. □

Implementation details

Just like in the case of the MH random walk we will make a few comments about the actual simulation of the slice sampling algorithm and for this, we will assume $\mathcal{X} \subseteq \mathbb{R}^d$.

The main difficulty in the implementation is the sampling of a uniform distribution on a slice S. In practice it is not even possible to calculate the slice but one can exploit the following observation. Assume that we are able to simulate a uniform distribution on a set C that contains the slice S. Then the following algorithm – which is nothing but the conditioning of this uniform distribution on the event that the outcome is in S – samples uniformly from S.

Algorithm 5 Sampling from a uniform distribution on a subset $S \subseteq C$

```
Input: $S$ and $C \supseteq S$
$x \sim \mathcal{U}(C)$
while $x \notin S$ do
    $x \sim \mathcal{U}(C)$
end while
return $x$
```

An obvious choice for C would be a cuboid

$$C = \prod_{i=1}^{d} [a_i, b_i]$$

since it is straight forward to sample from a uniform distribution on a cuboid. Namely one only has to sample the individual coordinates uniformly in the intervals $[a_i, b_i]$. The problem still remains how one can find a cuboid that surely contains the whole slice S. The short answer is that there is no general way to do this. However, not everything is lost, since we can use random cuboids that have the property that every part of the slice is contained in the cuboid with positive probability. This will be crucial in retaining the irreducibility of the Markov chain. In fact it has been found that in applications the following procedure works well (cf. [1]). Given the current state x_n of the Markov chain, we propose a random interval $[a_i, b_i]$ around the i-th component of x_n. Then we extend those intervals until the endpoints a and b of the cuboid do not lie in the slice anymore which is described in Algorithm 6.

Algorithm 6 Sampling a random cuboid

```
Input: Current state $x_n$ of the Markov chain, parameter $\alpha > 0$
for $i = 1, \dots, d$ do
    $a_i, b_i \sim \mathcal{E}(\alpha)$
end for
$a \leftarrow (a_1, \dots, a_d), b \leftarrow (b_1, \dots, b_d)$
while $x - a \in S$ do
    $a \leftarrow 2 \cdot a$
end while
while $x + b \in S$ do
    $b \leftarrow 2 \cdot b$
end while
return $(x - a, x + b)$
```

Here $\mathcal{E}(\alpha)$ denotes the exponential distribution with parameter α and determines how large the first proposed intervals are. It is straight forward and computationally very easy to determine whether a point x is in the slice $S(y)$ since one only has to check $f(x) \geq y$. The reason for the choice of the exponential distribution is that this ensures that the cuboid can get arbitrarily large with positive probability. This leads to the effect that the Markov chain one obtains in exchanging the sample from $\mathcal{U}(S)$ by a sample from $\mathcal{U}(S \cap C)$ still is irreducible. To see this we can slightly modify the proof of irreducibility, so for $A \subseteq \mathcal{X}$ with positive probability we choose $y > 0$ such that $\mu(A \cap S(y)) > 0$. Further we can choose a cuboid C around x such that $\mu\big(A \cap S(y) \cap C\big) > 0$. Further this cuboid is contained in the cuboid proposed by Algorithm 6 with positive probability and hence we have

$$K(x, A) = \mathbb{P}_x(X_1 \in A) > 0.$$

The algorithm that arises from the combination of the usual slice sampling method and the approximation of the uniform distribution on the slice is presented in Algorithm 7. It shall be noted, that this algorithm also uses a point $\hat{x}$ of positive density, which can be determined easily for a lot of densities f. If this is not straight forward, one can sample x_0 according to a normal distribution until we select a point of positive density.

Obviously the algorithm presented above produces a Markov chain that is not identical with the one presented in the theoretical discussion of the slice sampling method. However, if one wants

Algorithm 7 Algorithm for the slice sampling

```
Input: Unnormalised density $f$, starting value $x_0$, desired length $n$ of the chain, $\alpha > 0$
if $f(x_0) = 0$ then
    $x_0 \leftarrow \hat{x}$
end if
for $i = 0, \dots, n-1$ do
    $y \sim \mathcal{U}([0, f(x_i)])$
    $C$ random cuboid around $x_i$ with parameter $\alpha$
    $x \sim \mathcal{U}(C)$
    while $f(x) < y$ do
        $x \sim \mathcal{U}(C)$
    end while
    $x_{i+1} \leftarrow x$
end for
return $x = (x_0, \dots, x_n)$
```

to ensure the convergence of this slightly modified Markov chain, one has to check whether π remains a stationary distribution and whether the chain is still ergodic. This is usually done in the specific setting one works in, cf. [44]. We will quickly discuss this in a very easy case. Namely let us assume $d = 1$ and that f is continuous and has only one local maximum. We call f *unimodal* in this case and note that every slice $S(y)$ is an interval. Hence, the proposed cuboid is an interval around $x_n \in S(y)$ such that both endpoints are outside of the slice $S(y)$ and hence we have $S(y) \subseteq C$. Therefore, the algorithm above is equivalent to the original slice sampling method and hence produces an ergodic Markov chain with the desired invariant distribution.

3.30 The choice of α. One could think that a small choice of α – which relates into large values of a_i and b_i – would be the best since this increases the probability that the whole slice S is contained in the cuboid C. There is some truth in this approach, since $\mathcal{U}(S \cap C)$ is a better approximation of $\mathcal{U}(S)$ if C is larger and further the while loops in Algorithm 6 need less repetitions if a_i and b_i initially are big. However, one should not choose α too small, because a large cuboid C also means that a lot of samples from $\mathcal{U}(C)$ will lie outside of $S \cap C$. Hence, Algorithm 5 that samples from $\mathcal{U}(S \cap C)$ will get slower as it will reject a lot of samples.

In conclusion there is a trade off in terms of computation time between the choice of too small and too large values for α. However not always the parameter α that minimises the simulation time is the most suitable, since the autocorrelation decreases together with the parameter α. Hence, computation time should rather be compared to the effective sample size.

Those effects of α on the auto correlation and therefore effective sample size can be seen in Figure 3.4 where the procedure of Example 3.22 is repeated but this time with the slice sampling method. The sample size remains $2 \cdot 10^4$ and the different parameter choices where $\alpha = 0.01, 0.5, 10$. The according computation times where approximately 26s for $\alpha = 0.01$, 1.7s for $\alpha = 0.5$ and 1.7s for $\alpha = 10$. In regard of the decay of the autocorrelation functions and the resulting effective sample sizes, it is apparent that the choice $\alpha = 0.5$ would be the most sensible one in this case.

III.2.4 Variational MCMC methods

Now that we have presented a general setup for MCMC methods we wish to use them to approximated the posterior distribution which is given by the unnormalised density

$$f(\theta) = f_\Theta(\theta) \prod_{i=1}^{n} \frac{\det(L(\theta)_{Y_i})}{\det(L(\theta) + I)}. \tag{3.9}$$

In the light of the theoretical guarantees this will surely work and actually Figure 3.1 has been created this way. However, the evaluation of this unnormalised density f can take several seconds or even minutes itself since it involves the computation of the determinant of the $N \times N$ matrix $L(\theta) + I$. However, one can efficiently compute bounds of the unnormalised density and we will provide a general setup of how the MH random walk and slice sampling can be expressed using those bounds. This will lead to significantly shorter simulation times for the respective MCMC methods.

3.31 Setting. Let $\mathcal{X}$ be a measurable space, μ a measure on that space and $f \colon \mathcal{X} \to [0, \infty]$ a function with finite positive integral

$$Z = \int_{\mathcal{X}} f(x)\mu(\mathrm{d}x) \in (0, \infty).$$

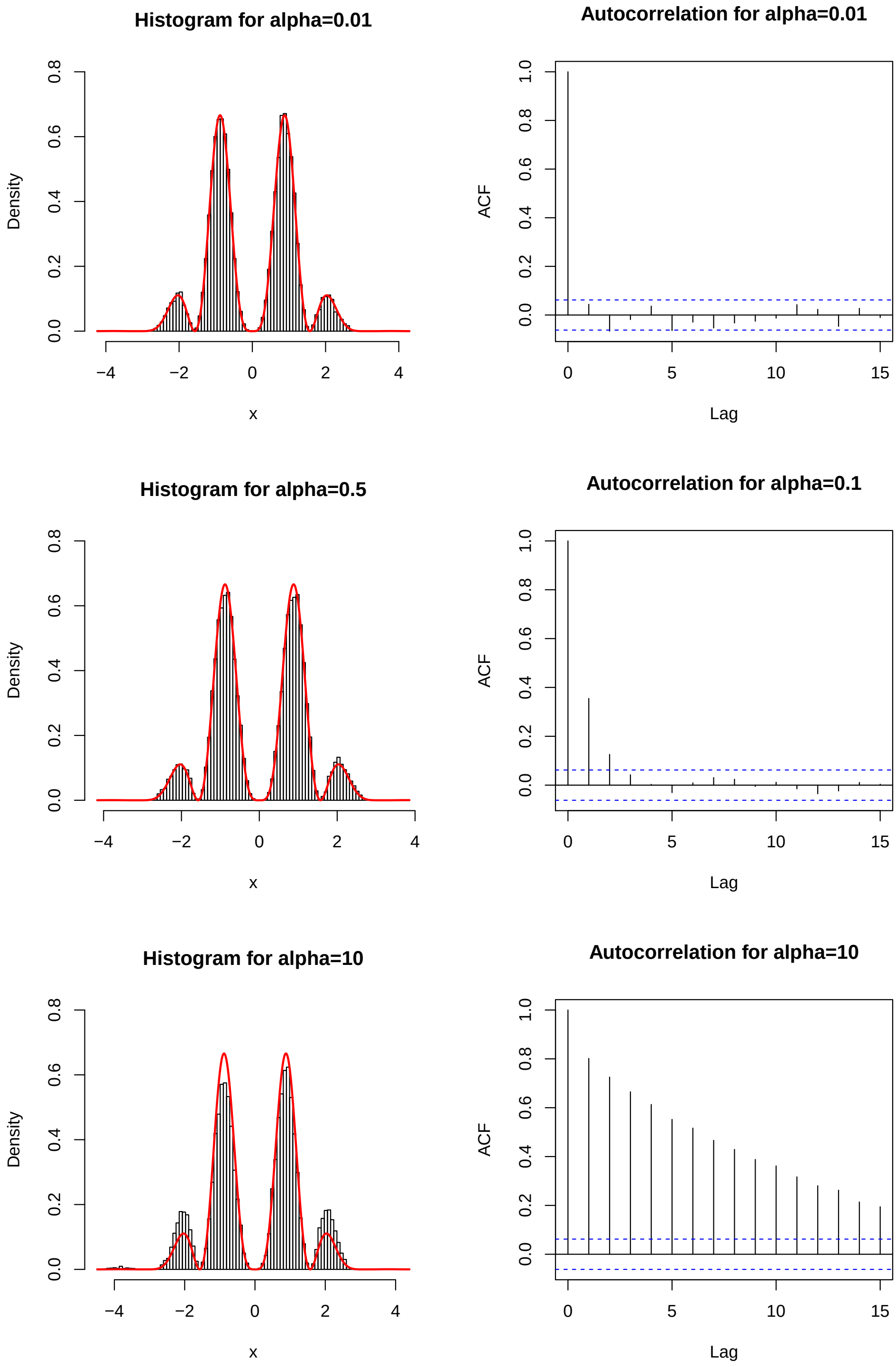
Let further $f \le \|f\|_{L^\infty(\mu)}$ and let

$$\{f(\cdot|x) \mid x \in \mathcal{X}\}$$

be a family of proposal distributions. Let now $f_n^-, f_n^+ \colon \mathcal{X} \to [0, \infty]$ be functions such that $f_n^-(x) \le f(x) \le f_n^+(x)$ for all $x \in \mathcal{X}$ as well as

$$f_n^\pm(x) \xrightarrow{n\to\infty} f(x) \quad \text{for all } x \in \mathcal{X}.$$

We seek an expression of the MH random walk and the slice sampling method that purely relies on those bounds f_n^- and f_n^+ of the unnormalised density.



