

Numéro d'ordre: 395

Centrale Lille

THESE

Présentée en vue d'obtenir le grade de

DOCTEUR

En

Spécialité : Automatique, Génie informatique, Traitement du signal et image

Par

GUILLAUME GAUTIER

Doctorat délivré par Centrale Lille

Titre de la thèse

On sampling determinantal point processes

Sur l'échantillonnage des processus ponctuels déterminantaux

Soutenue le 19 mai 2020 devant le jury d'examen:

Président	Pierre-Olivier AMBLARD	Directeur de Recherche CNRS, Université de Grenoble-Alpes
Rapporteurs	Agnès DESOLNEUX	Directrice de Recherche CNRS, ENS Paris-Saclay
	Romain COUILLET	Professeur des universités, CentraleSupélec Paris
Examinateurs	Frédéric LAVANCIER	Maître de conférences, Université de Nantes
	Sheehan OLVER	Reader, Imperial College, London
	Michalis TITSIAS	Research Scientist, DeepMind, London
Directeurs de thèse	Michal VALKO	Chargé de recherche INRIA, Lille - DeepMind, Paris
	Rémi BARDENET	Chargé de Recherche CNRS, Université de Lille

Thèse préparée dans le Laboratoire

Centre de Recherche en Informatique Signal et Automatique de Lille
Université de Lille, Centrale Lille, CNRS, UMR 9189 - CRISTAL
École Doctorale SPI 072



Université
de Lille



Remerciements

Au delà de l'aventure scientifique, la thèse est avant tout une aventure humaine ; autant sur le plan personnel qu'au contact et à la rencontre de l'autre. Les quelques lignes qui suivent ne sont pas suffisantes pour exprimer toute ma gratitude envers les personnes avec lesquelles j'ai eu la chance et le plaisir d'évoluer.

Mes premiers remerciements s'adressent aux membres du jury, Pierre-Olivier Amblard, Frédéric Lavancier, Sheehan Olver et Michalis Titsias et en particulier aux rapporteurs Agnès Desolneux et Romain Couillet d'avoir accepté d'évaluer ce travail de thèse. Merci pour vos éclairages, questionnements, critiques et encouragements qui ont nourri ce travail. Dans le contexte particulier de la pandémie de Covid-19, vous avez répondu présent et avez rendu possible la reprogrammation de ma soutenance à distance, je vous en suis profondément reconnaissant.

Cette thèse n'aurait pu se faire sans mes encadrants Michal Valko et Rémi Bardenet. Merci pour la liberté, l'autonomie et la confiance qui m'ont été accordées, ainsi que pour les remises en questions nécessaires qui ont rythmées ces trois dernières années. Merci de m'avoir encouragé à aller à la rencontre de différentes communautés via des visites, des séminaires ou des conférences. Merci en particulier à Rémi, avec qui j'ai eu la chance d'évoluer au quotidien, merci pour ces moments de qualité, ton enthousiasme, ton écoute, tes conseils et ton goût du travail bien fait.

La recherche c'est aussi une histoire d'équipe ; merci à tous les membres passés et présents des équipes SigMA et SequeL d'avoir su créer une ambiance propice à l'épanouissement scientifique de chacun. Un merci particulier à Pierre qui m'a donné les clés de certaines portes du monde de la recherche, ainsi qu'à Patrick pour sa bienveillance. Un grand merci à Arnaud, Théo, Maxime, Solène, Quentin et Rui pour votre sourire et votre soutien au quotidien. Merci à Clément et à Phuong pour m'avoir montré le chemin en début de thèse. Merci à Lilian, Mathieu, Nicolas, Edouard, Xuedong, Omar, Sadegh, Dorian et Yoan, pour les bons moments passés à l'INRIA et en dehors.

Merci Chi pour m'avoir accompagné dans ma première expérience de recherche au travers de mon mémoire de master. Merci aux différents groupes de travail du laboratoire Painlevé, pour ces séances stimulantes et enrichissantes. Un merci particulier à Adrien et Mylène, pour avoir créé et animé le cours sur les DPPs qui m'a permis de mieux appréhender cet objet complexe en début de thèse.

Merci Guillermo Polito d'avoir engagé ton énergie et ton temps dans le groupe de travail sur la recherche reproductible. Merci pour ta patience et tes précieux conseils qui m'ont permis de développer DPPy. J'en profite pour remercier Daniele, Lilian et Slim d'avoir contribué à

ce projet.

Merci Augustin, Emilie et Pierre de m'avoir fait confiance et de m'avoir accompagné dans l'expérience enrichissante de l'enseignement.

Many thanks to Alex, Jenny and Krzysztof for your valuable time and the interesting discussions we had during my short visit at Google NYC.

Merci à l'ensemble des acteurs mettant à disposition du contenu en libre accès, qu'ils soient enseignants, chercheurs, programmeurs, ou simplement passionnés sans étiquette. Même si vous ne recevez pas forcément la reconnaissance que vous méritez, vos contributions et votre impact sont bien réels.

Merci à mes partenaires sportifs, Christophe, François, Victor, Kévin, Yannis, Romain, Dorian et Nathan, avec qui j'ai pris beaucoup de plaisir sur les terrains de basket, de badminton et de squash. Merci Laurentiu de m'avoir initié à la boxe, d'avoir partagé de super moments sur les chemins de course, ainsi que pour ton énergie positive et tes nombreux conseils.

Merci à mes amis qui, chacun à leur façon et malgré la distance, me font grandir, m'encouragent, me stimulent, voire me fascinent. Merci Océane, Marine, Lucie, Louis et Antoine pour cette belle amitié qui traverse les années et frontières. Merci Thibault, Julien, Virginie, Baptiste, Mathieu pour votre grain de folie. Merci Laurent, Jonas, Reda, Michael et toute la bande lyonnaise pour ces moments intenses vécus en prépa et ceux partagés à chacune de nos retrouvailles. Merci Pierre, Guillaume, Antoine, Yassine, Sylvain, Rita, Anabelle, Alexis, Valentin pour ces belles années centraliennes et celles qui ont suivi. Mention spéciale pour Nabil, merci pour ton amitié indéfectible et pour m'avoir tiré vers le haut en toute circonstance. Merci Julien pour ton amitié précieuse et ces moments particuliers passés en dehors du cadre scientifique, notamment en compagnie de Laura. Thank you Ellen and Nicole for your friendship which goes beyond the Atlantic.

Merci à mes parents et à ma sœur qui m'ont toujours accordé toute leur confiance, leur amour et m'ont toujours encouragé à suivre mon instinct et à me dépasser.

Merci à toute la famille Hultz qui est devenue ma deuxième famille, merci pour votre accueil, votre écoute, et votre amour.

Finalement, un merci passionné à ma douce Jeanne, merci pour ton amour inconditionnel, avec toi l'impossible n'est rien. You're my rock!

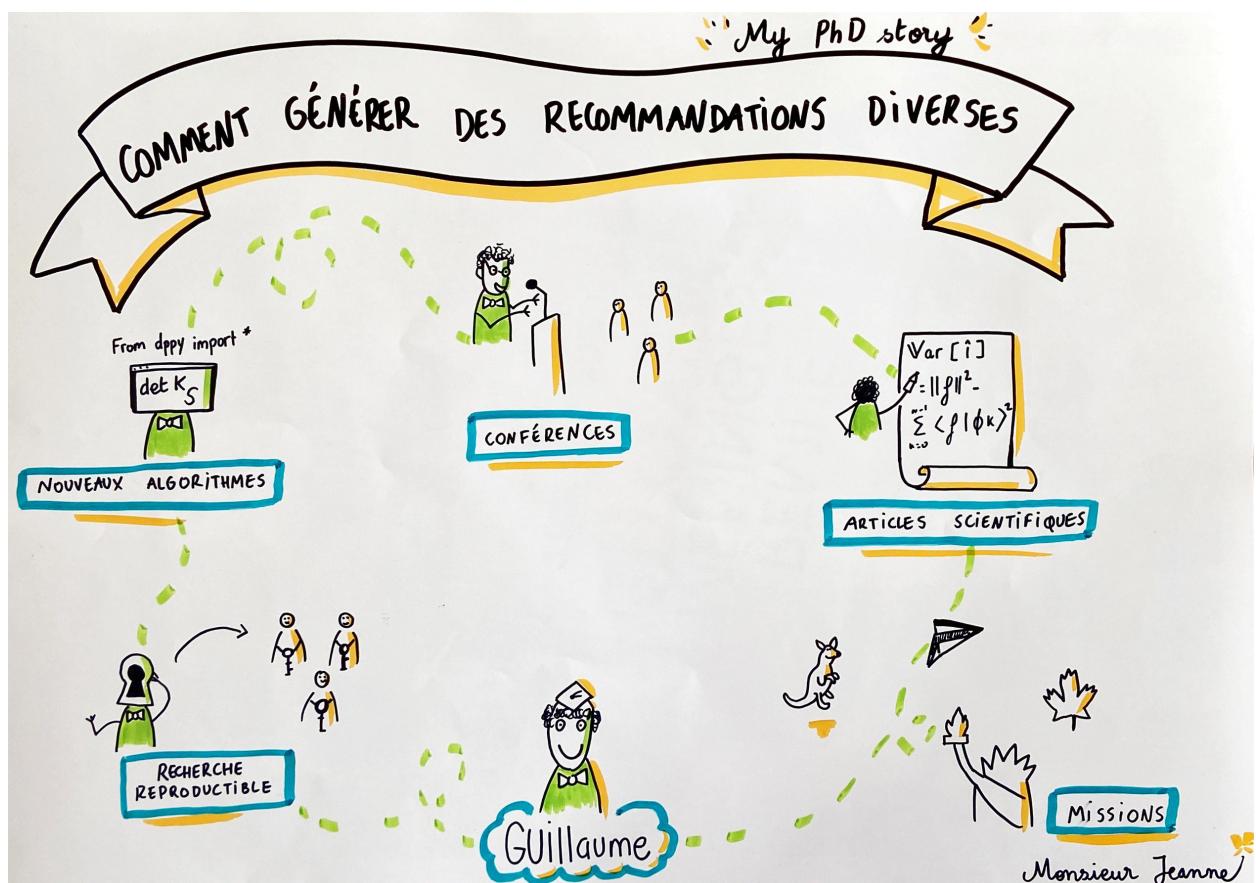


Figure 1: Résumé graphique de ma thèse réalisé par Monsieur Jeanne!

Contents

0 INTRODUCTION	9
General introduction	9
Contributions	14
Outline of the manuscript	16
1 DETERMINANTAL POINT PROCESSES	19
1.1 Definitions	19
1.2 How to construct a DPP?	27
APPENDICES	
1.A Construction of fixed-sized point processes from exchangeable variables	31
1.B Classical matrix results	32
1.C Cauchy-Binet formulas	33
1.D Stability properties of finite DPPs	34
1.E Expectation and variance of linear statistics	39
2 EXACT DPP SAMPLING	41
2.1 Sampling from projection DPPs	42
2.1.1 <i>The continuous case</i>	42
2.1.2 <i>The finite case</i>	45
2.2 Exact sampling from non-projection DPPs	48
2.2.1 <i>Finite DPPs defined by their correlation kernel</i>	48
2.2.2 <i>Finite DPPs defined by their likelihood kernel</i>	53
2.2.3 <i>The continuous case</i>	57
APPENDICES	
2.A Specialization of the sequential sampler for Hermitian DPPs	58
2.B A note on the sequential thinning procedure	58
3 APPROXIMATE DPP SAMPLING	61
3.1 Kernel approximation methods	61
3.2 Monte Carlo Markov chain methods	61
3.3 The zonotope viewpoint on finite projection DPPs	62
3.3.1 <i>Hit-and-run and the Simplex Algorithm</i>	66
3.3.2 <i>From Volume to Squared Volume</i>	68
3.4 Experiments	68
3.4.1 <i>Non-uniform Spanning Trees</i>	68
3.4.2 <i>Text Summarization</i>	71
3.5 Discussion	72

4 APPLICATION OF DPP SAMPLING TO MONTE CARLO INTEGRATION 73

4.1	Standard quadrature	74
4.2	The multivariate Jacobi ensemble	74
4.3	Description of the two DPP-based estimators	75
4.3.1	<i>A natural estimator</i>	75
4.3.2	<i>The Ermakov-Zolotukhin estimator</i>	76
4.4	Sampling from orthogonal projection DPPs	79
4.4.1	<i>Sampling from the multivariate Jacobi ensemble</i>	81
4.5	Empirical investigation	83
4.5.1	<i>The bump experiment</i>	84
4.5.2	<i>Integrating sums of eigenfunctions</i>	84
4.5.3	<i>Further experiments</i>	85
4.6	Discussion	85

APPENDICES

4.A	Further experiments	87
4.A.1	<i>Integrating absolute value</i>	87
4.A.2	<i>Integrating Heaviside</i>	87
4.A.3	<i>Integrating cosine</i>	88
4.A.4	<i>Integrating a mixture of smooth and non smooth functions</i>	88

5 FAST SAMPLING FROM β -ENSEMBLES 89

5.1	Classical β -ensembles and their tridiagonal models	91
5.2	Atomic measures, moments and Jacobi matrices	93
5.2.1	<i>Orthogonal polynomials and Jacobi matrices</i>	93
5.2.2	<i>Orthogonal polynomials and moments</i>	95
5.3	Making the change of variables	97
5.4	Proving the three classical tridiagonal models	100
5.4.1	<i>The HβE and its tridiagonal model</i>	100
5.4.2	<i>The LβE and its tridiagonal model</i>	101
5.4.3	<i>The JβE and its tridiagonal model</i>	102
5.5	Gibbs sampling tridiagonal models associated to polynomial potentials	105
5.5.1	<i>Sampling from the conditionals</i>	105
5.5.2	<i>Example simulations and empirical study of the convergence</i>	106
5.6	Conclusion	112

DISCUSSION 113

RÉSUMÉ EN FRANÇAIS 117

BIBLIOGRAPHY 123

Introduction

0

0.1 GENERAL INTRODUCTION

0.1	General introduction	9
0.2	Contributions	14
0.3	Outline of the manuscript	16

Consider searching for the word “bolt” in an image search engine. Given the vagueness of the query, one may expect from a good engine to be shown pictures of the champion athlete Usain Bolt, a car with the same name, the poster of the Disney’s movie “Bolt” or a kind of fastener; in short a collection of images representing the polysemy of the word “bolt”.



Figure 1: For the query “bolt”, a properly designed DPP would assign higher probability of display to the second row than the first one.

A DETERMINANTAL POINT PROCESS (DPP) IS A PROBABILISTIC MODEL WITH THE RIGHT PROPERTIES TO MAKE RECOMMENDATIONS UNDER DIVERSITY CONSTRAINTS. DPPs assign higher probability to sets of images that are diverse, in a sense to be defined, while guaranteeing that each image is marginally relevant for the query.

THE NOTION OF SIMILARITY BETWEEN IMAGES x AND y IS ENCODED AS THE ENTRY \mathbf{K}_{xy} OF A POSITIVE SEMI-DEFINITE KERNEL MATRIX. The probability that a set of N images labeled $x_1, \dots, x_N \in \mathbb{N}$ is part of the final display expresses as the determinant of the corresponding submatrix. In an informal sense

$$\mathbb{P} \left[\begin{array}{c} \text{images } x_1, \dots, x_N \\ \text{are present in the display} \end{array} \right] = \det \begin{bmatrix} \mathbf{K}_{x_1 x_1} & \cdots & \mathbf{K}_{x_1 x_N} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{x_N x_1} & \cdots & \mathbf{K}_{x_N x_N} \end{bmatrix}. \quad (0.1.1)$$

In particular, the determinant vanishes if two images x_i, x_j are the same, that is, identical images cannot be displayed simultaneously.

THIS DETERMINANTAL STRUCTURE ENFORCES NEGATIVE DEPENDENCE BETWEEN ITEMS. In particular, for two distinct images

x and y , and further assuming that \mathbf{K} is symmetric,

$$\mathbb{P} \left[\begin{array}{l} \text{images } x \text{ and } y \\ \text{are present in the display} \end{array} \right] = \mathbf{K}_{xx} \mathbf{K}_{yy} - \mathbf{K}_{xy}^2. \quad (0.1.2)$$

In other words, the larger the magnitude of the similarity \mathbf{K}_{xy} between images x and y , the less likely they co-occur in the query's result. In fact, DPPs were introduced long before image search engines. Originally, DPPs arose in the context where items are actually points living in a continuous domain.

IN THE CONTINUOUS SETTING, THE NOTION OF DIVERSITY BETWEEN ITEMS IS REPLACED BY A NOTION OF REPULSION BETWEEN POINTS: THE POINTS TEND TO REPEL EACH OTHER, see Figure 2. In the pioneering works of Ginibre (1965), Wigner (1967), and Dyson (1962) describing the energy levels of physical systems as the eigenvalues of large random Hermitian matrices, DPPs actually characterize the very distribution of these eigenvalues. In particular, the study of the eigenvalues of the sample covariance matrix of complex random Gaussian vectors, started by Wishart (1928), found applications in biology (Arnold, Gundlach, and Demetrius, 1994), finance (Laloux et al., 2000), and telecommunications (Couillet and Debbah, 2011).

While working on a mathematical framework to explain an optical phenomenon known as the anti-bunching effect,¹ Macchi (1975) rigorously defined fermion processes – later renamed determinantal points processes – as a model of the position of particles in a beam, accounting for the fact that the probability of detecting two fermions in a short interval of time is expected to be smaller than if the positions were independently distributed, hence the name “anti-bunching”. Fermion processes were characterized by the determinantal form of their coincidence densities, also called correlation functions. These densities are informally defined as

$$\mathbb{P} \left[\begin{array}{l} \text{there is one point in each} \\ \text{infinitesimal volume} \\ dx_1, \dots, dx_N \\ \text{around } x_1, \dots, x_n \end{array} \right] = \det[K(x_i, x_j)]_{i,j=1}^n dx_1, \dots, dx_n, \quad (0.1.3)$$

where the function K , called the correlation kernel, plays the role of \mathbf{K} in our search engine example.²

AS A MATTER OF FACT, DPPS APPEAR IN A WIDE VARIETY OF CONTEXTS, ranging from number theory (Rudnick and Sarnak, 1996), combinatorics (Baik, Deift, and Johansson, 1999; Borodin, Okounkov, and Olshanski, 2000), zeros of random analytic functions (Peres and Virág, 2003; Hough et al., 2009), spatial statistics (Lavancier, Møller, and Rubak, 2015), in connection with random graphs and signal processing (Burton and Pemantle, 2004; Tremblay, Amblard, and Barthelme, 2017; Avena et al., 2018), telecommunication networks (Li et al., 2015), statistical mechanics (Pathria and Beale,

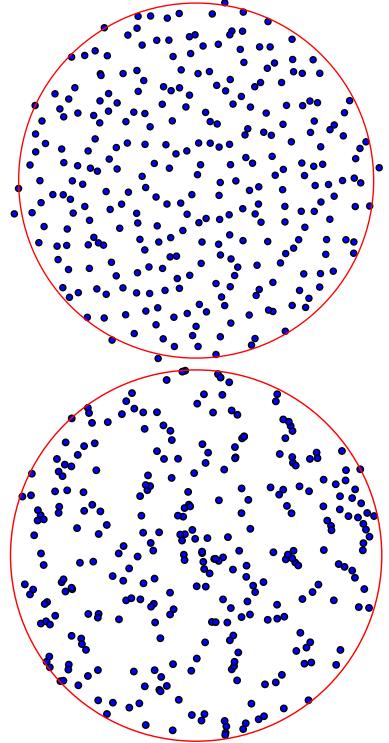


Figure 2: The eigenvalues of a Hermitian Gaussian matrix of size 200 form a DPP called the Ginibre ensemble (above) tend to spread more homogeneously than 200 points drawn independently uniformly at random (below).

¹ The anti-bunching effect of fermions was anticipated by the theory but could not be verified empirically because the poor temporal resolution of the detectors at that time.

² We use the terminology “correlation kernel” as reminder for Korrelation.

2011), machine learning (Kulesza and Taskar, 2012), survey sampling (Loonis and Mary, 2015), or models for neural signals (Snoek, Zemel, and Adams, 2013).

For surveys on DPPs we refer to Johansson (2006) and Anderson, Guionnet, and Zeitouni (2009) for the connection with random matrix theory; to Soshnikov (2000), Shirai and Takahashi (2003), Lyons (2002), Hough et al. (2006), and Borodin (2015) for a probabilistic viewpoint; to Lavancier, Møller, and Rubak (2015) for their use in spatial statistics and to Kulesza and Taskar (2012) for their application to machine learning.

THE TWO NATURAL TASKS THAT ARISE TO UNDERSTAND AND USE DPPS AS A STATISTICAL MODEL OR AS A COMPUTATIONAL TOOL, ARE INFERENCE AND SAMPLING. The main aspect of inference consists in inferring the kernel K of the DPP from data, like learning the notion of similarity between images from a user point of view. Sampling corresponds to generating configurations of points distributed according to a DPP, like a search engine answering a query under diversity constraints.

In the finite case, point processes are a probabilistic model over subsets of a ground set of items. Naive approaches to both inference and sampling face a formidable combinatorial problem. However, the special algebraic structure of DPPs makes them a singular model, and offers polynomial-time inference and sampling algorithms with respect to the natural quantities describing the problem. Such quantities are typically the total number M of items in the database, the expected number of items in the realizations of the process, and potentially the number d of features representing each item.

In the continuous case, there are infinitely many possible configurations of points in an ambient space of dimension d . The usual class of point processes considered to model points in repulsive interaction is the class of Gibbs processes. However, Gibbs processes usually have an intractable normalization constant and require elaborate sampling methods (Møller and Waagepetersen, 2004). In contrast, the determinantal structure of the correlation functions of DPPs reflects on the likelihood of the process, namely

$$\mathbb{P} \left[\begin{array}{l} \text{there are exactly } n \text{ points,} \\ \text{one in each infinitesimal} \\ \text{volume } dx_1, \dots, dx_N \\ \text{around } x_1, \dots, x_n \end{array} \right] \propto \det[L(x_i, x_j)]_{i,j=1}^n dx_1, \dots, dx_n \quad (0.1.4)$$

where L is called the likelihood kernel.³

The normalization constant of (0.1.4) is actually available in closed form, which opens the way to likelihood-based inference of the kernel L . However, the maximization of the log-likelihood is a non convex optimization problem. Bayesian and Expectation-Maximization methods have been employed to learn parametric likelihood kernels (Affandi et al., 2014; Bardenet and Titsias, 2015), while a fixed-point method applies to the non parametric case (Mariet and Sra, 2015) in the dis-

³ We use this terminology as a reminder for Likelihood.

Note that, in the finite setting, DPPs defined through their likelihood kernel are also called L -ensembles (Borodin and Rains, 2004; Kulesza and Taskar, 2012).

crete setting. Besides, Gillenwater et al. (2014), Urschel et al. (2017), and Brunel (2018) also developed moment methods for learning the kernel \mathbf{K} instead, and Gartrell et al. (2019) recently considered learning non-Hermitian likelihood kernels. For a theoretical study of the maximum likelihood estimator in the finite setting, we refer to Brunel et al. (2017a, 2017b).

IN THIS THESIS, WE FOCUS ON THE SAMPLING TASK, for instance DPP samples can be used i) to empirically check the validity of theoretical results in random matrix theory, formulate or test conjectures. Olver, Nadakuditi, and Trogdon (2014) numerically investigate universality phenomena. ii) as a way to generate diverse sets of items for recommendation systems (Gillenwater et al., 2019), text summarization (Dupuy and Bach, 2018) etc. iii) in a Monte Carlo framework to compute estimates of the integral of a function of interest (Bardenet and Hardy, 2020; Belhadji, Bardenet, and Chainais, 2019; Gautier, Bardenet, and Valko, 2019b). iv) to select rows or columns of a design matrix in linear regression or experimental design (Deshpande and Rademacher, 2010; Dereziński et al., 2018; Pukelsheim, 2006; Mariet and Sra, 2017) or for feature selection (Kojima and Komaki, 2016; Belhadji, Bardenet, and Chainais, 2018).

MORE SPECIFICALLY, WE FOCUS ON SAMPLING FROM SO-CALLED PROJECTION DPPs, IN BOTH THE FINITE AND THE CONTINUOUS CASES. The terminology comes from the fact that the underlying kernel characterizes a projection operator. In particular, projection DPPs generate samples with N points almost surely;⁴ this is a convenient property for the display of a fixed number of images or to control the cardinality of the configurations of points to be generated. More generally, projection DPPs⁵ are the building blocks of the DPP model. Indeed, Hough et al. (2006) showed that general DPPs defined by an Hermitian⁶ correlation kernel K , can be expressed as a mixture of projection DPPs. The weights in the mixture are functions of the eigenvalues of the operator with kernel K . From this perspective, the authors derived an exact procedure for sampling generic DPPs (with Hermitian kernels), which requires the eigendecomposition of the kernel. Furthermore, this generic sampling scheme internally calls a projection DPP sampling routine, which does not require preprocessing the projection kernel.

PROJECTION DPPs ADMIT AN EXACT SAMPLING SCHEME, AT LEAST ON PAPER. In the continuous case, the chain-rule based method of Hough et al. (2006) requires generating exact samples from the conditionals. This non-trivial matter is usually tackled using rejection sampling (Lavancier, Møller, and Rubak, 2015), which in turn requires tailored proposal distributions (Gautier, Bardenet, and Valko, 2019b). Putting aside the costs involved in the rejection steps and evaluations of the kernel K , the overall cost of this method for sampling continuous projection DPPs is cubic in the number of points N to be

⁴ Strictly speaking, the projection operator also needs to have finite rank. For instance, the sine kernel ($K(x, y) = \frac{\sin \pi(x-y)}{\pi(x-y)}$) arising in the random matrix literature (Mehta and Gaudin, 1960) generates infinite configurations of points on the real line.

⁵ Projection DPPs are also called elementary DPPs in the machine learning literature (Kulesza and Taskar, 2012).

⁶ K satisfies $K(x, y) = \overline{K(y, x)}$.

generated. In the finite case, if we note M the total number of items, the adaptation⁷ of the chain-rule based method of Hough et al. (2006) yields exact projection DPP samples in $\mathcal{O}(MN^2)$; with an additional preprocessing cost of order $\mathcal{O}(M^3)$ associated to the eigendecomposition of the underlying kernel in the non-projection case.

Some special DPPs arising in random matrix theory can be sampled by computing the eigenvalues of a properly randomized $N \times N$ matrix. From a computational viewpoint, this yields a practical rejection-free exact sampling method with $\mathcal{O}(N^3)$ time-complexity. The equivalent tridiagonal models devised by Dumitriu and Edelman (2002) and Killip and Nenciu (2004) further reduce the sampling cost from cubic to quadratic in N . This has triggered our interest in using such random tridiagonal matrix models for sampling more general DPPs.

Decreusefond, Flint, and Low (2013) also considered adapting the elaborate procedure⁸ of Kendall and Møller (2000) to generate exact DPP samples associated to more general Hermitian kernels. This procedure is shown to have a time-to-coalescence – roughly, the computational cost – which only depends linearly and logarithmically on the trace of the likelihood kernel.⁹ We mention that we have not investigated this line of research, but it might be of interest in the finite case.

Finally, there exist special instances of projection DPPs admitting alternative exact samplers, e.g., the uniform measure on spanning trees of a graph is actually a projection DPP. For well-connected graphs, random-walk-based methods (Propp and Wilson, 1998; Broder, 1989; Aldous, 1990) can produce uniform spanning trees in roughly $\mathcal{O}(M \log(M))$ steps. Like the random tridiagonal matrix models, this kind of examples leaves the possibility of fast DPP sampling open.

WITH TODAY’S DELUGE OF DATA, THE POLYNOMIAL-TIME SAMPLING ALGORITHMS CAN BECOME IMPRACTICAL FOR LARGE SCALE APPLICATIONS, where M and potentially N can be very large: in the order of millions. In such scenarios, the above generic procedure and other matrix-factorization based methods (Launay, Galerne, and Desolneux, 2018; Poulson, 2019) become too costly. Even the linear dependency in M , the total number of items, may be problematic. Gillenwater et al. (2019) exploit a binary tree structure to turn the $\mathcal{O}(MN^2)$ cost of the original projection DPP sampler into $\mathcal{O}(\log(M)N^4)$, which becomes practical when the expected number of points N is small: typically a few tens in recommendation systems.

For generic Hermitian DPPs, the essential bottleneck is the cubic preprocessing cost for the eigendecomposition of the kernel. A possible workaround is to consider a factored form of the likelihood kernel $\mathbf{L} = \Phi^\top \Phi$, where each item is represented by a feature vector of size d , stored as a column of Φ (Kulesza and Taskar, 2012). In this setting, the diagonalization step is performed on the so-called “dual kernel” $\Phi \Phi^\top$ of size $d \times d$, which makes the method practical when $d \ll M$. Dereziński, Calandriello, and Valko (2019) devised an alternative methodology shifting the computational overhead of sampling

⁷ See, e.g., Gillenwater (2014), Tremblay, Bartheleme, and Amblard (2018), and Poulson (2019).

⁸ which is an instance of perfect simulation, based on the method of coupling from the past, see, e.g., Propp and Wilson (1998) and Huber (2016).

⁹ The time-to-coalescence is of order $\mathcal{O}(\int L(x, x) dx \log(\int L(x, x) dx))$.

from a target DPP defined on a potentially large ground set, onto sampling from another DPP defined on a smaller subset of items. To do this, they replace the computation of the spectral content of the kernel, by cheaper-to-compute approximate statistics.¹⁰ From these approximate statistics, they construct an intermediate distribution whose realizations serve as ground set for another tailored DPP. Finally, they prove that downsampling realizations of such intermediate distribution with this tailored DPP actually corrects the bias introduced at the intermediate step; so that the whole procedure yields exact DPP samples.

ANOTHER LINE OF RESEARCH FOCUSES ON APPROXIMATE SAMPLING METHODS. The main methods for sampling approximate from DPPs can be clustered into two categories. In the first class, the approximation is made on the kernel defining the underlying DPP either using random projections or low-rank factorization techniques. The second class relies on Monte Carlo Markov chain (MCMC) methods.

0.2 CONTRIBUTIONS

The original ambition of the thesis was to develop new methods to generate exact DPP samples more efficiently than the original technique of Hough et al. (2006); with a special focus on projection DPPs. The ultimate goal was to make Monte Carlo integration with DPPs (Bardenet and Hardy, 2020) fast and efficient in practice.

IN THIS THESIS, WE TACKLE BOTH FINITE AND CONTINUOUS PROJECTION DPP SAMPLING FROM NON-CONVENTIONAL PERSPECTIVES. In the finite case, we exploit the geometrical structure of projection DPPs to establish the link between sampling and the resolution of randomized linear programs. In particular, we build a novel MCMC sampler that combines ideas from combinatorial geometry, linear programming, and Monte Carlo methods. This method relies on the embedding of the finite support of projection DPPs into a continuous convex domain, and yields a more sample-efficient exploration of the state space than previous MCMC approaches.

In an attempt to generalize this idea to the continuous setting, we investigated the randomization of semi-infinite linear programs.¹¹ The theoretical and technical challenges were too big and we focused on a special instance related to truncated moment problems, which are optimization problems over measures with moment constraints.¹² The fact that some of these optimization problems have solution measures supported on a finite set of points, led us to consider generating DPP samples through a proper randomization of the moments constraints. Restricting to the univariate case, there exists in fact a canonical way of parameterizing measures supported on a compact set. These parameters, derived from the natural moments, are called canonical moments by Dette and Studden (1997). They actually characterize the coefficients of the three term recurrence relation satisfied by the orthogonal

¹⁰ These statistics are an instance of the ridge leverage scores used in kernel approximation methods, see, e.g., Alaoui and Mahoney (2015).

¹¹ See, e.g., Goberna and López (2014).

¹² See, e.g., Lasserre (2010).

polynomials associated to the underlying measure. This recurrence relation can also be encoded by a tridiagonal matrix. In the end, leveraging all these connections, we use elementary arguments to unify the treatment of the random tridiagonal matrix models for the three classical β -ensembles, which were respectively proved using different techniques by Dumitriu and Edelman (2002) and Killip and Nenciu (2004). Then, we conduct an empirical study of a promising fast mixing Monte Carlo Markov chain on tridiagonal matrices to simulate more general β -ensembles.

In the context of Monte Carlo integration, we implemented an efficient version of the original sampler of Hough et al. (2006) tailored to the projection DPP used by Bardenet and Hardy (2020), see Figure 3. This allowed us to empirically test the properties and explore the behavior of two DPP-based Monte Carlo estimators, in regimes yet unexplored. The first one due to Bardenet and Hardy (2020) works as a random multivariate equivalent of the Gauss quadrature. The second was actually devised by Ermakov and Zolotukhin (1960), but remained mostly unknown in the literature. Thus, establishing its intrinsic link with projection DPPs sheds a new light on this estimator based on the resolution of a randomized linear system involving the integrand and the eigenfunctions of the DPP kernel.

FINALLY, OUR DPPY⁶ TOOLBOX IS A NEW COMPUTATIONAL ENTRY-POINT TO THE DPP MODEL. For better reproducibility, we created the DPPy Python toolbox (Gautier et al., 2019), which comes with an extensive documentation⁷ explaining and illustrating the various properties of DPPs and the corresponding state-of-the-art sampling methods. We mention that, apart from our own works, several researchers already make use of this toolbox, like Kammoun (2018), Burt, Rasmussen, and Wilk (2019), and Dereziński (2019). Figure 4 shows the current (01/30/2020) activity status of the hosting repository.

BELOW IS A LIST OF OUR CONTRIBUTIONS.

Journal paper (almost all the figures of the thesis were generated with DPPy)

- G. Gautier, G. Polito, R. Bardenet, and M. Valko. 2019. *DPPy: DPP Sampling with Python*. Journal of Machine Learning Research - Machine Learning Open Source Software (JMLR-MLOSS). arXiv:1809.07258.

Submitted to a journal (cf. Chapter 5)

- G. Gautier, R. Bardenet, and M. Valko. 2020. *Fast sampling from β -ensembles*. ArXiv e-prints. arXiv:2003.02344.

Conference papers (cf. Chapters 3 and 4)

- G. Gautier, R. Bardenet, and M. Valko. 2017. *Zonotope hit-and-run for efficient sampling from projection DPPs*. In International Conference on Machine Learning (ICML). arXiv:1705.10498.
- G. Gautier, R. Bardenet, and M. Valko. 2019b. *On two ways to use determinantal point processes for Monte Carlo integration*. In Advances in Neural Information Processing Systems (NeurIPS).

Workshop papers

- G. Gautier, R. Bardenet, and M. Valko. 2019c. *On two ways to use determinantal point processes for Monte Carlo integration*. In Workshop on Negative Dependence in Machine Learning, International Conference on Machine Learning (ICML).

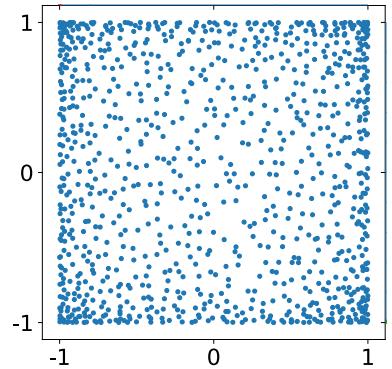


Figure 3: A sample with $N = 1000$ points from the two-dimensional projection DPP used for Monte Carlo integration.

DPPy: DPP Sampling with Python

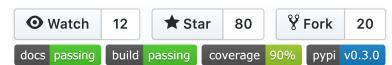


Figure 4: Links to DPPy

- github.com/guilgautier/DPPy
- dppy.readthedocs.io.

 G. Gautier, R. Bardenet, and M. Valko. 2019a. *Les processus ponctuels déterminantaux en apprentissage automatique*. In French Colloquium on Signal and Image Processing (GRETSI).

0.3 OUTLINE OF THE MANUSCRIPT

In the same spirit as the documentation of DPPy, we strive to give intuitions and clear explanations of the definitions and mathematical properties of DPPs. At the end of each chapter, appendices collect complementary remarks, related results and proofs. We exploit Tufte’s book style (Tufte, 2006) and use its wide margins to accompany the main body of the text with comments, or display figures (mostly) generated with DPPy.

THE MANUSCRIPT IS DIVIDED INTO FIVE CHAPTERS.

CHAPTER 1 lays the ground material for the subsequent chapters. We introduce the main definitions and properties of DPPs in both the continuous and finite settings, and explicit the informal descriptions given in the introduction.

CHAPTER 2 discusses several methods available to generate exact DPP samples in both the finite and continuous settings. In particular, special sections are dedicated to projection DPPs to emphasize the special role they play in the construction of the DPP model. We also insist on the geometrical interpretation of the associated sampling procedures.

CHAPTER 3 discusses various methods to generate approximate samples in both finite and continuous cases. The last section includes material from an international conference,¹³ where we develop a Monte Carlo Markov Chain sampler for the simulation of finite projection DPPs. In this work, we combine the geometrical properties of the DPP model with linear programming to view DPP samples as the solution of a randomized linear problem.

CHAPTER 4 includes material accepted to an international conference,¹⁴ where we consider the problem of estimating the integral of a function f with a DPP-based Monte Carlo method. Our motivation comes from the recent result of Bardenet and Hardy (2020), who showed that the samples of a specific projection DPP can be used to construct unbiased estimates of the integral of interest with a variance that decays faster than classical Monte Carlo. Retrospectively, the first DPP-based Monte Carlo estimator was devised by Ermakov and Zolotukhin (1960), some fifteen years before Macchi (1975) even formalized DPPs. In this contribution we first reveal the link between this estimator involving the resolution of a random linear system and projection DPPs, and analyze its properties using modern DPP machinery. In particular, to get the best estimation guarantee with a fixed budget of points, the definition of the estimator suggests a spe-

¹³ G. Gautier, R. Bardenet, and M. Valko. 2017. *Zonotope hit-and-run for efficient sampling from projection DPPs*. In International Conference on Machine Learning (ICML). arXiv:1705.10498.

¹⁴ G. Gautier, R. Bardenet, and M. Valko. 2019b. *On two ways to use determinantal point processes for Monte Carlo integration*. In Advances in Neural Information Processing Systems (NeurIPS).

cific but very natural choice for the projection kernel. Assuming the function f to integrate has fast decaying coefficients in a given orthogonal basis, it is strongly suggested to take as eigenfunctions for the DPP kernel the orthogonal basis functions onto which f has the largest coefficients. Then we investigate the empirical behavior of both the estimator of Bardenet and Hardy (2020) and that of Ermakov and Zolotukhin (1960) in various regimes, considering the so-called multivariate Jacobi-ensemble as projection DPP. To do this, we implement a tailored efficient version of the exact sampling procedure originally derived by Hough et al. (2006). In this respect, the exact sampling scheme for orthogonal projection DPPs is presented in this chapter.

CHAPTER 5 includes material submitted to an international journal.¹⁵ We consider a class of repulsive point processes on the real line called β -ensembles, which contains projection DPP instances when $\beta = 2$.

Our main motivation comes from the fact that β -ensembles appear as the eigenvalues of random tridiagonal matrices. From a sampling perspective, computing the eigenvalues of a properly randomized tridiagonal matrix is a way to generate exact samples in $\mathcal{O}(N^2)$ time complexity.

In this chapter, we give an unifying and elementary treatment of the tridiagonal models corresponding to the classical Hermite, Laguerre and Jacobi β -ensembles. In these special cases, the coefficients defining the associated tridiagonal matrix are independent with easy-to-sample distributions. When targeting more general β -ensembles, the independence vanishes, but the coefficients interact only within a short range. We exploit this property and derive a Gibbs sampling strategy to sample from β -ensembles with polynomial potentials.

We provide a tailored implementation, which allows us to test the properties of the method, in regimes yet unexplored by the literature. Our experiments reveal surprisingly fast convergence of the Gibbs sampler. In particular, within ten Gibbs passes only, even for a large tridiagonal matrices, the marginal behavior of the eigenvalues fits very well the theoretical expectations.

THE FINAL SECTION contains a discussion on the different parts and contributions presented in the manuscript, along with potential lines of improvements and open questions regarding DPP sampling.

¹⁵ G. Gautier, R. Bardenet, and M. Valko. 2020. *Fast sampling from β -ensembles*. ArXiv e-prints. arXiv:2003.02344.

Determinantal point processes

1

A point process can be viewed as a random collection of points living in an arbitrary domain. For example, the points may identify the position of trees in a forest, or particles in a physical system but also correspond to the items listed in a catalog or a database. Thus, point processes may be considered as probabilistic models accounting for the various types of interactions that may exist between these points.

Throughout the manuscript we only consider the case where the points live in a continuous or finite space $\mathbb{X} = \mathbb{R}, (0, +\infty), [0, 1], [-1, 1]^d$ or $\{1, \dots, M\}$. When the ambient space is continuous, realizations of the point process materialize as a cloud of points $\{x_1, \dots, x_N\} \subset \mathbb{X}$, where the number of points N may vary from one realization to another. When the ambient space is finite, e.g., $\mathbb{X} = \{1, \dots, M\}$, realizations of the point process can be understood as a bag of items extracted from a data base.

Among point processes, Determinantal Point Processes (DPPs) are a parametric family of point processes parametrized by a kernel function K . Their essential characteristic is that the correlation between the points is encoded under the special algebraic form of determinants.

The goal of this chapter is first to formalize the intuitions given in the introduction. Then, we give the main properties induced by this singular determinantal structure and discuss various ways of constructing a valid kernel.

For more details and illustrations we invite the reader to have a look at the documentation of DPPy. 

1.1	Definitions	19
1.2	How to construct a DPP?	27
APPENDICES		
1.A	Construction of fixed-sized point processes from exchangeable variables	31
1.B	Classical matrix results	32
1.C	Cauchy-Binet formulas	33
1.D	Stability properties of finite DPPs	34
1.E	Expectation and variance of linear statistics	39

 dppy.readthedocs.io.

1.1 DEFINITIONS

In this section we give a brief account of the theory of point processes. For more generality and technical details regarding the definition of point processes we refer the reader to, e.g., Daley and Vere-Jones (2003) and Møller and Waagepetersen (2004).

We note \mathbb{X} the ambient space where the points live and restrict to *simple* point processes for which each point can appear only once. Thus a simple point process can be viewed as a random subset $\mathcal{X} \subset \mathbb{X}$, i.e., a random point pattern on \mathbb{X} . Moreover to avoid any accumulation of points, we consider *locally finite* point processes, i.e., for any bounded set $B \in \mathcal{B}(\mathbb{X})$, the number of points of \mathcal{X} falling in B is finite.

More formally, consider the measure space $(\mathbb{X}, \mathcal{B}(\mathbb{X}), \mu)$ where $\mathcal{B}(\mathbb{X})$ is the Borel σ -algebra associated to \mathbb{X} and μ is a reference measure. A point process \mathcal{X} on $(\mathbb{X}, \mathcal{B}(\mathbb{X}), \mu)$ is a measurable mapping from a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ onto the measurable space $(N_{lf}, \mathcal{N}_{lf})$ of

locally finite configurations of points of \mathbb{X} where

$$N_{lf} \triangleq \{S \subset \mathbb{X} \mid |S \cap B| < \infty \text{ for all bounded } B \in \mathcal{B}(\mathbb{X})\}. \quad (1.1.1)$$

and \mathcal{N}_{lf} is the smallest σ -algebra on N_{lf} making the mappings

$$\begin{aligned} N_{lf} &\rightarrow \mathbb{N} \\ S &\mapsto |S \cap B| \end{aligned}$$

measurable for all bounded $B \in \mathcal{B}(\mathbb{X})$.

There exist several ways to define point processes, e.g., through their void probabilities, their Laplace transform, their correlation functions and Janossy densities (Daley and Vere-Jones, 2003; Møller and Waagepetersen, 2004; Shirai and Takahashi, 2003).

In this manuscript, we characterize point processes and define determinantal point processes (DPPs) through their so-called correlation functions.¹ Intuitively, the n -th correlation function of \mathcal{X} describes the inclusion probabilities of n points in the process

$$\mathbb{P}\left[\begin{array}{l} n \text{ points of the process are} \\ \text{located in the infinitesimal balls} \\ B(x_1, dx_1), \dots, B(x_n, dx_n) \end{array}\right] = \rho_n(x_1, \dots, x_n) \mu(dx_1) \cdots \mu(dx_n). \quad (1.1.2)$$

More formally, the correlation functions of a point process can be defined in the following way.

Definition 1.1.1. (*Correlation functions*) Let \mathcal{X} be a (finite) point process defined on \mathbb{X} with reference measure μ . The correlation functions of \mathcal{X} , noted ρ_n for $n \geq 1$, are symmetric nonnegative and locally integrable functions such that for any measurable function $f : \mathbb{X}^n \rightarrow [0, \infty)$

$$\mathbb{E}\left[\sum_{\substack{(\mathbf{x}_1, \dots, \mathbf{x}_n) \\ \mathbf{x}_1 \neq \dots \neq \mathbf{x}_n \in \mathcal{X}}} f(\mathbf{x}_1, \dots, \mathbf{x}_n)\right] = \int_{\mathbb{X}^n} f(x_1, \dots, x_n) \rho_n(x_1, \dots, x_n) \mu(dx_1) \cdots \mu(dx_n), \quad (1.1.3)$$

with the condition that $\rho_n(x_1, \dots, x_n) = 0$, if $x_i = x_j$ for some $i \neq j$.

Another characterization could be given in terms of the so-called Janossy densities, which characterize the n -points likelihoods of the process. Again, in an informal way

$$\mathbb{P}\left[\begin{array}{l} \text{the process has exactly } n \text{ points} \\ \text{one in each } B(x_1, dx_1), \dots, B(x_n, dx_n) \end{array}\right] = j_n(x_1, \dots, x_n) \mu(dx_1) \cdots \mu(dx_n) \quad (1.1.4)$$

Paraphrasing Daley and Vere-Jones (2003, Section 5.4), from an experimental point of view, the correlation functions (1.1.2) can be estimated from the results of n observations at specific times or places, whereas the Janossy densities (1.1.4) require indefinitely many observations to determine the exact (total) number of occurrences. For this reason, the correlation functions (1.1.2) are in principle amenable to experimental determination (through ‘coincidence’ experiments, as called by Macchi (1975)) in a way that Janossy densities are not, at

¹ also called product densities or joint intensities.

least in the context of counting particles. However, the inclusion-exclusion principle² allows us to link the correlation functions ρ_n and the Janossy densities j_n of a point process.

Lemma 1.1.2. (*Daley and Vere-Jones, 2003, Lemma 5.4.III*) Consider a finite point process $\mathcal{X} \subset \mathbb{X}$ with reference measure μ , such that the correlation functions $(\rho_n)_{n \geq 1}$ and Janossy densities $(j_n)_{n \geq 1}$ exist. Then, for any $k \geq 1$ and $x_1, \dots, x_k \in \mathbb{X}$ we have

$$\begin{aligned}\rho_k(x_1, \dots, x_k) &= \sum_{n=0}^{\infty} \frac{1}{n!} \int_{\mathbb{X}^n} j_{k+n}(x_1, \dots, x_k, y_1, \dots, y_n) \mu(dy_1) \cdots \mu(dy_n), \\ j_k(x_1, \dots, x_k) &= \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \int_{\mathbb{X}^n} \rho_{k+n}(x_1, \dots, x_k, y_1, \dots, y_n) \mu(dy_1) \cdots \mu(dy_n).\end{aligned}$$

When \mathbb{X} is finite, for any $A \subset \mathbb{X}$, this reads

$$\begin{aligned}\mathbb{P}[A \subset \mathcal{X}] &= \sum_{A \subset S \subset \mathbb{X}} \mathbb{P}[\mathcal{X} = S], \\ \mathbb{P}[\mathcal{X} = A] &= (-1)^{-|A|} \sum_{A \subset S \subset \mathbb{X}} (-1)^{|S|} \mathbb{P}[S \subset \mathcal{X}].\end{aligned}$$

By Definition 1.1.1, the correlation functions are useful to express moments of linear statistics of the point process.³ For example, the first correlation function ρ_1 , also called the intensity function of the process, allows us to compute the expectation of the number of points that fall in different regions of the ambient space. Indeed, for any measurable set $B \subset \mathbb{X}$, $\mathbb{E}[|\mathcal{X} \cap B|] = \int_B \rho_1(x) \mu(dx)$. More generally, for any family of mutually disjoint measurable subsets $B_1, \dots, B_n \subset \mathbb{X}$, if we take $f = \prod_{i=1}^n \mathbf{1}_{B_i}$ then (1.1.3) becomes

$$\mathbb{E}\left[\prod_{i=1}^n |\mathcal{X} \cap B_i|\right] = \int_{B_1} \cdots \int_{B_n} \rho_n(x_1, \dots, x_n) \mu(dx_1) \cdots \mu(dx_n), \quad (1.1.5)$$

and for any $n_1, \dots, n_k \geq 1$ such that $\sum_{i=1}^k n_i = n$, we obtain

$$\mathbb{E}\left[\prod_{i=1}^k \binom{|\mathcal{X} \cap B_i|}{n_i} n_i!\right] = \int_{B_1^{n_1}} \cdots \int_{B_k^{n_k}} \rho_n(x_1, \dots, x_n) \mu(dx_1) \cdots \mu(dx_n). \quad (1.1.6)$$

Now, the ‘‘determinantal’’ term in Determinantal Point Process refers to the fact that correlation functions express as the determinant of a matrix with entry ij given by the evaluation of a kernel function K .

Definition 1.1.3 (Determinantal Point Process). Let $(\mathbb{X}, \mathcal{F}, \mu)$ be a measurable space and consider a measurable function $K : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{C}$. A point process \mathcal{X} on \mathbb{X} is said to be determinantal with reference measure μ and kernel K , when for any $n \geq 1$ and $x_1, \dots, x_n \in \mathbb{X}$, the correlation functions of \mathcal{X} take the form⁴

$$\rho_n(x_1, \dots, x_n) = \det[K(x_i, x_j)]_{i,j=1}^n, \quad (1.1.7)$$

in which case we note $\mathcal{X} \sim \text{DPP}(\mu, K)$.

Furthermore, when the kernel is Hermitian, i.e.,

$$\overline{K(y, x)} = K(x, y), \quad \forall x, y \in \mathbb{X}, \quad (1.1.8)$$

² See also Theorem 1.D.1 for a different application of the inclusion-exclusion principle when \mathbb{X} is finite.

³ See also Proposition 1.E.1 where we compute the expectation and the variance of linear statistics of DPPs.

⁴ A necessary condition on K for the existence is that $\det[K(x_i, x_j)]_{i,j=1}^n \geq 0, \forall n \geq 1$ and $x_1, \dots, x_n \in \mathbb{X}$.

the process \mathcal{X} may be called a Hermitian DPP and we use symmetric instead of Hermitian when $K(x, y) \in \mathbb{R}$.

The finite counterpart of Definition 1.1.3 takes the following form.

Definition 1.1.4. Let \mathbb{X} be finite with size $|\mathbb{X}| = M$. Consider a vector⁵ $\omega \triangleq (\omega_1, \dots, \omega_M) \in [0, +\infty)^M$ and a matrix $\mathbf{K} \in \mathbb{C}^{M \times M}$. The point process \mathcal{X} on \mathbb{X} is said to be determinantal with weight vector ω and correlation⁶ kernel \mathbf{K} , when for any subset $A \subset \mathbb{X}$,

$$\mathbb{P}[A \subset \mathcal{X}] = \det \mathbf{K}_A \prod_{i \in A} \omega_i, \quad \text{where } \mathbf{K}_A \triangleq [\mathbf{K}_{ij}]_{i,j \in A}, \quad (1.1.9)$$

in which case we note $\mathcal{X} \sim \text{DPP}(\omega, \mathbf{K})$. When $\omega_m = 1, \forall m \in \mathbb{X}$, we simply note $\mathcal{X} \sim \text{DPP}(\mathbf{K})$. By convention $\det \mathbf{K}_\emptyset = 1$. Furthermore, if $\mathbf{K} \in \mathbb{C}^{M \times M}$ is Hermitian,⁷ i.e., $\mathbf{K}^H = \mathbf{K}$, the process \mathcal{X} is called a Hermitian DPP and we use symmetric instead of Hermitian when $\mathbf{K}^T = \mathbf{K}$ is real-valued.

An alternative way of introducing DPPs, is to define them through their likelihood function, i.e., their Janossy densities.⁸ To avoid the technicalities of the continuous case regarding the definition of the normalization constant as a Fredholm determinant, we only consider the finite case.

Definition 1.1.5 (L-ensembles). Let \mathbb{X} be finite and consider a matrix $\mathbf{L} \in \mathbb{C}^{M \times M}$. The point process $\mathcal{X} \subset \mathbb{X}$ with joint distribution⁹

$$\mathbb{P}[\mathcal{X} = S] = \frac{\det \mathbf{L}_S}{\det[I + \mathbf{L}]}, \quad \forall S \subset \mathbb{X}, \quad (1.1.10)$$

is called an L-ensemble with likelihood kernel \mathbf{L} .

In fact, there is a correspondence between the determinantal structure of the likelihood (1.1.10) of L-ensembles and the determinantal structure of the inclusion probabilities (1.1.9) of DPPs.

Proposition 1.1.6 (L-ensembles are DPPs). Consider the point process \mathcal{X} to be an L-ensemble with kernel $\mathbf{L} \in \mathbb{C}^{|\mathbb{X}| \times |\mathbb{X}|}$, as in Definition 1.1.5. Then $\mathcal{X} \sim \text{DPP}(\mathbf{K} = \mathbf{L}(I + \mathbf{L})^{-1})$.

We note $\mathcal{X} \sim \text{DPP}(\mathbf{L})$ to emphasize that it is defined via its joint distribution and call \mathbf{L} the likelihood¹⁰ kernel.

Proof. Let us compute the inclusion probabilities of this point process. For any $A \subset \mathbb{X}$,

$$\begin{aligned} \mathbb{P}[A \subset \mathcal{X}] &= \sum_{A \subset S \subset \mathbb{X}} \mathbb{P}[\mathcal{X} = S] \\ &= \frac{1}{\det[I + \mathbf{L}]} \sum_{A \subset S \subset \mathbb{X}} \det \mathbf{L}_S \\ &= \frac{1}{\det[I + \mathbf{L}]} \det \left[I^A \mathbf{L} + I^{A^c} (I + \mathbf{L}) \right] \\ &= \det \left[I^A \mathbf{L}(I + \mathbf{L})^{-1} + I^{A^c} \right] \\ &= \det \left[\mathbf{L}(I + \mathbf{L})^{-1} \right]_A. \end{aligned}$$

⁵ ω plays the role of the reference measure $\mu = \sum_{m=1}^M \omega_m \delta_m$. The entry ω_m can be interpreted as the marginal relevance or quality of item m .

⁶ It is also called *marginal* kernel in the literature, but we prefer the term *correlation* as a reminder for Korrelation.

⁷ the symbol H means conjugate transpose.

⁸ See their informal definition (1.1.4).

⁹ Assuming $\det \mathbf{L}_S \geq 0, \forall S \subset \mathbb{X}$.

The normalizing constant of (1.1.10) is given by Theorem 1.B.2 $\sum_{S \subset \mathbb{X}} \det \mathbf{L}_S = \det[I + \mathbf{L}]$.

See also Kulesza and Taskar (2012, Theorem 2.2) or Borodin, Okounkov, and Olshanski (2000, Proposition A.6).

¹⁰ The terminology serves as reminder for Likelihood.

By Theorem 1.B.2, where $[I^A]_{ij} = \mathbb{1}_{i=j \in A}$.

□

Proposition 1.1.6 shows that an L -ensemble with likelihood kernel \mathbf{L} can be realized as a DPP with correlation kernel $\mathbf{K} = \mathbf{L}(I + \mathbf{L})^{-1}$. Conversely, the computation of the likelihood of $\mathcal{X} \sim \text{DPP}(\mathbf{K})$ reveals that this relation can be inverted, that is $\text{DPP}(\mathbf{K})$ can be realized as an L -ensemble with likelihood kernel $\mathbf{L} = \mathbf{K}(I - \mathbf{K})^{-1}$, if and only if $\mathbb{P}[\mathcal{X} = \emptyset] > 0$.¹¹ In particular, if \mathbf{K} is assumed to be Hermitian, this inversion fails when \mathbf{K} has some eigenvalues equal to one.

¹¹ See Corollary 1.D.3.

DO WE NEED COMPLEX KERNELS? Let us take a simple toy example; let $\mathbb{X} = \{1, 2, 3\}$ and consider the point process \mathcal{X} with likelihood

$$\mathbb{P}[\mathcal{X} = S] = \begin{cases} 0, & \text{if } S = \emptyset \text{ or } \mathbb{X}, \\ \frac{1}{6}, & \text{if } |S| = 1, 2. \end{cases} \quad (1.1.11)$$

Is \mathcal{X} actually a determinantal point process? The answer is positive, \mathcal{X} is indeed a DPP but the corresponding kernel \mathbf{K} cannot be real-valued. To see this, we can write the inclusion probabilities and the compatibility relations as

$$\mathbb{P}[S \subset \mathcal{X}] = \begin{cases} 1, & \text{if } S = \emptyset, \\ \frac{1}{2}, & \text{if } |S| = 1, \\ \frac{1}{6}, & \text{if } |S| = 2, \\ 0, & \text{if } |S| = \mathbb{X}, \end{cases} \quad \text{and} \quad \begin{cases} \mathbf{K}_{11} = \mathbf{K}_{22} = \mathbf{K}_{33} = \frac{1}{2}, \\ \mathbf{K}_{12}\mathbf{K}_{21} = \mathbf{K}_{13}\mathbf{K}_{31} = \mathbf{K}_{23}\mathbf{K}_{32} = \frac{1}{12}, \\ (\mathbf{K}_{12}\mathbf{K}_{31}\mathbf{K}_{23})^2 + \frac{1}{12^3} = 0. \end{cases}$$

Observe that the latter condition is not feasible when $\mathbf{K}_{ij} \in \mathbb{R}$. However one can take the complex Hermitian kernel

$$\mathbf{K} = \begin{bmatrix} \frac{1}{2} & \frac{i}{\sqrt{12}} & -\frac{i}{\sqrt{12}} \\ -\frac{i}{\sqrt{12}} & \frac{1}{2} & \frac{i}{\sqrt{12}} \\ \frac{i}{\sqrt{12}} & -\frac{i}{\sqrt{12}} & \frac{1}{2} \end{bmatrix},$$

and realize (1.1.11) as $\text{DPP}(\mathbf{K})$.

We have just seen a simple toy example of $\text{DPP}(\mathbf{K})$ but we seek for more general conditions on K , \mathbf{K} or \mathbf{L} for the corresponding point process to exist.

WHEN DO DPPs ACTUALLY EXIST? We start with the so-called projection kernels, which play a fundamental role in the construction of more general DPPs, as we detail in Section 1.2, see, e.g., Theorems 1.2.3 and 1.2.4.

Definition 1.1.7 (Projection and orthogonal projection kernels). A function $K : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{C}$ is called a *projection kernel with rank $N \in \mathbb{N}^*$* if it satisfies

$$K(x, y) = \int_{\mathbb{X}} K(x, z)K(z, y)\mu(dz), \quad \forall x, y \in \mathbb{X}, \quad (1.1.12)$$

and

$$\int_{\mathbb{X}} K(x, x)\mu(dx) = N. \quad (1.1.13)$$

If K is also Hermitian, we call it an *orthogonal projection kernel*.¹²

¹² Orthogonal projection kernels are instances of reproducing kernels, see, e.g., Berlinet and Thomas-Agnan (2004, Example 1).

Proposition 1.1.8 (Projection and orthogonal projection DPPs).

According to Definition 1.1.7, let K be either

(a) a projection kernel with rank N , such that

$$\det[K(x_i, x_j)]_{i,j=1}^N \geq 0, \quad \forall x_1, \dots, x_N \in \mathbb{X}, \text{ or} \quad (1.1.14)$$

(b) an orthogonal projection kernel with rank N .

