1-1-2014

Learning, Large Scale Inference, and Temporal Modeling of Determinantal Point Processes

Raja Mohd Hafiz Affandi Raja Ahmad
University of Pennsylvania, rajahafizaffandi@gmail.com

Learning, Large Scale Inference, and Temporal Modeling of Determinantal Point Processes

Abstract
Determinantal Point Processes (DPPs) are random point processes well-suited for modelling repulsion. In discrete settings, DPPs are a natural model for subset selection problems where diversity is desired. For example, they can be used to select relevant but diverse sets of text or image search results. Among many remarkable properties, they offer tractable algorithms for exact inference, including computing marginals, computing certain conditional probabilities, and sampling.

In this thesis, we provide four main contributions that enable DPPs to be used in more general settings. First, we develop algorithms to sample from approximate discrete DPPs in settings where we need to select a diverse subset from a large amount of items.

Second, we extend this idea to continuous spaces where we develop approximate algorithms to sample from continuous DPPs, yielding a method to select point configurations that tend to be overly-dispersed.

Our third contribution is in developing robust algorithms to learn the parameters of the DPP kernels, which is previously thought to be a difficult, open problem.

Finally, we develop a temporal extension for discrete DPPs, where we model sequences of subsets that are not only marginally diverse but also diverse across time.

Degree Type
Dissertation

Degree Name
Doctor of Philosophy (PhD)

Graduate Group
Statistics

First Advisor
Eric T. Bradlow

Keywords
Determinantal Point Processes, diversity, large-scale, learning, point processes, repulsion

Subject Categories
Computer Sciences | Statistics and Probability

LEARNING, LARGE SCALE INFERENCE, AND TEMPORAL MODELING OF DETERMINANTAL POINT PROCESSES

Raja Mohd Hafiz Affandi Raja Ahmad

A DISSERTATION

in

Statistics

For the Graduate Group in Managerial Science and Applied Economics

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2014

Supervisor of Dissertation

____

Eric Bradlow, K.P. Chao Professor, Marketing, Statistics and Education

Graduate Group Chairperson

____

Eric Bradlow, K.P. Chao Professor, Marketing, Statistics and Education

Dissertation Committee
Dean Foster, Marie and Joseph Melone Professor of Statistics
Alexander Rakhlin, Assistant Professor, Department of Statistics
Emily Fox, Assistant Professor, Amazon Professor of Machine Learning, University of Washington

LEARNING, LARGE SCALE INFERENCE, AND TEMPORAL MODELING OF DETERMINANTAL POINT PROCESSES

Raja Mohd Hafiz Affandi Raja Ahmad

This thesis is dedicated to my beloved, late mother, Raja Hafizah Raja Khalid.

Acknowledgments

In the name of God, the Most Gracious, the Most Merciful. My highest gratitude goes my Lord for His blessing and everything that I have.

My first debt of appreciation must go to my advisor, Emily Fox, for her endless encouragement and guidance without which this work would not have been possible. She is undoubtedly a great mentor and is someone who I look up to throughout my years as her student. This thesis is written in the memory of Ben Taskar, who was not only the smartest person I've known but also possessed a gentle and loving heart. His guidance, patience and kindness will always be remembered. To Alessandro Rinaldo, who was the first to pique my interests in statistics– without you, I wouldn't have even started this journey towards my Ph.D. I would also like to thank my committee members, Eric Bradlow, Dean Foster and Sasha Rakhlin for their valuable advice, helpful suggestions and constructive criticisms in finishing this thesis.

I am forever indebted to the friendship and constant love and support from my family and friends. First and foremost, this thesis is dedicated to my late mother, Raja Hafizah Raja Khalid, whose neverending love made me strive to be the best that I can. To my father, Raja Ahmad Raja Shah Kobat– my deepest love for never giving up on me even when at times it seems that I'm stumbling through life. To my sisters and brother, Mimi (Ayong), Elina (Angah) and Kamil (Abang), thank you for being there and reminding me how important family is. To my aunt and uncle, Raja Hamizah and Zairi, words cannot express how indebted I am to you for treating me like your own son. To my fiancée, Ada– I honestly don't think I could have finished this thesis without you. You are such an inspiration in my life and your endless love and encouragement carries me through to the end.

ABSTRACT

LEARNING, LARGE SCALE INFERENCE, AND TEMPORAL MODELING OF DETERMINANTAL POINT PROCESSES

Raja Mohd Hafiz Affandi Raja Ahmad

Eric Bradlow

Determinantal Point Processes (DPPs) are random point processes well-suited for modelling repulsion. In discrete settings, DPPs are a natural model for subset selection problems where diversity is desired. For example, they can be used to select relevant but diverse sets of text or image search results. Among many remarkable properties, they offer tractable algorithms for exact inference, including computing marginals, computing certain conditional probabilities, and sampling.

In this thesis, we provide four main contributions that enable DPPs to be used in more general settings. First, we develop algorithms to sample from approximate discrete DPPs in settings where we need to select a diverse subset from a large amount of items.

Second, we extend this idea to continuous spaces where we develop approximate algorithms to sample from continuous DPPs, yielding a method to select point configurations that tend to be overly-dispersed.

Our third contribution is in developing robust algorithms to learn the parameters of the DPP kernels, which is previously thought to be a difficult, open problem.

Finally, we develop a temporal extension for discrete DPPs, where we model sequences of subsets that are not only marginally diverse but also diverse across time.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A determinantal point process (DPP) provides a distribution over configurations of points. The defining characteristic of the DPP is that it is a repulsive point process, which makes it useful for modeling diversity. DPPs were first identified as a class by Macchi [1975] who used them to model the distributions of fermion systems at thermal equilibrium. Since the Pauli exclusion principle states that no two fermions can occupy the same quantum state, this repulsiveness is aptly described by a DPP.

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

$$\mathbb{P}_L(A) \propto \det(L_A) \ , \tag{1.1}$$

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. To see this, if we we let B the matrix of feature vectors of points in Ω and let $L = B^\top B$, then the subdeterminant in Eq. (1.1) is proportional to the square of the volume spanned by the kernel vectors associated with the points in A.

Despite the interesting repulsive characteristic they present, DPPs did not receive much attention beyond the community of mathematical physics and probability for a few decades. This changed in the mid-to-late 2000s, when it was found that DPPs in discrete spaces yield appealing mathematical properties including computation of marginal and conditional probabilities [Borodin and Rains, 2005, Borodin, 2009] and efficient sampling algorithms [Hough et al., 2006].

With the advent of the theory in discrete settings, DPPs have recently played an increasingly important role in machine learning and statistics. With discrete DPPs' growing recognition as a method for diverse subset collection, they have been widely applied to tasks such as pose estimation [Kulesza and Taskar, 2010], image search [Kulesza and Taskar, 2011a], news thread discovery [Gillenwater et al., 2012] and neural spiking models [Snoek et al., 2013].

For discrete DPPs, there exists a sampling algorithm that is exact and efficient [Hough et al., 2006]. Given N base points (or items), $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} = \Omega$, sampling can be done using an eigendecomposition of the DPP kernel matrix L and it runs in time $O(N^3)$ despite sampling from a distribution over 2^N subsets. However, when N is very large, an $O(N^3)$ algorithm can be prohibitively slow; for instance, when selecting a subset of frames to summarize a long video. Furthermore, while storing a vector of N items might be feasible, storing an $N \times N$ matrix often is not. The first contribution of this thesis is then to provide an approximate algorithm to sample from large-scale DPPs using low-rank approximations of the kernel matrix. While many low-rank approximation algorithms exist with guaranteed performance in terms of the matrix norm, it is not clear how the errors propagate to the preservation of the distributions defined by DPPs. In this thesis, we will provide bounds on the error of DPPs based on low-rank approximated kernels. Our algorithm will provide a method for diverse subset sampling in large-scale settings. For example, we will consider the application to motion capture video summarization.

DPPs defined on continuous spaces have also been found useful in applications such as repulsive mixture modeling [Zou and Adams, 2012] where generating point configurations that tend to be spread out is desirable. However, the use of DPPs are somewhat limited due to the lack of efficient sampling algorithms. Our second main contribution is this thesis is to propose an efficient algorithm to sample from DPPs in continuous spaces using low-rank approximations of the kernel function. We investigate two such schemes: Nyström and random Fourier features. Our approach utilizes a *dual representation* of the DPP, a technique that has proven useful in the discrete Ω setting as well [Kulesza and Taskar, 2010]. For k-DPPs, which only place positive probability on sets of cardinality k [Kulesza and Taskar, 2012a], we also devise a Gibbs sampler that iteratively samples points in the k-set conditioned on all $k-1$ other points. The derivation relies on representing the conditional DPPs using the Schur complement of the kernel. As a result, our methods allow for the sampling of repulsive point configurations in continuous spaces. We consider their applications to repulsive Gaussian mixture models and synthesis of human motion.

Despite many remarkably efficient algorithms for inference of DPPs, 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 and Taskar, 2012a]. Intuitively, the issue arises from the fact that in seeking to maximize the log-likelihood of DPPs, the numerator yields a concave log-determinant term whereas the normalizer contributes a convex term, leading to a non-convex objective. This non-convexity holds even under various simplifying assumptions on the form of L. Our third main contribution in this thesis is to provide robust algorithms to learn the parameters of the DPP kernel. We propose Bayesian methods to learn the DPP kernel parameters. These methods can be used to efficiently learn both discrete and continuous DPPs, even in cases where gradient-based optimization methods are not even feasible. We provide applications to classifying nerve fiber data in diabetic patients and human judgement of image diversity.

Finally, we present a temporal extension to discrete DPPs, where we model diverse *sequences* of subsets. In many applications, it is desirable to select subsets that are not only marginally diverse but also diverse relative to those previously shown. For example, in displaying news headlines from day-to-day, one aims to select articles that are relevant and diverse on any given day and also diverse to the articles that are shown in the previous days. We construct a *Markov* DPP (M-DPP) for a sequence of random sets that defines a stationary process that maintains DPP margins, implying that the subset chosen is encouraged to be diverse at a particular time t. Crucially, the union of consecutive subsets is also marginally DPP-distributed implying diversity across time as well. We will consider the application of M-DPP in displaying daily news headline.

We believe this thesis will open up opportunities to use DPPs as parts of many models, both in discrete and continuous settings, as well as spur further research in developing theoretical properties and efficient algorithms for general classes of DPPs.

1.1 Contributions

Our main contributions are summarized below.

- We provide an approximate algorithm to sample from large-scale DPPs using low-rank approximations of the kernel matrix. We provide bounds on the error of DPPs based on these low-rank approximated kernels.

- We propose propose an efficient algorithm to sample from DPPs in continuous spaces using low-rank approximations of the kernel function. We provide bounds on the error of DPPs based on these low-rank approximations. For fixed-sized k-DPPs, we also devise a Gibbs sampler that iteratively samples points in the k-set conditioned on all $k-1$ other points.

- We provide robust algorithms to learn the parameters of the DPP kernel using Bayesian methods. These methods can be extended to efficiently learn large-scale discrete and continuous DPPs.

- We present a temporal extension to discrete DPPs, where we model diverse *sequences* of subsets. Our construction defines a stationary process that maintains DPP margins, implying that the subset chosen is encouraged to be not only diverse at a particular time t but the union of consecutive subsets are also marginally DPP-distributed implying diversity across time as well.