Histogram for alpha=0.01

Autocorrelation for alpha=0.01

Histogram for alpha=0.5

Autocorrelation for alpha=0.1

Histogram for alpha=10

Autocorrelation for alpha=10

Figure 3.4: Histograms and autocorrelation functions for the choices of $\alpha = 0.01, 0.5, 10$. The auto correlation obviously decreases the fastest for $\alpha = 0.01$, however the computation time is much higher than for the parameter the other parameters.

Variational MH random walk

We note that the only part in the algorithm for the MH random walk where f is needed itself is the accept-reject step, hence, it suffices to adjust this step. In order to achieve this we bound the acceptance rate through

$$\rho_n^{\pm}(x,y) := \min\left\{\frac{f_n^{\pm}(y)f(x|y)}{f_n^{\mp}(x)f(y|x)}, 1\right\}.$$

In fact we obviously have $\rho_n^-(x,y) \leq \rho(x,y) \leq \rho_n^+(x,y)$ as well as

$$\rho_n^{\pm}(x,y) \xrightarrow{n\to\infty} \rho(x,y) \quad \text{for all } x, y \in \mathcal{X}.$$

Hence if we want to decide whether a number a satisfies $a \leq \rho(x,y)$ we can iteratively tighten the upper and lower bounds on ρ until we either have $a \leq \rho_n^-(x,y)$ and thus $a \leq \rho(x,y)$ or $a > \rho_n^+(x,y)$ and therefore $a > \rho(x,y)$. Now we can adjust the algorithm of the MH random walk accordingly and obtain Algorithm 8.

Algorithm 8 One step in the variational MH random walk

```
Input: Current state $x_n$ of the MH random walk
$y \sim f(\cdot|x_n)\mathrm{d}\mu$
$a \sim \mathcal{U}([0,1])$
$k \leftarrow 1$
while $a > \rho_k^-(x_n, y)$ and $a \leq \rho_k^+(x_n, y)$ do
    $k \leftarrow k + 1$
end while
if $a \leq \rho_k^-(x_n, y)$ then
    $x_{n+1} \leftarrow y$
else
    $x_{n+1} \leftarrow x_n$
end if
return $x_{n+1}$
```

Variational slice sampling

In the slice sampling we use the unnormalised density twice. The first time when sampling $y \sim \mathcal{U}([0, f(x_n)])$ and the second time when checking $x \in S(y)$ or equivalently $f(x) \geq y$. For the first problem we note that we surely have $[0, f(x_n)] \subseteq [0, f_1^+(x_n)]$ and hence we can use Algorithm 5 to sample uniformly from $[0, f(x_n)]$. However, in this algorithm we need to check $y \in [0, f(x_n)]$ or equivalently $f(x_n) \geq y$ which is just what we had to do determine whether $x \in S(y)$. Therefore, it suffices to see how one can check $f(x) \geq y$ which we will do analogously to the variational MH random walk by gradually tightening the bounds. This yields Algorithm 9 that returns ‚TRUE‘ if $f(x) \geq y$ and ‚FALSE‘ otherwise.

Algorithm 9 Deciding $f(x) \geq y$ through the bounds

```
Input: $y \in \mathbb{R}$ and $x \in \mathcal{X}$
$k \leftarrow 1$
while $y > f_k^-(x)$ and $y \leq f_k^+(x)$ do
    $k \leftarrow k + 1$
end while
if $y \leq f_k^-(x_n, y)$ then
    return TRUE
else
    return FALSE
end if
```

In conclusion we can express both MCMC methods exactly through those bounds as long as the bounds converge. This enables a fast simulation of the Markov chains if the unnormalised density is slow but the bounds $f_n^{\pm}$ are easy to compute. In the case that f is the posterior (3.9) of a DPP such bounds are given in [1] and [3].

Chapter IV

A toy example: Learning the log linearity constant of a spatial DPP

We will apply the MLE and the Bayesian estimation for one log linear model in a controlled environment, i.e. where the data is generated by ourselves. We will see how the Bayesian approach allows to encode more information and how the regulariser or prior affects the estimation and will quickly discuss how this impacts the noise sensitivity of the estimation.

We continue the example of the DPP on a two dimensional grid in the unit square from the first chapter. For this we note that for a 100×100 grid the evaluation of the elementary probabilities

$$f_{\mathbf{Y}|\Theta}(A|\theta) = \frac{\det(L(\theta)_A)}{\det(L(\theta) + I)}$$

would involve the calculation of a determinant of a $10^4 \times 10^4$ matrix and even the storage of such a matrix would pose a problem since it consists of 10^8 numbers. If the storage of a real number is done in the double-precision floating-point format, it takes 64bit per number and the space required to store the entire matrix is $64 \times 10^8\text{bit} = 800\text{MB}$, so almost one Gigabyte.[1] This makes even the computation of the log likelihood function very time consuming, let alone its maximisation. Because of those computational hindrances we will decrease the size but the ideas remain exactly the same.

4.1 Setting. We set

$$\mathcal{Y} := 39^{-1}\{0, \dots, 39\}^2$$

and obtain a 40×40 grid in the unit square. We again choose $\mathcal{R} := \mathcal{Y}$ and f to be

$$f(x) := \exp\left(-8 \cdot x^2\right)$$

[1] One byte is defined to be 8 bits. The units of Megabytes and Gigabytes are defined in the familiar way and denoted by MB and GB respectively.

and set

$$(\phi_i)_j \propto f(\|i-j\|) \quad \text{for } i,j \in \mathcal{Y}.$$

Further we choose the qualities to be be decreasing with the distance from the centre m of the and set

$$q_i := e^6 \cdot \exp\big(-10\,\|i-m\|\big) = \exp\big(-10\,\|i-m\|+6\big).$$

The goal is to estimate the two parameters that characterise the qualities, which are e^6 and -10. In order to do this we note that the qualities are given by a log linear model since we have

$$q_i = \exp(\theta_0^T f_i) \quad \text{where } f_i = \begin{pmatrix} \|i-m\| \\ 1 \end{pmatrix} \text{ and } \theta_0 = \begin{pmatrix} -10 \\ 6 \end{pmatrix}.$$

Hence we should be able to estimate this log linearity constant $\theta \in \mathbb{R}^2$ based on some data that is distributed according to this DPP. To do this we generate $n = 20$ samples $Y_1, \dots, Y_n$ from the DPP using the sampling algorithm introduced in the first chapter.

IV.1 MLE and regularised MLE

In order to perform the maximum likelihood estimation for the log linearity constant, we need to fix a similarity kernel $\hat{S}$, but since we know the exact kernel, we can simply set $\hat{S}_{ij} := \phi_i^T \phi_j$. Then we maximise the log likelihood over $\mathbb{R}^2$ using a pre-implemented optimisation algorithm in R. The resulting estimate was

$$\hat{\theta} = \begin{pmatrix} -10.000250 \\ 6.007382 \end{pmatrix} \tag{4.1}$$

but from the consistency results we already knew that it should be close to the actual parameter for large sample sizes. Since we also want to investigate the effect of a regulariser, we define two different regularisers

$$R_1(\theta) := -\frac{\|\theta\|^2}{2^4} \quad \text{and } R_1(\theta) := -\frac{\|\theta\|^2}{2}$$

as a regulariser. Note that those corresponds to the priors

$$f_{\Theta_1}(\theta) = \exp\left(-\frac{\|\theta\|^2}{2^4}\right) \quad \text{and } f_{\Theta_2}(\theta) = \exp\left(-\frac{\|\theta\|^2}{2}\right) \tag{4.2}$$

which are – up to scaling – Gaussian priors with different variance. Under those regularisations, the respective MAP estimators obtained were

$$\hat{\theta}_1 = \begin{pmatrix} -9.870850 \\ 5.952101 \end{pmatrix} \quad \text{and } \hat{\theta}_1 = \begin{pmatrix} -9.050763 \\ 5.597739 \end{pmatrix}.$$

It is not surprising that regularised MLEs are closer to zero, since the regulariser is built to penalise large parameter values. Further the effect of the Gaussian prior with smaller variance – which is the second one – is larger, which is consistent with the heuristic explanation that it cooperates a stronger prior believe, since it expects the parameter to be close to zero.

We chose the sample size of $n = 20$ relatively small and want to show the effect the sample size has on the estimation. Obviously, we know from the second chapter that the different MLE will converge to the actual parameter. To show this convergence in our case, we iteratively raised the sample size from 1 to 30 and obtained the maximum likelihood estimators that are shown in Figure 4.1. We can see that a stronger regularisation leads to a slower convergence, which is to be expected since it takes longer for the contribution $\frac{1}{n} \cdot R$ to the maximised function to decrease.

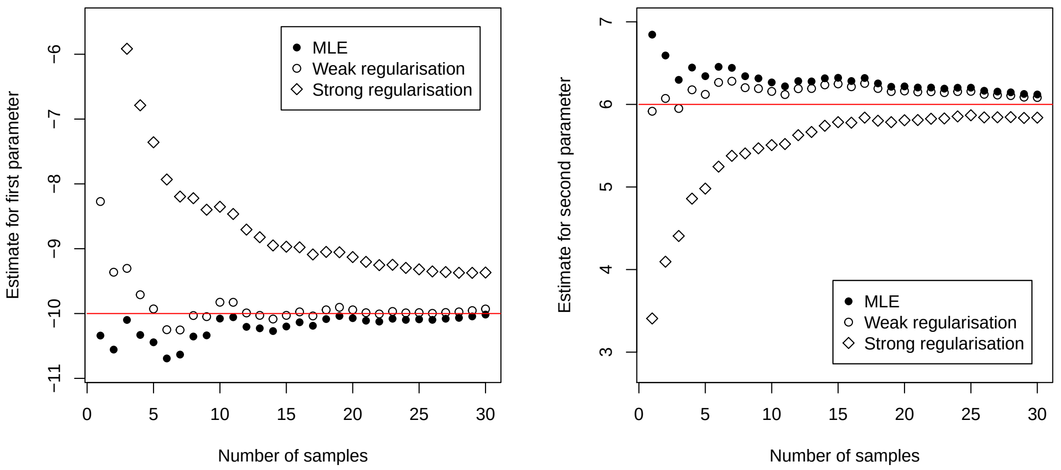


Figure 4.1: Plot of the progression of the MLE and regularised MLE. The real parameter is marked by the red lines and with increased sample size they both estimators are within a reasonably small margin.

