Learning the Parameters of Determinantal Point Process Kernels

Raja Hafiz Affandi RAJARA@WHARTON.UPENN.EDU
Department of Statistics, University of Pennsylvania

Emily B. Fox EBFOX@STAT.WASHINGTON.EDU
Department of Statistics, University of Washington

Ryan P. Adams RPA@SEAS.HARVARD.EDU
Department of Statistics, Harvard Univerity

Ben Taskar TASKAR@CS.WASHINGTON.EDU
Department of Computer Science & Engineering, University of Washington

Abstract

Determinantal point processes (DPPs) are well-suited for modeling repulsion and have proven useful in applications where diversity is desired. While DPPs have many appealing properties, learning the parameters of a DPP is difficult, as the likelihood is non-convex and is infeasible to compute in many scenarios. Here we propose Bayesian methods for learning the DPP kernel parameters. These methods are applicable in large-scale discrete and continuous DPP settings, even when the likelihood can only be bounded. We demonstrate the utility of our DPP learning methods in studying the progression of diabetic neuropathy based on the spatial distribution of nerve fibers, and in studying human perception of diversity in images.

1. Introduction

A determinantal point process (DPP) is a distribution over configurations of points. The defining characteristic of the DPP is that it is repulsive, which makes it useful for modeling diversity. Recently, DPPs have played an increasingly important role in machine learning and statistics with applications both in the discrete setting—where they are used as a diverse subset selection method (Affandi et al., 2012; 2013b; Gillenwater et al., 2012; Kulesza & Taskar, 2010; 2011a; Snoek et al., 2013)— and in the continuous setting for generating point configurations that tend to be spread out(Affandi et al., 2013a; Zou & Adams, 2012).

Formally, given a space $\Omega \subseteq \mathbb{R}^d$, a specific point configuration $A \subseteq \Omega$, and a positive semi-definite kernel function $L : \Omega \times \Omega \to \mathbb{R}$, the probability density under a DPP with kernel L is given by

$$\mathcal{P}_L(A) \propto \det(L_A) , \quad (1)$$

where L_A is the $|A| \times |A|$ matrix with entries $L(\mathbf{x}, \mathbf{y})$ for each $\mathbf{x}, \mathbf{y} \in A$. This defines a repulsive point process since point configurations that are more spread out according to the metric defined by the kernel L have higher densities.

Building on work of Kulesza & Taskar (2010), it is intuitive to decompose the kernel L as

$$L(\mathbf{x}, \mathbf{y}) = q(\mathbf{x})k(\mathbf{x}, \mathbf{y})q(\mathbf{y}) , \quad (2)$$

where $q(\mathbf{x})$ can be interpreted as the quality function at point $\mathbf{x}$ and $k(\mathbf{x}, \mathbf{y})$ as the similarity kernel between points $\mathbf{x}$ and $\mathbf{y}$. The ability to bias the quality in certain locations while still maintaining diversity via the similarity kernel offers great modeling flexibility.

One of the remarkable aspects of DPPs is that they offer efficient algorithms for inference, including computing marginal and conditional probabilities (Kulesza & Taskar, 2012), sampling (Affandi et al., 2013a;b; Hough et al., 2006; Kulesza & Taskar, 2010), and restricting to fixed-sized point configurations (k-DPPs) (Kulesza & Taskar, 2011a). However, an important component of DPP modeling, learning the DPP kernel parameters, is still considered a difficult, open problem. Even in the discrete Ω setting, DPP kernel learning has been conjectured to be NP-hard (Kulesza & Taskar, 2012). Intuitively, the issue arises from the fact that in seeking to maximize the log-likelihood of Eq. (1), the numerator yields a concave log-determinant term and the normalizer a convex term, leading to a non-convex objective.

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

This non-convexity holds even under various simplifying assumptions on the form of L. Furthermore, when Ω is either a large, discrete set or a continuous subspace, computation of the likelihood is inefficient or infeasible, respectively. This precludes the use of gradient-based and black-box optimization methods.

Attempts to partially learn the kernel have been studied by, for example, learning the parametric form of the quality function $q(\mathbf{x})$ for fixed similarity $k(\mathbf{x}, \mathbf{y})$ (Kulesza & Taskar, 2011b), or learning a weighting on a fixed set of kernel experts (Kulesza & Taskar, 2011a). So far, the only attempt to learn the parameters of the similarity kernel $k(\mathbf{x}, \mathbf{y})$ has used Nelder-Mead optimization (Lavancier et al., 2012), which lacks theoretical guarantees about convergence to a stationary point. Moreover, the use of Nelder-Mead (and other black-box optimization methods) relies heavily on exact computation of the likelihood.

In this paper, we consider parametric forms for the quality function $q(\mathbf{x})$ and similarity kernel $k(\mathbf{x}, \mathbf{y})$ and propose Bayesian methods to learn the DPP kernel parameters Θ using Markov chain Monte Carlo (MCMC). In addition to capturing posterior uncertainty rather than a single point estimate, our proposed methods apply without approximation to large-scale discrete and continuous DPPs when the likelihood can only be bounded (with any desired precision).

In Sec. 2, we review DPPs and their fixed-sized counterpart (k-DPPs). We then explore maximum likelihood estimation (MLE) algorithms for learning DPP and k-DPP kernels. After examining the shortcomings of the MLE approach, we propose a set of techniques for Bayesian posterior inference of the kernel parameters in Sec. 3. In Sec. 4, we derive a set of DPP moments that can be used for model assessment, MCMC convergence diagnostics, and in low-dimensional settings for learning kernel parameters via numerical techniques. Finally, in Sec. 5 we use DPP learning to study the progression of diabetic neuropathy based on the spatial distribution of nerve fibers and also to study human perception of diversity of images.