In both cases consider $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ with joint probability distribution

$$\frac{1}{N!} \det[K(x_i, x_j)]_{i,j=1}^N \mu^{\otimes N}(dx_1, \dots, dx_N). \quad (1.1.15)$$

Then, for any $1 \leq n \leq N$, $(\mathbf{x}_1, \dots, \mathbf{x}_n)$ has probability distribution

$$\frac{(N-n)!}{N!} \det[K(x_i, x_j)]_{i,j=1}^n \mu^{\otimes n}(dx_1, \dots, dx_n). \quad (1.1.16)$$

and $\{\mathbf{x}_1, \dots, \mathbf{x}_N\} \sim \text{DPP}(\mu, K)$, which we call a projection DPP, respectively an orthogonal projection DPP.

Proof. If we assume that (1.1.15) is a well-defined probability distribution with marginals (1.1.16), we can use the permutation invariance of (1.1.15) and plug the marginals $u_n(x_1, \dots, x_n) = (1.1.16)$, into Proposition 1.A.1 to see that the correlation functions of $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ are determinantal:

$$\rho_n(x_1, \dots, x_n) = \det[K(x_i, x_j)]_{i,j=1}^n, \quad \forall 1 \leq n \leq N.$$

It remains to prove that (1.1.15) indeed defines a probability distribution with marginals (1.1.16). First, the density associated to (1.1.15) is non-negative by assumption in case (a), while in case (b) it is a consequence of the fact that K is an *orthogonal projection* kernel,

$$\begin{aligned} \det[K(x_i, x_j)]_{i,j=1}^N &= \det \left[\int_{\mathbb{X}} K(x_i, y) K(y, x_j) \mu(dy) \right]_{i,j=1}^N \\ &= \frac{1}{N!} \int_{\mathbb{X}^N} \det[K(x_i, y_j)]_{i,j=1}^N \det[K(y_j, x_i)]_{i,j=1}^N \mu^{\otimes N}(dy_1, \dots, dy_N) \\ &= \frac{1}{N!} \int_{\mathbb{X}^N} \det[K(x_i, y_j)]_{i,j=1}^N \det[\overline{K(x_i, y_j)}]_{i,j=1}^N \mu^{\otimes N}(dy_1, \dots, dy_N) \\ &= \frac{1}{N!} \int_{\mathbb{X}^N} |\det[K(x_i, y_j)]_{i,j=1}^N|^2 \mu^{\otimes N}(dy_1, \dots, dy_N) \geq 0. \end{aligned} \quad (1.1.17)$$

In both cases, starting from $\det[K(x_i, x_j)]_{i,j=1}^N \geq 0$, successive applications of Lemma 1.A.2 reveal the $N!$ normalization of (1.1.15) along with the $(N-n)!$ term and the non-negativity of the minors

$$\det[K(x_i, x_j)]_{i,j=1}^n \geq 0, \quad \forall N \geq n \geq 1, \quad (1.1.18)$$

which define the marginals (1.1.16). \square

When the correlation kernel of $\text{DPP}(\mu, K)$, resp. $\text{DPP}(\mathbf{K})$ is assumed to be Hermitian, the existence of the DPP is guaranteed by a special condition on the eigenvalues of the underlying kernel, or alternatively as positive semi-definite conditions.

This corresponds to a generalization of Hough et al. (2009, Exercise 4.1.1) where we start from a projection kernel K itself which is (a) not assumed Hermitian, (b) not given by its eigen-decomposition.

By construction, projection DPPs have N points μ -almost surely.

Proposition 1.1.9 (Existence of Hermitian DPPs). *Let $K : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{C}$ be a continuous function, which is Hermitian*

See also Macchi (1975) or Soshnikov (2000, Theorem 3).

$$K(y, x) = \overline{K(x, y)}, \quad \forall x, y \in \mathbb{X}, \quad (1.1.19)$$

positive semi-definite, i.e., for all $f \in L^2(\mu)$

$$\int_{\mathbb{X}^2} f(x)K(x, y)\overline{f(y)}\mu(dx)\mu(dy) \geq 0, \quad (1.1.20)$$

and satisfies the trace class condition

$$\int_{\mathbb{X}} K(x, x)\mu(dx) < \infty, \quad (1.1.21)$$

together¹³ with the Hilbert-Schmidt condition

$$\int_{\mathbb{X}^2} |K(x, y)|^2\mu(dx)\mu(dy) < \infty. \quad (1.1.22)$$

Then the eigendecomposition of the kernel reads

$$K(x, y) = \sum_{k=1}^{\infty} \lambda_k \phi_k(x) \overline{\phi_k(y)}, \quad \text{where } \int_{\mathbb{X}} \phi_k(z) \overline{\phi_\ell(z)}\mu(dz) = \delta_{k\ell}. \quad (1.1.23)$$

Furthermore, $\text{DPP}(\mu, K)$ exists if and only if $0 \leq \lambda_k \leq 1$, for all $k \geq 1$.

We also mention that, when the kernel is assumed stationary, i.e., $K(x, y) = K_0(y - x)$, existence conditions reflect on the Fourier transform of K_0 , see, e.g., Lavancier, Møller, and Rubak (2015, Proposition 3.1). Indeed, under suitable conditions, if we note $\mathcal{F}(K_0)$ the Fourier transform,¹⁴ the existence condition simply reads as $\mathcal{F}(K_0) \leq 1$.

Next, we give the proof of the necessary and sufficient condition for a finite Hermitian DPP(\mathbf{K}) to exist.

Corollary 1.1.10. *Let $\mathbb{X} = \{1, \dots, M\}$ be finite and take $\mathbf{K} \in \mathbb{C}^{M \times M}$ to be Hermitian. Then $\text{DPP}(\mathbf{K})$ exists if and only if $0 \preceq \mathbf{K} \preceq I$.*

Proof. \Rightarrow Given that \mathbf{K} is Hermitian, $I - \mathbf{K}$ is Hermitian too. Then, since $\text{DPP}(\mathbf{K})$ exists then the complementary process $\mathcal{X}^c \sim \text{DPP}(I - \mathbf{K})$ also exists¹⁵ and we must have, for any $A \subset \mathbb{X}$,

$$\begin{cases} \mathbb{P}[A \subset \mathcal{X}] = \det[\mathbf{K}]_A \geq 0, \\ \mathbb{P}[A \subset \mathcal{X}^c] = \det[I - \mathbf{K}]_A \geq 0, \end{cases} \quad (1.1.24)$$

which is equivalent to $\mathbf{K} \succeq 0$ and $I - \mathbf{K} \succeq 0$ by Sylvester's criterion.¹⁶
 \Leftarrow Given that \mathbf{K} is Hermitian, since $0 \preceq \mathbf{K} \preceq I$, we can write the eigendecomposition

$$\mathbf{K} = \sum_{n=1}^M \lambda_n u_n u_n^H = U \Lambda U^H, \quad \text{with } 0 \leq \lambda_n \leq 1 \text{ and } U^H U = I, \quad (1.1.25)$$

to apply Theorem 1.2.4 to this finite setting. \square

In the same spirit, one can show that the Hermitian DPP(\mathbf{L}), as defined in Proposition 1.1.6, exists if and only if $\mathbf{L} \succeq 0$.

¹³ Note that when K is an orthogonal projection kernel, cf. Definition 1.1.7, conditions (1.1.21) and (1.1.22) are the same.

¹⁴ For any Borel function $h : \mathbb{R}^d \rightarrow \mathbb{C}$, $\mathcal{F}(h)(x) = \int h(y)e^{-2\pi i x^T y} dy, x \in \mathbb{R}^d$

¹⁵ See Corollary 1.D.4.

¹⁶ See, e.g., Horn and Johnson (2012, Theorem 7.2.5).

Observe the contrast with the existence condition for Hermitian DPP(\mathbf{K}), which requires two positive semi-definite constraints, namely $0 \preceq \mathbf{K} \preceq I$.¹⁷ As a consequence, the simple requirement $\mathbf{L} \succeq 0$ makes the L -ensemble viewpoint more practical for modeling purposes. Indeed, if we represent each item $m \in \mathbb{X}$ by a feature vector $\phi_m \in \mathbb{R}^d$ and store these vectors in a feature matrix $\Phi = [\phi_1, \dots, \phi_M] \in \mathbb{R}^{d \times M}$, then $\mathbf{L} = \Phi^\top \Phi \succeq 0$ defines a valid symmetric DPPs.

¹⁷ See Corollary 1.1.10.

CAN TWO KERNELS K AND K' DEFINE THE SAME DPP? In other words what is the equivalence class of kernels that give the same determinantal point process? We say that two DPP kernels K and K' are equivalent when the correlation functions of $\text{DPP}(\mu, K)$ and $\text{DPP}(\mu, K')$ match, that is, $\forall n \leq 1$ and x_1, \dots, x_n

$$\det[K(x_i, x_j)]_{i,j=1}^n = \det[K'(x_i, x_j)]_{i,j=1}^n. \quad (1.1.26)$$

Using the invariance by transposition of the determinant, a first simple example is $K'(x, y) = K(y, x)$. A second example relies on the multilinearity property of the determinant. If we consider a function g which does not vanish on \mathbb{X} , then we can take

$$K'(x, y) = g(x)K(x, y)g(y)^{-1} \quad (1.1.27)$$

Stevens (2019, Conjecture 1.4) even conjectures that these two cases describe the entire equivalent class of kernels. If we restrict to kernels that satisfy the symmetry condition $K(x, y) = K(y, x)$, Stevens (2019, Theorem 1) proves that the corresponding equivalence class is exactly described by (1.1.27) with $g : \mathbb{X} \rightarrow \{-1, 1\}$. For other insights in the discrete case, we refer to Kulesza (2012, Section 4.3.1, Theorem 4.1) and Poulson (2019, Proposition 2). This identifiability issue makes the inference of the kernel a subtle problem.

Moreover, we note that when the reference measure μ has a density ω w.r.t. another measure, say λ ,¹⁸ that is $\mu(dx) = \omega(x)\lambda(dx)$, then $\text{DPP}(\mu, K)$ can be alternatively seen as $\text{DPP}(\lambda, K')$ where

$$K'(x, y) = \sqrt{\omega(x)}K(x, y)\sqrt{\omega(y)}. \quad (1.1.28)$$

In the discrete case this reads $\mathbf{K}' = G\mathbf{K}G^{-1}$, where G is a diagonal matrix with no zero elements.

¹⁸ For instance, the Lebesgue measure.

We refer to this transformation as a change of base measure. In an informal way, the base measure can be either put into the kernel or taken out from the kernel. In the discrete case, one can work equivalently with $\text{DPP}(\boldsymbol{\omega}, \mathbf{K})$ or $\text{DPP}(\boldsymbol{\Omega}^{\frac{1}{2}} \mathbf{K} \boldsymbol{\Omega}^{\frac{1}{2}})$, where $\boldsymbol{\Omega} \triangleq \text{diag}(\boldsymbol{\omega})$.

1.2 HOW TO CONSTRUCT A DPP?

We present three ways to construct valid DPP kernels, either starting from a valid kernel K , a set of linearly independent functions or simply vectors.

RESCALING THE KERNEL K BY A CONSTANT YIELDS A ANOTHER DPP.

Proposition 1.2.1. (*Stability by thinning*). *Let $\lambda > 1$ and $\mathcal{X} \sim \text{DPP}(\mu, K)$. Consider the point process \mathcal{X}^λ whose realizations correspond to the ones of \mathcal{X} where each point is deleted independently with probability $1 - \frac{1}{\lambda}$. Then $\mathcal{X}^\lambda \sim \text{DPP}(\mu, \frac{1}{\lambda}K)$.*

Proof. Let $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ and compute the correlation functions of the thinned process \mathcal{X}^λ

$$\begin{aligned} & \mathbb{E} \left[\sum_{\substack{(\mathbf{x}_1, \dots, \mathbf{x}_n) \\ \mathbf{x}_1 \neq \dots \neq \mathbf{x}_n \in \mathcal{X}}} f(\mathbf{x}_1, \dots, \mathbf{x}_n) \prod_{i=1}^n \mathbb{1}_{\{\mathbf{x}_i \in \mathcal{X}^\lambda\}} \right] \\ &= \mathbb{E} \left[\mathbb{E} \left[\sum_{\substack{(\mathbf{x}_1, \dots, \mathbf{x}_n) \\ \mathbf{x}_1 \neq \dots \neq \mathbf{x}_n \in \mathcal{X}}} f(\mathbf{x}_1, \dots, \mathbf{x}_n) \prod_{i=1}^n \mathbb{1}_{\{\mathbf{x}_i \in \mathcal{X}^\lambda\}} \mid \mathcal{X} \right] \right] \\ &= \mathbb{E} \left[\sum_{\substack{(\mathbf{x}_1, \dots, \mathbf{x}_n) \\ \mathbf{x}_1 \neq \dots \neq \mathbf{x}_n \in \mathcal{X}}} f(\mathbf{x}_1, \dots, \mathbf{x}_n) \prod_{i=1}^n \mathbb{E} \left[\mathbb{1}_{\{\mathbf{x}_i \in \mathcal{X}^\lambda\}} \mid \mathcal{X} \right] \right] \\ &= \mathbb{E} \left[\sum_{\substack{(\mathbf{x}_1, \dots, \mathbf{x}_n) \\ \mathbf{x}_1 \neq \dots \neq \mathbf{x}_n \in \mathcal{X}}} f(\mathbf{x}_1, \dots, \mathbf{x}_n) \frac{1}{\lambda^n} \right] \\ &= \int_{\mathbb{X}^n} f(x_1, \dots, x_n) \frac{1}{\lambda^n} \det[K(x_i, x_j)]_{i,j=1}^n \mu^{\otimes n}(dx_1, \dots, dx_n) \\ &= \int_{\mathbb{X}^n} f(x_1, \dots, x_n) \det \left[\frac{1}{\lambda} K(x_i, x_j) \right]_{i,j=1}^n \mu^{\otimes n}(dx_1, \dots, dx_n) \end{aligned}$$

Conditionally on the realizations of \mathcal{X} , the deletions are independent.

Each point is deleted with probability $1 - 1/\lambda$, thus kept with probability $1/\lambda$.

By definition of the correlation functions, cf. Definition 1.1.1.

□

PROJECTION DPPS CAN BE CONSTRUCTED FROM A SET OF FUNCTIONS OR VECTORS. Depending on the task at hand, an adapted choice of projection kernel may provide strong theoretical guarantees. In particular, tailored choices of projection DPPs proved to be useful, e.g., in the context of Monte Carlo integration (Bardenet and Hardy, 2020; Gautier, Bardenet, and Valko, 2019b; Mabroyer, Coeurjolly, and Amblard, 2019),¹⁹ but also for kernel quadrature or column subset selection (Belhadji, Bardenet, and Chainais, 2019, 2018). Moreover, as we will see in the subsequent chapters, projection DPPs can be sampled efficiently which allow to turn theory into practice.

¹⁹ This is the purpose of Chapter 4.

Proposition 1.2.2 (Biorthogonal ensembles). *For a fixed $N \geq 1$, take linearly independent measurable functions ϕ_1, \dots, ϕ_N , resp. ψ_1, \dots, ψ_N , such that $\int_{\mathbb{X}} |\psi_k(x)\phi_\ell(x)|\mu(dx) < \infty, \forall 1 \leq k, \ell \leq N$.*

Consider $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ with probability distribution²⁰

$$\frac{1}{N!} \frac{\det[\phi_k(x_n)]_{k,n=1}^N \det[\overline{\psi_k(x_n)}]_{k,n=1}^N}{\det \left[\int_{\mathbb{X}} \overline{\psi_k(z)}\phi_\ell(z)\mu(dz) \right]_{k,\ell=1}^N} \mu^{\otimes N}(dx_1, \dots, dx_N). \quad (1.2.1)$$

Then, $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ defines a projection DPP(μ, K) with kernel²¹

$$K(x, y) = \sum_{k,\ell=1}^N \phi_k(x) [\mathbf{A}^{-1}]_{k\ell} \overline{\psi_\ell(y)}, \quad (1.2.2)$$

where $\mathbf{A} = \left[\int_{\mathbb{X}} \overline{\psi_k(z)}\phi_\ell(z)\mu(dz) \right]_{k,\ell=1}^N$. We call $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ a biorthogonal ensemble.

The prototypical example of a biorthogonal system of function on $\mathbb{X} = [-\pi, \pi]$ is $\{\phi_k(x) = \cos(kx), \psi_k(x) = \sin(kx)\}_{k \in \mathcal{I}}$ where $|\mathcal{I}| = N$. Intuitively, this example might find interesting applications in signal processing. In the case where $\psi_k \equiv \phi_k$ are orthogonal polynomials w.r.t. μ , the corresponding DPP also refers to as an orthogonal polynomial ensemble (OPE). They notably appear in random matrix theory, where they characterize the eigenvalue distribution of some special random matrices (König, 2004; König, O'Connell, and Roch, 2002). We mention that, in Chapter 4, we consider the points of a particular multivariate OPE as random quadrature nodes in the context of Monte Carlo integration. Moreover, OPEs are in turn a specific instance of β -ensembles ($\beta = 2$). Sampling from β -ensembles is the purpose of Chapter 5.

IN THE FINITE CASE, ORTHOGONAL PROJECTION KERNELS WITH RANK N CAN BE CONSTRUCTED FROM A SET OF N LINEARLY INDEPENDENT VECTORS. Consider that each item $m \in \mathbb{X}$ is represented by a feature vector $\phi_m \in \mathbb{R}^N$, so that the feature matrix $\Phi \in \mathbb{R}^{N \times M}$ is full row rank, i.e., $\text{rank } \Phi = N \leq M$. Then $\mathbf{K} = \Phi^\top (\Phi \Phi^\top)^{-1} \Phi$ defines the orthogonal projection onto the feature space, i.e., the vector space spanned by the rows $\Phi_{1:}, \dots, \Phi_{N:} \in \mathbb{R}^M$.

Besides, since the orthogonal projection DPP(\mathbf{K}) has almost surely N points, the likelihood of a set $S \subset \mathbb{X}$ reads

$$\mathbb{P}[\mathcal{X} = S] = \det[\Phi^\top (\Phi \Phi^\top)^{-1} \Phi]_{S,S} \mathbb{1}_{|S|=N} = \frac{(\det \Phi_{:,S})^2}{\det \Phi \Phi^\top} \mathbb{1}_{|S|=N}, \quad (1.2.3)$$

which interprets as

$$\mathbb{P}[\mathcal{X} = S] \propto \text{volume}^2 \{\phi_m\}_{m \in S} \mathbb{1}_{|S|=N}. \quad (1.2.4)$$

That is, the larger the volume spanned by the feature vectors $\{\phi_m\}_{m \in S}$ the more likely S appears as a realization of DPP(\mathbf{K}).

This corresponds to Johansson (2006, Proposition 2.11) where $\psi \equiv \bar{\psi}$.

²⁰ assuming the ratio is nonnegative with nonzero denominator, e.g., $\psi \equiv \phi$. The determinant in the denominator corresponds to the normalizing constant and is given by the Cauchy-Binet formula (1.C.2).

²¹ The kernel K characterizes the oblique projection operator onto $\text{span}\{\phi_1, \dots, \phi_N\}$ along the orthocomplement of $\text{span}\{\psi_1, \dots, \psi_N\}$.

We mention that this geometrical formulation was the main source of inspiration for the conception of our approximate projection DPP sampler, see Section 3.3.

SOME MORE GENERAL DPPS CAN BE CONSTRUCTED AS MIXTURES OF PROJECTION DPPS.

Theorem 1.2.3 (DPPs as mixtures of projection DPPs). *Let ϕ_1, \dots, ϕ_N and ψ_1, \dots, ψ_N be as in Proposition 1.2.2 with the additional assumption $\int_{\mathbb{X}} \phi_k(z) \psi_\ell(z) dz = \delta_{k\ell}, \forall 1 \leq k, \ell \leq N$ and $0 \leq \lambda_1, \dots, \lambda_N \leq 1$. Consider the point process \mathcal{X} generated as follows*

1. Draw independent random variables $B_k \sim \text{Ber}(\lambda_k)$ for $1 \leq k \leq N$.
2. Conditionally on the realization of the Bernoulli variables, sample from the projection DPP(μ, K^B) where

$$K^B(x, y) = \sum_{k=1}^N B_k \phi_k(x) \overline{\psi_k(y)}. \quad (1.2.5)$$

Then, the process \mathcal{X} defines a determinantal point process DPP(μ, K) in the sense of Definition 1.1.3, with kernel

$$K(x, y) = \sum_{k=1}^N \lambda_k \phi_k(x) \overline{\psi_k(y)}. \quad (1.2.6)$$

In particular the number of points $|\mathcal{X}|$ is random with distribution²²

$$|\mathcal{X}| \stackrel{\text{law}}{=} \sum_{k=1}^N B_k. \quad (1.2.7)$$

This corresponds to a variant of Hough et al. (2009, Theorem 4.5.3).

Proof. First, observe that the process \mathcal{X} constructed in Theorem 1.2.3 is well defined. Conditionally on the Bernoullis, K^B is a projection kernel with rank equal to $\sum_{k=1}^N B_k$, and Proposition 1.1.8 ensures that DPP(μ, K^B) defines in turn a projection DPP with $\sum_{k=1}^N B_k$ points, hence (1.2.7). Then, to prove that $\mathcal{X} \sim \text{DPP}(\mu, K)$ with kernel (1.2.6), we check that the correlation functions of \mathcal{X} are indeed equal to

$$\rho_n(x_1, \dots, x_n) = \det[K(x_i, x_j)]_{i,j=1}^N.$$

Let f be a suitable test function

$$\begin{aligned} & \mathbb{E} \left[\sum_{\substack{(\mathbf{x}_1, \dots, \mathbf{x}_n) \\ \mathbf{x}_1 \neq \dots \neq \mathbf{x}_n \in \mathcal{X}}} f(\mathbf{x}_1, \dots, \mathbf{x}_n) \right] \\ &= \mathbb{E} \left[\mathbb{E} \left[\sum_{\substack{(\mathbf{x}_1, \dots, \mathbf{x}_n) \\ \mathbf{x}_1 \neq \dots \neq \mathbf{x}_n \in \mathcal{X}}} f(\mathbf{x}_1, \dots, \mathbf{x}_n) \mid B_1, \dots, B_N \right] \right] \\ &= \mathbb{E} \left[\int_{\mathbb{X}^n} f(x_1, \dots, x_n) \det[K^B(x_i, x_j)]_{i,j=1}^n \mu^{\otimes n}(dx_1, \dots, dx_n) \right] \quad \text{Using Cauchy-Binet formula (1.C.1).} \\ &= \mathbb{E} \left[\int_{\mathbb{X}^n} f(x_1, \dots, x_n) \sum_{\substack{S \subset \{1, \dots, N\} \\ |S|=n}} \det[B_j \phi_j(x_i)]_{i=1, j \in S}^n \det[\overline{\psi_i(x_j)}]_{j=1, i \in S}^n \mu^{\otimes n}(dx_1, \dots, dx_n) \right] \\ &= \int_{\mathbb{X}^n} f(x_1, \dots, x_n) \sum_{\substack{S \subset \{1, \dots, N\} \\ |S|=n}} \mathbb{E} \left[\prod_{j \in S} B_j \right] \det[\phi_j(x_i)]_{i=1, j \in S}^n \det[\overline{\psi_i(x_j)}]_{j=1, i \in S}^n \mu^{\otimes n}(dx_1, \dots, dx_n) \end{aligned}$$

$$\begin{aligned}
&= \int_{\mathbb{X}^n} f(x_1, \dots, x_n) \sum_{\substack{S \subset \{1, \dots, N\} \\ |S|=n}} \prod_{j \in S} \lambda_j \det[\phi_j(x_i)]_{i=1, j \in S}^n \det[\overline{\psi_i(x_j)}]_{j=1, i \in S}^n \mu^{\otimes n}(dx_1, \dots, dx_n) \\
&= \int_{\mathbb{X}^n} f(x_1, \dots, x_n) \sum_{\substack{S \subset \{1, \dots, N\} \\ |S|=n}} \det[\lambda_j \phi_j(x_i)]_{i=1, j \in S}^n \det[\overline{\psi_i(x_j)}]_{j=1, i \in S}^n \mu^{\otimes n}(dx_1, \dots, dx_n) \\
&= \int_{\mathbb{X}^n} f(x_1, \dots, x_n) \det \left[\sum_{n=1}^N \lambda_n \phi_n(x_i) \overline{\psi_n(x_j)} \right]_{i,j=1}^n \mu^{\otimes n}(dx_1, \dots, dx_n). \quad \text{Using Cauchy-Binet formula (1.C.1).}
\end{aligned}$$

□

The previous construction can be adapted to the case where K is Hermitian. This allows us to create Hermitian DPPs with at most N points from a set of N orthonormal functions.

Theorem 1.2.4 (Hermitian DPPs as mixtures of orthogonal projection DPPs). *Let $\phi_1, \dots, \phi_N \in L^2(\mu)$ be orthonormal, that is*

$$\int_{\mathbb{X}} \phi_k(z) \overline{\phi_\ell(z)} dz = \delta_{k\ell}, \forall 1 \leq k, \ell \leq N, \quad (1.2.8)$$

and take $0 \leq \lambda_1, \dots, \lambda_N \leq 1$. Consider the point process \mathcal{X} generated as follows

1. *Draw independent random variables $B_k \sim \text{Ber}(\lambda_k)$ for $1 \leq k \leq N$.*
2. *Conditionally on the realization of the Bernoulli variables, sample from the orthogonal projection DPP(μ, K^B) where*

$$K^B(x, y) = \sum_{k=1}^N B_k \phi_k(x) \overline{\phi_k(y)}. \quad (1.2.9)$$

Then, according to Definition 1.1.3, the process \mathcal{X} defines an orthogonal determinantal point process DPP(μ, K) with kernel

$$K(x, y) = \sum_{k=1}^N \lambda_k \phi_k(x) \overline{\phi_k(y)}. \quad (1.2.10)$$

*In particular the number of points $|\mathcal{X}|$ is random with distribution*²³

²³ It is called Poisson Binomial.

$$|\mathcal{X}| \stackrel{\text{law}}{=} \sum_{k=1}^N B_k \quad (1.2.11)$$

Conversely, a kernel in the form of (1.2.10) defines the Hermitian DPP(μ, K).

This corresponds to a variant of Hough et al. (2009, Theorem 4.5.3).

APPENDICES

We compile a list of results, which are found to be useful in the main text or simply to fulfill the curiosity of the reader. For the results which were simply stated in the literature or left as an exercise, we strive to give new or at least more explicit proofs.

1.A CONSTRUCTION OF FIXED-SIZED POINT PROCESSES FROM EXCHANGEABLE VARIABLES

Given the joint probability density function of a random vector $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ which is invariant by permutation of its coordinates, the ordering of the coordinates can be removed to consider $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ as a point process with exactly N points.

Proposition 1.A.1. *Consider a random vector $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ whose probability density $u_N(x_1, \dots, x_N)$ w.r.t. $\mu^{\otimes N}$ is invariant to permutation of the coordinates. Then $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ defines a point process with correlation functions ρ_1, \dots, ρ_N given by*

$$\rho_n(x_1, \dots, x_n) = \frac{N!}{(N-n)!} u_n(x_1, \dots, x_n), \quad (1.A.1)$$

This corresponds to Exercise 1.2.5 of Hough et al. (2009).

where $u_n(x_1, \dots, x_n) \triangleq \int_{\mathbb{X}^{N-n}} u_N(x_1, \dots, x_N) \mu^{\otimes N-n}(dx_{n+1}, \dots, dx_N)$ denotes the marginal probability density function of n points.

Proof. For $1 \leq n \leq N$, denote

$$\mathcal{I}_n^N \triangleq \{\tau : \{1, \dots, n\} \rightarrow \{1, \dots, N\} \mid \tau \text{ is injective}\}, \quad (1.A.2)$$

and derive the correlation function ρ_n from Definition 1.1.1

$$\begin{aligned} \mathbb{E} \left[\sum_{\substack{(\mathbf{x}_1, \dots, \mathbf{x}_n) \\ \mathbf{x}_1 \neq \dots \neq \mathbf{x}_n \in \mathcal{X}}} f(\mathbf{x}_1, \dots, \mathbf{x}_n) \right] &= \mathbb{E} \left[\sum_{\tau \in \mathcal{I}_n^N} f(\mathbf{x}_{\tau(1)}, \dots, \mathbf{x}_{\tau(n)}) \right] \\ &= \int_{\mathbb{X}^N} \sum_{\tau \in \mathcal{I}_n^N} f(x_{\tau(1)}, \dots, x_{\tau(n)}) u_N(x_1, \dots, x_N) \mu^{\otimes N}(dx_1, \dots, dx_N), \end{aligned}$$

since u_n is invariant to permutation and the integration is over \mathbb{X}^N

$$= \sum_{\tau \in \mathcal{I}_n^N} \int_{\mathbb{X}^N} f(x_{\tau(1)}, \dots, x_{\tau(n)}) u_N(x_{\tau(1)}, \dots, x_{\tau(n)}, y_1, \dots, y_{N-n}) \mu^{\otimes N}(dx_{\tau(1)}, \dots, dx_{\tau(n)}, dy_1, \dots, dy_{N-n}),$$

but $x_{\tau(1)}, \dots, x_{\tau(n)}$ are dummy variables, we can write

$$= \sum_{\tau \in \mathcal{I}_n^N} \int_{\mathbb{X}^n} f(x_1, \dots, x_n) \left(\int_{\mathbb{X}^{N-n}} u_N(x_1, \dots, x_n, x_{n+1}, \dots, x_N) \mu^{\otimes N-n}(dx_{n+1}, \dots, dx_N) \right) \mu^{\otimes n}(dx_1, \dots, dx_n),$$

the dependence on τ vanishes and there are $|\mathcal{I}_n^N| = N!/(N-n)!$ such injective mappings

$$= \int_{\mathbb{X}^n} f(x_1, \dots, x_n) \underbrace{\left(\frac{N!}{(N-n)!} \int_{\mathbb{X}^{N-n}} u_N(x_1, \dots, x_N) \mu^{\otimes N-n}(dx_{n+1}, \dots, dx_N) \right)}_{=\rho_n(x_1, \dots, x_n)} \mu^{\otimes n}(dx_1, \dots, dx_n).$$

□

In Proposition 1.1.8 we combine the previous result with the following lemma to validate the construction of projection DPPs. This lemma will provide the determinantal formulation of the correlation functions.

Lemma 1.A.2. *For any rank- N projection kernel and $1 \leq n \leq N$,*

$$\int_{\mathbb{X}} \det[K(x_i, x_j)]_{i,j=1}^n \mu(dx_n) = (N - (n - 1)) \det[K(x_i, x_j)]_{i,j=1}^{n-1}, \quad (1.A.3)$$

which specializes to the finite setting as

$$\sum_{x_n=1}^{|\mathbb{X}|} \det[\mathbf{K}_{x_i, x_j}]_{i,j=1}^n = (N - (n - 1)) \det[\mathbf{K}_{x_i, x_j}]_{i,j=1}^{n-1}. \quad (1.A.4)$$

Proof. We can apply Lemma 1.B.1 to express

$$\begin{aligned} & \int_{\mathbb{X}} \det[K(x_i, x_j)]_{i,j=1}^n \mu(dx_n) \\ &= \int_{\mathbb{X}} [K(x_n, x_n) \det K(x_{1:n-1}, x_{1:n-1}) - K(x_n, x_{1:n-1}) \operatorname{adj}(K(x_{1:n-1}, x_{1:n-1})) K(x_{1:n-1}, x_n)] \mu(dx_n) \\ &= \det K(x_{1:n-1}, x_{1:n-1}) \int_{\mathbb{X}} K(x_n, x_n) \mu(dx_n) - \sum_{i,j=1}^{n-1} \int_{\mathbb{X}} K(x_n, x_i) [\operatorname{adj} K(x_{1:n-1}, x_{1:n-1})]_{ij} K(x_j, x_n) \mu(dx_n) \\ &= \det K(x_{1:n-1}, x_{1:n-1}) \int_{\mathbb{X}} K(x_n, x_n) \mu(dx_n) - \sum_{i,j=1}^{n-1} [\operatorname{adj} K(x_{1:n-1}, x_{1:n-1})]_{ij} \int_{\mathbb{X}} K(x_j, x_n) K(x_n, x_i) \mu(dx_n) \\ &= \det K(x_{1:n-1}, x_{1:n-1}) \int_{\mathbb{X}} K(x_n, x_n) \mu(dx_n) - \operatorname{Tr} \left[\operatorname{adj}(K(x_{1:n-1}, x_{1:n-1})) \left[\int_{\mathbb{X}} K(x_i, x_n) K(x_n, x_j) \mu(dx_n) \right]_{i,j=1}^{n-1} \right], \end{aligned}$$

and conclude using the fact that K is a rank- N projection kernel

$$\begin{aligned} & \int_{\mathbb{X}} \det[K(x_i, x_j)]_{i,j=1}^n \mu(dx_n) \\ &= N \det K(x_{1:n-1}, x_{1:n-1}) - \operatorname{Tr}[\operatorname{adj}(K(x_{1:n-1}, x_{1:n-1})) K(x_{1:n-1}, x_{1:n-1})] \quad \text{By (1.1.13) and (1.1.12).} \\ &= N \det K(x_{1:n-1}, x_{1:n-1}) - \det K(x_{1:n-1}, x_{1:n-1}) \operatorname{Tr}[I_{n-1}] \\ &= (N - (n - 1)) \det[K(x_i, x_j)]_{i,j=1}^{n-1}. \end{aligned}$$

By definition of the adjugate: $\operatorname{adj}(A)A = \det[A]I$.

□

1.B CLASSICAL MATRIX RESULTS

First, we prove a classical and useful lemma to compute the determinant of a block matrix $\begin{bmatrix} A & U \\ V & C \end{bmatrix}$ and give a slight variant²⁴ when A is not invertible and $C \in \mathbb{R}^{1 \times 1}$. Then we state a crucial result characterizing sums of principal minors of a symmetric kernel. This result is the main ingredient used in Appendix 1.D, where we derive some stability properties of finite DPPs.

²⁴ used in the proof of Lemma 1.A.2.

Lemma 1.B.1 (Determinant and Schur complement). *Let $A \in \mathbb{C}^{m \times m}$, $C \in \mathbb{C}^{n \times n}$, $U \in \mathbb{C}^{m \times n}$ and $V \in \mathbb{C}^{n \times m}$.*

$$\det \begin{bmatrix} A & U \\ V & C \end{bmatrix} = \begin{cases} \det A \det(C - VA^{-1}U), & \text{if } A \text{ is invertible,} \\ C \det A - V \operatorname{adj}(A)U, & \text{if } A \text{ is not invertible and } n = 1, \\ \det C \det(A - UC^{-1}V), & \text{if } C \text{ is invertible.} \end{cases} \quad (1.B.1)$$

Proof. If A is invertible, an LDU decomposition yields

$$\begin{bmatrix} A & U \\ V & C \end{bmatrix} = \begin{bmatrix} I_m & 0 \\ VA^{-1} & I_n \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & C - VA^{-1}U \end{bmatrix} \begin{bmatrix} I_m & A^{-1}U \\ 0 & I_n \end{bmatrix}, \quad (1.B.2)$$

and the first result follows. If A is not invertible and $n = 1$, we use the fact that the set of invertible matrices is dense in the set of matrices. More specifically, consider a sequence of invertible matrices such that $M_k \xrightarrow{k \rightarrow \infty} A$. For any $k \geq 1$, M_k is invertible, thus the first result applies and the adjugate reads $\operatorname{adj}(M_k) = M_k^{-1} \det M_k$. Then,

$$\begin{aligned} \det \begin{bmatrix} M_k & U \\ V & C \end{bmatrix} &= \det M_k \det[C - VM_k^{-1}U] \\ &= C \det M_k - V \operatorname{adj}(M_k)U. \end{aligned}$$

Finally, the two sides of the above relation being continuous functions in the entries of M_k , we can take the limit and the result follows. \square

Theorem 1.B.2 (Sum of principal minors). *Consider a matrix $K \in \mathbb{C}^{|\mathbb{X}| \times |\mathbb{X}|}$. For any $A \subset B \subset \mathbb{X}$, we have*

$$\sum_{A \subset S \subset B} \det K_S = \det \left[I^A K + I^{A^c} (I + K) \right]_B. \quad (1.B.3) \quad \text{where } [I^A]_{ij} = \mathbf{1}_{i=j \in A}.$$

Proof. See, e.g., Kulesza and Taskar (2012, Theorem 2.1) or Borodin, Okounkov, and Olshanski (2000, Proposition A.4). \square

1.C CAUCHY-BINET FORMULAS

The Cauchy-Binet formula and its extension to the continuous case will prove to be useful in various situations,²⁵ to identify the determinant of the product of two matrices as a sum of products of determinants.

Proposition 1.C.1 (Cauchy-Binet formula for matrices).

Let $1 \leq m \leq n$ and consider two rectangular matrices $A \in \mathbb{C}^{m \times n}$ and $B \in \mathbb{C}^{n \times m}$. Then, the Cauchy-Binet formula reads

$$\det AB = \sum_{\substack{S \subset \{1, \dots, n\} \\ |S|=m}} \det A_{:,S} \det B_{S,:}, \quad (1.C.1)$$

where $A_{:,S} = [A_{ij}]_{i=1,j \in S}^m$ and $B_{S,:} = [B_{ij}]_{j=1,i \in S}^m$.

Proposition 1.C.2 (Generalized Cauchy-Binet formula).

Let $(\mathbb{X}, \mathcal{B}, \mu)$ be a measurable space and consider some measurable functions ϕ_1, \dots, ϕ_N and ψ_1, \dots, ψ_N , such that for all $1 \leq i, j \leq N$,

²⁵ See, e.g., Proposition 1.1.8, Proposition 1.2.2, Theorem 1.2.3, Chapter 4.

The determinant of the product of two rectangular matrices is the sum over all square submatrices of the product of determinants.

This corresponds to Johansson (2006, Proposition 2.10).

$\int_{\mathbb{X}} |\phi_i(x)\psi_j(x)|\mu(dx) < \infty$. Then, we have

$$\begin{aligned} & \det \left[\int_{\mathbb{X}} \phi_i(x)\psi_j(x)\mu(dx) \right]_{i,j=1}^N \\ &= \frac{1}{N!} \int_{\mathbb{X}^N} \det[\phi_i(x_j)]_{i,j=1}^N \det[\psi_i(x_j)]_{i,j=1}^N \mu^{\otimes N}(dx_1, \dots, dx_N) \quad (1.C.2) \\ &= \frac{1}{N!} \int_{\mathbb{X}^N} \det \Phi(x_{1:N}) \det \Psi(x_{1:N}) \mu^{\otimes N}(dx_1, \dots, dx_N). \quad (1.C.3) \end{aligned}$$

$$\begin{aligned} \Phi(x_{1:N}) &= \begin{bmatrix} \phi_1(x_1) & \dots & \phi_N(x_1) \\ \vdots & & \vdots \\ \phi_1(x_N) & \dots & \phi_N(x_N) \end{bmatrix} \\ \Psi(x_{1:N}) &= \begin{bmatrix} \psi_1(x_1) & \dots & \psi_N(x_1) \\ \vdots & & \vdots \\ \psi_1(x_N) & \dots & \psi_N(x_N) \end{bmatrix} \end{aligned}$$

1.D STABILITY PROPERTIES OF FINITE DPPs

In this section, the ground space \mathbb{X} is assumed to be finite. We focus on proving some of the stability properties of DPPs under different set operations $f(\mathcal{X})$ or conditioning $\mathcal{X} | ?$. We first derive the inclusion-exclusion principle to express quantities of the form $\mathbb{P}[A \subset \mathcal{X}, \mathcal{X} \cap B = \emptyset]$ for general point processes \mathcal{X} . Then, we use Theorem 1.B.2 to specialize the inclusion-exclusion principle to finite DPPs, which reveals the determinantal structure of the previous quantities. As a direct application we can explicitly derive the likelihood $\mathbb{P}[\mathcal{X} = A]$ dear to people using the L -ensemble viewpoint and the inclusion probabilities of the complementary process $\mathbb{X} \setminus \mathcal{X}$ which proves to be a DPP with correlation kernel $I - \mathbf{K}$.

Notation For any $A \subset \mathbb{X}$, we use the notation I^A to denote the indicator matrix of the set A , i.e., the $M \times M$ diagonal matrix with ones only at indices of A . More formally,

$$[I^A]_{ij} \triangleq \mathbb{1}_{i=j \in A}, \quad \text{for all } i, j \in \mathbb{X}. \quad (1.D.1)$$

Theorem 1.D.1 (Inclusion-exclusion principle for point processes). *Let \mathbb{X} be finite and consider the point process $\mathcal{X} \subset \mathbb{X}$. For any disjoint subsets $A, B \subset \mathbb{X}$, the following holds*

$$\begin{aligned} \mathbb{P}[A \subset \mathcal{X}, \mathcal{X} \cap B = \emptyset] &= \sum_{S \subset B} (-1)^{|S|} \mathbb{P}[A \cup S \subset \mathcal{X}] \\ &= (-1)^{|A|} \sum_{A \subset S \subset A \cup B} (-1)^{|S|} \mathbb{P}[S \subset \mathcal{X}]. \end{aligned}$$

Proof. If $\mathbb{P}[A \subset \mathcal{X}] = 0$ then $\mathbb{P}[A \subset \mathcal{X}, \mathcal{X} \cap B = \emptyset] = 0$, otherwise we can write

$$\begin{aligned} & \mathbb{P}[A \subset \mathcal{X}, \mathcal{X} \cap B = \emptyset] \\ &= \mathbb{P}[A \subset \mathcal{X}] \mathbb{P}[\mathcal{X} \cap B = \emptyset | A \subset \mathcal{X}] \\ &= \mathbb{P}[A \subset \mathcal{X}] \mathbb{P}\left[\bigcap_{b \in B} \{b \notin \mathcal{X}\} | A \subset \mathcal{X}\right] \\ &= \mathbb{P}[A \subset \mathcal{X}] \left[1 - \mathbb{P}\left[\bigcup_{b \in B} \{b \in \mathcal{X}\} | A \subset \mathcal{X}\right]\right] \\ &= \mathbb{P}[A \subset \mathcal{X}] \left[1 - \sum_{n=1}^{|B|} (-1)^{n-1} \sum_{\substack{S \subset B \\ |S|=n}} \mathbb{P}\left[\bigcap_{b \in S} \{b \in \mathcal{X}\} | A \subset \mathcal{X}\right]\right] \end{aligned}$$

For $M = 4$ and $A = \{2, 4\}$
 $I^A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

By the inclusion-exclusion principle.

$$\begin{aligned}
&= \mathbb{P}[A \subset \mathcal{X}] \left[\mathbb{P}[\emptyset \subset \mathcal{X} \mid A \subset \mathcal{X}] + \sum_{n=1}^{|B|} (-1)^n \sum_{\substack{S \subset B \\ |S|=n}} \mathbb{P}[S \subset \mathcal{X} \mid A \subset \mathcal{X}] \right] \\
&= \sum_{S \subset B} (-1)^{|S|} \mathbb{P}[S \subset \mathcal{X}, A \subset \mathcal{X}] = \sum_{S \subset B} (-1)^{|S|} \mathbb{P}[A \sqcup S \subset \mathcal{X}] \\
&= (-1)^{|A|} \sum_{A \subset S \subset A \sqcup B} (-1)^{|S|} \mathbb{P}[S \subset \mathcal{X}].
\end{aligned}$$

Since A and B are disjoint.

□

Theorem 1.D.2 (Inclusion-exclusion principle for DPPs). *Let \mathbb{X} be finite and consider $\mathcal{X} \sim \text{DPP}(\mathbf{K})$. Then, for any disjoint subsets $A, B \subset \mathbb{X}$, we have*

$$\mathbb{P}[A \subset \mathcal{X}, \mathcal{X} \cap B = \emptyset] = \det \left[I^A \mathbf{K} + I^{A^c} (I - \mathbf{K}) \right]_{A \sqcup B}. \quad (1.D.2)$$

In particular,

$$\mathbb{P}[A \subset \mathcal{X}, \mathcal{X} \cap B = \emptyset] = \begin{cases} \det[I - \mathbf{K}]_B \det[\mathbf{K} + \mathbf{K}_{:B} [I - \mathbf{K}]_B^{-1} \mathbf{K}_{B:}]_A, & \text{if } \mathbb{P}[\mathcal{X} \cap B = \emptyset] > 0, \\ \det[\mathbf{K}]_A \det[I - (\mathbf{K} - \mathbf{K}_{:A} \mathbf{K}_A^{-1} \mathbf{K}_{A:})]_B, & \text{if } \mathbb{P}[A \subset \mathcal{X}] > 0. \end{cases} \quad (1.D.3)$$

Proof. This is a simple application of the inclusion-exclusion principle given in Theorem 1.D.1 combined with Theorem 1.B.2.

$$\begin{aligned}
\mathbb{P}[A \subset \mathcal{X}, \mathcal{X} \cap B = \emptyset] &= (-1)^{|A|} \sum_{A \subset S \subset A \sqcup B} (-1)^{|S|} \mathbb{P}[S \subset \mathcal{X}] \\
&= (-1)^{|A|} \sum_{A \subset S \subset A \sqcup B} \det[-\mathbf{K}]_S \\
&= (-1)^{|A|} \det \left[I^A (-\mathbf{K}) + I^{A^c} (I - \mathbf{K}) \right]_{A \sqcup B} \\
&= \det \left[I^A \mathbf{K} + I^{A^c} (I - \mathbf{K}) \right]_{A \sqcup B}.
\end{aligned} \quad (1.D.4)$$

As a consequence, we have $\mathbb{P}[\mathcal{X} \cap B = \emptyset] = \det[I - \mathbf{K}]_B$ and we recover $\mathbb{P}[A \subset \mathcal{X}] = \det \mathbf{K}_A$. Hence, if $\mathbb{P}[\mathcal{X} \cap B = \emptyset] > 0$ then $[I - \mathbf{K}]_B$ is invertible and (1.D.4) becomes

$$\begin{aligned}
\mathbb{P}[A \subset \mathcal{X}, \mathcal{X} \cap B = \emptyset] &= \det \begin{bmatrix} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ [I - \mathbf{K}]_{BA} & [I - \mathbf{K}]_{BB} \end{bmatrix} \\
&= \det[I - \mathbf{K}]_B \det[\mathbf{K}_{AA} - \mathbf{K}_{AB} [I - \mathbf{K}]_B^{-1} [I - \mathbf{K}]_{BA}] \\
&= \det[I - \mathbf{K}]_B \det[\mathbf{K}_{AA} + \mathbf{K}_{AB} [I - \mathbf{K}]_B^{-1} \mathbf{K}_{BA}].
\end{aligned}$$

Using Lemma 1.B.1.

Since A, B are disjoint, $I_{BA} = 0$.

With the same arguments, if $\mathbb{P}[A \subset \mathcal{X}] > 0$ then \mathbf{K}_A is invertible and (1.D.4) becomes

$$\begin{aligned}
\mathbb{P}[A \subset \mathcal{X}, \mathcal{X} \cap B = \emptyset] &= \det \begin{bmatrix} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ [I - \mathbf{K}]_{BA} & [I - \mathbf{K}]_{BB} \end{bmatrix} \\
&= \det \mathbf{K}_A \det[[I - \mathbf{K}]_{BB} - [I - \mathbf{K}]_{BA} \mathbf{K}_A^{-1} \mathbf{K}_{AB}] \\
&= \det \mathbf{K}_A \det[I - \mathbf{K} + \mathbf{K}_{:A} \mathbf{K}_A^{-1} \mathbf{K}_{A:}]_B.
\end{aligned}$$

Using Lemma 1.B.1.

Since A, B are disjoint, $I_{BA} = 0$.

□

Intuitively the inclusion-exclusion principle combines what happens in $\mathcal{X} \sim \text{DPP}(\mathbf{K})$ and $\mathcal{X}^c \sim \text{DPP}(I - \mathbf{K})$, that A is included and $B \subset A^c$ is excluded.

See also Launay, Galerne, and Desolneux (2018, Theorem 2) for a different approach.

By Theorem 1.D.1.

$$\mathbb{P}[S \subset \mathcal{X}] = \det \mathbf{K}_S.$$

By Theorem 1.B.2.

Several important corollaries follow from this inclusion-exclusion principle, like the expression of the joint probability densities of $\text{DPP}(\mathbf{K})$, and several stability properties of the DPP model.

By way of illustration, we use the margins to display the kernel of the DPPs resulting from the different transformations applied to $\mathcal{X} \sim \text{DPP}(\mathbf{K})$. The original correlation kernel \mathbf{K} is shown in Figure 1.D.1, and the correlation kernels resulting from the various transformation of \mathcal{X} are displayed on the same scale for comparison.

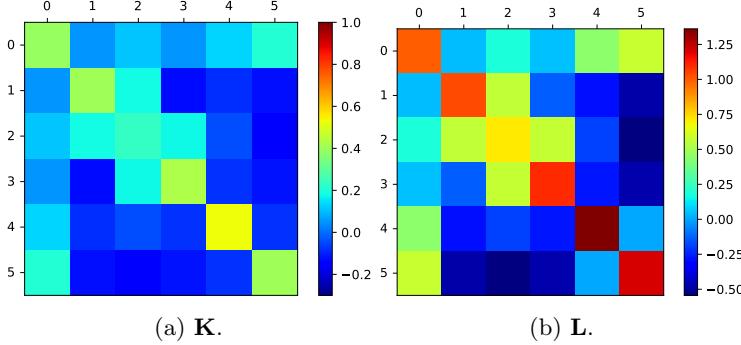


Figure 1.D.1: The two types of kernel associated to \mathcal{X} : (a) Correlation kernel \mathbf{K} , (b) Likelihood kernel $\mathbf{L} = \mathbf{K}(I - \mathbf{K})^{-1}$.

Corollary 1.D.3 (Likelihood of $\text{DPP}(\mathbf{K})$). *Let \mathbb{X} be finite and consider $\mathcal{X} \sim \text{DPP}(\mathbf{K})$. Then, for any $A \subset \mathbb{X}$ we have*

$$\begin{aligned}\mathbb{P}[\mathcal{X} = A] &= \det[I^A \mathbf{K} + I^{A^c} (I - \mathbf{K})] \\ &= (-1)^{|A^c|} \det[\mathbf{K} - I^{A^c}].\end{aligned}\quad (1.D.5)$$

When $\mathbb{P}[\mathcal{X} = \emptyset] > 0$, we have

$$\mathbb{P}[\mathcal{X} = A] = \det[I - \mathbf{K}] \det[\mathbf{K}(I - \mathbf{K})^{-1}]_A. \quad (1.D.6)$$

Proof. For any $A \subset \mathbb{X}$,

$$\begin{aligned}\mathbb{P}[\mathcal{X} = A] &= \mathbb{P}[A \subset \mathcal{X}, A^c \cap \mathcal{X} = \emptyset] \\ &= \det[I^A \mathbf{K} + I^{A^c} (I - \mathbf{K})]_{A \sqcup A^c} \\ &= \det[I^A \mathbf{K} + I^{A^c} (I - \mathbf{K})] \\ &= (-1)^{|A^c|} \det[I^A \mathbf{K} + I^{A^c} (\mathbf{K} - I)] \\ &= (-1)^{|A^c|} \det[\mathbf{K} - I^{A^c}]\end{aligned}\quad (1.D.7)$$

See also Kulesza and Taskar (2012, Section 3.5) Soshnikov (2000, Equation 1.36), Daley and Vere-Jones (2003, Exercise 5.4.7) or Macchi (1975, Theorem 12).

By Theorem 1.D.2,

In particular $\mathbb{P}[\mathcal{X} = \emptyset] = \det[I - \mathbf{K}]$.

If $\mathbb{P}[\mathcal{X} = \emptyset] = \det[I - \mathbf{K}] > 0$, then $I - \mathbf{K}$ is invertible and (1.D.7) becomes

$$\begin{aligned}\mathbb{P}[\mathcal{X} = A] &= \det[I - \mathbf{K}] \det[I^A \mathbf{K}(I - \mathbf{K})^{-1} + I^{A^c}] \\ &= \det[I - \mathbf{K}] \det[\mathbf{K}(I - \mathbf{K})^{-1}]_A.\end{aligned}$$

□

Corollary 1.D.4 (Complementary of a DPP). *Let \mathbb{X} be finite and consider $\mathcal{X} \sim \text{DPP}(\mathbf{K})$. Then the complementary process*

$$\mathcal{X}^c \triangleq \mathbb{X} \setminus \mathcal{X} \sim \text{DPP}(I - \mathbf{K}). \quad (1.D.8)$$

Proof. Let us compute the inclusion probabilities of the complementary process. Theorem 1.D.2 yields for any $A \subset \mathbb{X}$,

$$\begin{aligned} \mathbb{P}[A \subset \mathcal{X}^c] &= \mathbb{P}[\mathcal{X} \cap A = \emptyset] \\ &= \mathbb{P}[\emptyset \subset \mathcal{X}, \mathcal{X} \cap A = \emptyset] \\ &= \det[I^\emptyset \mathbf{K} + I^{\emptyset^c} (I - \mathbf{K})]_{\emptyset \sqcup A} \\ &= \det[I - \mathbf{K}]_A. \end{aligned}$$

□

Corollary 1.D.5 (DPP conditioned on $B \subset \mathcal{X}$). *Let \mathbb{X} be finite, and consider $\mathcal{X} \sim \text{DPP}(\mathbf{K})$. Then, for any $B \subset \mathbb{X}$ such that $\mathbb{P}[B \subset \mathcal{X}] > 0$, the conditioned process*

$$\mathcal{X} \mid B \subset \mathcal{X} \sim \text{DPP}(I^B + \mathbf{K} - \mathbf{K}_{:B} \mathbf{K}_B^{-1} \mathbf{K}_{B:}). \quad (1.D.9)$$

It could also be defined as the process $\text{DPP}(\mathbf{K} - \mathbf{K}_{:B} \mathbf{K}_B^{-1} \mathbf{K}_{B:})$ defined on $B^c = \mathbb{X} \setminus B$.

Proof. Let us compute the inclusion probabilities of this conditioned process. Since we assumed $\mathbb{P}[B \subset \mathcal{X}] = \det \mathbf{K}_B > 0$, then \mathbf{K}_B is invertible. For any $A \subset \mathbb{X}$,

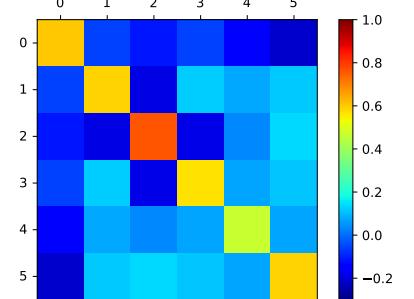
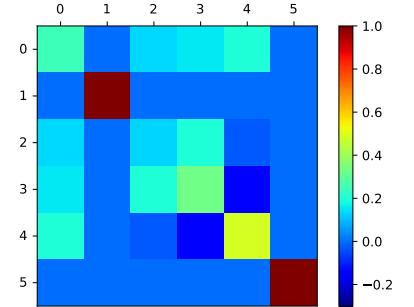
$$\begin{aligned} \mathbb{P}[A \subset \mathcal{X} \mid B \subset \mathcal{X}] &= \frac{\mathbb{P}[A \subset \mathcal{X}, B \subset \mathcal{X}]}{\mathbb{P}[B \subset \mathcal{X}]} \\ &= \frac{\mathbb{P}[(A \cap B^c) \sqcup B \subset \mathcal{X}]}{\mathbb{P}[B \subset \mathcal{X}]} = \frac{\det \mathbf{K}_{(A \cap B^c) \sqcup B}}{\det \mathbf{K}_B} \\ &= \frac{1}{\det \mathbf{K}_B} \det \begin{bmatrix} \mathbf{K}_{A \cap B^c} & \mathbf{K}_{A \cap B^c, B} \\ \mathbf{K}_{B, A \cap B^c} & \mathbf{K}_B \end{bmatrix} \\ &= \det \left[\mathbf{K}_{A \cap B^c} - \mathbf{K}_{A \cap B^c, B} [\mathbf{K}_B]^{-1} \mathbf{K}_{B, A \cap B^c} \right] \\ &= \det \left[\mathbf{K} - \mathbf{K}_{:B} [\mathbf{K}_B]^{-1} \mathbf{K}_{B:} \right]_{A \cap B^c} \\ &= \det \left[I^B + I^{B^c} \left[\mathbf{K} - \mathbf{K}_{:B} [\mathbf{K}_B]^{-1} \mathbf{K}_{B:} \right] I^{B^c} \right]_A \\ &= \det \left[I^B + \mathbf{K} - \mathbf{K}_{:B} [\mathbf{K}_B]^{-1} \mathbf{K}_{B:} \right]_A. \end{aligned}$$

□

Corollary 1.D.6 (DPP conditioned on $\mathcal{X} \cap B = \emptyset$). *Let \mathbb{X} be finite, and consider $\mathcal{X} \sim \text{DPP}(\mathbf{K})$. Then, for any $B \subset \mathbb{X}$ such that $\mathbb{P}[\mathcal{X} \cap B = \emptyset] > 0$, we have*

$$\mathcal{X} \mid B \cap \mathcal{X} = \emptyset \sim \text{DPP}\left(I^{B^c} \left[\mathbf{K} + \mathbf{K}_{:B} [I - \mathbf{K}]_B^{-1} \mathbf{K}_{B:} \right] I^{B^c}\right). \quad (1.D.10)$$

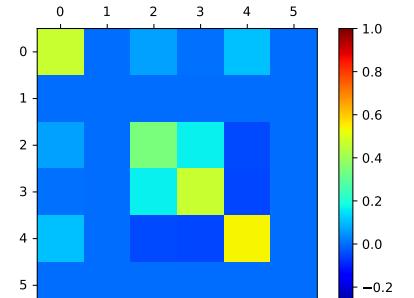
It could also be defined as the process $\text{DPP}(\mathbf{K} + \mathbf{K}_{:B} [I - \mathbf{K}]_B^{-1} \mathbf{K}_{B:})$ defined on $B^c = \mathbb{X} \setminus B$.

Figure 1.D.2: $\mathbb{X} \setminus \mathcal{X}$.Figure 1.D.3: $\mathcal{X} \mid \{1, 5\} \subset \mathcal{X}$.

By Definition 1.1.4.

Using Lemma 1.B.1.

Since $[\mathbf{K} - \mathbf{K}_{:B} [\mathbf{K}_B]^{-1} \mathbf{K}_{B:}]_B = 0$.

Figure 1.D.4: $\mathcal{X} \mid \{1, 5\} \cap \mathcal{X} = \emptyset$.