In the light of the comparison of the different regularisers to the unregularised MLE and also the true parameter values, it is evident that the first regularisation R_1 is more suitable. We will also explain how one can compare those two regularisers without knowing the true parameter values or even without solving the maximisation problem assiciated with the MAP estimation.

IV.2 Bayesian estimation using MCMC methods

We will use the same data set consisting of $n = 20$ samples and will use the first prior in (4.2) since we have seen that it is more appropriate and will see that also the Bayes factor strongly supports this choice. The posterior of the log linearity parameter is given by

$$f_{\Theta|\mathbf{Y}^n}(\theta|Y_1,\dots,Y_n) \propto f_\Theta(\theta)\prod_{i=1}^n \frac{\det(L(\theta)_{Y_i})}{\det(L(\theta)+I)}.$$

and we will use the MH random walk to approximate it. But before we do this, we will shortly discuss how the Bayes factor can be used to decide between two different priors if one has not access to the unregularised and regularised MLE like we did before.

Comparing the different priors via the Bayes factor

We have introduced the Bayes factor as the ratio of the total observation probabilities under two models. In order to compare the two different priors (4.2), we need to integrate the observation probabilities over the parameter space, i.e. compute the integrals

$$\int_\Theta f_{\mathbf{Y}^n|\Theta}(Y_1,\dots,Y_n|\theta) f_{\Theta_i}(\theta)\nu(\mathrm{d}\theta)$$

which we will do numerically. The order of magnitude of the approximated Bayes factor between the two models arising from the priors (4.2) was 10^{22} which strongly supports the claim that the first prior is the more sensible choice. Therefore we will only work with this one in the remainder.

It shall be noted that the numerical integration carried out above can only be performed in an efficient way if the parameter space Θ is rather low dimensional. If this is not the case one can exploit probabilistic approaches based Monte Carlo simulations to calculate this normalisation constant. Details on such approaches can be found in the section on Monte Carlo integration in [50].

Approximation of the posterior using MCMC simulations

In order to approximate the posterior density through a MH random walk we proceed in the following steps.

4.2 First burn in to find a starting point. In the first phase we want to find an area of high density and in order to do this we simulate MH random walks with different starting positions and try to identify the regions where they get stuck in. This will typically happen in areas of at least locally highest probability. We have already seen that in order to obtain a reasonable MH random walk one has to choose a suitable proposal family. We use Gaussian proposals $f(\cdot|\theta)$ that are centered at θ and adjust the variance such that we obtain an acceptance rate of roughly $25\% - 75\%$.

Although there is no rigorous method to choose the variance of the proposal distribution at this point we make to general observations. A very high acceptance rate hints to the fact that every proposed step is within a region of almost equal density and hence one probably has to increase the variance and hence the proposed step size. On the other hand if the acceptance rate is close to zero is usually due to the fact that one proposes mostly steps into areas of very low density and hence it is reasonable in most cases to decrease the variance.

Once the variance is adjusted we run a first simulation of length $2 \cdot 10^2$ in order to see where the MH random walk is going to focus. We take the mean value of the second half of the samples as a measure of the area where the MH random walk spends most of its time. Here, we neglect the first half since it is very highly dependent on the starting point and it shall be noted that if a state of the Markov chain has high density, the chances are rather high that it will stay there for a few more steps and hence this point is weighted more heavily in the mean of the random walk. The positions of the Markov chain are shown in Figure 4.2 for different starting positions of the MH random walk and we notice that the mean of their second half hardly depends on the initial state.

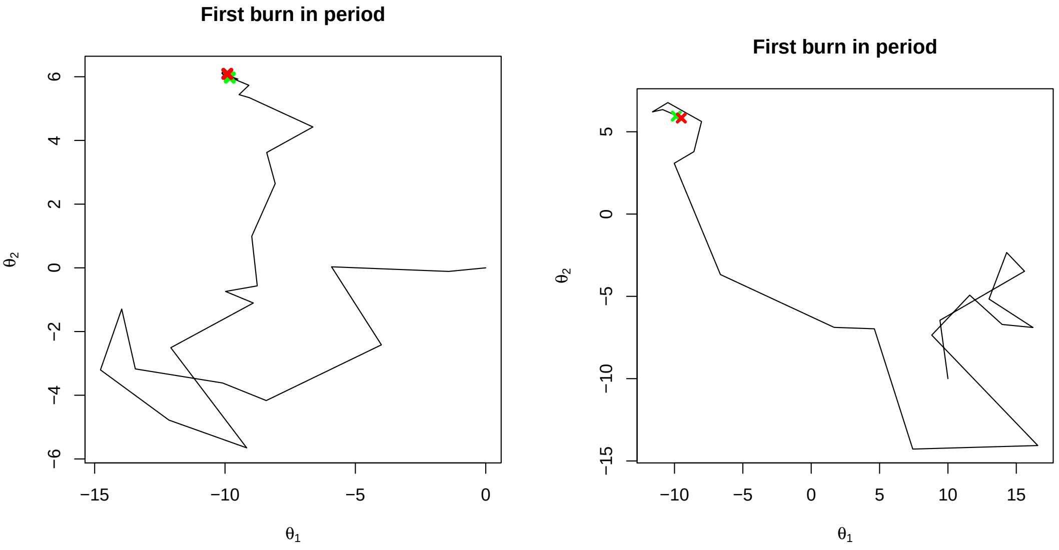


Figure 4.2: A plot of the first burn in period with two different starting points – the origin and $(10, -10)$. The regularised MLE for the log linearity constant is marked by the green cross and the mean of the second half of the random walk by the red cross.

Further the plots of the states of the Markov chain in Figure 4.3 show that the acceptance rate drops significantly after roughly 20 steps. This is a sign that we were successful in the process of finding an area with high density, since a lot of rejections imply, that the proposed points had significantly lower density.

One could argue that it would be reasonable to choose the (regularised) MLE as a starting point for the random walk. However, since we partly motivated the Bayesian approach to be an alternative to the infeasible maximum likelihood estimation for the elementary kernel L, we

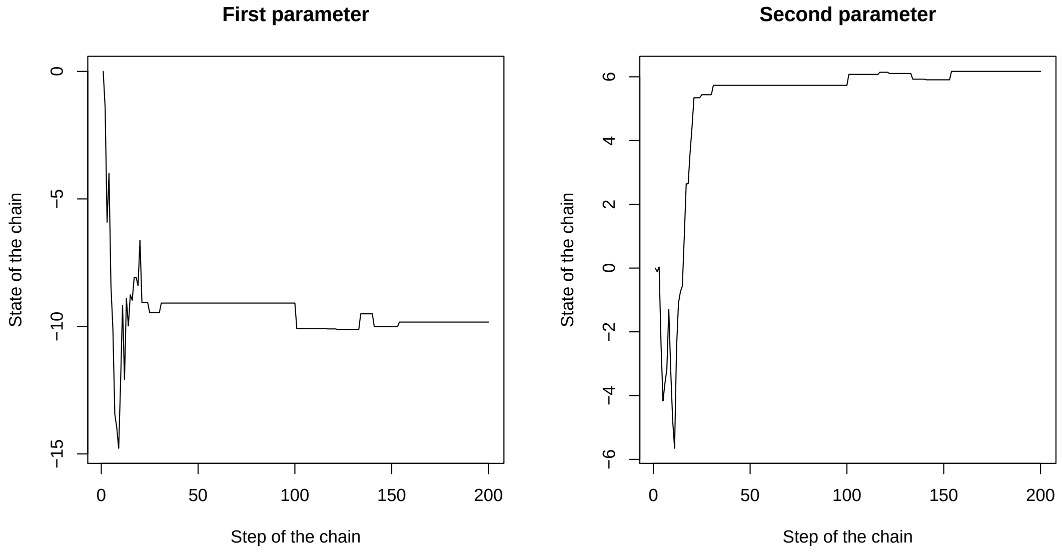


Figure 4.3: Plots of the two state parameters of the MH random walk starting at the origin. The acceptance rate drops significantly and hardly any proposals are accepted.

presented the procedure above that can also be used for the estimation of L.

4.3 Second burn in to tune the proposal. We use the second burn in period to tune the proposal according to 3.23 for the final simulation. To do this we first select a starting point according to the result of the first burn in period. Then we adjust the variance of the Gaussian proposals such that we obtain a reasonable acceptance rate. The variance of the proposals will be much smaller than the one of the first burn in since we have seen in the state plots of the first burn in period that the acceptance rate decreased heavily. In a heuristic way it can be said that one now works ‚locally‘ and tries to explore the finer structure of the distribution and has to take smaller steps in order to do so. We run this MH random walk for $5 \cdot 10^2$ samples which are shown in Figure 4.4 and calculate their empirical covariance $\Sigma \in \mathbb{R}^{2\times 2}$. We see that the points are located around the regularised MLE and we can get a first idea along which direction the parameter is more uncertain.

4.4 The actual MCMC simulation. In this final step we simulate a MH random walk of length 10^4 and with the same starting point as in the second step. Now we use the prior adjusted according the second burn in period. This means we choose $f(\cdot|\theta)$ to be the density of a normal distribution centered at θ and with covariance Σ. This leads to a higher acceptance rate of 60% compared to 21% in the second burn in period which can also be seen in the according state plots in Figure 4.5 and 4.6. Further we see in Figure 4.7 that the auto correlation function decreases faster with this tuned proposal. Finally we use a pre-implemented interpolation method to obtain a smoothed twodimensional histogram – also called a heat plot – which is shown in Figure 4.8.