2. Background

2.1. Discrete DPPs/k-DPPs

For a discrete base set $\Omega = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, a DPP defined by an $N \times N$ positive semi-definite kernel matrix L is a probability measure on the 2^N possible subsets A of Ω:

$$\mathcal{P}_L(A) = \frac{\det(L_A)}{\det(L+I)} \; . \tag{3}$$

Here, $L_A \equiv [L_{ij}]_{\mathbf{x}_i, \mathbf{x}_j \in A}$ is the submatrix of L indexed by the elements in A and I is the $N \times N$ identity matrix (Borodin & Rains, 2005).

In many applications, we are instead interested in the probability distribution which gives positive mass only to subsets of a fixed size, k. In these cases, we consider fixed-sized DPPs (or k-DPPs) with probability distribution on sets A of cardinality k given by

$$\mathcal{P}_L^k(A) = \frac{\det(L_A)}{e_k(\lambda_1, \dots, \lambda_N)} \; , \tag{4}$$

where $\lambda_1, \dots, \lambda_N$ are the eigenvalues of L and $e_k(\lambda_1, \dots, \lambda_N)$ is the kth elementary symmetric polynomial (Kulesza & Taskar, 2011a). Note that $e_k(\lambda_1, \dots, \lambda_N)$ can be efficiently computed using recursion (Kulesza & Taskar, 2012).

2.2. Continuous DPPs/k-DPPs

Consider now the case where $\Omega \subseteq \mathbb{R}^d$ is a continuous space. DPPs extend to this case naturally, with L now a kernel operator instead of a matrix. Again appealing to Eq. (1), the DPP probability density for point configurations $A \subset \Omega$ is given by

$$\mathcal{P}_L(A) = \frac{\det(L_A)}{\prod_{n=1}^{\infty}(\lambda_n + 1)} \; , \tag{5}$$

where $\lambda_1, \lambda_2, \dots$ are eigenvalues of the operator L.

The k-DPP also extends to the continuous case with

$$\mathcal{P}_L^k(A) = \frac{\det(L_A)}{e_k(\lambda_{1:\infty})} \; , \tag{6}$$

where $\lambda_{1:\infty} = (\lambda_1, \lambda_2, \dots)$.

In contrast to the discrete case, the eigenvalues λ_i for continuous DPP kernels are generally unknown; exceptions include a few kernels such as the Gaussian.

3. Learning Parametric DPPs

Assume that we are given a training set consisting of samples $A^1, A^2, \dots, A^T$, and that we model these data using a DPP/k-DPP with parametric kernel

$$L(\mathbf{x}, \mathbf{y}; \Theta) = q(\mathbf{x}; \Theta) k(\mathbf{x}, \mathbf{y}; \Theta) q(\mathbf{y}; \Theta) \; , \tag{7}$$

with parameters Θ. We denote the associated kernel matrix for a set A^t by $L_{A^t}(\Theta)$ and the full kernel matrix/operator by $L(\Theta)$. Likewise, we denote the kernel eigenvalues by $\lambda_i(\Theta)$. In this section, we explore various methods for DPP/k-DPP learning.

3.1. Learning using Optimization Methods

To learn the parameters Θ of a discrete DPP model, recalling Eq. (3) we can maximize the log-likelihood

$$\mathcal{L}(\Theta) = \sum_{t=1}^{T} \log\det(L_{A^t}(\Theta)) - T \log\det(L(\Theta) + I) \; .$$

Lavancier et al. (2012) suggest using the Nelder-Mead simplex algorithm (Nelder & Mead, 1965). This method evaluates the objective function at the vertices of a simplex, then iteratively shrinks the simplex towards an optimal point. Although straightforward, this procedure does not necessarily converge to a stationary point (McKinnon, 1998). Gradient ascent and stochastic gradient ascent are attractive due to their theoretical guarantees, but require knowledge of the gradient of $\mathcal{L}(\Theta)$. In the discrete DPP setting, this gradient can be computed straightforwardly, and we provide examples for discrete Gaussian and polynomial kernels in the Supplement.

We note, however, that both of these methods are susceptible to convergence to local optima due to the non-convex likelihood landscape. Furthermore, these methods (and many other black-box optimization techniques) require that the likelihood is known exactly. From the determinant in the denominator of Eq. (3), we see that when the number of base items N is large, computing the likelihood or its derivative is inefficient. A similar inefficiency arises when we expect large sets A^t, as determined by Θ. Both of these challenges limit the general applicability of these MLE approaches. Instead, in Sec. 3.3, we develop a Bayesian method that only requires an upper and lower bound on the likelihood. We focus on the large N challenge and discuss in the Supplement how analogous methods can be used for handling large observation sets, A^t.

The log-likelihood of the k-DPP kernel parameter is

$$\mathcal{L}(\Theta) = \sum_{t=1}^{T} \log\det(L_{A^t}(\Theta)) - T\log\sum_{|B|=k}\det(L_B(\Theta))\ ,$$

which presents an addition complication due to needing a sum over $\binom{n}{k}$ terms in the gradient.

For continuous DPPs/k-DPPs, once again, both MLE optimization-based methods require that the likelihood is computable. Recalling Eq. (5), we note the infinite product in the denominator. As such, for kernel operators with infinite rank (such as the Gaussian), we are forced to consider approximate MLE methods based on an explicit truncation of the eigenvalues. Gradient ascent using such truncations further relies on having a known eigendecomposition with a differentiable form for the eigenvalues. Unfortunately, such approximate gradients are not unbiased estimates of the true gradient, so the theory associated with stochastic gradient based approaches does not hold.