- We provide a variety of experimental results along the way, demonstrating the successful application of our methods to real-world tasks including video summarization, repulsive mixture model, repulsive social network clustering, human motion synthesis, diabetic neuropathy classification, human judgement of image diversity and news retrieval.

1.2 Thesis Outline

We summarize the chapters that follow.

- **Chapter 2: Background** We present a survey of DPPs including existing sampling and learning algorithms. We will also present a number of well-known low-rank approximation methods and MCMC-based algorithms which we will heavily rely on in sampling and learning large-scale/continuous DPPs.

- **Chapter 3: Large-Scale Inference of Discrete DPPs** We provide our approximate sampling algorithm for large-scale discrete DPPs along with error bounds associated with our methods and an empirical study of these bounds, along with applications to motion capture video summarization.

- **Chapter 4: Inference of Continuous DPPs** We extend the sampling algorithm to the continuous case in addition to a Gibbs sampling-type algorithm for fixed-sized DPPs and apply them to repulsive Gaussian mixture models and synthesis of human motion.

- **Chapter 5: Large-Scale Learning of DPPs**: We propose Bayesian algorithms to learn kernel parameters of DPPs. We show how MCMC-type algorithms can be modified to enable efficient learning of large-scale discrete and continuous DPPs, even when likelihoods cannot be exactly or efficiently computed but can be bounded instead. We provide applications to classifying nerve fiver data in diabetic patients and human judgement of image diversity.

- **Chapter 6: Markov DPPs** We present our construction of *Markov* DPPs that defines a stationary process that maintains individual diversity as well as diversity across time. We provide their applications to displaying daily news headline.

- **Chapter 7: Conclusion** We summarize our contributions and discuss their significance and limitations. We mention possibilities for future work.

Chapter 2

Background

In this chapter, we first present a survey of DPPs. We provide a definition of DPPs in discrete spaces and its interpretation and explore existing sampling and learning algorithms. We then present the extension of DPPs to continuous spaces. Finally, we present a number of well-known low-rank approximation methods and MCMC-based algorithms which we will heavily rely on in sampling and learning large-scale/continuous DPPs.

2.1 Discrete DPPs

A random point process $\mathcal{P}$ on a discrete base set $\Omega = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ is a probability measure on the set 2^Ω of all subsets of Ω. $\mathcal{P}$ is called a determinantal point process (DPP) if there exists a positive semidefinite matrix L indexed by elements of Ω such that the probability of sampling a set $A \subseteq \Omega$ is given by [Borodin and Rains, 2005]:

$$\mathcal{P}_L(A) = \frac{\det(L_A)}{\det(L+I)} \ , \tag{2.1}$$

$L_A \equiv [L_{ij}]_{\boldsymbol{x}_i, \boldsymbol{x}_j \in A}$ is the submatrix of L indexed by the elements in A and I is the $N \times N$ identity matrix. Here we adopt the convention that $\det(L_\emptyset) = 1$. This representation of DPPs is known as the L-ensemble [Borodin and Rains, 2005].

The subdeterminant in Eq. (2.1) has an intuitive geometric interpretation. If we let B be a $D \times N$ matrix such that $L = B^\top B$, then if we denote the columns of B by B_i, $i = 1, 2, \ldots, N$, we get the following expression:

$$\det(L_A) = \mathrm{Vol}^2(\{B_i\}_{\boldsymbol{x}_i \in A}) \tag{2.2}$$

where the right hand side denotes the square $|A|$-dimensional volume of the parallelepiped spanned by the columns of B corresponding to the elements in A. For this reason, DPPs give high probability to subsets that are diverse since they are more orthogonal (and hence span larger volumes). Fig. 2.1 shows the difference between sampling a set of points in the plane using a DPP (with L_{ij} inversely related to the distance between points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$) and sampling points independently. While the latter results in random clumping of the points, DPP sampling, on the other hand, leads to a relatively uniformly spread set with good coverage.

Due to the fact that DPPs favor selecting subsets that are diverse, they have played an importand role in many machine learning tasks such as pose estimation [Kulesza and Taskar, 2010], image search [Kulesza and Taskar, 2011a], salient news thread discovery [Gillenwater et al., 2012] and neural spiking [Snoek et al., 2013] where getting a good coverage of items is key. Coupled with appealing mathematical properties and efficient algorithms explored in the next few subsections, discrete DPPs are fast becoming an integral part of machine learning and statistics.

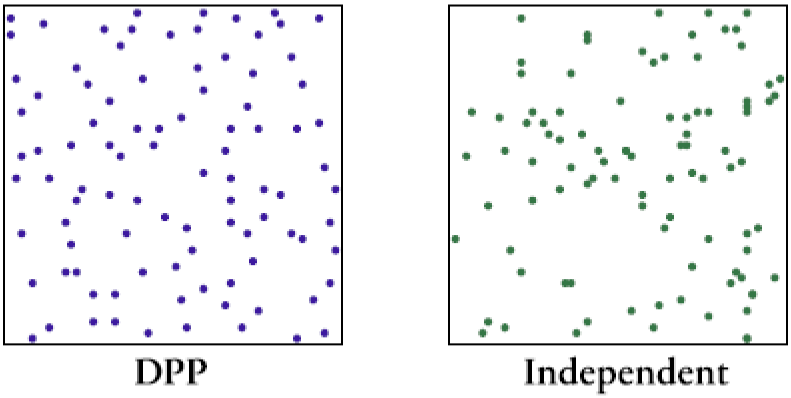


Figure 2.1: A set of points in the plane drawn from a DPP (left), and the same number of points sampled independently using a Poisson point process (right). This figure is taken from Kulesza and Taskar [2012a].

2.1.1 Marginals and Conditionals

Marginals A DPP can also be represented in terms of its marginal kernel. Let K be a semidefinite matrix such that $K \preceq I$ (all eigenvalues less than or equal to 1). Then if $\boldsymbol{Y}$ is a random set drawn according to the DPP with marginal kernel K, then for every $A \subseteq \Omega$:

$$\mathcal{P}(\boldsymbol{Y} \supseteq A) = \det(K_A) \ . \tag{2.3}$$

Once again, $K_A \equiv [K_A]_{\boldsymbol{x}_i, \boldsymbol{x}_j \in A}$ denotes the submatrix of K indexed by elements in A, and we adopt the convention that $\det(K_\emptyset) = 1$. If we think of K_{ij} as measuring the similarity between items i and j, then

$$\mathcal{P}(\boldsymbol{Y} \supseteq \{\boldsymbol{x}_i, \boldsymbol{x}_j\}) = K_{ii}K_{jj} - K_{ij}^2 \tag{2.4}$$

implies that $\boldsymbol{Y}$ is unlikely to contain both $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ when they are very similar. Here as well, a DPP can be seen as modeling a collection of diverse items from the base set Ω. It can be shown that an L-ensemble is a DPP with marginal kernel

$K = L(I+L)^{-1} = I - (L+I)^{-1}$ [Macchi, 1975]. Conversely, a DPP with marginal kernel K has L-ensemble kernel $L = K(I-K)^{-1}$ (when the inverse exists).

Conditionals For any $A, B \subseteq \mathcal{Y}$ with $A \cap B = \emptyset$, it can be shown that [Kulesza and Taskar, 2012a]

$$\mathcal{P}_L(\boldsymbol{Y} = A \cup B | \boldsymbol{Y} \supseteq A) = \frac{\det(L_{A \cup B})}{\det(L + I_{\Omega \setminus A})} \ , \tag{2.5}$$

where $I_{\Omega \setminus A}$ is a matrix with ones on the diagonal entries indexed by the elements of $\Omega \setminus A$ and zeros elsewhere.

This conditional distribution is itself a DPP over the elements of $\Omega \setminus A$ [Borodin and Rains, 2005]. In particular, suppose $\boldsymbol{Y}$ is DPP-distributed with L-ensemble kernel L, and condition on the fact that $\boldsymbol{Y} \supseteq A$. Then the set $\boldsymbol{Y} \setminus A$ is DPP-distributed with marginal and L-ensemble kernels

$$K^A = \left[I - (L + I_{\Omega \setminus A})^{-1}\right]_{\Omega \setminus A} \tag{2.6}$$

$$L^A = \left(\left[(L + I_{\Omega \setminus A})^{-1}\right]_{\Omega \setminus A}\right)^{-1} - I \ . \tag{2.7}$$

Here, $[\cdot]_{\Omega \setminus A}$ denotes the submatrix of the argument indexed by elements in $\Omega \setminus A$. Thus, DPPs as a class are closed under most natural conditioning operations.

2.1.2 Sampling Discrete DPPs

Hough et al. [2006] first described the DPP sampling algorithm shown in Algorithm 1. Phase 1 is to compute an eigendecomposition $L = \sum_{n=1}^{N} \lambda_n v_n v_n^\top$ of the kernel matrix; from this, a random subset V of the eigenvectors is chosen by using the eigenvalues to bias a sequence of coin flips. The Phase 2 of algorithm proceeds iteratively, on each iteration selecting a new item y_i to add to the sample and then

Algorithm 1 Sampling discrete DPPs

```
Input: L-ensemble kernel matrix $L$
$\{(v_n, \lambda_n)\}_{n=1}^{N} \leftarrow$ eigenvector/value pairs of $L$
$J \leftarrow \emptyset$
for $n = 1, \ldots, N$ do
    $J \leftarrow J \cup \{n\}$ with prob. $\frac{\lambda_n}{\lambda_n + 1}$
$V \leftarrow \{v_n\}_{n \in J}$
$Y \leftarrow \emptyset$
while $|V| > 0$ do
    Select $y_i$ from $\Omega$ with $\Pr(y_i)= \frac{1}{|V|} \sum_{v \in V} (v^\top \boldsymbol{e}_i)^2$
    $Y \leftarrow Y \cup \{y_i\}$
    $V \leftarrow V_\perp$, an orthonormal basis for the subspace of $V$ orthogonal to $e_i$
Output: $Y$
```

updating V in a manner that de-emphasizes items similar to the one just selected. Note that $\boldsymbol{e}_i$ is the ith elementary basis vector whose elements are all zero except for a one in position i. Alg. 1 runs in time $O(N^3 + Nk^3)$, where N is the number of available items and k is the cardinality of the returned sample.

The sampling algorithm has an intuitive interpretation. The initial sampling of a point is proportional to the square of the projection of the points to the collection of selected eigenvectors. Thus points that align closer to the eigenvectors have a higher probability of being chosen. However, as each successive point is selected and V is updated by Gram-Schmidt orthogonalization, the distribution shifts to avoid points near those already chosen. Fig. 2.2 shows a progression for a DPP sampling over points in the unit square.