Proof. Let us compute the inclusion probabilities of this conditioned process. Since we assumed $\mathbb{P}[B \cap \mathcal{X} = \emptyset] = \det[I - \mathbf{K}]_B > 0$, then $[I - \mathbf{K}]_B$ is invertible. For any $A \subset \mathbb{X}$,

$$\begin{aligned}\mathbb{P}[A \subset \mathcal{X} \mid B \cap \mathcal{X} = \emptyset] &= \frac{\mathbb{P}[A \subset \mathcal{X}, B \cap \mathcal{X} = \emptyset]}{\mathbb{P}[B \cap \mathcal{X} = \emptyset]} \mathbf{1}_{A \subset B^c} \\ &= \frac{\mathbb{P}[A \cap B^c \subset \mathcal{X}, B \cap \mathcal{X} = \emptyset]}{\mathbb{P}[B \cap \mathcal{X} = \emptyset]} \mathbf{1}_{A \subset B^c} \\ &= \frac{\det[I^{A \cap B^c} \mathbf{K} + I^{(A \cap B^c)^c} (I - \mathbf{K})]_{(A \cap B^c) \cup B}}{\det[I - \mathbf{K}]_B} \mathbf{1}_{A \subset B^c} \quad \text{By Theorem 1.D.2.} \\ &= \frac{1}{\det[I - \mathbf{K}]_B} \det \begin{bmatrix} \mathbf{K}_{A \cap B^c} & \mathbf{K}_{A \cap B^c, B} \\ \mathbf{K}_{B, A \cap B^c} & [I - \mathbf{K}]_B \end{bmatrix} \mathbf{1}_{A \subset B^c} \quad \text{Since } I_{B, A \cap B^c} = 0. \\ &= \det \left[\mathbf{K}_{A \cap B^c} + \mathbf{K}_{A \cap B^c, B} [I - \mathbf{K}]_B^{-1} \mathbf{K}_{B, A \cap B^c} \right] \mathbf{1}_{A \subset B^c} \quad \text{Using Lemma 1.B.1.} \\ &= \det \left[\mathbf{K} + \mathbf{K}_{:B} [I - \mathbf{K}]_B^{-1} \mathbf{K}_{B,:} \right]_{A \cap B^c} \mathbf{1}_{A \subset B^c} \\ &= \det \left[I^{B^c} \left[\mathbf{K} + \mathbf{K}_{:B} [I - \mathbf{K}]_B^{-1} \mathbf{K}_{B,:} \right] I^{B^c} \right]_A.\end{aligned}$$

□

Lemma 1.D.7 (DPP union a subset B). *Let $B \subset \mathbb{X}$ and consider $\mathcal{X} \sim \text{DPP}(\mathbf{K})$. Then the union process*

$$\mathcal{X} \cup B \sim \text{DPP}(I^B + I^{B^c} \mathbf{K} I^{B^c}). \quad (1.D.11)$$

It could also be defined as the process $\text{DPP}(\mathbf{K}_{B^c})$ defined on $B^c = \mathbb{X} \setminus B$.

Proof. Let us compute the inclusion probabilities of the union process $\mathcal{X} \cup B$. For $A \subset \mathbb{X}$, we have

$$\begin{aligned}\mathbb{P}[A \subset \mathcal{X} \cup B] &= \mathbb{P}[A \cap B^c \subset \mathcal{X}] \\ &= \det \mathbf{K}_{A \cap B^c} \\ &= \det \left[I^B + I^{B^c} \mathbf{K} I^{B^c} \right]_A.\end{aligned}$$

□

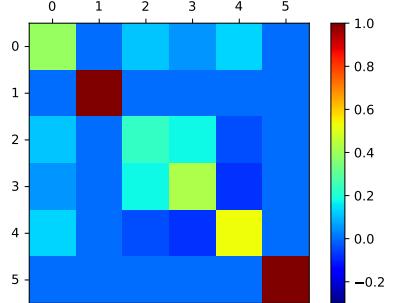


Figure 1.D.5: $\mathcal{X} \cup \{1, 5\}$.

Lemma 1.D.8 (DPP restricted to a subset B). *Let $B \subset \mathbb{X}$ and consider $\mathcal{X} \sim \text{DPP}(\mathbf{K})$. Then the restricted process*

$$\mathcal{X} \cap B \sim \text{DPP}(I^B \mathbf{K} I^B). \quad (1.D.12)$$

It could also be defined as the process $\text{DPP}(\mathbf{K}_B)$ defined on $B \subset \mathbb{X}$.

Proof. Let us compute the inclusion probabilities of the restricted process $\mathcal{X} \cap B$. For $A \subset \mathbb{X}$, we have

$$\begin{aligned}\mathbb{P}[A \subset \mathcal{X} \cap B] &= \mathbb{P}[A \subset \mathcal{X}] \mathbf{1}_{A \subset B} \\ &= \det K_A \mathbf{1}_{A \subset B} \\ &= \det [I^B \mathbf{K} I^B]_A.\end{aligned}$$

□

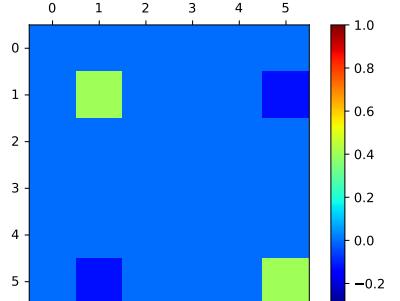


Figure 1.D.6: $\mathcal{X} \cap \{1, 5\}$.

We finally derive a non-symmetric DPP! by considering the symmetric difference between a DPP and a given set B .

Corollary 1.D.9 (Symmetric difference with a subset B). *Let \mathbb{X} be finite, consider $\mathcal{X} \sim \text{DPP}(\mathbf{K})$ and take $B \subset \mathbb{X}$. Then the symmetric difference process $\mathcal{X}\Delta B \triangleq (\mathcal{X} \cup B) \setminus (\mathcal{X} \cap B)$ is a non-symmetric DPP,*

$$\mathcal{X}\Delta B \sim \text{DPP}(I^{B^c}\mathbf{K} + I^B(I - \mathbf{K})). \quad (1.D.13)$$

Proof. Let us compute the inclusion probabilities of this symmetric difference process. For any $A \subset \mathbb{X}$,

$$\begin{aligned} & \mathbb{P}[A \subset \mathcal{X}\Delta B] \\ &= \mathbb{P}[A \cap B^c \subset \mathcal{X}, (A \cap B) \cap \mathcal{X} = \emptyset] \\ &= \det[I^{A \cap B^c}\mathbf{K} + I^{A \cap B}(I - \mathbf{K})]_{(A \cap B^c) \sqcup (A \cap B)} \quad \text{By Theorem 1.D.2.} \\ &= \det[I^{A \cap B^c}\mathbf{K} + I^{A \cap B}(I - \mathbf{K})]_A \\ &= \det[I^{B^c}\mathbf{K} + I^B(I - \mathbf{K})]_A. \end{aligned}$$

□

See also Borodin, Okounkov, and Olshanski (2000, Proposition A.8).

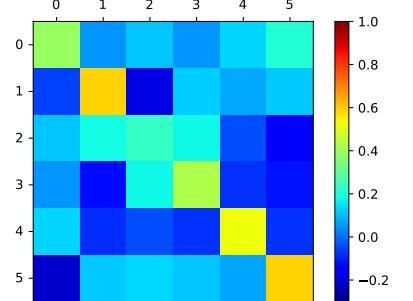


Figure 1.D.7: $\mathcal{X}\Delta\{1, 5\}$.

1.E EXPECTATION AND VARIANCE OF LINEAR STATISTICS

The computation of expectations and variances of linear statistics $\sum_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$ of a process \mathcal{X} may appear to be useful in various contexts. For example, when f corresponds to the indicator function of a given region, we could express analytically the expectation and the variance of a number of points falling in this region. In Chapter 4, we notably consider approximating integrals $\int f(x)\mu(dx)$ with a Monte Carlo estimator, which expresses as a linear statistic of the DPP. In particular, the bias of an estimator is obtained by the computation of its expectation, and its variance gives non-asymptotic guarantees on the quality of the estimation.

Proposition 1.E.1.

Let f be a suitable test function and consider $\mathcal{X} \sim \text{DPP}(\mu, K)$.

Then, we have

$$\mathbb{E} \left[\sum_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \right] = \int_{\mathbb{X}} f(x)K(x, x)\mu(dx) \quad (1.E.1)$$

$$\mathbb{V}\text{ar} \left[\sum_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \right] = \int_{\mathbb{X}} f(x)^2 K(x, x)\mu(dx) - \int_{\mathbb{X}^2} f(x)f(y)K(x, y)K(y, x)\mu(dx)\mu(dy). \quad (1.E.2)$$

If K is Hermitian, i.e., $K(y, x) = \overline{K(x, y)}$,

$$\mathbb{V}\text{ar} \left[\sum_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \right] = \int_{\mathbb{X}^2} f(x)^2 K(x, x)\mu(dx) - \int_{\mathbb{X}} f(x)f(y)|K(x, y)|^2\mu(dx)\mu(dy). \quad (1.E.3)$$

If K satisfies (1.1.12), in particular $K(x, x) = \int_{\mathbb{X}} K(x, y)K(y, x)\mu(dy)$,

$$\mathbb{V}\text{ar} \left[\sum_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \right] = \frac{1}{2} \int_{\mathbb{X}^2} [f(x) - f(y)]^2 K(x, y)K(y, x)\mu(dx)\mu(dy). \quad (1.E.4)$$

Proof of Proposition 1.E.1. By definition of the first order correlation function of $\text{DPP}(\mu, K)$ we have

$$\mathbb{E}\left[\sum_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})\right] = \int_{\mathbb{X}} f(x)\rho_1(x)\mu(dx) = \int_{\mathbb{X}} f(x)K(x, x)\mu(dx). \quad (1.E.5)$$

Then, for any point process \mathcal{X} which admits ρ_1 and ρ_2 as its first two correlation functions we have

$$\begin{aligned} \text{Var}\left[\sum_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})\right] &= \mathbb{E}\left[\sum_{\mathbf{x} \neq \mathbf{y} \in \mathcal{X}} f(\mathbf{x})f(\mathbf{y}) + \sum_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})^2\right] - \left(\mathbb{E}\left[\sum_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})\right]\right)^2 \\ &= \int_{\mathbb{X}^2} f(x)f(y)\rho_2(x, y)\mu(dx)\mu(dy) \\ &\quad + \int_{\mathbb{X}} f(x)^2\rho_1(x)\mu(dx) - \left(\int_{\mathbb{X}} f(x)\rho_1(x)\mu(dx)\right)^2. \end{aligned} \quad (1.E.6)$$

When $\rho_2(x, y) = \rho_1(x)\rho_1(y)$, e.g., for the Poisson point process, we have $\text{Var}[\sum_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})] = \int_{\mathbb{X}} f(x)^2\rho_1(x)\mu(dx)$.

Considering $X \sim \text{DPP}(\mu, K)$,

$$\begin{aligned} (1.E.6) &= \int_{\mathbb{X}} f(x)f(y)[\cancel{K(x, x)}\cancel{K(y, y)} - K(x, y)K(y, x)]\mu(dx)\mu(dy). & \rho_2(x, y) &= \det\begin{bmatrix} K(x, x) & K(x, y) \\ K(y, x) & K(y, y) \end{bmatrix}. \\ &\quad + \int_{\mathbb{X}} f(x)^2K(x, x)\mu(dx) - \left(\int_{\mathbb{X}} f(x)\cancel{K(x, x)}\mu(dx)\right)^2 & \rho_1(x) &= K(x, x). \\ &= \int_{\mathbb{X}} f(x)^2K(x, x)\mu(dx) - \int_{\mathbb{X}^2} f(x)f(y)K(x, y)K(y, x)\mu(dx)\mu(dy). \end{aligned} \quad (1.E.7)$$

If the kernel K is Hermitian, i.e., $K(y, x) = \overline{K(x, y)}$,

$$(1.E.7) = \int_{\mathbb{X}^2} f(x)^2K(x, x)\mu(dx) - \int_{\mathbb{X}} f(x)f(y)|K(x, y)|^2\mu(dx)\mu(dy).$$

For $f \geq 0$, there is variance reduction in the sense $\text{Var}_{\text{DPP}} < \text{Var}_{\text{Poisson}}$.

If the kernel K satisfies (1.1.12), then $K(x, x) = \int_{\mathbb{X}} K(x, y)K(y, x)\mu(dy)$ and we have

$$\begin{aligned} (1.E.7) &= \int_{\mathbb{X}} f(x)^2\left(\int_{\mathbb{X}} K(x, y)K(y, x)\mu(dy)\right)\mu(dx) \\ &\quad - \int_{\mathbb{X}^2} f(x)f(y)K(x, y)K(y, x)\mu(dx)\mu(dy) \\ &= \int_{\mathbb{X}^2} \left(\frac{f(x)^2 + f(y)^2}{2}K(x, y)K(y, x) - f(x)f(y)K(x, y)K(y, x)\right)\mu(dx)\mu(dy) \\ &= \frac{1}{2} \int_{\mathbb{X}^2} [f(x) - f(y)]^2K(x, y)K(y, x)\mu(dx)\mu(dy). \end{aligned} \quad (1.E.8)$$

□

Exact DPP sampling

2

Exact sampling methods guarantee that the statistical properties of the random cloud of points or the random subset of items generated by the procedure match with the ones prescribed by target DPP model.

On the theoretical side, generating exact samples is usually used as a way to illustrate or validate empirically some results but also to make new conjectures. In this setting, the efficiency of the sampling method may not be the primary concern but generating exact samples is key for the theoretical guarantees of the approach to hold.

From a more application oriented perspective, finite DPP samples are used, e.g., in a recommendation system to make diverse recommendations of items or to generate summaries of large text corpus, see, e.g., Wilhelm et al. (2018); Warlop (2018, Chapter 4); Kulesza and Taskar (2012). This requires efficient sampling methods scaling both with the number of items expected and the size of dataset.

In the context of Monte Carlo integration, DPP samples can be used to construct estimators of the integral of a function of interest, cf. Chapter 4. In order to make this approach applicable, the need for exact sampling procedures scaling with the number of sample points and the ambient dimension becomes critical.

The original exact sampling procedure was devised by Hough et al. (2006, Algorithm 18) to generate samples from orthogonal projection DPPs. It can serve as a core routine for sampling Hermitian DPPs given the eigendecomposition of the kernel.

In this chapter, we review the different exact sampling methods for continuous and finite DPPs. We start with the case where \mathbb{X} is continuous, e.g., $[-1, 1]^d, \mathbb{R}^d$, the unit circle, etc. Then, we specialize the sampling schemes to the finite case which is the most prominent use-case in the ML literature. As in the previous Section 1.2 we emphasize the distinction between projection and general DPPs.

Note that we intentionally avoid to deal with the so-called k -DPPs, which can be understood as DPPs conditioned to have exactly k points (Kulesza and Taskar, 2011). The main reason is because k -DPPs are not DPPs in general; the only intersection between DPPs and k -DPPs is when the underlying kernel is a projection DPP and k is equal to the rank of the kernel. The second reason comes from the fact that sampling schemes for DPPs can be adapted to sample from k -DPPs. We mention that, when the kernel is indeed of projection type, and only in that case, sampling exactly from the corresponding k -DPP amounts to performing the first k steps of the classical routines for sampling the associated projection DPP.

A substantial part of the material presented in this chapter is available in the documentation  of the DPPy Python toolbox , where the text is illustrated with practical implementations using DPPy objects.

2.1 Sampling from projection DPPs	42
The continuous case	
The finite case	
2.2 Exact sampling from non-projection DPPs	48
Finite DPPs defined by their correlation kernel	
Finite DPPs defined by their likelihood kernel	
The continuous case	
APPENDICES	
2.A Specialization of the sequential sampler for Hermitian DPPs	58
2.B A note on the sequential thinning procedure	58

2.1 SAMPLING FROM PROJECTION DPPs

The motivation for this section is twofold. First, we present generic sampling methods that give a unifying view on the potentially different models they come from. Second, recalling that Hermitian DPPs can be seen as a mixture of orthogonal projection DPPs, cf. Theorem 1.2.4, sampling methods for orthogonal projection DPPs serve the more general purpose of sampling Hermitian DPPs, cf. Section 2.2.

In this section we lay the ground material for projection DPP sampling and emphasize the geometrical interpretation of the generic sampling scheme for orthogonal projection DPPs.

2.1.1 The continuous case

Apart from some specific instances like the projection DPPs arising in random matrix theory as the eigenvalues of random matrices, the usual sampling scheme for continuous projection DPPs relies on the chain rule. Each of the N points forming the resulting sample $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ is drawn sequentially conditionally on the previously selected points so that $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ has distribution (1.1.15). The fact that the order the points were selected does not matter comes from the fact that the joint distribution of the points is invariant by permutation of its coordinates, i.e., the variables $\mathbf{x}_1, \dots, \mathbf{x}_N$ are exchangeable, see also Proposition 1.1.8.

PROJECTION DPP SAMPLING CAN BE DONE USING THE CHAIN RULE. Given an oracle to evaluate the kernel $K(x, y)$ and sample from the successive conditional distributions.

Proposition 2.1.1 (Projection DPP sampling given the kernel).

To generate a sample $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ from a projection DPP(μ, K), as defined by Proposition 1.1.8, it is sufficient to sequentially sample $\mathbf{x}_1, \dots, \mathbf{x}_N$ using the following chain rule scheme and forget the order the points were selected.

For $n = 1$, sample \mathbf{x}_1 from

$$\frac{1}{N} K(x, x) \mu(dx). \quad (2.1.1)$$

For $2 \leq n \leq N$, sample $\mathbf{x}_n \mid \mathbf{x}_1, \dots, \mathbf{x}_{n-1}$ from¹

$$\frac{K(x, x) - K(x, \mathbf{x}_{1:n-1}) K(\mathbf{x}_{1:n-1}, \mathbf{x}_{1:n-1})^{-1} K(\mathbf{x}_{1:n-1}, x)}{N - (n - 1)} \mu(dx). \quad (2.1.2)$$

¹ Computationally, updating of the inverse of the growing matrix involved in (2.1.2) can be done using the LDU factorization (1.B.2).

Proof. We build on the elements of the proof of the construction of projection DPPs given in Proposition 1.1.8. Assuming the matrix $K(\mathbf{x}_{1:n-1}, \mathbf{x}_{1:n-1})$ is indeed invertible we can use Lemma 1.B.1 to rewrite (2.1.2) as

$$\frac{\det \begin{bmatrix} [K(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^{n-1} & K(\mathbf{x}_{1:n-1}, x) \\ K(x, \mathbf{x}_{1:n-1}) & K(x, x) \end{bmatrix}}{(N - (n - 1)) \det [K(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^{n-1}} \mu(dx). \quad (2.1.3)$$

To prove the validity of the chain rule, it is enough to show that the marginal (2.1.1) and conditional densities (2.1.2) (w.r.t. μ) are well defined probability distributions, since the telescopic product (2.1.1) $\times \prod_{n=2}^N$ (2.1.3) = (1.1.15) yields $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ with the right distribution. The marginal (2.1.1) and conditional densities (2.1.3) are indeed nonnegative by (1.1.18) and integrate to one by (1.A.3).

The sequential nature of the chain rule makes the ratio (2.1.3) well defined. Indeed, after sampling \mathbf{x}_1 according to (2.1.1) we have $K(\mathbf{x}_1, \mathbf{x}_1) > 0, \mu$ -almost surely. This makes $\mathbf{x}_2 | \mathbf{x}_1$ given by (2.1.2) well defined. Thus, after sampling from this conditional, $(\mathbf{x}_1, \mathbf{x}_2)$ has joint density proportional to $\det[K(x_i, x_j)]_{i,j=1}^2$. A simple recursion shows that after the first $n - 1$ steps of the chain rule, $(\mathbf{x}_1, \dots, \mathbf{x}_{n-1})$ has joint density proportional to $\det[K(x_i, x_j)]_{i,j=1}^{n-1}$ as in (1.1.16) so that $\det[K(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^{n-1} > 0, \mu$ -almost surely, which makes (2.1.2) well defined. At the end of the day, the chain rule is valid and generates $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ with the right distribution (1.1.15). \square

OVERALL, THE MAIN DIFFICULTY WHEN USING THE CHAIN RULE FOR PROJECTION DPP SAMPLING IS TO FIND EFFICIENT WAYS TO SAMPLE FROM THE CONDITIONALS. Typically, the cost of sampling from a projection DPP with N points in dimension d using the chain rule is of order $\mathcal{O}(N^3)$. The dependence in the dimension hides in the evaluation of $K(x, y)$ or the feature vector $\Phi(x)$ when K is Hermitian, but also in the sampling of the conditionals. As we will see below, when the kernel is assumed Hermitian, there is a direct way of bounding the density of the conditionals which suggests using rejection sampling. In the one-dimensional case, we note that there are some specific orthogonal projection DPPs related to the eigenvalues of random matrices that can be sampled without rejection sampling in $\mathcal{O}(N^2)$, cf. Chapter 5.

When the kernel K is assumed Hermitian, K becomes an orthogonal projection kernel. The corresponding $DPP(\mu, K)$ is usually called an orthogonal projection DPP, but it is also referred to as an elementary DPP, e.g., in the machine learning literature (Kulesza and Taskar, 2012). The latter terminology highlights the special role they play. From a constructive perspective they can be understood as the building blocks of more general Hermitian DPPs.² And in terms of sampling, they carry an additional enjoyable geometrical interpretation which we present in the remaining part of the section.

THE CHAIN RULE APPLIED TO SAMPLE FROM ORTHOGONAL PROJECTION DPP HAS A NATURAL GEOMETRICAL INTERPRETATION. In this case, the kernel has a Gram formulation,³ thus kernel evaluations can be written as an inner product either in a functional space $K(x, y) = \langle \psi(x), \psi(y) \rangle_{L^2(\mu)}$ where $\psi(x) = K(x, \cdot)$ or in a finite dimensional space $K(x, y) = \Phi(y)^H \Phi(x)$ with $\Phi(x) = (\phi_1(x), \dots, \phi_N(x))$ where ϕ_n corresponds to the n -th eigenfunction of K . In this setting, the conditionals (2.1.2) can be expressed as a ratio of determinants of Gram matrices (2.1.3) which in turn reads as a

² See Hermitian DPPs as mixtures of orthogonal projection DPPs (Theorem 1.2.4).

³ In the discrete case, we have $\mathbf{K}^2 = \mathbf{K}$ and $\mathbf{K}^H = \mathbf{K}$ so that $\mathbf{K} = \mathbf{K}^H \mathbf{K}$, see also Section 2.1.2.

squared distance. In the end the likelihood of the resulting configuration of points $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ is proportional to the squared volume of the parallelopope spanned by the corresponding feature vectors. Proposition 2.1.2 treats the case $K(x, y) = \Phi(y)^H \Phi(x)$.

Proposition 2.1.2 (Orthogonal projection DPP sampling given the eigendecomposition of the kernel). *To generate a sample $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ from an orthogonal projection DPP(μ, K), with kernel given by*

$$K(x, y) = \sum_{k=1}^N \phi_k(x) \overline{\phi_k(y)}, \text{ with } \int_{\mathbb{X}} \phi_k(z) \overline{\phi_\ell(z)} \mu(dz) = \delta_{k\ell}, \quad (2.1.4)$$

the formulation (2.1.2) can be exploited further to express $K(x, y)$ as an inner product in \mathbb{C}^N : for all $x, y \in \mathbb{X}$,

$$K(x, y) = \Phi(y)^H \Phi(x), \text{ with } \Phi(x) \triangleq (\phi_1(x), \dots, \phi_N(x))^\top. \quad (2.1.5)$$

As a consequence, the joint distribution (1.1.15) of $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ reads⁴

$$\frac{1}{N!} \det[\Phi(x_j)^H \Phi(x_i)]_{i,j=1}^N \mu^{\otimes N}(dx_1, \dots, dx_N) \quad (2.1.6)$$

$$= \frac{1}{N!} \text{volume}^2\{\Phi(x_1), \dots, \Phi(x_N)\} \mu^{\otimes N}(dx_1, \dots, dx_N), \quad (2.1.7)$$

and the chain rule takes the following form.

For $n = 1$, sample \mathbf{x}_1 from

$$\frac{1}{N} K(x, x) \mu(dx) = \frac{1}{N} \|\Phi(x)\|^2 \mu(dx). \quad (2.1.8)$$

For $2 \leq n \leq N$, sample $\mathbf{x}_n | \mathbf{x}_1, \dots, \mathbf{x}_{n-1}$ from

$$\frac{K(x, x) - K(\mathbf{x}_{1:n-1}, x)^H K(\mathbf{x}_{1:n-1}, \mathbf{x}_{1:n-1})^{-1} K(\mathbf{x}_{1:n-1}, x)}{N - (n-1)} \mu(dx) \quad (2.1.9)$$

$$= \frac{\det \begin{bmatrix} [\Phi(x_j)^H \Phi(x_i)]_{i,j=1}^{n-1} & [\Phi(x)^H \Phi(x_i)]_{i=1}^{n-1} \\ \left[\overline{\Phi(x)^H \Phi(x_j)} \right]_{j=1}^{n-1} & \langle \Phi(x)^H \Phi(x) \rangle \end{bmatrix}}{(N - (n-1)) \det[\Phi(x_j)^H \Phi(x_i)]_{i,j=1}^{n-1}} \mu(dx) \quad (2.1.10)$$

$$= \frac{\text{distance}^2(\Phi(x), \text{span}\{\Phi(\mathbf{x}_1), \dots, \Phi(\mathbf{x}_{n-1})\})}{N - (n-1)} \mu(dx). \quad (2.1.11)$$

The chain rule has a strong geometrical flavor reflected by the base \times height formula (2.1.7) = (2.1.8) \times $\prod_{n=2}^N$ (2.1.11) corresponding to the sequential Gram-Schmidt orthogonalization of the feature vectors $\Phi(\mathbf{x}_1), \dots, \Phi(\mathbf{x}_N)$.

As mentioned above, given the formulation (2.1.11) of the conditional densities, the squared distance in the numerator can always be bounded by $K(x, x)$, i.e., the numerator of the marginal density. To see this take (2.1.11) and write

$$\text{distance}^2(\Phi(x), \text{span}\{\Phi(\mathbf{x}_1), \dots, \Phi(\mathbf{x}_{n-1})\}) \leq \|\Phi(x)\|^2 = K(x, x). \quad (2.1.12)$$

This domination of the conditional densities by the marginal density suggests deriving the chain rule with a rejection sampling mechanism to sample each conditional in turn, with the same proposal distribution $N^{-1} K(x, x) \mu(dx)$. We give more details on this perspective in Chapter 4, where we use a special orthogonal projection DPP in the context of Monte Carlo integration.

⁴ The larger the volume of the parallelopope spanned by $\Phi(x_1), \dots, \Phi(x_N) \in \mathbb{C}^N$, the more likely x_1, \dots, x_N co-occur.

Note that the numerator corresponds to the incremental posterior variance of a noise-free Gaussian process model with kernel K (Rasmussen and Williams, 2006), giving yet another intuition for repulsion.

Observe that if $\Phi(x) \in \text{span}\{\Phi(\mathbf{x}_1), \dots, \Phi(\mathbf{x}_{n-1})\}$ the point x will not be sampled since the square distance (2.1.11) becomes zero.

2.1.2 The finite case

The main difference from the previous section is that the state space \mathbb{X} is now finite and sampling from the conditionals boils down to sampling from a probability vector. We detail only the case where \mathbf{K} is an orthogonal projection kernel. For non-orthogonal projection kernels, one can use Proposition 2.1.1 and update the conditionals (2.1.2) based on the LDU factorization (1.B.2) of the matrix involved in (2.1.2). At iteration n this update can be done in time $\mathcal{O}(n^2)$.

Proposition 2.1.3. *Let $\mathbb{X} = \{1, \dots, M\}$. To generate a sample $\{\mathbf{j}_1, \dots, \mathbf{j}_N\}$ from an orthogonal projection DPP($\boldsymbol{\omega}, \mathbf{K}$), one can use the fact that the kernel factorizes as*

$$\mathbf{K} = \mathbf{K}\mathbf{K}^\text{H} = \mathbf{K}^\text{H}\mathbf{K}, \quad \text{i.e., } \mathbf{K}_{ij} = \mathbf{K}_{i:}\mathbf{K}_{j:}^\text{H} = \mathbf{K}_{:i}^\text{H}\mathbf{K}_{:j}. \quad (2.1.13)$$

As a consequence, the joint distribution (1.1.15) of $(\mathbf{j}_1, \dots, \mathbf{j}_N)$ reads⁵

$$\begin{aligned} \frac{1}{N!} \det[\mathbf{K}_{\mathbf{j}_k:\mathbf{K}_{\mathbf{j}_\ell:}^\text{H}}]_{k,\ell=1}^N \prod_{n=1}^N \boldsymbol{\omega}_n &= \frac{1}{N!} \text{volume}^2[\mathbf{K}_{\mathbf{j}_1:}, \dots, \mathbf{K}_{\mathbf{j}_N:}] \prod_{n=1}^N \boldsymbol{\omega}_n \\ \frac{1}{N!} \det[\mathbf{K}_{:\mathbf{j}_k}^\text{H}\mathbf{K}_{:\mathbf{j}_\ell}]_{k,\ell=1}^N \prod_{n=1}^N \boldsymbol{\omega}_n &= \frac{1}{N!} \text{volume}^2[\mathbf{K}_{:\mathbf{j}_1}, \dots, \mathbf{K}_{:\mathbf{j}_N}] \prod_{n=1}^N \boldsymbol{\omega}_n \end{aligned} \quad (2.1.14)$$

and the chain rule takes the following form.

For $n = 1$, sample \mathbf{j}_1 from

$$\frac{1}{N} \mathbf{K}_{jj} \boldsymbol{\omega}_j = \frac{1}{N} \|\mathbf{K}_{j:}\|^2 = \frac{1}{N} \|\mathbf{K}_{:j}\|^2 \boldsymbol{\omega}_j. \quad (2.1.15)$$

For $2 \leq n \leq N$, note $J_{n-1} \triangleq \{\mathbf{j}_1, \dots, \mathbf{j}_{n-1}\}$ and sample $\mathbf{j}_n \mid \mathbf{j}_1, \dots, \mathbf{j}_{n-1}$ from

$$\begin{aligned} \frac{\mathbf{K}_{jj} - \mathbf{K}_{jJ_{n-1}} \mathbf{K}_{J_{n-1}}^{-1} \mathbf{K}_{J_{n-1}:}^\text{H}}{N - (n-1)} \boldsymbol{\omega}_j &= \frac{\text{distance}^2(\mathbf{K}_{j:}, \text{span}\{\mathbf{K}_{\mathbf{j}_1:}, \dots, \mathbf{K}_{\mathbf{j}_{n-1}:}\})}{N - (n-1)} \boldsymbol{\omega}_j \\ \frac{\mathbf{K}_{jj} - \mathbf{K}_{J_{n-1}j}^\text{H} \mathbf{K}_{J_{n-1}}^{-1} \mathbf{K}_{J_{n-1}j}}{N - (n-1)} \boldsymbol{\omega}_j &= \frac{\text{distance}^2(\mathbf{K}_{:j}, \text{span}\{\mathbf{K}_{:\mathbf{j}_1}, \dots, \mathbf{K}_{:\mathbf{j}_{n-1}}\})}{N - (n-1)} \boldsymbol{\omega}_j. \end{aligned} \quad (2.1.16)$$

When \mathbf{K} is an orthogonal projection kernel with $\text{rank}(\mathbf{K}) = N$, the eigenfactorization $\mathbf{K} = UU^\text{H}$ (where $U \in \mathbb{C}^{M \times N}$ with $U^\text{H}U = I_N$) provides another way to write the chain rule.

Proposition 2.1.4. *Let $\mathbb{X} = \{1, \dots, M\}$. To generate a sample $\{\mathbf{j}_1, \dots, \mathbf{j}_N\}$ from an orthogonal projection DPP($\boldsymbol{\omega}, \mathbf{K}$), given the eigen-decomposition*

$$\mathbf{K} = UU^\text{H} = \sum_{n=1}^N U_{:n} U_n^\text{H}, \quad \text{with } U^\text{H} \text{diag}(\boldsymbol{\omega}) U = I_N, \quad (2.1.17)$$

As a consequence, the joint distribution (1.1.15) of $(\mathbf{j}_1, \dots, \mathbf{j}_N)$ reads⁶

$$\frac{1}{N!} \det[U_{\mathbf{j}_k:\mathbf{U}_{\mathbf{j}_\ell:}^\text{H}}]_{k,\ell=1}^N \prod_{n=1}^N \boldsymbol{\omega}_n = \frac{1}{N!} \text{volume}^2[U_{\mathbf{j}_1:}, \dots, U_{\mathbf{j}_N:}] \prod_{n=1}^N \boldsymbol{\omega}_n \quad (2.1.18)$$

⁵The larger the volume of the parallelopiped spanned by $\mathbf{K}_{:j_1}, \dots, \mathbf{K}_{:j_N} \in \mathbb{R}^M$, the more likely j_1, \dots, j_N co-occur.

Observe that if $j \in \{\mathbf{j}_1, \dots, \mathbf{j}_{n-1}\}$ it cannot be sampled again since the square distance (2.1.16) becomes zero.

⁶The larger the volume of the parallelopiped spanned by $U_{j_1:}, \dots, U_{j_N:} \in \mathbb{R}^N$, the more likely j_1, \dots, j_N co-occur.

Algorithm 1: Generate a sample \mathbf{X} from an *orthogonal projection DPP(\mathbf{K})* with $\text{rank}(\mathbf{K}) = N$, given a factor U such that $\mathbf{K} = UU^H$.

Require: Factor U of the decomposition $\mathbf{K} = UU^H$.

```

1:  $\mathbf{X}, \mathbf{A} = \emptyset, \{0, \dots, M - 1\}$ 
2:  $\mathbf{C} = \text{zeros}(M, N)$ 
3:  $d = \begin{cases} \text{diagonal}(\mathbf{K}) & \text{if } U = \mathbf{K} \\ (\|U[0,:]\|^2, \dots, \|U[M-1,:]\|^2) & \text{otherwise} \end{cases}$ 
4: for  $n$  in range( $N$ ) do
5:   draw  $j$  from  $\mathbf{A}$  with probability  $d[\mathbf{A}]$ 
6:    $\mathbf{X}, \mathbf{A} = \mathbf{X} \cup \{j\}, \mathbf{A} \setminus \{j\}$ 
7:    $C[\mathbf{A}, n] = \begin{cases} \mathbf{K}[\mathbf{A}, j] & \text{if } U = \mathbf{K} \\ U[\mathbf{A}, :]U[j, :]^H & \text{otherwise} \end{cases} - C[\mathbf{A}, :n] C[j, :n]^H$ 
8:    $C[\mathbf{A}, n] /= \sqrt{d[j]}$ 
9:    $d[\mathbf{A}] -= |C[\mathbf{A}, n]|^2$ 
10: end for
11: return  $\mathbf{X}$   $\# \prod d[\mathbf{X}] = \mathbb{P}[\mathcal{X} = \mathbf{X}]$ 

```

and the chain rule takes the following form.

For $n = 1$, sample \mathbf{j}_1 from

$$\frac{1}{N} \mathbf{K}_{jj} \boldsymbol{\omega}_j = \frac{1}{N} \|U_{j,:}\|^2 = \frac{1}{N} \|U_{:,j}\|^2 \boldsymbol{\omega}_j. \quad (2.1.19)$$

For $2 \leq n \leq N$, note $J_{n-1} \triangleq \{\mathbf{j}_1, \dots, \mathbf{j}_{n-1}\}$ and sample $\mathbf{j}_n \mid \mathbf{j}_1, \dots, \mathbf{j}_{n-1}$ from

$$\frac{\mathbf{K}_{jj} - \mathbf{K}_{jJ_{n-1}} \mathbf{K}_{J_{n-1}}^{-1} \mathbf{K}_{J_{n-1}}^H \mathbf{K}_{jJ_{n-1}}}{N - (n - 1)} \boldsymbol{\omega}_j = \frac{\text{distance}^2(U_{j,:}, \text{span}\{U_{\mathbf{j}_1,:}, \dots, U_{\mathbf{j}_{n-1},:}\})}{N - (n - 1)} \boldsymbol{\omega}_j \quad (2.1.20)$$

The geometrical perspective of the chain rule expressed in Proposition 2.1.3, resp. Proposition 2.1.4, allows us to view orthogonal projection DPP sampling as a sequential Gram-Schmidt orthogonalization of the rows of \mathbf{K} , resp. the rows of the factor U in the decomposition $\mathbf{K} = UU^H$. The matrix U usually corresponds to the matrix of eigenvectors of the correlation kernel, w.r.t. to nonzero eigenvalues. For a practical implementation we refer to Algorithm 1 which can be understood as a combination of the views of Gillenwater (2014, Algorithms 2), Poulson (2019, Algorithm 3) and Tremblay, Barthelme, and Ambard (2018, Algorithm 3).

However, when the size of the ground set $|\mathbb{X}| = M$ is too large but the size of the samples is relatively small $N \ll M$,⁷ the $\mathcal{O}(MN)$ cost required by Algorithm 1 (see 1.7-9) to update the size M vector describing the n -th conditional probabilities (2.1.20) becomes prohibitive. This has motivated research to find alternative methods scaling sub-linearly with the total number M of items. In the following we present two very different works which go in this direction. The first, due to Gillenwater et al. (2019), applies to orthogonal projection DPPs while the second, due to Dereziński, Calandriello, and Valko (2019) applies more generally to symmetric DPP(\mathbf{L}).

Observe that if $j \in \{\mathbf{j}_1, \dots, \mathbf{j}_{n-1}\}$ it cannot be sampled again since the square distance (2.1.20) becomes zero.

⁷ This is usually a practical case in recommendation systems where the catalog contains millions of items but only a few of them are proposed to the user.

SAMPLING FROM THE CONDITIONALS DRIVING THE CHAIN RULE CAN BE DONE THROUGH THE EXPLORATION OF A BINARY TREE WITH A LOGARITHMIC DEPENDENCY IN THE TOTAL NUMBER OF ITEMS. Given the eigendecomposition $\mathbf{K} = UU^\top$, where $U \in \mathbb{R}^{M \times N}$, Gillenwater et al. (2019) proposed a binary tree structure storing appropriate summary statistics of the eigenvectors to generate repeated samples from orthogonal projection DPPs in time $\mathcal{O}(\log(M)N^4)$.

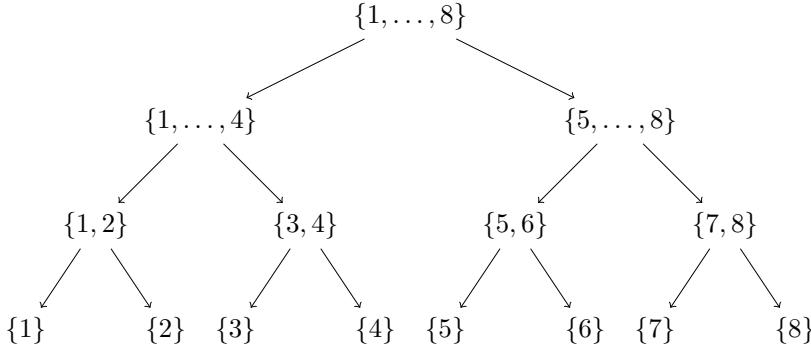


Figure 2.1: An example of the binary tree structure used by Gillenwater et al. (2019), with $M = 8$. Each conditional is sampled by traversing the tree from the root down to a leaf.

First we present the structure of the tree, see also Figure 2.1. The statistics that are stored at each node will become apparent as we explain the main idea of Gillenwater et al. (2019). Each node of this binary tree represents a set of items $S \subset \mathbb{X}$. The root corresponds to the ground set \mathbb{X} while the leafs represent each item individually. Starting from the root node, each node S is partitioned into two child nodes C_ℓ, C_r ⁸ of approximately the same size until all child nodes contain a single item.

Since we are in a finite setting, the conditional distribution of n -th item to be drawn is a multinomial distribution with marginal probabilities

$$\begin{cases} \frac{1}{N-(n-1)}(\mathbf{K}_{jj} - \mathbf{K}_{jJ_{n-1}}\mathbf{K}_{J_{n-1}}^{-1}\mathbf{K}_{jJ_{n-1}}^\top), & \text{if } j \notin J_{n-1} \\ 0, & \text{otherwise.} \end{cases} \quad (2.1.21)$$

The update of such quantities using Algorithm 1 (see 1.7-9) has roughly $\mathcal{O}(M(N + n))$ time complexity.

To amortize this linear dependency on M , Gillenwater et al. (2019) propose to traverse the precomputed binary tree from top to bottom in the following way. At each node S , proceed to the left child C_ℓ with probability⁹

$$\frac{\sum_{j \in C_\ell} \mathbf{K}_{jj} - \mathbf{K}_{jJ_{n-1}}\mathbf{K}_{J_{n-1}}^{-1}\mathbf{K}_{jJ_{n-1}}^\top}{\sum_{j \in S} \mathbf{K}_{jj} - \mathbf{K}_{jJ_{n-1}}\mathbf{K}_{J_{n-1}}^{-1}\mathbf{K}_{jJ_{n-1}}^\top} \quad (2.1.22)$$

or to the right child with complementary probability. This procedure generates a valid sample of the n -th conditional distribution (2.1.21). To see this, assume the exploration ends at the leaf associated to item j , then the corresponding transition probabilities (2.1.22) telescope so that the remaining term is indeed (2.1.21).

In fact, the crux of the approach is to use the precomputed summary statistics available at node C_ℓ to evaluate the above sum as a whole

⁸ C_ℓ for left child and C_r for right child, so that $S = C_\ell \cup C_r$.

⁹ At the root node $S = \mathbb{X}$ we have

$$(2.1.22) = \frac{\sum_{j \in C_\ell} \mathbf{K}_{jj} - \mathbf{K}_{jJ}\mathbf{K}_J^{-1}\mathbf{K}_{jJ}^\top}{N-(n-1)}$$

 If the path ends at j ,

$$(2.1.22) \propto \mathbb{K}_{jj}$$

instead of calculating each term individually to then sum them. For simplicity, let's drop the indexing of J_{n-1} . The required statistics will appear naturally after rewriting the numerator of (2.1.22) as

$$\begin{aligned}
& \sum_{j \in C_\ell} \mathbf{K}_{jj} - \mathbf{K}_{jJ} \mathbf{K}_J^{-1} \mathbf{K}_{jJ}^\top \\
&= \sum_{j \in C_\ell} \mathbf{K}_{jj} - \sum_{j \in C_\ell} \text{Tr}[\mathbf{K}_J^{-1} \mathbf{K}_{jJ}^\top \mathbf{K}_{jJ}] \\
&= \sum_{j \in C_\ell} \mathbf{K}_{jj} - \sum_{j \in C_\ell} \text{Tr}[\mathbf{K}_J^{-1} (U_{j:} U_{:,J}^\top)^T U_{j:} U_{:,J}^\top] \\
&= \sum_{j \in C_\ell} \mathbf{K}_{jj} - \text{Tr} \left[\mathbf{K}_J^{-1} U_{J:} \left(\sum_{j \in C_\ell} U_{j:}^\top U_{j:} \right) U_{J:}^\top \right] \\
&= \sum_{j \in C_\ell} \mathbf{K}_{jj} - \mathbf{1}^\top \left[\mathbf{K}_J^{-1} \circ U_{J:} \left(\sum_{j \in C_\ell} U_{j:}^\top U_{j:} \right) U_{J:}^\top \right] \mathbf{1}. \quad (2.1.23)
\end{aligned}$$

Now, at the n -th step of the chain rule, if the following quantities were stored at node C_ℓ ¹⁰

$$\sum_{j \in C_\ell} \mathbf{K}_{jj} = \sum_{j \in C_\ell} \|U_{j:}\|^2 \quad \text{and} \quad \sum_{j \in C_\ell} U_{j:}^\top U_{j:}, \quad (2.1.24)$$

then the probability (2.1.22) of transiting from S to its left child C_ℓ evaluates in $\mathcal{O}(nN^2)$ using (2.1.23). Sampling from each of the N conditional distributions, requires to perform $\log(M)$ such transitions. In the end, given the eigendecomposition $\mathbf{K} = UU^\top$, this tree-based method generates samples in $\mathcal{O}(\log(M)N^4)$ time complexity and requires $\mathcal{O}(MN^2)$ space in memory to store all the summary statistics.

Observe that if we consider the row $U_{j:}$ as a latent feature vector of item j , then the summary statistics (2.1.24) that are stored at each node refer to the squared norm and the covariance matrix of the corresponding feature vectors.

Using the factorization $\mathbf{K} = UU^\top$.

For any two square matrices $\text{Tr}[AB] = \mathbf{1}^\top [A \circ B^\top] \mathbf{1}$ where $A \circ B$ corresponds to the entrywise product and $\mathbf{1}$ denotes the vector with unit entries of the appropriate dimension.

¹⁰which do not depend on J but would cost respectively $\mathcal{O}(|C_\ell|N)$ and $\mathcal{O}(|C_\ell|N^2)$ to evaluate on the fly. For the nodes at the top of the tree $|C_\ell| \approx M$, this would imply a linear dependency on M , which is exactly what we want to avoid!

2.2 EXACT SAMPLING FROM NON-PROJECTION DPPs

2.2.1 Finite DPPs defined by their correlation kernel

We survey different exact sampling methods for finite DPPs defined by their correlation kernel \mathbf{K} , which is not assumed to be of projection type. The first ones by Poulson (2019) and Launay, Galerne, and Desolneux (2018) are based on matrix factorization techniques and apply generically, even to non-Hermitian correlation kernels.

First, let us try to be generic and consider a correlation kernel \mathbf{K} which is not assumed Hermitian nor of projection type.

FROM A PROBABILISTIC VIEWPOINT, A FIRST WAY OF THINKING THE SAMPLING OF DPP(\mathbf{K}) IS TO PERFORM A SEQUENTIAL BOTTOM-UP CHAIN RULE ON SETS, i.e., starting from the empty set, each item $1, \dots, M$ is decided in turn to be added or excluded to form the final sample \mathcal{X} (Launay, Galerne, and Desolneux, 2018, Section

2.2). This can be formalized as a top-down exploration of the binary probability tree displayed in Figure 2.2.

Example 2.2.1. For example, take $M = 5$ and assume the sample $\mathcal{X} = \{1, 4\}$ was generated. The corresponding chain rule steps

$$\begin{aligned}\mathbb{P}[\mathcal{X} = \{1, 4\}] &= \mathbb{P}[1 \in \mathcal{X}] \\ &\times \mathbb{P}[2 \notin \mathcal{X} \mid 1 \in \mathcal{X}] \\ &\times \mathbb{P}[3 \notin \mathcal{X} \mid 1 \in \mathcal{X}, 2 \notin \mathcal{X}] \\ &\times \mathbb{P}[4 \in \mathcal{X} \mid 1 \in \mathcal{X}, \{2, 3\} \cap \mathcal{X} = \emptyset] \\ &\times \mathbb{P}[5 \notin \mathcal{X} \mid \{1, 4\} \subset \mathcal{X}, \{2, 3\} \cap \mathcal{X} = \emptyset],\end{aligned}\tag{2.2.1}$$

are highlighted as a blue path in the binary tree of Figure 2.2.

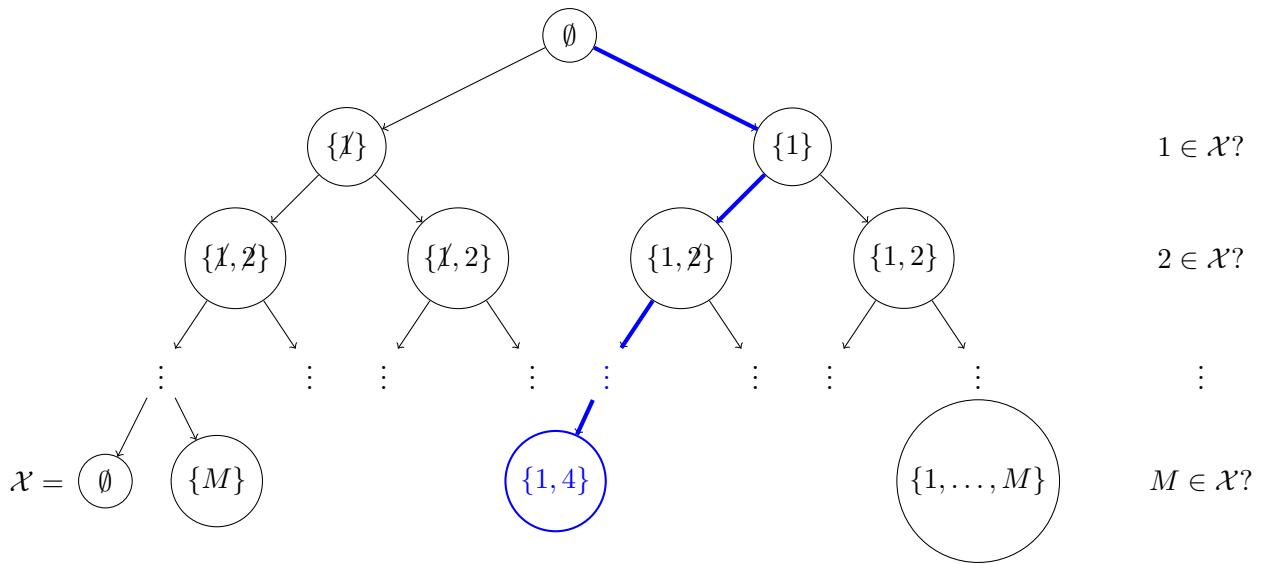


Figure 2.2: Chain rule on sets

In practice, at the m -th step of the procedure one needs to compute the incremental conditional probabilities of inclusion of item m given the past history of the exploration. To do this, the inclusion-exclusion principle given in Theorem 1.D.2 helps us to compute these quantities. At the m -th step of the chain rule, given that the subset A was included and B was excluded during the previous steps, the conditional probability of adding item m reads

$$\begin{aligned}\mathbb{P}[m \in \mathcal{X} \mid A \subset \mathcal{X}, \mathcal{X} \cap B = \emptyset] &= \frac{\mathbb{P}[A \cup \{m\} \subset \mathcal{X}, \mathcal{X} \cap B = \emptyset]}{\mathbb{P}[A \subset \mathcal{X}, \mathcal{X} \cap B = \emptyset]} \\ &= \frac{\det[\mathbf{K} - \mathbf{K}_{:B}[\mathbf{K} - I]_B^{-1}\mathbf{K}_{B:}]_{A \cup \{m\}}}{\det[\mathbf{K} - \mathbf{K}_{:B}[\mathbf{K} - I]_B^{-1}\mathbf{K}_{B:}]_A}.\end{aligned}$$

Using (1.D.3) in Theorem 1.D.2.

Following Launay, Galerne, and Desolneux (2018, Corollary 3), we note $H^B = \mathbf{K} - \mathbf{K}_{:B}[\mathbf{K} - I]_B^{-1}\mathbf{K}_{B:}$, so that

$$\begin{aligned}\mathbb{P}[m \in \mathcal{X} \mid A \subset \mathcal{X}, \mathcal{X} \cap B = \emptyset] &= \frac{\det[H^B]_{A \cup \{m\}}}{\det[H^B]_A} = [H^B]_{mm} - [H^B]_{mA}[H^B]_A^{-1}[H^B]_{Am}. \quad (2.2.2)\end{aligned}$$

Using Lemma 1.B.1.

However, it may become cumbersome and expensive to sequentially update the previous quantities naively.

More pragmatically, Poulson (2019, Algorithm 1, Theorem 2) put into correspondence the chain rule factorization of the likelihood $\mathbb{P}[\mathcal{X} = \mathbf{X}] = |\det[\mathbf{K} - I^{X^c}]|$ of the resulting sample with a slight modification of the sequential LU factorization procedure¹¹ applied to $\mathbf{K} - I^{X^c}$. More specifically, we can identify the factors coming from different expansions of the likelihood. During the first $m - 1$ steps of the decision process, if we note A_{m-1}, B_{m-1} the set of indices respectively included and excluded of \mathcal{X} , the likelihood of the sample factorizes as

$$\begin{aligned}\mathbb{P}[\mathcal{X} = \mathbf{X}] &= \mathbb{P}[X \subset \mathcal{X}, X^c \cap \mathcal{X} = \emptyset] \\ &= \prod_{x \in X} \mathbb{P}[x \in \mathcal{X} \mid A_{x-1} \subset \mathcal{X}, B_{x-1} \cap \mathcal{X} = \emptyset] \\ &\quad \times \prod_{y \in X^c} \mathbb{P}[y \notin \mathcal{X} \mid A_{y-1} \subset \mathcal{X}, B_{y-1} \cap \mathcal{X} = \emptyset].\end{aligned}\quad (2.2.3)$$

Correspondingly, from the matrix factorization viewpoint, the LU decomposition of $K - I^{X^c}$ allows us to write

$$\begin{aligned}\mathbb{P}[\mathcal{X} = \mathbf{X}] &= (-1)^{|X^c|} \det[\mathbf{K} - I^{X^c}] \\ &= (-1)^{|X^c|} \det[LU] = (-1)^{|X^c|} \prod_{i=1}^M U_{ii} \\ &= \prod_{x \in X} U_{xx} \prod_{y \in X^c} -U_{yy}.\end{aligned}\quad (2.2.4)$$

As shown by Poulson (2019, Theorem 2), the unblocked and unpivoted version of the sequential LU decomposition of $\mathbf{K} - I^{X^c}$ performed by Algorithm 2, guarantees the identification¹² of the terms in the products composing (2.2.3) and (2.2.4), that is

$$U_{mm} = \begin{cases} \mathbb{P}[m \in \mathcal{X} \mid A_{m-1} \subset \mathcal{X}, B_{m-1} \cap \mathcal{X} = \emptyset], & \text{if } m \in X, \\ -\mathbb{P}[m \notin \mathcal{X} \mid A_{m-1} \subset \mathcal{X}, B_{m-1} \cap \mathcal{X} = \emptyset] & \text{if } m \notin X. \\ = \mathbb{P}[m \in \mathcal{X} \mid A_{m-1} \subset \mathcal{X}, B_{m-1} \cap \mathcal{X} = \emptyset] - 1 & \end{cases}\quad (2.2.5)$$

In the end, the likelihood (2.2.3) of the sample can be computed via (2.2.4) as $\mathbb{P}[\mathcal{X} = \mathbf{X}] = \prod_{i=1}^M |U_{ii}|$.

We mention that the decision of updating of the pivot element $C[m, m]$ made at 1.4-7 of Algorithm 2 makes the division by $C[m, m]$ stable with high probability. Hence the stability of the global procedure. Indeed, after $m - 1$ steps, if the m -th pivot element is ϵ close to zero, i.e., $C[m, m] = \mathbb{P}[m \in \mathcal{X} \mid A_{m-1} \subset \mathcal{X}, B_{m-1} \cap \mathcal{X} = \emptyset] < \epsilon$ then is updated to $C[m, m] - 1 \leq -1 + \epsilon$ with probability at least $1 - \epsilon$. Hence the stability of the division by $C[m, m]$ at 1.9 of Algorithm 2.

When the kernel \mathbf{K} is Hermitian, one can leverage the symmetries to replace the LU factorization-based Algorithm 2 by an equivalent LDL^H version, see Algorithm 5.

In the end, the overall cost of generating one sample from DPP(\mathbf{K}) using Algorithm 2 is of order $\mathcal{O}(M^3)$. To get a new sample, one

¹¹ See, e.g., Golub and Van Loan (2013, Algorithm 3.2.1) for the classical LU decomposition.

By Corollary 1.D.3.

Since U is upper triangular and L is lower triangular with unit diagonal.

¹² We invite the reader to check on Example 2.2.1 that Algorithm 2 performs the right incremental updates (2.2.2).

$$\mathbf{A} = \begin{bmatrix} & & & \\ & \ddots & & \\ & & \mathbf{U} & \\ & & & \\ & \mathbf{L} \setminus \text{diag} & & \end{bmatrix}$$

Figure 2.3: Inplace LU factorization of $\mathbf{K} - I^{X^c} = LU$ returned by Algorithm 2.

Algorithm 2: Generate a sample \mathbf{X} from a generic DPP(\mathbf{K}) (LU version) see also Poulson (2019, Algorithm 1) and Figures 2.2 and 2.3

Require: Correlation kernel \mathbf{K}

```

1:  $C = \mathbf{K}.\text{copy}()$ 
2:  $X = \emptyset$ 
3: for  $m$  in range( $M$ ) do
4:   if  $\text{rand}() < C[m, m]$  # = probability (2.2.2) then
5:      $X = X \cup \{m\}$ 
6:   else
7:      $C[m, m] -= 1$ 
8:   end if
9:    $C[m+1:, m] /= C[m, m]$ 
10:   $C[m+1:, m+1:] -= C[m+1:, m] C[m, m+1:]$  # outer product
11: end for
12: return  $X, C$  # C = inplace LU factorization of  $K - I^{X^c}$ , Figure 2.3

```

needs to restart the procedure from scratch as opposed to the spectral method presented below.

We mention that Poulson (2019) provides a fully optimized C++ package¹³ collecting implementations of this generic method along with several variants. These variants include orthogonal projection DPPs which is equivalent to Algorithm 1, Hermitian DPPs, see also Algorithm 5, and DPPs with sparse correlation kernels \mathbf{K} . We mention that the inclusion decisions can also be taken by block (Poulson, 2019, Algorithm 5), this permits semi-parallel updates of the factorization.

¹³ gitlab.com/hodge_star/catamari.

In fact, before the work of Poulson (2019), the connection between sequential DPP sampling and matrix factorization techniques started with Launay, Galerne, and Desolneux (2018, Section 2.2) where the focus was on symmetric DPP(\mathbf{K}), i.e., when $0 \preceq \mathbf{K} \preceq I$. On top of that, the authors introduced a particular coupling which favors a special blocked version of the sequential algorithm.

THE INCLUSION DECISIONS OF THE SEQUENTIAL PROCEDURE CAN BE FOCUSED ONTO A SMALLER SUBSET OF ITEMS THAN THE ENTIRE GROUND SET. Launay, Galerne, and Desolneux (2018, Section 3.2) derived the so-called sequential thinning procedure, by coupling a target DPP with another process, such the samples of the other process contain the DPP samples.

The thinning term comes from the fact that the decision of including item m into the final sample $\mathcal{X} \sim \text{DPP}(\mathbf{K})$ is conditioned on the presence of m into an intermediate sample $\mathcal{Y} \triangleq \{y_1, \dots, y_p\}$ drawn from a so-called *dominating process*, with the guarantee that $\mathcal{X} \subset \mathcal{Y}$. Only the items $m \in \mathcal{Y}$ are decided to be included or not in \mathcal{X} but we stress that this sequential thinning procedure does not reduce to subsampling \mathcal{X} from \mathcal{Y} directly.¹⁴ Indeed, because of the sequential nature of the procedure, the inclusion decision of adding y_1 to \mathcal{X} and then excluding $y_1 + 1, \dots, y_2 - 1$ impacts the decision of adding y_2 into \mathcal{X} , etc. These conditional information can be passed via block updates

¹⁴ Dereziński, Calandriello, and Valko (2019) adopt a different thinning strategy to effectively draw DPP samples from a preselected subset. This idea is presented in Section 2.2.2.

of the underlying factors.

The dominating process used by Launay, Galerne, and Desolneux (2018) is a Bernoulli process. Sampling from it is easy since it boils down to drawing independent Bernoulli variables, but the corresponding parameters

$$q_m \triangleq \mathbb{P} \left[m \in \mathcal{X} \mid \underbrace{\{1, \dots, m-1\} \cap \mathcal{X}}_{\triangleq \tilde{B}_{m-1}} = \emptyset \right] \quad (2.2.6)$$

$$= \mathbf{K}_{mm} - \mathbf{K}_{m, \tilde{B}_{m-1}} [\mathbf{K} - I]_{\tilde{B}_{m-1}}^{-1} \mathbf{K}_{\tilde{B}_{m-1} m}, \quad (2.2.7)$$

are expensive to compute. In practice, Launay, Galerne, and Desolneux (2018) propose to evaluate the q_m s from the precomputation, in $\mathcal{O}(M^3)$, of the Cholesky decomposition of $I - \mathbf{K} = LL^\top$:

$$q_m = \mathbf{K}_{mm} + \mathbf{K}_{\tilde{B}_{m-1} m}^\top [I - \mathbf{K}]_{\tilde{B}_{m-1}}^{-1} \mathbf{K}_{\tilde{B}_{m-1} m} \quad (2.2.8)$$

$$= \mathbf{K}_{mm} + \left\| [L_{\tilde{B}_{m-1}}]^{-1} \mathbf{K}_{\tilde{B}_{m-1} m} \right\|^2. \quad (2.2.9)$$

However, we point out that these computations are unnecessary since

$$q_m = \mathbb{P}[m \in \mathcal{X} \mid \{1, \dots, m-1\} \cap \mathcal{X} = \emptyset],$$

can be identified directly from a factored form of $\mathbf{K} - I$ or $\mathbf{K} - I$. To see this, consider the generic case and the factorization $\mathbf{K} - I = LU$. Plugging $X = \emptyset$ into (2.2.4)-(2.2.5) yields the correspondence $q_m = 1 + U_{mm}$. From $I - \mathbf{K} = LL^\top$, the identification is $q_m = 1 - L_{mm}^2$. For more details on the latter identification we refer to Appendix 2.B.