3.2. Bayesian Learning for Discrete DPPs

Instead of optimizing the likelihood to get an MLE, we propose a Bayesian approach to estimating the posterior distribution over kernel parameters:

$$\mathcal{P}(\Theta|A^1,\ldots,A^T) \propto \mathcal{P}(\Theta)\prod_{t=1}^{T}\frac{\det(L_{A^t}(\Theta))}{\det(L(\Theta)+I)} \tag{8}$$

for the DPP and, for the k-DPP,

$$\mathcal{P}(\Theta|A^1,\ldots,A^T) \propto \mathcal{P}(\Theta)\prod_{t=1}^{T}\frac{\det(L_{A^t}(\Theta))}{e_k(\lambda_1(\Theta),\ldots,\lambda_N(\Theta))}. \tag{9}$$

Here, $\mathcal{P}(\Theta)$ is the prior on Θ. Since neither Eq. (8) nor Eq. (9) yield a closed-form posterior, we resort to approximate techniques based on Markov chain Monte Carlo (MCMC). We highlight two techniques: random-walk Metropolis-Hastings (MH) and slice sampling. We note, however, that other MCMC methods can be employed without loss of generality, and may be more efficient in some scenarios.

In random-walk MH, we use a proposal distribution $f(\hat{\Theta}|\Theta_i)$ to generate a candidate value $\hat{\Theta}$ given the current parameters Θ_i, which are then accepted or rejected with probability $\min\{r,1\}$ where

$$r = \left(\frac{\mathcal{P}(\hat{\Theta}|A^1,\ldots,A^T)}{\mathcal{P}(\Theta_i|A^1,\ldots,A^T)}\frac{f(\Theta_i|\hat{\Theta})}{f(\hat{\Theta}|\Theta_i)}\right)\ . \tag{10}$$

The proposal distribution $f(\hat{\Theta}|\Theta_i)$ is chosen to have mean Θ_i. The hyperparameters of $f(\hat{\Theta}|\Theta_i)$ tune the width of the distribution, determining the average step size. See Alg. 1 of the Supplement.

While random-walk MH can provide a straightforward means of sampling from the posterior, its efficiency requires tuning the proposal distribution. Choosing an aggressive proposal can result in a high rejection rate, while choosing a conservative proposal can result in inefficient exploration of the parameter space. To avoid the need to tune the proposal distribution, we can instead use slice sampling (Neal, 2003). We first describe this method in the univariate case, following the "linear stepping-out" approach described in Neal (2003). Given the current parameter Θ_i, we first sample $y \sim \text{Uniform}[0, \mathcal{P}(\Theta_i|A^1,\ldots,A^T)]$. This defines our *slice* with all values of Θ with $\mathcal{P}(\Theta|A^1,\ldots,A^T)$ greater than y included in the slice. We then define a random interval around Θ_i with width w that is linearly expanded until neither endpoint is in the slice. We propose $\hat{\Theta}$ uniformly in the interval. If $\hat{\Theta}$ is in the slice, it is accepted. Otherwise, $\hat{\Theta}$ becomes the new boundary of the interval, shrinking it so as to still include the current state of the Markov chain. This procedure is repeated until a proposed $\hat{\Theta}$ is accepted. See Alg. 2 of the Supplement.

There are many ways to extend this algorithm to a multidimensional setting. We consider the simplest extension proposed by Neal (2003) where we use hyperrectangles instead

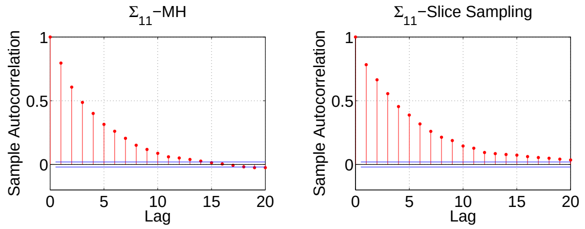

Figure 1. Sample autocorrelation function for posterior samples of the slowest mixing kernel parameter in Eq. (11) and Eq. (12), sampled using MH and slice sampling.

of intervals. A hyperrectangle region is constructed around Θ_i and the edge in each dimension is expanded or shrunk depending on whether its endpoints lie inside or outside the slice. One could alternatively consider coordinate-wise or random-direction approaches to multidimensional slice sampling.

As an illustrative example, we consider synthetic data generated from a two-dimensional discrete DPP with

$$q(\mathbf{x}_i) = \exp\left\{-\frac{1}{2}\mathbf{x}_i^\top \Gamma^{-1}\mathbf{x}_i\right\} \tag{11}$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^\top \Sigma^{-1}(\mathbf{x}_i - \mathbf{x}_j)\right\}, \tag{12}$$

where $\Gamma = \mathrm{diag}(0.5, 0.5)$ and $\Sigma = \mathrm{diag}(0.1, 0.2)$. We consider Ω to be a grid of 100 points evenly spaced in a 10×10 unit square and simulate 100 samples from this DPP. We then condition on these simulated data and perform posterior inference of the kernel parameters using MCMC. Fig. 1 shows the sample autocorrelation function of the slowest mixing parameter, Σ_{11}, learned using random-walk MH and slice sampling. Furthermore, we ran a Gelman-Rubin test (Gelman & Rubin, 1992) on five chains starting from overdispersed starting positions and found that the average partial scale reduction function across the four parameters to be 1.016 for MH and 1.023 for slice sampling, indicating fast mixing of the posterior samples.

3.3. Bayesian Learning for Large-Scale Discrete and Continuous DPPs

When the number of items, N, for discrete Ω is large or when Ω is continuous, evaluating the normalizers $\det(L(\Theta) + I)$ or $\prod_{n=1}^{\infty}(\lambda_n(\Theta) + 1)$, respectively, can be inefficient or infeasible. Even in cases where an explicit form of the truncated eigenvalues can be computed, this will only lead to approximate MLE solutions, as discussed in Sec. 3.1.