2.1.3 Fixed-sized kDPPs

In selecting a diverse collection of elements in Ω, a DPP jointly models both the size of a set and its content. In some applications, the goal is to select (diverse) sets of a fixed size. In order to achieve this goal, we can instead consider a fixed-size determinantal point processes, or kDPP [Kulesza and Taskar, 2011a], which gives

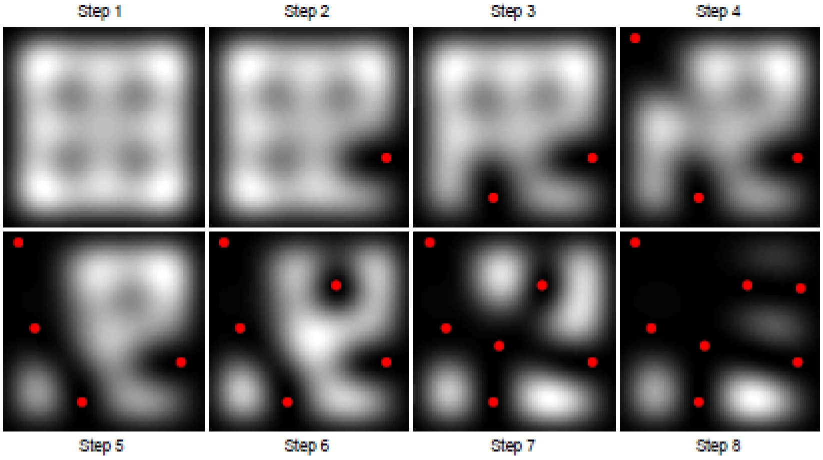


Figure 2.2: Sampling DPP over two-dimensional grid positions. Red circles indicate already selected positions. On the bottom,lighter color corresponds to higher probability. The DPP naturally reduces the probabilities for positions that are similar to those already selected. This figure is taken from Kulesza and Taskar [2012a].

a distribution over all random subsets $\boldsymbol{Y} \subseteq \Omega$ with fixed cardinality k. The L-ensemble construction of a kDPP, denoted $\mathcal{P}_L^k$, gives probabilities

$$\mathcal{P}_L^k(\boldsymbol{Y} = A) = \frac{\det(L_A)}{\sum_{|B|=k} \det(L_B)} \tag{2.8}$$

for all sets A with cardinality k and any positive semidefinite kernel L. While the normalization constant cannot expressed as conveniently as the regular DPP, Kulesza and Taskar [2011a] show that the kDPP distribution can be written as

$$\mathcal{P}_L^k(A) = \frac{\det(L_A)}{e_k(\lambda_1, \ldots, \lambda_N)} \;, \tag{2.9}$$

Algorithm 2 Sampling from a kDPP

```
Input: L-ensemble kernel matrix $L$, size $k$
$\{(v_n, \lambda_n)\}_{n=1}^N \leftarrow$ eigenvector/value pairs of $L$
$J \leftarrow \emptyset$
for $n = N, \ldots, 1$ do
  if $u \sim U[0,1] < \lambda_n \frac{e_{k-1}^{n-1}}{e_k^n}$ then
    $J \leftarrow J \cup \{n\}$
    $k \leftarrow k - 1$
    if $k = 0$ then
      break
{continue with the rest of Algorithm 1}
```

where $\lambda_1, \ldots, \lambda_N$ are eigenvalues of L and $e_k(\lambda_1, \ldots, \lambda_N)$ is the kth elementary symmetric polynomial:

$$e_k(\lambda_1, \ldots, \lambda_N) = \sum_{|J|=k} \prod_{n \in J} \lambda_n \ . \tag{2.10}$$

Note that $e_k(\lambda_1, \ldots, \lambda_N)$ can be efficiently computed using recursion [Kulesza and Taskar, 2012b].

The algorithm to sample from a kDPP is highlighted in Alg. 2.

2.1.4 Quality-Similarity Decomposition

An intuitive way to think of the L-ensemble kernel L is as a Gram matrix [Kulesza and Taskar, 2010]:

$$L(\boldsymbol{x}, \boldsymbol{y}) = q(\boldsymbol{x}) k(\boldsymbol{x}, \boldsymbol{y}) q(\boldsymbol{y}) \ , \tag{2.11}$$

interpreting $q(\boldsymbol{x}) \in \mathbb{R}^+$ as representing the intrinsic *quality* of an item $\boldsymbol{x}$, and $k(\boldsymbol{x}, \boldsymbol{y}) \in [-1, 1]$ as unit representing the *similarity* between items $\boldsymbol{x}$ and $\boldsymbol{y}$. Under this framework, we can model quality and similarity separately to encourage the DPP to choose *high quality* items that are *dissimilar* to each other. This is very useful in many applications such as image or document search where, in response to a search query, we can provide a very relevant (i.e. high quality) but diverse (i.e. dissimilar) list of results.

2.1.5 Dual Representation of DPPs

While DPPs are remarkable in that sampling requires only $O(N^3)$ time despite covering 2^N sets, they are still computationally prohibitive when N is large. In special cases where L is a linear kernel of low dimension, Kulesza and Taskar [2010] showed that the complexity of sampling from these DPPs can be be significantly reduced. In particular, when $L = B^\top B$, with B a $D \times N$ matrix and $D \ll N$, the complexity of the sampling algorithm can be reduced to $O(D^3)$. This arises from the fact that L and the dual kernel matrix $C = BB^\top$ share the same nonzero eigenvalues, and for each eigenvector v_k of L, Bv_k is the corresponding eigenvector of C. This leads to the sampling algorithm given in Alg. 3, which takes time $O(D^3 + ND)$ and space $O(ND)$.

Even in cases where the dimension D is large but finite, the complexity of the DPP sampling algorithm can be further reduced by randomly projecting the linear kernel onto a much lower dimension. Gillenwater et al. [2012] showed how such random projections yield an approximate model with bounded variational distance to the original DPP.

2.1.6 Learning Discrete DPPs

Assume we observe data $A_1, A_2, \ldots, A_T$ with $A_t \subseteq \Omega$ and we model our DPP kernel as $L(\boldsymbol{x}, \boldsymbol{y}; \Theta)$ with parameters Θ. Our log-likelihood then is given by

$$\mathcal{L}(\Theta) = \sum_{t=1}^{T} \log(\det(L_{A_t}(\Theta))) - T \log(\det(L(\Theta) + I)). \tag{2.12}$$

Algorithm 3 Dual-DPP-Sample(B)

```
Input: $B$ such that $L = B^\top B$.
$\{(\hat{\boldsymbol{v}}_n, \lambda_n)\}_{n=1}^N \leftarrow$ eigendecomposition of $C = BB^\top$
$J \leftarrow \emptyset$
for $n = 1, \ldots, N$ do
    $J \leftarrow J \cup \{n\}$ with prob. $\frac{\lambda_n}{\lambda_n + 1}$
$\hat{V} \leftarrow \left\{ \frac{\hat{\boldsymbol{v}}_n}{\sqrt{\hat{\boldsymbol{v}}^\top C \hat{\boldsymbol{v}}}} \right\}_{n \in J}$
$Y \leftarrow \emptyset$
while $|\hat{V}| > 0$ do
    Select $\boldsymbol{y}_i$ from $\Omega$ with $\Pr(\boldsymbol{y}_i) = \frac{1}{|\hat{V}|} \sum_{\hat{\boldsymbol{v}} \in \hat{V}} (\hat{\boldsymbol{v}}^\top B_i)^2$
    $Y \leftarrow Y \cup \{\boldsymbol{y}_i\}$
    Let $\hat{\boldsymbol{v}}_0$ be a vector in $\hat{V}$ with $B_i^\top \hat{\boldsymbol{v}}_0 \neq 0$
    Update $\hat{V} \leftarrow \left\{ \hat{\boldsymbol{v}} - \frac{\hat{\boldsymbol{v}}^\top B_i}{\hat{\boldsymbol{v}}_0^\top B_i} \hat{\boldsymbol{v}}_0 \mid \hat{\boldsymbol{v}} \in \hat{V} - \{\hat{\boldsymbol{v}}_0\} \right\}$
    Orthonormalize $\hat{V}$ w.r.t. $\langle \hat{\boldsymbol{v}}_1, \hat{\boldsymbol{v}}_2 \rangle = \hat{\boldsymbol{v}}_1^\top C \hat{\boldsymbol{v}}_2$
Output: $Y$
```

Despite many remarkable properties of discrete DPPs presented in the preceeding subsections, learning the DPP kernel parameters Θ is still considered a difficult open problem. In particular, if we considered decomposing the kernel into quality and similarity, as in Sec. 2.1.4, Kulesza and Taskar [2012a] have conjectured that learning the parameters of the diversity kernel is NP-hard. Intuitively, the issue arises from the fact that in seeking to maximize the log-likelihood of DPPs in Eq. (2.12), the term associated with numerator yields a concave log-determinant term whereas the the term associated with normalizer contributes a convex term, leading to a non-convex objective. This non-convexity holds even under various simplifying assumptions on the form of L. However, attempts have been made at kernel learning as discussed below.

Quality Learning. Assuming the diversity kernel, $k(\boldsymbol{x}, \boldsymbol{y})$ is held fixed, Kulesza and Taskar [2011b] presented a method to learn the quality kernel. In particular, given features $f(\boldsymbol{x}_i)$ of a point $\boldsymbol{x}_i$, they modeled the quality function as

$$q(\boldsymbol{x}_i) = \exp\left(\frac{1}{2}\theta^\top f(\boldsymbol{x}_i)\right), \tag{2.13}$$

and showed that Eq. (2.12) is now concave in Θ. They proved that the gradient can be computed efficiently as

$$\nabla\mathcal{L}(\theta) = \sum_{t=1}^{T}\left[\sum_{\boldsymbol{x}_i \in A_t} f(\boldsymbol{x}_i) - \sum_{i=1}^{N} K_{ii} f(\boldsymbol{x}_i)\right] \tag{2.14}$$

where K is the marginal kernel in Section 2.1.1.

Mixture of Experts. Kulesza and Taskar [2011a] considered the case where we are given a set $L^{(1)}, L^{(2)}, \ldots, L^{(D)}$ of available *expert* kernel matrices and define the convex combination model

$$\mathcal{P}_\alpha(A) = \sum_{d=1}^{D} \alpha_d \mathcal{P}_{L^{(d)}}(A), \tag{2.15}$$

where $\sum_{d=1}^{D} \alpha_d = 1$. Given observations $A_1, A_2, \ldots, A_T$, α is learned by optimizing the logistic loss measure:

$$\begin{aligned} \min_{\alpha} \quad & \mathcal{L}(\alpha) = \sum_{t=1}^{T} \log(1 + e^{\gamma[\mathcal{P}_\alpha(A_t) - \mathcal{P}_\alpha(\Omega \setminus A_t)]}) \\ \text{s.t.} \quad & \sum_{d=1}^{D} \alpha_d = 1, \end{aligned} \tag{2.16}$$

where γ is a hyperparameter that controls how aggresively we penalize non-included sets. Kulesza and Taskar [2011a] show that this optimization problem is convex and can be solved using projected gradient methods.

Nelder-Mead Optimization The only known method for parameter learning of general kernels is suggested by Lavancier et al. [2012] using Nelder-Mead optimization to maximize Eq. (2.12). Nelder-Mead optimization [Nelder and Mead, 1965] is a heuristic optimization technique that uses the multidimensional simplex to locate a local optimum. As such the method does not require the knowledge of the derivative of the log-likelihood. Note, however, this method can converge to a non-stationary point [McKinnon, 1998] and even when it does converge to a stationary point, it cannot be guaranteed to be the global optimum.

