Journal of Machine Learning Research xx (2018) xx-xx Submitted 9/18; Published xx/xx

DPPy: Sampling Determinantal Point Processes with Python

Guillaume Gautier[*,†] G.GAUTIER@INRIA.FR
Rémi Bardenet[†] REMI.BARDENET@GMAIL.COM
Michal Valko[*] MICHAL.VALKO@INRIA.FR
[*] *SequeL team, INRIA Lille - Nord Europe, 40, avenue Halley 59650, Villeneuve d'Ascq, France*
[†] *Univ. Lille, CNRS, Centrale Lille, UMR 9189 - CRIStAL, 59651 Villeneuve d'Ascq, France*

Editor:

Abstract

Determinantal point processes (DPPs) are specific probability distributions over clouds of points that are used as models and computational tools across physics, probability, statistics, and more recently machine learning. Sampling from DPPs is a challenge and therefore we present DPPy, a Python toolbox that gathers known exact and approximate sampling algorithms. The project is hosted on GitHub and equipped with an extensive documentation. This documentation takes the form of a short survey of DPPs and relates each mathematical property with DPPy objects.

Keywords: determinantal point processes, sampling schemes

1. Introduction

Determinantal point processes (DPPs) are distributions over configurations of points that encode diversity through a kernel function K. They were introduced by Macchi (1975) as models for beams of fermions, and they have since found applications in fields as diverse as probability (Soshnikov, 2000; König, 2004; Hough et al., 2006), statistical physics (Pathria & Beale, 2011), Monte Carlo methods (Bardenet & Hardy, 2016), spatial statistics (Lavancier et al., 2012), and machine learning (ML, Kulesza & Taskar, 2012).

In ML, DPPs mainly serve to model diverse sets of items, as in recommendation (Kathuria et al., 2016; Gartrell et al., 2016) or text summarization (Dupuy & Bach, 2018). Consequently, MLers use mostly finite DPPs, which are distributions over subsets of a finite *ground set* of cardinality M, parametrized by an $M \times M$ kernel matrix $\mathbf{K}$. Routine inference tasks such as normalization, marginalization, or sampling have complexity $\mathcal{O}(M^3)$ (Kulesza & Taskar, 2012). For large M, $\mathcal{O}(M^3)$ is a bottleneck, as for other kernel methods, see also Tremblay et al. (2018) for a survey on exact sampling. Efficient approximate samplers have thus been developed, ranging from kernel approximation (Affandi et al., 2013) to MCMC samplers (Anari et al., 2016; Li et al., 2016; Gautier et al., 2017).

In terms of software, the R library spatstat (Baddeley & Turner, 2005), a general-purpose toolbox on spatial point processes, includes sampling and learning of continuous DPPs with stationary kernels, as described by Lavancier et al. (2012). On the other hand, we propose DPPy, a turnkey implementation of all known general algorithms to sample *finite* DPPs. We also provide a few algorithms for non-stationary continuous DPPs that are related to random projections and random covariance matrices, which can be of interest for MLers.

https://github.com/guilgautier/DPPy
https://dppy.readthedocs.io

DPPy is hosted on GitHub. We use Travis for continuous integration. Moreover, DPPy is supported by an extensive documentation which provides the essential mathematical background and illustrates some key properties of DPPs through DPPy objects and their methods. DPPy thus also serves as a tutorial. Along the paper, words in magenta point to the documentation.

2. Determinantal point processes

We introduce DPPs and the main sampling algorithm; see Hough et al. (2006) for details.

2.1 Definition

A point process $\mathcal{X}$ on $\mathbb{X}$ is a random subset of points $\{X_1, \dots, X_N\} \subset \mathbb{X}$, where the number of points N is itself random. We further add to the definition that N should be almost surely finite and that all points in a sample are distinct. Given a reference measure μ on $\mathbb{X}$, a point process is usually characterized by its k-correlation function ρ_k for all k, where

$$\mathbb{P}\left[\begin{array}{c}\exists \text{ one point of the process in} \\ \text{each ball } B(x_i, \mathrm{d}x_i), \forall i = 1, \dots, k\end{array}\right] = \rho_k(x_1, \dots, x_k) \prod_{i=1}^{k} \mu(\mathrm{d}x_i),$$

see Møller & Waagepetersen (2004, Section 4). The functions ρ_k describe the interaction among points in $\mathcal{X}$ by quantifying co-occurrence of points at a set of locations.