On the surface, it seems that most MCMC algorithms will suffer from the same problem since they require knowledge of the likelihood as well. However, we argue that for most of these algorithms, an upper and lower bound of the posterior probability is sufficient as long as we can control the accuracy of these bounds. We denote the upper and lower bounds by $\mathcal{P}^+(\Theta|A^1, \ldots, A^T)$ and $\mathcal{P}^-(\Theta|A^1, \ldots, A^T)$, respectively. In the random-walk MH algorithm we can then compute the upper and lower bounds on the acceptance ratio,

$$r^+ = \left(\frac{\mathcal{P}^+(\hat{\Theta}|A^1, \ldots, A^T)}{\mathcal{P}^-(\Theta_i|A^1, \ldots, A^T)}\frac{f(\Theta_i|\hat{\Theta})}{f(\hat{\Theta}|\Theta_i)}\right) \tag{13}$$

$$r^- = \left(\frac{\mathcal{P}^-(\hat{\Theta}|A^1, \ldots, A^T)}{\mathcal{P}^+(\Theta_i|A^1, \ldots, A^T)}\frac{f(\Theta_i|\hat{\Theta})}{f(\hat{\Theta}|\Theta_i)}\right). \tag{14}$$

The threshold $u \sim \mathrm{Uniform}[0, 1]$ can be precomputed, so we can often accept or reject the proposal $\hat{\Theta}$ even if these bounds have not completely converged. All that is necessary is for $u < \min\{1, r^-\}$ (immediately reject) or $u > \min\{1, r^+\}$ (immediately accept). In the case that $u \in (r^-, r^+)$, we can perform further computations to increase the accuracy of our bounds until a decision can be made. As we only sample u once in the beginning, this iterative procedure yields a Markov chain with the exact target posterior as its stationary distribution; all we have done is "short-circuit" the computation once we have bounded the acceptance ratio r away from u. We show this procedure in Alg. 3 of the Supplement.

The same idea applies to slice sampling. In the first step of generating a slice, instead of sampling $y \sim \mathrm{Uniform}[0, \mathcal{P}(\Theta_i|A^1, \ldots, A^T)]$, we use a rejection sampling scheme to first propose a candidate slice

$$\hat{y} \sim \mathrm{Uniform}[0, \mathcal{P}^+(\Theta_i|A^1, \ldots, A^T)]\,. \tag{15}$$

We then decide whether $\hat{y} < \mathcal{P}^-(\Theta_i|A^1, \ldots, A^T)$, in which case we know $\hat{y} < \mathcal{P}(\Theta_i|A^1, \ldots, A^T)$ and we accept $\hat{y}$ as the slice and set $y = \hat{y}$. In the case where $\hat{y} \in (\mathcal{P}^-(\Theta_i|A^1, \ldots, A^T), \mathcal{P}^+(\Theta_i|A^1, \ldots, A^T))$, we keep increasing the tightness of our bounds until a decision can be made. If at any point $\hat{y}$ exceeds the newly computed $\mathcal{P}^+(\Theta_i|A^1, \ldots, A^T)$, we know that $\hat{y} > \mathcal{P}(\Theta_i|A^1, \ldots, A^T)$ so we reject the proposal. In this case, we generate a new $\hat{y}$ and repeat.

Upon accepting a slice y, the subsequent steps for proposing a parameter $\hat{\Theta}$ proceed in a similarly modified manner. For the interval computation, the endpoints Θ_e are each examined to decide whether $y < \mathcal{P}^-(\Theta_e|A^1, \ldots, A^T)$ (endpoint is not in slice) or $y > \mathcal{P}^+(\Theta_e|A^1, \ldots, A^T)$ (endpoint is in slice). The tightness of the posterior bounds is increased until a decision can be made and the interval adjusted, if need be. After convergence, $\hat{\Theta}$ is generated uniformly over the interval and is likewise tested for acceptance. We illustrate this procedure in Fig. 1 of the Supplement.

The lower and upper posterior probability bounds can be incorporated in many MCMC algorithms, and provide an effective means of garnering posterior samples assuming the bounds can be efficiently tightened. For DPPs, the upper and lower bounds depend on the truncation of the kernel eigenvalues and can be arbitrarily tightened by including more terms.

In the discrete DPP/k-DPP settings, the eigenvalues can be efficiently computed to a specified point using methods such as power law iterations. The corresponding bounds for a 3600×3600 Gaussian kernel example are shown in Fig. 2. In the continuous setting, explicit truncation can be done when the kernel has Gaussian quality and similarity, as we show in Sec. 5.1. For other continuous DPP kernels, low-rank approximations can be used (Affandi et al., 2013a) resulting in approximate posterior samples (even after convergence of the Markov chain). We believe these methods could be used to get *exact* posterior samples by extending the discrete-DPP Nyström theory of Affandi et al. (2013b), but this is beyond the scope of this paper. In contrast, a gradient ascent algorithm for MLE is not even feasible: we do not know the form of the approximated eigenvalues, so we cannot take their derivative.

Explicit forms for the DPP/k-DPP posterior probability bounds as a function of the eigenvalue truncations follow from Prop. 3.1 and 3.2 combined with Eqs. (8) and (9), respectively. Proofs are in the Supplement.