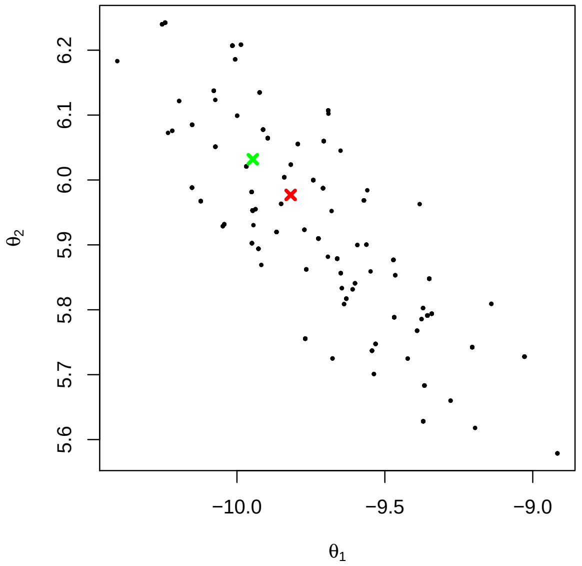


Figure 4.4: A plot of the samples of the MH of the second burn in period. One can see how the points are distributed around the regularised MLE which is marked red, the MLE is marked green. Their empirical covariance will be used to tune the proposal.

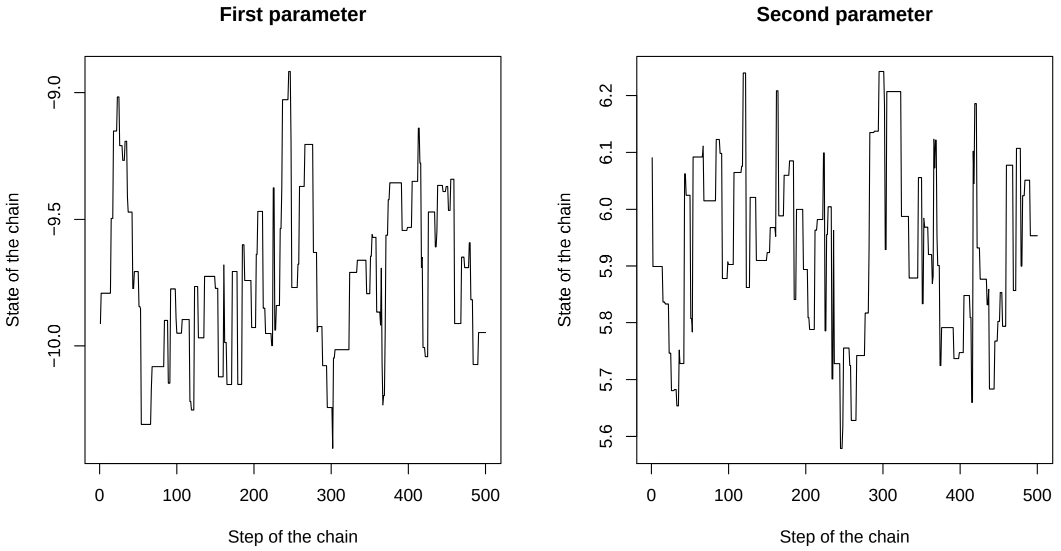


Figure 4.5: State plots of the second burn in period. One can see that the acceptance rate does not change like in the first burn in.

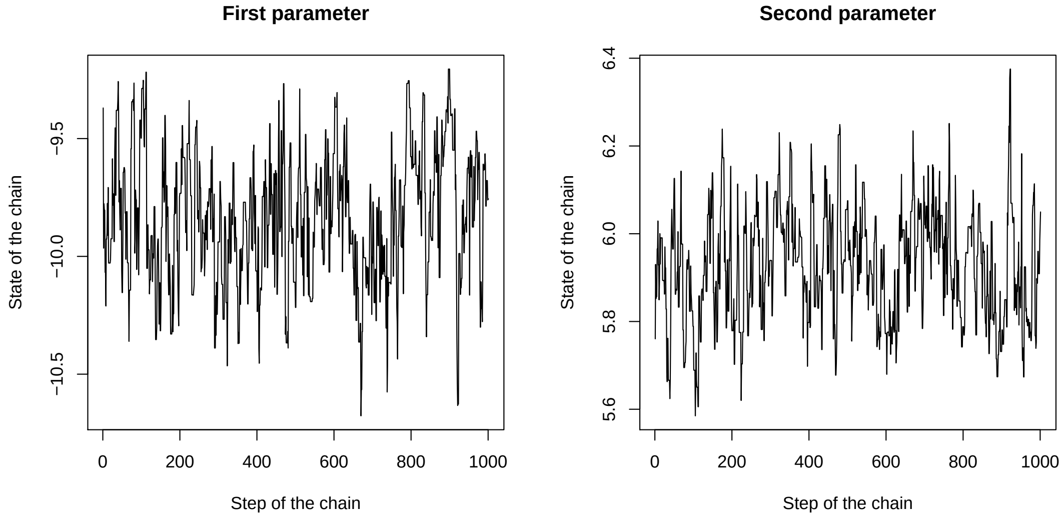


Figure 4.6: State plots of the final MH random walk on the bottom. One can see how the tuned proposal gives a higher acceptance rate than in the second burn in period.

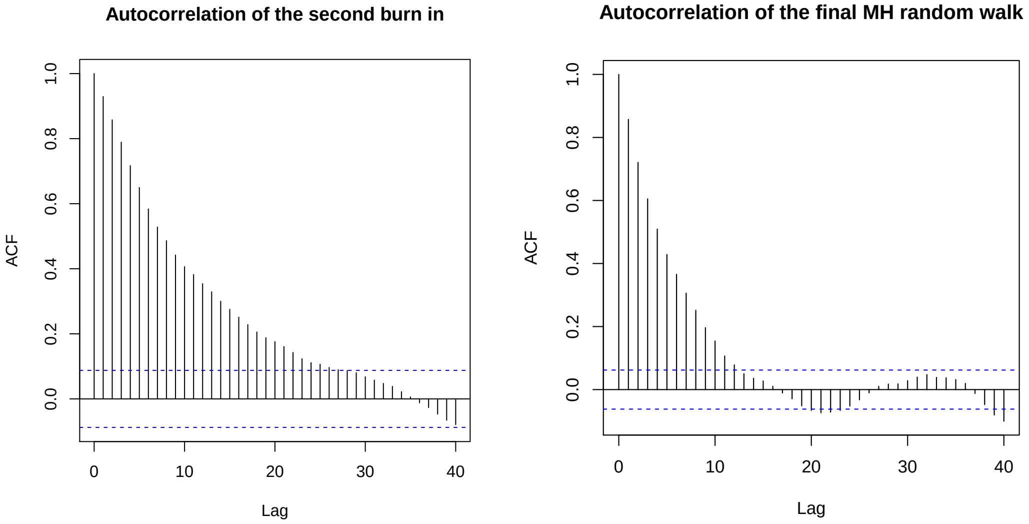


Figure 4.7: A plot of the autocorrelation functions of the second burn in period and the final MH random walk. The latter one decreases faster which hints to a faster convergence due to the tuned proposal distributions.

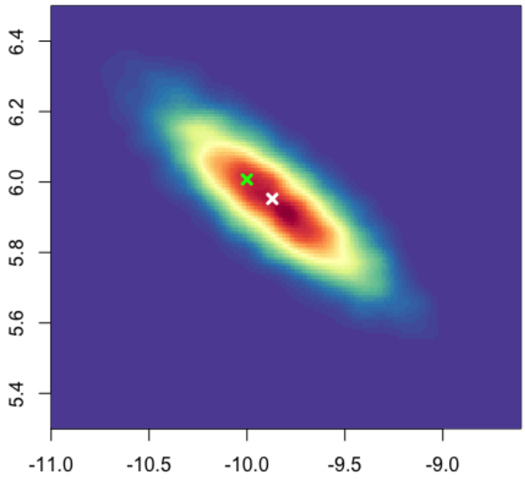


Figure 4.8: Heat map of the MH random walks with 10^4 iterations. The regularised MLE estimator is shown as a white and the MLE as a green cross. The regularised MLE is the maximum of the (approximated) posterior.

4.5 GELMAN-RUBIN DIAGNOSTIC. In order to justify the length of 10^4 of our final MCMC simulation for the approximation of the posterior we use the Gelman-Rubin diagnostic. Hence, we run a second chain with a random starting value sampled from a Gaussian distribution centered around the mean of the second half of the first burn in period and with twice the variance of the second burn in period. Then we use the pre-implemented R function `gelman.diag` that computes the $\hat{R}$ value and an upper estimate for it and obtain the following results.

	$\hat{R}$ value	upper estimation of $\hat{R}$
First parameter	1.01	1.06
Second parameter	1.02	1.09

Table IV.1: Table with $\hat{R}$ values for both coordinates of the parameter including upper estimates.

The small $\hat{R}$ values imply that the length of the MH random walk was not too short. On the other hand Figure 4.9 shows a plot of the evolution of the $\hat{R}$ value with increasing length of the chain and it suggests that the length of the chain was not unreasonably long.

BAYESIAN APPROACH WITHOUT PRIOR

We have seen that the prior or regulariser influences the estimation and will often lead to worse estimates. However, we have discussed shortly how we can follow a generalised Bayesian ap-

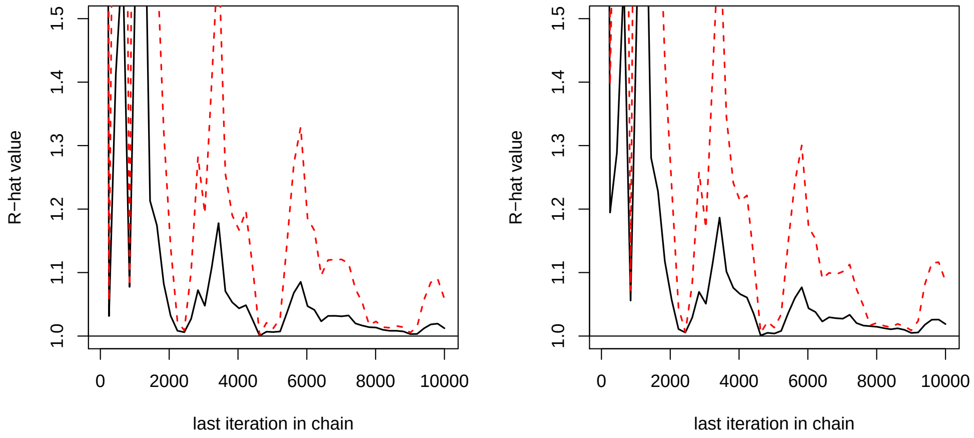


Figure 4.9: Plot of the evolution of the $\hat{R}$ value for first (left) and second (right) coordinate of the parameter in dependency of the length of the Markov chain. The upper estimates for the $\hat{R}$ value are depicted in red.

proach without a prior which results in having the likelihood function as a posterior,

$$f_{\Theta|X}(\theta|x) = f_{X|\Theta}(x|\theta).$$

In order to see that the MCMC approximation still works for this function, we note that

$$\mathrm{d}\pi := f_{\Theta|X} \cdot \mathrm{d}\mu$$

is a σ-finite measure on the parameter space Θ. To approximate this measure, one can use the following more general version of the ergodic theorem (cf. Theorem 17.3.2 [42]).