A point process $\mathcal{X}$ on $(\mathbb{X}, \mu)$ parametrized by a kernel $K : \mathbb{X} \times \mathbb{X} \to \mathbb{C}$ is said to be *determinantal*, denoted as $\mathcal{X} \sim \mathrm{DPP}(K)$, if its k-correlation functions satisfy

$$\rho_k(x_1, \dots, x_k) = \det\left[K(x_i, x_j)\right]_{i,j=1}^{k}, \quad \forall k \geq 1.$$

Most DPPs in ML correspond to the finite case where $\mathbb{X} = \{1, \dots, M\}$ and the reference measure is $\mu = \sum_{i=1}^{M} \delta_i$ (Kulesza & Taskar, 2012). In this context, the kernel function becomes an $M \times M$ matrix $\mathbf{K}$, and the correlation functions refer to inclusion probabilities. DPPs are thus often defined in ML as $\mathcal{X} \sim \mathrm{DPP}(\mathbf{K})$ if

$$\mathbb{P}\left[S \subset \mathcal{X}\right] = \det \mathbf{K}_S, \quad \forall S \subset \mathbb{X}, \tag{1}$$

where $\mathbf{K}_S$ denotes the submatrix of $\mathbf{K}$ formed by the rows and columns indexed by S. If we further assume that the kernel matrix $\mathbf{K}$ is Hermitian, then the existence and unicity of the DPP in Equation 1 is equivalent to the condition that the spectrum of $\mathbf{K}$ lies in $[0, 1]$. The result also holds for general Hermitian kernel functions K with additional assumptions (Soshnikov, 2000, Theorem 3). We note that there are also DPPs with nonsymmetric kernels (Borodin et al., 2010).

The main interest in DPPs in ML is that they can model diversity while being tractable. To see how diversity shows, we simply observe that Equation 1 entails

$$\mathbb{P}\left[\{i, j\} \subset \mathcal{X}\right] = \mathbf{K}_{ii}\mathbf{K}_{jj} - \mathbf{K}_{ij}\mathbf{K}_{ji} = \mathbb{P}\left[\{i\} \subset \mathcal{X}\right] \times \mathbb{P}\left[\{i\} \subset \mathcal{X}\right] - \mathbf{K}_{ij}\mathbf{K}_{ji}.$$

If $\mathbf{K}$ is Hermitian, the quantity $\mathbf{K}_{ij}\mathbf{K}_{ji} = |\mathbf{K}_{ij}|^2$ encodes how less often i and j should co-occur, compared to independent sampling with the same marginals. Most point processes that encode diversity are not tractable, in the sense that we do not have efficient algorithms to sample, marginalize, or compute normalization constants. DPPs are amenable to these tasks (Kulesza & Taskar, 2012).

https://travis-ci.com/guilgautier/DPPy

2.2 Sampling

We assume henceforth that K is Hermitian and satisfies suitable conditions (Soshnikov, 2000, Theorem 3) so that its spectral decomposition is available

$$K(x,y) \triangleq \sum_{i=1}^{\infty} \lambda_i \phi_i(x)\overline{\phi_i(y)}, \quad \text{with} \int \phi_i(x)\overline{\phi_j(x)}\mu(\mathrm{d}x) = \delta_{ij}.$$

In the finite case, the eigendecomposition of a Hermitian kernel $\mathbf{K}$ can always be computed. Hough et al. (2006, Theorem 7) proved that sampling DPP(K) can be done in two steps:

1. draw $B_i \sim \mathcal{B}\mathrm{er}(\lambda_i)$ independently and denote $\{i_1, \dots, i_N\} = \{i : B_i = 1\}$,
2. sample from the DPP with kernel $\widetilde{K}(x,y) = \sum_{n=1}^{N} \phi_{i_n}(x)\overline{\phi_{i_n}(y)}$.

In particular, this shows that all DPPs are mixtures of *projection* DPPs, that is, DPPs parametrized by an orthogonal projection kernel. The second step of the sampling scheme can be performed using Algorithm 18 of Hough et al. (2006), which we now describe.

Hough et al. (2006) first prove that a projection DPP($\widetilde{K}$) generates configurations of $N = \operatorname{Tr}\widetilde{K}$ points almost surely, and therefore, it is enough to sample $(X_1, \dots, X_N)$ with distribution

$$\frac{1}{N!} \det\left[\widetilde{K}(x_m, x_n)\right]_{m,n=1}^{N} \prod_{n=1}^{N} \mathrm{d}\mu(x_n) = \frac{1}{N!} \operatorname{Vol}^2\{\Phi(x_1), \dots, \Phi(x_n)\} \prod_{n=1}^{N} \mathrm{d}\mu(x_n), \tag{2}$$

where $\Phi(x) \triangleq (\phi_{i_1}(x), \dots, \phi_{i_N}(x))$ denotes the *feature vector* associated with $x \in \mathbb{X}$, so that $\widetilde{K}(x,y) = \Phi(y)^\dagger \Phi(x)$. The algorithm boils down to expressing the chain rule for Equation 2 as