After drawing \mathcal{Y} from the dominating process one can adapt the sequential Algorithm 2 and concentrate the decision of adding an item or updating the pivots only at indices $m \in \mathcal{Y}$ with probability

$$\frac{U_{mm}}{q_m} = \frac{\mathbb{P}[m \in \mathcal{X} \mid A_{m-1} \subset \mathcal{X}, B_{m-1} \cap \mathcal{X} = \emptyset]}{\mathbb{P}[m \in \mathcal{X} \mid \{1, \dots, m-1\} \cap \mathcal{X} = \emptyset]}. \quad (2.2.10)$$

This can be seen as an importance sampling phase which compensates the fact that $m \in \mathcal{Y}$ has not the correct distribution, see Algorithm 3.

Finally, we mention that the sequential thinning procedure, originally developed for symmetric DPPs, can be extended directly to the general case using a block LU factorization.¹⁵

¹⁵ See, e.g., Golub and Van Loan (2013, Section 3.2.11).

WHEN THE KERNEL \mathbf{K} IS HERMITIAN, WE CAN LEVERAGE THE FACT THAT HERMITIAN DPPS ARE MIXTURES OF ORTHOGONAL PROJECTION DPPS TO GENERATE REPEATED SAMPLES GIVEN THE SPECTRAL CONTENT OF THE KERNEL. This method is commonly called the *spectral method* since it requires the spectral/eigen decomposition of the Hermitian kernel

$$\mathbf{K} = U \Lambda U^\text{H} = \sum_{m=1}^M \lambda_m U_{:m} U_{:m}^\text{H}, \quad (2.2.11)$$

where $U^\text{H} \text{diag}(\boldsymbol{\omega}) U = I_M$ and $0 \leq \lambda_m \leq 1$. After this expensive $\mathcal{O}(M^3)$ preprocessing step, the spectral content of the kernel can be reused to generate new samples. In fact, Theorem 1.2.4 already provides the two steps of this spectral method:

Algorithm 3: Generate a sample \mathbf{X} from a generic DPP(\mathbf{K}) with the sequential thinning method (unblocked way)

Require: Correlation kernel \mathbf{K} and Bernoulli parameters q of the dominating process

```

1:  $C = \mathbf{K}.\text{copy}()$ 
2:  $\mathcal{Y} = \{m; \text{rand}() < q[m]\}$  # = draw from the dominating process
3:  $X = \emptyset$ 
4: for  $m$  in  $\text{range}(M)$  do
5:   if  $m \in \mathcal{Y}$  and  $\text{rand}() < \frac{C[m, m]}{q[m]}$  # = importance sampling then
6:      $X = X \cup \{m\}$ 
7:   else
8:      $C[m, m] -= 1$ 
9:   end if
10:   $C[m+1:, m] /= C[m, m]$ 
11:   $C[m+1:, m+1:] -= C[m+1:, m] C[m, m+1:]$  # outer product
12: end for
13: return  $X, C$ 
```

Algorithm 4: Generate a sample \mathbf{X} from a Hermitian DPP(\mathbf{K}) with the spectral method (pseudo code)

Require: The eigendecomposition $\mathbf{K} = U \Lambda U^H = \sum_{m=1}^M \lambda_m u_{:m} u_{:m}^H$

```

1: Draw independently  $B_1, \dots, B_M$  where  $B_m \sim \text{Bernoulli}(\lambda_m)$ 
2: Set  $\mathcal{B} \triangleq \{1 \leq m \leq M ; B_m = 1\}$ .
3: Generate a sample  $\mathbf{X}$  from the orthogonal projection DPP( $\mathbf{K} = U_{:\mathcal{B}} U_{:\mathcal{B}}^H$ ) using, e.g., Algorithm 1.
4: return  $X$ 
```

1. Select a component of the mixture by drawing independent Bernoulli variables with parameters the eigenvalues of \mathbf{K} .
2. Generate a sample from the selected orthogonal projection DPP.

In the end, the spectral method generates samples in time $\mathcal{O}(M(\text{Tr } \mathbf{K})^2)$ on average, where $\text{Tr } \mathbf{K} = \mathbb{E}[|\mathcal{X}|] = \sum_{m=1}^M \lambda_m$. The procedure is summarized as a pseudo code in Algorithm 4.

Contrary to the sequential factorization-based methods, where each sample requires to restart the whole procedure from scratch, the spectral method offers a practical alternative. Indeed, given the eigendecomposition of the kernel, both the eigenvalues and eigenvectors can then be reused to generate new samples.

2.2.2 Finite DPPs defined by their likelihood kernel

In the Hermitian case, while the correlation kernel must satisfy two positive semi-definite constraints, namely $0 \preceq \mathbf{K} \preceq I$, the likelihood kernel needs only to be positive semi-definite, i.e., $\mathbf{L} \succeq 0$. In terms of modelization, the L -ensemble viewpoint is thus easier to handle.

$\mathbf{K} = \mathbf{L}(I + \mathbf{L})^{-1}$, cf. Proposition 1.1.6, if one can afford the inversion of $I + \mathbf{L}$ followed by the multiplication with \mathbf{L} ,¹⁶ sampling from an L -ensemble boils down to sample from DPP($\mathbf{K} = \mathbf{L}(I + \mathbf{L})^{-1}$). This can be done using Algorithm 2 (Poulson, 2019, Algorithm 1) where each sample costs $\mathcal{O}(M^3)$. However, if one can afford the $\mathcal{O}(M^3)$ cost for the inversion and sampling, one can also use Algorithm 4 involving the eigendecomposition of the underlying kernel.

IF THE EIGENDECOMPOSITION OF THE LIKELIHOOD KERNEL IS AVAILABLE or if one can afford to compute it in $\mathcal{O}(M^3)$,

$$\mathbf{L} = \sum_{j=1}^M \gamma_j v_j v_j^\top = V \Gamma V^\top, \quad (2.2.12)$$

the corresponding correlation kernel $\mathbf{K} = \mathbf{L}(I + \mathbf{L})^{-1}$ reads

$$\mathbf{K} = \sum_{j=1}^M \frac{\gamma_j}{1 + \gamma_j} v_j v_j^\top = V \Gamma (I + \Gamma)^{-1} V^\top, \quad (2.2.13)$$

Then, using the spectral method, i.e., Algorithm 2, with $\lambda_j = \frac{\gamma_j}{1 + \gamma_j}$ and $U = V$, each sample can be generated in $\mathcal{O}(M(\text{Tr } \mathbf{K})^2)$ on average, where $\text{Tr } \mathbf{K} = \mathbb{E}[|\mathcal{X}|] = \sum_{j=1}^M \frac{\gamma_j}{1 + \gamma_j}$.

WHEN THE LIKELIHOOD KERNEL IS CONSTRUCTED AS A GRAM MATRIX $\mathbf{L} = \Phi^\top \Phi$, where each item $j \in \{1, \dots, M\}$ is represented by a feature vector $\phi_j \in \mathbb{R}^d$, one can consider shifting the computational overhead on the so-called *dual kernel*.¹⁷ This may become particularly useful when the number of items M is too large to work efficiently with \mathbf{L} directly but the dimension d of the features is much smaller, i.e., $d \ll M$. In this setting, the dual kernel is of size $d \times d$ which is much smaller than the original likelihood kernel, and its eigendecomposition¹⁸

$$\Phi \Phi^\top = \sum_{j=1}^d \theta_j w_j w_j^\top = W \Theta W^\top, \quad (2.2.14)$$

becomes cheaper to compute, in $\mathcal{O}(d^3)$. Observing that the eigendecomposition of the likelihood kernel (2.2.12) and its dual relate in the following way¹⁹

$$\mathbf{L} = \underbrace{\Phi^\top W \Theta^{-\frac{1}{2}}}_{V(M \times d)} \underbrace{\Theta}_{\Gamma(d \times d)} \underbrace{(W \Theta^{-\frac{1}{2}})^\top}_{V^\top}, \quad (2.2.15)$$

we can resort again to the spectral method, i.e., Algorithm 4, with $V = \Phi^\top W \Theta^{-\frac{1}{2}}$ and $\Lambda = \Gamma$. The overall cost of the procedure can be summarized as follows. The preprocessing cost inherent to the computation of $\Phi \Phi^\top$, its eigendecomposition and the reconstruction of the eigenvectors of the likelihood kernel as $\Phi^\top W \Theta^{-\frac{1}{2}}$ is of order $\mathcal{O}(Md^2 + d^3)$. This is much lower than the original $\mathcal{O}(M^3)$ cost to eigendecompose \mathbf{L} in the first place. Then, using Algorithm 4 with $\lambda_j = \frac{\theta_j}{1 + \theta_j}$ and $U = \Phi^\top W \Theta^{-\frac{1}{2}}$ allows us to generate each sample in $\mathcal{O}(M(\text{Tr } \mathbf{K})^2)$ on average, where $\text{Tr } \mathbf{K} = \mathbb{E}[|\mathcal{X}|] = \sum_{j=1}^d \frac{\theta_j}{1 + \theta_j} < d$.

¹⁶If $\mathbf{L} \succeq 0$, we can write $\mathbf{L}(I + \mathbf{L})^{-1} = I - (I + \mathbf{L})^{-1}$ and only the inversion of $I + \mathbf{L}$ is required.

¹⁷See Kulesza and Taskar (2012, Section 3.3).

¹⁸one may also consider computing the singular value decomposition of Φ .

¹⁹See, e.g., Kulesza and Taskar (2012, Proposition 3.1).

We also mention that the tree-based method we chose to present in the context of orthogonal projection DPPs, was originally developed by Gillenwater et al. (2019) to generate samples from $\text{DPP}(\mathbf{L} = \Phi^\top \Phi)$ using the dual viewpoint. The tree structure converts the $\mathcal{O}(M(\text{Tr } \mathbf{K})^2)$ complexity of second phase of the spectral method into $\mathcal{O}(\log(M)(\text{Tr } \mathbf{K})^4 + d)$ on average, where $\text{Tr } \mathbf{K} = \mathbb{E}[|\mathcal{X}|] = \sum_{j=1}^d \frac{\theta_j}{1+\theta_j} < d$. This method becomes a viable alternative to the spectral method when the total number of items M is large and when the dimensionality d of the features and the expected sample size are very small compared to M .

Motivated by recommendation applications where it is desirable to make repeated personalized propositions to potentially a very large number of users, Gillenwater et al. (2019, Section 4) also introduce a *personalized* variant of this tree-based sampling scheme. The ultimate goal is to recycle the tree structure and preserve the logarithmic dependency on the total number of items and in order to scale to potentially very large number of users. In a few words, to achieve this, instead of changing the reference weights ω of $\text{DPP}(\omega, \mathbf{L} = \Phi^\top \Phi)$ for each user which would result in a M dependency, the d features are reweighted on a user basis. As a consequence, for each user, a full eigendecomposition of the rescaled dual kernel of size $d \times d$ is required but the same tree structure with summary statistics shared by each user, allows us to incorporate user preferences at query time and yields repeated samples in $\mathcal{O}(\log(M)d^2(\text{Tr } \mathbf{K})^2 + d^3)$.

EXACT DPP SAMPLING CAN ALSO BE PERFORMED BY THINNING AN EASY-TO-SAMPLE INTERMEDIATE PROCESS WITH ANOTHER TAILORED DPP. In a stream of works, Dereziński (2019) and Dereziński, Calandriello, and Valko (2019) developed a thinning procedure to generate exact samples from $\text{DPP}(\mathbf{L})$. The authors first draw an intermediate sample $\mathcal{Y} \subset \mathbb{X}$ from an easy-to-sample distribution. The bias introduced at this first step is then corrected by downsampling/thinning of \mathcal{Y} with a specific $\text{DPP}(\tilde{\mathbf{L}})$. This method allows us to shift the computational cost of exact DPP sampling from a potentially large ground set $\mathbb{X} = \{1, \dots, M\}$ onto a smaller subset $\mathcal{Y} \subset \mathbb{X}$. The choice of the intermediate sampling distribution relies on the connection between DPPs and so-called *ridge leverage scores* (RLS),²⁰ which are commonly used for sampling in randomized linear algebra.

²⁰ See, e.g., Alaoui and Mahoney, 2015.

Definition 2.2.2 (Ridge leverage scores and effective dimension). *Let \mathbf{L} be a positive semi-definite matrix, i.e., $\mathbf{L} \succeq 0$. For any $\lambda > 0$, the λ -ridge leverage scores (λ -RLS) associated to \mathbf{L} are defined as*

$$\tau_i(\lambda) \triangleq [\mathbf{L}(\lambda I + \mathbf{L})^{-1}]_{ii}. \quad (2.2.16)$$

We denote $d_{\text{eff}}(\lambda) \triangleq \sum_{i=1}^M \tau_i(\lambda)$ the corresponding effective dimension.

In fact, the 1-RLS naturally arise in our setting²¹ as the first order inclusion probabilities of $\mathcal{X} \sim \text{DPP}(\mathbf{L})$,

$$\tau_i(1) = [\mathbf{L}(I + \mathbf{L})^{-1}]_{ii} = \mathbf{K}_{ii} = \mathbb{P}[i \in \mathcal{X}],$$

²¹ See Proposition 1.1.6.

and the corresponding effective dimension is exactly the expected number of points

$$d_{\text{eff}}(1) = \text{Tr}[\mathbf{K}] = \mathbb{E}[|\mathcal{X}|].$$

Intuitively, if one samples n items y_1, \dots, y_n , proportionally to the 1-RLS, i.e., i.i.d. with replacement from \mathbb{X} according to the probability vector

$$\frac{1}{d_{\text{eff}}}(\tau_1, \dots, \tau_M) = \frac{1}{\text{Tr}[\mathbf{K}]}(\mathbf{K}_{11}, \dots, \mathbf{K}_{MM}), \quad (2.2.17)$$

the random set $\mathcal{Y} = \{y_1, \dots, y_n\}$ (the duplicates have been removed) is similar to a sample of $\text{DPP}(\text{diag}(\mathbf{K}))$ where there is no correlation between the points. Thus, if the resulting set \mathcal{Y} is sufficiently large, typically²² when $n = \mathcal{O}(\text{Tr}[\mathbf{K}]^2)$ where $\text{Tr}[\mathbf{K}] = \mathbb{E}[|\mathcal{X}|]$, it is likely that a proper sample $\mathcal{X} \sim \text{DPP}(\mathbf{L})$ will be contained within \mathcal{Y} . Then, they showed that a carefully designed $\text{DPP}(\tilde{\mathbf{L}})$ defined on \mathcal{Y} allows us to correct the bias of the i.i.d. sampling (2.2.17) and produce an exact DPP sample. In other words, subsampling the intermediate realization \mathcal{Y} with $\text{DPP}(\tilde{\mathbf{L}})$ re-injects the proper correlations between the items, and offers substantial computational savings when $|\mathcal{Y}| \ll M$.

In practice, exact calculation of the 1-RLS costs $\mathcal{O}(M^3)$. For the same cost, one could resort to the spectral method by computing the eigendecomposition of \mathbf{L} . The main contribution of Dereziński, Calandriello, and Valko (2019) was to show that one can work with a cheaper approximate version of the leverage scores, while maintaining the exactness of the output samples.

Finally, the authors show that their method generates exact DPP samples with high probability. They can guarantee that, with probability $1 - \delta$ ($\delta > 0$), the first sample can be generated (preprocessing included) in time-complexity

$$\mathcal{O}\left(M \text{Tr}[\mathbf{K}]^6 \log^2\left(\frac{M}{\delta}\right) + \text{Tr}[\mathbf{K}]^9 \log^3\left(\frac{M}{\delta}\right) + \text{Tr}[\mathbf{K}]^3 \log^4\left(\frac{M}{\delta}\right)\right).$$

Then, repeated samples can be generated in

$$\mathcal{O}\left(\text{Tr}[\mathbf{K}]^6 \log\left(\frac{M}{\delta}\right) + \log^4\left(\frac{M}{\delta}\right)\right).$$

Looking at the above costs, the overall procedure excels in the regime where the total number of items M is potentially very large and the expected number of points $\mathbb{E}[|\mathcal{X}|] = \text{Tr}[\mathbf{K}]$ is comparatively very small. Nonetheless, some parameters of the algorithm, like the level of approximation of the leverage scores and the expected size of the intermediate sample, may need to be tuned. Besides, sampling $\text{DPP}(\tilde{\mathbf{L}})$ in the last phase, relies on classical DPP samplers.

For the details of this intricate procedure, combining the estimation of the 1-RLS, Nyström approximation of \mathbf{L} , concentration inequalities, DPP sampling and rejection sampling, we refer directly to Dereziński, Calandriello, and Valko (2019). We mention that we have worked with Daniele Calandriello²³ to make the actual sampler available in DPPy.

²² Dereziński, Calandriello, and Valko (2019) used the concentration results of Pemantle and Peres (2014).

²³ github.com/danielecc

2.2.3 The continuous case

Comparatively to the finite case, much less efforts have been put in the sampling phase of generic continuous DPPs. In particular, we are not aware of any algorithm like Algorithm 2 which would apply generically to any correlation kernel K . However, the challenges that arise in the continuous case differ radically from the finite setting. In the following, we consider only Hermitian DPPs, that is $K(x, y) = \overline{K(y, x)}$.

In the continuous case, some additional conditions on the kernel are required²⁴ to be able to write its eigendecomposition (1.1.23) $K(x, y) = \sum_{k=1}^{\infty} \lambda_k \phi_k(x) \overline{\phi_k(y)}$. Starting from this expansion, one could think of generalizing the spectral procedure presented in Theorem 1.2.4. The two main challenges are (a) the infinite (but countable) number of eigenvalues, (b) the evaluation of the eigenfunctions.

²⁴ See, e.g., Proposition 1.1.9.

Regarding challenge (a), drawing an infinite number of independent Bernoulli variables with parameters the eigenvalues $\lambda_1, \lambda_2, \dots$, is not feasible without further assumption on the decay of $(\lambda_k)_{k \geq 1}$. However, the condition (1.1.13) on the trace of K is equivalent to $\sum_{k \geq 0} \lambda_k < \infty$ and Borel-Cantelli's lemma gives that only a finite number of Bernoulli variables can realize as ones (almost surely). Thus, the initial phase of the spectral algorithm can be performed by first generating

$$M_{\max} \triangleq \max(k \geq 1 \mid B_k = 1), \quad \text{with } B_k \sim \text{Bernoulli}(\lambda_k), \quad (2.2.18)$$

followed by an independent sampling of Bernoulli variables with parameters $\lambda_1, \dots, \lambda_{M_{\max}}$. In fact, sampling from (2.2.18) in the first place is not granted in practice, since it requires a full access to the infinite sequence of eigenvalues, or for instance, a recursive relation that is simple to propagate. We refer to Lavancier, Møller, and Rubak (2015, Appendix D) for more details on this matter. A potential workaround would be to set a particular criterion to truncate the series (??), but the resulting samples would not be exact.

Challenge (b) requires an efficient way to evaluate the entries of the projection kernel (1.2.5), selected at the first stage. The formulation (??) is given on paper but an analytic and tractable form of the eigenfunctions is really the bottleneck for practical implementation. Sometimes, the eigenfunctions are special functions like orthogonal polynomials. In the latter case, the orthogonal polynomials are linked together through the so-called three-term recurrence relation, which becomes a practical way of evaluating the eigenfunctions, see, e.g., Olver, Nadakuditi, and Trogdon (2015).

Again, these challenges are inherent to the spectral method itself, and conceptual shifts must be initiated to make exact sampling more practical. In this direction, we mention the work of Decreusefond, Flint, and Low (2013), where the authors adapt a perfect simulation scheme, which seemingly requires only access to the trace of the likelihood kernel.

All the constraints and limitations arising in both the finite and the continuous cases, have motivated research towards approximate methods.

APPENDICES

2.A SPECIALIZATION OF THE SEQUENTIAL SAMPLER FOR HERMITIAN DPPs

We saw that Algorithm 2 (Poulson, 2019, Algorithm 1) provides a compact sequential procedure to sample from a generic DPP(\mathbf{K}). It reveals the intrinsic link between DPP sampling and the LU factorization method. However, when the kernel is Hermitian, symmetries can be exploited to replace the sequential LU factorization-based sampler by the sequential LDL^H factorization-based method.²⁵ given in Algorithm 5. The latter generates exact samples distributed according to an Hermitian DPP(\mathbf{K}).

Algorithm 5: Generate a sample \mathbf{X} from a Hermitian DPP(\mathbf{K}), LDL^H variant of Algorithm 2

Require: Correlation kernel \mathbf{K}

```

1:  $\mathbf{C} = \mathbf{K}.copy()$ 
2:  $\mathbf{X} = \emptyset$ 
3:  $\mathbf{v} = \text{zeros}(M)$ 
4: for  $m$  in range( $M$ ) do
5:    $\mathbf{v}[:,m] = \mathbf{C}[m, :m] * \text{diagonal}(\mathbf{C}[:,m, :m])$ 
6:    $\mathbf{C}[m, m] -= \mathbf{C}[m, :m] \mathbf{v}[:,m]^H$ 
7:   if  $\text{rand}() < \mathbf{C}[m, m]$  # = probability (2.2.2) then
8:      $\mathbf{X} = \mathbf{X} \cup \{m\}$ 
9:   else
10:     $\mathbf{C}[m, m] -= 1$ 
11:   end if
12:    $\mathbf{C}[m+1:, m] -= \mathbf{C}[m+1:, :m] \mathbf{v}[:,m]^H$ 
13:    $\mathbf{C}[m + 1:, m] /= \mathbf{C}[m, m]$ 
14: end for
15: return  $\mathbf{X}, \mathbf{C}$  #  $\mathbf{C} = \text{inplace } LDL^H \text{ factorization of } K - I^{X^c}$ 

```

²⁵ See, e.g., Golub and Van Loan (2013, Section 4.1.2) for the original LDL^H factorization procedure.

2.B A NOTE ON THE SEQUENTIAL THINNING PROCEDURE

Originally developed for sampling from symmetric DPPs, the sequential thinning procedure can be easily extended to Hermitian DPPs and even generic DPP(\mathbf{K}).

In particular, we want to point out that the computations of the Bernoulli parameters driving the dominating process can be deduced directly from a factored form of $\mathbf{K} - I$ or $I - \mathbf{K}$ with no extra computations contrary to the original proposition of Launay, Galerne, and Desolneux (2018, Algorithm 3 1.). We recall that a sample \mathcal{Y} from the dominating process is easy to generate: each item $m \in \mathbb{X}$ is decided to be selected, independently, with probability

$$q_m \triangleq \mathbb{P}\left[m \in \mathcal{X} \mid \underbrace{\{1, \dots, m-1\}}_{\triangleq B_{m-1}} \cap \mathcal{X} = \emptyset\right]. \quad (2.B.1)$$

In the original setting of Launay, Galerne, and Desolneux (2018) where \mathbf{K} is real symmetric, the Bernoulli parameters (2.B.1) are derived using the Cholesky decomposition of $I - \mathbf{K}$, that is precomputed in advance. More specifically, given the factorization $I - \mathbf{K} = LL^\top$, Launay, Galerne, and Desolneux (2018, Algorithm 3 1.) compute

$$q_m = \mathbf{K}_{mm} + \mathbf{K}_{B_{m-1}m}^\top [I - \mathbf{K}]_{B_{m-1}}^{-1} \mathbf{K}_{B_{m-1}m} \quad (2.B.2)$$

$$= \mathbf{K}_{mm} + \| [L_{B_{m-1}}]^{-1} \mathbf{K}_{B_{m-1}m} \|^2. \quad (2.B.3)$$

In fact, these computations are unnecessary. To see this, consider for simplicity the case $0 \preceq \mathbf{K} \prec I$. For any $m \geq 1$, we can express $\mathbb{P}[\{1, \dots, m\} \cap \mathcal{X} = \emptyset]$ in two different ways. First, using the chain rule

$$\begin{aligned} \mathbb{P}[\{1, \dots, m\} \cap \mathcal{X} = \emptyset] &= \prod_{i=1}^m \mathbb{P}[i \notin \mathcal{X} \mid B_{i-1} \cap \mathcal{X} = \emptyset] \\ &= \prod_{i=1}^m 1 - \mathbb{P}[i \in \mathcal{X} \mid B_{i-1} \cap \mathcal{X} = \emptyset] \\ &= \prod_{i=1}^m 1 - \underbrace{(\mathbf{K}_{ii} - \mathbf{K}_{i, B_{i-1}} [\mathbf{K} - I]_{B_{i-1}}^{-1} \mathbf{K}_{B_{i-1}i})}_{= q_i}. \end{aligned} \quad \text{Using (1.D.10).}$$

From the matrix factorization view point, given either the LU, LDL^\top or Cholesky decomposition of $I - \mathbf{K}$, we obtain

$$\begin{aligned} \mathbb{P}[\{1, \dots, m\} \cap \mathcal{X} = \emptyset] &= \det[I - \mathbf{K}]_{\{1, \dots, m\}} \\ &= \det[LU]_{\{1, \dots, m\}} \quad \left| \begin{array}{l} = \det[LDL^\top]_{\{1, \dots, m\}} \\ = \det D_{\{1, \dots, m\}} \end{array} \right. \\ &= \det L_{\{1, \dots, m\}} U_{\{1, \dots, m\}} \quad \left| \begin{array}{l} = |\det L_{\{1, \dots, m\}}|^2 \\ = \prod_{i=1}^m |L_{ii}|^2 \end{array} \right. \\ &= \prod_{i=1}^m U_{ii} \end{aligned}$$

By Theorem 1.D.2.

The second equality follows from the Cauchy-Binet formula (1.C.1) where we exploit the triangular structure of U and L and the fact that L has unit diagonal in case of LU and LDL^\top .

Since this holds for any $1 \leq m \leq M$, we can recursively identify the terms of each product, so that $1 - q_m = U_{mm} = D_{mm} = |L_{mm}|^2$. Finally, we can summarize the different cases in the following way. For a generic (valid) correlation kernel \mathbf{K} :

- given the U factor of the decomposition $\mathbf{K} - I = LU$ or $I - \mathbf{K} = LU$, we have $q_m = 1 - |U_{mm}|$.
- If \mathbf{K} is Hermitian, i.e., $0 \preceq \mathbf{K} \preceq I$, we can focus only on the diagonal factor D of the decomposition $\mathbf{K} - I = LDL^\top$ or $I - \mathbf{K} = LDL^\top$, and take $q_m = 1 - |D_{mm}|$.
- If $0 \preceq \mathbf{K} \prec I$, we can consider the L factor of the Cholesky decomposition $I - \mathbf{K} = LL^\top$ and take $q_m = 1 - |L_{mm}|^2$.

Finally, note that, starting from \mathbf{K} , it is less expensive to compute with $\mathbf{K} - I$ than $I - \mathbf{K}$.

Approximate DPP sampling

3

In this chapter, we consider methods generating approximate samples from DPPs. These methods can be clustered into two categories. In the first class, the approximation is made on the kernel defining the underlying DPP either using random projections or low-rank factorization techniques. The second class relies on Monte Carlo Markov chain (MCMC) methods. Our contribution,¹ presented in Section 3.3, falls within the second cluster and it is the main focus of the chapter. More details on approximate DPP sampling can also be found in the Ph.D. dissertation of Affandi (2014).

3.1 KERNEL APPROXIMATION METHODS

Random projection methods Kulesza and Taskar (2012) and Gillenwater (2014, Section 3.4) are usually applied in the setting where the likelihood kernel is defined as a Gram matrix $\mathbf{L} = \Phi^\top \Phi$, where Φ is a feature matrix of size $d \times M$. When $d \ll M$, we have seen in Section 2.2.2 that working with the so-called “dual” kernel $\Phi\Phi^\top$ can be an effective approach to sample exactly from DPP(\mathbf{L}). However, when the number of features d becomes too large, typically of the order or greater than M , the computational gain of working with the dual kernel is lost. This is where random projection methods are useful, they allow to reduce feature dimension guaranteeing that inner products, and thus volumes are preserved within a good approximation.

Another kernel approximation technique used in practice is called the Nyström method, (Belabbas and Wolfe, 2009; Affandi et al., 2013). It involves selecting a subset of items to construct a low-rank projection approximation of the likelihood kernel \mathbf{L} .

3.2 MONTE CARLO MARKOV CHAIN METHODS

While fast Markov-chain based exact algorithms exist for specific DPPs such as uniform spanning trees (Aldous, 1990; Propp and Wilson, 1998), generic DPPs have so far been addressed with approximate MCMC algorithms (Belabbas and Wolfe, 2009; Kang, 2013; Rebeschini and Karbasi, 2015; Anari, Gharan, and Rezaei, 2016; Li, Jegelka, and Sra, 2016a, 2016b).

In all these works, transitions are allowed only between states S and S' that differ by at most one element, that is $|S \Delta S'| \in \{0, 1\}$. In the following, we may name transitions:

- $S \rightarrow S' = S \cup \{j\}$ as “add” moves,
- $S \rightarrow S' = S \setminus \{i\}$ as “delete” moves,
- $S \rightarrow S' = (S \setminus \{i\}) \cup \{j\}$ as called “exchange” or “swap” moves.

3.1 Kernel approximation methods	61
3.2 Monte Carlo Markov chain methods	61
3.3 The zonotope viewpoint on finite projection DPPs	62
Hit-and-run and the Simplex Algorithm	
From Volume to Squared Volume	
3.4 Experiments	68
Non-uniform Spanning Trees	
Text Summarization	
3.5 Discussion	72

¹ G. Gautier, R. Bardenet, and M. Valko. 2017. *Zonotope hit-and-run for efficient sampling from projection DPPs*. In International Conference on Machine Learning (ICML). arXiv:1705.10498.

github.com/guilgautier/DPPy

In practice, transitions are made using a simple Metropolis kernel, with $\text{DPP}(\mathbf{L})$ as invariant measure. More precisely, the transition $S \rightarrow S'$ is accepted with probability proportional to

$$\min\left(1, \frac{\det \mathbf{L}_{S'}}{\det \mathbf{L}_S}\right) \quad (3.2.1)$$

Seemingly naive, these natural MCMC methods have polynomial bounds on their mixing rates with arbitrary DPPs as their limiting measure; see Anari, Gharan, and Rezaei (2016) for cardinality-constrained DPPs, and Li, Jegelka, and Sra (2016a, 2016b) for the general case.

We mention a recent result devised by Hermon and Salez (2019); the exchange random walk applied to generate samples from a projection DPP(\mathbf{K}) with a kernel of rank N has a mixing time of order

$$\mathcal{O}\left(MN \log\left(\log\left(\frac{1}{\det \mathbf{K}_{S_0}}\right)\right)\right), \quad (3.2.2)$$

where S_0 is the initial state of the chain. Comparatively to previous approaches the authors bring an additional log term, which represents an important improvement guarding against a bad initialization.

However, the mixing time of these chains does not reflect entirely the overall cost of the algorithm. Indeed, the mixing time characterizes the number of iterations needed to reach ϵ -convergence to the limiting distribution and does not take into account the cost of computing the acceptance ratio (3.2.1) at each step. A naive computation of this ratio of determinants costs $\mathcal{O}(N^3)$ but smarter updates can be performed using, e.g., Lemma 1.B.1 or tight lower/upper bounds (Li, Sra, and Jegelka, 2016).

3.3 THE ZONOTYPE VIEWPOINT ON FINITE PROJECTION DPPs

Our contribution is the construction of a fast-mixing Markov chain with limiting distribution a given projection DPP. The main assumption is that we require the feature matrix $\Phi \in \mathbb{R}^{N \times M}$ to be full row-rank, that is $\text{rank}(\Phi) = N$. In that case, the (orthogonal) projection DPP(\mathbf{K}) with kernel

$$\mathbf{K} = \Phi^\top (\Phi \Phi^\top)^{-1} \Phi, \quad (3.3.1)$$

is well defined. In particular, as we saw in Section 1.1.8 the likelihood of DPP(\mathbf{K}) reads

$$\mathbb{P}[\mathcal{X} = B] = \frac{(\det \Phi_{:B})^2}{\det \Phi \Phi^\top} \mathbb{1}_{|B|=N}. \quad (3.3.2)$$

In other words, this projection DPP assigns positive probability to sets $B \subset \mathbb{X}$ made of exactly N items, such that the corresponding feature vectors are linearly independent: $\{\Phi_{:m}\}_{m \in B}$ form a basis of \mathbb{R}^N .

Our goal is now to design a Markov chain on

$$\mathcal{B} \triangleq \left\{ B \subset \mathbb{X} ; |B| = N, \{\Phi_{:j}\}_{j \in B} \text{ are independent} \right\}, \quad (3.3.3)$$

where the elements $B \in \mathcal{B}$ are naturally called “bases”, in the matroid literature.²

The state space \mathcal{B} of the chain can be described geometrically as the collection of sets of N vectors of \mathbb{R}^N spanning a non-zero volume. Hence, each sample $B \in \mathcal{B}$ of DPP(\mathbf{K}) is associated to a non-degenerate parallelopiped spanned by the corresponding feature vectors $\{\Phi_j\}_{j \in B}$. Figure 3.1 illustrates the situation in the case where $N = 2$.

We exploit further this geometrical embedding to construct a Markov Chain which explores better the state space \mathcal{B} by allowing bigger jumps than the previous exchange walk, cf. Section 3.2, which performs only very local moves. To do this, we introduce the notion of zonotope.

Definition 3.3.1 (Zonotope). *The zonotope associated to $\Phi \in \mathbb{R}^{N \times M}$ is defined as*

$$\mathcal{Z}(\Phi) \triangleq \Phi[0, 1]^M = \left\{ x \in \mathbb{R}^N \mid x = \sum_{m=1}^M y_j \Phi_j, \text{ with } 0 \leq y_j \leq 1 \right\}.$$

It can be identified as the affine transformation of the unit hypercube $[0, 1]^M$ by Φ . In particular, $\mathcal{Z}(\Phi)$ is a convex polytope, see Figure 3.2.

Under the row-rank assumption of Φ , the zonotope $\mathcal{Z}(\Phi)$ characterizes a N -dimensional volume of \mathbb{R}^N . Moreover, each sample $B \in \mathcal{B}$ can be represented by the parallelopiped $\mathcal{Z}(\Phi_{:B})$, as in Figure 3.1.

LINEAR PROGRAMMING GIVES A WAY TO LOCATE THE POINTS OF THE ZONOTOPE $\mathcal{Z}(\Phi)$ IN REGIONS SHAPED BY THE PARALLELTOPES $\mathcal{Z}(\Phi_{:B})$, REPRESENTING THE SUPPORT OF DPP(\mathbf{K}). Combining a continuous Markov chain on the zonotope $\mathcal{Z}(\Phi)$ with linear programming, we can explore the finite support of the target DPP($\Phi^\top (\Phi \Phi^\top)^{-1} \Phi$) more freely than the exchange random walk. The crux of our method relies on the proof of the following theorem, involving linear programming.

Proposition 3.3.2 (Dyer and Frieze, 1994, Theorem 3). *The zonotope $\mathcal{Z}(\Phi)$ can be partitioned in regions shaped as $\mathcal{Z}(\Phi_{:B})$, for each $B \in \mathcal{B}$. In particular,*

$$\text{volume}(\mathcal{Z}(\Phi)) = \sum_{B \in \mathcal{B}} \text{volume}(\mathcal{Z}(\Phi_{:B})) = \sum_{B \in \mathcal{B}} |\det \Phi_{:B}|. \quad (3.3.4)$$

Proof. First, for any $B \in \mathcal{B}$, $\mathcal{Z}(\Phi_{:B}) = \Phi_{:B}[0, 1]^N$ corresponds to the N -dimensional parallelopiped spanned by $\{\Phi_{:b}\}_{b \in B}$, so that

$$\text{volume}(\mathcal{Z}(\Phi_{:B})) = \sqrt{\det \Phi_{:B}^\top \Phi_{:B}} = |\det \Phi_{:B}| > 0.$$

The rest of the proof given by Dyer and Frieze (1994) is not crystal clear; for this reason we can only give the main ideas.

For any $x \in \mathcal{Z}(\Phi)$, Dyer and Frieze (1994) consider the following linear program (LP),

$$\begin{aligned} \min_y \quad & c^\top y \\ \text{s.t.} \quad & \Phi y = x \\ & 0 \leq y \leq 1 \end{aligned} \quad P_x(\Phi, c)$$

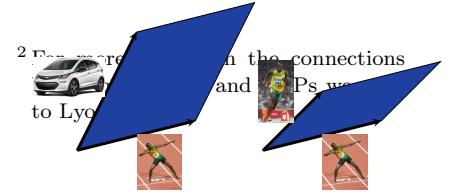


Figure 3.1: For $N = 2$, each item is associated to a two-dimensional feature vector. Samples of DPP(\mathbf{K}) can be represented by a parallelogram, and have likelihood (3.3.2) proportional to the squared area.

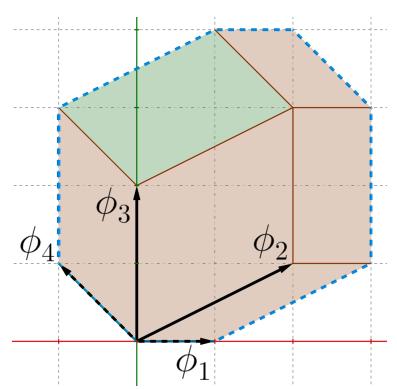


Figure 3.2: $\mathcal{Z}(\Phi)$ where

$$\Phi = \begin{bmatrix} 1 & 2 & 0 & -1 \\ 0 & 1 & 2 & 1 \end{bmatrix}.$$

where the linear objective $c \notin \text{span } \Phi^\top$. We propose to consider the corresponding standard formulation

$$\begin{aligned} \min_{y,z} \quad & [c^\top \ 0] \begin{bmatrix} y \\ z \end{bmatrix}, \\ \text{s.t.} \quad & \begin{bmatrix} \Phi & 0_{N,M} \\ I_M & I_M \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} x \\ 1 \end{bmatrix}, \\ & y \geq 0, z \geq 0, \end{aligned} \quad (3.3.5)$$

to give more details on the analysis of $P_x(\Phi, c)$. Solving (3.3.5) using the simplex method, see, e.g., Luenberger and Ye (2016, Chapter 3), allows us to focus on the so-called optimal basic solution vector (y^*, z^*) , which has exactly $M + N$ nonzero components. A first observation is that $B_x \triangleq Y_{[0,1]} = \{i \mid 0 < y_i^* < 1\}$ has cardinality N .³ Indeed, if we further denote $Y_0 = \{i \mid y_i^* = 0\}$, $Y_1 = \{i \mid y_i^* = 1\}$ and use the same notations for z^* , the number of nonzero components of the optimal solution vector $(y^*, z^* = 1 - y^*)$ reads

$$\mathcal{M} + N = |Y_{[0,1]}| + |\underbrace{Z_{[0,1]}}_{=Y_0}| + |Y_1| + |\underbrace{Z_1}_{=Y_0}| = |B_x| + \mathcal{M}.$$

Next we prove that $B_x \in \mathcal{B}$, by showing that $\det \Phi_{:B_x} \neq 0$. To see this, we first highlight the columns of the constraint matrix of (3.3.5) associated to **basic** and **non-basic** variables of $(y^*, 1 - y^*)$

$$\left[\begin{array}{ccc|ccc} \Phi_{:B_x} & \Phi_{:Y_1} & \Phi_{:Y_0} & 0_N & 0_{N,|Y_1|} & 0_{N,|Y_0|} \\ \hline I_N & 0 & 0 & I_N & 0 & 0 \\ 0 & I_{|Y_1|} & 0 & 0 & I_{|Y_1|} & 0 \\ 0 & 0 & I_{|Y_0|} & 0 & 0 & I_{|Y_0|} \end{array} \right],$$

and note

$$\mathbf{B} = \left[\begin{array}{c|cc} \Phi_{:B_x} & 0_N & \Phi_{:Y_1} & 0_{N,|Y_0|} \\ \hline I_N & & & \\ 0_{|Y_1|,N} & & I_M & \\ 0_{|Y_0|,N} & & & \end{array} \right] \quad \text{and} \quad \mathbf{N} = \left[\begin{array}{cc} \Phi_{:Y_0} & 0_{N,|Y_1|} \\ 0_{N,|Y_0|} & 0_{N,|Y_1|} \\ 0_{|Y_1|,|Y_0|} & I_{|Y_1|} \\ I_{|Y_0|} & 0_{|Y_0|,|Y_1|} \end{array} \right].$$

Since the matrix \mathbf{B} is associated to basic variables, it is invertible and Lemma 1.B.1 gives

$$\begin{aligned} \det \mathbf{B} &= \det I_M \times \det (\Phi_{:B_x} - \underbrace{[0_N, \Phi_{:Y_1}, 0_{N,|Y_0|}] I_M^{-1} [I_N, 0_{N,|Y_1|}, 0_{N,|Y_0|}]^\top}_{=0_N}) \\ &= \det \Phi_{:B_x} \neq 0. \end{aligned}$$

This shows that, solving $P_x(\Phi, c)$ allows to locate any point $x \in \mathcal{Z}(\Phi)$, as falling inside the region⁴

$$x \in \Phi \underline{y}^* + \mathcal{Z}(\Phi_{:B_x}), \quad (3.3.6)$$

where $0 < y_{B_x}^* < 1$, and $\underline{y}^* \in \{0, 1\}^M$ such that $\underline{y}_i^* \triangleq \begin{cases} 0, & \text{if } i \in B_x, \\ y_i^*, & \text{if } i \notin B_x. \end{cases}$

Besides, the optimality conditions of the standard formulation (3.3.5)

³Note that $N = \text{rank } \Phi$ also corresponds to the cardinality of the samples of our projection DPP(\mathbf{K}). The set B_x will actually materialize the DPP samples, see also Algorithm 7.

⁴For example, in Figure 3.2, the green region can be described as $\Phi(0, 0, 1, 0)^\top + \mathcal{Z}(\Phi_{:\{2,4\}})$.

read

$$\begin{aligned} & [c_{Y_0}^\top, 0_{1,|Y_1|}] - [c_{B_x}^\top, 0_{1,N}, c_{Y_1}^\top, 0_{1,|Y_1|}] \mathbf{B}^{-1} \mathbf{N} \geq 0 \\ \iff & \begin{cases} c_{Y_0}^\top - c_{B_x}^\top \Phi_{:B_x}^{-1} \Phi_{:Y_0} \geq 0, \text{ and} \\ c_{Y_1}^\top - c_{B_x}^\top \Phi_{:B_x}^{-1} \Phi_{:Y_1} \leq 0. \end{cases} \end{aligned} \quad (3.3.7)$$

In particular, consider x' that belongs to the same region (3.3.6) as x , i.e., $x' \in \Phi \underline{y}^* + \mathcal{Z}(\Phi_{:B_x})$, and such that the optimal solution y' of $P_{x'}$ satisfies $0 < y'^*_{B_x} < 1$. Then $P_{x'}$ and P_x share the same optimality conditions (3.3.7), so that $B_{x'} = B_x$ and $\underline{y}'^* = \underline{y}^*$.

More generally, Dyer and Frieze (1994) show that any $B \in \mathcal{B}$ is associated to a unique region of $\mathcal{Z}(\Phi)$ described as a potentially shifted $\mathcal{Z}(\Phi_{:B})$. To see this, consider $\eta(B) \in \{0, 1\}^M$ such that $\eta(B)_B = 0$. Then, solving $P_x(\Phi, c)$ for all the points $x \in \mathcal{Z}(\Phi)$ falling strictly ($0 < x_B < 1$) in the region

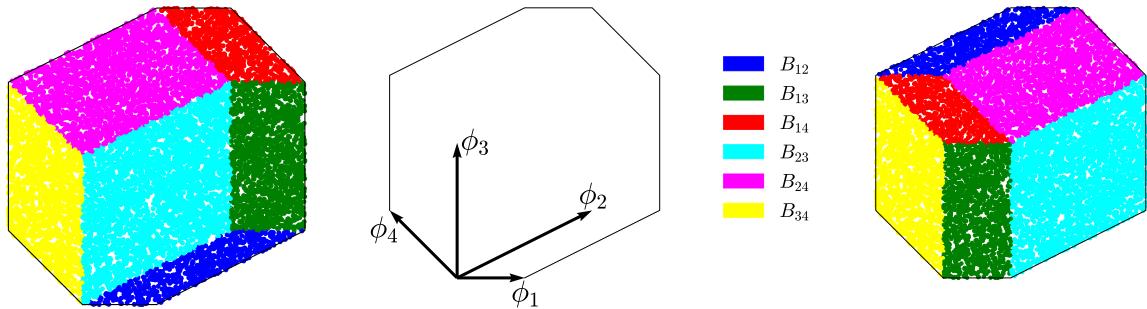
$$\Phi \eta(B) + \mathcal{Z}(\Phi_{:B}),$$

yields $B_x = B$ and $\xi(x) = \eta(B)$, so that $\eta(B)$ is unique. Finally, for a given linear objective $c \notin \text{span } \Phi^\top$, solving $P_x(\Phi, c)$ for all the points $x \in \mathcal{Z}(\Phi)$ provides the decomposition $\mathcal{Z}(\Phi) = \bigcup_{x \in \mathcal{Z}(\Phi)} \{\Phi \xi(x) + \mathcal{Z}(\Phi_{:B_x})\}$, which in turn gives rise to the partition

$$\mathcal{Z}(\Phi) = \bigsqcup_{B \in \mathcal{B}} \{\Phi \eta(B) + \mathcal{Z}(\Phi_{:B})\}.$$

□

We note that the tiling of $\mathcal{Z}(\Phi)$ is not unique, indeed different linear objectives $c \in \mathbb{R}^M$ may give different tilings, see Figure 3.3 for an illustration. An arbitrary c gives a valid tiling, as long as there are no ties when solving $P_x(\Phi, c)$. Dyer and Frieze (1994) use a nonlinear mathematical trick to fix c . In practice, we generate a random Gaussian vector c once and for all, which makes sure no ties appear during the execution, with probability one.



More specifically, the linear programming arguments used by Dyer and Frieze (1994) to prove this result were our main inspiration for devising our sampling strategy.

Remark 3.3.3. We propose to interpret the proof of Proposition 3.3.2 as a volume sampling algorithm: if one manages to sample an x uniformly on $\mathcal{Z}(\Phi)$ and extract the corresponding basis $B = B_x$ by solving $P_x(\Phi, c)$, then B is drawn with probability proportional to $\text{volume}(\Phi_{:B}) = |\det \Phi_{:B}|$.

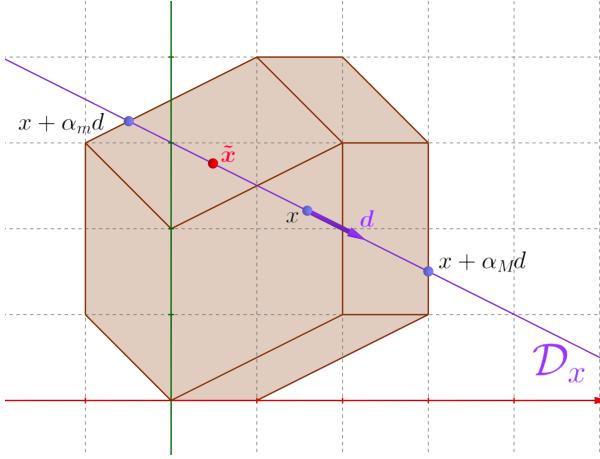
Figure 3.3: Different tilings obtained by solving $P_x(\Phi, c)$ with different linear objectives.

Remark 3.3.3 is close to what we want, but there is missing *square* exponent, compared to (3.3.2). In the rest of this section, we explain how to efficiently sample x uniformly on $\mathcal{Z}(\Phi)$, and how to change the volume into its square.

3.3.1 Hit-and-run and the Simplex Algorithm

$\mathcal{Z}(\Phi)$ is a convex set. Approximate uniform sampling on large-dimensional convex bodies is one of the core questions in MCMC, see e.g., Cousins and Vempala (2016) and Chen et al. (2018) and references therein. The hit-and-run chain (Turčin, 1971; Smith, 1984) is one of the preferred practical and theoretical solutions (Cousins and Vempala, 2016).

We describe the Markov kernel of the hit-and-run Markov chain for a generic target distribution π supported on a convex set C . Sample a point y uniformly on the unit sphere centered at x . Letting $d = y - x$, this defines the line $\mathcal{D}_x \triangleq \{x + \alpha d ; \alpha \in \mathbb{R}\}$. Then, sample z from any Markov kernel $Q(x, \cdot)$ supported on \mathcal{D}_x that leaves the restriction of π to \mathcal{D}_x invariant. In particular, Metropolis-Hastings kernel (MH, Robert and Casella, 2004) is often used with uniform proposal on \mathcal{D}_x , which favors large moves across the support C of the target, see Figure 3.4.



The resulting Markov kernel leaves π invariant, see e.g., Andersen and Diaconis (2007) for a general proof. Furthermore, the hit-and-run Markov chain has polynomial mixing time for log concave π (Lovász and Vempala, 2003, Theorem 2.1).

To implement Remark 3.3.3, the distribution to target on the zonotope is the uniform measure, i.e., $\pi_u \propto \mathbf{1}_{\mathcal{Z}(\Phi)}$, see Figure 3.5.

In practice, we can choose the secondary Markov kernel $Q(x, \cdot)$ to be MH with uniform proposal on \mathcal{D}_x , as long as we can determine the endpoints $x + \alpha_m(y - x)$ and $x + \alpha_M(y - x)$ of $\mathcal{D}_x \cap \mathcal{Z}(\Phi)$, see Figure 3.4. In fact, even an oracle saying whether a point belongs to the zonotope requires solving LPs (basically, it is Phase I of the simplex algorithm). As noted by Lovász and Vempala (2003, Section 4.4), hit-and-run with LP is the state-of-the-art method for computing the volume of large-scale zonotopes. Thus, by definition of $\mathcal{Z}(\Phi)$, this amounts to solving

Figure 3.4: A step of the Hit-and-Run chain. Starting at x the proposed point is \tilde{x}

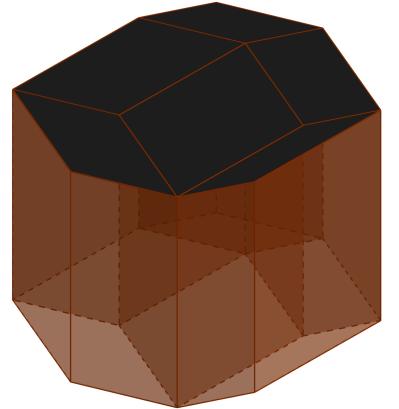


Figure 3.5: Uniform target distribution π_u on $\mathcal{Z}(\Phi)$. The probability of falling into a given tile is proportional to its volume. π_u is the limiting distribution of Algorithm 6.

Algorithm 6: `unifZonoHitAndRun` Generate an approximate sample of π_u , the uniform measure on $\mathcal{Z}(\Phi)$.

Require: Φ

- 1: $i \leftarrow 0$
 - 2: $x_0 \leftarrow \Phi u$ with $u \sim \mathcal{U}_{[0,1]^N}$
 - 3: **while** Not converged **do**
 - 4: Draw $d \sim \mathcal{U}_{\mathbb{S}^{r-1}}$ and let $\mathcal{D}_{x_i} \triangleq x_i + \mathbb{R}d$
 - 5: Draw $\tilde{x} \sim \mathcal{U}_{\mathcal{D}_{x_i} \cap \mathcal{Z}(A)}$ **# Solve 2 LPs, see (3.3.8)**
 - 6: $x_{i+1} \leftarrow \tilde{x}$
 - 7: $i \leftarrow i + 1$
 - 8: **end while**
-

Algorithm 7: `extractBasis` Solve $P_x(\Phi, c)$ and keep the indices of the basic variables.

Require: $\Phi, c, x \in \mathcal{Z}(\Phi)$

- 1: Compute y^* the optimal solution of $P_x(\Phi, c)$ **# 1 LP**
 - 2: $B \leftarrow \{i ; y_i^* \in]0, 1[\}$
 - 3: **return** B
-

two more LPs: α_m is the optimal solution to the linear program

$$\begin{array}{ll} \min_{\lambda \in \mathbb{R}^M, \alpha \in \mathbb{R}} & \alpha \\ \text{s.t.} & x + \alpha d = \Phi \lambda \\ & 0 \leq \lambda \leq 1, \end{array} \quad (3.3.8)$$

while α_M is the optimal solution of the same linear program with objective $-\alpha$. Thus, a combination of hit-and-run and LP solvers such as Dantzig's simplex algorithm (Luenberger and Ye, 2016) yields a Markov kernel with invariant distribution the uniform measure π_u . This is summarized in Algorithm 6. The acceptance in MH is 1 due to our choice of the proposal and the target. By the proof of Proposition 3.3.2, running Algorithm 6, taking the output chain (x_i) and extracting the bases (B_{x_i}) with Algorithm 7, we obtain a chain on \mathcal{B} with invariant distribution proportional to the volume $|\det \Phi_{:B}|$ of the tile associated to $B \in \mathcal{B}$.

In terms of theoretical performance, this Markov chain inherits Lovász and Vempala (2003) mixing time as it is a simple transformation of hit-and-run targeting the uniform distribution on a convex set. We underline that this is not a pathological case and it already covers a range of applications, as changing the feature matrix Φ yields another zonotope, but the target distribution on the zonotope stays uniform. Machine learning practitioners do not use volume sampling for diversity sampling yet, but nothing prevents it, as it already encodes the same feature-based diversity as squared volume sampling (i.e., DPPs). Nevertheless, our initial goal was to sample from the projection DPP(\mathbf{K}) given by (3.3.2). We now modify the Markov chain just constructed to achieve that.

 Algorithm 8: `volZonoHitAndRun`

Require: Φ, c, x, B

- 1: Draw $d \sim \mathcal{U}_{\mathbb{S}^{r-1}}$ and let $\mathcal{D}_x \triangleq x + \mathbb{R}d$
 - 2: Draw $\tilde{x} \sim \mathcal{U}_{\mathcal{D}_x \cap \mathcal{Z}(\Phi)}$ # Solve 2 LPs, see (3.3.8)
 - 3: $\tilde{B} \leftarrow \text{extractBasis}(\Phi, c, \tilde{x})$ # Solve 1 LP, see $P_x(\Phi, c)$
 - 4: Draw $u \sim \mathcal{U}_{[0,1]}$
 - 5: **if** $u < \frac{\text{volume}(\Phi_{:\tilde{B}})}{\text{volume}(\Phi_{:B})} = \left| \frac{\det \Phi_{:\tilde{B}}}{\det \Phi_{:B}} \right|$ **then**
 - 6: **return** \tilde{x}, \tilde{B}
 - 7: **else**
 - 8: **return** x, B
 - 9: **end if**
-

 Algorithm 9: `zonotopeSampler` Generate an approximate sample B of $\text{DPP}(\Phi^\top (\Phi \Phi^\top)^{-1} \Phi)$.

Require: Φ, c

- 1: $i \leftarrow 0$
 - 2: $x_i \leftarrow \Phi u$, with $u \sim \mathcal{U}_{[0,1]^N}$
 - 3: $B_i \leftarrow \text{extractBasis}(\Phi, c, x_i)$
 - 4: **while** Not converged **do**
 - 5: $x_{i+1}, B_{i+1} \leftarrow \text{volZonoHitAndRun}(\Phi, c, x_i, B_i)$
 - 6: $i \leftarrow i + 1$
 - 7: **end while**
-

3.3.2 From Volume to Squared Volume

Consider the probability density function on $\mathcal{Z}(\Phi)$

$$\pi_v(x) = \frac{|\det \Phi_{:B_x}|}{\det \Phi \Phi^\top} \mathbf{1}_{\mathcal{Z}(\Phi)}(x), \quad (3.3.9)$$

represented on our example in Figure 3.6. In particular, observe that π_v is constant on each tile. Running the hit-and-run algorithm with π_v as target instead of π_u in Section 3.3.1, and extracting bases using Algorithm 7 again, we obtain a Markov chain on \mathcal{B} with limiting distribution (3.3.2), as required. To see this, consider the tiling of $\mathcal{Z}(\Phi)$ and simply integrate π_v over the tile associated to each $B \in \mathcal{B}$.

The resulting algorithm is given in Algorithm 9. Note that, contrary to Algorithm 6 where the target of the hit-and-run algorithm was uniform, the subroutine Algorithm 8 now involves a rejection step.

3.4 EXPERIMENTS

We investigate the behavior of our Algorithm 9 on synthetic graphs in Section 3.4.1 and on a summary extraction task in Section 3.4.2.

3.4.1 Non-uniform Spanning Trees

We compare Algorithm 10 studied by Anari, Gharan, and Rezaei (2016) and Li, Jegelka, and Sra (2016b) and our Algorithm 9 on two types of graphs, in two different settings. The graphs we consider are

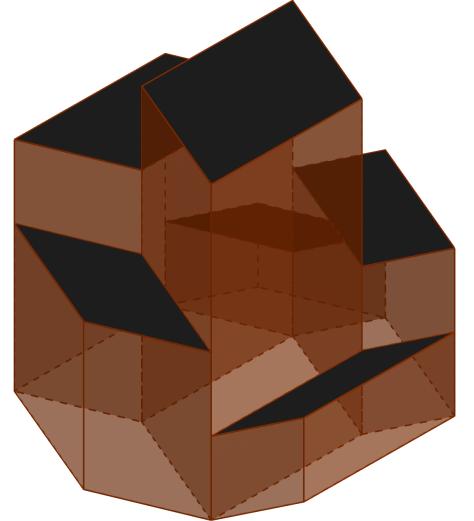


Figure 3.6: Target distribution π_v on $\mathcal{Z}(\Phi)$. The probability of falling in a given tile is proportional to its square volume. π_v is the limiting distribution of Algorithm 9.

 Algorithm 10: `basisExchangeSampler`

Require: Φ or \mathbf{K} , initial basis $B_0 \in \mathcal{B}$ (3.3.3)

- 1: $i \leftarrow 0$
- 2: **while** Not converged **do**
- 3: Draw $u \sim \mathcal{U}_{[0,1]}$
- 4: **if** $u < \frac{1}{2}$ **then**
- 5: Draw $s \sim \mathcal{U}_{B_i}$ and $t \sim \mathcal{U}_{[n] \setminus B_i}$
- 6: $P \leftarrow (B_i \setminus \{s\}) \cup \{t\}$
- 7: Draw $u' \sim \mathcal{U}_{[0,1]}$
- 8: **if** $u' < \frac{\text{volume}^2(\Phi_{:P})}{\text{volume}^2(\Phi_{:B_i}) + \text{volume}^2(\Phi_{:P})} = \frac{\det \mathbf{K}_P}{\det \mathbf{K}_{B_i} + \det \mathbf{K}_P}$ **then**
- 9: $B_{i+1} \leftarrow P$
- 10: **else**
- 11: $B_{i+1} \leftarrow B_i$
- 12: **end if**
- 13: **else**
- 14: $B_{i+1} \leftarrow B_i$
- 15: **end if**
- 16: $i \leftarrow i + 1$
- 17: **end while**

the complete graph K_{10} with 10 vertices (and 45 edges) and a realization BA(20, 2) of a Barabási-Albert graph with 20 vertices and parameter 2. We chose BA as an example of structured graph, as it has the preferential attachment property present in social networks (Barabási and Albert, 1999). The input matrix Φ is a weighted version of the oriented vertex-edge incidence matrix of each graph, for which we keep only the 9 (resp. 19) first rows, so that Φ is indeed full row-rank. For more generality, we introduce artificially a weight vector,⁵ by reweighting the columns of Φ with i.i.d. uniform weights ω_m in $[0, 1]$. Samples from the corresponding projection DPP are thus spanning trees drawn proportionally to the products of their edge weights.