4.6 Theorem (Ergodic theorem). *Let $(X_n)_{n\in\mathbb{N}}$ be a Markov chain with σ-finite stationary distribution π and let*

$$\hat{\mathbb{P}}_n := \frac{1}{n}\sum_{i=1}^{n}\delta_{X_i}$$

be the empirical measures associated with the Markov chain. Then the following two statements are equivalent:

(*i*) *For all π-integrable functions f, g with $\int g(x)\pi(\mathrm{d}x) \neq 0$ we have*

$$\frac{\int f(x)\hat{\mathbb{P}}_n(\mathrm{d}x)}{\int g(x)\hat{\mathbb{P}}_n(\mathrm{d}x)} \xrightarrow{n\to\infty} \frac{\int f(x)\pi(\mathrm{d}x)}{\int g(x)\pi(\mathrm{d}x)}.$$

(*ii*) *The Markov chain $(X_n)_{n\in\mathbb{N}}$ is Harris recurrent.*

This implies directly that the restrictions of $\hat{\mathbb{P}}_n$ onto sets of finite measure $\pi(A) < \infty$ converge weakly towards π up to normalisation. The argument for the stationarity of the σ-finite measure π stays exactly the same as before. Hence, we can take a completely analogue approach for the approximation of the likelihood function, but we only present the result of the third and final MCMC simulation in Figure 4.10.

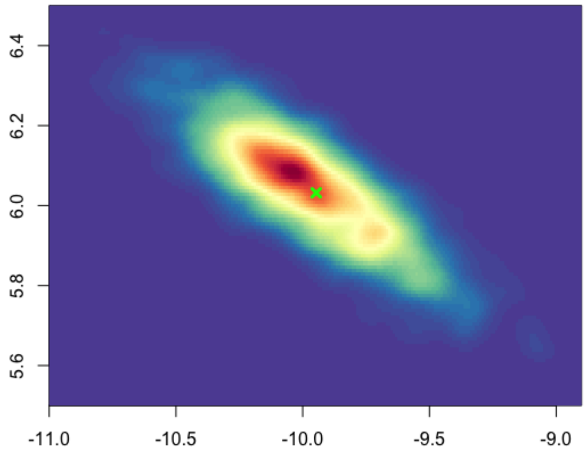


Figure 4.10: Heat map of the MH random walks with 10^4 iterations. The regularised MLE estimator is shown as a white and the MLE as a green cross. The MLE is the maximum of the (approximated) likelihood.

It is evident from both, theoretical considerations and the experimental results of the maximum likelihood estimations and the approximations of the posterior that the regulariser or prior always forces the estimation closer to the origin. Obviously this can make the estimation better, if for example the unregularised MLE is larger than the actual parameter, but then it makes the estimation better by pure luck. We will see later that the influence of the prior can be a little bit more positive if the data is perturbed by random noise.

A naive approximation of the posterior

The motivation for the use of MCMC methods was that one wants to obtain an approximation of the posterior. We present here a different and naive approach, which works a lot faster at least in our toy example. However, we will see later that this approach suffers from what is known as the *curse of dimensionality*,[2] i.e. the time needed to perform it will grow exponentially with the

[2] This name is used for pretty much all phenomena that grow exponentially with the dimension of the problem.

dimension of the parameter that should be estimated.

Let us assume that we have performed the two burn in periods of the MH random walk presented above. Then we roughly know the location of the high density from the first burn in period and also the approximate shape of it from the second one. Now we place a 40 × 40 grid above this box and evaluate the unnormalised posterior at those grid points. Then we use an interpolation algorithm to obtain an approximation of the unnormalised posterior density. This interpolation usually comes in a quite simply form – for example piecewise linear – and can therefore by explicitely expressed and integrated in order to normalise the approximate posterior. The results for this approach can be seen in Figure 4.11 and we will discuss the advantages and hindrances in the next paragraph.

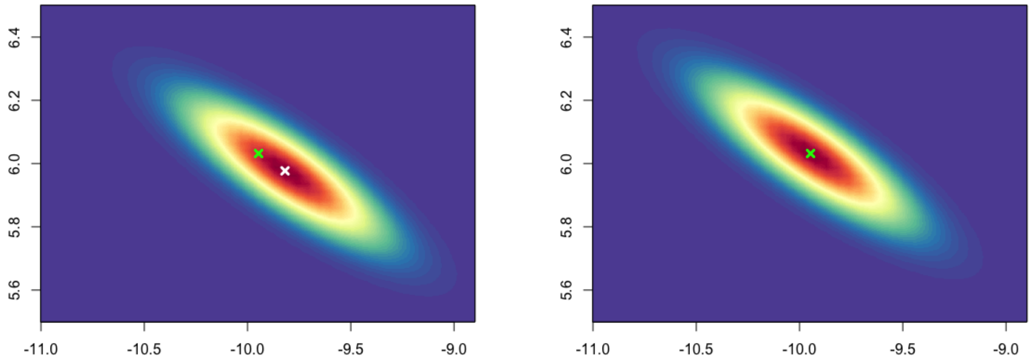


Figure 4.11: Approximations of the posterior (left) and of the likelihood function (right) obtained by the interpolation between breakpoints. Just like in the approximations using MCMC methods, the reguralised MLE is marked white and the MLE green.

Complexity of the different approaches

In large examples the evaluation of the likelihood function

$$f_{\mathbf{Y}^n|\Theta}(Y_1,\dots,Y_n|\theta) \propto \prod_{i=1}^{n} \frac{\det(L(\theta)_{Y_i})}{\det(L(\theta)+I)}$$

involves the computation of a $N \times N$ matrix. This can be done explicitly using Gauss elimination which does not change the determinant – at least up to a sign – and can be performed in at most N^3 steps, cf. [61].[3] Hence, the time needed for the computation of the likelihood function can

[3] Actually one can even do better, but those algorithms come with greater implementation challenges.

be bounded – up to a constant – by

$$N^3 + \sum_{i=1}^{n} |Y_i|^3 \leq (n+1) \cdot N^3$$

and we say it can be performed in $O(N^3)$ time.[4] In practice this will take a significant amount of time and this was also the motivation for the variational MCMC methods. For example in our toy example the computation took roughly 1.5 seconds on a 1.8GHz Intel Core i5 with 8GB RAM[5] using the determinant algorithm in R.

For a single step of the MH random walk the unnormalised posterior needs to be evaluated twice for the computation of the acceptance threshold $\rho(x, y)$, cf. (3.7). Usually one will work with a prior that is easy to compute and with a proposal that can be simulated fast and hence we will neglect its contribution and thus one step of the performance of one MH random walk can be carried out in $O(N^3)$ time. If T denotes the length of the MCMC method, the time needed for its simulation is

$$O\left(T \cdot N^3\right). \tag{4.3}$$

In the final step of the MH random walk, we set $T = 10^4$ as the length and this relates to an approximate simulation time of

$$10^4 \cdot 2 \cdot 1.5\text{s} = 3 \cdot 10^4\text{s} \approx 8\text{h}$$

The strength of the naive approximation based on the interpolation between breakpoints is that one only has to evaluate the unnormalised posterior $40^2 = 1.6 \cdot 10^3 \ll 10^4$ times. The time needed for this is approximately

$$1.6 \cdot 10^3 \cdot 1.5\text{s} = 2.4 \cdot 10^3\text{s} = 40\text{min}.$$

However, if we denote the dimension of the parameter by M, then the size of the grid of breakpoints needed for the interpolation grows exponentially in M. Let R denote the number of grid lines along each coordinate, then one needs to evaluate the posterior R^M times and the times for this behaves like

$$O\left(R^M \cdot N^3\right). \tag{4.4}$$

In 10 dimension and with 40 grid lines along all dimensions, this corresponds to $40^{10} > 10^{16}$ evaluations which can not be performed in reasonable time. This exponential increase makes this direct approach impossible if the parameter one wishes to estimate is not very low dimensional. Nevertheless it might be possible to modify this approach in a suitable way to make it more promising in higher dimension. For example one could try to iteratively raise the resolution of

[4] See also the Landau or ‚big O‘ notation in the nomenclature.

[5] I used a MacBook Air from 2012.

the grid on the places where one expects a high value of the function or high changes of the function. Alternatively it might be worth to investigate how different approximation algorithms of high dimensional functions could help in this approach. Obviously the desirable length T of the Markov chain will increase with the dimension of the parameter space, however this will typically not be the case exponentially in M, cf. [31].

The complexity of the slice sampling algorithm can not be given this easily, at least for the version we presented. This is because the approximate sampling from the uniform distribution on the slice proposed in Algorithm 7 can need arbitrarily many samples from the uniform distribution on the proposed cuboid. Although those uniform samples can be generated efficiently one has to evaluate the unnormalised posterior each time to check whether the proposed sample is actually contained in the slice.

Comments on real world applications

Although this is just a controlled toy example, this procedure can easily be generalised to real world settings. However, one has to face the following two major challenges:

(i) In practice one will not know the feature vectors f_i like we did, so one will have to model those. Usually one would put all quantitative properties into this vector that one would believe could have an effect on the quality of an item. For example if the DPP should model the picnic positions of people in a park one could argue that the quality, i.e. the popularity of a picnic spot depends amongst other things on the distance to the next trash bin, the next toilet and overall noise level. Although one thinks that those parameters can play a role, we could not argue a priori whether they have a positive or a negative impact. For example if the toilets are nice and clean it might be favourable to be closer to them, if they are dirty it might be better to be far away from them in order to avoid their unpleasant odour. However, one does not have to know this straight away as this effect is determined by the according log linearity constant and hence can be estimated in the above manner.

(ii) Secondly and maybe even more importantly the actual similarity kernel is also unknown and hence one also has to come up with a reasonable model for it. Either this can be done by purely relying on models created by people familiar with the real world phenomenon that is being investigated, or one could also try to estimated the similarity kernel itself. However, estimating the whole similarity kernel is equivalent to a maximum likelihood estimation of the whole elementary kernel L and we have seen in the previous discussion about computability that this results in an optimisation problem that can not be solved efficiently. However, the proposed estimation of the repulsiveness parameter might help here, since one only has to model the repuslive structure qualitatively and the estimation of the parameter will take care of the strength of the repulsion.

IV.3 Stability under noise – does the regularisation help?

In real world applications the observed data will almost never be free from noise and outer influences. Therefore one wants to establish stable estimation techniques in the sense that small changes in the data should only lead to small changes in the estimation. This is nothing but the question of continuity of the estimation rule and in a lot of scenarios a regulariser or prior can help to lower the effect that random perturbations of the data have on the estimation, cf. [12]. Thus, we want to investigate whether this is the case for the parameter estimation of discrete DPPs. First we have to specify what noise we are going to consider in the case of discrete DPPs. If one works with continuous DPPs one could assume a perturbation of the exact positions of the observed points, however in the discrete setting this does only make limited sense. Therefore we will work with the observations of a DPP where points are randomly added or deleted and will specify this later.