$$\underbrace{\frac{1}{N}\|\phi(x_1)\|^2 \mathrm{d}\mu(x_1)}_{p(x_1)} \prod_{n=2}^{N} \underbrace{\frac{1}{N-(n-1)} \left\|\Pi_{H_{n-1}^{\perp}} \phi(x_n)\right\|^2 \mathrm{d}\mu(x_n)}_{p(x_n|x_1,\dots,x_{n-1})}, \tag{3}$$

where $\Pi_{H_{n-1}}$ is the orthogonal projection onto $H_{n-1} \triangleq \operatorname{span}\{\phi(x_1), \dots, \phi(x_{n-1})\}$. The crux of Hough et al. (2006, Algorithm 18) is that since $\widetilde{K}$ is a projection kernel, each term in Equation 3 is a well-normalized conditional density, making Equation 3 a chain rule. It is then enough to sample from each conditional sequentially.

We make few remarks. First, each conditional in Equation 3 can be expressed as a ratio of two determinants and further expanded with Woodbury's formula. In that case, in each conditional, ML practitioners will recognize a posterior variance in Gaussian process regression with kernel $\widetilde{K}$ (Rasmussen & Williams, 2006, Equation 2.26). We insist that this link with Gaussian processes is a consequence of $\widetilde{K}$ being a *projection* kernel. Second, we observe that the chain rule in Equation 3 has a strong Gram-Schmidt flavor since it actually comes from a recursive application of the base×height formula. In the end, DPPs favor configuration of points whose feature vectors $\Phi(x_1), \dots, \Phi(x_N)$ span a large volume, which is another way of understanding repulsiveness.

The previous sampling scheme is exact and generic but requires the eigendecomposition of the underlying kernel. In the finite setting, this corresponds to an initial $\mathcal{O}(M^3)$ cost,

but then the procedure has an average cost of $\mathcal{O}(M\,[\mathrm{Tr}\,\mathbf{K}]^2)$ (Tremblay et al., 2018). In the continuous case, there is an additional cost of the rejection sampling routine required for sampling each conditional in Equation 3. In applications where these costs are a bottleneck, users rely on approximate sampling. Research has focused on two main directions: kernel approximation (Affandi et al., 2013) and MCMC samplers (Anari et al., 2016; Li et al., 2016; Gautier et al., 2017).

To conclude, some specific DPPs admit more efficient exact samplers that do not rely on Equation 3, e.g., uniform spanning trees (Propp & Wilson, 1998) or eigenvalues of random matrices. For instance, a β-ensemble is a set of N points $\{X_1, \dots, X_N\} \subset \mathbb{R}$ with pdf

$$\frac{1}{Z_{N,\beta}} \prod_{m<n} |x_m - x_n|^\beta \prod_{i=1}^{N} \omega(x_i),$$

where $\beta > 0$. For some choices of the weight function ω, the β-ensemble can be sampled as the spectrum of simple tridiagonal random matrices (Dumitriu & Edelman, 2002; Killip & Nenciu, 2004). When $\beta = 2$, β-ensembles are projection DPPs, and therefore examples of continuous DPPs that can be sampled in $\mathcal{O}(N^2)$. Some of these ensembles are of direct interest to MLers. For instance, the *Laguerre* ensemble, where ω is a Gamma pdf, is the distribution of the eigenvalues of the empirical covariance matrix of i.i.d. Gaussian vectors.

3. The DPPy toolbox

DPPy handles objects that fit the natural definition of the different DPPs models.

- The DPPy object corresponding to the finite DPP($\mathbf{K}$) can be instantiated as
  `Finite_DPP(kernel_type="inclusion", projection=False, **{"K":K})`.
  It has two main sampling methods, namely `.sample_exact()` and `.sample_mcmc()`, implementing different variants of the exact sampling scheme and current state-of-the-art MCMC samplers.

- The DPPy object corresponding to β-ensembles can be instantiated as
  `Beta_Ensemble(ensemble_name="laguerre", beta=3.14)`.
  It has one sampling method
  `.sample(sampling_mode="banded", **{"shape":10, "scale":2.0, "size":50})`
  and two methods for display: `.plot()` to plot the last realization and `.hist()` to construct the empirical distribution.

More information can be found in the documentation and the corresponding Jupyter notebooks, which showcase DPPy objects.

4. Conclusion and future work

DPPy can readily serve as research and teaching material. DPPy is also ready for other contributors to add content and enlarge its scope, e.g., with procedures for learning kernels.