For Algorithm 9, a value of the linear objective c is drawn once and for all, for each graph, from a standard Gaussian distribution. This is enough to make sure no ties appear during the execution, as mentioned in Section 3.3. This linear objective is kept fixed throughout the experiments so that the tiling of the zonotope remains the same. We run both algorithms for 70 seconds, which corresponds to roughly 50 000 iterations of Algorithm 9. Moreover, we run 100 chains in parallel for each of the two algorithms. For each of the 100 repetitions, we initialize the two algorithms with the same random initial basis, obtained by solving $P_x(\Phi, c)$ once, with $x = \Phi u$ and $u \sim \mathcal{U}_{[0,1]^N}$. For both graphs, the total number $|\mathcal{B}|$ of bases is of order 10^8 , so computing total variation distances is impractical. We instead compare Algorithms 10 and 9 based on the estimation of inclusion probabilities $\mathbb{P}[S \subset B]$ for various subsets $S \subset [n]$ of size 3. We observed similar behaviors across 3-subsets, so we display here the typical behavior on a 3-subset.

The inclusion probabilities are estimated via a running average of

⁵ As in Definition 1.1.4.

the number of bases containing the subsets S . Figures 3.7(a) and 3.8(a) show the behavior of both algorithms vs. MCMC iterations for the complete graph K_{10} and a realization of $\text{BA}(20, 2)$, respectively. Figures 3.7(b) and 3.8(b) show the behavior of both algorithms vs. wall-clock time for the complete graph K_{10} and a realization of $\text{BA}(20, 2)$, respectively. In these four figures, bold curves correspond to the median of the relative errors, whereas the frontiers of colored regions indicate the first and last deciles of the relative errors.

In Figures 3.7(c) and 3.8(c) we compute the Gelman-Rubin statistic (Gelman and Rubin, 1992), also called the potential scale reduction factor (PSRF). We use the PSRF implementation of CODA (Karen et al., 2006) in R, on the 100 binary chains indicating the presence of the typical 3-subset in the current basis.

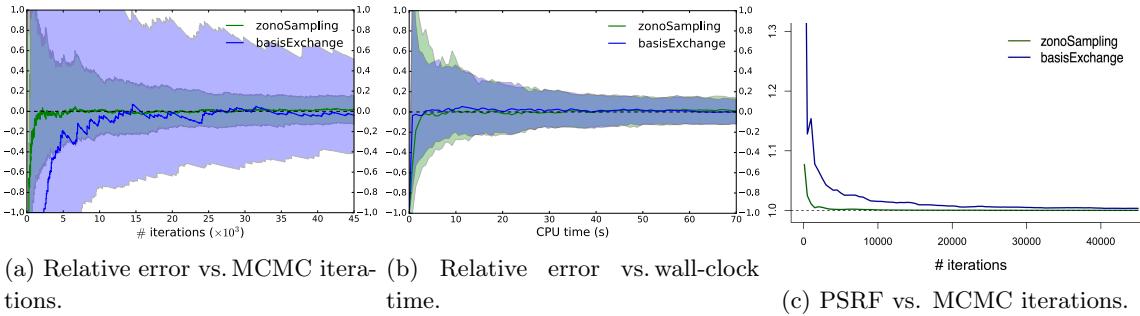


Figure 3.7: Comparison of Algorithms 10 and 9 on the complete graph K_{10} .

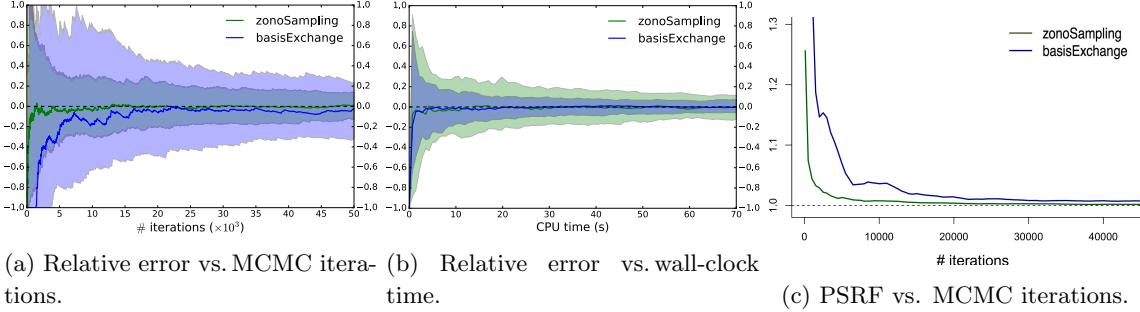


Figure 3.8: Comparison of Algorithms 10 and 9 on a realization of $\text{BA}(20, 2)$.

In terms of number of iterations, our Algorithm 9 clearly mixes faster. Relatedly, we observed typical acceptance rates for our algorithm an order of magnitude larger than Algorithm 10, while simultaneously attempting more global moves than the local basis-exchange moves of Algorithm 10. The high acceptance is partly due to the structure of the zonotope: the uniform proposal in the hit-and-run algorithm already favors bases with large determinants, as the length of the intersection of D_x in Algorithm 8 with any $\mathcal{Z}(\Phi_{:B})$ is an indicator of its volume, see also Figure 3.4.

Under the time-horizon constraint, see Figures 3.7(b) and 3.8(b), Algorithm 10 has time to perform more than 10^6 iterations compared to roughly 50 000 steps for our chain. The acceptance rate of Algo-

rithm 9 is still 10 times larger, but the time required to solve the linear programs at each MCMC iteration clearly hinders our algorithm in terms of CPU time. Both algorithms are comparable in performance, but given its large acceptance, we would expect our algorithm to perform better if it was allowed to do 10 times more iterations. Now this is implementation-dependent, and our current implementation of Algorithm 9 is relatively naive, calling the simplex algorithm in the GLPK (Oki, 2012) solver with CVXOPT (Andersen, Dahl, and Vandenberghe, 2008) from Python. We think there are big potential speed-ups to realize in the integration of linear programming solvers in our code. Moreover, we initialize our simplex algorithms randomly, while the different LPs we solve are related, so there may be additional smart mathematical speed-ups in using the path followed by one simplex instance to initialize the next.

Finally, we note that the performance of our Algorithm 9 seems stable and independent of the structure of the graph, while the performance of the basis-exchange Algorithm 10 seems more graph-dependent. Further investigation is needed to make stronger statements.

3.4.2 Text Summarization

Looking at Figures 3.7 and 3.8, our algorithm will be most useful when the bottleneck is mixing vs. number of iterations rather than CPU time. For instance, when integrating a costly-to-evaluate function against a projection DPP, the evaluation of the integrand may outweigh the cost of one iteration. To illustrate this, we adapt an experiment of Kulesza and Taskar (2012, Section 4.2.1) on minimum Bayes risk decoding for summary extraction. The idea is to find a subset Y of sentences of a text that maximizes

$$\frac{1}{R} \sum_{i=1}^R \text{ROUGE-1F}(Y, B_i), \quad (3.4.1)$$

where B_1, \dots, B_R are to be sampled from a projection DPP. ROUGE-1F is a measure of similarity of two sets of sentences. We consider making 11-sentence summaries of the 64-sentence article entitled *Scientists, Stop Thinking Explaining Science Will Fix Things* by subsampling the sentences of this text using a projection DPP with kernel of the form (3.3.1). Next, we describe how we build the corresponding 11×64 feature matrix Φ . For each sentence, we compute its number of characters and its number of words. Then, we apply a Porter stemmer (Steven Bird, Ewan Klein, and Edward Loper, 2009) and count again the number of characters and words in each sentence. In addition, we sum the tf-idf values of the words in each sentence and compute the average cosine distance to all other sentences. Finally, we compute the position of the sentence in the original article and generate binary features indicating positions 1–5.

In this setting, evaluating once ROUGE-1F in the sum (3.4.1) takes 0.1s on a modern laptop, while one iteration of our algorithm is $10^{-3}s$. Our Algorithm 9 can thus compute (3.4.1) for $R = 10\,000$ in about the

same CPU time as Algorithm 10, an iteration of which costs $10^{-5}s$. We show in Figure 3.9 the value of (3.4.1) for 3 possible summaries $(Y^{(i)})_{i=1}^3$ chosen uniformly at random in \mathcal{B} , over 50 independent runs. The variance of our estimates is smaller, and the number of different summaries explored is about 50%, against 10% for Algorithm 10. Evaluating (3.4.1) using our algorithm is thus expected to be closer to the maximum of the underlying integral.

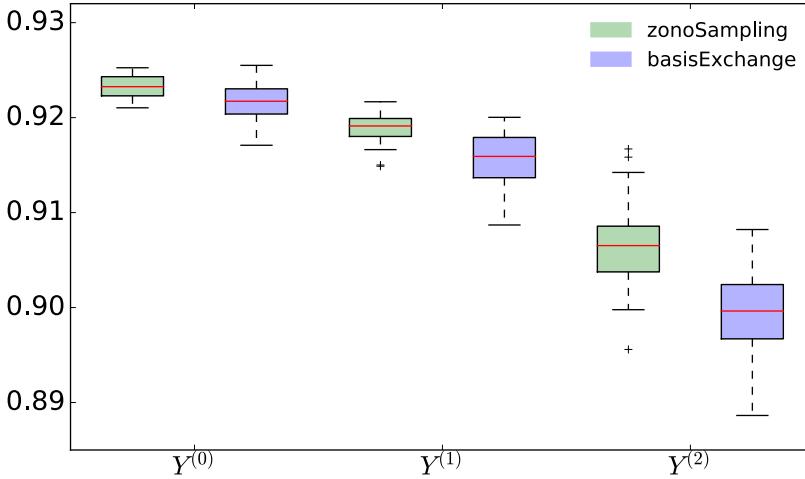


Figure 3.9: Summary extraction results

3.5 DISCUSSION

We proposed a new MCMC kernel with limiting distribution being an arbitrary projection DPP. This MCMC kernel leverages optimization algorithms to help making global moves on a convex body that represents the DPP. We provided empirical results supporting its fast mixing when compared to the state-of-the-art basis-exchange chain of Anari, Gharan, and Rezaei (2016) and Li, Jegelka, and Sra (2016b). Future work will focus on an efficient implementation: while our MCMC chain mixes faster, when compared based on CPU time our algorithm suffers from having to solve linear programs at each iteration. We note that answering the question whether a given point belongs to a zonotope involves linear programming, so that chord-finding procedures used in slice sampling (Neal, 2003, Sections 4 and 5) would not provide significant computational savings.

We also plan to investigate theoretical bounds on the mixing time of our Algorithm 8. We can build upon the work of Anari, Gharan, and Rezaei (2016), as our Algorithm 8 is also a weighted extension of our Algorithm 6, and the polynomial bounds for the vanilla hit-and-run algorithm (Lovász and Vempala, 2003) already apply to the latter. Note that while not targeting a DPP, our Algorithm 6 already samples items with feature-based repulsion, and could be used independently if the determinantal aspect is not crucial to the application.

Application of DPP sampling to Monte Carlo integration

4

This chapter presents our contribution¹ on the use of DPPs in the context of Monte Carlo integration.

NUMERICAL INTEGRATION IS A CORE TASK OF MACHINE LEARNING, INCLUDING MOST BAYESIAN METHODS (ROBERT, 2007). Both deterministic (Davis and Rabinowitz, 1984; Dick and Pillichshammer, 2010) and random (Robert and Casella, 2004) procedures have been proposed; see also Evans and Swartz (2000) for a survey. All methods require evaluating the integrand at carefully chosen points, called *quadrature nodes*, and combining these evaluations to minimize the approximation error.

Recently, a stream of work has made use of prior knowledge on the smoothness of the integrand using kernels. Oates, Girolami, and Chopin (2017) and Liu and Lee (2017) used kernel-based control variates, splitting the computational budget into regressing the integrand and integrating the residual. Bach (2017) looked for the best way to sample i.i.d. nodes and combine the resulting evaluations. Finally, Bayesian quadrature (O'Hagan, 1991; Huszár and Duvenaud, 2012; Briol et al., 2015), herding (Chen, Welling, and Smola, 2010; Bach, Lacoste-Julien, and Obozinski, 2012), or the biased importance sampling estimate of Delyon and Portier (2016) all favor *dissimilar* nodes, where dissimilarity is measured by a kernel. Our work falls in this last cluster.

WE BUILD ON THE PARTICULAR APPROACH OF BARDENET AND HARDY (2020) FOR MONTE CARLO INTEGRATION BASED ON PROJECTION DPPS. Fifteen years before Macchi (1975) even formalized DPPs, Ermakov and Zolotukhin (EZ, 1960) had the intuition to use a determinantal structure to sample quadrature nodes, followed by solving a linear system to aggregate the evaluations of the integrand into an unbiased estimator. This linear system yields a simple and interpretable characterization of the variance of their estimator. Ermakov and Zolotukhin's result did not diffuse much² in the Monte Carlo community, partly because the mathematical and computational machinery to analyze and implement it was not available. Unaware of this previous work, Bardenet and Hardy (2020) came up with a more natural estimator of the integral of interest, and they could build upon the thorough study of DPPs in random matrix theory (Johansson, 2006) to obtain a fast central limit theorem (CLT). Since then, DPPs with stationary kernels have also been used by Mazoyer, Coeurjolly, and Amblard (2019) and Coeurjolly, Mazoyer, and Amblard (2020) for

4.1 Standard quadrature	74
4.2 The multivariate Jacobi ensemble	74
4.3 Description of the two DPP-based estimators	75
A natural estimator	
The Ermakov-Zolotukhin estimator	
4.4 Sampling from orthogonal projection DPPs	79
Sampling from the multivariate Jacobi ensemble	
4.5 Empirical investigation	83
The bump experiment	
Integrating sums of eigenfunctions	
Further experiments	
4.6 Discussion	85
APPENDICES	
4.A Further experiments	87
Integrating absolute value	
Integrating Heaviside	
Integrating cosine	
Integrating a mixture of smooth and non smooth functions	

¹ G. Gautier, R. Bardenet, and M. Valko. 2019b. *On two ways to use determinantal point processes for Monte Carlo integration*. In Advances in Neural Information Processing Systems (NeurIPS).

github.com/guilgautier/DPPy

² Many thanks to Mathieu Gerber of Univ. Bristol, UK, for digging up this result from his human memory.

Monte Carlo integration. In any case, these DPP-based Monte Carlo estimators crucially depend on efficient sampling procedures for the corresponding (potentially multidimensional) DPP.

Our point is not to compare DPP-based Monte Carlo estimators to the wide choice of numerical integration algorithms, but to get a fine understanding of their properties so as to fine-tune their design and guide theoretical developments.

4.1 STANDARD QUADRATURE

Following Briol et al. (2015, Section 2.1), let $\mu(dx) = \omega(x)dx$ be a positive Borel measure on $\mathbb{X} \subset \mathbb{R}^d$ with finite mass and density ω w.r.t. the Lebesgue measure. Our goal is to compute integrals of the form $\int f(x)\mu(dx)$ for some test function $f : \mathbb{X} \rightarrow \mathbb{R}$. A quadrature rule approximates such integrals as a weighted sum of evaluations of f at some *nodes* $\{x_1, \dots, x_N\} \subset \mathbb{X}$,

$$\int f(x)\mu(dx) \approx \sum_{n=1}^N \omega_n f(x_n), \quad (4.1.1)$$

where the weights $\omega_n \triangleq \omega_n(x_1, \dots, x_N)$ do not need to be non-negative nor sum to one.

Among the many quadrature designs, see, e.g., Evans and Swartz (2000, Section 5), we pay special attention to the textbook example of the (deterministic) Gauss-Jacobi rule. This scheme applies to dimension $d = 1$, for $\mathbb{X} \triangleq [-1, 1]$ and $\omega(x) \triangleq (1-x)^a(1+x)^b$ with $a, b > -1$. In this case, the nodes $\{x_1, \dots, x_N\}$ are taken to be the zeros of p_N , the orthonormal Jacobi polynomial of degree N , and the weights $\omega_n \triangleq 1/K(x_n, x_n)$ with $K(x, x) \triangleq \sum_{k=0}^{N-1} p_k(x)^2$. In particular, this specific quadrature rule allows us to perfectly integrate polynomials up to degree $2N - 1$ (Davis and Rabinowitz, 1984, Section 2.7). In a sense, the DPPs of Bardenet and Hardy (2020) are a random, multivariate generalization of Gauss-Jacobi quadrature, as we shall see in Section 4.3.1.

Monte Carlo integration can be defined as random choices of nodes in (4.1.1). Importance sampling, for instance, corresponds to i.i.d. nodes, while Markov chain Monte Carlo corresponds to nodes drawn from a carefully chosen Markov chain; see, e.g., Robert and Casella (2004) for more details. Finally, quasi-Monte Carlo (QMC, Dick and Pillichshammer, 2010) applies to μ uniform over a compact subset of \mathbb{R}^d , and constructs deterministic nodes that spread uniformly, as measured by their *discrepancy*.

4.2 THE MULTIVARIATE JACOBI ENSEMBLE

In this part, we specify the kernel

$$K(x, y) = \sum_{k=0}^{N-1} \phi_k(x)\phi_k(y), \quad \text{where } \langle \phi_k, \phi_\ell \rangle = \delta_{k\ell}, \quad (4.2.1)$$

We recall the notation $\langle \phi_k, \phi_\ell \rangle \triangleq \int \phi_k(x)\phi_\ell(x)\mu(dx)$.

of the (orthogonal) projection DPP used in the following Monte Carlo methods. We follow Bardenet and Hardy (2020) and take multivariate orthonormal polynomials as eigenfunctions ϕ_k for the kernel. In dimension $d = 1$, letting $(\phi_k)_{k \geq 0}$ in (4.2.1) be the orthonormal polynomials w.r.t. μ results in a projection DPP called an *orthogonal polynomial ensemble* (OPE, König, 2004). When $d > 1$, orthonormal polynomials can still be uniquely defined by applying the Gram-Schmidt procedure to a set of monomials, provided the base measure is not pathological. However, there is no natural order on multivariate monomials: an ordering $\mathfrak{b} : \mathbb{N}^d \rightarrow \mathbb{N}$ must be picked before we apply Gram-Schmidt to the monomials in $L^2(\mu)$. We follow Bardenet and Hardy (2020, Section 2.1.3) and consider multi-indices $k \triangleq (k^1, \dots, k^d) \in \mathbb{N}^d$ ordered by their maximum degree $\max_i k^i$, and for constant maximum degree, by the usual lexicographic order. We still denote the corresponding multivariate orthonormal polynomials by $(\phi_k)_{k \in \mathbb{N}^d}$.

By multivariate OPE we mean the projection DPP with base measure $\mu(dx) \triangleq \omega(x)dx$ and orthogonal projection kernel $K(x, y) \triangleq \sum_{\mathfrak{b}(k)=0}^{N-1} \phi_k(x)\phi_k(y)$. When the base measure is separable, i.e., $\omega(x) = \omega^1(x^1) \times \dots \times \omega^d(x^d)$, multivariate orthonormal polynomials are products of univariate ones, and the kernel (4.2.1) reads

$$K(x, y) = \sum_{\mathfrak{b}(k)=0}^{N-1} \prod_{i=1}^d \phi_{k^i}^i(x^i)\phi_{k^i}^i(y^i), \quad (4.2.2)$$

where $(\phi_\ell^i)_{\ell \geq 0}$ are the orthonormal polynomials w.r.t. $\omega^i(z)dz$. For $\mathbb{X} = [-1, 1]^d$ and $\omega^i(z) = (1 - z)^{a^i}(1 + z)^{b^i}$, with $a^i, b^i > -1$, the resulting DPP is called a *multivariate Jacobi ensemble*.

4.3 DESCRIPTION OF THE TWO DPP-BASED ESTIMATORS

OUR GOAL IS TO DESIGN RANDOM QUADRATURE RULES ON $\mathbb{X} \triangleq [-1, 1]^d$ WITH DESIRABLE PROPERTIES. We focus on computing $\int f(x)\mu(dx)$ with the two unbiased DPP-based Monte Carlo estimators of Bardenet and Hardy (BH, 2020) and Ermakov and Zolotukhin (EZ, 1960). We start by presenting the natural BH estimator which, when associated to the multivariate Jacobi ensemble, comes with a CLT with a faster rate than classical Monte Carlo. Then, we survey the properties of the less obvious EZ estimator. Using the generalized Cauchy-Binet formula³ we provide a slight improvement of the key result of EZ. Despite the lack of result illustrating a fast convergence rate, the EZ estimator has a practical and interpretable variance.

4.3.1 A natural estimator

For $f \in L^1(\mu)$, Bardenet and Hardy (2020) consider

$$\widehat{I}_N^{\text{BH}}(f) \triangleq \sum_{n=1}^N \frac{f(\mathbf{x}_n)}{K(\mathbf{x}_n, \mathbf{x}_n)}, \quad (4.3.1)$$

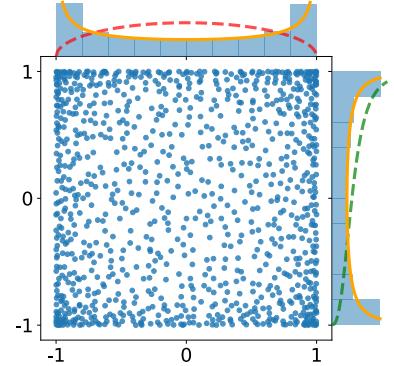


Figure 4.1: A sample of a 2D Jacobi ensemble with $N = 1000$ points. The normalized reference densities, proportional to $\omega^1(x) = (1 - x)^{a^1}(1 + x)^{b^1}$ and $\omega^2(y) = (1 - y)^{a^2}(1 + y)^{b^2}$, are displayed in dashed lines. The empirical marginal densities which converges to the arcsine density $\omega_{\text{eq}}(x) = \frac{1}{\pi\sqrt{1-x^2}}$ is plotted in solid line.

³ See Proposition 1.C.2.

as an unbiased estimator of $\int f(x)\mu(dx)$, with variance⁴

$$\mathbb{V}\text{ar}\left[\widehat{I}_N^{\text{BH}}(f)\right] = \frac{1}{2} \int \left(\frac{f(x)}{K(x,x)} - \frac{f(y)}{K(y,y)} \right)^2 K(x,y)^2 \mu(dx)\mu(dy), \quad (4.3.2)$$

which clearly captures a notion of smoothness of f w.r.t. K but its interpretation is not obvious.

For $\mathbb{X} = [-1, 1]^d$, the interest in multivariate Jacobi ensemble among DPPs comes from the fact that (4.3.1) can be understood as a (randomized) multivariate counterpart of the Gauss-Jacobi quadrature introduced in Section 4.1. Moreover, for f essentially C^1 , Bardenet and Hardy (2020, Theorem 2.7) proved a CLT with faster-than-classical-Monte-Carlo decay,

$$\sqrt{N^{1+1/d}} \left(\widehat{I}_N^{\text{BH}}(f) - \int f(x)\mu(dx) \right) \xrightarrow[N \rightarrow \infty]{\text{law}} \mathcal{N}(0, \Omega_{f,\omega}^2), \quad (4.3.3)$$

with $\Omega_{f,\omega}^2 \triangleq \frac{1}{2} \sum_{k \in \mathbb{N}^d} (k^1 + \dots + k^d) \mathcal{F}_{\omega_{\text{eq}}}^f(k)^2$, where \mathcal{F}_g denotes the Fourier transform of g , and $\omega_{\text{eq}}(x) \triangleq 1/\prod_{i=1}^d \pi \sqrt{1 - (x^i)^2}$. In the fast CLT (4.3.3), the asymptotic variance also captures a notion of smoothness of f : $\Omega_{f,\omega}$ is a measure of the decay of the Fourier coefficients of the integrand.

4.3.2 The Ermakov-Zolotukhin estimator

We start by stating⁵ the main finding of Ermakov and Zolotukhin (1960). To the best of our knowledge, we are the first to make the connection with projection DPPs. Using the generalized Cauchy-Binet formula,⁶ we provide a simpler proof of the original result, along with a slight improvement. Finally, we apply the technique of Ermakov and Zolotukhin (1960) to build an unbiased estimator of $\int f(x)\mu(dx)$, which comes with a practical and interpretable variance.

Theorem 4.3.1. *Take $f \in L^2(\mu)$ and let $\phi_0, \dots, \phi_{N-1}$ be orthonormal functions with respect to μ . Consider $\{\mathbf{x}_1, \dots, \mathbf{x}_N\} \sim \text{DPP}(\mu, K)$, with kernel $K(x, y) = \sum_{k=0}^{N-1} \phi_k(x)\phi_k(y)$ and build the linear system*

$$\begin{bmatrix} \phi_0(\mathbf{x}_1) & \dots & \phi_{N-1}(\mathbf{x}_1) \\ \vdots & & \vdots \\ \phi_0(\mathbf{x}_N) & \dots & \phi_{N-1}(\mathbf{x}_N) \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_N) \end{bmatrix}. \quad (4.3.4)$$

Then, the solution vector of (4.3.4) is unique, μ -almost surely, with coordinates given by Cramer's rules:

$$y_k = \frac{\det \Phi_{\phi_{k-1}, f}(\mathbf{x}_{1:N})}{\det \Phi(\mathbf{x}_{1:N})}, \quad (4.3.5)$$

where $\Phi_{\phi_{k-1}, f}(\mathbf{x}_{1:N})$ is the matrix obtained by replacing the k -th column of $\Phi(\mathbf{x}_{1:N})$ by $f(\mathbf{x}_{1:N})$. Moreover, for all $1 \leq k \leq N$, we have

$$\mathbb{E}[y_k] = \langle f, \phi_{k-1} \rangle, \quad \text{and} \quad (4.3.6)$$

$$\mathbb{V}\text{ar}[y_k] = \|f\|^2 - \sum_{\ell=0}^{N-1} \langle f, \phi_\ell \rangle^2. \quad (4.3.7)$$

In addition,⁷ we have that $\text{Cov}[y_j, y_k] = 0$, for all $1 \leq j \neq k \leq N$.

⁴ See Proposition 1.E.1.

⁵ See also Evans and Swartz (2000, Section 6.4.3) and references therein.

⁶ See Proposition 1.C.2.

Cramer's rules applied to the system $\begin{bmatrix} 1 & -2 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -3 \\ 5 \end{bmatrix}$ yield
 $x = \frac{\det \begin{bmatrix} -3 & -2 \\ 3 & 1 \end{bmatrix}}{\det \begin{bmatrix} 1 & -2 \\ 3 & 1 \end{bmatrix}} = 1$, and
 $y = \frac{\det \begin{bmatrix} 1 & -3 \\ 1 & -2 \end{bmatrix}}{\det \begin{bmatrix} 1 & -2 \\ 3 & 1 \end{bmatrix}} = 2$.

⁷ This is our slight improvement of the original result.

Before we provide the proof, several remarks are in order. We start by considering a function $f \triangleq \sum_{k=0}^{M-1} \langle f, \phi_k \rangle \phi_k$, $1 \leq M \leq \infty$, where $(\phi_k)_{k \geq 0}$ forms an orthonormal basis of $L^2(\mu)$, e.g., the Fourier basis or wavelet bases (Mallat and Peyré, 2009). Next, we build the orthogonal projection kernel K onto $\mathcal{H}_N \triangleq \text{span}\{\phi_0, \dots, \phi_{N-1}\}$, i.e., $K(x, y) = \sum_{k=0}^{N-1} \phi_k(x) \phi_k(y)$. Then, Theorem 4.3.1 shows that solving (4.3.4), with points $\{\mathbf{x}_1, \dots, \mathbf{x}_N\} \sim \text{DPP}(\mu, K)$, provides unbiased estimates of the N Fourier-like coefficients $(\langle f, \phi_k \rangle)_{k=0}^{N-1}$. Remarkably, these estimates are uncorrelated and have the same variance (4.3.7) equal to the residual $\sum_{k=N}^{\infty} \langle f, \phi_k \rangle^2$. The faster the decay of the coefficients, the smaller the variance. In particular, for $M \leq N$, i.e., $f \in \mathcal{H}_N$, the estimators have zero variance. Put differently, f can be reconstructed perfectly from only one sample of $\text{DPP}(\mu, K)$.

Proof of Theorem 4.3.1. First, observe that the joint distribution (2.1.6) of $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ can be written as

$$\frac{1}{N!} (\det \Phi(x_{1:N}))^2 \mu^{\otimes N}(dx). \quad (4.3.8)$$

Thus, $\det \Phi(\mathbf{x}_{1:N}) \neq 0$, μ -almost surely. Hence, the matrix $\Phi(\mathbf{x}_{1:N})$ defining the linear system (4.3.4) is invertible, μ -almost surely. The expression of the coordinates (4.3.5) follows from Cramer's rule. Then, we treat the case $k = 1$, the others follow the same lines. The proof relies on the orthonormality of the ϕ_k s. The expectation (4.3.7) reads

$$\begin{aligned} \mathbb{E} \left[\frac{\det \Phi_{\phi_0, f}(\mathbf{x}_{1:N})}{\det \Phi(\mathbf{x}_{1:N})} \right] &= \frac{1}{N!} \int \det \Phi_{\phi_0, f}(x_{1:N}) \det \Phi(x_{1:N}) \mu^{\otimes N}(dx) && \text{Using (4.3.8).} \\ &= \det \begin{bmatrix} \langle f, \phi_0 \rangle & (\langle f, \phi_\ell \rangle)_{\ell=1}^{N-1} \\ (\langle \phi_k, \phi_0 \rangle)_{k=1}^{N-1} & (\langle \phi_k, \phi_\ell \rangle)_{k,\ell=1}^{N-1} \end{bmatrix} \\ &= \det \begin{bmatrix} \langle f, \phi_0 \rangle & (\langle f, \phi_\ell \rangle)_{\ell=1}^{N-1} \\ 0_{N-1,1} & I_{N-1} \end{bmatrix} \\ &= \langle f, \phi_0 \rangle. && \text{By Cauchy-Binet formula (1.C.2).} \\ &&& \text{Using the orthonormality of } (\phi_k). \end{aligned} \quad (4.3.9)$$

Similarly, the second moment reads

$$\begin{aligned} \mathbb{E} \left[\left(\frac{\det \Phi_{\phi_0, f}(\mathbf{x}_{1:N})}{\det \Phi(\mathbf{x}_{1:N})} \right)^2 \right] &= \frac{1}{N!} \int \det \Phi_{\phi_0, f}(x_{1:N}) \det \Phi_{\phi_0, f}(x_{1:N}) \mu^{\otimes N}(dx) && \text{Using (4.3.8).} \\ &= \det \begin{bmatrix} \|f\|^2 & (\langle f, \phi_\ell \rangle)_{\ell=1}^{N-1} \\ (\langle f, \phi_k \rangle)_{k=1}^{N-1} & I_{N-1} \end{bmatrix} \\ &= \|f\|^2 - \sum_{k=1}^{N-1} \langle f, \phi_k \rangle^2. && \text{Using Cauchy-Binet formula (1.C.2) and the orthonormality of } (\phi_k). \end{aligned} \quad (4.3.10)$$

Finally, the variance (4.3.7) = (4.3.10) - (4.3.9)². With the same arguments, we can compute the covariance $\text{Cov}[y_j, y_k]$. We treat only the case $j = 1, k = 2$, the general case follows the same lines.

$$\begin{aligned}
& \text{Cov}[y_1, y_2] \\
&= \mathbb{E} \left[\frac{\det \Phi_{\phi_0, f}(\mathbf{x}_{1:N})}{\det \Phi(\mathbf{x}_{1:N})} \frac{\det \Phi_{\phi_1, f}(\mathbf{x}_{1:N})}{\det \Phi(\mathbf{x}_{1:N})} \right] - \mathbb{E} \left[\frac{\det \Phi_{\phi_0, f}(\mathbf{x}_{1:N})}{\det \Phi(\mathbf{x}_{1:N})} \right] \mathbb{E} \left[\frac{\det \Phi_{\phi_1, f}(\mathbf{x}_{1:N})}{\det \Phi(\mathbf{x}_{1:N})} \right] \\
&= \frac{1}{N!} \int \det \Phi_{\phi_0, f}(x_{1:N}) \det \Phi_{\phi_1, f}(x_{1:N}) \mu^{\otimes N}(dx) - \langle f, \phi_0 \rangle \langle f, \phi_1 \rangle \quad \text{Using (4.3.8).} \\
&= \det \begin{bmatrix} \langle f, \phi_0 \rangle & \langle f, f \rangle & (\langle f, \phi_\ell \rangle)_{\ell=2}^{N-1} \\ (\langle \phi_k, \phi_0 \rangle)_{k=1}^{N-1} & (\langle \phi_k, f \rangle)_{k=1}^{N-1} & (\langle \phi_k, \phi_\ell \rangle)_{k=1, \ell=2}^{N-1} \end{bmatrix} - \langle f, \phi_0 \rangle \langle f, \phi_1 \rangle \quad \text{By Cauchy-Binet formula (1.C.2).} \\
&= \det \begin{bmatrix} \langle f, \phi_0 \rangle & \|f\|^2 & (\langle f, \phi_\ell \rangle)_{\ell=2}^{N-1} \\ 0 & \langle \phi_1, f \rangle & 0 \\ 0_{N-2,1} & (\langle \phi_k, f \rangle)_{k=2}^{N-1} & I_{N-2} \end{bmatrix} - \langle f, \phi_0 \rangle \langle f, \phi_1 \rangle \quad \text{Using the orthonormality of } (\phi_k). \\
&= \langle f, \phi_0 \rangle \langle f, \phi_1 \rangle - \langle f, \phi_0 \rangle \langle f, \phi_1 \rangle = 0.
\end{aligned}$$

□

Corollary 4.3.2. *In the setting of Theorem 4.3.1, if ϕ_0 is constant, then $\frac{y_1}{\phi_0}$ defines an unbiased estimator of $\int_{\mathbb{X}} f(x) \mu(dx)$ with variance equal to $\mu(\mathbb{X}) \times (4.3.7)$. In addition, this estimator can be seen as a random quadrature rule (4.1.1) with weights summing to $\mu(\mathbb{X})$.*

Proof. We readily have $\mathbb{E} \left[\frac{y_1}{\phi_0} \right] = \frac{1}{\phi_0} \mathbb{E}[y_1] = \frac{1}{\phi_0} \langle f, \phi_0 \rangle = \int_{\mathbb{X}} f(x) dx$.

Then, since ϕ_0 is constant with unit norm we have $\phi_0 = \mu(\mathbb{X})^{-1/2}$ and the variance reads $\text{Var} \left[\frac{y_1}{\phi_0} \right] = \frac{1}{\phi_0^2} \text{Var}[y_1] = \mu(\mathbb{X}) \times (4.3.7)$. In addition, we can rewrite

$$\begin{aligned}
\frac{y_1}{\phi_0} &= \frac{1}{\phi_0} \frac{\det \Phi_{\phi_{k-1}, f}(\mathbf{x}_{1:N})}{\det \Phi(\mathbf{x}_{1:N})} = \frac{1}{\phi_0^2} \frac{\det \Phi_{\phi_0, f}(\mathbf{x}_{1:N})}{\det \Phi_{\phi_0, 1}(\mathbf{x}_{1:N})} \quad (4.3.11) \\
&= \mu(\mathbb{X}) \frac{\det \Phi_{\phi_0, f}(\mathbf{x}_{1:N})}{\det \Phi_{\phi_0, 1}(\mathbf{x}_{1:N})},
\end{aligned}$$

and the expansion of the numerator w.r.t. the first column yields

$$\frac{y_1}{\phi_0} = \sum_{n=1}^N f(\mathbf{x}_n) \underbrace{\frac{\mu(\mathbb{X})}{\det \Phi_{\phi_0, 1}(\mathbf{x}_{1:N})} (-1)^{1+n} \det(\phi_k(x_p))_{k=1, p=1 \neq n}^{N-1, N}}_{\triangleq \omega_n(\mathbf{x}_{1:N})}. \quad (4.3.12)$$

Note that there is a priori no reason for the weights to be nonnegative.

Finally, summing the weights gives

$$\sum_{n=1}^N \omega_n(\mathbf{x}_{1:N}) = \underbrace{\frac{\mu(\mathbb{X})}{\det \Phi_{\phi_0, 1}(\mathbf{x}_{1:N})}}_{=\det \Phi_{\phi_0, 1}(\mathbf{x}_{1:N})} \sum_{n=1}^N (-1)^{1+n} \det(\phi_k(x_p))_{k=1, p=1 \neq n}^{N-1, N}.$$

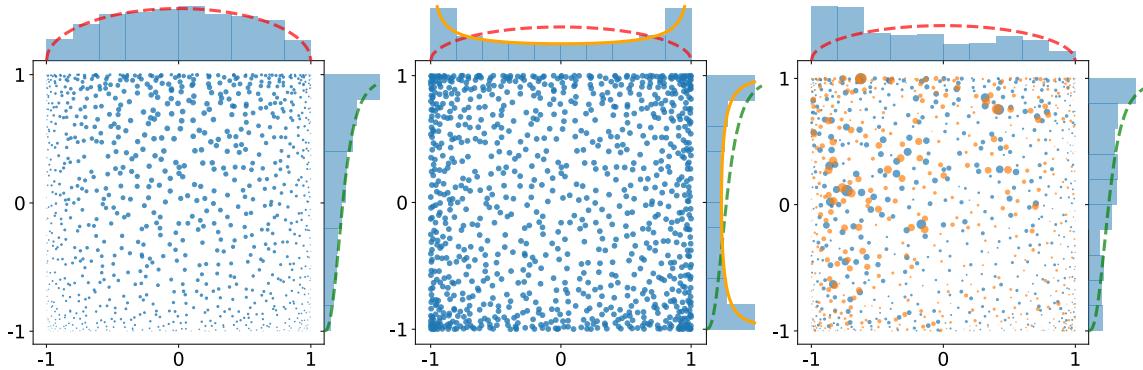
□

In the framework of the Jacobi ensemble described in Section 4.2, we indeed have ϕ_0 constant. We use Corollary 4.3.2 to define our second DPP-based estimator:

$$\widehat{I}_N^{\text{EZ}}(f) \triangleq \mu([-1, 1])^{1/2} \frac{\det \Phi_{\phi_{k-1}, f}(\mathbf{x}_{1:N})}{\det \Phi(\mathbf{x}_{1:N})}. \quad (4.3.13)$$

Unlike the variance of $\widehat{I}_N^{\text{BH}}(f)$ in (4.3.2), the variance of $\widehat{I}_N^{\text{EZ}}(f)$ clearly reflects the accuracy of the approximation of f by its projection onto \mathcal{H}_N . In particular, it allows us to integrate and interpolate polynomials up to “degree” $b^{-1}(N - 1)$, perfectly. Nonetheless, its limiting theoretical properties, like a CLT, look hard to establish. Indeed, the facts that the estimator has zero variance on the class of functions belonging to \mathcal{H}_N and that each quadrature weight depends on all quadrature nodes make it a peculiar object that doesn’t fit the assumptions of traditional CLTs for DPPs (Soshnikov, 2000).

Figure 4.2 displays a sample of a $d = 2$ Jacobi ensemble with $N = 1000$ points and compares how each estimator would reweight the points.



4.4 SAMPLING FROM ORTHOGONAL PROJECTION DPPs

In this section, we first give generic guidelines for a practical implementation of the chain-rule-based algorithm of Hough et al. (2006, Algorithm 18) for the simulation of orthogonal projection DPPs. For simplicity, we consider the setting of Proposition 2.1.2. We then tailor this procedure to the multivariate Jacobi ensemble.

We let $\mu(dx) = \omega(x)dx$ and consider a real-valued orthogonal projection kernel K , such that

$$K(x, y) = \sum_{k=1}^N \phi_k(x)\phi_k(y) \quad (4.4.1)$$

$$= \Phi(x)^\top \Phi(y), \quad (4.4.2)$$

where $\Phi(x) \triangleq (\phi_1(x), \dots, \phi_N(x))^\top$.

Figure 4.2: (middle) Same sample as in Figure 4.1 Then we plot the same sample where the disk centered at \mathbf{x}_n has now an area proportional to: (left) the weight $1/K(\mathbf{x}_n, \mathbf{x}_n)$ of $\widehat{I}_N^{\text{BH}}$ in (4.3.1), observe that these weights serve as a proxy for the reference measure, like Gaussian quadrature. (right) the weight of $\widehat{I}_N^{\text{EZ}}$ given by (4.3.12), observe that they can be either positive (blue disks) or negative (orange disks). The histogram of the absolute value of the weights is plotted on the marginal axes.

Algorithm 11: Generate a sample \mathbf{X} from an *orthogonal projection* DPP(μ, K) with N points

Require: Φ

```

1:  $\mathbf{X}, \mathbf{C}, \mathbf{d} = \emptyset, \text{zeros}(N, N), 0$ 
2: for  $n$  in range( $N$ ) do
3:   while not Accepted do
4:     Sample  $x \sim \frac{1}{N} K(x, x) \omega(x) dx$ 
5:      $\mathbf{C}[:, n] = \Phi(x)$ 
6:      $\mathbf{K}_{xx} = \mathbf{C}[:, n]^T \mathbf{C}[:, n]$ 
7:      $\mathbf{C}[:, n] -= \mathbf{C}[:, :n] \mathbf{C}[:, :n]^T \mathbf{C}[:, :n]$ 
8:      $\mathbf{d} = \mathbf{C}[:, n]^T \mathbf{C}[:, n]$ 
9:     if  $\text{rand()} < \frac{d}{\mathbf{K}_{xx}}$  then
10:       $\mathbf{C}[:, n] /= \sqrt{d}$ 
11:      Accepted
12:    end if
13:  end while
14:   $\mathbf{X} = \mathbf{X} \cup \{x\}$ 
15: end for
16: return  $\mathbf{X}$ 

```

WE APPLY THE CHAIN RULE USING A TWO-LAYER REJECTION SAMPLING SCHEME. In this scenario, we consider the one-point marginal distribution

$$\frac{1}{N} K(x, x) \omega(x) dx, \quad (4.4.3)$$

as unique proposal for sampling each conditional

$$\frac{K(x, x) - K(\mathbf{x}_{1:n-1}, x)^H K(\mathbf{x}_{1:n-1}, \mathbf{x}_{1:n-1})^{-1} K(\mathbf{x}_{1:n-1}, x)}{N - (n - 1)} \mu(dx) \quad (4.4.4)$$

$$= \frac{\text{distance}^2(\Phi(x), \text{span}\{\Phi(\mathbf{x}_1), \dots, \Phi(\mathbf{x}_{n-1})\})}{N - (n - 1)} \mu(dx). \quad (4.4.5)$$

Indeed, combining the fact that K is assumed Hermitian and that $\det K(\mathbf{x}_{1:n-1}, \mathbf{x}_{1:n-1}) \geq 0$ by (1.1.18), the quadratic form in (4.4.4) is non-negative and we can bound the numerator by $K(x, x)$. Thus, given an oracle generating samples from the marginal distribution (4.4.3), we can use a rejection sampling mechanism to sample from the successive conditional distributions. The rejection constant associated to the n -th conditional reads

$$\begin{aligned}
& \frac{(N - (n - 1))^{-1} (K(x, x) - K(\mathbf{x}_{1:n-1}, x)^H K(\mathbf{x}_{1:n-1}, \mathbf{x}_{1:n-1})^{-1} K(\mathbf{x}_{1:n-1}, x)) \omega(x)}{N^{-1} K(x, x) \omega(x)} \\
&= \frac{N}{N - (n - 1)} \frac{K(x, x) - K(\mathbf{x}_{1:n-1}, x)^H K(\mathbf{x}_{1:n-1}, \mathbf{x}_{1:n-1})^{-1} K(\mathbf{x}_{1:n-1}, x)}{K(x, x)} \\
&\leq \frac{N}{N - (n - 1)}. \tag{4.4.6}
\end{aligned}$$

The overall procedure is summarized in Algorithm 11.

4.4.1 Sampling from the multivariate Jacobi ensemble

In the case of (multivariate) orthogonal polynomial ensembles,⁸ evaluations of the kernel can be performed using the Gram representation $K(x, y) = \Phi(x)^\top \Phi(y)$ and one can leverage the three-term recurrence relations satisfied by each of the univariate Jacobi polynomials $(\phi_\ell^i)_\ell$. This is what we do in our special case, we use the dedicated function `scipy.special.eval_jacobi` to evaluate, up to depth $\sqrt[d]{N}$, the three-term recurrence relations satisfied by each of the univariate Jacobi polynomials. Instead of calling the recursive routine internally to evaluate $\Phi(x)$, the corresponding $d\sqrt[d]{N}$ univariate polynomials or N multivariate polynomials could be stored in some way and evaluated pointwise on the fly. The preprocessing time and the memory required would increase but it might accelerate the evaluation of $\Phi(x)$.

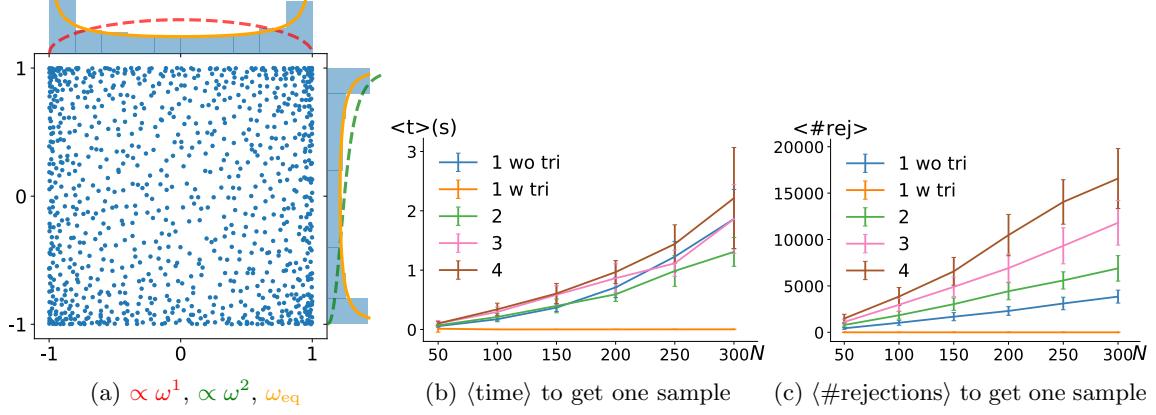
In this work, we take $\mathbb{X} = [-1, 1]^d$ and focus on sampling the multivariate Jacobi ensemble with parameters $|a^i|, |b^i| \leq 1/2$, cf. Section 4.2. We remodeled the original implementation⁹ of the multivariate Jacobi ensemble sampler accompanying the work of Bardenet and Hardy (BH, 2020) in a more Pythonic way. In particular, we address the previous challenges in the following way:

⁸ See Section 4.2.

⁹ github.com/rbardenet/dppmc.

1. contrary to BH, we leverage the Gram structure of the kernel to vectorize the computations and consider (4.4.5) instead of (4.4.4), and evaluate $K(x, y)$ via (4.4.2) instead of (4.2.2). The overall procedure is akin to a sequential Gram-Schmidt orthogonalization of the feature vectors $\Phi(x_1), \dots, \Phi(x_N)$. Moreover we pay special attention to avoiding unnecessary evaluations of orthogonal polynomials (OP) when computing a feature vector $\Phi(x)$. This is done by coupling the slicing feature of the Python language with the dedicated method `scipy.special.eval_jacobi`, used to evaluate the three-term recurrence relations satisfied by each of the univariate Jacobi polynomials. Given the chosen ordering \mathfrak{b} , the computation of $\Phi(x)$ requires the evaluation of d recurrence relations up to depth $\sqrt[d]{N}$.
2. like BH, we sample each conditional in turn using a rejection sampling mechanism with the same proposal distribution. But BH take as proposal $\omega_{\text{eq}}(x)dx$, which corresponds to the limiting marginal of the multivariate Jacobi ensemble as N goes to infinity; see Simon (2011, Section 3.11). On our side, we use a two-layer rejection sampling scheme. We rather sample from the n -th conditional using the marginal distribution $N^{-1}K(x, x)\omega(x)dx$ as proposal and rejection constant $N/(N - (n - 1))$. This allows us to reduce the number of (costly) evaluations of the acceptance ratio. The marginal distribution itself is sampled using the same proposal $\omega_{\text{eq}}(x)dx$ and rejection constant as BH. The rejection constant, of order 2^d , is derived in Proposition 4.4.1. We further reduced the number of OP evaluations by considering $N^{-1}K(x, x)\omega(x)dx$ as a mixture, where each component in (4.2.2) involves only one OP. In the end, the expected total number of rejections is of order $2^d N \log N$. Finally, we implemented a specific rejection free method for the univariate

Jacobi ensemble; a special continuous projection DPP which can be sampled exactly in $\mathcal{O}(N^2)$, by computing the eigenvalues of a random tridiagonal matrix (Killip and Nenciu, 2004, Theorem 2).



All these improvements resulted in dramatic speedups. For example, on a modern laptop, sampling a 2D Jacobi ensemble with $N = 1000$ points, see Figure 4.3, takes less than a minute, compared to hours previously.

In dimension $d = 1$, we implemented the random tridiagonal matrix model of Killip and Nenciu (2004, Theorem 2) to sample from the univariate Jacobi ensemble, with base measure $\mu(dx) = (1-x)^a(1+x)^b dx$, where $a, b > -1$.¹⁰ That is to say, this one dimensional continuous projection DPP with N points can be sampled in $\mathcal{O}(N^2)$, by computing the eigenvalues of a $N \times N$ random tridiagonal matrix with independent coefficients.

For $d \geq 2$, we detail the procedure for sampling exactly from the multivariate Jacobi ensemble with parameters $|a^i|, |b^i| \leq \frac{1}{2}$, for all $1 \leq i \leq d$. In other words we want to generate exact samples from the (orthogonal) projection DPP(μ, K) where

- $\mu(dx) = \omega(x)dx$, with

$$\omega(x) = \prod_{i=1}^d \omega^i(x^i), \quad \text{where} \quad \omega^i(z) = \prod_{i=1}^d (1-z)^{a^i} (1+z)^{b^i}, \quad \text{and} \quad |a^i|, |b^i| \leq \frac{1}{2}. \quad (4.4.7)$$

- $K(x, y) = \sum_{b(b)=0}^{N-1} \phi_k(x)\phi_k(y)$, with

$$\phi_k(x) = \prod_{i=1}^d \phi_{k^i}^i(x^i), \quad \text{where} \quad \int_{-1}^1 \phi_u^i(z)\phi_v^i(z)\omega^i(z)dz = \delta_{uv}. \quad (4.4.8)$$

Thus, sampling from (4.4.3) can be done in two steps:

- select a multi-index $k = \mathbf{b}^{-1}(n)$ with n drawn uniformly at random in $\{0, \dots, N-1\}$
- sample from $\phi_k(x)^2 \omega(x)dx$

We perform Step (ii) using rejection sampling with proposal

$$\omega_{\text{eq}}(x)dx = \prod_{i=1}^d \frac{1}{\pi \sqrt{1-(x^i)^2}} dx^i, \quad (4.4.9)$$

Figure 4.3: (a) A sample from a 2D Jacobi ensemble with $N = 1000$ points. (b)-(c) $a^i, b^i = -1/2$, the colors and numbers correspond to the dimension. For $d = 1$, the tridiagonal model (tri) of Killip and Nenciu offers tremendous time savings. (c) The total number of rejections grows as $2^d N \log(N)$.

¹⁰ See also Theorem 5.1.3.

which corresponds to the limiting marginal distribution of the multivariate Jacobi ensemble as N goes to infinity; see Simon (2011, Section 3.11) and Figure 4.1. The acceptance ratio writes

$$\begin{aligned} \frac{\phi_k(x)^2 \omega(x)}{\omega_{\text{eq}}(x)} &= \prod_{i=1}^d \frac{\phi_{k^i}^i(x^i)^2 \times (1-x^i)^{a^i}(1+x^i)^{b^i}}{\pi^{-1}(1-x^i)^{-\frac{1}{2}}(1+x^i)^{-\frac{1}{2}}} \\ &= \prod_{i=1}^d \pi(1-x^i)^{a^i+\frac{1}{2}}(1+x^i)^{b^i+\frac{1}{2}} \phi_{k^i}^i(x^i)^2. \end{aligned} \quad (4.4.10)$$

A more pragmatic reason which guided this choice of proposal lies in the following result.

Proposition 4.4.1. *Let $(\phi_k)_{k \geq 0}$ be the (univariate) orthonormal polynomials w.r.t. $(1-x)^a(1+x)^b dx$ with $|a| \leq \frac{1}{2}, |b| \leq \frac{1}{2}$.*

Then, for any $x \in [-1, 1]$ and $k \geq 1$,

$$\pi(1-x)^{a+\frac{1}{2}}(1+x)^{b+\frac{1}{2}} \phi_k(x)^2 \leq C_k \triangleq \frac{2 \Gamma(k+a+b+1) \Gamma(k+\max(a,b)+1)}{k! (k+\frac{a+b+1}{2})^{2\max(a,b)} \Gamma(k+\min(a,b)+1)}. \quad (4.4.11)$$

Each of the terms that appear in (4.4.10) can be bounded using the following recipe:

- For $k^i = 0$, ϕ_0^i is constant with unit norm, i.e.,

$$(\phi_0^i)^2 \int_{-1}^1 (1-x)^{a^i}(1+x)^{b^i} dx = 1 \iff (\phi_0^i)^2 = \frac{1}{2^{a^i+b^i+1} B(a^i+1, b^i+1)}, \quad (4.4.12)$$

so that the corresponding term in (4.4.10) becomes

$$\frac{\pi(1-x)^{a^i+\frac{1}{2}}(1+x)^{b^i+\frac{1}{2}}}{2^{a^i+b^i+1} B(a^i+1, b^i+1)} \leq \frac{\pi(1-m)^{a^i+\frac{1}{2}}(1+m)^{b^i+\frac{1}{2}}}{2^{a^i+b^i+1} B(a^i+1, b^i+1)} \triangleq C_{k^i=0} \leq 2, \quad (4.4.13)$$

where

$$m = \underset{-1 \leq x \leq 1}{\operatorname{argmax}} (1-x)^{a^i+\frac{1}{2}}(1+x)^{b^i+\frac{1}{2}} = \begin{cases} 0, & \text{if } a^i = b^i = -\frac{1}{2}, \\ \frac{b^i-a^i}{a^i+b^i+1}, & \text{otherwise.} \end{cases}$$

- For $k^i \geq 1$, we use the bound $C_{k^i \geq 1}$ (4.4.11) provided originally by Chow, Gatteschi, and Wong (1994). As mentioned by Gautschi (2009), this bound is probably maximal for $k^i = 1$ and parameters $a^i \approx -0.0691, b^i = 1/2$, with value $\approx 0.64297807\pi \approx 2.02$.

Finally, the expected number of rejections to perform Step (ii) is equal to $\prod_{i=1}^d C_{k^i}$ which is of order 2^d , and the expected total number of rejections of the chain rule is of order

$$\sum_{n=1}^N 2^d \frac{N}{N-(n-1)} = 2^d N \sum_{n=1}^N \frac{1}{n} \approx 2^d N \log(N). \quad (4.4.14)$$

4.5 EMPIRICAL INVESTIGATION

We perform three main sets of experiments to investigate the properties of the BH (4.3.1) and EZ (4.3.13) estimators of the integral

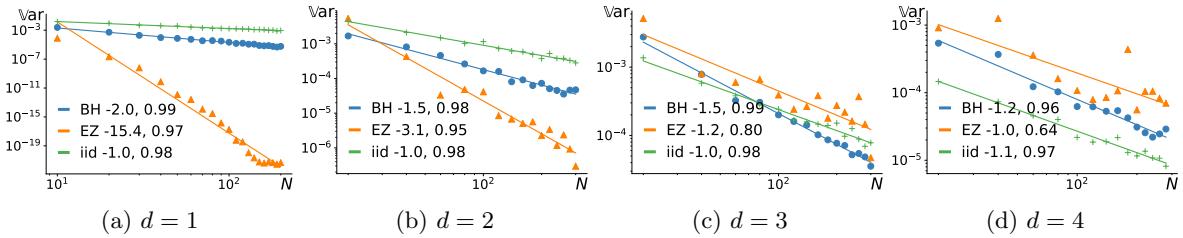
See also Chow, Gatteschi, and Wong (1994) and Gautschi (2009, Equation 1.3).

$\int f(x)\mu(dx)$. We add the baseline vanilla Monte Carlo, where points are drawn i.i.d. proportionally to μ . The two estimators are built from the multivariate Jacobi ensemble, cf. Section 4.2. First, we extend, for larger N , the experiments of Bardenet and Hardy (2020) illustrating their fast CLT (4.3.3) on a smooth *bump* function. Then, we illustrate Theorem 4.3.1 by considering polynomial functions that can be either fully or partially decomposed in the eigenbasis of the DPP kernel. Finally, we compare the potential of both estimators on various non smooth functions.

4.5.1 The bump experiment

Bardenet and Hardy (2020, Section 3) illustrate the behavior of \hat{I}_N^{BH} and its CLT (4.3.3) on a unimodal, smooth *bump* function. The expected variance decay is of order $1/N^{1+1/d}$. We reproduce their experiment in Figure 4.4 for larger N , and compare with the behavior of \hat{I}_N^{EZ} . In short, \hat{I}_N^{EZ} dramatically outperforms \hat{I}_N^{BH} in $d \leq 2$, with surprisingly fast empirical convergence rates. When $d \geq 3$, performance decreases, and \hat{I}_N^{BH} shows both faster and more regular variance decay.

To know whether we can hope for a CLT for \hat{I}_N^{EZ} , we performed Kolmogorov-Smirnov tests for $N = 300$, which yielded small p -values across dimensions, from 0.03 to 0.24. This is compared to the same p -values for \hat{I}_N^{BH} , which range from 0.60 to 0.99. The lack of normality of \hat{I}_N^{EZ} is partly due to a few outliers. Where these outliers come from is left for future work; ill-conditioning of the linear system (4.3.4) is an obvious candidate.



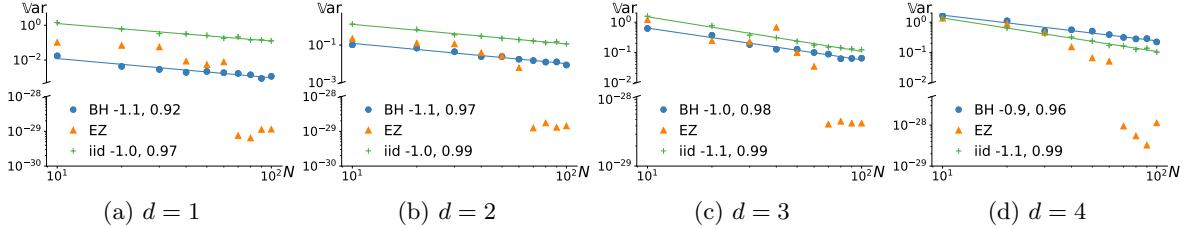
4.5.2 Integrating sums of eigenfunctions

In the next series of experiments, we are mainly interested in testing the variance decay of $\hat{I}_N^{\text{EZ}}(f)$ prescribed by Theorem 4.3.1. To that end, we consider functions of the form

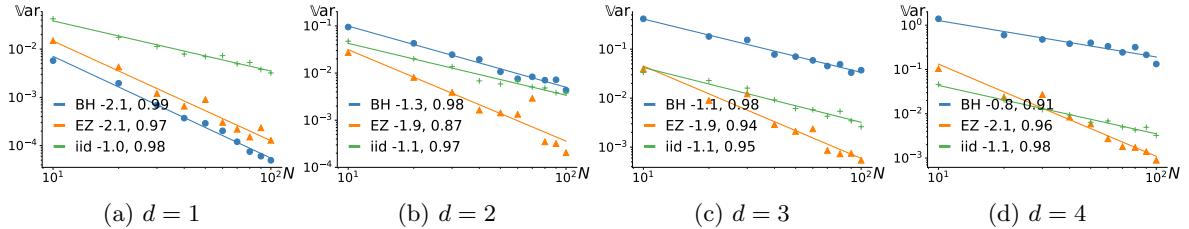
$$f(x) = \sum_{\mathfrak{b}(k)=0}^{M-1} \frac{1}{\mathfrak{b}(k)+1} \phi_k(x), \quad (4.5.1)$$

whose integral w.r.t. μ is to be estimated based on realizations of the multivariate Jacobi ensemble with kernel $K(x, y) = \sum_{\mathfrak{b}(k)=0}^{N-1} \phi_k(x)\phi_k(y)$, where $N \neq M$ a priori. This means that the function f can be either fully ($M \leq N$) or partially ($M > N$) decomposed in the eigenbasis of the kernel. In both cases, we let the number of points N used to build the two estimators vary from 10 to 100 in dimensions $d = 1$ to 4.