Before we investigate the stability properties of the estimation in the specific setup for DPPs, we should make some general statements. The estimation can be seen as the following two stage process

$$\text{data } x \xrightarrow{\text{evaluation of } f_{X|\Theta}} \text{posterior } f_{X|\Theta}(x|\cdot)f_\Theta(\cdot) \xrightarrow{\text{maximisation}} \text{MAP estimator } \hat{\theta}.$$

If one wants to investigate the stability properties of the estimation which is nothing but the continuity, then it is reasonable to do this for both steps separately. The second step is in general discontinuous since the maximisation of a function is not a continuous operation under the usual topologies on functions corresponding to uniform or pointwise convergence or integral norms.

Usually the first step will be continuous in some notion, for example if all densities $f_{X|\Theta}(x|\theta)$ are continuous in x, then the posterior depends continuously on the data x in terms of pointswise convergence which corresponds to the product topology. The choice of the prior can possibly strengthen this continuity property and lead to a uniform convergence. If additionally all posterior densities have a unique maximiser, then the maximisation is continuous on this subclass of functions with respect to the uniform topology. In summary we have seen that the prior comes into play at two points, the first one to strengthen the continuity of the first step and then to lead to a possibly more well behaved class of posterior densities.

In the case of discrete DPPs, our space of observations $2^{\mathcal{Y}}$ is discrete and hence the only reasonable topology on it is the discrete topology, i.e. the powerset itself. Every mapping is continuous with respect to this topology and hence the prior is not needed for this qualitative property. Thus, there is no apparent reason why the regularisation or the prior should bring any benefits. We will see in our examples that it can actually be used to regularise certain parameters of the DPP but only if one has a very clear understanding of how those parameters are influenced by the noise.

Experiments

First we explain which kind of noise we will consider.

4.7 Setting. Let $B_1, \dots, B_n$ be independent realisations of a DPP $\mathbb{P}$. Let further $C_1, \dots, C_n$ be independent realisations of an independent Poisson point process. We assume that we have given the data

$$Y_i := B_i \setminus C_i \cap C_i \setminus B_i \quad \text{for } i = 1, \dots, n.$$

The observations Y_i correspond exactly to the observation of a DPP where points were randomly deleted and added.

We will generate noisy data consisting of $n = 8$ samples of a DPP perturbed by a Poisson point processes with marginal kernel $\rho \cdot I$ where we call $\rho \in (0, 1)$ the *intensity* of the point process. We calculate the MLE and regularised MLE corresponding to the regularisation given by the prior (4.2). We use an intensity of $\rho = \frac{1}{400}$ and repeat this procedure eight times and the results of this are fixed in Table IV.2.

	MLE	regularised MLE
1	(-10.05, 7.22)	(-9.76, 7.09)
2	(-9.44, 6.67)	(-9.16, 6.55)
3	(-10.51, 7.05)	(-10.20, 6.91)
4	(-9.79, 6.45)	(-9.49, 6.32)
5	(-11.03, 7.06)	(-10.70, 6.92)
6	(-10.07, 6.97)	(-9.78, 6.85)
7	(-10.04, 6.53)	(-9.74, 6.40)
8	(-9.76, 6.83)	(-9.47, 6.70)

Table IV.2: Table with the MLE and regularised MLE for noisy data of a DPP perturbed by a Poisson point process with intensity $\rho = \frac{1}{400}$.

Like in the case of estimation without noise, the regularised MLE is closer to the origin than the unregularised one. In the first component, this leads to worse estimates and in the second one to better ones. The reason that the estimation of the second parameter benefits from the regularisation is the following. If the cardinality of the DPP is smaller than $N/2$, then the presence of noise – at least the one we are considering – leads to a higher expected cardinality in the data since more points are added than deleted because we expect

$$|Y_i \cap B_i| \leq \left|B_i \setminus Y_i\right|.$$

This leads to an estimation of higher qualities and the magnitude of the qualities is controlled through the second parameter. Hence, the regulariser forces the second component MLE into the right direction. However this kind of regularisation can only be successful, if one has a clear understanding into which direction the noise will perturb the estimates. If for example the cardinality of the DPP is larger than $N/2$ the Poisson noise will lower the cardinality of the data and the regulariser could increase this effect instead of weakening it.

To conclude, we found that a regulariser can be used to lower the effect random perturbations have on the estimation, but only if the qualitative effect of the noise on the certain parameters is understood from theoretical considerations. In this case, however, it might be enough to note this effect or to correct it directly and not through a regulariser.

Chapter V

Summary and conclusion

In this thesis we gave a short but mostly self contained introduction to the basic notions of discrete determinantal point processes. This also included results concerning their existence as well as an explicit construction of them which can be used to simulate them.

Based on those preliminaries we have presented different approaches to the estimation of several parametric models of discrete DPPs. First we presented a point estimator that reconstructs an estimation for the marginal kernel according to the empirical marginal densities. The central tool for this is the solution of the principle minor assignment problem that is concerned with the existence and construction of a matrix with prediscribed principle minors. We showed how this can be reduced to the solution of a set of linear equations over the prime field $\mathbb{F}_2$. One drawback of this approach is, that one has to calculate a minimal shortest cycle basis of the estimated adjacency graph which is not straight forward to implement. Further we have seen that this estimator is consistent, but the results in [60] also imply that the convergence might be very slow.

The second approach was based on the well established theory of maximum likelihood estimation, which yields another point estimator. However the main difficulty here is that the log likelihood function for the whole elementary kernel L is not concave and therefore hard to maximise in practice. Nevertheless, we have seen that this problem can be solved by the use of a log linear model for the qualities. The trade off is that this model has a lower descriptive power and that one has to model the similarity of the DPP which determines the structure and strength of the repulsion. Finally, we proved that the MLEs exist with increasing probability and that they are consistent estimators for the respective parameters which we also generalised to the MAP estimation.

In the third chapter we introduced the Bayesian approach to parameter estimation which is fundamentally different in the sense that it treats the estimated parameter as a random varibale rather than a single value. This does not only allow to capture the uncertainty of the estimation but also has a regularising effect in the sense that the posterior distribution always exists even if the according maximum likelihood estimator doesn't. We have seen that the prior corresponds exactly to the explicit regularisation of the MLE we considered earlier. We have seen how the posterior density of the parameter can be approximated using different MCMC methods even if the MLE is impossible to compute.

In the last chapter we performed the MLE and Bayesian estimation for a toy model introduced earlier. We saw how the sample size affects the quality of the estimation, in particular with different regularisations. In general, it can be noted that a wrong regularisation is especially bad for small sample sizes and hence different priors should be compared, for example using the Bayes factor. Further we used the MH random walk to approximate the posterior distribution and applied two different burn in periods. It is apparent that the tuning of the proposal lead to a higher acceptance rate and a faster decreasing auto correlation function. Finally we investigated the effect random noise has on the estimation. In fact, some parameters get distorted by the presence of Poisson noise, however we argue that it is only reasonable to use a suitable regulariser if one has a clear understanding of the qualitative structure of this influence.

Further work

During the work on this dissertation the following questions arose that might be worth to consider further.

(*i*) Can one effectively perform maximum likelihood estimation of the repulsiveness parameter σ, in the best case even simultaneously to the log linearity constant θ of the quality? If not, could this be done by iteratively optimising σ and θ after another? If one of those procedures works theoretically, does it provide any significant improvement over the other estimations?

(*ii*) What does the geometry of the log likelihood function of the whole elementary kernel look like? Are there other critical points than to the global maximiser?

(*iii*) Are the presented point estimators unbiased?

(*iv*) How do the different point estimators perform compared to each other and can one put those findings onto rigorous base in the sense that one is the optimal estimation for some given observations? Does that performance change under the presence of noise?

(*v*) Investigate whether the ‚naive‘ approach for the approximation of the posterior could somehow be saved in higher dimension. Past work on the approximation of high dimensional functions could help here as well.

(*vi*) Find further applications for the use of DPPs.

To conlude, we want to emphasise that we believe that determinantal point processes will continue to get attention from the research communities concerned with machine learning, data science and computational statistics. We assume that they can help to improve various current techniques in those fields and think that they are already on the way of doing this.

Chapter A

Auxiliary results

The only result we used during the thesis that we consider not general mathematical knowledge is the intersection theorem of Cantor. Since it is hard to find a reference of the general statement we quickly present the proof that can also be found on the Wikipedia entry.

A.1 Theorem (Cantor's intersection theorem). *Let $\mathcal{X}$ be a topological Hausdorff space and let $K_1 \supseteq K_2 \supseteq \dots$ be a sequence of descending, non empty compact sets. Then also the intersection*

$$\bigcap_{n=1}^{\infty} K_n$$

is non empty.

Proof. Assume that the intersection would be empty, and set $U_n := \mathcal{X} \setminus K_n$ which is open since K_n is closed as the compact subset of a Hausdorff space. Then $(U_n)_{n\in\mathbb{N}}$ is an open covering of K_1 since we have

$$\bigcup_{n=1}^{\infty} U_n = \mathcal{X} \setminus \left(\bigcap_{n=1}^{\infty} K_n \right) = \mathcal{X}.$$

Hence we can select a finite subcover and obtain

$$K_1 \subseteq \bigcup_{n=1}^{N} U_n = \mathcal{X} \setminus \left(\bigcap_{n=1}^{N} K_n \right) = \mathcal{X} \setminus K_N.$$

However since $K_n \subseteq K_1$ this implies $K_N = \varnothing$ which is a contradiction. □

Chapter B

Generated code

All the coding was done in R and the code of the sampling algorithm, the general MCMC methods and also the estimation of the log linearity constant will be provided here.