2.2 Continuous DPPs

Let $\Omega \subseteq \mathbb{R}^d$ be a continuous space. To define DPPs on Ω, we first consider a function $L : \Omega \times \Omega \to \mathbb{R}$ with

$$\int_\Omega \int_\Omega |L(\boldsymbol{x}, \boldsymbol{y})|^2 d\boldsymbol{x} d\boldsymbol{y} < \infty \ . \tag{2.17}$$

Such a function is called the Hilbert-Schmidt kernel with the associated Hilbert-Schmidt operator T given by

$$(Tu)(\boldsymbol{x}) = \int_\Omega L(\boldsymbol{x}, \boldsymbol{y}) u(\boldsymbol{y}) d\boldsymbol{y} \ . \tag{2.18}$$

We also assume that $L(\boldsymbol{x}, \boldsymbol{y}) = L(\boldsymbol{y}, \boldsymbol{x})$ and that $L(\boldsymbol{x}, \boldsymbol{y})$ is a positive semidefinite kernel function. Thus, T is a compact, self-adjoint operator and so $L(\boldsymbol{x}, \boldsymbol{y})$ has the eigenfunction expansion

$$L(\boldsymbol{x}, \boldsymbol{y}) = \sum_{n=1}^{\infty} \lambda_n \phi_n(\boldsymbol{x}) \phi_n(\boldsymbol{y}) \ , \tag{2.19}$$

with $\lambda_n \geq 0$ for all n.

Finally, we assume that

$$\mathrm{tr}(T) = \int_{\Omega} L(\boldsymbol{x}, \boldsymbol{x}) d\boldsymbol{x} < \infty \ , \tag{2.20}$$

which defines T as a trace class operator with a well-defined Fredholm determinant,

$$\det(T + I) = \prod_{n=1}^{\infty} (1 + \lambda_n) \ , \tag{2.21}$$

since $\det(T + I) \leq e^{\mathrm{tr}(T)}$.

For notational convenience, in the remainder of this thesis, we suppress the use of the operator T. Thus, we will use expressions such as $\mathrm{tr}(L)$, $\det(L)$ and $\lambda(L)$ with the understanding that those quantities are defined by the Hilbert-Schmidt operator associated with kernel $L(\boldsymbol{x}, \boldsymbol{y})$.

Given a symmetric, positive semidefinite Hilbert-Schmidt kernel, L, we can now define a DPP on Ω with kernel L with probability density given by Eq. (2.1) where L_A is the $|A| \times |A|$ matrix with entries $L(\mathbf{x}, \mathbf{y})$ for each $\mathbf{x}, \mathbf{y} \in A$ and $\det(L + I)$ is the Fredholm determinant of the associated operator defined by Eq. (2.21).

DPPs extend to the continuous settings naturally, with L now a kernel operator instead of a matrix. Again appealing to Eq. (2.1), the DPP probability density for point configurations $A \subseteq \Omega$ is given by

$$\mathcal{P}_L(A) = \frac{\det(L_A)}{\prod_{n=1}^{\infty}(\lambda_n + 1)} \ , \tag{2.22}$$

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

The kDPP also extends to the continuous case with

$$\mathcal{P}_L^k(A) = \frac{\det(L_A)}{e_k(\lambda_{1:\infty})} \ , \tag{2.23}$$

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

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 exponentiated quadratic.

2.2.1 Example of Continuous Kernels

Here we present a few standard continuous kernels and results associated with them. We first decompose the kernel L in terms of the quality and similarity component, akin to what is done in Sec. 2.1.4.

Gaussian Quality and Similarity

Consider $\Omega = \mathbb{R}^D$. Let the the quality and similarity be

$$q(\boldsymbol{x}) = \sqrt{\alpha} \prod_{d=1}^{D} \frac{1}{\sqrt{\pi \rho_d}} \exp\left\{ -\frac{x_d^2}{2\rho_d} \right\} \tag{2.24}$$

and

$$k(\boldsymbol{x}, \boldsymbol{y}) = \prod_{d=1}^{D} \exp\left\{ -\frac{(x_d - y_d)^2}{2\sigma_d} \right\}, \mathbf{x}, \mathbf{y} \in \mathbb{R}^D. \tag{2.25}$$

This kernel is appealing since many of its mathematical properties are known. In particular, the eigenvalues and eigenfunctions are given by [Fasshauer and McCourt, 2012]:

$$\lambda_{\mathbf{n}} = \alpha \prod_{d=1}^{D} \sqrt{\frac{1}{\frac{\beta_d^2+1}{2} + \frac{1}{2\gamma_d}}} \left(\frac{1}{\gamma_d(\beta_d^2+1)+1} \right)^{n_d - 1}, \tag{2.26}$$

and

$$\phi_{\mathbf{n}}(\boldsymbol{x}) = \prod_{d=1}^{D} \left(\frac{1}{\pi \rho_d^2} \right)^{\frac{1}{4}} \sqrt{\frac{\beta_d}{2^{n_d-1}\Gamma(n_d)}} \exp\left\{ -\frac{\beta_d^2 x^2}{2\rho_d^2} \right\} H_{n_d-1}\left(\frac{\beta_d x_d}{\sqrt{\rho_d^2}} \right), \tag{2.27}$$

where $\gamma_d = \frac{\sigma_d}{\rho_d}$, $\beta_d = (1 + \frac{2}{\gamma_d})^{\frac{1}{4}}$ and $\mathbf{n} = (n_1, n_2, \ldots, n_D)$ is a multi index.

Uniform Quality and Gaussian Similarity

Consider $\Omega = [-\frac{1}{2}, \frac{1}{2}]^D$, a hypercube of volume 1. Let the the quality and similarity be

$$q(\boldsymbol{x}) = \sqrt{\frac{\alpha}{\pi^D}} 1_{\boldsymbol{x} \in [-\frac{1}{2}, \frac{1}{2}]^D} \ , \tag{2.28}$$

$$k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left\{-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{\sigma}\right\} \ . \tag{2.29}$$

This kernel represents models where points are given uniform quality within a hypercube and points repulse one another depending on their Gaussian distances. Unfortunately, no known exact eigendecomposition is known. However, since the kernel is translationally invariant (ie. $L(\boldsymbol{x}, \boldsymbol{y}) = L(\boldsymbol{x} - \boldsymbol{y})$) and defined over a compact space, we can approximate its eigenstructure using the Fourier basis

$$\phi_{\mathbf{n}}(\boldsymbol{x}) = \exp\left\{2\pi i \mathbf{n}^\top \boldsymbol{x}\right\}, \quad \mathbf{n} \in \mathbb{Z}^D. \tag{2.30}$$

The approximate eigenvalues are then given by [Lavancier et al., 2012]

$$\lambda_{\mathbf{n}} = \alpha\sigma e^{-\pi^2\sigma\|\mathbf{n}\|^2}. \tag{2.31}$$

Uniform Quality and Matérn Similarity

Let $\Omega = [-\frac{1}{2}, \frac{1}{2}]^D$, a hypercube of volume 1. Let the the quality and similarity be

$$q(\boldsymbol{x}) = \sqrt{\rho \frac{2^{1-\nu}}{\Gamma(\nu)\alpha^\nu}} 1_{\boldsymbol{x} \in [-\frac{1}{2}, \frac{1}{2}]^d} \ , \tag{2.32}$$

$$k(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|^\nu \mathcal{K}_\nu(\frac{\|\boldsymbol{x} - \boldsymbol{y}\|}{\alpha}), \tag{2.33}$$

where $\mathcal{K}_\nu$ is the modified Bessel function of the second kind.

No known exact eigendecomposition is known. However, since the kernel is also translationally invariant the approximate eigenvalues using the Fourier basis in Eq. (2.30) is [Lavancier et al., 2012]

$$\lambda_{\mathbf{n}} = 4\pi^D \rho \frac{\nu}{(1 + 4\pi^{2D}\alpha^2 \|\mathbf{n}\|^2)^{1+\nu}}. \tag{2.34}$$

Uniform Quality and Generalized Cauchy Similarity

Let $\Omega = [-\frac{1}{2}, \frac{1}{2}]^D$, a hypercube of volume 1. Let the the quality and similarity be

$$q(\boldsymbol{x}) = \sqrt{\rho} 1_{\boldsymbol{x} \in [-\frac{1}{2}, \frac{1}{2}]^d} , \tag{2.35}$$

$$k(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{(1 + \frac{1}{\alpha^2}\|\boldsymbol{x} - \boldsymbol{y}\|^2)^{1+\nu}} . \tag{2.36}$$

Again, no known exact eigendecomposition is known. The approximate eigenvalues using the Fourier basis in Eq. (2.30) is [Lavancier et al., 2012]

$$\lambda_{\mathbf{n}} = \frac{2^{1-\nu}\pi\alpha^2\rho}{\Gamma(1+\nu)} \|2\pi\alpha\mathbf{n}\|^\nu \mathcal{K}_\nu(\|2\pi\alpha\mathbf{n}\|), \tag{2.37}$$

where $\mathcal{K}_\nu$ is the modified Bessel function of the second kind.

2.2.2 Sampling from a Continuous DPPs

When the eigendecomposition of a continuous kernel $L(\boldsymbol{x}, \boldsymbol{y})$ is known and the DPP is defined on a compact space, Lavancier et al. [2012] suggest a modification of the Hough et al. [2006] algorithm, presented in Sec. 2.1.2. In this case, the first phase of the algorithm involves choosing a random set of *eigenfunctions*, $\phi_i(\boldsymbol{x})$ by using the eigenvalues to bias the sequence of coin flips. However, in the second phase of the algorithm, instead of sampling points from discrete distribution

$$Pr(\boldsymbol{y}_i) = \frac{1}{|V|} \sum_{v \in \hat{V}} (v^\top e_i)^2, \tag{2.38}$$

we have to sample from a continuous distribution

$$f(\boldsymbol{y}_i) = \frac{1}{|V|} \sum_{\phi \in \Phi} \|\phi_n(\boldsymbol{y}_i)\|^2, \tag{2.39}$$

where Φ is the set of selected eigenfunctions. Lavancier et al. [2012] suggest using rejection sampling with uniform density to sample from this distribution. There are serious drawbacks for this method. Note that this method only works for DPPs defined over a compact space since the rejection sampling step relies heavily on using the uniform density over the space as a proposal. Even in this case, an exact eigendecomposition is needed for the sampling algorithm and in most cases, as suggested by Lavancier et al. [2012], we have to resort to an eigenstructure approximation method, such as the Fourier basis which results only in an approximate sampling algorithm.

2.2.3 Learning Continuous DPPs

In the continuous setting, no known exact learning algorithm exists. The only known approximate method for parameter learning of continuous kernels is suggested by Lavancier et al. [2012] using Nelder-Mead optimization to maximize Eq. (2.12) as in the discrete case highlighted in Sec. 2.1.6. Here, however, eigendecompositions are not known for many continuous kernels. In the case that the kernel L is translationally invariant (ie. $L(\boldsymbol{x}, \boldsymbol{y}) = L(\boldsymbol{x} - \boldsymbol{y})$) and defined over a unit square, the eigenvalues of the kernel can be approximated using a Fourier basis, as in Sec. 2.2.1, and thus the approximate log-likelihood can be computed from Eq. (2.22).