Figure 4.4: Variance of the different estimators as a function of the number of points, in the context of Section 4.5.1. The numbers in the legend are the slope and R^2 .



In the first setting, we set $M = 70$. Thus, N eventually reaches the number of functions used to build f in (4.5.1), after what $\widehat{I}_N^{\text{EZ}}$ is an exact estimator, see Figure 4.5. For each dimension d , we indeed observe a drop in the variance of $\widehat{I}_N^{\text{EZ}}$ once the number of points of the DPP hits the threshold $N = M$. This is in perfect agreement with Theorem 4.3.1: once $f \in \mathcal{H}_M \subseteq \mathcal{H}_N$, the variance in (4.3.7) is zero.



The second setting has $M = N + 1$, so that the number of points N is never enough for the variance in (4.3.7) to be zero, see Figure 4.6. In the second setting, as N increases the contribution of the extra mode $\phi_{b^{-1}(N)}$ in (4.5.1) decreases as N^{-1} . Hence, from Theorem 4.3.1 we expect a variance decay of order N^{-2} , which we observe in practice.

4.5.3 Further experiments

In Appendices 4.A.1-4.A.4 we test the robustness of both BH and EZ estimators, when applied to functions presenting discontinuities or which do not belong to the span of the eigenfunctions of the kernel. Although the conditions of the CLT (4.3.3) associated to \widehat{I}^{BH} are violated, the corresponding variance decay is smooth but not as fast. For \widehat{I}^{EZ} , the performance deteriorates with the dimension. Indeed, the cross terms arising from the Taylor expansion of the different functions introduce monomials, associated to large coefficients, that do not belong to \mathcal{H}_N . Sampling more points would reduce the variance (4.3.7). But more importantly, for EZ to excel, this suggests to adapt the kernel to the basis where the integrand is known to be sparse or to have fast-decaying coefficients. In regimes where BH and EZ do not shine, vanilla Monte Carlo becomes competitive for small values of N .

4.6 DISCUSSION

Ermakov and Zolotukhin (EZ, 1960) proposed a non-obvious unbiased Monte Carlo estimator using projection DPPs. It requires solving a linear system, which in turn involves evaluating both the N eigenfunc-

Figure 4.5: Variance of the different estimators as a function of the number of points N , for f of the form (4.5.1) with $M = 70$. The numbers in the legend are the slope and R^2 .

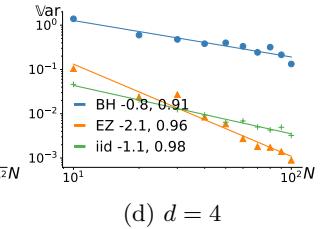


Figure 4.6: Variance of the different estimators as a function of the number of points N , for f of the form (4.5.1) with $M = N+1$. The numbers in the legend are the slope and R^2 .

tions of the corresponding kernel and the integrand at the N points of the DPP sample. This is yet another connection between DPPs and linear algebra. In fact, solving this linear system provides unbiased estimates of the Fourier-like coefficients of the integrand f with each of the N eigenfunctions of the DPP kernel. Remarkably, these estimators have identical variance, and this variance measures the accuracy of the approximation of f by its projection onto these eigenfunctions. With modern arguments, we have provided a much shorter proof of these properties than in the original work of (Ermakov and Zolotukhin, 1960). Beyond this, little is known on the EZ estimator. While coming with a less interpretable variance, the more direct estimator proposed by Bardenet and Hardy (BH, 2020) has an intrinsic connection with the classical Gauss quadrature and further enjoys stronger theoretical properties when using multivariate Jacobi ensemble.

Our experiments highlight the key features of both estimators when the underlying DPP is a multivariate Jacobi ensemble, and further demonstrate the known properties of the BH estimator in yet unexplored regimes. Although EZ shows a *surprisingly fast* empirical convergence rate for $d \leq 2$, its behavior is more erratic for $d \geq 3$. Ill-conditioning of the linear system is a potential source of outliers in the distribution of the estimator. Regularization may help but would introduce a stability/bias trade-off. More generally, EZ seems worth investigating for integration or even interpolation tasks where the function is known to be decomposable in the eigenbasis of the kernel, i.e., in a setting similar to the one of Bach (2017). Finally, the new implementation of an exact sampler for multivariate Jacobi ensemble unlocks more large-scale empirical investigations and asks for more theoretical work.

APPENDICES

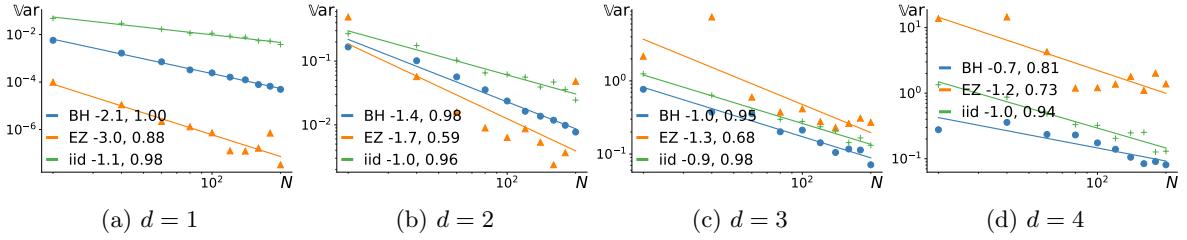
4.A FURTHER EXPERIMENTS

4.A.1 Integrating absolute value

We consider estimating the integral

$$\int_{[-1,1]^d} \prod_{i=1}^d |x^i| (1-x^i)^{a^i} (1+x^i)^{b^i} dx^i, \quad (4.A.1)$$

where $a^1, b^1 = -\frac{1}{2}$ and a^i, b^i i.i.d. uniformly in $[-\frac{1}{2}, \frac{1}{2}]$, using BH (4.3.1) and EZ (4.3.13) estimators. Results are given in Figure 4.A.1.



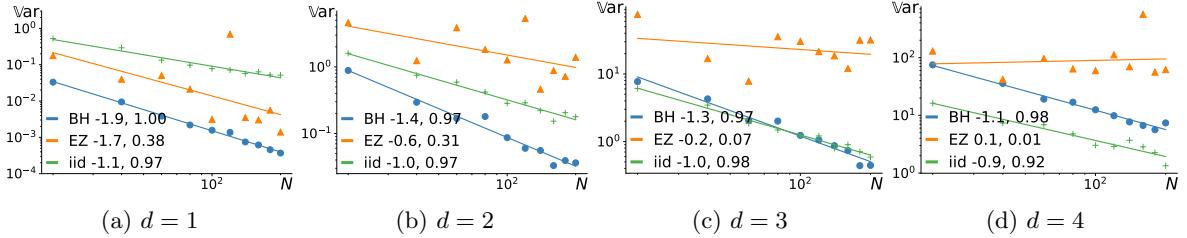
In dimension $d = 1$, the absolute value is well approximated by its truncated Taylor series of low order and EZ performs very well, but as the dimension increases, its performance is more erratic. For $d \leq 2$, the performance of BH is smooth and better than vanilla Monte Carlo. In particular, for $d \leq 2$, the rate $1/N^{1+1/d}$ seems to hold for BH while the conditions for the CLT (4.3.3) are not satisfied. But it seems no longer true in larger dimension.

4.A.2 Integrating Heaviside

Let $H(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}$. We consider estimating the integral

$$\int_{[-1,1]^d} \prod_{i=1}^d 2 \left(H(x^i) - \frac{1}{2} \right) (1-x^i)^{a^i} (1+x^i)^{b^i} dx^i, \quad (4.A.2)$$

where $a^1, b^1 = -\frac{1}{2}$ and a^i, b^i i.i.d. uniformly in $[-\frac{1}{2}, \frac{1}{2}]$, using BH (4.3.1) and EZ (4.3.13) estimators. Results are given in Figure 4.A.2.



The EZ estimator behaves in a very erratic way; it does not seem robust to the discontinuity we have introduced. This can be explained by

Figure 4.A.1: Comparison of $\widehat{I}_N^{\text{BH}}$ and $\widehat{I}_N^{\text{EZ}}$ for absolute value, cf. Section 4.5.3.

Figure 4.A.2: Comparison of $\widehat{I}_N^{\text{BH}}$ and $\widehat{I}_N^{\text{EZ}}$ for Heaviside function, cf. Section 4.5.3.

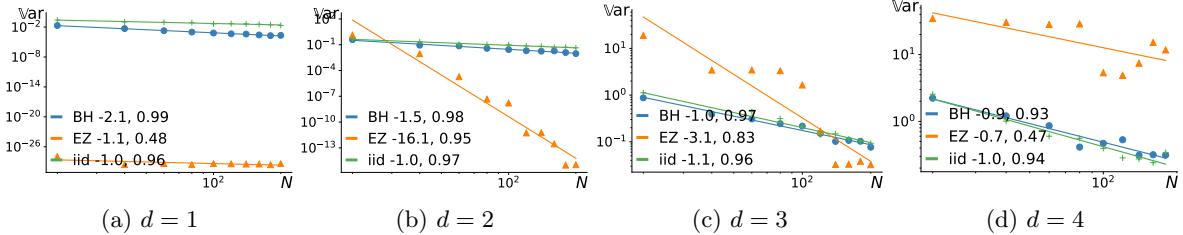
considering $H(x) = \frac{1}{2} \lim_{\epsilon \rightarrow 0} 1 + \tanh \frac{x}{\epsilon}$ and taking the product of the Taylor series expansions of \tanh ; the square of the coefficients in front of the monomials in such expansion become very large as $\epsilon \rightarrow 0$. One could expect better behavior for very large N . The performance of BH is smooth and the rate $1/N^{1+1/d}$ seems to hold despite the conditions for the CLT (4.3.3) are not satisfied.

4.A.3 Integrating cosine

We consider estimating the integral

$$\int_{[-1,1]^d} \prod_{i=1}^d \cos(\pi x^i) (1-x^i)^{a^i} (1+x^i)^{b^i} dx^i, \quad (4.A.3)$$

where $a^1, b^1 = -\frac{1}{2}$ and a^i, b^i i.i.d. uniformly in $[-\frac{1}{2}, \frac{1}{2}]$, using BH (4.3.1) and EZ (4.3.13) estimators. Results are given in Figure 4.A.3.



The EZ estimator behaves well for $d \leq 2$ but its performance deteriorates for $d \geq 3$. Indeed, the cross terms arising from the Taylor expansion of the different $\cos(\pi x^i)$ introduce monomials, associated to large coefficients, that do not belong to \mathcal{H}_N . One could expect better behavior for very large N . For $d \leq 2$, the rate $1/N^{1+1/d}$ for BH seems to hold despite the conditions for the CLT (4.3.3) are not satisfied. For $d \geq 3$, BH and vanilla Monte Carlo behave similarly.

4.A.4 Integrating a mixture of smooth and non smooth functions

Let $f(x) = H(x)(\cos(\pi x) + \cos(2\pi x) + \sin(5\pi x))$. We consider estimating the integral

$$\int_{[-1,1]^d} \prod_{i=1}^d f(x^i) (1-x^i)^{a^i} (1+x^i)^{b^i} dx^i, \quad (4.A.4)$$

where $a^1, b^1 = -\frac{1}{2}$ and a^i, b^i i.i.d. uniformly in $[-\frac{1}{2}, \frac{1}{2}]$, using BH (4.3.1) and EZ (4.3.13) estimators.

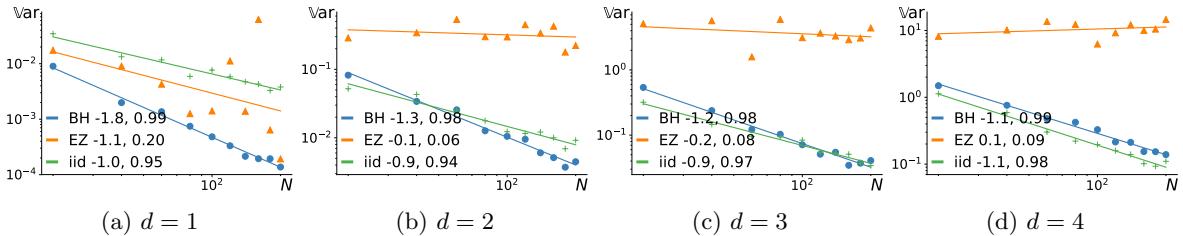


Figure 4.A.3: Comparison of \hat{I}_N^{BH} and \hat{I}_N^{EZ} for cosine, cf. Section 4.5.3.

Figure 4.A.4: Comparison of \hat{I}_N^{BH} and \hat{I}_N^{EZ} , cf. Section 4.5.3.

Fast sampling from β -ensembles

5

This chapter presents our contribution¹ regarding sampling algorithms for β -ensembles with time complexity less than cubic in the cardinality of the ensemble. Since there are many long equations, we shift to a larger text layout.

β -ensembles are probability distributions of the form

$$\frac{1}{Z_{\beta,N}} |\Delta(x_1, \dots, x_N)|^\beta \prod_{n=1}^N e^{-V(x_n)} dx_n, \quad x_1, \dots, x_N \in \mathbb{R}, \quad (5.0.1)$$

where $\Delta(x_1, \dots, x_N) = \prod_{i < j} (x_j - x_i)$ is the Vandermonde determinant, $\beta > 0$ is akin to an inverse temperature in statistical physics, and $V : \mathbb{R} \rightarrow \mathbb{R}$ is called the *potential*. Loosely speaking, one can think of (5.0.1) as representing the position of N particles living on the real line, confined by the potential V , and repelling each other through the Vandermonde determinant. As this interpretation suggests, β -ensembles arise as models in statistical physics (Forrester, 2010, Chapters 1 to 3). They are also famous as eigenvalue distribution of the classical random matrix models. The particular values $\beta \in \{1, 2, 4\}$ respectively appear when considering specific random matrices with real, complex, or quaternionic Gaussian entries; see e.g., Forrester (2010), or (Anderson, Guionnet, and Zeitouni, 2009, Chapter 4).

The case $\beta = 2$ is of particular interest, since the distribution of $\{x_1, \dots, x_N\}$ then becomes a particular projection DPP, called an *orthogonal polynomial ensemble* (OPE, König, 2004). In the context of Monte Carlo integration, see Chapter 4, we already used the so-called Jacobi OPE where $e^{-V(x)}$ is proportional to the density of a Beta distribution.

Numerical procedures to generate samples from β -ensembles are also needed to establish conjectures in statistical physics or random matrix theory. For instance, using a tailored version of the generic DPP sampler of Hough et al. (2006), Olver, Nadakuditi, and Trogdon (2014) explore so-called *universality properties* in random matrix theory, and make conjectures on the law of $\max x_i$ when $\beta = 2$ and V is a polynomial of degree 4. Chafaï and Ferré (2018) rather use Hamiltonian Monte Carlo to approximately sample from various *Coulomb gases*, including (5.0.1) with $\beta = 2$ and $V(x) = x^4/4$, and investigate their limiting features when $N \rightarrow \infty$. From a different perspective, Li and Menon (2013) view (5.0.1) as the equilibrium distribution for the Dyson Brownian motion associated to the potential V . When $\beta = 2$, they generate approximate samples by discretizing the corresponding stochastic differential equation.

Sampling algorithms for β -ensembles come in three different guises, which we describe in increasing order of complexity. First, when $\beta > 0$ and V is the negative logarithm of a Gaussian, gamma, or beta pdf, we speak of the Hermite, Laguerre, and Jacobi β -ensemble, respectively. Dumitriu and Edelman (2002) showed that the Hermite and Laguerre β -ensembles can be characterized as the eigenvalue distribution of a random tridiagonal matrix with easy-to-sample independent entries. This gives a $\mathcal{O}(N^2)$ sampling algorithm. Dumitriu and Edelman (2002) expected the same to hold for the Jacobi β -ensemble, which was later proved by Killip and Nenciu (2004).

Second, when $\beta = 2$, the generic projection DPP sampler of Hough et al. (2006) applies. That there

5.1	Classical β -ensembles and their tridiagonal models	91
5.2	Atomic measures, moments and Jacobi matrices	
	93	
	Orthogonal polynomials and Jacobi matrices	
	Orthogonal polynomials and moments	
5.3	Making the change of variables	97
5.4	Proving the three classical tridiagonal models	100
	The $H\beta E$ and its tridiagonal model	
	The $L\beta E$ and its tridiagonal model	
	The $J\beta E$ and its tridiagonal model	
5.5	Gibbs sampling tridiagonal models associated to polynomial potentials	105
	Sampling from the conditionals	
	Example simulations and empirical study of the convergence	
5.6	Conclusion	112

¹ G. Gautier, R. Bardenet, and M. Valko. 2020. *Fast sampling from β -ensembles*. ArXiv e-prints. arXiv:2003.02344.

github.com/guilgautier/DPPy

actually exists an exact sampler is maybe surprising, and it is a particular feature of DPPs among interacting particle systems. The procedure remains costly, though. It has at least a cubic cost in N , with the total cost further depending on rejection sampling subroutines, the cost of which is case-dependent and has been left uninvestigated. Additionally, it is required in this procedure to numerically evaluate the first N orthonormal polynomials $p_k, k = 0, \dots, N - 1$ with respect to $e^{-V(x)} dx$. This is traditionally done using their recurrence relation

$$\sqrt{b_{k-1}} p_{k-1}(x) + a_k p_k(x) + \sqrt{b_k} p_{k+1}(x) = x p_k(x), \quad (5.0.2)$$

see e.g., Gautschi (2004). In the Hermite, Laguerre, and Jacobi case, the recurrence coefficients a_k, b_k are known, but as we just saw, these three cases are already covered by a computationally more efficient tridiagonal matrix model. When the coefficients in (5.0.2) are not known, one can either rely on the Stieltjes algorithm (Gautschi, 2004, Section 2.2) or numerically solve a Riemann-Hilbert problem (Olver, 2011). The latter is theoretically only an $\mathcal{O}(N)$ overcost.

A third algorithm is Markov chain Monte Carlo (MCMC, see e.g., Robert and Casella, 2004), which is in principle valid for any $\beta > 0$ and any V that gives a well-defined distribution in (5.0.1). MCMC only requires to evaluate the pdf in (5.0.1) pointwise and up to a constant, but it only delivers approximate samples of (5.0.1), in the sense that it outputs a sample from a Markov chain with (5.0.1) as its limiting distribution. The issue is that the performance of MCMC samplers – the mixing time of the Markov chain – deteriorates when $N \gg 1$, which is typically the regime of interest for conjectures in random matrix theory or statistical physics. Hybrid Monte Carlo (HMC, Duane et al., 1987; Neal, 2011) is an MCMC sampler that has demonstrated good mixing in high-dimensional problems, provided one can evaluate the gradient of the pdf in (5.0.1). For β -ensembles with $\beta = 2$, Chafaï and Ferré (2018) provide empirical evidence that the output of HMC successfully reproduces known limiting features of the large N regime, and they raise new conjectures. The main limitation of this approach is the large number of MCMC iterations required by HMC: Chafaï and Ferré (2018) require at least 10^4 iterations and are restricted to $N \leq 50$.

In this chapter, we further investigate fast samplers of β -ensembles. Our contributions are twofold. First we gather existing tools from different communities to give an elementary treatment of the tridiagonal models for the Hermite, Laguerre, and Jacobi β -ensembles. This proof crucially relies on successive reparametrizations of the recurrence coefficients in (5.0.2) and unifies the treatment of tridiagonal models for the three classical β -ensembles, pioneered with two different methods by Dumitriu and Edelman (2002) and Killip and Nenciu (2004). We take no credit for the originality of the proof: the credit should go – among others cited below – to Dette and Nagel (2012), who studied distributions on the space of moments, and recognized these three β -ensembles as corresponding to natural distributions over moments. We rather take credit for a stand-alone and elementary version of this unifying proof, using only basic facts on orthogonal polynomials and linear algebra.

Our second contribution is an MCMC sampler that applies to polynomial potentials. For V of degree at most 6, we give experimental evidence that the resulting Markov chain mixes extremely fast, which confirms an intuition of Krishnapur, Rider, and Virág (2016, Section 2). On a variety of potentials, we demonstrate that our simple Gibbs Markov kernel yields a much cheaper (although approximate) sampler than the exact procedure of Hough et al. (2006, for $\beta = 2$). Importantly, our Markov kernel outperforms the HMC approach of Chafaï and Ferré (2018) applied to β -ensembles. To give an idea, we are able to reproduce known features of (5.0.1) for values of N in the hundreds, using only a few Gibbs sweeps, totaling a few seconds on a modern laptop: it takes roughly 10s for $N = 200$ points and less than a minute for $N = 1000$ points. That such a basic Gibbs kernel can outperform HMC may seem surprising. The key is that we exploit the structure of β -ensembles by defining a Markov chain on the recurrence coefficients of orthogonal polynomials. These recurrence coefficients are defined similarly to (5.0.2), but this time using the orthogonal polynomials with respect to a random discrete measure, the support of which is the β -ensemble. Intuitively, in that new parametrization, the interaction between variables is short-range compared to the interaction among particles in (5.0.1), and Gibbs sampling thus becomes easier. In this sense, our MCMC kernel extends the tridiagonal models of the three classical β -ensembles. Finally, we note that all experiments

can be reproduced using our DPPy toolbox (Gautier et al., 2019, <https://github.com/guilgautier/DPPy>), which features all samplers described here.

The rest of the chapter is organized as follows. In Section 5.1, we survey existing results on tridiagonal models for β -ensembles. Known exact sampling results actually take the form of diagonalizing random Jacobi matrices, that is, tridiagonal matrices with entries the recurrence coefficients of a sequence of orthogonal polynomials. We introduce the necessary background on orthogonal polynomials in Section 5.2. In Section 5.3, we perform the change of variables between the points of a β -ensemble augmented with weights and the entries of a Jacobi matrix. In Section 5.4, we give an elementary proof of the known results on tridiagonal models. Finally, in Section 5.5, we demonstrate the potential of a simple MCMC scheme based on a Gibbs kernel, to sample Jacobi matrices corresponding to β -ensembles with polynomial potentials.

5.1 CLASSICAL β -ENSEMBLES AND THEIR TRIDIAGONAL MODELS

The Hermite, Laguerre and Jacobi β -ensembles were originally defined for $\beta \in \{1, 2, 4\}$, as the eigenvalue distribution of some random full matrices; see e.g., Anderson, Guionnet, and Zeitouni (2009). The latter matrices are symmetrizations of matrices filled with i.i.d. real, complex, or quaternionic Gaussian variables when β is respectively 1, 2, and 4. In this section, we recall the seminal results of Dumitriu and Edelman (2002) and Killip and Nenciu (2004) regarding the construction of real-symmetric tridiagonal random matrices, whose eigenvalues follow the classical Hermite, Laguerre and Jacobi β -ensembles. These results actually allow any $\beta \in (0, +\infty)$, and can be interpreted as samplers with $\mathcal{O}(N^2)$ time complexity, by simply diagonalizing the proposed tridiagonal matrices.

Let $\mathbf{a} \triangleq (a_1, \dots, a_N) \in \mathbb{R}^N$, $\mathbf{b} \triangleq (b_1, \dots, b_{N-1}) \in (0, +\infty)^{N-1}$, and define the tridiagonal matrix

$$J_{\mathbf{a}, \mathbf{b}} \triangleq \begin{bmatrix} a_1 & \sqrt{b_1} & & (0) \\ \sqrt{b_1} & a_2 & \ddots & \\ \ddots & \ddots & \ddots & \sqrt{b_{N-1}} \\ (0) & & \sqrt{b_{N-1}} & a_N \end{bmatrix}. \quad (5.1.1)$$

Such a matrix is called a *Jacobi matrix*. As we will see in Section 5.2, Jacobi matrices naturally arise in the study of orthogonal polynomials.

To derive the random tridiagonal matrix model for the Hermite β -ensemble, Dumitriu and Edelman (2002) started from the original random full matrix model for the Hermite ensemble with $\beta = 1$. More specifically, they considered the symmetric part of a random matrix filled with i.i.d. unit Gaussians, and applied Householder transformations to reduce it to tridiagonal form, as in, e.g., Golub and Van Loan (2013, Section 5.4.8).

Theorem 5.1.1 (Dumitriu and Edelman, 2002, II C, for $\mu = 0$ and $\sigma = 1$). *The Hermite β -ensemble, defined as (5.0.1) with potential $V(x) = \frac{1}{2\sigma^2}(x - \mu)^2$, corresponds to the eigenvalue distribution of the tridiagonal matrix $J_{\mathbf{a}, \mathbf{b}}$ in (5.1.1), with entries drawn independently as*

$$a_n \sim \mathcal{N}(\mu, \sigma^2), \quad \text{and} \quad b_n \sim \Gamma\left(\frac{\beta}{2}(N-n), \sigma^2\right). \quad (5.1.2)$$

For the Laguerre β -ensemble, Dumitriu and Edelman (2002) used the same linear algebra techniques starting from the original full matrix model defining the Laguerre β -ensemble for $\beta = 1$. The latter corresponds to the eigenvalue distribution of the covariance matrix XX^\top of i.i.d. $\mathcal{N}(0, I)$ vectors. More specifically, they reduced the matrix X to bidiagonal form, see, e.g., Golub and Van Loan (2013, Section 8.3.1).

Theorem 5.1.2 (Dumitriu and Edelman, 2002, III B, for $k = \frac{\beta}{2}(M - N + 1)$ and $\theta = 2$).

The Laguerre β -ensemble, defined as (5.0.1) with potential $V(x) = -(k-1)\log(x) + \frac{x}{\theta}$, corresponds to the eigenvalue distribution of the tridiagonal matrix $J_{\mathbf{a}, \mathbf{b}}$ in (5.1.1) parametrized by

$$\begin{aligned} a_1 &= \xi_1, \quad \text{and} \quad a_n = \xi_{2n-2} + \xi_{2n-1}, \quad \text{for } 2 \leq n \leq N, \quad \text{and} \\ b_n &= \xi_{2n-1}\xi_{2n}, \quad \text{for } 1 \leq n \leq N-1, \end{aligned} \tag{5.1.3}$$

with independent coefficients

$$\xi_{2n-1} \sim \Gamma\left(\frac{\beta}{2}(N-n) + k, \theta\right), \quad \text{and} \quad \xi_{2n} \sim \Gamma\left(\frac{\beta}{2}(N-n), \theta\right). \tag{5.1.4}$$

Dumitriu and Edelman (2002) left the construction of a tridiagonal model for the Jacobi β -ensemble as an open problem. Killip and Nenciu (2004) found such a model as a byproduct of their study of the Circular β -ensemble. The latter ensemble is originally defined, for $\beta \in \{1, 2, 4\}$, as the eigenvalue distribution of orthogonal, unitary and symplectic matrices drawn uniformly at random from the corresponding Haar measures. First, Killip and Nenciu (2004) applied Householder transformations to reduce to quindagonal form a unitary matrix drawn uniformly at random. Second, they projected the resulting eigenvalues onto the real line to obtain the tridiagonal model for the Jacobi β -ensemble.

Theorem 5.1.3 (Killip and Nenciu, 2004, Theorem 2). *The Jacobi β -ensemble, defined as (5.0.1) with potential $V(x) = -[(a-1)\log(x) + (b-1)\log(1-x)]$, corresponds to the eigenvalue distribution of the tridiagonal matrix $J_{\mathbf{a}, \mathbf{b}}$ in (5.1.1) parametrized by*

$$\begin{aligned} a_1 &= c_1, & a_n &= (1 - c_{2n-3})c_{2n-2} + (1 - c_{2n-2})c_{2n-1}, & \text{for } 2 \leq n \leq N, \\ b_1 &= c_1(1 - c_1)c_2, & b_n &= (1 - c_{2n-2})c_{2n-1}(1 - c_{2n-1})c_{2n}, & \text{for } 2 \leq n \leq N-1, \end{aligned} \tag{5.1.5}$$

with independent coefficients

$$\begin{aligned} c_{2n-1} &\sim \text{Beta}\left(\frac{\beta}{2}(N-n) + a, \frac{\beta}{2}(N-n) + b\right), \quad \text{and} \\ c_{2n} &\sim \text{Beta}\left(\frac{\beta}{2}(N-n), \frac{\beta}{2}(N-n-1) + a + b\right). \end{aligned} \tag{5.1.6}$$

Observe how the stars align for these three special β -ensembles: Hermite, Laguerre, and Jacobi. The coefficients in successive parameterizations of the Jacobi matrix $J_{\mathbf{a}, \mathbf{b}}$ are independent with easy-to-sample distributions. From a practical point of view, for any $\beta > 0$, the computation of the eigenvalues of these random real-symmetric tridiagonal matrices can be seen as a $\mathcal{O}(N^2)$ sampler for each of the model; see Coakley and Rokhlin (2013) for practical approaches to diagonalizing such matrices that can even run in quasi-linear time.

Studying distributions over the space of moments, Dette and Nagel (2012) elegantly derived the three classical tridiagonal models as the supports of random atomic measures corresponding to natural moment distributions. On our side, we provide a unified treatment of these three classical models using a more pedestrian, sampling-motivated approach. To do this, we consider an atomic measure $\mu = \sum_{n=1}^N \omega_n \delta_{x_n}$, whose support points are distributed as a target β -ensemble, and take the Jacobi matrix $J_{\mathbf{a}, \mathbf{b}}$ in (5.1.1) with coefficients the recurrence coefficients (5.0.2) of the orthonormal polynomials w.r.t. μ . We shall see in Section 5.2 that the recurrence coefficients are a suitable reparametrization of the atomic measure μ . In particular, the support of μ actually coincides with the eigenvalues of $J_{\mathbf{a}, \mathbf{b}}$, so that a tridiagonal model for the support of μ follows from knowing how to randomize $J_{\mathbf{a}, \mathbf{b}}$.

The first step of our proof will be to rederive Theorem 5.1.4, which allows changing variables from the nodes and weights of an atomic measure μ to the recurrence coefficients defining the Jacobi matrix $J_{\mathbf{a}, \mathbf{b}}$. Note that the specific choice of distribution on the weights is simply of mathematical convenience.

Theorem 5.1.4 (Krishnapur, Rider, and Virág, 2016, Proposition 2). *Consider a random atomic measure $\mu = \sum_{n=1}^N \omega_n \delta_{x_n}$, with nodes and weights independently distributed according to a β -ensemble with potential V (5.0.1) and a Dirichlet $\text{Dir}(\beta/2)$, respectively. Otherly put, the joint distribution of $(x_1, \dots, x_N, w_1, \dots, w_N)$ is proportional to*

$$|\Delta(x_1, \dots, x_N)|^\beta e^{-\sum_{n=1}^N V(x_n)} dx_{1:N} \prod_{n=1}^N w_n^{\frac{\beta}{2}-1} \mathbb{1}_{w_n \geq 0} \mathbb{1}_{\sum_{n=1}^N w_n = 1} dw_{1:N-1}. \quad (5.1.7)$$

Then, the entries of $J_{\mathbf{a}, \mathbf{b}}$ in (5.1.1), i.e., the recurrence coefficients associated to μ have joint distribution proportional to

$$\prod_{n=1}^{N-1} b_n^{\frac{\beta}{2}(N-n)-1} e^{-\text{Tr}[V(J_{\mathbf{a}, \mathbf{b}})]} da_{1:N} db_{1:N-1}. \quad (5.1.8)$$

In Section 5.3, we first re-prove that the change of variables underlying Theorem 5.1.4 is valid. Then, in Section 5.4, we obtain the three classical tridiagonal models of Theorems 5.1.1, 5.1.2, and 5.1.3 as instances of this result, using further smart-but-simple changes of variables. Before delving into the proof, we survey how Jacobi matrices naturally appear in the theory of orthogonal polynomials.

5.2 ATOMIC MEASURES, MOMENTS AND JACOBI MATRICES

Throughout this section, we let $\mu = \sum_{n=1}^N w_n \delta_{x_n}$ be a discrete probability measure on \mathbb{R} with N distinct atoms x_1, \dots, x_N and positive weights $\omega_1, \dots, \omega_N$. We further denote its moments by

$$m_k \triangleq \sum_{n=1}^N w_n x_n^k, \quad k \geq 0.$$

5.2.1 Orthogonal polynomials and Jacobi matrices

This section closely follows Simon (2011, Section 1.3), to which we refer for details. Applying the Gram-Schmidt procedure in $L^2(\mu)$ to the monomials $(x \mapsto x^k)_{k=0}^{N-1}$ yields monic polynomials $(P_k)_{k=0}^{N-1}$ with $\deg P_k = k$ and

$$\langle P_k, P_\ell \rangle_\mu \triangleq \sum_{n=1}^N w_n P_k(x_n) P_\ell(x_n) = 0, \quad k \neq \ell. \quad (5.2.1)$$

These polynomials are called the *monic orthogonal polynomials* (monic OPs, in short) with respect to μ . We define the N -th monic OP as

$$P_N(x) = \prod_{n=1}^N (x - x_n).$$

Since $\|P_N\|_\mu \triangleq \langle P_N, P_N \rangle_\mu = 0$, P_N is the zero vector of $L^2(\mu)$: it is orthogonal to all P_k with $k \leq N-1$.

Furthermore, for any $n < N$, since $\langle x P_n, P_k \rangle_\mu = \langle P_n, x P_k \rangle_\mu = 0$ for $k < n-1$, the polynomial $x P_n$ can be uniquely expressed using only P_{n-1} , P_n and P_{n+1} . This is usually phrased as follows. The monic OPs satisfy a three-term recurrence relation involving two sequences of *recurrence coefficients*, namely

$$\begin{aligned} P_{-1} &\equiv 0, P_0 \equiv 1 \text{ and} \\ x P_n(x) &= b_n P_{n-1}(x) + a_{n+1} P_n(x) + P_{n+1}(x), \quad \forall 0 \leq n < N, \end{aligned} \quad (5.2.2)$$

where $\mathbf{a} = a_{1:N} = (a_n) \in \mathbb{R}^N$, and $\mathbf{b} = b_{1:N-1} = (b_n) \in (0, +\infty)^{N-1}$. These relations can be written in matrix form as

$$\begin{bmatrix} a_1 & 1 & & (0) \\ b_1 & a_2 & \ddots & \\ & \ddots & \ddots & 1 \\ (0) & & b_{N-1} & a_N \end{bmatrix} \begin{bmatrix} P_0(x) \\ \vdots \\ P_{N-2}(x) \\ P_{N-1}(x) \end{bmatrix} = x \begin{bmatrix} P_0(x) \\ \vdots \\ P_{N-2}(x) \\ P_{N-1}(x) \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ P_N(x) \end{bmatrix}. \quad (5.2.3)$$

From (5.2.3), it is clear that the roots of P_N are also eigenvalues of the tridiagonal matrix $T_{\mathbf{a}, \mathbf{b}}$ appearing on the left-hand side. Roots and eigenvalues actually coincide since P_N has N distinct roots by definition.

A lot more can be said on the links between OPs and their recurrence coefficients. For instance, Proposition 5.2.1 will be of use later on.

Proposition 5.2.1. *The squared norms of the monic polynomials $(P_n)_{n=0}^{N-1}$ can be expressed as*

$$\|P_0\|_\mu^2 = 1, \quad \|P_k\|_\mu^2 = \prod_{n=1}^k b_n, \quad \forall 1 \leq k \leq N-1. \quad (5.2.4)$$

Proof. For $k = 0$, $\|P_0\|_\mu^2 = \sum_{n=1}^N w_n = 1$. Then, for any $1 \leq k \leq N-1$,

$$\begin{aligned} \langle (5.2.2), P_{k-1} \rangle_\mu &\iff \langle xP_k, P_{k-1} \rangle_\mu = \langle b_k P_{k-1}, P_{k-1} \rangle_\mu \\ &\iff \langle P_k, xP_{k-1} \rangle_\mu = b_k \langle P_{k-1}, P_{k-1} \rangle_\mu \\ &\iff \langle P_k, x^n \rangle_\mu = b_k \|P_{k-1}\|_\mu^2 \\ &\iff \|P_k\|_\mu^2 = b_k \|P_{k-1}\|_\mu^2, \end{aligned}$$

and a simple recursion provides $\|P_k\|_\mu^2 = \prod_{n=1}^k b_n > 0$. \square

Denoting by $D = \text{diag}(\|P_0\|, \dots, \|P_{N-1}\|)$, Proposition 5.2.1 yields $J_{\mathbf{a}, \mathbf{b}} = D^{-1} T_{\mathbf{a}, \mathbf{b}} D$, where we recall that the Jacobi matrix $J_{\mathbf{a}, \mathbf{b}}$ was defined in (5.1.1). This yields the following proposition.

Proposition 5.2.2. *The atoms of μ , coincide with the eigenvalues of $J_{\mathbf{a}, \mathbf{b}}$, where the coefficients of the matrix are taken to be the recurrence coefficients of the monic OPs with respect to μ .*

Proposition 5.2.2 already gives a tentative $\mathcal{O}(N^2)$ sampling algorithm for β -ensembles: find a distribution over Jacobi matrices such that the eigenvalues form the desired β -ensemble. This is precisely what the tridiagonal models of Dumitriu and Edelman (2002) do; see Theorem 5.1.1. To give a complete elementary proof, we need to perform a change of variables from the atoms and weights of μ to the recurrence coefficients. The rest of this section introduces the tools needed for this change of variables, which is then performed in Section 5.3.

So far, we have explained how to obtain a Jacobi matrix from an atomic measure with finite support. The reverse construction is also possible and elementary. This is called Favard's theorem for atomic measures with finite support. To save space and because our proof would be a simple copy of Simon's book, we only give a reference. We have used the same notation as Simon throughout this section, for ease of reference.

Theorem 5.2.3 (Simon, 2011, Theorem 1.3.3). *Let*

$$\mathbb{R}_>^N \triangleq \{x_1, \dots, x_N \in \mathbb{R} \mid x_1 > \dots > x_N\} \quad \text{and} \quad S_N \triangleq \left\{ \omega_1, \dots, \omega_{N-1} > 0 \mid \sum_{n=1}^{N-1} \omega_n < 1 \right\}. \quad (5.2.5)$$

Favard's map

$$\begin{aligned} \psi : \quad \mathbb{R}_>^N \times S_N &\longrightarrow \mathbb{R}^N \times (0, +\infty)^{N-1} \\ (x_{1:N}, w_{1:N-1}) &\longmapsto (a_{1:N}, b_{1:N-1}) \end{aligned} \quad (5.2.6)$$

linking the nodes and weights of $\mu = \sum_{n=1}^N w_n \delta_{x_n}$ with the entries of the corresponding Jacobi matrix $J_{\mathbf{a}, \mathbf{b}}$ defined in (5.1.1), is one-to-one and onto.

Note that whenever $w_{1:N-1} \in S_N$, we always set $w_N = 1 - \sum_{n=1}^{N-1} \omega_n$, so that μ is a probability measure. As a side remark, the weights $w_{1:N}$ of μ can also be expressed using evaluations of the monic OPs on the support of μ (Simon, 2011, Proposition 1.3.1): for all $n = 1, \dots, N$,

$$w_n = \frac{1}{K_N(x_n, x_n)}, \quad \text{with } K_N(x, y) = \sum_{k=0}^{N-1} \frac{P_k(x)P_k(y)}{\|P_k\|_\mu^2}. \quad (5.2.7)$$

These weights are reminiscent of Gaussian quadrature (Gautschi, 2004, Section 1.4.2), where the OPs are usually w.r.t. a non-atomic measure.

5.2.2 Orthogonal polynomials and moments

We know from Theorem 5.2.3 that the change of variables ψ is a bijection. In order to prove that ϕ is a C^1 -diffeomorphism and compute its Jacobian in Section 5.3, we pause to introduce an intermediate parametrization through moments. Intuitively, the moments are responsible for the Vandermonde determinant in (5.0.1).

The monic orthogonal polynomials $(P_n)_{n=0}^N$ w.r.t. μ can also be expressed in terms of the moments (m_k) of μ . First, define the following *moment matrices*, see, e.g., Dette and Studden (1997, Equation 1.4.3).

Definition 5.2.4. Let

$$\underline{H}_{2n} = [m_{i+j}]_{i,j=0}^n = \begin{bmatrix} m_0 & \cdots & m_n \\ \vdots & \ddots & \vdots \\ m_n & \cdots & m_{2n} \end{bmatrix} \quad (5.2.8)$$

$$\underline{H}_{2n+1} = [m_{i+j+1}]_{i,j=0}^n = \begin{bmatrix} m_1 & \cdots & m_{n+1} \\ \vdots & \ddots & \vdots \\ m_{n+1} & \cdots & m_{2n+1} \end{bmatrix} \quad (5.2.9)$$

$$\overline{H}_{2n+1} = [m_{i+j} - m_{i+j+1}]_{i,j=0}^n = \begin{bmatrix} m_0 - m_1 & \cdots & m_n - m_{n+1} \\ \vdots & \ddots & \vdots \\ m_n - m_{n+1} & \cdots & m_{2n} - m_{2n+1} \end{bmatrix}. \quad (5.2.10)$$

where H stands for *Hankel matrix*.

In the definition of β -ensembles (5.0.1), the determinant of the Vandermonde matrix

$$\Delta(x_1, \dots, x_n) \triangleq \begin{bmatrix} 1 & \cdots & 1 \\ x_1 & \cdots & x_n \\ \vdots & & \vdots \\ x_1^{n-1} & \cdots & x_n^{n-1} \end{bmatrix}, \quad (5.2.11)$$

comes out naturally when taking the determinant of moment matrices associated to discrete measures.

Lemma 5.2.5. It holds that

$$|\underline{H}_{2n-2}| = \begin{cases} > 0, & \text{for any } 1 \leq n \leq N, \\ |\Delta(x_1, \dots, x_N)|^2 \prod_{n=1}^N w_n, & \text{for } n = N, \\ 0, & \text{for } n > N. \end{cases} \quad (5.2.12)$$

Moreover, we have

$$|\underline{H}_{2N-1}| = |\underline{H}_{2N-2}| \prod_{n=1}^N x_n \quad \text{and} \quad |\overline{H}_{2N-1}| = |\underline{H}_{2N-2}| \prod_{n=1}^N (1 - x_n). \quad (5.2.13)$$

Proof. For any $1 \leq n \leq N$, the Cauchy-Binet formula yields

$$\begin{aligned} |\underline{H}_{2n-2}| &= \left| \sum_{k=1}^N w_k x_k^{i+j} \right|_{i,j=0}^{n-1} \\ &= \left| \begin{bmatrix} 1 & \cdots & 1 \\ x_1 & \cdots & x_N \\ \vdots & \vdots & \vdots \\ x_1^{n-1} & \cdots & x_N^{n-1} \end{bmatrix} \begin{bmatrix} w_1 & & & \\ & \ddots & & \\ & & w_N & \end{bmatrix} \begin{bmatrix} 1 & x_1 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_N & \cdots & x_N^{n-1} \end{bmatrix} \right| \\ &= \sum_{\{i_1, \dots, i_n\} \subset [N]} |\Delta(x_{i_1}, \dots, x_{i_n})|^2 \prod_{k=1}^n w_{i_k} > 0. \end{aligned} \quad (5.2.14)$$

The particular case $n = N$ yields

$$|\underline{H}_{2N-2}| = |\Delta(x_1, \dots, x_N)|^2 \prod_{n=1}^N w_n.$$

In the same vein, the two other determinants are obtained starting from

$$|\underline{H}_{2N-1}| = \left| \sum_{n=1}^N w_n x_n^{i+j} x_n \right|_{i,j=0}^{N-1} \quad \text{and} \quad |\overline{H}_{2N-1}| = \left| \sum_{n=1}^N w_n x_n^{i+j} (1 - x_n) \right|_{i,j=0}^{N-1}.$$

For $n > N$, (5.2.14) clearly shows that \underline{H}_{2n-2} is rank deficient. \square

Moment matrices also provide an alternative description of orthogonal polynomials.

Proposition 5.2.6. *The monic polynomials $(P_n)_{n=0}^N$ orthogonal with respect to $\mu = \sum_{n=1}^N w_n \delta_{x_n}$ admit the following expression*

$$P_0 = 1 \quad \text{and} \quad P_n(x) = \frac{1}{|\underline{H}_{2n-2}|} \begin{vmatrix} 1 & & & \\ \underline{H}_{2n-2} & & & \\ \vdots & & & \\ m_n & \cdots & m_{2n-1} & x^n \end{vmatrix}, \quad \forall 1 \leq n \leq N. \quad (5.2.15)$$

Besides,

$$\|P_0\|_\mu^2 = 1 \quad \text{and} \quad \|P_n\|_\mu^2 = \frac{|\underline{H}_{2n}|}{|\underline{H}_{2n-2}|}, \quad \forall 1 \leq n \leq N. \quad (5.2.16)$$

In particular, $P_N(x) = \prod_{n=1}^N (x - x_n)$ is the zero vector of $L^2(\mu)$.

Proof. The previous Lemma 5.2.5 validates the definition of $(P_n)_{n=0}^N$ as a sequence of monic polynomials with $\deg P_n = n$ since the denominator $|\underline{H}_{2n-2}| > 0$. They are also mutually orthogonal. To see this, let $1 \leq n \leq N$, then

$$\langle P_n, x^k \rangle_\mu = \frac{1}{|\underline{H}_{2n-2}|} \begin{vmatrix} m_0 & \cdots & m_{n-1} & m_k \\ \vdots & & \vdots & \vdots \\ m_n & \cdots & m_{2n-1} & m_{n+k} \end{vmatrix} = 0, \quad \forall k < n.$$

Moreover, $\forall 1 \leq n \leq N$,

$$\|P_n\|_\mu^2 = \langle P_n, P_n \rangle_\mu = \langle P_n, x^n \rangle_\mu = \frac{|\underline{H}_{2n}|}{|\underline{H}_{2n-2}|}.$$

Then, Lemma 5.2.5 yields $\|P_N\|_\mu^2 = \frac{|\underline{H}_{2N}|}{|\underline{H}_{2N-2}|} = 0$. Thus, the distinct support points of μ are zeros of P_N . But the latter is monic with $\deg P_N = N$, hence $P_N = \prod_{n=1}^N (x - x_n)$. \square

The next result further relates moment matrices and the recurrence coefficients.

Lemma 5.2.7. *The moment matrix \underline{H}_{2N-2} associated to $\mu = \sum_{n=1}^N w_n \delta_{x_n}$ has determinant*

$$|\underline{H}_{2N-2}| = |\Delta(x_1, \dots, x_N)|^2 \prod_{n=1}^N w_n = \prod_{n=1}^{N-1} b_n^{N-n}. \quad (5.2.17)$$

Proof. The first equality was established in Lemma 5.2.5. The second results from a simple recursion combining (5.2.4) (5.2.16). For any $1 \leq k \leq N-1$,

$$\|P_k\|_\mu^2 = \frac{|\underline{H}_{2k}|}{|\underline{H}_{2k-2}|} = \prod_{n=1}^k b_n \implies |\underline{H}_{2k}| = \prod_{\ell=1}^k \prod_{n=1}^\ell b_n = \prod_{n=1}^k b_n^{k+1-n}. \quad (5.2.18)$$

\square

In order to generate samples from a β -ensemble, the previous Lemma 5.2.7 already hints what tridiagonal models can achieve: if we see the β -ensemble as the support of a random atomic measure, which is parametrized by its recurrence coefficients, then the complex interaction term that is the Vandermonde determinant in (5.0.1) gets replaced by a simple product of powers of b_n s. This intuition, formalized in Theorem 5.1.4, requires to make explicit the change of variables between the nodes and weights of μ and the recurrence coefficients of the corresponding Jacobi matrix $J_{\mathbf{a}, \mathbf{b}}$.

5.3 MAKING THE CHANGE OF VARIABLES

To compute the Jacobian of Favard's map $(x_{1:N}, \omega_{1:N}) \mapsto (a_{1:N}, b_{1:N-1})$, defined in Theorem 5.2.3, we first compute the Jacobian of the moment map $(x_{1:N}, \omega_{1:N}) \mapsto (m_{1:2N-1})$, and then use the lattice path construction of Hardy (2017) to express the Jacobian of $(m_{1:2N-1}) \mapsto (a_{1:N}, b_{1:N-1})$. We mention that the overall Jacobian has already been derived, in a more concise style, by Forrester and Rains (2006) and Krishnapur, Rider, and Virág (2016). Our contribution in this section is to give all details while remaining as elementary as possible. In particular, we only rely on Favard's theorem for atomic measures, and the proof of Theorem 5.1.4 boils down to checking that the changes of variables are C^1 -diffeomorphisms.

Let $\phi : \mathbb{R}_>^N \times S_N \rightarrow \mathbb{R}^{2N-1}$ map a set of N distinct atoms and $N - 1$ positive weights to their moments (m_k) . Let $\mathcal{M} \subset \mathbb{R}^{2N-1}$ be the image of ϕ .

Proposition 5.3.1 (From atomic measures to moments). $\mathcal{M} \subset \mathbb{R}^{2N-1}$ is open, ϕ is a C^1 -diffeomorphism from $\mathbb{R}_>^N \times S_N$ onto \mathcal{M} , and

$$\left| \frac{\partial m_{1:2N-1}}{\partial x_{1:N}, w_{1:N-1}} \right| = |\Delta(x_1, \dots, x_N)|^4 \prod_{n=1}^N w_n = \frac{|\underline{H}_{2N-2}|^2}{\prod_{n=1}^N w_n}, \quad (5.3.1)$$

where the Hankel matrix \underline{H}_{2N-2} is defined by (5.2.8).

Proof. Moments define monic OPs; see Proposition 5.2.15. By Favard's Theorem 5.2.3, monic OPs in turn define the atoms and weights of μ uniquely. Thus, ϕ is injective. Moreover $\mathbb{R}_>^N \times S_N \subset \mathbb{R}^{2N-1}$ is open, and ϕ is C^1 . By the classical inverse function theorem, see e.g., Cartan (1971, Corollary 4.2.2), it is thus enough to show that the Jacobian of ϕ never vanishes.

The i -th moment of μ can be written in two forms

$$m_i = \sum_{j=1}^N w_j x_j^i = \sum_{j=1}^{N-1} w_j (x_j^i - x_N^i) + x_N^i, \quad (5.3.2)$$

so that

$$\frac{\partial m_i}{\partial x_j} = i w_j x_j^{i-1} \quad \text{and} \quad \frac{\partial m_i}{\partial w_j} = x_j^i - x_N^i. \quad (5.3.3)$$

Thus,

$$\begin{aligned} \left| \frac{\partial m_{1:2N-1}}{\partial x_{1:N}, w_{1:N-1}} \right| &= \left| \begin{bmatrix} \left[\frac{\partial m_i}{\partial x_j} \frac{\partial m_i}{\partial w_j} \right]_{j=1}^{N-1} & \left[\frac{\partial m_i}{\partial x_N} \right]_{i=1}^{2N-1} \end{bmatrix} \right| \\ &= \left| \begin{bmatrix} [i w_j x_j^{i-1} \ x_j^i - x_N^i]_{j=1}^{N-1} & i w_N x_N^{i-1} \end{bmatrix}_{i=1}^{2N-1} \right| \\ &= \left| \begin{bmatrix} [i x_j^{i-1} \ x_j^i - x_N^i]_{j=1}^{N-1} & i x_N^{i-1} \end{bmatrix}_{i=1}^{2N-1} \right| \times \prod_{n=1}^N w_n \\ &= \left| \begin{bmatrix} [(i-1)x_j^{i-2} \ x_j^{i-1} - x_N^{i-1}]_{j=1}^{N-1} & (i-1)x_N^{i-2} \ x_N^{i-1} \end{bmatrix}_{i=1}^{2N} \right| \times \prod_{n=1}^N w_n \\ &= \left| \begin{bmatrix} (i-1)x_j^{i-2} \ x_j^{i-1} \end{bmatrix}_{i=1, j=1}^{2N, N} \right| \prod_{n=1}^N w_n. \end{aligned} \quad (5.3.4)$$

The last equality is obtained by adding the last column to all other even columns. The determinant in (5.3.4) is called a *confluent* Vandermonde determinant, and has closed form expression

$$\left| \begin{bmatrix} (i-1)x_j^{i-2} & x_j^{i-1} \end{bmatrix}_{i=1,j=1}^{2N,N} \right| = \prod_{1 \leq i < j \leq N} (x_j - x_i)^{2 \times 2} = |\Delta(x_1, \dots, x_N)|^4,$$

see, e.g., Ha and Gibson (1980, Corollary 1, with $\eta_i \equiv 2$) In particular, (5.3.4) never vanishes on $\mathbb{R}_>^N \times S_N$. \square

Let us now consider the map

$$\rho : \mathcal{M} \rightarrow \mathbb{R}^N \times (0, +\infty)^{N-1}, \quad (5.3.5)$$

that takes moments $m_{1:2N-1}$ and returns the recurrence coefficients $(a_{1:N}, b_{1:N-1})$.

Proposition 5.3.2 (From recurrence coefficients to moments). ρ is a C^1 -diffeomorphism from \mathcal{M} onto $\mathbb{R}^N \times (0, +\infty)^{N-1}$, and

$$\left| \frac{\partial m_{1:2N-1}}{\partial a_{1:N}, b_{1:N-1}} \right| = \prod_{n=1}^{N-1} b_n^{2(N-n)-1} = \frac{|\underline{H}_{2N-2}|^2}{\prod_{n=1}^{N-1} b_n}, \quad (5.3.6)$$

where the Hankel matrix \underline{H}_{2N-2} is defined by (5.2.8).

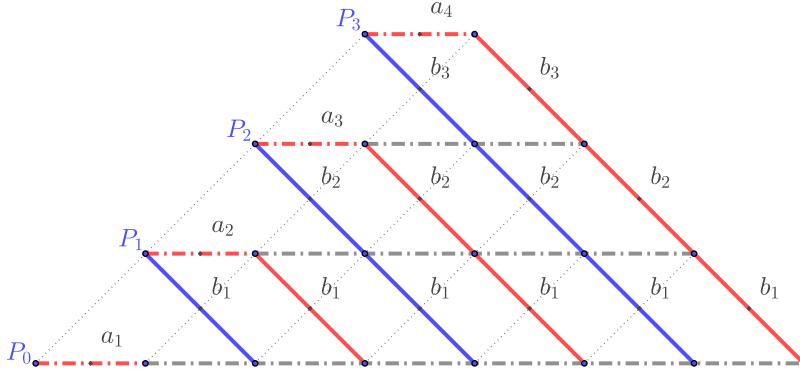
Proof. Using Theorem 5.2.3 and Proposition 5.3.1, $\rho = \psi \circ \phi^{-1}$, so that ρ is bijective. As in the proof of Proposition 5.3.1, we apply the inverse function theorem (Cartan, 1971, Corollary 4.2.2), but this time to ρ^{-1} . We first note that $\mathbb{R}^N \times (0, +\infty)^{N-1} \subset \mathbb{R}^{2N-1}$ is open. It is thus enough to show that ρ^{-1} is C^1 and that its Jacobian never vanishes. To this end, we borrow an elegant lattice path representation of the recurrence relations for OPs from Hardy (2017, Equation 1.8). This allows us to express the successive moments as polynomials in the recurrence coefficients.

To provide intuition, we first compute the first few moments by hand, recursively applying the recurrence relation (5.2.2). It comes

$$\begin{aligned} m_1 &= \langle xP_0, P_0 \rangle = 1 \cdot \cancel{\langle P_1, P_0 \rangle} + a_1 \cdot \langle P_0, P_0 \rangle + 0 = \textcolor{red}{a_1}, \\ m_2 &= \langle x^2 P_0, P_0 \rangle = 1 \cdot \langle xP_1, P_0 \rangle + a_1 \cdot \langle xP_0, P_0 \rangle + 0 \\ &\quad = 1 \cdot (1 \cdot \cancel{\langle P_2, P_0 \rangle} + a_2 \cdot \cancel{\langle P_1, P_0 \rangle} + b_1 \langle P_0, P_0 \rangle) \\ &\quad \quad + a_1 \cdot (1 \cdot \cancel{\langle P_1, P_0 \rangle} + a_1 \cdot \langle P_0, P_0 \rangle + 0) \\ &\quad = \textcolor{blue}{1 \cdot b_1} + a_1 \cdot a_1, \\ m_3 &= \langle x^3 P_0, P_0 \rangle = 1 \cdot \langle x^2 P_1, P_0 \rangle + a_1 \cdot \langle x^2 P_0, P_0 \rangle + 0 \\ &\quad = 1 \cdot (1 \cdot \langle xP_2, P_0 \rangle + a_2 \cdot \langle xP_1, P_0 \rangle + b_1 \cdot \langle xP_0, P_0 \rangle) \\ &\quad \quad + a_1 \cdot (1 \cdot \langle xP_1, P_0 \rangle + a_1 \cdot \langle xP_0, P_0 \rangle + 0) \\ &\quad = 1 \cdot 1 \cdot (1 \cdot \cancel{\langle P_3, P_0 \rangle} + a_3 \cdot \cancel{\langle P_2, P_0 \rangle} + b_3 \cdot \cancel{\langle P_1, P_0 \rangle}) \\ &\quad \quad + 1 \cdot a_2 \cdot (1 \cdot \cancel{\langle P_2, P_0 \rangle} + a_2 \cdot \cancel{\langle P_1, P_0 \rangle} + b_1 \cdot \langle P_0, P_0 \rangle) \\ &\quad \quad + 1 \cdot b_1 \cdot (1 \cdot \cancel{\langle P_1, P_0 \rangle} + a_1 \cdot \langle P_0, P_0 \rangle + 0) \\ &\quad \quad + a_1 \cdot 1 \cdot (1 \cdot \cancel{\langle P_2, P_0 \rangle} + a_2 \cdot \cancel{\langle P_1, P_0 \rangle} + b_1 \cdot \langle P_0, P_0 \rangle) \\ &\quad \quad + a_1 \cdot a_1 \times (1 \cdot \cancel{\langle P_1, P_0 \rangle} + a_1 \cdot \langle P_0, P_0 \rangle + 0) \\ &\quad = \textcolor{red}{1 \cdot a_2 \cdot b_1} + a_1 \cdot 1 \cdot b_1 + 1 \cdot b_2 \cdot a_1 + a_1 \cdot a_1 \cdot a_1. \end{aligned} \quad (5.3.7)$$

More generally, when computing $m_k = \langle x^k P_0, P_0 \rangle$, the recursive application of the recurrence relation (5.2.2) allows to decrease the power of x from k to 0 until each term in the development is proportional to the inner product of $P_0 = 1$ with another monic OP. The only nonzero such inner product is $\langle P_0, P_0 \rangle = 1$. Consequently, each nonzero term in the final development of m_k corresponds to a path of length at most k

that leaves from the lower left corner of the graph in Figure 5.1 and ends up on the bottom row. In between, the path has to remain above the bottom row, and can only move North-East, East, or South-East. Each edge corresponds to picking one of the three terms in the recurrence relation (5.2.2). For example, the development of m_3 in (5.3.7) corresponds to three such paths, shown in orange in Figure 5.1. The product of the coefficients along each path forms the resulting term in the development.



(a) The North-East, East, South-East edges associated to weights 1, a_n , b_n are respectively represented as dashed, dashdotted and solid lines. Note that on each dashed and dashdotted line, the weight is constant.

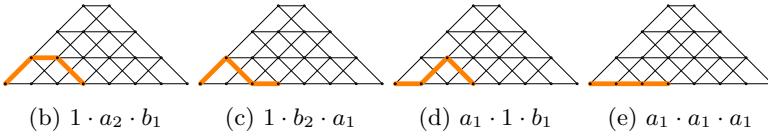


Figure 5.1: The lattice path of Hardy (2017) used to compute $m_n = \langle x^n P_0, P_0 \rangle$ is displayed in (a). The paths used for the computation of m_3 (5.3.7) are highlighted in (b)-(e) with the corresponding weight as caption.