Proposition 3.1. *Let $\lambda_{1:\infty}$ be the eigenvalues of kernel L. Then*

$$\prod_{n=1}^{M}(1+\lambda_n) \leq \prod_{n=1}^{\infty}(1+\lambda_n) \tag{16}$$

and

$$\prod_{n=1}^{\infty}(1+\lambda_n) \leq \exp\left\{\text{tr}(L) - \sum_{n=1}^{M}\lambda_n\right\}\left[\prod_{n=1}^{M}(1+\lambda_n)\right] .$$

Proposition 3.2. *Let $\lambda_{1:\infty}$ be the eigenvalues of kernel L. Then*

$$e_k(\lambda_{1:M}) \leq e_k(\lambda_{1:\infty}) \tag{17}$$

and

$$e_k(\lambda_{1:\infty}) \leq \sum_{j=0}^{k}\frac{(\text{tr}(L) - \sum_{n=1}^{M}\lambda_n)^j}{j!}e_{k-j}(\lambda_{1:M}) .$$

Note that the expression $\text{tr}(L)$ in the bounds can be easily computed as either $\sum_{i=1}^{N} L_{ii}$ in the discrete case or $\int_\Omega L(\mathbf{x},\mathbf{x})d\mathbf{x}$ in the continuous case.

4. Method of Moments

In this section, we derive a set of DPP moments that can be used in a variety of ways. For example, we can compute the theoretical moments associated with each of our posterior samples and use these as summary statistics in assessing convergence of the MCMC sampler, e.g., via Gelman-Rubin diagnostics (Gelman & Rubin, 1992). Likewise, if we observe that these posterior-sample-based moments do not cover the empirical moments of the data, this can usefully hint at a lack of posterior consistency and a potential need to revise the misspecified prior.

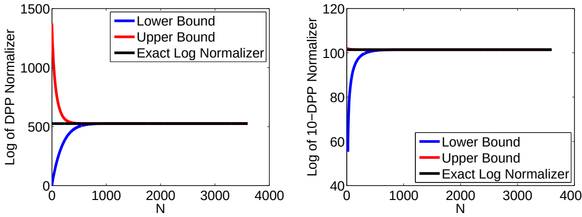

Figure 2. Normalizer bounds for a discrete DPP (*left*) and a 10-DPP (*right*) with Gaussian quality and similarity as in Eqs. (11) and (12) and Ω a grid of 3600 points.

In the discrete case, we first need to compute the marginal probabilities. Borodin (2009) shows that the marginal kernel, K, can be computed directly from L:

$$K = L(I+L)^{-1} . \tag{18}$$

The mth moment can then be calculated via

$$\mathbb{E}[\mathbf{x}^m] = \sum_{i=1}^{N}\mathbf{x}_i^m K(\mathbf{x}_i,\mathbf{x}_i) . \tag{19}$$

In the continuous case, given the eigendecomposition of the kernel operator, $L(\mathbf{x},\mathbf{y}) = \sum_{n=1}^{\infty}\lambda_n\phi_n(\mathbf{x})^*\phi_n(\mathbf{y})$ (where $\phi_n(\mathbf{x})^*$ denotes the complex conjugate of the nth eigenfunction), the mth moment is

$$\mathbb{E}[\mathbf{x}^m] = \int_\Omega \sum_{n=1}^{\infty}\frac{\lambda_n}{\lambda_n+1}\mathbf{x}^m\phi_n(\mathbf{x})^2 d\mathbf{x} . \tag{20}$$

Note that Eq. (20) generally cannot be evaluated in closed form since the eigendecompositions of most kernel operators are not known. However, in certain cases, such as the Gaussian kernel of Sec. 5.1 with eigenfunctions given by Hermite polynomials, the moments can be directly computed. In the Supplement, we derive the mth moment for this Gaussian kernel setting.

Unfortunately, the method of moments can be challenging to use for direct parameter learning since Eqs. (19) and (20) rarely yield analytic forms that are solvable for Θ. In low dimensions, Θ can be estimated numerically, but it is an open question to estimate these moments for large-scale problems.

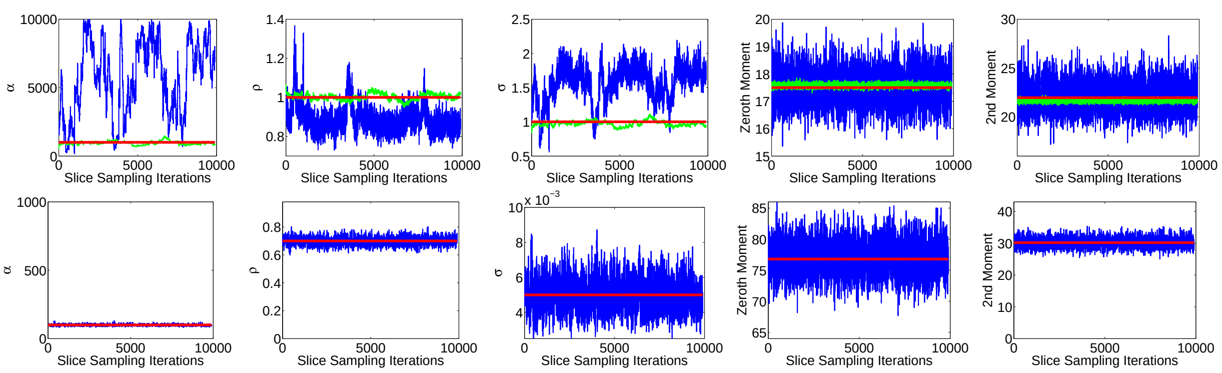

Figure 3. For a continuous DPP with Gaussian quality and similarity, from left to right: Posterior samples of α, ρ and σ, and associated zeroth and second moments. The top row are samples from Scenario (i) (blue) and Scenario (ii) (green) while the second row are samples from Scenario (iii). Red lines indicate the true parameter values that generated the data and their associated theoretical moments. The y-axis scaling aims to place all scenarios on equal footing.