Note, that even in this case, we have to truncate the eigenvalues to be able to approximately evaluate the denominator in Eq. (2.22). For non-translationally invariant kernels or DPPs defined over non-compact spaces, no known learning algorithms (even approximate) exist.

2.3 Low-Rank Kernel Approximations

2.3.1 Nyström Approximation

A common method to improve scalibilty of many kernel-based algorithms involves a low rank approximation to the high-dimensional kernel matrix. For many applications, including SVM-based classification, Gaussian process regression, PCA, and, in our case, sampling DPPs, fundamental algorithms require kernel matrix operations of space $O(N^2)$ and time $O(N^3)$. A common way to improve scalability is to create a low-rank approximation to the high-dimensional kernel matrix. One such technique is known as the Nyström method, which involves selecting a small number of landmarks and then using them as the basis for a low rank approximation.

Given a sample W of r landmark items corresponding to a subset of the indices of an $N \times N$ symmetric positive semidefinite matrix L, let $\overline{W}$ be the complement of W (with size $N - r$), let L_W and $L_{\overline{W}}$ denote the principal submatrices indexed by W and $\overline{W}$, respectively, and let $L_{\overline{W}W}$ denote the $(N-r) \times r$ submatrix of L with row indices from $\overline{W}$ and column indices from W. Then we can write L in block form as

$$L = \begin{pmatrix} L_W & L_{W\overline{W}} \\ L_{\overline{W}W} & L_{\overline{W}} \end{pmatrix} . \tag{2.40}$$

If we denote the pseudo-inverse of L_W as L_W^+, then the Nyström approximation of

L using W is

$$\tilde{L} = \begin{pmatrix} L_W & L_{W\overline{W}} \\ L_{\overline{W}W} & L_{W\overline{W}} L_W^+ L_{\overline{W}W} \end{pmatrix} . \tag{2.41}$$

Fundamental to this method is the choice of W. Various techniques have been proposed; some have theoretical guarantees, while others have only been demonstrated empirically. Williams and Seeger [2000] first proposed choosing W by uniform sampling without replacement. A variant of this approach was proposed by Frieze et al. [2004] and Drineas and Mahoney [2005], who sample W *with* replacement, and with probabilities proportional to the squared diagonal entries of L. This produces a guarantee that, with high probability,

$$\|L - \tilde{L}\|_2 \leq \|L - L_r\|_2 + \epsilon \sum_{i=1}^{N} L_{ii}^2 . \tag{2.42}$$

where L_r is the best rank-r approximation to L.

Kumar et al. [2012] later proved that the same rate of convergence applies for uniform sampling without replacement, and argued that uniform sampling outperforms other non-adaptive methods for many real-world problems while being computationally cheaper.

In cases where there are large enough eigengaps in the spectrum of L, Jin et al. [2011] proved that we can further improve the spectral norm bound to:

$$\|L - \tilde{L_r}\|_2 \leq \lambda_{r+1} + O\left(\frac{N}{\sqrt{r}}\right) . \tag{2.43}$$

Instead of sampling elements of W from a fixed distribution, Deshpande et al. [2006] introduced the idea of *adaptive* sampling, which alternates between selecting landmarks and updating the sampling distribution for the remaining items. Intuitively, items whose kernel values are poorly approximated under the existing sample are more likely to be chosen in the next round.

By sampling in each round landmarks W_t chosen according to probabilities $p_i^{(t)} \propto \|L_i - \tilde{L}_i(W_1 \cup \cdots \cup W_{t-1})\|_2^2$ (where L_i denotes the ith column of L), we are guaranteed that

$$\mathbb{E}\left(\|L - \tilde{L}(W)\|_F\right) \leq \frac{\|L - L_r\|_F}{1-\epsilon} + \epsilon^T \sum_{i=1}^{N} L_{ii}^2 \ . \tag{2.44}$$

where L_r is the best rank-r approximation to L and $W = W_1 \cup \cdots \cup W_T$.

Kumar et al. [2012] argue that adaptive Nyström methods empirically outperform the non-adaptive versions in cases where the number of landmarks is small relative to N. In fact, their results suggest that the performance gains of adaptive Nyström methods relative to the non-adaptive schemes are inversely proportional to the percentage of items chosen as landmarks.

The Nyström method can also be extended in the continuous case. Given $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_r$ *landmarks* sampled from Ω, we can approximate the kernel function as,

$$\tilde{L}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{j=1}^{r} \sum_{k=1}^{r} W_{jk}^2 L(\boldsymbol{x}, \boldsymbol{z}_j) L(\boldsymbol{z}_k, \boldsymbol{y}), \tag{2.45}$$

where $W_{jk} = L(\boldsymbol{z}_j, \boldsymbol{z}_k)^{-1/2}$.

2.3.2 Random Fourier Features

In cases where the kernel matrix L is generated from a shift-invariant kernel function $k(\boldsymbol{x}, \boldsymbol{y}) = k(\boldsymbol{x} - \boldsymbol{y})$, we can construct a low-rank approximation using random Fourier features (RFF) [Rahimi and Recht, 2007]. This involves mapping each data point $\boldsymbol{x} \in \mathbb{R}^d$ onto a random direction $\boldsymbol{\omega}$ drawn from the Fourier transform of the kernel function. In particular, we draw $\boldsymbol{\omega} \sim p(\boldsymbol{\omega})$, where

$$p(\boldsymbol{\omega}) = \int_{\mathbb{R}^d} k(\boldsymbol{\Delta}) \exp(-i\boldsymbol{\omega}^\top \boldsymbol{\Delta}) d\boldsymbol{\Delta} \ , \tag{2.46}$$

and set $z_{\boldsymbol{\omega}}(\boldsymbol{x}) = e^{i\boldsymbol{\omega}^\top \boldsymbol{x}}$. It can be shown then that $z_{\boldsymbol{\omega}}(\boldsymbol{x})^* z_{\boldsymbol{\omega}}(\boldsymbol{y})$ is an unbiased estimator of $k(\boldsymbol{x} - \boldsymbol{y})$. Here we denote $z_{\boldsymbol{\omega}}(\boldsymbol{x})^*$ as the complex conjugate of $z_{\boldsymbol{\omega}}(\boldsymbol{x})$. Note that the shift-invariant property of the kernel function is crucial to ensure that $p(\boldsymbol{\omega})$ is a valid probability distribution, due to Bochner's Theorem. The variance of the estimate can be improved by drawing D random directions, $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_D \sim p(\boldsymbol{\omega})$ and estimating the kernel function with $k(\boldsymbol{x} - \boldsymbol{y})$ as $\frac{1}{D} \sum_{j=1}^{D} z_{\boldsymbol{\omega}_j}(\boldsymbol{x})^* z_{\boldsymbol{\omega}_j}(\boldsymbol{y})$.

To use RFF for approximating kernel matrices, we assume that the matrix L is generated from a shift-invariant kernel function, so that if $\boldsymbol{x}_i$ is the vector representing item i. Then

$$L_{ij} = k(\boldsymbol{x}_i - \boldsymbol{x}_j) \ . \tag{2.47}$$

We construct a $D \times N$ matrix B with

$$B_{ij} = \frac{1}{\sqrt{D}} z_{\boldsymbol{\omega}_i}(\boldsymbol{x}_j) \qquad i = 1, \ldots, D, j = 1, \ldots, N \ . \tag{2.48}$$

An unbiased estimator of the kernel matrix L is now given by $\tilde{L}^{\mathrm{RFF}} = B^\top B$.

2.4 Markov-Chain Monte Carlo (MCMC)

In Chapter 5, we will consider learning the parameters, Θ of a DPP kernel $L(\Theta)$ from observations $A^1, \ldots, A^T$ by sampling from the posterior distribution

$$\mathbb{P}(\Theta | A^1, \ldots, A^T) \propto \mathbb{P}(\Theta) \mathbb{P}(A^1, \ldots, A^T | \Theta) \ , \tag{2.49}$$

where $\mathbb{P}(A^1, \ldots, A^T|\Theta)$ is the likelihood of Θ given the observations $A^1, \ldots, A^T$ and $\mathbb{P}(\Theta)$ is the prior on Θ. For many combinations of prior and likelihood, including the DPP cases we consider in this thesis, Eq. (2.49) does not yield a closed-form that we can sample directly from. Thus we resort to approximate techniques based on Markov chain Monte Carlo (MCMC) [Robert and Casella, 2004]. MCMC methods provide a class of algorithms that produce estimates of the posterior samples based on iterative sampling, combining Monte Carlo integration with samples from a specially constructed Markov chain. The key feature of these methods is that the sampling procedure does not rely on sampling from the distribution in Eq. (2.49), which is assumed to have an arbitrarily complex form. Here we present two MCMC methods — Metropolis-Hastings (MH) and slice sampling.

2.4.1 Metropolis-Hastings (MH)

The *Metropolis-Hastings (MH)* algorithm provides a generic method for constructing an ergodic Markov chain, relying solely on defining a valid proposal distribution $f(\cdot|\cdot)$ and evaluation of the target distribution $\mathbb{P}(\Theta|A^1, \ldots, A^T)$ up to a normalization constant. It is assumed evaluating $\mathbb{P}(\Theta|A^1, \ldots, A^T)$ is easy, but sampling from this distribution is challenging. We use the 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

$$r = \left(\frac{\mathbb{P}(\hat{\Theta}|A^1, \ldots, A^T)}{\mathbb{P}(\Theta_i|A^1, \ldots, A^T)} \frac{f(\Theta_i|\hat{\Theta})}{f(\hat{\Theta}|\Theta_i)} \right) . \tag{2.50}$$

The MH algorithm is outlined in Alg. 4.

The *randon-walk Metropolis-Hastings* is an example of an MH algorithm where the proposal distribution $f(\hat{\Theta}|\Theta_i)$ is chosen to have mean Θ_i. The hyperparameters

Algorithm 4 Metropolis-Hastings

```
Input: Dimension: $D$, Starting point: $\Theta_0$, Prior distribution: $\mathbb{P}(\Theta)$, Proposal distribution $f(\hat{\Theta}|\Theta)$ with mean $\Theta$, Samples: $A^1, \dots, A^T$].
$\Theta = \Theta_0$
for $i = 0 : (\tau - 1)$ do
  $\hat{\Theta} \sim f(\hat{\Theta}|\Theta_i)$
  $r = \left( \frac{\mathbb{P}(\hat{\Theta}|A^1,\dots,A^T)}{\mathbb{P}(\Theta_i|A^1,\dots,A^T)} \frac{f(\Theta_i|\hat{\Theta})}{f(\hat{\Theta}|\Theta_i)} \right)$
  $u \sim$ Uniform[0,1]
  if $u < \min\{1, r\}$ then
    $\Theta_{i+1} = \hat{\Theta}$
Output: $\Theta_{0:\tau}$
```

of $f(\hat{\Theta}|\Theta_i)$ tune the width of the distribution, determining the average step size.