Acknowledgments

We would like to thank Guillermo Polito for leading the reproducible research work group without whom this project would have not existed. The research presented was supported by European CHIST-ERA project DELTA, French Ministry of Higher Education and Research, Nord-Pas-de-Calais Regional Council, Inria and Otto-von-Guericke-Universität Magdeburg associated-team north-European project Allocate, and French National Research Agency projects ExTra-Learn (n.ANR-14-CE24-0010-01) and BoB (n.ANR-16-CE23-0003).

References

Affandi, R. H., Kulesza, A., Fox, E. B., and Taskar, B. Nyström Approximation for Large-Scale Determinantal Processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2013.

Anari, N., Gharan, S. O., and Rezaei, A. Monte Carlo Markov Chain Algorithms for Sampling Strongly Rayleigh Distributions and Determinantal Point Processes. In *Conference on Learning Theory (COLT)*, 2016. arXiv:1602.05242.

Baddeley, A. and Turner, R. spatstat: an R package for analyzing spatial point patterns. *Journal Of Statistical Software*, 2005.

Bardenet, R. and Hardy, A. Monte Carlo with Determinantal Point Processes. *ArXiv e-prints*, 2016. arXiv:1605.00361.

Borodin, A., Diaconis, P., and Fulman, J. On adding a list of numbers (and other one-dependent determinantal processes). *Bulletin of the American Mathematical Society*, 47 (4):639–670, 2010. arXiv:0904.3740.

Dumitriu, I. and Edelman, A. Matrix Models for Beta Ensembles. *Journal of Mathematical Physics*, 43(11):5830–5847, 2002. arXiv:math-ph/0206043.

Dupuy, C. and Bach, F. Learning Determinantal Point Processes in Sublinear Time. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, Lanzarote, Spain, 2018. arXiv:1610.05925.

Gartrell, M., Paquet, U., and Koenigstein, N. Low-Rank Factorization of Determinantal Point Processes for Recommendation. In *AAAI Conference on Artificial Intelligence*, 2016. arXiv:1602.05436.

Gautier, G., Bardenet, R., and Valko, M. Zonotope Hit-and-run for Efficient Sampling from Projection DPPs. In *International Conference on Machine Learning (ICML)*, 2017. arXiv:1705.10498.

Hough, J. B., Krishnapur, M., Peres, Y., and Virág, B. Determinantal Processes and Independence. In *Probability Surveys*, volume 3, pp. 206–229. The Institute of Mathematical Statistics and the Bernoulli Society, 2006. arXiv:math/0503110.

Kathuria, T., Deshpande, A., and Kohli, P. Batched Gaussian Process Bandit Optimization via Determinantal Point Processes. In *Neural Information Processing Systems (NIPS)*, 2016. arXiv:1611.04088.

Killip, R. and Nenciu, I. Matrix models for circular ensembles. *International Mathematics Research Notices*, 2004. arXiv:math/0410034.

König, W. Orthogonal polynomial ensembles in probability theory. *Probab. Surveys*, 2: 385–447, 2004. arXiv:math/0403090.

Kulesza, A. and Taskar, B. Determinantal Point Processes for Machine Learning. *Foundations and Trends in Machine Learning*, 5(2-3):123–286, 2012. arXiv:1207.6083.

Lavancier, F., Møller, J., and Rubak, E. Determinantal point process models and statistical inference : Extended version. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 77(4):853–877, 2012. arXiv:1205.4818.

Li, C., Jegelka, S., and Sra, S. Fast Mixing Markov Chains for Strongly Rayleigh Measures, DPPs, and Constrained Sampling. In *Neural Information Processing Systems (NIPS)*, 2016. arXiv:1608.01008.

Macchi, O. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7(01):83–122, 1975.

Møller, J. and Waagepetersen, R. P. Statistical inference and simulation for spatial point processes, volume 23. Chapman & Hall/CRC. 2004. ISBN 1584882654.

Pathria, R. K. and Beale, P. D. Statistical Mechanics. Academic Press. 2011. ISBN 0123821894.

Propp, J. G. and Wilson, D. B. How to Get a Perfectly Random Sample from a Generic Markov Chain and Generate a Random Spanning Tree of a Directed Graph. *Journal of Algorithms*, 27(2):170–217, 1998.

Rasmussen, C. E. and Williams, C. K. I. Gaussian processes for machine learning. MIT Press. 2006. ISBN 026218253X.

Soshnikov, A. Determinantal random point fields. *Russian Mathematical Surveys*, 55(5): 923–975, 2000. arXiv:math/0002099.

Tremblay, N., Barthelme, S., and Amblard, P.-O. Optimized Algorithms to Sample Determinantal Point Processes. *ArXiv e-prints*, 2018. arXiv:1802.08471.