5. Experiments

5.1. Simulations

We provide an explicit example of Bayesian learning for a continuous DPP with the kernel defined by

$$q(\mathbf{x}) = \sqrt{\alpha} \prod_{d=1}^{D} \frac{1}{\sqrt{\pi \rho_d}} \exp\left\{-\frac{x_d^2}{2\rho_d}\right\} \tag{21}$$

$$k(\mathbf{x}, \mathbf{y}) = \prod_{d=1}^{D} \exp\left\{-\frac{(x_d - y_d)^2}{2\sigma_d}\right\}, \ \mathbf{x}, \mathbf{y} \in \mathbb{R}^D. \tag{22}$$

Here, $\Theta = \{\alpha, \rho_d, \sigma_d\}$ and the eigenvalues of the operator $L(\Theta)$ are given by (Fasshauer & McCourt, 2012),

$$\lambda_{\mathbf{m}}(\Theta) = \alpha \prod_{d=1}^{D} \sqrt{\frac{1}{\frac{\beta_d^2+1}{2} + \frac{1}{2\gamma_d}}} \left(\frac{1}{\gamma_d(\beta_d^2+1)+1}\right)^{m_d-1}, \tag{23}$$

where $\gamma_d = \frac{\sigma_d}{\rho_d}$, $\beta_d = (1 + \frac{2}{\gamma_d})^{\frac{1}{4}}$, and $\mathbf{m} = (m_1, \ldots, m_D)$ is a multi-index. Furthermore, the trace of $L(\Theta)$ can be easily computed as

$$\text{tr}(L(\Theta)) = \int_{\mathbb{R}^d} \alpha \prod_{d=1}^{D} \frac{1}{\pi \rho_d} \exp\left\{-\frac{x_d^2}{2\rho_d}\right\} d\mathbf{x} = \alpha \ . \tag{24}$$

We test our Bayesian learning algorithms on simulated data generated from a 2-dimensional isotropic kernel ($\sigma_d = \sigma$, $\rho_d = \rho$ for $d = 1, 2$) using Gibbs sampling (Affandi et al., 2013a). We then learn the parameters under weakly informative inverse gamma priors on σ, ρ and α. Details are in the Supplement. We consider the following simulation scenarios:

(i) 10 DPP samples with average number of points=18 using $(\alpha, \rho, \sigma) = (1000, 1, 1)$

(ii) 1000 DPP samples with average number of points=18 using $(\alpha, \rho, \sigma) = (1000, 1, 1)$

(iii) 10 DPP samples with average number of points=77 using $(\alpha, \rho, \sigma) = (100, 0.7, 0.05)$.

Fig. 3 shows trace plots of the posterior samples for all three scenarios. In the first scenario, the parameter estimates vary wildly whereas in the other two scenarios, the posterior estimates are more stable. In all cases, the zeroth and second moment estimated from the posterior samples are in the neighborhood of the corresponding empirical moments. This leads us to believe that the posterior is broad when we have both a small number of samples and few points in each sample. The posterior becomes more peaked as the total number of points increases. The stationary similarity kernel allows us to garner information either from few sets with many points or many sets of few points.

Dispersion Measure In many applications, we are interested in quantifying the overdispersion of point process data. In spatial statistics, a standard dispersion measure is the Ripley K-function (Ripley, 1977). We instead aim to use the learned DPP parameters (encoding repulsion) to quantify overdispersion. Importantly, our measure should be invariant to scaling. In the Supplement we derive that, as the data are scaled from $\mathbf{x}$ to $\eta\mathbf{x}$, the parameters scale from $(\alpha, \sigma_i, \rho_i)$ to $(\alpha, \eta\sigma_i, \eta\rho_i)$. This suggests that an appropriate scale-invariant repulsion measure is $\gamma_i = \sigma_i/\rho_i$.

5.2. Applications

5.2.1. Diabetic Neuropathy

Recent breakthroughs in skin tissue imaging have spurred interest in studying the spatial patterns of nerve fibers in diabetic patients. It has been observed that these nerve fibers become more clustered as diabetes progresses. Waller et al. (2011) previously analyzed this phenomena based on 6 thigh

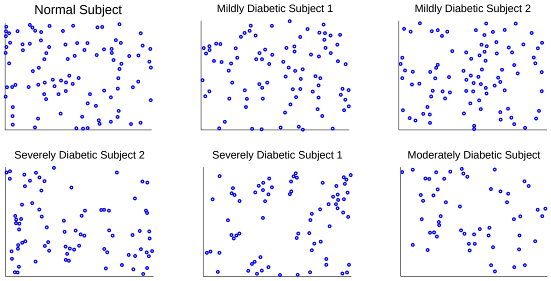

Figure 4. Nerve fiber samples. Clockwise: (i) Normal subject, (ii) Mildly Diabetic Subject 1, (iii) Mildly Diabetic Subject 2,(iv) Moderately Diabetic subject, (v) Severely Diabetic Subject 1 and (vi) Severely Diabetic Subject 2.

nerve fiber samples. These samples were collected from 5 diabetic patients at different stages of diabetic neuropathy and one healthy subject. On average, there are 79 points in each sample (see Fig. 4). Waller et al. (2011) analyzed the Ripley K-function and concluded that the difference between the healthy and severely diabetic samples is highly significant.

We instead study the differences between these samples by learning the kernel parameters of a DPP and quantifying the level of repulsion of the point process. Due to the small sample size, we consider a 2-class study of Normal/Mildly Diabetic versus Moderately/Severely Diabetic. We perform two analyses. In the first, we directly quantify the level of repulsion based on our scale-invariant statistic, $\gamma = \sigma/\rho$ (see Sec. 5.1). In the second, we perform a leave-one-out classification by training the parameters on the two classes with one sample left out. We then evaluate the likelihood of the held-out sample under the two learned classes. We repeat this for all six samples.