2.4.2 Slice Sampling

While simpler MCMC methods such as 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. For example, consider the case where we use a Gaussian proposal for some real-valued parameters. If we set the variance of this Gaussian proposal to be small, then the proposed values $\hat{\Theta}$ will be close to the current value, Θ_i, introducing high correlation in the samples. If instead we set a large variance, the proposed values can potentially be far away from the current parameter value. However, in this case, the rejection rate $r = \left(\frac{\mathbb{P}(\hat{\Theta}|A^1,\dots,A^T)}{\mathbb{P}(\Theta_i|A^1,\dots,A^T)} \frac{f(\Theta_i|\hat{\Theta})}{f(\hat{\Theta}|\Theta_i)} \right)$ can be potentially low resulting in a high rejection rate. Thus the hyperparameters of the proposal distrubution have to be tuned to provide balance between those two cases.

To avoid the need to tune the proposal distribution, we can instead use *slice sampling* [Neal, 2003], which automatically adjusts the step-size to match the local shape of the posterior probability while still satisfying detailed balance conditions. We first describe this method in the univariate case, following the "linear stepping-out" approach described by Neal [2003]. Given the current parameter Θ_i, we first sample $y \sim \text{Uniform}[0, \mathbb{P}(\Theta_i|A^1, \ldots, A^T)]$. This defines our *slice* with all values of Θ with $\mathbb{P}(\Theta|A^1, \ldots, A^T)$ 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. The details for univariate slice sampling are shown in Alg. 5 and illustrated in Fig. 2.3.

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 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.

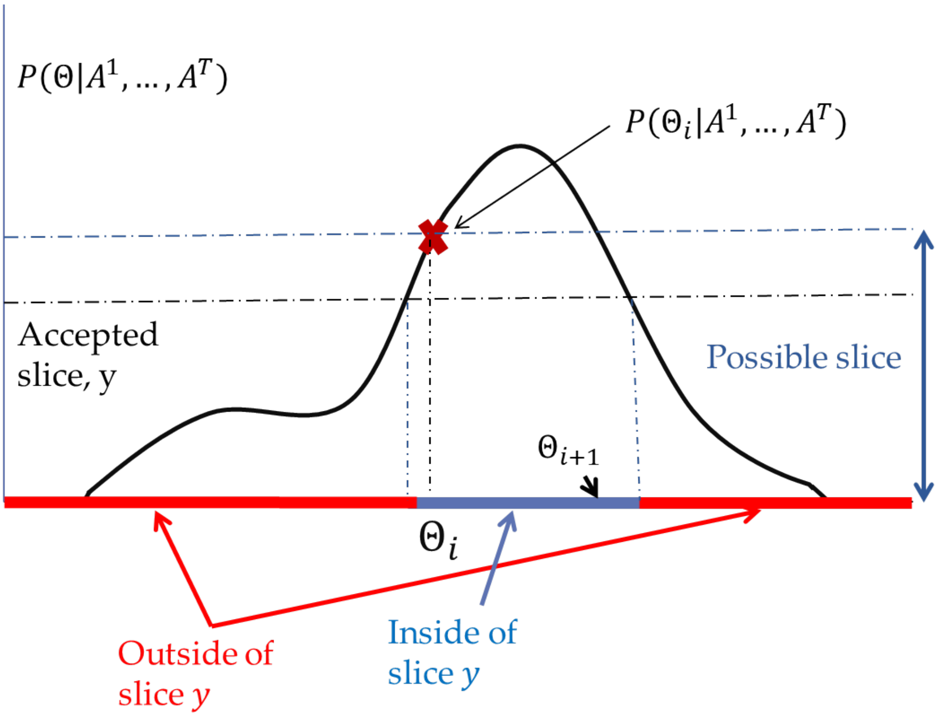


Figure 2.3: Illustration of the univariate slice sampling algorithm. In the first step, A slice y is generated by sampling from Uniform$[0, P(\Theta_i|A^1, \dots, A^T)]$. Once a slice is generated, we need to sample new parameters inside the slice. We start with a predetermined window around the previous parameters. If the endpoints fall inside the slice, the window is linearly expanded until neither endpoint is in the slice. We then 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, thereby shrinking the window. This procedure is repeated until a proposed $\hat{\Theta}$ is accepted.

Algorithm 5 Univariate Slice Sampling

```
Input: Starting point: $\Theta_0$, Initial width: $w$, Prior distribution: $\mathbb{P}(\Theta)$, Samples: $X = [X^1, \dots, X^T]$.
$\Theta = \Theta_0$
for $i = 0 : (\tau - 1)$ do
    $y \sim \text{Uniform}[0, \mathbb{P}(\Theta_i | A^1, \dots, A^T)]$
    $z \sim \text{Uniform}[0, 1]$
    $L = \Theta_i - z * \frac{w}{2}$
    $R = L + \frac{w}{2}$
    while $y > \mathbb{P}(L | A^1, \dots, A^T)$ do
        $L = L - \frac{w}{2}$
    while $y > \mathbb{P}(R | A^1, \dots, A^T)$ do
        $R = R + \frac{w}{2}$
    $\hat{\Theta} \sim \text{Uniform}[L, R]$
    if $\mathbb{P}(\hat{\Theta} | A^1, \dots, A^T) < y$ then
        while $\mathbb{P}(\hat{\Theta} | A^1, \dots, A^T) < y$ do
            if $\hat{\Theta} > \Theta$ then
                $R = \hat{\Theta}$
            else
                $L = \hat{\Theta}$
            $\hat{\Theta} \sim \text{Uniform}[L, R]$
    $\Theta_{i+1} = \hat{\Theta}$
Output: $\Theta_{0:\tau}$
```

Chapter 3

Large-Scale Inference of Discrete DPPs

An appealing aspect of a DPP defined on a discrete space, $\Omega = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ is its efficient sampling algorithm. Despite the probability distribution being defined on the 2^N of all possible subsets of Ω, the DPP sampler of Hough et al. [2006] leads to an $O(N^3)$ algorithm (see Sec. 2.1.2). Imagine, however, machine learning applications such as video summarization where each video might contain tens of thousands of frames or possibly more. Suppose we are interested in selecting a diverse subset from this set of frames by sampling from a discrete DPP. In this case, when N is very large, an $O(N^3)$ algorithm can be prohibitively slow. Furthermore, while storing a vector of N items might be feasible, storing an $N \times N$ matrix often is not.

When the kernel matrix can be decomposed as $L = B^\top B$, where B is a $D \times N$ matrix and $D \ll N$, as presented in Sec. 2.1.5, Kulesza and Taskar [2010] offer a solution by considering sampling from the *dual representation*. Recall that in these cases sampling can be done in $O(D^3)$ time without ever constructing L. If D is finite but large, the complexity of the algorithm can be further reduced by randomly projecting B into a lower-dimensional space. Gillenwater et al. [2012] showed how such random projections yield an approximate model with bounded variational error.

However, linear decomposition of the kernel matrix using low-dimensional (or even finite-dimensional) features may not be possible. Even a simple Gaussian kernel has an infinite-dimensional feature space, and for many applications, including video summarization, the kernel can be even more complex and nonlinear.

Here we address these computational issues by approximating a DPP kernel matrix L with a low rank matrix $\tilde{L}$. There are many low-rank approximation methods with results that are well documented both empirically and theoretically. However, there are significant challenges in extending these results to the DPP. Most existing theoretical results bound the Frobenius or spectral norm of the kernel error matrix, but we show that these quantities are insufficient to give useful bounds on distributional measures like variational distance. Instead, we derive novel bounds for low-rank approximation that are specifically tailored to DPPs, nontrivially characterizing the propagation of the approximation error through the structure of the process.

One such approximation method that satisfies the hypothesis in our bounds is the Nyström method presented in Sec. 2.3.1. Our bounds are provably tight in certain cases, and we demonstrate empirically that the bounds are informative for a wide range of real and simulated data. These experiments also show that the proposed method provides a close approximation for DPPs on large sets.

Another approximation method is the random Fourier features (RFF) presented in Sec. 2.3.2. Here, however, the method does not satisfy our hypothesis making theoretical analysis much more difficult. We will present an empirical comparison between the Nyström and RFF and show that the RFF method still proves to be useful in cases where there is low-correlation between the items.

Finally, we apply our techniques to select diverse and representative frames from a series of motion capture recordings. Based on a user survey, we find that the frames sampled from a Nyström-approximated DPP form better summaries than randomly chosen frames.

3.1 Low-Rank Approximated DPPs/kDPPs

While the *dual representation* of DPPs may not be available in many cases (such as cases where L is generated by infinite-dimensional features), the dual DPP sampling outlined in Sec. 2.1.5 can still be harnessed when one approximates L with a low-rank matrix $\tilde{L}$. Such an r-rank approximation to L combined with the application of the the dual sampling reduces the complexity to $O(r^3 + Nr)$ time and $O(Nr)$ space.

Most analysis of the error of low-rank approximations has been limited to the Frobenius and spectral norms of the residual matrix $L - \tilde{L}$, such as the ones presented in Secs. 2.3.1 and 2.3.2. While matrix norm bounds $\|L - \tilde{L}\|_F$ and $\|L - \tilde{L}\|_2$ are useful in quantifying the effectiveness of the low-rank approximation methods in recovering the kernel matrix, it is not clear how these error bounds propagate to quantifying the error in the DPP distributions. Specifically, since the probability of sampling a specific subset A under a DPP is related to the square of the volume spanned by the feature vectors of items in A (see Sec. 2.1). Unfortunately, for many of these low-rank approximation methods, no volumetric error bounds exist. The challenge here is to study how low-rank approximations simultaneously affects *all* possible minors of L.

In fact, a small error in the matrix norm can have a large effect on the minors of the matrix:

Example 1 *Consider matrices* $L = \mathrm{diag}(M, \ldots, M, \epsilon)$ *and* $\tilde{L} = \mathrm{diag}(M, \ldots, M, 0)$ *for some large* M *and small* ϵ. *Although* $\|L - \tilde{L}\|_F = \|L - \tilde{L}\|_2 = \epsilon$, *for any* A *that includes the final index, we have* $\det(L_A) - \det(\tilde{L}_A) = \epsilon M^{k-1}$, *where* $k = |A|$.

It is conceivable that while errors on some subsets are large, most subsets are well approximated. Unfortunately, this not generally true.

Definition 1 *The variational distance between the DPP with kernel* L *and the DPP with the low-rank approximated kernel* $\tilde{L}$ *is given by*

$$\|\mathcal{P}_L - \mathcal{P}_{\tilde{L}}\|_1 = \frac{1}{2} \sum_{A \in 2^\Omega} |\mathcal{P}_L(A) - \mathcal{P}_{\tilde{L}}(A)| \ . \tag{3.1}$$

The variational distance is a natural global measure of approximation that ranges from 0 to 1. Unfortunately, it is not difficult to construct a sequence of matrices where the matrix norm of $L - \tilde{L}$ tends to zero but the variational distance does not.