B.1 Sampling algorithm

```
# Implementation of the sampling algorithm as a function of the
# eigendecomposition of the elementary kernel L

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 e_i
    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)
}
```

B.2 Implementation of the MCMC methods

```
# First we implement the Metropolis-Hastings algorithm. We implement the
# propose and reject step. We use a Gaussian as a proposal with covariance
# matrix alpha times the identity.
# Load library for multivariate normal.
library(MASS)
Metropolis <- function(x, f, alpha=1){
  d <- length(x)
  if (length(alpha) == 1) {
    alpha <- diag(rep(alpha, d), d)
  }
  y <- mvrnorm(1, x, alpha)
  z <- f(y)
  if (is.nan(z) || runif(1) * f(x) > z) y <- x
  return(y)
}
```

```
# Now we turn towards slice sampling. Proposing a random interval that includes
# the slice. We use an exponential random variable to define the width of the
# interval.
RandomInterval <- function (x, y, f, alpha=1) {
  c <- f(x)
  # We make the interval the same length in every dimension.
  a <- rexp(1, rate=alpha)  # rexp(length(x), rate=alpha)
  b <- rexp(1, rate=alpha)  # rexp(length(x), rate=alpha)
  # One can check both endpoints simultaneously to avoid the need of two loops.
  while (TRUE) {
    help <- f(x - a)
    if (is.nan(help) || help < c * y) break
    a <- 2 * a
  }
  while (TRUE) {
    help <- f(x + b)
    if (is.nan(help) || help < c * y) break
    b <- 2 * b
  }
  return(matrix(c(x - a, x + b), length(x)))
}
# Doing a single slice sample.
SliceSampling <- function (x, f, alpha=1) {
  d <- length(x)
  a <- f(x)
  y <- runif(1)
  c <- RandomInterval(x, y, f, alpha)
  z <- runif(d, c[, 1], c[, 2])  # runif(1, -4, 4)
  while (TRUE) {
    help <- f(z)
    if (is.nan(help) || help < a * y) z <- runif(d, c[, 1], c[, 2])
    else break
  }
  return(z)
}

# Implementing the MCMC method. The function needs the unnormalised density f,
# a starting value x0, sample size T whether it should be MH or Slice Sampling
# and the parameter alpha, which either specifies the variance of the proposal
# which is multivariate normal or the rate of the exponential random variable
# which defines the thickness of the random interval.
MCMC <- function (f, x0, T=10^3, MH=TRUE, alpha=1) {
  d <- length(x0)
  x <- matrix(rep(x0, T), d)
  if (MH) {
    for (t in 2:T) x[, t] <- Metropolis(x[, t-1], f, alpha)
  }
  else {
    # Check whether starting value is impossible. In this case the slice is the
    # whole space and hence the endpoints of the random interval will diverge.
    while (is.nan(f(x0)) || f(x0)==0) {
      x0 <- mvrnorm(1, x0, diag(rep(alpha, d), d))
    }
    x[, 1] <- x0
    for (t in 2:T) x[, t] <- SliceSampling(x[, t-1], f, alpha)
  }
  return(x)
}
```

B.3 MLE and Bayesian estimation of the log linearity constant

```
# NEEDS: SamplingDPP, defineS, example of DPP on a two dimensional grid with
# log linear qualities including the eigendecompositon of L.

# 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 the data
T <- 20
data <- rep(list(0), T)
for (i in 1:T) {
  data[[i]] <- sort(SamplingDPP(lambda, eigenvectors))
}
# Exporting the data into a .txt file.
write.table(toString(data), "mydata.txt", sep="\n")

# 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)))
}
# Define the sum of the diversity features over a set A.
Feature <- function(A) {
  return(c(sum(DistanceNew(rep(5, length(A)), A, 2, m)), length(A)))
```

```
}
# Define the observation probability and the log likelihood function.
ObservationProbability <- function (theta, data) {
  T <- length(data)
  x <- 1
  a <- det(diag(rep(1, n)) + LFunction(theta))
  for (t in 1:T) {
    A <- data[[t]]
    x <- x * exp(2 * sum(theta * Feature(A))) * det(S[A, A]) / a * 10^45
  }
  return(x)
}
LogLikelihood <- function(theta, data) {
  return(-log(ObservationProbability(theta, data)))
}
RegLogLikelihood <- function(theta, data) {
  return(LogLikelihood(theta, data) + sum(theta^2) / 16)
}
NewRegLogLikelihood <- function(theta, data) {
  return(LogLikelihood(theta, data) + sum(theta^2) / 2)
}

# Maximum likelihood estimation of the log linearity constant theta.
sol <- nlm(LogLikelihood, c(-8, 6))
mle <- sol$estimate
mle

# Learning curve of the MLE
T <- 30
learningCurve <- matrix(rep(0, 2 * T), 2)
regLearningCurve <- matrix(rep(0, 2 * T), 2)
NewRegLearningCurve <- matrix(rep(0, 2 * T), 2)
target <- function(theta) {
  return(LogLikelihood(theta, data))
}
regTarget <- function(theta) {
  return(RegLogLikelihood(theta, data))
}
NewRegTarget <- function(theta) {
  return(NewRegLogLikelihood(theta, data))
}
for (i in 1:T) {
  if (i > 1) {
    help <- data
  }
  data <- rep(list(0), i)
  if (i > 1) {
    for (j in 1:(i - 1)) {
      data[[j]] <- help[[j]]
    }
  }
  data[[i]] <- sort(SamplingDPP(lambda, eigenvectors))
  learningCurve[, i] <- nlm(target, c(-10, 6))$estimate
  print(learningCurve[, i])
  regLearningCurve[, i] <- nlm(regTarget, c(-10, 6))$estimate
  print(regLearningCurve[, i])
  NewRegLearningCurve[, i] <- nlm(NewRegTarget, c(-10, 6))$estimate
  print(NewRegLearningCurve[, i])
}
plot(NewRegLearningCurve[1, 1:30], xlab="Number␣of␣samples", ylab="Estimate
␣␣␣␣␣for␣first␣parameter",
     pch=5, ylim=c(-10.85, -5.5))
points(learningCurve[1, 1:30], pch=16)
points(regLearningCurve[1, 1:30], pch=1)
lines(c(0, 45), c(-10, -10), col="red")
legend("topright", inset=.05, legend=c("MLE", "Weak␣regularisation", "Strong
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣regularisation")
       , pch=c(16, 1, 5))
plot(NewRegLearningCurve[2, 1:30], xlab="Number␣of␣samples", ylab="Estimate␣for
␣␣␣␣␣second␣parameter",
     pch=5, ylim=c(2.8, 7))
points(learningCurve[2, 1:30], pch=16)
points(regLearningCurve[2, 1:30], pch=1)
lines(c(0, 45), c(6, 6), col="red")
legend("bottomright", inset=.05, legend=c("MLE", "Weak␣regularisation", "Strong
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣regularisation"),
       pch=c(16, 1, 5))

# NEEDS: MCMC algorithm.

# We want to introduce the first example of Bayesian parameter estimation for
# DPPs. We start by estimating the log linearity constant of the qualities - a
# parameter for which we've already successfully done MLE.

# Run MCMC and create a plot that also shows the coordinates of the MLE.
# Putting a centered Gaussian as a prior.
target <- function(theta) {
  x <- exp(-sum(theta^2) / 2^4) * ObservationProbability(theta)
  return(x)
}
```

```
# Sampling a 200 samples to find a reasonable starting point. Agressitivity of
# the MH is adjusted so that a reasonable acceptence rate is obtained.
x <- MCMC(target, c(0, 0), MH=TRUE, 10^2, alpha=10)
plot(t(x), pch=16, col='black', cex=0.5, xlab="theta1", ylab="theta2")
points(mle[1], mle[2], pch=4, lwd=3, col="green")
points(mean(x[1, 100:200]), mean(x[2, 100:200]), pch=4, lwd=3, col="red")
# Calculating the acceptence rate for the MH algorithm; 25-75% is desired.
sum(x[, -1] != x[, 1:(length(x)/2 - 1)])/(length(x) - 2)
# Plot the autocorrelation function.
acf(x[1, 1:100], 40)

# Second burn in period consisting of 10^3 samples which will be used to tune
# the proposal. The parameter alpha is adjusted such that a reasonable
# acceptance rate is obtained.
x2 <- MCMC(target, c(mean(x[1, 100:200]), mean(x[2, 100:200])), MH=TRUE, 10^3,
           alpha=2)
plot(t(x2[,250:10^3]), pch=16, col='black', cex=0.5, xlab="theta1",
     ylab="theta2")
points(mle[1], mle[2], pch=4, lwd=3, col="green")
# Calculating the acceptence rate for the MH algorithm; 25-75% is desired.
sum(x2[, -1] != x2[, 1:(length(x2)/2 - 1)])/(length(x2) - 2)
# Plot the autocorrelation function.
acf(x2[1,1:1000], 100, main="Autocorrelation␣for␣alpha=10")

# Doing PCA with the first 100 samples in order to tune the proposal.
library(stats)
pc <- prcomp(t(x2[,250:10^3]))
sd <- t(pc[[2]]) %*% diag(c(pc[[1]][[1]], pc[[1]][[2]])^2) %*% pc[[2]]
# Let the main MCMC run with 10^4 samples. Also a nice plot is created.
xnew <- MCMC(target, c(mean(x[1, ]), mean(x[2, ])), MH=TRUE, 10^4, alpha=sd)
plot(t(xnew))
library(RColorBrewer)
rf <- colorRampPalette(rev(brewer.pal(11,'Spectral')))
r <- rf(64)
k <- kde2d(xnew[1, ], xnew[2, ], n=1000, lims = c(-12, -3, -1, 8))
image(k, col=r, xlim=c(-12, -8.5), ylim=c(5, 7))
points(mle[1], mle[2], pch=4, lwd=3, col="green")
points(-10, 6, pch=4, lwd=3, col="white")
# Calculating the acceptence rate for the MH algorithm; 25-75% is desired.
sum(xnew[, -1] != xnew[, 1:(length(xnew)/2 - 1)])/(length(xnew) - 2)
# Plot the autocorrelation function.
acf(xnew[1,1:1000], 100, main="Autocorrelation␣for␣alpha=10")

# Gelman-Rubin diagnostic:
```

```
# Run a second chain:
newStart <- mvrnorm(1, start, 2 * sd)
xnew2 <- MCMC(target, newStart, MH=TRUE, 10^4, alpha=sd)
install.packages("coda")
library(coda)
combinedchains <- mcmc.list(mcmc(data=t(xnew)), mcmc(data=t(xnew2)))
gelman.diag(combinedchains)
gelman.plot(combinedchains, ylim=c(1, 1.5), ylab="R-hat␣value", lwd=1.5)

# Stability analysis
# Define Poisson point process
poisson <- function(p, n) {
  return((1:n)[runif(n) < p])
}
poisson(0.2, 20)
symmdiff <- function(a, b) {
  return(union(setdiff(a, b), setdiff(b, a)))
}

T <- 8
perturbedData1 <- rep(list(0), T)
sampleSize <- 8
for (i in 1:sampleSize) {
  # Generation of data
  for (i in 1:T) {
    perturbedData1[[i]] <- sort(SamplingDPP(lambda, eigenvectors))
  }
  for (i in 1:T) {
    perturbedData1[[i]] <- symmdiff(perturbedData1[[i]], poisson(1/400, n))
  }
  # Estimators
  target <- function(theta) {
    return(LogLikelihood(theta, perturbedData1))
  }
  print(nlm(target, c(-10, 6.5))$estimate)
  target <- function(theta) {
    return(RegLogLikelihood(theta, perturbedData1))
  }
  print(nlm(target, c(-10, 6.5))$estimate)
  target <- function(theta) {
    return(NewRegLogLikelihood(theta, perturbedData1))
  }
  print(nlm(target, c(-10, 6.5))$estimate)
}
```

Nomenclature

arg max The arg max function selects one arbitrary maximiser given it exists.

$\mathbb{F}_2$ The finite field $\{0, 1\}$ with the addition and multiplication modulo 2.