We model our data using a 2-dimensional continuous DPP with Gaussian quality and similarity as in Eqs. (21) and (22). Since there is no observed preferred direction in the data, we use an isotropic kernel ($\sigma_d = \sigma$ and $\rho_d = \rho$ for $d = 1, 2$). We place weakly informative inverse gamma priors on (α, ρ, σ), as specified in the Supplement, and learn the parameters using slice sampling with eigenvalue bounds as outlined in Sec. 3.3. The results shown in Fig. 5 indicate that our γ measure clearly separates the two classes, concurring with the results of Waller et al. (2011). Furthermore, we are able to correctly classify all six samples. While the results are preliminary, being based on only six observations, they show promise for this task.

5.2.2. Diversity in Images

We also examine DPP learning for quantifying how visual features relate to human perception of diversity in different image categories. This is useful in applications such as image search, where it is desirable to present users with a set of images that are not only relevant to the query, but diverse as well.

Building on work by Kulesza & Taskar (2011a), three image categories—cars, dogs and cities—were studied. Within each category, 8-12 subcategories (such as *Ford* for cars, *London* for cities and *poodle* for dogs) were queried from Google Image Search and the top 64 results were retrieved. For a subcategory subcat, these images form our base set Ω_{subcat}. To assess human perception of diversity, annotated sets of size six were generated from these base sets. However, it is challenging to ask a human to coherently select six diverse images from a set of 64 total. Instead, Kulesza & Taskar (2011a) generated a *partial result set* of five images from a 5-DPP on each Ω_{subcat} with a kernel based on the SIFT256 features (see the Supplement). Human annotators (via Amazon Mechanical Turk) were then presented with two images selected at random from the remaining subcategory images and asked to add the image they felt was least similar to the partial result set. These experiments resulted in about 500 samples spread evenly across the different subcategories.

We aim to study how the human annotated sets differ from the top six Google results, Top-6. As in Kulesza & Taskar (2011a), we extracted three types of features from the images—color features, SIFT descriptors (Lowe, 1999; Vedaldi & Fulkerson, 2010) and GIST descriptors (Oliva & Torralba, 2006) described in the Supplement. We denote these features for image i as f_i^{color}, f_i^{SIFT}, and f_i^{GIST}, respectively. For each subcategory, we model our data as a discrete 6-DPP on Ω_{subcat} with kernel

$$L_{i,j}^{\text{subcat}} = \exp\left\{-\sum_{\text{feat}} \frac{\|f_i^{\text{feat}} - f_j^{\text{feat}}\|_2^2}{\sigma_{\text{feat}}^{\text{cat}}}\right\} \tag{25}$$

for feat $\in \{$color, SIFT, GIST$\}$ and i, j indexing the 64 images in Ω_{subcat}. Here, we assume that each category has the same parameters across subcategories, namely, $\sigma_{\text{feat}}^{\text{cat}}$ for subcat $\in$ cat and cat $\in \{$cars, dogs, cities$\}$.

To learn from the Top-6 images, we consider the samples as being generated from a 6-DPP. To emphasize the human component of the 5-DPP + human annotation sets, we examine a conditional 6-DPP (Kulesza & Taskar, 2012) that fixes the five partial results set images and only considers the probability of adding the human-annotated image. The Supplement provides details on this conditional k-DPP.

All subcategory samples within a category are assumed to be independent draws from a DPP on Ω_{subcat} with kernel L^{subcat} parameterized by a shared set of $\sigma_{\text{feat}}^{\text{cat}}$. As such, each of these samples equally informs the posterior of $\sigma_{\text{feat}}^{\text{cat}}$. We samples the posterior of the 6-DPP or conditional 6-DPP kernel parameters using slice sampling with weakly informative inverse gamma priors on the $\sigma_{\text{feat}}^{\text{cat}}$. Details are in the Supplement.

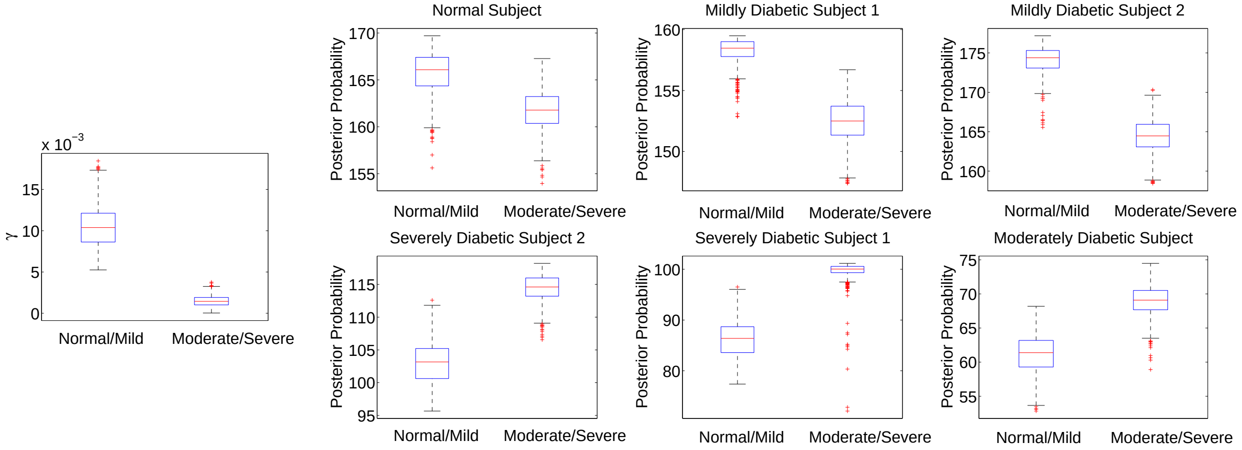

Figure 5. Left: Repulsion measure, γ. *Right:* Leave-one-out log-likelihood of each sample in Fig. 4 (in the same order) under the two learned DPP classes: Normal/Mildly Diabetic (left box) and Moderately/Severely Diabetic (right box).