Example 2 *Let* L *be a diagonal matrix with entries* $1/N$ *and* $\tilde{L}$ *be a diagonal matrix with* $N/2$ *entries equal to* $1/N$ *and the rest equal to 0. Note that* $||L - \tilde{L}||_F = 1/\sqrt{2N}$ *and* $||L - \tilde{L}||_2 = 1/N$, *which tend to zero as* $N \to \infty$. *However, the variational distance is bounded away from zero. To see this, note that the normalizers are* $\det(L + I) = (1 + 1/N)^N$ *and* $\det(\tilde{L} + I) = (1 + 1/N)^{N/2}$, *which tend to* e *and* $\sqrt{e}$, *respectively. Consider all subsets which have zero mass in the approximation,* $S = \{A : \det(\tilde{L}_A) = 0\}$. *Summing up the unnormalized mass of sets in the complement of* S, *we have* $\sum_{A \notin S} \det(L_A) = \det(\tilde{L} + I)$ *and thus* $\sum_{A \in S} \det(L_A) = \det(L + I) - \det(\tilde{L} + I)$. *Now consider the contribution of just* *the sets in S to the variational distance:*

$$\|\mathcal{P}_L - \mathcal{P}_{\tilde{L}}\|_1 \geq \frac{1}{2}\sum_{A\in S}\left|\frac{\det(L_A)}{\det(L+I)} - 0\right| \tag{3.2}$$

$$= \frac{\det(L+I) - \det(\tilde{L}+I)}{2\det(L+I)}\ , \tag{3.3}$$

which tends to $\frac{e-\sqrt{e}}{2e} \approx 0.1967$ as $N \to \infty$.

One might still hope that pathological cases occur only for diagonal matrices, or more generally for matrices that have high coherence [Candes and Romberg, 2007]. In fact, coherence has previously been used by Talwalkar and Rostamizadeh [2010] to analyze the error of Nyström approximations. Define the coherence

$$\mu(L) = \sqrt{N}\max_{i,j}|v_{ij}|\ , \tag{3.4}$$

where each $\boldsymbol{v}_i$ is a unit-norm eigenvector of L. A diagonal matrix achieves the highest coherence of $\sqrt{N}$ and a matrix with all entries equal to a constant has the lowest coherence of 1. Suppose that $f(N)$ is a sublinear but monotone increasing function with $\lim_{N\to\infty} f(N) = \infty$. We can construct a sequence of kernels L with $\mu(L) = \sqrt{f(N)} = o(\sqrt{N})$ for which matrix norms of the low-rank approximation error tend to zero, but the variational distance tends to a constant.

Example 3 *Let L be a block diagonal matrix with $f(N)$ constant blocks, each of size $N/f(N)$, where each non-zero entry is $1/N$. Let $\tilde{L}$ be structured like L except with half of the blocks set to zero. Note that $\mu^2(L) = f(N)$ by construction and that each block contributes a single eigenvalue of $\frac{1}{f(N)}$; the Frobenius and spectral norms of $L - \tilde{L}$ thus tend to zero as N increases. The DPP normalizers are given by $\det(L+I) = (1+1/f(N))^{f(N)} \to e$ and $\det(\tilde{L}+I) = (1+1/f(N))^{f(N)/2} \to \sqrt{e}$. By a similar argument to the one for diagonal matrices, we can show that variational* *distance tends to* $\frac{e-\sqrt{e}}{2e}$*.*

Unfortunately, in the cases above, the low-rank approximations will yield poor approximations to the original DPPs. Convergence of the matrix norm error alone is thus generally insufficient to obtain tight bounds on the resulting approximate DPP distribution. It turns out that the gap between the eigenvalues of the kernel matrix and the spectral norm error plays a major role in the effectiveness of low-rank approximations for DPPs, as we will show in Theorems 1 and 2. In the examples above, this gap is not large enough for a close approximation; in particular, the spectral norm errors are *equal* to the smallest non-zero eigenvalues. In the next subsection, we derive approximation bounds for DPPs that are applicable to many low-rank approximation algorithms.

3.1.1 Preliminaries

In analyzing the error in terms of the variational distance defined in Eq. (3.1), it is important that we are able to analyze the approximation error for each possible subset, $|\mathcal{P}_L(A) - \mathcal{P}_{\tilde{L}}(A)|$, $A \in 2^\Omega$. Since the DPP/kDPP distribution is given by the determinantal form in Eqs. (2.1) and (2.9), it is only natural that we analyze how the eigenvalues of the original kernel L are affected by the low-rank approximation methods. Below we present a result for positive semidefinite matrices known as Weyl's inequality that characterize this effect on the eigenvalues.

Lemma 1 *[Bhatia, 1997] Let* $L = \tilde{L} + E$*, where* $L, \tilde{L}$ *and* E *are all positive semidefinite* $N \times N$ *matrices with eigenvalues* $\lambda_1 \geq \ldots \geq \lambda_N \geq 0$*,* $\tilde{\lambda}_1 \geq \ldots \geq \tilde{\lambda}_N \geq 0$*, and* $\xi_1 \geq \ldots \geq \xi_N \geq 0$*, respectively. Then*

$$\lambda_n \leq \tilde{\lambda}_m + \xi_{n-m+1} \quad \textit{for} \quad m \leq n \ , \tag{3.5}$$

$$\lambda_n \geq \tilde{\lambda}_m + \xi_{n-m+N} \quad \textit{for} \quad m \geq n \ . \tag{3.6}$$

Going forward, we use the convention $\lambda_i = 0$ for $i > N$. Weyl's inequality gives the following two corollaries.

Corollary 1 *When $\xi_j = 0$ for $j = r+1, \dots, N$, then for $j = 1, \dots, N$,*

$$\lambda_j \geq \tilde{\lambda}_j \geq \lambda_{j+r} \ . \tag{3.7}$$

Proof For the first inequality, let $n = m = j$ in (3.6). For the second, let $m = j$ and $n = j + r$ in (3.5).

Corollary 2 *For $j = 1, \dots, N$,*

$$\lambda_j - \xi_N \geq \tilde{\lambda}_j \geq \lambda_j - \xi_1 \ . \tag{3.8}$$

Proof We let $n = m = j$ in (3.5) and (3.6), then rearrange terms to get the desired result.

3.1.2 Set-wise bounds for DPPs

We are now ready to state set-wise bounds on the approximation error for DPPs and kDPPs using certain classes of low-rank approximations In particular, we assume that under the low-rank approximation schemes, both the approximated kernel $\tilde{L}$ and the error matrix $E = L - \tilde{L}$ are positive semi-definite. Then, for each set $A \subseteq \Omega$, we want to bound the probability gap $|\mathcal{P}_L(A) - \mathcal{P}_{\tilde{L}}(A)|$. Going forward, we use $\mathcal{P}_A \equiv \mathcal{P}_L(A)$ and $\tilde{\mathcal{P}}_A \equiv \mathcal{P}_{\tilde{L}}(A)$.

Theorem 1 *Assume $\tilde{L}$ is positive semi-definite and has rank r, L has rank m with eigenvalues $\lambda_1 \geq, \lambda_2, \geq, \dots, \geq \lambda_m$, $|A| = k$, and L_A has eigenvalues $\lambda_1^A \geq \dots \geq \lambda_k^A$. Furthermore, assume that the error matrix $E = L - \tilde{L}$ is also postive semi-definite* *with rank $m-r$. Let*

$$\hat{\lambda}_i = \max\left\{\lambda_{i+(m-r)}, \lambda_i - \|L - \tilde{L}\|_2\right\} \tag{3.9}$$

$$\hat{\lambda}_i^A = \max\left\{\lambda_i^A - \|L - \tilde{L}\|_2, 0\right\} \ . \tag{3.10}$$

Then

$$\begin{aligned} &|\mathcal{P}_A - \tilde{\mathcal{P}}_A| \\ &\leq \mathcal{P}_A \max\left\{\left[1 - \frac{\prod_{i=1}^k \hat{\lambda}_i^A}{\prod_{i=1}^k \lambda_i^A}\right], \left[\frac{\prod_{i=1}^n (1+\lambda_i)}{\prod_{i=1}^n (1+\hat{\lambda}_i)} - 1\right]\right\} \ . \end{aligned} \tag{3.11}$$

Proof

$$\begin{aligned} \tilde{\mathcal{P}}_A - \mathcal{P}_A &= \mathcal{P}_A \left[\frac{\det(L+I)\det(\tilde{L}_A)}{\det(\tilde{L}+I)\det(L_A)} - 1\right] \\ &= \mathcal{P}_A \left[\left(\frac{\prod_{i=1}^n (1+\lambda_i)}{\prod_{i=1}^n (1+\tilde{\lambda}_i)}\right)\left(\frac{\prod_{i=1}^k \tilde{\lambda}_i^A}{\prod_{i=1}^k \lambda_i^A}\right) - 1\right] \ . \end{aligned} \tag{3.12}$$

Here $\tilde{\lambda}_1^A, \geq, \ldots, \geq \tilde{\lambda}_k^A$ are the eigenvalues of $\tilde{L}_A$. Now note that $L_A = \tilde{L}_A + E_A$. Since E is positive semidefinite, E_A is also positive semidefinite. Thus by Corollary 1 we have $\lambda_i^A \geq \tilde{\lambda}_i^A$, and so

$$\tilde{\mathcal{P}}_A - \mathcal{P}_A \leq \mathcal{P}_A \left[\frac{\prod_{i=1}^n (1+\lambda_i)}{\prod_{i=1}^n (1+\hat{\lambda}_i)} - 1\right] \ . \tag{3.13}$$

For the reverse inequality, we multiply the left hand side of the above expression by -1 and use the fact that $\lambda_i \geq \tilde{\lambda}_i$ and $\lambda_i^A \geq 0$. By Corrolary 2,

$$\tilde{\lambda}_i^A \geq \lambda_i^A - \xi_1^A \geq \lambda_i^A - \xi_1 = \lambda_i^A - \|L - \tilde{L}\|_2 \ , \tag{3.14}$$

resulting in the inequality

$$\mathcal{P}_A - \tilde{\mathcal{P}}_A \leq \mathcal{P}_A \left[1 - \frac{\prod_{i=1}^k \hat{\lambda}_i^A}{\prod_{i=1}^k \lambda_i^A} \right] . \tag{3.15}$$

The theorem follows by combining the two inequalities.

The theorem is tight if the approximation is exact ($\|L - \tilde{L}\|_2 = 0$). It can also be shown that these bounds are tight for the diagonal matrix examples discussed at the beginning of this section, where the spectral norm error is equal to the non-zero eigenvalues. Moreover, these bounds are convenient since they are expressed in terms of the spectral norm of the error matrix and therefore can be easily combined with existing approximation bounds for the Nyström method. Note that the eigenvalues of L and the size of the set A both play important roles in the bound. In fact, these two quantities are closely related; it is possible to show that the expected size of a set sampled from a DPP is

$$\mathbb{E}\left[|A|\right] = \sum_{n=1}^{N} \frac{\lambda_n}{\lambda_n + 1} . \tag{3.16}$$

Thus, if L has large eigenvalues, we expect the Nyström approximation error to be large as well since the DPP associated with L gives high probability to large sets.

3.1.3 Set-wise bounds for kDPPs

We can obtain similar results for low-rank approximated kDPPs. In this case, for each set A with $|A| = k$ we want to bound the probability gap $|\mathcal{P}_A^k - \tilde{\mathcal{P}}_A^k|$. Here we let $\mathcal{P}_A^k$ denote Eq. (2.9). As with the DPP case, this can be achieved by analyzing the effect of low-rank approximation to the eigenvalues of L. In the lemma below, we characterize how the denominator of Eq. (2.9) is affected by the low-rank approximation to L.