In the end, odd moments m_{2i-1} , resp. even moments m_{2i} , are the sum of the weights of the paths below the i -th red, respectively blue path, counting from the bottom left. More precisely,

$$m_{2i-1} = a_i \prod_{k=1}^{i-1} b_k + f_1(a_{1:i-1}, b_{1:i-2}) \quad \text{and} \quad m_{2i} = \prod_{k=1}^i b_k + f_2(a_{1:i}, b_{1:i-1}). \quad (5.3.8)$$

Thus, the Jacobian is the determinant of a triangular matrix

$$\left| \frac{\partial m_{1:2N-1}}{\partial a_{1:N}, b_{1:N-1}} \right| = \left| \begin{bmatrix} \frac{\partial m_{2i-1}}{\partial a_j} & \frac{\partial m_{2i-1}}{\partial b_j} \\ \frac{\partial m_{2i}}{\partial a_j} & \frac{\partial m_{2i}}{\partial b_j} \\ \frac{\partial m_{2N-1}}{\partial a_j} & \frac{\partial m_{2N-1}}{\partial b_j} \end{bmatrix} \right|_{i,j=1}^{N-1} = \prod_{i=1}^N \frac{\partial m_{2i-1}}{\partial a_i} \prod_{i=1}^{N-1} \frac{\partial m_{2i}}{\partial b_i}.$$

The formulation (5.3.8) yields

$$\frac{\partial m_{2i-1}}{\partial a_i} = \prod_{k=1}^{i-1} b_k \quad \text{and} \quad \frac{\partial m_{2i}}{\partial b_i} = \prod_{k=1}^{i-1} b_k.$$

Finally, we obtain

$$\left| \frac{\partial m_{1:2N-1}}{\partial a_{1:N}, b_{1:N-1}} \right| = \prod_{i=1}^N \prod_{k=1}^{i-1} b_k \prod_{i=1}^{N-1} \prod_{k=1}^{i-1} b_k = \frac{\left[\prod_{i=1}^N \prod_{k=1}^{i-1} b_k \right]^2}{\prod_{k=1}^{N-1} b_k} = \frac{\left[\prod_{n=1}^{N-1} b_n^{N-n} \right]^2}{\prod_{n=1}^{N-1} b_n},$$

which does not vanish since all b_n s are positive by construction. Finally, the last equality in (5.3.6) follows from Lemma 5.2.7. \square

Propositions 5.3.1 and 5.3.2 now allow us to conclude that Favard's map $\psi = \rho \circ \phi$ (cf. Theorem 5.2.3) is a C^1 -diffeomorphism, and compute its Jacobian.

Proposition 5.3.3. Favard's map ψ is a C^1 -diffeomorphism from $\mathbb{R}_>^N \times S_N$ onto $\mathbb{R}^N \times (0, +\infty)^{N-1}$, and

$$\left| \frac{\partial x_{1:N}, w_{1:N-1}}{\partial a_{1:N}, b_{1:N-1}} \right| = \prod_{n=1}^{N-1} b_n^{-1} \prod_{n=1}^N w_n. \quad (5.3.9)$$

We now have all the ingredients to give an explicit proof of Theorem 5.1.4, of which the three classical tridiagonal models of Section 5.4 will be seen to be corollaries.

Proof of Theorem 5.1.4. For simplicity we drop the indicator functions and rewrite the density of the nodes and weights as

$$(5.1.7) = \left(|\Delta(x_1, \dots, x_N)|^2 \prod_{n=1}^N w_n \right)^{\frac{\beta}{2}} e^{-\text{Tr}[V(\text{diag}(x_1, \dots, x_N))]} \prod_{n=1}^N w_n^{-1} dx_{1:N} dw_{1:N-1}$$

Combining Lemma 5.2.7, and the fact that x_1, \dots, x_N are the eigenvalues of $J_{\mathbf{a}, \mathbf{b}}$, the change of variables provided by Proposition 5.3.3 yields

$$\begin{aligned} (5.1.7) &= \left(\prod_{n=1}^{N-1} b_n^{N-n} \right)^{\frac{\beta}{2}} e^{-\text{Tr}[V(J_{\mathbf{a}, \mathbf{b}})]} \prod_{n=1}^N w_n^{-1} \left| \frac{\partial x_{1:N}, w_{1:N-1}}{\partial a_{1:N}, b_{1:N-1}} \right| da_{1:N} db_{1:N-1} \\ &= \prod_{n=1}^{N-1} b_n^{\frac{\beta}{2}(N-n)-1} e^{-\text{Tr}[V(J_{\mathbf{a}, \mathbf{b}})]} da_{1:N} db_{1:N-1}, \end{aligned}$$

where the last equality follows from (5.3.9). \square

5.4 PROVING THE THREE CLASSICAL TRIDIAGONAL MODELS

Theorem 5.1.4 gives the distribution over recurrence coefficients, from which one has to sample, in order for the atoms of the corresponding atomic measure to follow a given β -ensemble. When the potential of the β -ensemble is taken among three particular forms, the recurrence coefficients turn out to be independent with simple distributions. In particular, the recurrence coefficients are much simpler to sample than the complex joint distribution (5.0.1) of the atoms.

5.4.1 The $H\beta E$ and its tridiagonal model

The tridiagonal model associated to the Hermite β -ensemble, cf. Theorem 5.1.1, follows from a direct application of Theorem 5.1.4 and the following immediate lemma.

Lemma 5.4.1. Let $J_{\mathbf{a}, \mathbf{b}}$ be a Jacobi matrix as defined by (5.1.1), with eigenvalues x_1, \dots, x_N . It holds that

$$\sum_{n=1}^N x_n = \text{Tr } J_{\mathbf{a}, \mathbf{b}} = \sum_{n=1}^N a_n \quad \text{and} \quad \sum_{n=1}^N x_n^2 = \text{Tr } J_{\mathbf{a}, \mathbf{b}}^2 = \sum_{n=1}^N a_n^2 + 2 \sum_{n=1}^{N-1} b_n. \quad (5.4.1)$$

Proof of Theorem 5.1.1. Starting from Theorem 5.1.4 it remains to express the term $\text{Tr } V(J_{\mathbf{a}, \mathbf{b}})$, where $V(x) = \frac{(x-\mu)^2}{2\sigma^2} = \frac{1}{2\sigma^2}(x^2 - 2\mu x + \mu^2)$. To this end, Lemma 5.4.1 yields

$$\text{Tr } V(J_{\mathbf{a}, \mathbf{b}}) = \frac{1}{2\sigma^2} [\text{Tr } J_{\mathbf{a}, \mathbf{b}}^2 - 2\mu \text{Tr } J_{\mathbf{a}, \mathbf{b}} + N\mu^2] = \frac{1}{2\sigma^2} \sum_{n=1}^N (a_n - \mu)^2 + \frac{1}{\sigma^2} \sum_{n=1}^{N-1} b_n.$$

Finally, we can plug this expression back into (5.1.7) to see that the entries of $J_{\mathbf{a}, \mathbf{b}}$ are independently distributed, with joint distribution proportional to

$$\prod_{n=1}^{N-1} b_n^{\frac{\beta}{2}(N-n)-1} e^{-\frac{1}{\sigma^2} b_n} db_n \prod_{n=1}^N e^{-\frac{1}{2\sigma^2}(a_n - \mu)^2} da_n. \quad (5.4.2)$$

\square

Note that when μ is still supported on \mathbb{R} , but the potential is a more general polynomial, the recurrence parameters are no longer independent, but the interaction remains short range. This is what we later exploit in Section 5.5, where we derive a fast approximate sampler for various β -ensembles with polynomial potentials.

5.4.2 The L β E and its tridiagonal model

When the target β -ensemble is supported on $(0, +\infty)$, there is a natural reparametrization of the recurrence coefficients of μ , which allows to express other quantities than those in Lemma 5.4.1. This leads to the tridiagonal model of Theorem 5.1.2 for the Laguerre β -ensemble.

The reparametrization, denoted by $\xi_1, \dots, \xi_{2N-1} > 0$, arose in the work of Stieltjes (1894) on continued fractions; see also Chihara (1978; 1971, Equation 9.12 in Corollary of Theorem 9.1; Equation 2) in the context of three-term recurrence relations. To introduce these new parameters, first note that since μ is now supported on $(0, +\infty)$, the recurrence relation (5.2.2) implies

$$a_n = \|P_{n-1}\|^{-2} \langle x P_{n-1}, P_{n-1} \rangle = \|P_{n-1}\|^{-2} \int_0^\infty x P_{n-1}^2(x) \mu(dx) > 0, \quad n = 1, \dots, N. \quad (5.4.3)$$

Now, we set

$$\begin{aligned} a_1 &= \xi_1, \quad a_n = \xi_{2n-2} + \xi_{2n-1}, \quad \text{for } 2 \leq n \leq N, \\ \text{and } b_n &= \xi_{2n-1} \xi_{2n}, \quad \text{for } 1 \leq n \leq N-1. \end{aligned} \quad (5.4.4)$$

Equivalently, the new parameters correspond to the Cholesky factorization

$$J_{\mathbf{a}, \mathbf{b}} = \Xi \Xi^\top, \quad \text{where } \Xi = \begin{pmatrix} \sqrt{\xi_1} & & & (0) \\ \sqrt{\xi_2} & \sqrt{\xi_3} & & \\ \ddots & \ddots & \ddots & \\ (0) & & \sqrt{\xi_{2N-2}} & \sqrt{\xi_{2N-1}} \end{pmatrix}. \quad (5.4.5)$$

Note that this bidiagonal transformation is reminiscent of the construction of the tridiagonal model for the L β E, where Dumitriu and Edelman (2002) bidiagonalize a random Gaussian matrix. The following proposition shows that the change of variables replacing the recurrence coefficients by $\xi_{1:2N-1}$ is valid.

Proposition 5.4.2. *Consider μ supported on $(0, +\infty)$, then the corresponding Jacobi matrix (5.1.1) factorizes uniquely as $J_{\mathbf{a}, \mathbf{b}} = \Xi \Xi^\top$, where Ξ is given by (5.4.5). Moreover, the mapping*

$$(\xi_1, \dots, \xi_{2N-1}) \mapsto (a_{1:N}, b_{1:N-1}), \quad (5.4.6)$$

defined by (5.4.4) is a C^1 -diffeomorphism of $(0, +\infty)^{2N-1}$ onto itself, and its Jacobian reads

$$\left| \frac{\partial a_{1:N}, b_{1:N-1}}{\partial \xi_{1:2N-1}} \right| = \prod_{i=1}^{N-1} \xi_{2i-1}. \quad (5.4.7)$$

Proof. Given that $J_{\mathbf{a}, \mathbf{b}}$ is symmetric with positive eigenvalues, the Cholesky factorization $J_{\mathbf{a}, \mathbf{b}} = \Xi \Xi^\top$ is unique, see, e.g., Golub and Van Loan (2013, Theorem 4.2.7). Moreover, since $J_{\mathbf{a}, \mathbf{b}}$ is tridiagonal, the factor Ξ can only be bidiagonal. Hence, the mapping (5.4.6) is injective (and even bijective) and C^1 because it is polynomial. Finally, by definition of the transformation (5.4.4), the Jacobian reads as the determinant of a triangular matrix

$$\left| \frac{\partial a_{1:N}, b_{1:N-1}}{\partial \xi_{1:2N-1}} \right| = \left| \begin{bmatrix} \left[\frac{\partial a_i}{\partial \xi_{2j-1}}, \frac{\partial a_i}{\partial \xi_{2j}} \right]_{i,j=1}^{N-1} & \left[\frac{\partial a_i}{\partial \xi_{2N-1}}, \frac{\partial a_i}{\partial b_i} \right]_{i=1}^{N-1} \\ \left[\frac{\partial b_i}{\partial \xi_{2j-1}}, \frac{\partial b_i}{\partial \xi_{2j}} \right]_{i,j=1}^{N-1} & \frac{\partial a_N}{\partial \xi_{2N-1}} \end{bmatrix} \right| = \prod_{i=1}^N \underbrace{\frac{\partial a_i}{\partial \xi_{2i-1}}}_{=1} \prod_{i=1}^{N-1} \underbrace{\frac{\partial b_i}{\partial \xi_{2i}}}_{=\xi_{2i-1}}.$$

□

For our purpose, the Cholesky factorization (5.4.5) is ideal to express the key quantities that appear in the L β E. The proof of the corresponding tridiagonal model, cf. Theorem 5.1.2, follows from a direct application of Theorem 5.1.4 and the following immediate lemma.

Lemma 5.4.3. *Let $J_{\mathbf{a}, \mathbf{b}} = \Xi \Xi^T$ as in (5.4.5) and note x_1, \dots, x_N its eigenvalues. Then,*

$$\sum_{n=1}^N x_n = \text{Tr } J_{\mathbf{a}, \mathbf{b}} = \sum_{n=1}^{2N-1} \xi_n \quad \text{and} \quad \prod_{n=1}^N x_n = \det J_{\mathbf{a}, \mathbf{b}} = \prod_{n=1}^N \xi_{2n-1}. \quad (5.4.8)$$

Proof of Theorem 5.1.2. Applying Lemma 5.4.3 to the $V(x) = -(k-1) \log(x) + \frac{x}{\theta}$ yields

$$\begin{aligned} \exp[-\text{Tr } V(J_{\mathbf{a}, \mathbf{b}})] &= (\det J_{\mathbf{a}, \mathbf{b}})^{k-1} \exp\left(-\frac{1}{\theta} \text{Tr } J_{\mathbf{a}, \mathbf{b}}\right) \\ &\stackrel{(5.4.8)}{=} \prod_{n=1}^N \xi_{2n-1}^{k-1} \exp\left(-\frac{1}{\theta} \sum_{n=1}^{2N-1} \xi_n\right). \end{aligned} \quad (5.4.9)$$

Starting from (5.1.8), Proposition 5.4.2 gives the joint distribution of the underlying $\xi_{1:2N-1}$ parameters as proportional to

$$\begin{aligned} &\prod_{n=1}^{N-1} b_n^{\frac{\beta}{2}(N-n)-1} e^{-\text{Tr } V(J_{\mathbf{a}, \mathbf{b}})} da_{1:N} db_{1:N-1} \\ &\stackrel{(5.4.4)}{=} \prod_{n=1}^{N-1} (\xi_{2n-1} \xi_{2n})^{\frac{\beta}{2}(N-n)-1} e^{-\text{Tr } V(J_{\mathbf{a}, \mathbf{b}})} \left| \frac{\partial a_{1:N}, b_{1:N-1}}{\partial \xi_{1:2N-1}} \right| d\xi_{1:2N-1} \\ &\stackrel{(5.4.7)}{=} \prod_{n=1}^{N-1} \xi_{2n-1}^{\frac{\beta}{2}(N-n)-1} \cancel{\xi_{2n}^{\frac{\beta}{2}(N-n)-1}} e^{-\text{Tr } V(J_{\mathbf{a}, \mathbf{b}})} \prod_{n=1}^{N-1} \cancel{\xi_{2n-1}} d\xi_{1:2N-1} \\ &\stackrel{(5.4.9)}{=} \prod_{n=1}^N \xi_{2n-1}^{\frac{\beta}{2}(N-n)} \prod_{n=1}^{N-1} \xi_{2n}^{\frac{\beta}{2}(N-n)-1} \left(\prod_{n=1}^N \xi_{2n-1} \right)^{k-1} e^{-\frac{1}{\theta} \sum_{n=1}^{2N-1} \xi_n} d\xi_{1:2N-1} \end{aligned} \quad (5.4.10)$$

□

In the next section, we introduce another reparametrization of the recurrence coefficients, this time when μ is supported in a compact interval: the canonical moments of Dette and Studden (1997).

5.4.3 The J β E and its tridiagonal model

Finding a tridiagonal model for the J β E was left as an open problem by Dumitriu and Edelman (2002, IV B). The latter was addressed by Killip and Nenciu (2004, Theorem 2) in their study of the quindiaagonal model associated to the circular ensemble. However, the authors acknowledged that lifting the points on the unit circle to apply their result represents a winding detour to prove the J β E. Besides, the Jacobian required by this method was obtained by indirect means by Killip and Nenciu (2007, Lemma 4.3). Subsequently, Forrester and Rains (2006, Theorem 2) obtained the Jacobian more directly.

We can actually derive the tridiagonal model of Theorem 5.1.3 by reparametrizing the Jacobi matrix $J_{\mathbf{a}, \mathbf{b}}$ again, this time using *canonical moments* (Dette and Studden, 1997, Chapter 1). In essence, the result can be found in the work of Gamboa and Rouault (2010) and Dette and Nagel (2012), but we rephrase it as just another consequence of Theorem 5.1.4.

Before formally introducing them, let us mention that canonical moments and their complex counterpart were successfully used to investigate the connection between randomized moments problems, orthogonal polynomials, and optimal design (Dette and Studden, 1997) and in random matrix theory (Gamboa and Rouault, 2010; Gamboa, Nagel, and Rouault, 2016). In particular, canonical moments can be thought of as a reparametrization of the moments, where $0 < c_n < 1$ represents the relative position of the n -th

moment m_n in the range of all possible moments associated to measure with compatible previous moments m_1, \dots, m_{n-1} , see Dette and Studden (1997).

Throughout this section, we assume that the N -atomic measure μ is supported on $(0, 1)$. In particular, with (ξ_n) the parameters introduced in Section 5.4.2, it comes

$$0 < \xi_{2n-2} + \xi_{2n-1} = a_n = \|P_{n-1}\|^{-2} \langle x P_{n-1}, P_{n-1} \rangle < 1, \quad n = 2, \dots, N.$$

Similarly, $\xi_1 = a_1 \in (0, 1)$. This implies $0 < \xi_n < 1$ for all $1 \leq n \leq 2N - 1$. Following the work of Wall (1940) on chain sequences and continued fractions, we introduce a new parametrization of the recurrence coefficients.

Lemma 5.4.4 (Wall). *Assume μ is supported on $(0, 1)$, there exists a sequence $(c_n) \in (0, 1)^{\mathbb{N}}$ such that*

$$\xi_1 = c_1 \quad \text{and} \quad \xi_n = (1 - c_{n-1})c_n, \quad \forall 2 \leq n \leq 2N - 1. \quad (5.4.11)$$

We do not prove Lemma 5.4.4 and refer to Wall (1940, Theorem 6.1); see also Chihara (1978, Chapter 3) for more details on chain sequences. We simply note that defining (c_n) in (5.4.11) is straightforward, the nontrivial part of the lemma is that $0 < c_n < 1$ for all n . We also note that the c_n s are today known as the *canonical moments* of μ ; see the monograph of Dette and Studden (1997).

The following proposition shows that the change of variables replacing ξ by c is valid.

Proposition 5.4.5. *Consider μ supported on $(0, 1)$, then the corresponding Jacobi matrix (5.1.1) can be parametrized in terms of the canonical moments following (5.4.4) and (5.4.11). Moreover, the mapping*

$$(c_1, \dots, c_{2N-1}) \mapsto (\xi_1, \dots, \xi_{2N-1}), \quad (5.4.12)$$

defined by (5.4.11) is a C^1 -diffeomorphism of $(0, 1)^{2N-1}$ onto itself, and its Jacobian reads

$$\left| \frac{\partial \xi_{1:2N-1}}{\partial c_{1:2N-1}} \right| = \prod_{n=1}^{2N-2} (1 - c_n). \quad (5.4.13)$$

Proof. The map (5.4.12) is a bijection by definition and Lemma 5.4.4, and C^1 because it is polynomial. Then, by definition of the transformation (5.4.11), the Jacobian is the determinant of a triangular matrix

$$\left| \frac{\partial \xi_{1:2N-1}}{\partial c_{1:2N-1}} \right| = \prod_{n=1}^{2N-1} \frac{\partial \xi_n}{\partial c_n} = 1 \cdot \prod_{n=2}^{2N-1} (1 - c_{n-1}) = \prod_{n=1}^{2N-2} (1 - c_n).$$

□

For our purpose, the canonical moment parametrization is ideal to express the key quantities that appear in the J β E. The proof of the corresponding tridiagonal model in Theorem 5.1.3 is again a direct application of Proposition 5.4.5 and the following lemma.

Lemma 5.4.6. *It holds that*

$$\prod_{n=1}^N x_n = \det J_{\mathbf{a}, \mathbf{b}} = \prod_{n=1}^N c_{2n-1} \prod_{n=1}^{N-1} (1 - c_{2n}) \quad \text{and} \quad \prod_{n=1}^N (1 - x_n) = \det [I_N - J_{\mathbf{a}, \mathbf{b}}] = \prod_{n=1}^{2N-1} (1 - c_n). \quad (5.4.14)$$

Proof. First, combine the result of Lemma 5.4.3 and the definition of the canonical moments in (5.4.11) to get

$$\prod_{n=1}^N x_n = \xi_1 \prod_{n=2}^N \xi_{2n-1} = c_1 \prod_{n=2}^N (1 - c_{2n-2})c_{2n-1} = \prod_{n=1}^N c_{2n-1}(1 - c_{2n}). \quad (5.4.15)$$

Then, Lemma 5.2.5 yields

$$\prod_{n=1}^N (1 - x_n) = \frac{|\overline{H}_{2N-1}|}{|\underline{H}_{2N-2}|}. \quad (5.4.16)$$

The denominator can be expressed in terms of the $\xi_{1:2N-1}$ parameters

$$\left| \underline{H}_{2N-2} \right|^2 \stackrel{(5.2.17)}{=} \prod_{n=1}^{N-1} b_n^{N-n} \stackrel{(5.4.4)}{=} \prod_{n=1}^{N-1} [\xi_{2n-1} \xi_{2n}]^{N-n}. \quad (5.4.17)$$

For the numerator, we follow Dette and Studden (1997, Theorem 1.4.10) who introduced additional quantities $\gamma_{1:2N-1}$ to get

$$\left| \bar{H}_{2N-1} \right| = \gamma_1^N \prod_{n=1}^{N-1} [\gamma_{2n} \gamma_{2n+1}]^{N-n}, \quad (5.4.18)$$

where

$$\begin{cases} \xi_1 = c_1 \\ \gamma_1 = 1 - c_1 \end{cases} \quad \text{and} \quad \begin{cases} \xi_n = (1 - c_{n-1}) c_n \\ \gamma_n = c_{n-1} (1 - c_n) \end{cases} \quad \forall 2 \leq n \leq 2N-1. \quad (5.4.19)$$

We plug these results back into (5.4.16), and conclude that

$$\begin{aligned} \prod_{n=1}^N (1 - x_n) &= \frac{\left| \bar{H}_{2N-1} \right|}{\left| \underline{H}_{2N-2} \right|} = \gamma_1^N \prod_{n=1}^{N-1} \left[\frac{\gamma_{2n} \gamma_{2n+1}}{\xi_{2n-1} \xi_{2n}} \right]^{N-n} \\ &= (1 - c_1)^N \left[\frac{\cancel{\gamma_1}(1 - c_2) \cancel{\gamma_2}(1 - c_3)}{\cancel{\gamma_1}(1 - c_1) \cancel{\gamma_2}} \right]^{N-1} \prod_{n=2}^{N-1} \left[\frac{\cancel{c_{2n-1}}(1 - c_{2n}) \cancel{c_{2n}}(1 - c_{2n+1})}{(1 - c_{2n-2}) \cancel{c_{2n-1}}(1 - c_{2n-1}) \cancel{c_{2n}}} \right]^{N-n} \\ &= (1 - c_1)^N \prod_{n=1}^{N-1} \left[\frac{1 - c_{2n+1}}{1 - c_{2n-1}} \right]^{N-n} (1 - c_2)^{N-1} \prod_{n=2}^{N-1} \left[\frac{1 - c_{2n}}{1 - c_{2n-2}} \right]^{N-n} \\ &= \prod_{n=1}^N (1 - c_{2n-1}) \prod_{n=1}^{N-1} (1 - c_{2n}). \end{aligned}$$

□

Proof of Theorem 5.1.3. Considering the potential $V(x) = -[(a-1) \log(x) + (b-1) \log(1-x)]$, Lemma 5.4.6 yields

$$\begin{aligned} \exp[-\operatorname{Tr} V(J_{\mathbf{a}, \mathbf{b}})] &= (\det J_{\mathbf{a}, \mathbf{b}})^{a-1} (\det [I_N - J_{\mathbf{a}, \mathbf{b}}])^{b-1} \\ &\stackrel{(5.4.14)}{=} \prod_{n=1}^N c_{2n-1}^{a-1} (1 - c_{2n-1})^{b-1} \prod_{n=1}^{N-1} (1 - c_{2n})^{a+b-2}. \end{aligned} \quad (5.4.20)$$

Starting from (5.4.10), Proposition 5.4.5 allows us to express the joint distribution of the canonical moments as

$$\begin{aligned} &\prod_{n=1}^{N-1} \frac{[\xi_{2n-1} \xi_{2n}]^{\frac{\beta}{2}(N-n)}}{\xi_{2n}} e^{-\operatorname{Tr} V(J_{\mathbf{a}, \mathbf{b}})} \left| \frac{\partial \xi_{1:2N-1}}{\partial c_{1:2N-1}} \right| dc_{1:2N-1} \\ &\stackrel{(5.4.13)}{=} \prod_{n=1}^{N-1} \frac{[\xi_{2n-1} \xi_{2n}]^{\frac{\beta}{2}(N-n)}}{\xi_{2n}} \prod_{n=1}^{2N-2} (1 - c_n) e^{-\operatorname{Tr} V(J_{\mathbf{a}, \mathbf{b}})} dc_{1:2N-1} \\ &\stackrel{(5.4.11)}{=} \frac{[c_1(1 - c_1)c_2]^{\frac{\beta}{2}(N-1)}}{(1 - \cancel{c_1})c_2} \prod_{n=2}^{N-1} \frac{[(1 - c_{2n-2})c_{2n-1}(1 - c_{2n-1})c_{2n}]^{\frac{\beta}{2}(N-n)}}{(1 - \cancel{c_{2n-1}})c_{2n}} \\ &\quad \prod_{n=1}^{N-1} \frac{(1 - \cancel{c_{2n-1}})(1 - c_{2n})}{(1 - \cancel{c_{2n-1}})} e^{-\operatorname{Tr} V(J_{\mathbf{a}, \mathbf{b}})} dc_{1:2N-1} \\ &= \prod_{n=1}^N [c_{2n-1}(1 - c_{2n-1})]^{\frac{\beta}{2}(N-n)} \prod_{n=1}^{N-1} c_{2n}^{\frac{\beta}{2}(N-n)-1} (1 - c_{2n})^{\frac{\beta}{2}(N-n-1)+1} e^{-\operatorname{Tr} V(J_{\mathbf{a}, \mathbf{b}})} dc_{1:2N-1} \\ &\stackrel{(5.4.20)}{=} \prod_{n=1}^N c_{2n-1}^{\frac{\beta}{2}(N-n)+a-1} (1 - c_{2n-1})^{\frac{\beta}{2}(N-n)+b-1} \prod_{n=1}^{N-1} c_{2n}^{\frac{\beta}{2}(N-n)-1} (1 - c_{2n})^{\frac{\beta}{2}(N-n-1)+a+b-1} dc_{1:2N-1}. \end{aligned}$$

□

5.5 GIBBS SAMPLING TRIDIAGONAL MODELS ASSOCIATED TO POLYNOMIAL POTENTIALS

For the specific potentials associated to the classical β -ensembles, the successive reparametrizations of the Jacobi matrix $J_{\mathbf{a}, \mathbf{b}}$, presented in Section 5.4, yield independent coefficients with easy-to-sample distributions. Thus, after a proper randomization of $J_{\mathbf{a}, \mathbf{b}}$, the calculation of its eigenvalues gives $\mathcal{O}(N^2)$ exact samplers. However, when the potential V is generic, these Jacobi parameters may not be independent anymore. For polynomial potentials, this dependence remains mild, in the sense that each parameter remains independent from the rest conditionally on a few “neighboring” parameters. As we shall see in this section, simple Gibbs samplers in the space of these Jacobi parameters can provide surprisingly fast-mixing approximate samplers for β -ensembles. In short, we study a Gibbs sampler on tridiagonal matrices, that can be diagonalized in time complexity $\mathcal{O}(N^2)$ to obtain approximate samples from a given β -ensemble. This approach is in contrast with that of Li and Menon (2013) and Chafaï and Ferré (2018), who used MCMC directly on the original space where the particles $\{x_1, \dots, x_N\}$ live.

Our starting point is Proposition 2 of Krishnapur, Rider, and Virág (2016), which we rederived as Theorem 5.1.4. In short, a Jacobi matrix $J_{\mathbf{a}, \mathbf{b}}$ with coefficients distributed as

$$(a_1, b_1, \dots, a_{N-1}, b_{N-1}, a_N) \sim \prod_{i=1}^{N-1} b_i^{\frac{\beta}{2}(N-i)-1} \exp^{-\text{Tr } V(J_{\mathbf{a}, \mathbf{b}})} da_{1:N}, b_{1:N-1}, \quad (5.5.1)$$

has eigenvalues distributed according to the β -ensemble (5.0.1) with potential V . Krishnapur, Rider, and Virág (2016) already mention their intuition that a Gibbs chain with invariant measure (5.5.1) and a polynomial potential would mix fast, in $\mathcal{O}(\log N)$, due to the short range interaction between the coefficients. From an algorithmic point of view, the explicit conditionals in (5.5.1) similarly invite to use a Gibbs sampler, which we investigate in this section. For the sake of presentation, we fix the potential to be a polynomial with even degree at most 6 and positive leading coefficient, i.e.,

$$V(x) = g_6 x^6 + g_5 x^5 + g_4 x^4 + g_3 x^3 + g_2 x^2 + g_1 x. \quad (5.5.2)$$

The absence of a term of degree 5 in (5.5.2) comes from practical reasons detailed in Section 5.5.1. While the method applies more generally, we restrict ourselves to potentials of the form (5.5.2) because (i) it already goes beyond the numerical state-of-the-art, (ii) it is rich enough to require different sampling schemes for different conditionals depending on the coefficients in (5.5.2), and (iii) the theory of sextic potentials is advanced enough that we have means to empirically assess the convergence of our samplers.

5.5.1 Sampling from the conditionals

We implement a systematic scan Gibbs sampler (Robert and Casella, 2004, Chapter 10) to approximately sample from the distribution (5.5.1) on the Jacobi coefficients. Writing the conditionals in closed form for the generic sextic potential (5.5.2) is cumbersome, but we do it for a specific instance in Example 5.5.1 below. The expansion of $\text{Tr } V(J_{\mathbf{a}, \mathbf{b}})$ in (5.5.1) reveals that the size of the Markov blanket of each coefficient grows with degree V . Quoting Krishnapur, Rider, and Virág (2016, Section 1), *variables with indices that are degree $V/2$ apart are conditionally independent given the variables in between*. In other words, the Jacobi coefficients $a_1, \dots, a_N, b_1, \dots, b_{N-1}$ have a more short-range interaction than the corresponding particles x_1, \dots, x_N . Gibbs sampling can leverage that property.

For $1 \leq i \leq N$, let $\mathbf{a}_{\setminus i} = (a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_N)$. Similarly, for $1 \leq j \leq N-1$, let $\mathbf{b}_{\setminus j} = (b_1, \dots, b_{j-1}, b_{j+1}, \dots, b_{N-1})$. In practice, we define one complete Gibbs pass as sampling from $a_n | \mathbf{a}_{\setminus n}, \mathbf{b}$, and then $b_n | \mathbf{a}, \mathbf{b}_{\setminus n}$, for each n in turn. We avoid the term of degree 5 in (5.5.2) to make sure that the conditionals $b_n | \mathbf{a}, \mathbf{b}_{\setminus n}$ are always log-concave, while the conditionals $a_n | \mathbf{a}_{\setminus n}, \mathbf{b}$ are log-concave if $g_2 > 0$ and $g_3 = g_6 = 0$. Univariate log-concave densities are interesting from a sampling point of view, since they are usually amenable to efficient rejection sampling.

Algorithm 12: Gibbs sampler to sample from (5.0.1) with $\beta > 0$ and V as in (5.5.2)

Require: β parameter, potential V , number T of MCMC steps

```

1: Initialize  $a_1 = \dots = a_N = b_1 = \dots = b_{N-1} = 0$ 
2: for  $t = 1$  to  $T$  do
3:   for  $n = 1$  to  $N$  do
4:     Sample  $a_n | \mathbf{a}_{\setminus n}, \mathbf{b}$ 
5:     if  $n < N$  then
6:       Sample  $b_n | \mathbf{a}, \mathbf{b}_{\setminus n}$ 
7:     end if
8:   end for
9:    $x_1^t, \dots, x_N^t = \text{eigvals}(J_{\mathbf{a}, \mathbf{b}})$ 
10: end for
```

In our case, for every log-concave conditional, the mode of the corresponding density can be derived analytically. We can thus use the tailored rejection sampler of Devroye (2012), with an expected 5 rejection steps per draw; see Example 5.5.1 for details. The overall algorithm is given in Algorithm 12.

When non log-concave conditionals $a_n | \mathbf{a}_{\setminus n}, \mathbf{b}$, we switch from a Gibbs algorithm to a Metropolis-within-Gibbs algorithm, and replace exact sampling of the corresponding conditional by a draw from a Metropolis-Hastings kernel. More specifically, since the log of the conditional densities $a_n | \mathbf{a}_{\setminus n}, \mathbf{b}$ are polynomials, they are easy to differentiate, and we use their gradient in a Metropolis-adjusted Langevin kernel (MALA, see, e.g., Robert and Casella, 2004, Section 7.8.5).

Example 5.5.1 (Quartic potential). Let $V(x) = g_4x^4 + g_2x^2$. With the convention $a_0 = a_{N+1} = b_0 = b_N = 0$, the conditionals write as follows.

For each $1 \leq n \leq N$, the conditional $a_n | \mathbf{a}_{\setminus n}, \mathbf{b}$ has density proportional to

$$a_n | \mathbf{a}_{\setminus n}, \mathbf{b} \sim \exp \left[- \left[g_4 a_n^4 + a_n^2 [g_2 + 4g_4(b_{n-1} + b_n)] + 4g_4 a_n (a_{n-1} b_{n-1} + a_{n+1} b_n) \right] \right], \quad (5.5.3)$$

For each $1 \leq n \leq N-1$, the conditional $b_n | \mathbf{a}, \mathbf{b}_{\setminus n}$ has density proportional to

$$b_n | \mathbf{a}, \mathbf{b}_{\setminus n} \sim b_n^{\frac{\beta}{2}(N-i)-1} \exp \left[-2 \left[g_4 b_n^2 + b_n [g_2 + 2g_4(a_n^2 + a_n a_{n+1} + a_{n+1}^2 + b_{n-1} + b_{n+1})] \right] \right]. \quad (5.5.4)$$

In this case, for $g_2, g_4 > 0$, the conditionals given in (5.5.3) and (5.5.4) are unnormalized and log-concave, with easy-to-find modes. Thus, the rejection sampling technique of Devroye (2012) applies, with an expected number of rejections equal to 5. Given an unnormalized and log-concave target density π with mode $m = \operatorname{argmax}_y \pi(y)$, Devroye (2012) constructs a piecewise dominating function h comprising 3 plateaus and 2 exponential tails such that $\int h / \int \pi \leq 5$. The breakpoints $m+2u, m+u, m+v, m+2v$ are located on both sides of the mode, where $u < 0 < v$ satisfy $\pi(m+x) \geq \pi(m)/4 \geq \pi(m+2x)$. Such u and v can be found using a simple bisection method. In practice, we compute $u' < 0 < v'$ solutions of $\pi(m+x) = \pi(m)/4$ and assign $u = u'/2$ and $v = v'/2$, see Figure 5.2.

5.5.2 Example simulations and empirical study of the convergence

In this section, we investigate the convergence of the Gibbs sampler detailed in Section 5.5.1. We sample from β -ensembles with potential $W(x) = \frac{\beta N}{2} V(x)$, for various choices of V of the form (5.5.2). The rescaling in W is applied to capture the weak convergence of the empirical distribution of the particles towards the corresponding equilibrium measure μ_{eq} ; see e.g., Deift (2000, Section 6.1). Intuitively, the rescaling balances the effect of the Vandermonde determinant and that of the potential V in (5.0.1).

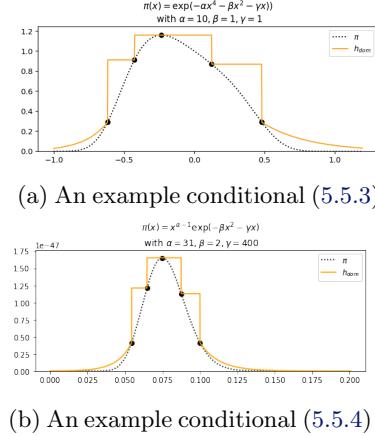


Figure 5.2: Construction of the dominating function h (solid line) of Devroye (2012) to perform rejection sampling with log-concave target π (dashed line), which needs to be log-concave with computable mode. The enveloppe h is made of three plateaus and two exponential tails.

Convergence of the marginals. Let $(x_n^t)_{1 \leq n \leq N}$ be the vector of ordered particles after t full Gibbs passes, that is, after t outer iterations of Algorithm 12. A first quantity to monitor is how well the empirical distribution $\hat{\mu}_N^t = N^{-1} \sum_{n=1}^N \delta_{x_n^t}$ approximates, as t grows, the empirical distribution of the target β -ensemble

$$\hat{\mu}_N = \frac{1}{N} \sum_{n=1}^N \delta_{x_n}, \quad \text{where } \{x_1, \dots, x_N\} \text{ is drawn from (5.0.1).}$$

It turns out that, under assumptions on the potential V that are satisfied by (5.5.2), the random measure $\hat{\mu}_N$ is itself well approximated when $N \gg 1$ by a (deterministic) measure μ_{eq} called the *equilibrium measure* of the potential. This statement can be made rigorous; see for instance the large deviation principle with fast rate $1/N^2$ in Serfaty, 2015, Theorem 2.3.

Two observations are in order. First, the fast rate hints that the approximation should hold even for moderate values of N , as we shall confirm later on in our simulations. Second, μ_{eq} is known analytically for a few choices of polynomial potentials. Thus, for these potentials, we compare draws from $\hat{\mu}_N^t = N^{-1} \sum_{n=1}^N \delta_{x_n^t}$ with μ_{eq} , to assess convergence of our marginals $\hat{\mu}_N^t$ as t grows. This is in line with the experiments of Li and Menon (2013), Olver, Nadakuditi, and Trogdon (2014) and Chafaï and Ferré (2018).

The quartic potential. The equilibrium measure μ_{eq} is available in closed form for potentials proportional to x^{2d} (Deift, 2000, Proposition 6.156). We consider again $V(x) = \frac{1}{4}x^4$, as in Example 5.5.1. In this case all conditionals are log-concave, and can thus be sampled exactly, cf. Section 5.5.1. Figure 5.3 shows the aggregation of the marginal histograms of 1000 independent runs, after each of the first few Gibbs passes.

Observe that convergence to the equilibrium measure is extremely fast: beyond $t = 3$ Gibbs passes, the histograms are visually indistinguishable from the equilibrium measure. This observation is quantitatively monitored in Figure 5.3, where we plot the logarithm of the L_∞ distance between the empirical cdf of $\hat{\mu}_N^t$ and the cdf of μ_{eq} .

Other potentials of degree 4.. We also consider the potential $V(x) = \frac{1}{20}x^4 - \frac{4}{15}x^3 + \frac{1}{5}x^2 + \frac{8}{5}x$ and potentials of the form $V(x) = g_2x^2 + \frac{1}{4}x^4$ where we vary g_2 . Except the case where $g_2 \geq 0$, the conditionals $a_n | \mathbf{a}_{\setminus n}, \mathbf{b}$ are not log concave and we sample from them using a few steps of MALA. This allows us to select various qualitative behaviors of μ_{eq} , which may become dissymmetric (Claeys, Krasovsky, and Its, 2009; Olver, Nadakuditi, and Trogdon, 2014, Example 1.2; Section 3.2), or supported by more than one connected component (Molinari, 2018, Figure 4). Our approach allows to simulate from the corresponding β -ensembles in regimes yet unexplored. Figure 5.4 shows good agreement of marginal histograms of a single sample of $N = 1000$ points with the equilibrium distribution after only $t = 10$ Gibbs passes.

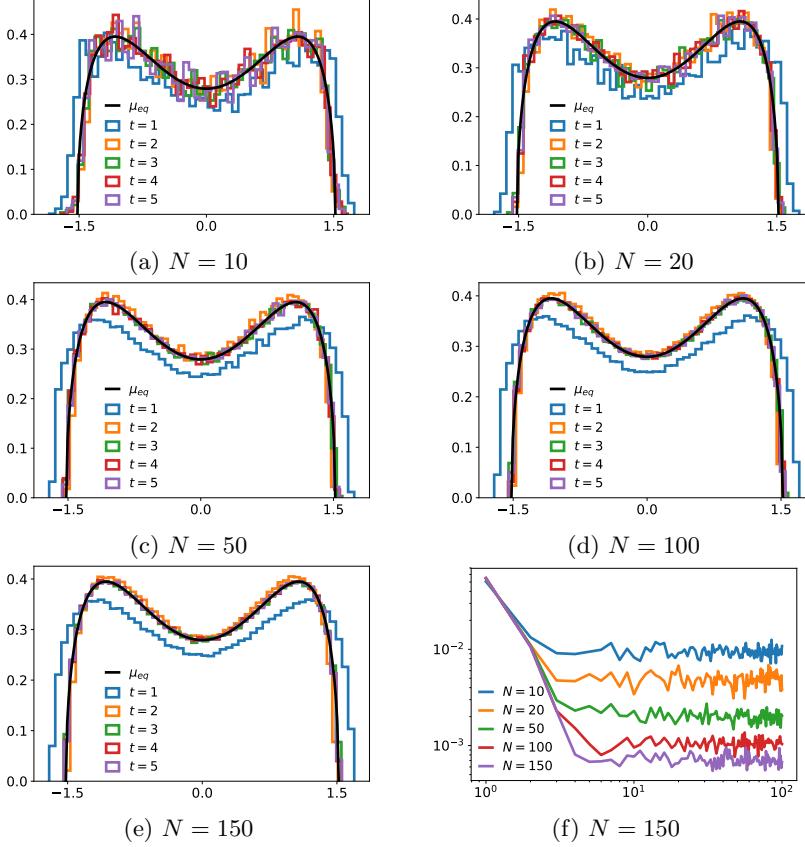


Figure 5.3: For $\beta = 2$ and $V(x) = \frac{1}{4}x^4$, panels (a)-(e) give a visual display of the convergence of the empirical marginal distribution μ_N^t of the eigenvalues constructed from 1000 independent chains. Each colored line corresponds to a Gibbs pass $t \in \{1, 2, 3, \dots\}$, while the equilibrium pdf is shown in black line on each panel. Different panels correspond to increasing values of N . Panel (f) shows the supremum norm of the difference between the cdf of μ_{eq} and the cdf of $\hat{\mu}_N^t$ as a function of the number t of Gibbs passes.

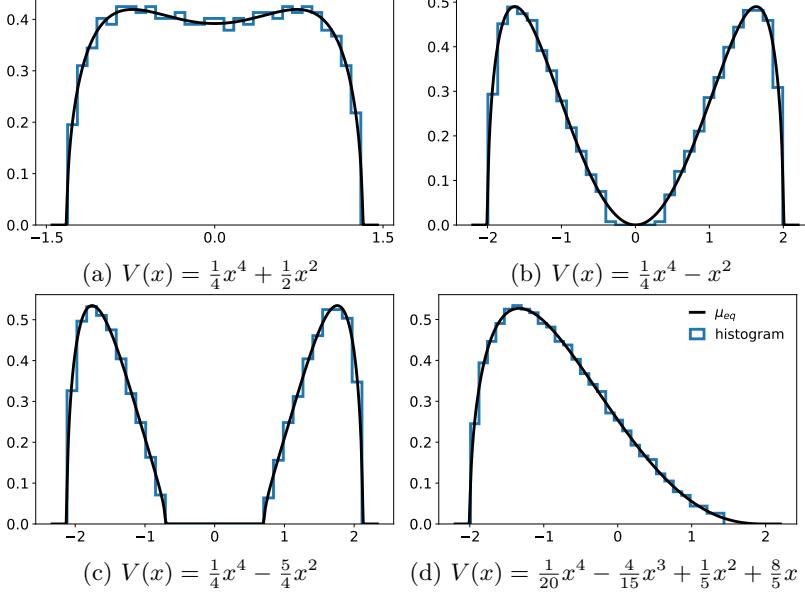


Figure 5.4: For various choices of potentials V of degree 4 and $\beta = 2$, each panel shows the histogram of a sample from μ_N^t , with $N = 1000$ points after $t = 10$ Gibbs passes. The corresponding equilibrium measures are superimposed in black.

The sextic potential. We extend the derivations of Example 5.5.1 and consider sampling from the β -ensemble with potential $V(x) = \frac{1}{6}x^6$. The corresponding equilibrium distribution can be derived from Deift (2000, Proposition 6.156). In this case, the conditionals $a_n | \mathbf{a}_{\setminus n}, \mathbf{b}$ are not log-concave and we cannot use the exact rejection sampler of Devroye (2012). Instead, we switch to a Metropolis-within-Gibbs sampler and make a few steps of MALA; see Section 5.5.1.

One free parameter is the number of MALA steps in one Gibbs pass. We empirically observed (not shown) that this number has an influence on the number of Gibbs passes needed to reach the plateau in

Figure 5.5. Manually setting the number of MALA steps per Gibbs pass to 100 was enough for rapid overall convergence in our experiments, and larger values did not significantly influence the fit in Figure 5.5, which is already striking after less than 10 Gibbs passes.

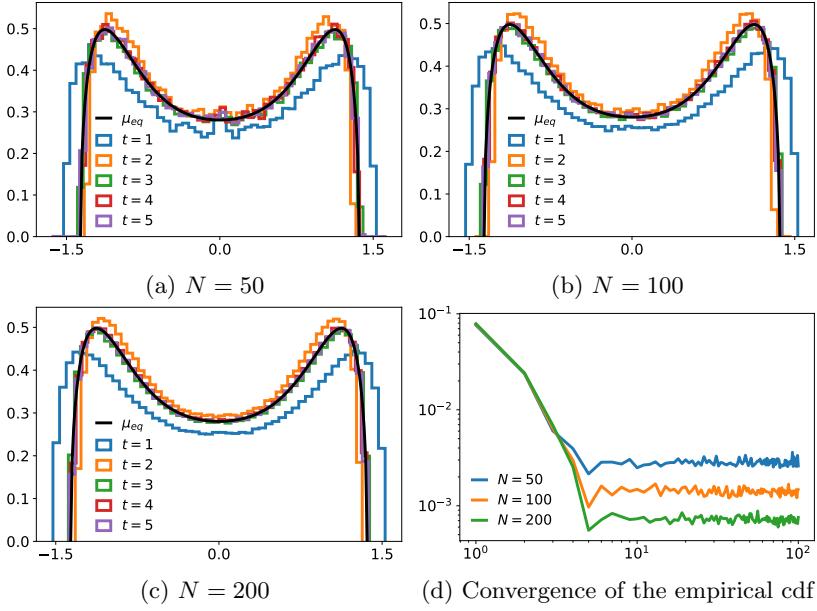


Figure 5.5: For $\beta = 2$ and $V(x) = \frac{1}{6}x^6$, panels (a)-(c) give a visual display of the convergence of the empirical marginal distribution μ_N^t of the eigenvalues constructed from 1000 independent chains. Each colored line corresponds to a Gibbs pass $t \in \{1, 2, 3, \dots\}$, while the equilibrium pdf is shown as a black line on each panel. Different panels correspond to increasing values of N . Panel (d) shows the supremum norm of the difference between the cdf of μ_{eq} and the cdf of $\hat{\mu}_N^t$ as a function of the number t of Gibbs passes.

Fluctuations of the largest eigenvalue. After looking at the global behavior of the eigenvalues, we zoom at the right edge of the support of μ to study the local behavior of our approximate samples. We do this for the quartic and sextic potentials. When $\beta = 2$, the target β -ensemble is determinantal and we can test the adequation of the largest atom of $\hat{\mu}_N^t$ to the universal Tracy-Widom limiting distribution (Deift and Gioev, 2005, Corollary 1.3). We implemented the cumulative distribution function (cdf) of the Tracy-Widom law following Bornemann (2009), and rescale the eigenvalues as Olver and Trogdon (2014, Section 3.2).

For each potential, we run 1000 independent chains and record only the largest eigenvalue of each chain after each Gibbs pass. Figures 5.6 and 5.7 show the histograms of the rescaled largest particles after a few Gibbs passes, respectively for the quartic and sextic potential. More quantitatively, in Figure 5.8 we monitor the convergence to the Tracy-Widom distribution across Gibbs passes, by computing the supremum distance between the empirical cdf of the largest eigenvalue and the cdf of the Tracy-Widom law.

For the quartic potential, we observe that the adequation with the cdf of the Tracy-Widom law gets tighter as N grows, and again only a few passes of the Gibbs sampler are sufficient to reach a plateau.

In contrast, for the sextic ensemble there seems to be an impassable gap, as if the rescaling was not adequate or the Tracy-Widom law was not the proper limiting distribution. This is despite the square root singularity at the right edge of the equilibrium distribution (Deift, 2000, Section 6.1). In particular, a simple Kolmogorov-Smirnov test at level 0.05 would reject the adequation to Tracy-Widom. Part of this effect might be due to the fact that the conditionals are not sampled exactly in the sextic case, though.

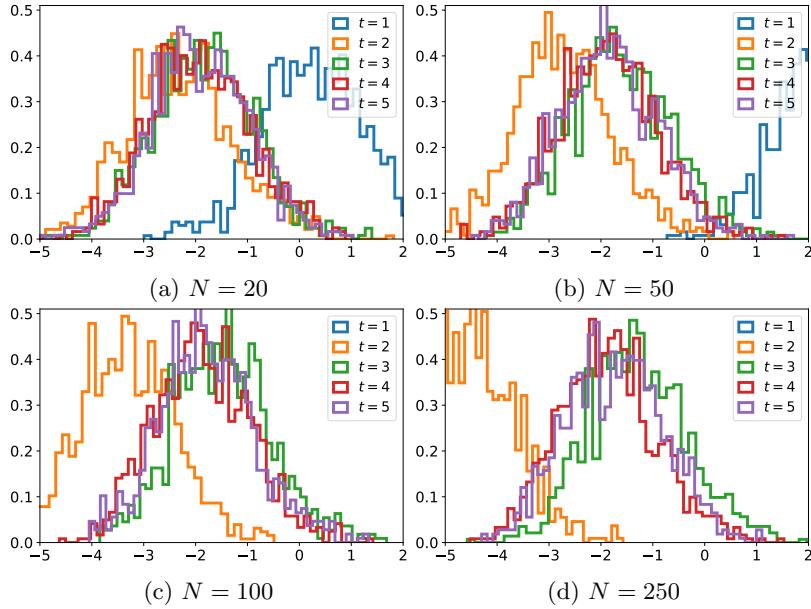


Figure 5.6: For $\beta = 2$ and $V(x) = \frac{1}{4}x^4$, we give a visual display of the convergence of the empirical distribution of the largest atom of μ_N^t constructed from 1000 independent chains to the Tracy-Widom distribution. Each colored line corresponds to a Gibbs pass $t \in \{1, 2, 3, \dots\}$. Different panels correspond to increasing values of N . As N increases the histogram of passes $t = 1$ and $t = 2$ are farther from matching the high density regions of the Tracy-Widom law but then the fit is good and fast; see also Figure 5.8 for a more quantitative monitoring.

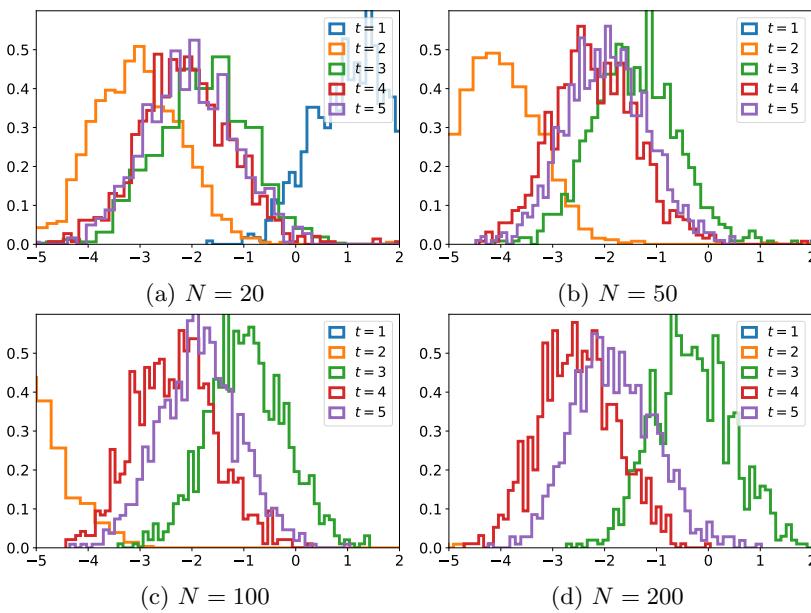


Figure 5.7: For $\beta = 2$ and $V(x) = \frac{1}{6}x^6$, we give a visual display of the convergence of the empirical distribution of the largest atom of μ_N^t constructed from 1000 independent chains to the Tracy-Widom distribution. Each colored line corresponds to a Gibbs pass $t \in \{1, 2, 3, \dots\}$, while the equilibrium pdf is shown as a black line on each panel. Different panels correspond to increasing values of N . As N increases the histogram of passes $t = 1$ and $t = 2$ are farther from matching the high density regions of the Tracy-Widom law and convergence is not as fast as for the quartic case; see also Figure 5.8 for a more quantitative monitoring.

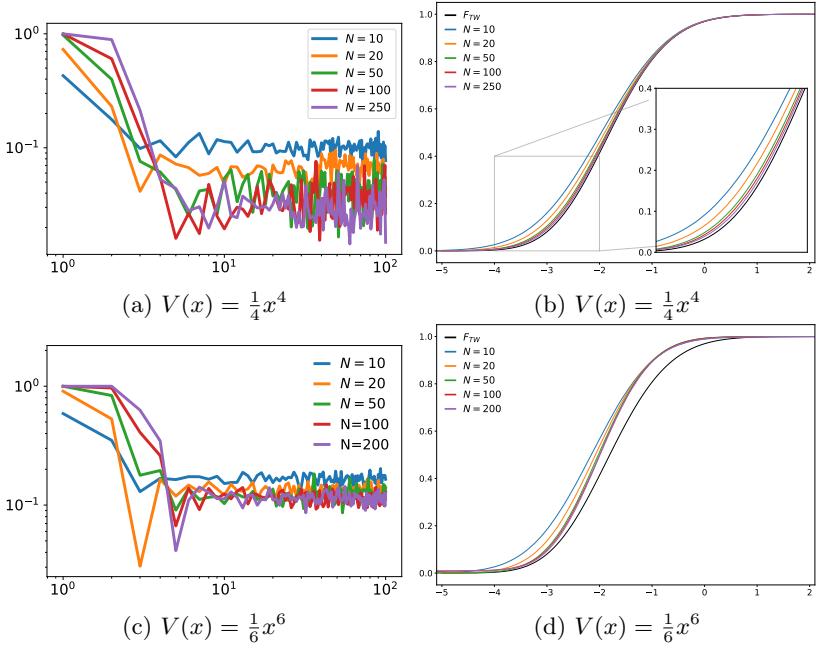


Figure 5.8: For $\beta = 2$ and $V(x) = \frac{1}{4}x^4$ and $\frac{1}{6}x^6$, we monitor the convergence of the empirical cdf of the largest atom of $\hat{\mu}_N^t$ to the expected Tracy-Widom distribution, for several values of the number of points N . Panels (a) and (c) show the supremum norm of the difference between the cdf of the Tracy-Widom distribution and the empirical cdf of the largest atom of $\hat{\mu}_N^t$, constructed from 1000 independent chains, as a function of the number $1 \leq t \leq 100$ of Gibbs passes. Panels (b) and (d) show the corresponding smoothed empirical cdf constructed from the aggregation of the 1000 independent chains over the $t = 100$ passes. Different panels correspond to increasing values of N .

5.6 CONCLUSION

First, we provide the details of an elementary treatment of the three classical tridiagonal models for β -ensembles. Most arguments of the proof already appeared in work by Dumitriu and Edelman (2002), Killip and Nenciu (2004), Forrester and Rains (2006), Gamboa and Rouault (2010), Dette and Nagel (2012), and Krishnapur, Rider, and Virág (2016) and we take no credit for the originality of the proof, only for a stand-alone and elementary version, akin to a survey. We hope that this version will help share the ideas of parametrizing a measure through its recurrence coefficients to computational scientists interested in interacting particle systems. Indeed, throughout the proof, we outline natural reparametrizations of β -ensembles through tridiagonal Jacobi matrices, in which the Vandermonde interaction disappears and leaves only a stream of easy-to-sample, independent matrix entries. Coupled with diagonalization of the underlying tridiagonal matrix, this gives a rejection-free, $\mathcal{O}(N^2)$ exact sampler for the three classical β -ensembles.

Second, when the potential is more generic, independence is lost, but the new interaction can be short-range. We exploited this property to implement a Gibbs kernel and a Metropolis-within-Gibbs variant, which sample β -ensembles with polynomial potentials. This leads to simple MCMC samplers that empirically mix much faster, even for a large number of points, than more sophisticated MCMC kernels working in the original domain of the particles (Li and Menon, 2013; Chafaï and Ferré, 2018). In particular, marginal behavior that matches known theoretical results can be obtained in a few Gibbs passes, totaling a few seconds on a laptop for hundreds of points. However, local behavior, such as the law of the largest particle in the β -ensemble, remains harder to approximate as the degree of the potential grows. Finally, to be fair, we note that the sampler of Chafaï and Ferré (2018) applies much more generally than ours, and in particular to multivariate β -ensembles.

Finally, we want to stress a third related approach, which we leave for future work. As we have seen, diagonalizing a random Jacobi matrix is equivalent to solving a randomized moment problem. One can thus cast sampling β -ensembles as a constrained optimization problem, namely a linear program, as in the work of Ryu and Boyd (2015), but with randomized constraints. Our own interest in tridiagonal models actually came from trying to generalize a sampler for finite determinantal point processes ((Gautier, Bardenet, and Valko, 2017)) of this very form. It is then tempting to look for multivariate versions of the corresponding randomized linear program. We conjecture that the semidefinite relaxations of Lasserre (2010) of multivariate moment problems, with properly randomized constraints, would lead to efficient samplers for multidimensional β -ensembles. This is a technically difficult next step, both in mathematical and computational terms, but it would be useful for Monte Carlo integration (Bardenet and Hardy, 2020; Coeurjolly, Mazoyer, and Amblard, 2020).

Discussion

In this discussion, I am making statements of opinions which can be subjective. In spite of many attempts, it may feel disappointing that, since Hough et al. (2006) derived the first exact DPP sampler, no real paradigm shift occurred. The spectral method seemingly remains the state of the art method for exact simulation of DPPs.

In the discrete case, the current main efforts of the machine learning community focus on making this procedure more scalable, in regimes where the total number of items can be very large but only a very few of them are to be selected. This may suit well the purpose of recommending roughly ten items representing the “diversity” of a large database, but there are some settings where the expected number of items is required to be much larger than a few tens, e.g., for graph signal reconstruction (Tremblay, Amblard, and Barthelme, 2017). Nonetheless, the strong connection between DPP sampling and randomized matrix factorization techniques have been successfully established by Poulson (2019). Besides, I see an interesting avenue of research in the ideas of Launay, Galerne, and Desolneux (2018) and Dereziński (2019) with the development of intermediate thinning strategies. Such thinning procedures can be understood as a way to draw exact DPP samples by first generating realizations of an easy-to-sample distribution followed by a carefully designed downsampling step correcting the initial bias. However, in both cases the downsampling step actually relies on classical routines for sampling DPPs.