Fig. 6 shows a comparison between $\sigma_{\text{feat}}^{\text{cat}}$ learned from the human annotated samples (conditioning on the 5-DPP partial result sets) and the Top-6 samples for different categories. The results indicate that the 5-DPP + human annotated samples differs significantly from the Top-6 samples in the features judged by human to be important for diversity in each category. For cars and dogs, human annotators deem color to be a more important feature for diversity than the Google search engine based on their Top-6 results. For cities, on the other hand, the SIFT features are deemed more important by human annotators than by Google. Keep in mind, though, that this result only highlights the diversity components of the results while ignoring quality. In real life applications, it is desirable to combine both the quality of each image (as a measure of relevance of the image to the query) and the diversity between the top results. Regardless, we have shown that DPP kernel learning can be informative of judgements of diversity, and this information could be used (for example) to tune search engines to provide results more in accordance with human judgement.

6. Conclusion

Determinantal point processes have become increasingly popular in machine learning and statistics. While many important DPP computations are efficient, learning the parameters of a DPP kernel is difficult. This is due to the fact that not only is the likelihood function non-convex, but in many scenarios the likelihood and its gradient are either unknown or infeasible to compute. We proposed Bayesian approaches using MCMC for inferring these parameters. In addition to providing a characterization of the posterior uncertainty, these algorithms can be used to deal with large-scale discrete and continuous DPPs based solely on likelihood bounds. We demonstrated the utility of learning DPP parameters in studying diabetic neuropathy and evaluating human perception of diversity in images.

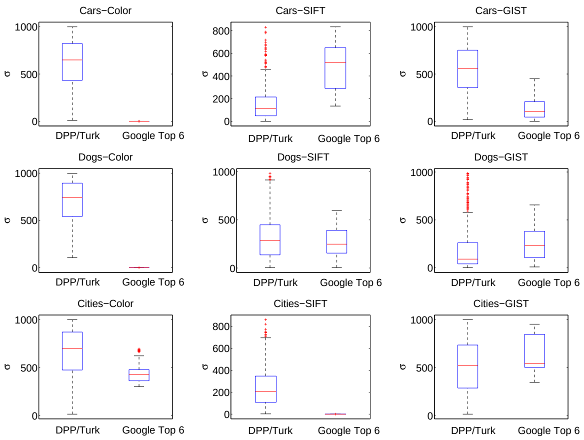

Figure 6. For the image diversity experiment, boxplots of posterior samples of (rom left to right) $\sigma_{\text{color}}^{\text{cat}}$, $\sigma_{\text{SIFT}}^{\text{cat}}$ and $\sigma_{\text{GIST}}^{\text{cat}}$. Each plot shows results for human annotated sets (left) versus Google Top 6 (right). Categories from top to bottom: (a) cars, (b) dogs and (c) cities.

Acknowledgments

This research was supported in part by AFOSR Grant FA9550-12-1-0453 and DARPA Young Faculty Award N66001-12-1-4219.

References

Affandi, R. H., Kulesza, A., and Fox, E. B. Markov determinantal point processes. In *Proc. UAI*, 2012.

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

Affandi, R.H., Kulesza, A., Fox, E.B., and Taskar, B. Nyström approximation for large-scale determinantal processes. In *Proc. AISTATS*, 2013b.

Borodin, A. Determinantal point processes. *arXiv preprint arXiv:0911.1153*, 2009.

Borodin, A. and Rains, E.M. Eynard-Mehta theorem, Schur process, and their Pfaffian analogs. *Journal of Statistical Physics*, 121(3):291–317, 2005.

Fasshauer, G.E. and McCourt, M.J. Stable evaluation of Gaussian radial basis function interpolants. *SIAM Journal on Scientific Computing*, 34(2):737–762, 2012.

Gelman, A. and Rubin, D. B. Inference from iterative simulation using multiple sequences. *Statistical Science*, pp. 457–472, 1992.

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

Hough, J.B., Krishnapur, M., Peres, Y., and Virág, B. Determinantal processes and independence. *Probability Surveys*, 3:206–229, 2006.

Kulesza, A. and Taskar, B. Structured determinantal point processes. In *Proc. NIPS*, 2010.

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

Kulesza, A. and Taskar, B. Learning determinantal point processes. In *Proc. UAI*, 2011b.

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

Lavancier, F., Møller, J., and Rubak, E. Statistical aspects of determinantal point processes. *arXiv preprint arXiv:1205.4818*, 2012.

Lowe, D. G. Object recognition from local scale-invariant features. In *Proc. IEEE International Conference on Computer Vision*, 1999.

McKinnon, K. I.M. Convergence of the Nelder–Mead simplex method to a nonstationary point. *SIAM Journal on Optimization*, 9(1):148–158, 1998.

Neal, R. M. Slice sampling. *Annals of Statistics*, pp. 705–741, 2003.

Nelder, J. and Mead, R. A simplex method for function minimization. *Computer Journal*, 7(4):308–313, 1965.

Oliva, A. and Torralba, A. Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research*, 155:23–36, 2006.

Ripley, B. D. Modelling spatial patterns. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 172–212, 1977.

Snoek, J., Zemel, R., and Adams, R. P. A determinantal point process latent variable model for inhibition in neural spiking data. In *Proc. NIPS*, 2013.

Vedaldi, A. and Fulkerson, B. Vlfeat: An open and portable library of computer vision algorithms. In *Proc. International Conference on Multimedia*, 2010.

Waller, L. A., Särkkä, A., Olsbo, V., Myllymäki, M., Panoutsopoulou, I.G., Kennedy, W.R., and Wendelschafer-Crabb, G. Second-order spatial analysis of epidermal nerve fibers. *Statistics in Medicine*, 30(23):2827–2841, 2011.

Zou, J. and Adams, R.P. Priors for diversity in generative latent variable models. In *Proc. NIPS*, 2012.