$\mathbb{N}$ The natural numbers.

$\mathbb{R}$ The real numbers.

$\mathbb{R}^d$ The Euclidean d-dimensional space with Euclidean norm $\|\cdot\|$.

$\mathbb{R}^{N\times N}_{\mathrm{sym},+}$ The set of non negative definite symmetric $N \times N$ matrices.

$\mathbb{R}_+$ The set of non negative real numbers $[0, \infty)$.

$\mathbb{Z}_2$ The cyclic group consisting of $\{0, 1\}$ with the addition modulo 2.

$\mathcal{E}(\alpha)$ The exponential distribution with parameter $\alpha > 0$ given by the density $\mathbb{1}_{[0,\infty)}(s)\alpha e^{-\alpha s}$.

$\mathcal{U}(S)$ The uniform distribution on a set S with respect to some measure that should be clear from the context.

rk The rank of a matrix.

sgn Either the sign of a number or the parity of a permutation.

span This denotes the span of a collection of vectors.

$A \geq B$ We write $A \geq 0$ if A is non negative definite and $A \geq B$ if $A - B \geq 0$ where A and B are symmetric matrices.

$D^2 f$ Hessian matrix, i.e. second derivative of a function $f : \mathbb{R}^d \to \mathbb{R}$.

$L^2(\mu)$ The space of square integrable functions with scalar product

$$(\phi, \psi)_{L^2(\mu)} = \int \phi(x)\psi(x)\mu(\mathrm{d}x).$$

$O(g(x))$ We write $f(x) = O(g(x))$ if $f(x) \leq Mg(x)$ for all $x \geq x_0$ and one $M > 0$.

S_n The permutation group of $\{1, \ldots, n\}$ or other sets with n elements.

$x \leftarrow y$ This denotes the assignment of y to the variable x in pseudocode.

$x \sim \mathbb{P}$ This denotes that x is a realisation of a random variable X with law $\mathbb{P}$.

Bibliography

[AFAT14] Raja Hafiz Affandi, Emily Fox, Ryan Adams, and Ben Taskar. Learning the parameters of determinantal point process kernels. In *International Conference on Machine Learning*, pages 1224–1232, 2014.

[AIR10] Edoardo Amaldi, Claudio Iuliano, and Romeo Rizzi. Efficient deterministic algorithms for finding a minimum cycle basis in undirected graphs. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 397–410. Springer, 2010.

[BA15] Rémi Bardenet and Michalis Titsias RC AUEB. Inference for determinantal point processes without spectral knowledge. In *Advances in Neural Information Processing Systems*, pages 3393–3401, 2015.

[BDF10] Alexei Borodin, Persi Diaconis, and Jason Fulman. On adding a list of numbers (and other one-dependent determinantal processes). *Bulletin of the American Mathematical Society*, 47(4):639–670, 2010.

[BG98] Stephen P Brooks and Andrew Gelman. General methods for monitoring convergence of iterative simulations. *Journal of computational and graphical statistics*, 7(4):434–455, 1998.

[Bil13] Patrick Billingsley. *Convergence of probability measures*. John Wiley & Sons, 2013.

[BK13] Paul Bourgade and Jonathan P Keating. Quantum chaos, random matrix theory, and the Riemann ζ-function. In *Chaos*, pages 125–168. Springer, 2013.

[BM73] Christine Benard and Odile Macchi. Detection and“emission”processes of quantum particles in a“chaotic state”. *Journal of Mathematical Physics*, 14(2):155–167, 1973.

[BM11] A. Bondy and U.S.R. Murty. *Graph Theory*. Graduate Texts in Mathematics. Springer London, 2011.

[Bor09] Alexei Borodin. Determinantal point processes. *arXiv preprint arXiv:0911.1153*, 2009.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[BVDG11] Peter Bühlmann and Sara Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

[DK14] Josip Djolonga and Andreas Krause. From MAP to marginals: Variational inference in bayesian submodular models. In *Advances in Neural Information Processing Systems*, pages 244–252, 2014.

[Dud10] Richard M Dudley. Distances of probability measures and random variables. In *Selected Works of RM Dudley*, pages 28–37. Springer, 2010.

[EP05] Ioannis Z Emiris and Victor Y Pan. Improved algorithms for computing determinants and resultants. *Journal of Complexity*, 21(1):43–71, 2005.

[GKT12] Jennifer Gillenwater, Alex Kulesza, and Ben Taskar. Near-optimal map inference for determinantal point processes. In *Advances in Neural Information Processing Systems*, pages 2735–2743, 2012.

[GR$^+$92] Andrew Gelman, Donald B Rubin, et al. Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472, 1992.

[Gra90] Robert M Gray. Entropy and information theory. Springer, 1990.

[GT06] Kent Griffin and Michael J Tsatsomeros. Principal minors, Part II: The principal minor assignment problem. *Linear Algebra and its applications*, 419(1):125–171, 2006.

[Has70] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[Hig90] Nicholas J Higham. Exploiting fast matrix multiplication within the level 3 BLAS. *ACM Transactions on Mathematical Software (TOMS)*, 16(4):352–368, 1990.

[HKP$^+$06] J Ben Hough, Manjunath Krishnapur, Yuval Peres, Bálint Virág, et al. Determinantal processes and independence. *Probability surveys*, 3:206–229, 2006.

[Hor87] Joseph Douglas Horton. A polynomial-time algorithm to find the shortest cycle basis of a graph. *SIAM Journal on Computing*, 16(2):358–366, 1987.

[Joh04] Kurt Johansson. Determinantal processes with number variance saturation. *Communications in mathematical physics*, 252(1-3):111–148, 2004.

[KR95] Robert E Kass and Adrian E Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

[KT10] Alex Kulesza and Ben Taskar. Structured determinantal point processes. In *Advances in neural information processing systems*, pages 1171–1179, 2010.

[KT11] Alex Kulesza and Ben Taskar. k-DPPs: Fixed-size determinantal point processes. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1193–1200, 2011.

[KT12a] Alex Kulesza and Ben Taskar. Learning determinantal point processes. *arXiv preprint arXiv:1202.3738*, 2012.

[KT$^+$12b] Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

[Kul12] Alex Kulesza. *Learning with Determinantal Point Processes*. PhD thesis, University of Pennsylvania, 2012.

[Kun17] Robert J Kunsch. High-dimensional function approximation: Breaking the curse with monte carlo methods. *arXiv preprint arXiv:1704.08213*, 2017.

[LC06] Erich L Lehmann and George Casella. *Theory of point estimation*. Springer Science & Business Media, 2006.

[LCYO] Donghoon Lee, Geonho Cha, Ming-Hsuan Yang, and Songhwai Oh. Individualness and determinantal point processes for pedestrian detection: Supplementary material.

[LG$^+$16] Jean-François Le Gall et al. *Brownian motion, martingales, and stochastic calculus*, volume 274. Springer, 2016.

[Lyo03] Russell Lyons. Determinantal probability measures. *Publications Mathématiques de l'Institut des Hautes Études Scientifiques*, 98(1):167–212, 2003.

[Mac03] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[ME11] Nathaniel FG Martin and James W England. *Mathematical theory of entropy*, volume 12. Cambridge university press, 2011.

[MG60] Madan Lal Mehta and Michel Gaudin. On the density of eigenvalues of a random matrix. *Nuclear Physics*, 18:420–427, 1960.

[MRR$^+$53] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[MS15] Zelda Mariet and Suvrit Sra. Fixed-point algorithms for learning determinantal point processes. In *International Conference on Machine Learning*, pages 2389–2397, 2015.

[MT02] Antonietta Mira and Luke Tierney. Efficiency and convergence properties of slice samplers. *Scandinavian Journal of Statistics*, 29(1):1–12, 2002.

[MT12] Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

[MZ08] Avner Magen and Anastasios Zouzias. Near optimal dimensionality reductions that preserve volumes. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 523–534. Springer, 2008.

[Nea03] Radford M Neal. Slice sampling. *Annals of statistics*, pages 705–741, 2003.

[NH44] Isaac Newton and Edmund Halley. *Philosophiæ naturalis principia mathematica*, volume 62. Jussu Societatis Regiae ac typis Josephi Streater, prostant venales apud Sam. Smith, 1744.

[NM94] Whitney K Newey and Daniel McFadden. Large sample estimation and hypothesis testing. *Handbook of econometrics*, 4:2111–2245, 1994.

[RC13] Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[RCC10] Christian P Robert, George Casella, and George Casella. *Introducing monte carlo methods with r*, volume 18. Springer, 2010.

[Rez12] Fraydoun Rezakhanlou. Lectures on Random Matrices. *Notes for a UC Berkeley topics course*, 2012.

[RGG$^+$97] Gareth O Roberts, Andrew Gelman, Walter R Gilks, et al. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The annals of applied probability*, 7(1):110–120, 1997.

[Ric06] John Rice. *Mathematical statistics and data analysis*. Nelson Education, 2006.

[RKT15] Justin Rising, Alex Kulesza, and Ben Taskar. An efficient algorithm for the symmetric principal minor assignment problem. *Linear Algebra and its Applications*, 473:126–144, 2015.

[Rob99] Christian P Robert. The Metropolis—Hastings Algorithm. In *Monte Carlo Statistical Methods*, pages 231–283. Springer, 1999.

[Rob12] Christian P Robert. *Discretization and MCMC convergence assessment*, volume 135. Springer Science & Business Media, 2012.

[RR09] Gareth O Roberts and Jeffrey S Rosenthal. Examples of adaptive mcmc. *Journal of Computational and Graphical Statistics*, 18(2):349–367, 2009.

[Rüs14] Ludger Rüschendorf. *Mathematische Statistik*. Springer, 2014.

[Sam59] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.

[Tao10] Terence Tao. A second draft of a non-technical article on universality. https://terrytao.wordpress.com, 2010.

[Tau85] George Tauchen. Diagnostic testing and evaluation of maximum likelihood models. *Journal of Econometrics*, 30(1-2):415–443, 1985.

[UBMR17] John Urschel, Victor-Emmanuel Brunel, Ankur Moitra, and Philippe Rigollet. Learning determinantal point processes with moments and cycles. *arXiv preprint arXiv:1703.00539*, 2017.

[Val79] Leslie G Valiant. The complexity of computing the permanent. *Theoretical computer science*, 8(2):189–201, 1979.

[Vav91] Stephen A Vavasis. *Nonlinear optimization: complexity issues*. Oxford University Press, Inc., 1991.

[Vav95] Stephen A Vavasis. Complexity issues in global optimization: a survey. In *Handbook of global optimization*, pages 27–41. Springer, 1995.

[Veb12] Oswald Veblen. An application of modular equations in analysis situs. *Annals of Mathematics*, 14(1/4):86–94, 1912.

[Vol09] Mikhail V Volkenstein. *Entropy and information*, volume 57. Springer Science & Business Media, 2009.