Applying the chain rule of Hough et al. (2006) in the continuous case requires a case-by-case basis study. Especially when using rejection sampling for simulating from the conditionals, tailored proposals need to be designed. In first approach, we recommend the choice of the one-point marginal distribution as single proposal to sample each conditional in turn, and focusing on efficient ways to sample from it.

Another interesting line of study that may be worth investigating is perfect simulation. A canonical example where perfect sampling and DPPs meet is the uniform measure on the spanning trees of a graph (Aldous, 1990; Wilson, 1996). In particular, I would start by specializing the work of Decreusefond, Flint, and Low (2013) to sample finite DPP(\mathbf{L}) defined by their likelihood kernel.

WHAT IS NEW IN THIS THESIS? I have tried to derive DPP samplers departing from the original spectral method of Hough et al. (2006). In the finite case, as presented in Section 3.3, I have embedded finite projection DPPs into a continuous convex domain, called zonotope. I am pleased by the elegance of this solution and the representation of DPP samples as the solution of a randomized linear program. However, sam-

pling exactly from the target distribution supported on the zonotope is intractable and MCMC alternatives are particularly costly. The main computational bottleneck remains solving linear programs at each iteration of the procedure. In fact, even answering the question whether a given point belongs to the zonotope involves linear programming. But it is hard to be conclusive without more theoretical investigation of the mixing time of our chain. The answer might be found exploring the links between matroids and projection DPPs (Lyons, 2002), which is a key property.

On the way to find candidate problems to extend the zonotope idea in the continuous case, I have tried to randomize the constraints of semi-infinite linear optimization problems. The ultimate goal was to target DPPs as the support points of the corresponding solution measure and use them to build a random analog of multivariate Gaussian quadrature (Xu, 1994; Ryu and Boyd, 2015). However, I faced important theoretical and technical challenges, while investigating the potential of randomizing moment problems and their semidefinite relaxations (Lasserre, 2010). I have spent some time implementing existing solvers for multivariate moment problems; the main bottlenecks of these techniques are computational, it is difficult to put the theory into practice. The algorithms advocated in the literature and employed to recover the support of the solution measure turned out to be pretty unstable in our setting and do not scale to more than 20 points in dimension two. Then, I decided to focus on the one-dimensional setting where the moment problems to be randomized were clearer (Dette and Studden, 1997). In fact, the stars align in the univariate case and I recovered random tridiagonal models associated to the classical β -ensembles, which I then tried to generalize using a Gibbs sampler having surprisingly fast empirical mixing behavior. Such tridiagonal models are a striking example of an elegant and very practical way of capturing the complex correlations arising in the input space and reducing them to only short-range interactions in the parameter space. In the end, I see a promising avenue in randomizing the constraints of moment problems to sample DPPs, but I have now a good idea of the theoretical and technical challenges and it is a highly non-trivial matter to make this a concrete alternative.

AN IMPORTANT STEP FOR GUIDING THE CHOICE OF THE METHOD TO USE IN PRACTICE, would be to benchmark fully optimized implementations of the different algorithms, exploring the regimes where each excel. Technically, this could be done at least in the finite setting. The difficulty in making fair comparisons of different sampling methods is at least twofold. A first point is the clarity of the description of the sampling method itself and the availability of, at least, a naive open-source implementation of the algorithm. Secondly, the programming skills of the developer making the implementation efficient, and the computational infrastructure that is used are critical. In this respect, let me highlight the work of Poulson (2019), which offers a top quality open-source implementation² of the matrix-factorization-based exact

² gitlab.com/hodge_star/catamari

samplers, but requires strong skills to interact with the code. With DPPy⁹ I have the modest ambition of providing simple and well documented implementations of the various sampling schemes, making the toolbox an accessible entry point to the DPP model, on a broad scale.

⁹github.com/guilgautier/DPPy

HOW TO CHOOSE THE KERNEL? The answer to this question is application-dependent. For instance, when spatial repulsion between the points is required, the kernel is usually assumed to be shift invariant or isotropic, and parametrized by a small number of coefficients characterizing local and global interactions (Biscio and Lavancier, 2016). When computational tractability is the main concern, the kernel is usually assumed to be in a low-rank factored form (Gartrell, Paquet, and Koenigstein, 2016; Dupuy and Bach, 2018). As emphasized along this thesis, I am a strong advocate of projection kernels, in particular, when a control on the number of selected items or points is needed. A first reason is that they usually come with strong theoretical guarantees (Bardenet and Hardy, 2020; Coeurjolly, Mazoyer, and Amblard, 2020; Belhadji, Bardenet, and Chainais, 2018, 2019). Secondly, generating exact samples does not require the eigendecomposition nor any costly preprocessing of the kernel. Finally, I want to point out that projection kernels can also benefit the k -DPP model, often considered as the practical alternative to constrain the sample size. For now, k -DPPs are almost exclusively defined through a generic (positive semi-definite) kernel,³ and suffer the same computational costs - if not more - as generic DPPs. But when they are defined with a projection kernel, they come with very special properties. Indeed, exact sampling simply requires to run a projection DPP sampling routine for k iterations, and if one more point is desired, one extra iteration suffices. Some authors have suggested sampling k -DPPs in this way without checking whether the kernel is a projection, this is at best a heuristic.

ARE DPPS “THE” RIGHT STATISTICAL MODEL? On paper, DPPs are very attractive. To some extent, they can be understood as the kernel machine of points processes. The model is easily described with a single parameter: a kernel function. When defined through a Hermitian or symmetric kernel, the determinantal structure of the correlation functions carries both the notion of repulsion between points and the geometrical interpretability of the model. However, one can wonder whether sampling exactly from an appealing but incorrect model is better than sampling approximately from it. Besides, there are obviously alternative models for repulsive point patterns, which might be less expressive than DPPs but cheaper to sample from, like the Poisson disk sampling strategy. One could also think of running only a few steps of a Markov chain leaving a DPP target invariant, like the ones presented in Chapter 3, as a way to produce “diverse” sets of points.

On the other hand, as a computational tool, DPPs offer strong guarantees. To enjoy this properties, exact samples are required, e.g., to guarantee fast rates in Monte Carlo integration (Bardenet and Hardy,

³In the discrete setting, for $k \geq 1$,

$$\mathbb{P}_{k\text{-DPP}}[\mathcal{X} = S] \propto \det \mathbf{L}_S \mathbb{I}_{|S|=k}.$$

2020; Coeurjolly, Mazoyer, and Amblard, 2020; Gautier, Bardenet, and Valko, 2019c).

More generally, probabilistic models exhibiting properties of negative dependence in their weak or strong forms (Borcea, Brändén, and Liggett, 2009) are gaining more and more interest from both the machine learning and the signal processing communities. In particular, several special events have been recently organized on these topics.^{4,5,6}

As a closing remark, I sometimes heard: “mathematicians are physicists who went astray”. This may suggest to go back to the roots of the model (Macchi, 1975; Ginibre, 1965) and gain some insight from the physics literature, which has proved to be very imaginative and creative in the design of efficient sampling algorithms for interacting particle systems (Liggett, 2005).

⁴Tutorial on Negative Dependence, Stable Polynomials, and All That, NeurIPS, 2018

⁵Workshop on Negative Dependence in Machine Learning, ICML, 2019

⁶Tutorial on Determinantal Point Processes in Signal Processing and Machine Learning, EUSIPCO, 2019

Résumé en français

UN PROCESSUS PONCTUEL DÉFINIT UNE CONFIGURATION DE POINTS ALÉATOIRE. Ces points représentent par exemple des particules en interaction ou les éléments d'un corpus, d'une base de données. Un processus ponctuel déterminantal (DPP) est un type de processus ponctuel dont les points ont tendance à se repousser; où les éléments sélectionnés représentent d'une certaine manière la diversité du corpus.

Au cours de l'élaboration d'un cadre mathématique permettant de modéliser le phénomène optique appelé l'effet d'anti-bunching, effet attendu mais pas encore observable à l'époque, Odile Macchi (1975) a défini rigoureusement les processus fermioniques, renommés plus tard processus ponctuels déterminantaux. Ce modèle caractérise la distribution des temps d'inter-arrivées de fermions émis par une source spécifique : la probabilité de détecter deux fermions dans un intervalle de temps court est plus faible que celle de deux particules qui auraient été émises de manière indépendantes, d'où le nom anti-bunching.

Depuis, les DPP sont devenus un outil pour l'étude des grandes matrices aléatoires (Johansson, 2006) et ont trouvé des applications aussi variées qu'en statistiques spatiales (Lavancier, Møller, and Rubak, 2015), intégration numérique (Bardenet and Hardy, 2020; Coeurjolly, Mazoyer, and Amblard, 2020), traitement du signal (Avena et al., 2018; Tremblay, Amblard, and Barthelme, 2017) et apprentissage artificiel (Kulesza and Taskar, 2012), où ils sont utilisés à la fois pour leur pouvoir de modélisation et comme outil d'échantillonnage intelligent.

Prenons l'exemple d'un processus ponctuel déterminantal \mathcal{X} utilisé pour sous-échantillonner, sous critère de diversité, un ensemble de données (images, musiques, catalogue d'objets, etc.) étiquetées de 1 à M . Ce DPP est paramétré par une matrice symétrique \mathbf{K} , de taille $M \times M$, de sorte que l'entrée $\mathbf{K}_{x,y}$ encode la similarité entre les items portant les étiquettes x et y . Cette matrice, appelée noyau du DPP, caractérise complètement le processus à travers ses probabilités d'inclusion : pour n'importe quel ensemble d'étiquettes $\{x_1, \dots, x_N\}$ on a,

$$\mathbb{P}[\{x_1, \dots, x_N\} \subset \mathcal{X}] = \det \begin{bmatrix} \mathbf{K}_{x_1 x_1} & \cdots & \mathbf{K}_{x_1 x_N} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{x_N x_1} & \cdots & \mathbf{K}_{x_N x_N} \end{bmatrix}.$$

Pour $N = 2$, l'équation précédente donne

$$\mathbb{P}[\{x_1, x_2\} \subset \mathcal{X}] = \mathbf{K}_{x_1 x_1} \mathbf{K}_{x_1 x_1} - \mathbf{K}_{x_1 x_2}^2, \quad (5.6.1)$$

ce qui s'interprète comme suit : la probabilité de trouver les deux items étiquetés x_1 et x_2 dans un même échantillon du DPP est d'autant plus faible que leur similarité $|\mathbf{K}_{x_1 x_2}|$ est grande.

LA STRUCTURE DÉTERMINANTALE ENCODE LA NOTION DIVERSITÉ OU DE DÉPENDANCE NÉGATIVE entre les différents items sélectionnés. Cette forme algébrique particulière associée à un choix judicieux du noyau K , confère aux DPP de nombreux avantages statistiques et computationnels. En un certain sens les DPP sont les machines à noyaux des processus ponctuels.

CETTE THÈSE SE CONCENTRE SUR LA PHASE D'ÉCHANTILLONNAGE DES DPP, c'est à dire sur la conception de méthodes de simulation efficaces pour ce type de processus particulier. Les échantillons d'un DPP sont utilisés par exemple pour i) confirmer expérimentalement la validité de résultats théoriques, notamment en théorie des matrices aléatoires ii) générer des ensembles d'items capturant la diversité d'une base de données dans le cas de systèmes de recommandation, ou d'un corpus de textes afin d'en produire un résumé représentatif, etc. iii) servir à l'estimation de l'intégrale d'une fonction d'intérêt dans une procédure Monte Carlo iv) pour sélectionner les lignes ou les colonnes d'une matrices d'attributs dans des problèmes de régression ou de design d'expériences.

Pour ce faire, il existe des procédures dites de simulation exacte et de simulation approchée. Les configurations de points générées par une méthode exacte possèdent bien les propriétés statistiques prescrites par le modèle déterminantal. Cependant, à l'instar des autres méthodes à noyaux, ces procédures ne passent pas à l'échelle car le coût de simulation dépend le plus souvent de manière polynomiale en la taille du problème: nombre de points, dimension de l'espace ambiant, taille de la base de données etc. D'autre part, il existe des algorithmes de simulation approchée qui cherchent à réduire ces coûts au prix d'une approximation des propriétés statistiques du DPP ciblé. Ceci introduit un compromis entre la qualité d'un échantillon, exact ou approché, et son coût de simulation.

CES TRAVAUX DE THÈSE SE PORTENT PRINCIPALEMENT SUR L'ÉCHANTILLONNAGE DES DPP DITS DE PROJECTION qui peuvent être compris comme les briques élémentaires du modèle ; tout DPP peut s'écrire comme un mélange de DPP de projection. Ces DPP particuliers, associés un noyau de projection orthogonale, permettent notamment un contrôle de la taille des échantillons, donnée par le rang du noyau. Dans le cadre fini, nous apportons un nouvel éclairage sur la simulation des DPP de projection en établissant la correspondance entre le problème d'échantillonnage et la résolution d'un problème d'optimisation linéaire dont les contraintes sont aléatoires. Nous en tirons une méthode d'échantillonnage par chaîne de Markov efficace. Sur la droite réelle, certains DPP classiques peuvent être simulés par le calcul des valeurs propres de matrices tridiagonales aléatoires bien choisies. Nous en fournissons une nouvelle preuve élémentaire et unifiante, dont nous tirons également un échantilleur approché efficace servant à l'étude de processus plus généraux appelés β -ensembles. En dimension supérieure, nous nous concentrerons sur une classe de DPP de

projection utilisée en intégration numérique. Les estimateurs Monte Carlo construits à partir de DPP offrant des garanties théoriques de convergence plus rapide que les estimateurs classiques

DANS LE CADRE D’UNE RECHERCHE REPRODUCTIBLE, NOUS AVONS DÉVELOPPÉ UNE BOÎTE À OUTILS OPEN-SOURCE, NOMMÉE DPPY⁹. Celle-ci rassemble une implémentation des différentes techniques de simulations actuelles et s’accompagne d’une documentation¹⁰ complète et illustrée.

C’est dans ce même esprit que nous nous efforçons de donner les intuitions et des explications claires aux définitions et propriétés des DPP à travers le manuscrit. Les annexes placées en fin de chapitres rassemblent des remarques, résultats et preuves complémentaires au texte principal. Le format du texte est basé sur le style Tufte (2006) dont les larges marges nous permettent distiller des commentaires et d’illustrer le corps du texte avec des figures le plus souvent réalisées avec DPPy.

Le manuscrit se divise en cinq chapitres. Les deux premiers peuvent être compris comme une revue de l’état de l’art alors que les trois suivants présentent essentiellement les grandes contributions de la thèse.

LE CHAPITRE 1 présente le formalisme des processus déterminantaux et introduit les outils mathématiques utilisés dans les chapitres suivants. En annexe, nous avons souhaité présenter des preuves explicites de résultats simplement énoncés ou laissés en exercices dans la littérature. En particulier, nous recensons et démontrons différentes propriétés de stabilité des DPP sous certains conditionnements probabilistes et opérations ensemblistes.

LE CHAPITRE 2 présente les différentes méthodes d’échantillonnage exact connues pour les DPP à espace d’état continu et fini. Une place spéciale est accordée aux DPP de projection, afin de souligner leur rôle particulier dans la construction et la simulation de DPP plus généraux.

Dans le cas fini, la méthode d’échantillonnage classique de Hough et al. (2006) requiert la décomposition spectrale du noyau \mathbf{K} , dont le coût est d’ordre $\mathcal{O}(M^3)$. Il reste ensuite à simuler d’un DPP de projection construit par un tirage aléatoire des vecteurs propres à partir de variables de Bernoulli de paramètre les valeurs propres. Le schéma de simulation exact des DPP de projection s’apparente ensuite à une procédure d’orthogonalisation de Gram-Schmidt randomisée sur des vecteurs d’attributs latents obtenus à partir des vecteurs propres ou simplement des colonnes du noyau de projection sélectionné. Ainsi le coût de simulation d’un DPP de projection est d’ordre $\mathcal{O}(MN^2)$, où N est le rang du noyau de projection et qualifie également le nombre de items contenus dans l’échantillon généré. Cette procédure générique de simulation des DPP offre une belle interprétation géométrique. Cependant le coût initial cubique en M rend la méthode impraticable à large échelle, lorsque le nombre d’items de la base de données est très grand.

G. Gautier, G. Polito, R. Bardenet, and M. Valko. 2019. *DPPy: DPP Sampling with Python*. Journal of Machine Learning Research - Machine Learning Open Source Software (JMLR-MLOSS). arXiv:1809.07258.
github.com/guilgautier/DPPy
dppy.readthedocs.io.

Dans le cas continu, le challenge est d'autant plus grand, car il n'existe pas de procédure numérique permettant d'obtenir la décomposition spectrale du noyau. Et même pour les DPP de projection, la procédure d'échantillonnage à la Gram-Schmidt requiert la capacité de simuler de manière exacte selon des distributions conditionnelles complexes.

Des méthodes exactes alternatives reposent sur la donnée au préalable d'une forme factorisée du noyau, ou bien appliquent une procédure classique sur un ensemble de taille plus petite que M formé d'items soigneusement présélectionnés de manière aléatoire. Ces techniques permettent un échantillonnage moins coûteux dans certains régimes, e.g., base de données de grande taille et échantillons de petite taille. Lorsque la simulation d'échantillons exacts n'est pas cruciale ou trop coûteuse pour l'application visée, on peut se tourner vers des méthodes approchées.

LE CHAPITRE 3 traite des différentes méthodes d'échantillonnage approchées. Nous commençons par un rappel rapide des différentes techniques de simulation approchées basées sur des approximations de faible rang ou des projections aléatoires du noyau, ainsi que sur des chaînes de Markov autorisant des transitions entre états différents d'au plus un item. Le reste du chapitre est consacré à la présentation d'une contribution de la thèse. Nous développons une méthode à base de chaînes de Markov pour simuler de manière approchée des DPP de projection de noyau $\mathbf{K} = \Phi^\top(\Phi\Phi^\top)^{-1}\Phi$, où $\Phi \in \mathbb{R}^{N \times M}$ est une matrice d'attributs. Dans ce cadre, la probabilité d'obtenir un échantillon $X = \{x_1, \dots, x_N\}$ donné est proportionnelle à $(\det \Phi_{:X})^2$, ce qui correspond au carré du volume du parallélotope engendré par les vecteurs d'attributs $\Phi_{:x_1}, \dots, \Phi_{:x_N}$. C'est à dire que chaque échantillon possible, i.e., chaque élément du support du DPP de projection considéré, est représenté par un parallélotope ; plus son volume est grand, plus l'échantillon associé à de chance d'être obtenu. Il s'avère que l'ensemble de ces parallélotopes peut être arrangé de sorte à former un domaine continu et convexe appelé zonotope. Nous nous appuyons sur une preuve de ce résultat faisant intervenir un problème d'optimisation linéaire sous contraintes permettant d'identifier un pavage du zonotope (Dyer and Frieze, 1994). Ainsi, en simulant des points dans le zonotope selon une distribution bien choisie, les échantillons du DPP de projection sous-jacent sont obtenus par résolution d'un problème d'optimisation linéaire sous contraintes randomisées. L'approche développée offre un point de vue nouveau sur l'échantillonnage des DPP de projection qui contraste avec l'aspect Gram-Schmidt de la procédure d'échantillonnage exact classique. Cependant, la génération de points sur le zonotope étant très complexe de manière exacte nous utilisons une chaîne de Markov ce qui donne le caractère approché à notre technique de simulation. Cette nouvelle méthode permet de s'affranchir des contraintes de transitions extrêmement locales inhérentes aux techniques approchées existantes. De plus, elle offre empiriquement une meilleure qualité d'approximation, au prix d'un coût computationnel plus élevé impliquant la résolution de problèmes d'optimisation.

G. Gautier, R. Bardenet, and M. Valko. 2017. *Zonotope hit-and-run for efficient sampling from projection DPPs*. In International Conference on Machine Learning (ICML). arXiv:1705.10498.

LE CHAPITRE 4 présente notre contribution à l'utilisation du caractère répulsif des DPP dans une procédure d'intégration numérique par méthode de Monte Carlo. Ce travail a été motivé par celui de Bardenet and Hardy (2020) ayant prouvé que les échantillons d'un DPP de projection spécifique permettent de construire un estimateur non-biaisé – proche d'une version aléatoire de la quadrature Gauss – ayant une variance qui décroît plus rapidement que la variance obtenue par les méthodes d'intégration Monte Carlo classiques. L'idée originale d'utiliser les DPP pour l'intégration Monte Carlo revient rétrospectivement à Ermakov and Zolotukhin (1960), soit une quinzaine d'années avant qu'Odile Macchi (1975) n'introduise le formalisme des DPP! Dans cette contribution nous révélons le lien intrinsèque entre cet estimateur basé sur la résolution d'un système linéaire randomisé et les DPP de projection. Grâce aux outils modernes des DPP, nous proposons une analyse actualisée des propriétés de cet estimateur. En particulier, afin d'obtenir la meilleure garantie d'approximation à partir d'un échantillon de taille fixée, la définition même de l'estimateur suggère un choix naturel pour le noyau du DPP. Supposons que la fonction f à intégrer se décompose sur une base de fonctions dans laquelle les coefficients associés décroissent rapidement, il est alors suggéré de considérer le noyau de projection construit à partir des fonctions de base sur lesquelles la fonction f a ses plus grands coefficients. Dans un second temps nous proposons une étude comparative des propriétés de convergence empirique des estimateurs respectifs de Bardenet and Hardy et Ermakov and Zolotukhin. Pour ce faire, nous considérons le DPP de projection construit à partir de polynômes orthogonaux multivariés invoqué par Bardenet and Hardy. De plus, nous proposons une version adaptée et efficace du schéma d'échantillonnage classique d'Hough et al. (2006), qui nous a permis de gagner plusieurs ordres de grandeur sur le temps de simulation de ce DPP spécifique par rapport aux échantilleurs existants, et ainsi d'explorer les propriétés des deux estimateurs dans des régimes nouveaux.

LE CHAPITRE 5 présente une contribution relative à l'échantillonnage d'une classe de processus ponctuels répulsifs sur la ligne réelle appelés β -ensembles apparaissant dans l'étude du comportement des grandes matrices aléatoires. Le cas particulier $\beta = 2$ correspond à une classe de DPP de projection construits à partir de polynômes orthogonaux. Cette classe contient notamment le DPP de projection unidimensionnel utilisé au chapitre 4 dans le cadre de l'intégration Monte Carlo avec DPP. Notre motivation principale, vient du fait que les β -ensembles correspondent à la distribution des valeurs propres de matrices tridiagonales aléatoires. Du point de vue de l'échantillonnage, le calcul des valeurs propres d'une matrice tridiagonale aléatoire constitue une méthode exacte pour générer des échantillons d'un processus continu, de manière efficace en coût $\mathcal{O}(N^2)$, où N correspond à la taille de matrice qui dicte le nombre de points.

Dans ce chapitre nous proposons un traitement nouveau et élémentaire des modèles tridiagonaux associés aux processus classiques des Her-

G. Gautier, R. Bardenet, and M. Valko. 2019b. *On two ways to use determinantal point processes for Monte Carlo integration*. In Advances in Neural Information Processing Systems (NeurIPS).

G. Gautier, R. Bardenet, and M. Valko. 2020. *Fast sampling from β -ensembles*. ArXiv e-prints. arXiv:2003.02344.

mite, Laguerre et Jacobi β -ensembles. Dans ces cas spécifiques, les coefficients définissant la matrice tridiagonale impliquée sont indépendants et suivent des lois simples à simuler. En revanche, pour des β -ensembles plus généraux, il n'y a plus d'indépendance entre les coefficients, mais ces derniers interagissent à faible échelle. En un certain sens, les coefficients de la matrice tridiagonale vivent dans un espace de représentation très spécial qui permet de réduire drastiquement la complexité de l'interaction présente entre les points d'un β -ensembles dans l'espace initial.

Nous exploitons cette propriété dans un schéma d'échantillonnage de Gibbs pour simuler β -ensembles associés à des potentiels polynomiaux. Notre étude expérimentale met en évidence une convergence étonnamment rapide de l'échantillonneur de Gibbs. En particulier, après seulement dix passes de Gibbs, même pour de grandes matrices tridiagonales, le comportement marginal empirique des valeurs propres suit bien le comportement prescrit par la théorie.

DANS LA SECTION FINALE, nous discutons des différentes parties du manuscrit et des contributions de la thèse. Nous présentons des pistes d'améliorations et des questions ouvertes concernant l'échantillonnage des processus ponctuels déterminantaux.

Bibliography

- Affandi, R. H. 2014. *Learning, Large Scale Inference, and Temporal Modeling of Determinantal Point Processes*. PhD dissertation, University of Pennsylvania. (see p. 61).
- Affandi, R. H., E. B. Fox, R. P. Adams, and B. Taskar. 2014. *Learning the Parameters of Determinantal Point Process Kernels*. In International Conference on Machine Learning (ICML). (see p. 11).
- Affandi, R. H., A. Kulesza, E. B. Fox, and B. Taskar. 2013. *Nyström Approximation for Large-Scale Determinantal Processes*. In International Conference on Artificial Intelligence and Statistics (AISTATS). (see p. 61).
- Alaoui, A. E., and M. W. Mahoney. 2015. *Fast Randomized Kernel Ridge Regression with Statistical Guarantees*. In Neural Information Processing Systems (NIPS). (see pp. 14, 55).
- Aldous, D. J. 1990. *The Random Walk Construction of Uniform Spanning Trees and Uniform Labelled Trees*. SIAM Journal on Discrete Mathematics. (see pp. 13, 61, 113).
- Anari, N., S. O. Gharan, and A. Rezaei. 2016. *Monte Carlo Markov Chain Algorithms for Sampling Strongly Rayleigh Distributions and Determinantal Point Processes*. In Conference on Learning Theory (COLT). arXiv:1602.05242. (see pp. 61, 62, 68, 72).
- Andersen, H. C., and P. W. Diaconis. 2007. *Hit and Run as a Unifying Device*. Journal de la Société Française de Statistique. (see p. 66).
- Andersen, M., J. Dahl, and L. Vandenberghe. 2008. *CVXOPT: A Python package for convex optimization*. (see p. 71).
- Anderson, G. W., A. Guionnet, and O. Zeitouni. 2009. *An Introduction to Random Matrices*. Cambridge University Press. arXiv:arXiv:1011.1669v3. (see pp. 11, 89, 91).
- Arnold, L., V. M. Gundlach, and L. Demetrius. 1994. *Evolutionary Formalism for Products of Positive Random Matrices*. (see p. 10).
- Avena, L., F. Castell, A. Gaudillière, and C. Mélot. 2018. *Intertwining wavelets or multiresolution analysis on graphs through random forests*. arXiv:1707.04616. (see pp. 10, 117).
- Bach, F. 2017. *No Title*. Journal of Machine Learning Research. arXiv:1502.06800. (see pp. 73, 86).
- Bach, F., S. Lacoste-Julien, and G. Obozinski. 2012. *On the Equivalence between Herding and Conditional Gradient Algorithms*. In International Conference on Machine Learning (ICML). arXiv:1203.4523. (see p. 73).
- Baik, J., P. Deift, and K. Johansson. 1999. *On the distribution of the length of the longest increasing subsequence of random permutations*. Journal of the American Mathematical Society. arXiv:math/9810105. (see p. 10).
- Barabási, A.-L., and R. Albert. 1999. *Emergence of scaling in random networks*. Science. (see p. 69).

- Bardenet, R., and A. Hardy. 2020. *Monte Carlo with Determinantal Point Processes*. Annals of Applied Probability. arXiv:1605.00361. (see pp. 12, 14, 15, 16, 17, 27, 73, 74, 75, 76, 81, 84, 86, 112, 115, 117, 121).
- Bardenet, R., and M. K. Titsias. 2015. *Inference for determinantal point processes without spectral knowledge*. In Advances in Neural Information Processing Systems (NIPS). arXiv:1507.01154. (see p. 11).
- Belabbas, M.-A., and P. J. Wolfe. 2009. *On landmark selection and sampling in high-dimensional data analysis*. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. arXiv:0906.4582. (see p. 61).
- Belhadji, A., R. Bardenet, and P. Chainais. 2018. *A determinantal point process for column subset selection*. ArXiv e-prints. arXiv:1812.09771. (see pp. 12, 27, 115).
- . 2019. *Kernel quadrature with DPPs*. In Advances in Neural Information Processing Systems (NeurIPS). arXiv:1906.07832. (see pp. 12, 27, 115).
- Berlinet, A., and C. Thomas-Agnan. 2004. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science+Business Media. (see p. 23).
- Biscio, C. A. N., and F. Lavancier. 2016. *Quantifying repulsiveness of determinantal point processes*. Bernoulli. (see p. 115).
- Borcea, J., P. Brändén, and T. M. Liggett. 2009. *Negative dependence and the geometry of polynomials*. Journal of the American Mathematical Society. arXiv:0707.2340. (see p. 116).
- Bornemann, F. 2009. *On the numerical evaluation of Fredholm determinants*. Mathematics of Computation. arXiv:0804.2543. (see p. 109).
- Borodin, A. 2015. *Determinantal point processes*. Edited by G. Akemann, J. Baik, and P. Di Francesco. The Oxford Handbook of Random Matrix Theory. arXiv:0911.1153. (see p. 11).
- Borodin, A., A. Okounkov, and G. Olshanski. 2000. *Asymptotics of Plancherel measures for symmetric groups*. Journal of the American Mathematical Society. arXiv:math/9905032. (see pp. 10, 22, 33, 39).
- Borodin, A., and E. M. Rains. 2004. *Eynard-Mehta theorem, Schur process, and their pfaffian analogs*. Journal of Statistical Physics. arXiv:math-ph/0409059. (see p. 11).
- Briol, F.-X., C. J. Oates, M. Girolami, and M. A. Osborne. 2015. *Frank-Wolfe Bayesian Quadrature: Probabilistic Integration with Theoretical Guarantees*. In Advances in Neural Information Processing Systems (NeurIPS). arXiv:1506.02681. (see pp. 73, 74).
- Broder, A. 1989. *Generating random spanning trees*. In Annual Symposium on Foundations of Computer Science (Proceedings). Publ by IEEE. (see p. 13).
- Brunel, V.-E. 2018. *Learning Signed Determinantal Point Processes through the Principal Minor Assignment Problem*. In Advances in Neural Information Processing Systems (NeurIPS). arXiv:1811.00465. (see p. 12).
- Brunel, V.-E., A. Moitra, P. Rigollet, and J. Urschel. 2017a. *Maximum likelihood estimation of determinantal point processes*. arXiv:1701.06501. (see p. 12).
- . 2017b. *Rates of estimation for determinantal point processes*. In Conference on Learning Theory (COLT), edited by S. Kale and O. Shamir. Proceedings of Machine Learning Research. Amsterdam, Netherlands: PMLR. arXiv:1706.00961. (see p. 12).

- Burt, D., C. E. Rasmussen, and M. V. D. Wilk. 2019. *Rates of Convergence for Sparse Variational Gaussian Process Regression*. In International Conference on Machine Learning (ICML). arXiv:1903.03571. (see p. 15).
- Burton, R., and R. Pemantle. 2004. *Local Characteristics, Entropy and Limit Theorems for Spanning Trees and Domino Tilings via Transfer-Impedances*. The Annals of Probability. arXiv:math/0404048. (see p. 10).
- Cartan, H. 1971. *Differential calculus*. Hermann. (see pp. 97, 98).
- Chafaï, D., and G. Ferré. 2018. *Simulating Coulomb and Log-Gases with Hybrid Monte Carlo Algorithms*. Journal of Statistical Physics. arXiv:1806.05985. (see pp. 89, 90, 105, 107, 112).
- Chen, Y., R. Dwivedi, M. Wainwright, and Y. Bin. 2018. *Fast MCMC Sampling Algorithms on Polytopes*. Journal of Machine Learning Research. (see p. 66).
- Chen, Y., M. Welling, and A. Smola. 2010. *Super-Samples from Kernel Herding*. In Conference on Uncertainty in Artificial Intelligence (UAI). AUAI Press. arXiv:1203.3472. (see p. 73).
- Chihara, T. S. 1971. *On the true interval of orthogonality*. The Quarterly Journal of Mathematics. (see p. 101).
- . 1978. *An introduction to orthogonal polynomials*. Gordon / Breach. (see pp. 101, 103).
- Chow, Y., L. Gatteschi, and R. Wong. 1994. *A Bernstein-type inequality for the Jacobi polynomial*. Proceedings of the American Mathematical Society. (see p. 83).
- Claeys, T., I. Krasovsky, and A. Its. 2009. *Higher-order analogues of the Tracy-Widom distribution and the Painlevé II hierarchy*. Communications on Pure and Applied Mathematics. arXiv:0901.2473. (see p. 107).
- Coakley, E. S., and V. Rokhlin. 2013. *A fast divide-and-conquer algorithm for computing the spectra of real symmetric tridiagonal matrices*. Applied and Computational Harmonic Analysis. (see p. 92).
- Cœurjolly, J.-F., A. Mazoyer, and P.-O. Amblard. 2020. *Monte Carlo integration of non-differentiable functions on $[0, 1]^\iota$, $\iota = 1, \dots, d$, using a single determinantal point pattern defined on $[0, 1]^d$* . ArXiv e-prints. arXiv:2003.10323. (see pp. 73, 112, 115, 116, 117).
- Couillet, R., and M. Debbah. 2011. *Random Matrix Methods for Wireless Communications*. Cambridge University Press. (see p. 10).
- Cousins, B., and S. Vempala. 2016. *A practical volume algorithm*. Mathematical Programming Computation. (see p. 66).
- Daley, D. J., and D. Vere-Jones. 2003. *An Introduction to the Theory of Point Processes. Volume I: Elementary Theory and Methods*. 2nd ed. Probability and its Applications. New York, USA: Springer-Verlag New York. (see pp. 19, 20, 21, 36).
- Davis, P. J., and P. Rabinowitz. 1984. *Methods of numerical integration*. Academic Press. (see pp. 73, 74).
- Decreusefond, L., I. Flint, and K. C. Low. 2013. *Perfect Simulation of Determinantal Point Processes*. ArXiv e-prints. arXiv:1311.1027. (see pp. 13, 57, 113).
- Deift, P. 2000. *Orthogonal polynomials and random matrices : a Riemann-Hilbert approach*. American Mathematical Society. (see pp. 106, 107, 108, 109).

- Deift, P., and D. Gioev. 2005. *Universality at the edge of the spectrum for unitary, orthogonal and symplectic ensembles of random matrices*. Communications on Pure and Applied Mathematics. arXiv:math-ph/0507023. (see p. 109).
- Delyon, B., and F. Portier. 2016. *Integral approximation by kernel smoothing*. Bernoulli. arXiv:1409.0733. (see p. 73).
- Dereziński, M. 2019. *Fast determinantal point processes via distortion-free intermediate sampling*. In Conference on Learning Theory (COLT). arXiv:1811.03717. (see pp. 15, 55, 113).
- Dereziński, M., D. Calandriello, and M. Valko. 2019. *Exact sampling of determinantal point processes with sublinear time preprocessing*. In Advances in Neural Information Processing Systems (NeurIPS), edited by H. W. Garnett, H. Larochelle, A. Beygelzimer, F. D'Alché-Buc, E. Fox, and R. Garnett. Vancouver, Canada: Curran Associates, Inc. arXiv:1905.13476. (see pp. 13, 46, 51, 55, 56).
- Dereziński, M., M. K. Warmuth, S. Cruz, and M. Mahoney. 2018. *Reverse Iterative Volume Sampling for Linear Regression*. Journal of Machine Learning Research. (see p. 12).
- Deshpande, A., and L. Rademacher. 2010. *Efficient volume sampling for row/column subset selection*. IEEE 51st Annual Symposium on Foundations of Computer Science. arXiv:1004.4057. (see p. 12).
- Dette, H., and J. Nagel. 2012. *Distributions on unbounded moment spaces and random moment sequences*. Annals of Probability. (see pp. 90, 92, 102, 112).
- Dette, H., and W. J. Studden. 1997. *The theory of canonical moments with applications in statistics, probability, and analysis*. Wiley. (see pp. 14, 95, 102, 103, 104, 114).
- Devroye, L. 2012. *A note on generating random variables with log-concave densities*. Technical report. (see pp. 106, 107, 108).
- Dick, J., and F. Pillichshammer. 2010. *Digital nets and sequences : discrepancy and quasi-Monte Carlo integration*. Cambridge University Press. (see pp. 73, 74).
- Duane, S., A. D. Kennedy, B. J. Pendleton, and D. Roweth. 1987. *Hybrid Monte Carlo*. Physics Letters B. (see p. 90).
- Dumitriu, I., and A. Edelman. 2002. *Matrix models for beta ensembles*. Journal of Mathematical Physics. arXiv:math-ph/0206043. (see pp. 13, 15, 89, 90, 91, 92, 94, 101, 102, 112).
- Dupuy, C., and F. Bach. 2018. *Learning Determinantal Point Processes in Sublinear Time*. In International Conference on Artificial Intelligence and Statistics (AISTATS). Lanzarote, Spain. arXiv:1610.05925. (see pp. 12, 115).
- Dyer, M., and A. Frieze. 1994. *Random walks, totally unimodular matrices, and a randomised dual simplex algorithm*. Mathematical Programming. (see pp. 63, 65, 120).
- Dyson, F. J. 1962. *Statistical theory of the energy levels of complex systems. I.* Journal of Mathematical Physics. (see p. 10).
- Ermakov, S. M., and V. G. Zolotukhin. 1960. *Polynomial Approximations and the Monte-Carlo Method*. Theory of Probability and Its Applications. (see pp. 15, 16, 17, 73, 75, 76, 85, 86, 121).
- Evans, M., and T. Swartz. 2000. *Approximating integrals via Monte Carlo and deterministic methods*. Oxford University Press. (see pp. 73, 74, 76).
- Forrester, P. J. 2010. *Log-gases and random matrices*. Princeton University Press. (see p. 89).

- Forrester, P. J., and E. M. Rains. 2006. *Jacobians and rank 1 perturbations relating to unitary Hessenberg matrices*. International Mathematics Research Notices. arXiv:[math/0505552](#). (see pp. 97, 102, 112).
- Gamboa, F., and A. Rouault. 2010. *Canonical Moments and Random Spectral Measures*. Journal of Theoretical Probability. arXiv:[0801.4400](#). (see pp. 102, 112).
- Gamboa, F., J. Nagel, and A. Rouault. 2016. *Sum rules via large deviations*. Journal of Functional Analysis. arXiv:[1407.1384](#). (see p. 102).
- Gartrell, M., V.-E. Brunel, E. Dohmatob, and S. Krichene. 2019. *Learning Nonsymmetric Determinantal Point Processes*. In Advances in Neural Information Processing Systems (NeurIPS). arXiv:[1905.12962](#). (see p. 12).
- Gartrell, M., U. Paquet, and N. Koenigstein. 2016. *Low-Rank Factorization of Determinantal Point Processes for Recommendation*. In AAAI Conference on Artificial Intelligence. arXiv:[1602.05436](#). (see p. 115).
- Gautier, G., R. Bardenet, and M. Valko. 2017. *Zonotope hit-and-run for efficient sampling from projection DPPs*. In International Conference on Machine Learning (ICML). arXiv:[1705.10498](#). (see pp. 15, 16, 61, 112, 120).
- . 2019a. *Les processus ponctuels déterminantaux en apprentissage automatique*. In French Colloquium on Signal and Image Processing (GRETSI). (see p. 16).
- . 2019b. *On two ways to use determinantal point processes for Monte Carlo integration*. In Advances in Neural Information Processing Systems (NeurIPS). (see pp. 12, 15, 16, 27, 73, 121).
- . 2019c. *On two ways to use determinantal point processes for Monte Carlo integration*. In Workshop on Negative Dependence in Machine Learning, International Conference on Machine Learning (ICML). (see pp. 15, 116).
- . 2020. *Fast sampling from β -ensembles*. ArXiv e-prints. arXiv:[2003.02344](#). (see pp. 15, 17, 89, 121).
- Gautier, G., G. Polito, R. Bardenet, and M. Valko. 2019. *DPPy: DPP Sampling with Python*. Journal of Machine Learning Research - Machine Learning Open Source Software (JMLR-MLOSS). arXiv:[1809.07258](#). (see pp. 15, 41, 91, 119).
- Gautschi, W. 2004. *Orthogonal polynomials : computation and approximation*. Oxford University Press. (see pp. 90, 94).
- . 2009. *How sharp is Bernstein's Inequality for Jacobi polynomials?* Electronic Transactions on Numerical Analysis. (see p. 83).
- Gelman, A., and D. B. Rubin. 1992. *Inference from iterative simulation using multiple sequences*. Statistical Science. (see p. 70).
- Gillenwater, J. 2014. *Approximate inference for determinantal point processes*. PhD dissertation, University of Pennsylvania. (see pp. 13, 46, 61).
- Gillenwater, J., A. Kulesza, E. Fox, and B. Taskar. 2014. *Expectation-Maximization for Learning Determinantal Point Processes*. In Advances in Neural Information Processing Systems (NIPS). (see p. 12).
- Gillenwater, J., A. Kulesza, Z. Mariet, and S. Vassilvitskii. 2019. *A Tree-Based Method for Fast Repeated Sampling of Determinantal Point Processes*. In International Conference on Machine Learning (ICML). (see pp. 12, 13, 46, 47, 55).
- Ginibre, J. 1965. *Statistical ensembles of complex, quaternion, and real matrices*. Journal of Mathematical Physics. (see pp. 10, 116).

- Goberna, M. A., and M. A. López. 2014. *Post-Optimal Analysis in Linear Semi-Infinite Optimization*. Springer, New York, NY. (see p. 14).
- Golub, G. H., and C. F. Van Loan. 2013. *Matrix computations*. Johns Hopkins Univ Press. (see pp. 50, 52, 58, 91, 101).
- Ha, T., and J. Gibson. 1980. *A note on the determinant of a functional confluent vandermonde matrix and controllability*. Linear Algebra and its Applications. (see p. 98).
- Hardy, A. 2017. *Polynomial Ensembles and Recurrence Coefficients*. Constructive Approximation. arXiv:1709.01287. (see pp. 97, 98, 99).
- Hermon, J., and J. Salez. 2019. *Modified log-Sobolev inequalities for strong-Rayleigh measures*. arXiv:1902.02775. (see p. 62).
- Horn, R. a., and C. R. Johnson. 2012. *Matrix Analysis, Second Edition*. Cambridge University Press. (see p. 25).
- Hough, J. B., M. Krishnapur, Y. Peres, and B. Virág. 2006. *Determinantal Processes and Independence*. In Probability Surveys. arXiv:math/0503110. (see pp. 11, 12, 13, 14, 15, 17, 41, 79, 89, 90, 113, 119, 121).
- . 2009. *Zeros of Gaussian analytic functions and determinantal point processes*. American Mathematical Society. (see pp. 10, 24, 29, 30, 31).
- Huber, M. L. 2016. *Perfect simulation*. Chapman & Hall/CRC. (see p. 13).
- Huszár, F., and D. Duvenaud. 2012. *Optimally-Weighted Herding is Bayesian Quadrature*. In Conference on Uncertainty in Artificial Intelligence (UAI). AUAI Press. arXiv:1204.1664. (see p. 73).
- Johansson, K. 2006. *Random matrices and determinantal processes*. Les Houches Summer School Proceedings. arXiv:math-ph/0510038. (see pp. 11, 28, 33, 73, 117).
- Kammoun, M. S. 2018. *Monotonous subsequences and the descent process of invariant random permutations*. Electronic Journal of Probability. arXiv:1805.05253. (see p. 15).
- Kang, B. 2013. *Fast determinantal point process sampling with application to clustering*. In Advances in Neural Information Processing Systems (NIPS). (see p. 61).
- Karen, M. P. and, N. Best, K. Cowles, and Vines. 2006. *CODA: Convergence Diagnosis and Output Analysis for MCMC*. (see p. 70).
- Kendall, W. S., and J. Møller. 2000. *Perfect Simulation Using Dominating Processes on Ordered Spaces, with Application to Locally Stable Point Processes*. (see p. 13).
- Killip, R., and I. Nenciu. 2004. *Matrix models for circular ensembles*. International Mathematics Research Notices. arXiv:math/0410034. (see pp. 13, 15, 82, 89, 90, 91, 92, 102, 112).
- . 2007. *CMV: The unitary analogue of Jacobi matrices*. Communications on Pure and Applied Mathematics. arXiv:math/0508113. (see p. 102).
- Kojima, M., and F. Komaki. 2016. *Determinantal point process priors for Bayesian variable selection in linear regression*. Statistica Sinica. arXiv:1406.2100. (see p. 12).
- König, W. 2004. *Orthogonal polynomial ensembles in probability theory*. Probability Surveys. arXiv:math/0403090. (see pp. 28, 75, 89).
- König, W., N. O'Connell, and S. Roch. 2002. *Non-colliding random walks, tandem queues, and discrete orthogonal polynomial ensembles*. Electronic Journal of Probability. (see p. 28).

- Krishnapur, M., B. Rider, and B. Virág. 2016. *Universality of the Stochastic Airy Operator*. Communications on Pure and Applied Mathematics. arXiv:[arXiv:1306.4832](#). (see pp. 90, 93, 97, 105, 112).
- Kulesza, A. 2012. *Learning with Determinantal Point Processes*. PhD dissertation, University of Pennsylvania. (see p. 26).
- Kulesza, A., and B. Taskar. 2011. *k-DPPs: Fixed-Size Determinantal Point Processes*. In International Conference on Machine Learning (ICML). Bellevue, WA, USA. (see p. 41).
- . 2012. *Determinantal Point Processes for Machine Learning*. Foundations and Trends in Machine Learning. arXiv:[1207.6083](#). (see pp. 11, 12, 13, 22, 33, 36, 41, 43, 54, 61, 71, 117).
- Laloux, L., P. Cizeau, M. Potters, and J.-P. Bouchaud. 2000. *Random matrix theory and financial correlations*. International Journal of Theoretical and Applied Finance. (see p. 10).
- Lasserre, J.-B. 2010. *Moments, positive polynomials and their applications*. Imperial College Press. (see pp. 14, 112, 114).
- Launay, C., B. Galerne, and A. Desolneux. 2018. *Exact Sampling of Determinantal Point Processes without Eigendecomposition*. ArXiv e-prints. arXiv:[1802.08429](#). (see pp. 13, 35, 48, 49, 51, 52, 58, 59, 113).
- Lavancier, F., J. Møller, and E. Rubak. 2015. *Determinantal point process models and statistical inference: Extended version*. Journal of the Royal Statistical Society. Series B: Statistical Methodology. arXiv:[1205.4818](#). (see pp. 10, 11, 12, 25, 57, 117).
- Li, C., S. Jegelka, and S. Sra. 2016a. *Fast Mixing Markov Chains for Strongly Rayleigh Measures, DPPs, and Constrained Sampling*. In Advances in Neural Information Processing Systems (NIPS). Barcelona, Spain. arXiv:[1608.01008](#). (see pp. 61, 62).
- . 2016b. *Fast Sampling for Strongly Rayleigh Measures with Application to Determinantal Point Processes*. ArXiv e-prints. arXiv:[1607.03559](#). (see pp. 61, 62, 68, 72).
- Li, C., S. Sra, and S. Jegelka. 2016. *Gaussian quadrature for matrix inverse forms with applications*. In International Conference on Machine Learning (ICML). arXiv:[1512.01904](#). (see p. 62).
- Li, X. H., and G. Menon. 2013. *Numerical Solution of Dyson Brownian Motion and a Sampling Scheme for Invariant Matrix Ensembles*. Journal of Statistical Physics. arXiv:[1306.1179](#). (see pp. 89, 105, 107, 112).
- Li, Y., F. Baccelli, H. S. Dhillon, and J. G. Andrews. 2015. *Statistical Modeling and Probabilistic Analysis of Cellular Networks with Determinantal Point Processes*. IEEE Transactions on Communications. arXiv:[1412.2087](#). (see p. 10).
- Liggett, T. M. 2005. *Interacting Particle Systems*. Classics in Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg. (see p. 116).
- Liu, Q., and J. D. Lee. 2017. *Black-Box Importance Sampling*. In Internation Conference on Artificial Intelligence and Statistics (AISTATS). arXiv:[1610.05247](#). (see p. 73).
- Loonis, V., and X. Mary. 2015. *Determinantal Sampling Designs*. ArXiv e-prints. arXiv:[1510.06618](#). (see p. 11).
- Lovász, L., and S. Vempala. 2003. *Hit-and-Run is Fast and Fun*. Technical report. Microsoft Research. (see pp. 66, 67, 72).
- Luenberger, D. G., and Y. Ye. 2016. *Linear and Nonlinear Programming*. (see pp. 64, 67).
- Lyons, R. 2002. *Determinantal probability measures*. Publications mathématiques de l’IHÉS. arXiv:[math/0204325](#). (see pp. 11, 63, 114).

- Macchi, O. 1975. *The coincidence approach to stochastic point processes*. Advances in Applied Probability. (see pp. 10, 16, 20, 25, 36, 73, 116, 117, 121).
- Mallat, S., and G. Peyré. 2009. *A wavelet tour of signal processing : the sparse way*. Elsevier/Academic Press. (see p. 77).
- Mariet, Z., and S. Sra. 2015. *Fixed-point algorithms for learning determinantal point processes*. In International Conference on Machine Learning (ICML). Lille, France: PMLR. arXiv:1508.00792. (see p. 11).
- . 2017. *Elementary Symmetric Polynomials for Optimal Experimental Design*. In Advances in Neural Information Processing Systems (NIPS). arXiv:1705.09677. (see p. 12).
- Mazoyer, A., J.-F. Coeurjolly, and P.-O. Amblard. 2019. *Projections of determinantal point processes*. ArXiv e-prints. arXiv:1901.02099. (see pp. 27, 73).
- Mehta, M. L., and M. Gaudin. 1960. *On the density of Eigenvalues of a random matrix*. Nuclear Physics. (see p. 12).
- Mehta, M. L. 1990. *Random Matrices, 2nd Edition*. 2nd ed. (see p. 32).
- . 2004. *Random Matrices, 3rd Edition*. (see p. 32).
- Molinari, L. G. 2018. *Notes on Random Matrices*. Technical report. (see p. 107).
- Møller, J., and R. P. Waagepetersen. 2004. *Statistical inference and simulation for spatial point processes*. Chapman & Hall/CRC. (see pp. 11, 19, 20).
- Neal, R. M. 2003. *Slice sampling*. The Annals of Statistics. (see p. 72).
- . 2011. *MCMC Using Hamiltonian Dynamics*. Chap. 5 in Handbook of Markov Chain Monte Carlo, edited by S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. Chapman & Hall, CRC Press. (see p. 90).
- Oates, C. J., M. Girolami, and N. Chopin. 2017. *Control functionals for Monte Carlo integration*. Journal of the Royal Statistical Society: Series B (Statistical Methodology). arXiv:1410.2392. (see p. 73).
- O'Hagan, A. 1991. *Bayes–Hermite quadrature*. Journal of Statistical Planning and Inference. (see p. 73).
- Oki, E. 2012. *GNU Linear Programming Kit, Version 4.61*. In Linear Programming and Algorithms for Communication Networks - A Practical Guide to Network Design, Control, and Management. (see p. 71).
- Olver, S. 2011. *Computation of equilibrium measures*. Journal of Approximation Theory. (see p. 90).
- Olver, S., R. R. Nadakuditi, and T. Trogdon. 2014. *Sampling unitary invariant ensembles*. Random Matrices: Theory and Applications. arXiv:1404.0071. (see pp. 12, 89, 107).
- . 2015. *Sampling unitary ensembles*. Random Matrices: Theory and Applications. (see p. 57).
- Olver, S., and T. Trogdon. 2014. *Numerical Solution of Riemann-Hilbert Problems: Random Matrix Theory and Orthogonal Polynomials*. Constructive Approximation. arXiv:1210.2199. (see p. 109).
- Pathria, R. K., and P. D. Beale. 2011. *Statistical Mechanics*. Academic Press. (see p. 10).
- Pemantle, R., and Y. Peres. 2014. *Concentration of Lipschitz functionals of determinantal and other strong Rayleigh measures*. Combinatorics Probability and Computing. arXiv:1108.0687. (see p. 56).
- Peres, Y., and B. Virág. 2003. *Zeros of the i.i.d. Gaussian power series: a conformally invariant determinantal process*. Acta Mathematica. arXiv:math/0310297. (see p. 10).

- Poulson, J. 2019. *High-performance sampling of generic Determinantal Point Processes*. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. arXiv:[1905.00165](#). (see pp. 13, 26, 46, 48, 50, 51, 54, 58, 113, 114).
- Propp, J. G., and D. B. Wilson. 1998. *Coupling from the Past: a User's Guide*. Microsurveys in Discrete Probability. (see pp. 13, 61).
- Pukelsheim, F. 2006. *Optimal Design of Experiments*. Society for Industrial / Applied Mathematics. (see p. 12).
- Rasmussen, C. E., and C. K. I. Williams. 2006. *Gaussian processes for machine learning*. MIT Press. (see p. 44).
- Rebeschini, P., and A. Karbasi. 2015. *Fast Mixing for Discrete Point Processes*. In Conference on Learning Theory (COLT). (see p. 61).
- Robert, C. P. 2007. *The Bayesian choice : from decision-theoretic foundations to computational implementation*. Springer. (see p. 73).
- Robert, C. P., and G. Casella. 2004. *Monte Carlo statistical methods*. Springer-Verlag New York. (see pp. 66, 73, 74, 90, 105, 106).
- Rudnick, Z., and P. Sarnak. 1996. *Zeros of principal L-functions and random matrix theory*. Duke Mathematical Journal. (see p. 10).
- Ryu, E. K., and S. P. Boyd. 2015. *Extensions of Gauss Quadrature Via Linear Programming*. Foundations of Computational Mathematics. (see pp. 112, 114).
- Serfaty, S. 2015. *Coulomb Gases and Ginzburg–Landau Vortices*. Zuerich, Switzerland: European Mathematical Society Publishing House. (see p. 107).
- Shirai, T., and Y. Takahashi. 2003. *Random point fields associated with certain Fredholm determinants I: fermion, Poisson and boson point processes*. Journal of Functional Analysis. (see pp. 11, 20).
- Simon, B. 2011. *Szegő's theorem and its descendants*. Princeton University Press. (see pp. 81, 83, 93, 94).
- Smith, R. L. 1984. *Efficient Monte-Carlo procedures for generating points uniformly distributed over bounded regions*. Operations Research. (see p. 66).
- Snoek, J., R. Zemel, and R. P. Adams. 2013. *A determinantal point process latent variable model for inhibition in neural spiking data*. In Advances in Neural Information Processing Systems (NIPS). (see p. 11).
- Soshnikov, A. 2000. *Determinantal random point fields*. Russian Mathematical Surveys. arXiv:[math/0002099](#). (see pp. 11, 25, 36, 79).
- Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media. (see p. 71).
- Stevens, M. 2019. *Equivalent symmetric kernels of determinantal point processes*. arXiv:[1905.08162](#). (see p. 26).
- Stieltjes, T.-J. 1894. *Recherches sur les fractions continues*. Annales de la Faculté des sciences de Toulouse : Mathématiques. (see p. 101).
- Tremblay, N., P.-O. Amblard, and S. Barthelme. 2017. *Graph sampling with determinantal processes*. In European Signal Processing Conference (EUSIPCO). IEEE. arXiv:[1703.01594](#). (see pp. 10, 113, 117).

- Tremblay, N., S. Barthelme, and P.-O. Amblard. 2018. *Optimized Algorithms to Sample Determinantal Point Processes*. ArXiv e-prints. arXiv:1802.08471. (see pp. 13, 46).
- Tufte, E. R. 2006. *Beautiful Evidence*. First. Graphics Press, LLC. (see pp. 16, 119).
- Turčin, V. F. 1971. *On the Computation of Multidimensional Integrals by the Monte-Carlo Method*. Theory of Probability and Its Applications. (see p. 66).
- Urschel, J., V.-E. Brunel, A. Moitra, and P. Rigollet. 2017. *Learning Determinantal Point Processes with Moments and Cycles*. In International Conference on Machine Learning (ICML). arXiv:1703.00539. (see p. 12).
- Wall, H. S. 1940. *Continued Fractions and Totally Monotone Sequences*. Transactions of the American Mathematical Society. (see p. 103).
- Warlop, R. 2018. *Novel learning and exploration-exploitation methods for effective recommender systems*. PhD dissertation, Université de Lille. (see p. 41).
- Wigner, E. P. 1967. *Random Matrices in Physics*. SIAM Review. (see p. 10).
- Wilhelm, M., A. Ramanathan, A. Bonomo, S. Jain, E. H. Chi, and J. Gillenwater. 2018. *Practical diversified recommendations on YouTube with determinantal point processes*. In International Conference on Information and Knowledge Management, Proceedings. New York, New York, USA: Association for Computing Machinery. (see p. 41).
- Wilson, D. B. 1996. *Generating random spanning trees more quickly than the cover time*. (see p. 113).
- Wishart, J. 1928. *The Generalised Product Moment Distribution in Samples from a Normal Multivariate Population*. Biometrika. (see p. 10).
- Xu, Y. 1994. *Common zeros of polynomials in several variables and higher dimensional quadrature*. Longman Scientific & Technical. (see p. 114).

On sampling determinantal point processes

Determinantal point processes (DPPs) generate random configuration of points where the points tend to repel each other. The notion of repulsion is encoded by the sub-determinants of a kernel matrix, in the sense of kernel methods in machine learning. This special algebraic form makes DPPs attractive both in statistical and computational terms. This thesis focuses on sampling from such processes, that is on developing simulation methods for DPPs. Applications include numerical integration, recommender systems or the summarization of a large corpus of data. In the finite setting, we establish the correspondence between sampling from a specific type of DPPs, called projection DPPs, and solving a randomized linear program. In this light, we devise an efficient Markov chain-based sampling method. In the continuous case, some classical DPPs can be sampled by computing the eigenvalues of carefully randomized tridiagonal matrices. We provide an elementary and unifying treatment of such models, from which we derive an approximate sampling method for more general models. In higher dimension, we consider a special class of DPPs used for numerical integration. We implement a tailored version of a known exact sampler, which allows us to compare the properties of Monte Carlo estimators in new regimes. In the context of reproducible research, we develop an open-source Python toolbox, named DPPy, which implements the state-of-the-art sampling methods for DPPs.

Keywords: Determinantal point processes, sampling, simulation, Monte Carlo methods, random matrices

Sur l'échantillonnage des processus ponctuels déterminantaux

Un processus ponctuel déterminantal (DPP) génère des configurations aléatoires de points ayant tendance à se repousser. La notion de répulsion est encodée par les sous-déterminants d'une matrice à noyau, au sens des méthodes à noyau en apprentissage artificiel. Cette forme algébrique particulière confère aux DPP de nombreux avantages statistiques et computationnels. Cette thèse porte sur l'échantillonnage des DPP, c'est à dire sur la conception d'algorithmes de simulation pour ce type de processus. Les motivations pratiques sont l'intégration numérique, les systèmes de recommandation ou encore la génération de résumés de grands corpus de données. Dans le cadre fini, nous établissons la correspondance entre la simulation de DPP spécifiques, dits de projection, et la résolution d'un problème d'optimisation linéaire dont les contraintes sont randomisées. Nous en tirons une méthode efficace d'échantillonnage par chaîne de Markov. Dans le cadre continu, certains DPP classiques peuvent être simulés par le calcul des valeurs propres de matrices tridiagonales aléatoires bien choisies. Nous en fournissons une nouvelle preuve élémentaire et unificatrice, dont nous tirons également un échantillonneur approché pour des modèles plus généraux. En dimension supérieure, nous nous concentrons sur une classe de DPP utilisée en intégration numérique. Nous proposons une implémentation efficace d'un schéma d'échantillonnage exact connu, qui nous permet de comparer les propriétés d'estimateurs Monte Carlo dans de nouveaux régimes. En vue d'une recherche reproductible, nous développons une boîte à outils open-source, nommée DPPy, regroupant les différents outils d'échantillonnage sur les DPP.

Mots-clés : Processus ponctuels déterminantaux, échantillonnage, simulation, méthodes Monte Carlo, matrices aléatoires