Lemma 2 *Let e_k once again denote the kth elementary symmetric polynomial of L:*

$$e_k(\lambda_1, \ldots, \lambda_N) = \sum_{|J|=k} \prod_{n \in J} \lambda_n \ , \tag{3.17}$$

and

$$\hat{\lambda}_i = \max \left\{ \lambda_{i+(m-r)}, \lambda_i - \|L - \tilde{L}\|_2 \right\} \ , \tag{3.18}$$

where m is the rank of L and r is the rank of $\tilde{L}$. Then

$$e_k(\lambda_1, \ldots, \lambda_N) \geq e_k(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_N) \geq e_k(\hat{\lambda}_1, \ldots, \hat{\lambda}_N) \ . \tag{3.19}$$

Proof

$$e_k(\lambda_1, \ldots, \lambda_N) = \sum_{|J|=k} \prod_{n \in J} \lambda_n \geq \sum_{|J|=k} \prod_{n \in J} \tilde{\lambda}_n = e_k(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_N) \ ,$$

by Corollary 1.

On the other hand,

$$e_k(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_N) = \sum_{|J|=k} \prod_{n \in J} \tilde{\lambda}_n \geq \sum_{|J|=k} \prod_{n \in J} \hat{\lambda}_n = e_k(\hat{\lambda}_1, \ldots, \hat{\lambda}_N) \ ,$$

by Corollary 1 and Corollary 2.

Since

$$\mathcal{P}_L^k(A) = \frac{\det(L_A)}{\sum_{|A'|=k} \det(L_{A'})} = \frac{\det(L_A)}{\sum_{|J|=k} \prod_{n \in J} \lambda_n} = \frac{\det(L_A)}{e_k(\lambda_1, \ldots, \lambda_N)} \ , \tag{3.20}$$

we can now prove the following theorem:

Theorem 2 *Under the conditions of Theorem 1,*

$$\begin{aligned} &|\mathcal{P}_A^k - \tilde{\mathcal{P}}_A^k| \\ &\le \mathcal{P}_A^k \max\left\{ \left[\frac{e_k(\lambda_1,\dots,\lambda_N)}{e_k(\hat{\lambda}_1,\dots,\hat{\lambda}_N)} - 1\right], \left[1 - \frac{\prod_{i=1}^k \hat{\lambda}_i^A}{\prod_{i=1}^k \lambda_i^A}\right] \right\} . \end{aligned} \tag{3.21}$$

Proof

$$\begin{aligned} \tilde{\mathcal{P}}_A^k - \mathcal{P}_A^k &= \mathcal{P}_A^k \left[\frac{e_k(\lambda_1,\dots,\lambda_N)\det(\tilde{L}_A)}{e_k(\tilde{\lambda}_1,\dots,\tilde{\lambda}_N)\det(L_A)} - 1\right] \\ &= \mathcal{P}_A^k \left[\left(\frac{e_k(\lambda_1,\dots,\lambda_N)}{e_k(\tilde{\lambda}_1,\dots,\tilde{\lambda}_N)}\right)\left(\frac{\prod_{i=1}^k \tilde{\lambda}_i^A}{\prod_{i=1}^k \lambda_i^A}\right) - 1\right] . \end{aligned} \tag{3.22}$$

Here $\tilde{\lambda}_1^A, \ge, \dots, \ge \tilde{\lambda}_k^A$ are the eigenvalues of $\tilde{L}_A$. Now note that $L_A = \tilde{L}_A + E_A$. Since E is positive semidefinite, it follows that E_A is also positive semidefinite. Thus by Corollary 1, we have $\lambda_i^A \ge \tilde{\lambda}_i^A$ and so

$$\tilde{\mathcal{P}}_A^k - \mathcal{P}_A^k \le \mathcal{P}_A^k \left[\frac{e_k(\lambda_1,\dots,\lambda_N)}{e_k(\tilde{\lambda}_1,\dots,\tilde{\lambda}_N)} - 1\right] \le \mathcal{P}_A^k \left[\frac{e_k(\lambda_1,\dots,\lambda_N)}{e_k(\hat{\lambda}_1,\dots,\hat{\lambda}_N)} - 1\right] ,$$

where the last inequality follows from Lemma 2.

On the other hand,

$$\mathcal{P}_A^k - \tilde{\mathcal{P}}_A^k = \mathcal{P}_A^k \left[1 - \left(\frac{e_k(\lambda_1,\dots,\lambda_N)}{e_k(\tilde{\lambda}_1,\dots,\tilde{\lambda}_N)}\right)\left(\frac{\prod_{i=1}^k \tilde{\lambda}_i^A}{\prod_{i=1}^k \lambda_i^A}\right)\right] . \tag{3.23}$$

By Corrolary 2,

$$\tilde{\lambda}_i^A \ge \lambda_i^A - \xi_1^A \ge \lambda_i^A - \xi_1 = \lambda_i^A - \|L - \tilde{L}\|_2 . \tag{3.24}$$

We also note that $\tilde{\lambda}_i^A \ge 0$. Since $e_k(\lambda_1,\dots,\lambda_N) \ge e_k(\tilde{\lambda}_1,\dots,\tilde{\lambda}_N)$ by Lemma 2, we have

$$\mathcal{P}^k_A - \tilde{\mathcal{P}}^k_A \leq \mathcal{P}^k_A \left[1 - \frac{\prod_{i=1}^k \hat{\lambda}_i^A}{\prod_{i=1}^k \lambda_i^A} \right] . \tag{3.25}$$

The theorem follows by combining the two inequalities.

Note that the scale of the eigenvalues has no effect on the kDPP; we can directly observe from Eq. (2.8) that scaling L does not change the kDPP distribution since any constant factor appears to the kth power in both the numerator and denominator.

3.1.4 Nyström-approximated DPPs

In Secs. 3.1.2 and 3.1.3, we presented the theory for low-rank approximated DPP kernels with certain conditions on the approximation method. In particular, we assume that both the approximated kernel $\tilde{L}$ and the error matrix $E = L - \tilde{L}$ are positive semi-definite. One such technique that satisfies these conditions in Theorems 1 and 2 is the Nyström method. To see this, we invoke the following lemma:

Lemma 3 *[Arcolano, 2011] Let $\tilde{L}$ be a Nyström approximation of L. Let $E = L - \tilde{L}$ be the corresponding error matrix. Then E is positive semidefinite with* $\mathrm{rank}(E) = \mathrm{rank}(L) - \mathrm{rank}(\tilde{L})$.

In this section we present empirical results on the performance of the Nyström approximation for kDPPs using three datasets small enough for us to perform ground-truth inference in the original kDPP. Two of the datasets are derived from real-world applications available on the UCI repository[1]—the first is a linear kernel

[1] `http://archive.ics.uci.edu/ml/`

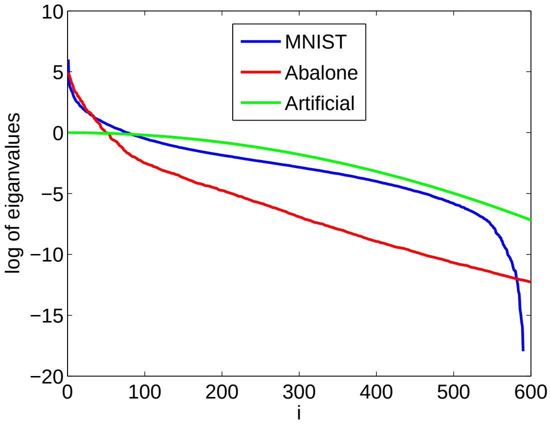


Figure 3.1: The first 600 log-eigenvalues for each dataset.

matrix constructed from 1000 MNIST images, and the second an RBF kernel matrix constructed from 1000 Abalone data points—while the third is synthetic and comprises a 1000 × 1000 diagonal kernel matrix with exponentially decaying diagonal elements. Fig. 3.1 displays the log-eigenvalues for each dataset.

On each dataset, we perform the Nyström approximation with three different sampling schemes: stochastic adaptive, greedy adaptive, and uniform. The stochastic adaptive sampling technique is a simplified version of the scheme used in Deshpande et al. [2006], where, on each iteration of landmark selection, we update $E = L - \tilde{L}$ and then sample landmarks with probabilities proportional to E_{ii}^2. In the greedy scheme, we perform a similar update, but always choose the landmarks with the maximum diagonal value E_{ii}. Finally, for the uniform method, we simply sample the landmarks uniformly without replacement.

In Fig. 3.2 (top), we plot $\log ||L - \tilde{L}||_2$ for each dataset as a function of the number of landmarks sampled. For the MNIST data all sampling algorithms initially perform equally well, but uniform sampling becomes relatively worse after about 550 landmarks are sampled. For the Abalone data the adaptive methods perform much better than uniform sampling over the entire range of sampled

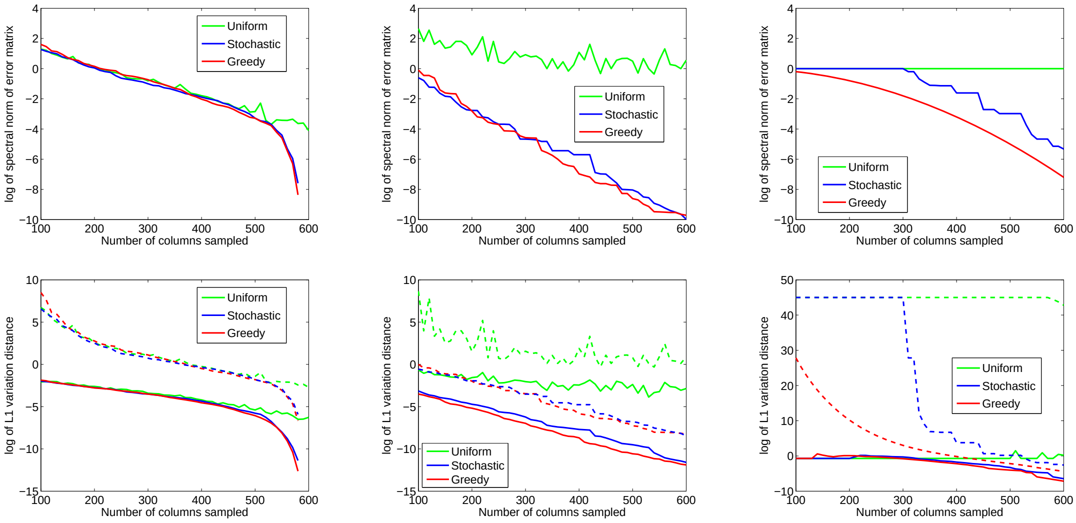


Figure 3.2: Error of Nyström approximations. *Top:* $\log(\|L - \tilde{L}\|_2)$ as a function of number of landmarks sampled. *Bottom:* $\log(\|\mathcal{P} - \tilde{\mathcal{P}}\|_1)$ as a function of number of landmarks sampled. The dashed lines show the bounds derived in Sec. 3.1. From left to right, the datasets used are MNIST, Abalone and Artificial.

landmarks. This phenomenon is perhaps explained by the analysis of Talwalkar and Rostamizadeh [2010], which suggests that uniform sampling works well for the MNIST data due to its relatively low coherence ($\mu(L) = 0.5\sqrt{N}$), while performing poorly on the higher-coherence Abalone dataset ($\mu(L) = 0.8\sqrt{N}$). For both of the UCI datasets, the stochastic and greedy adaptive methods perform similarly. However, for our artificial dataset it is easy to see that the greedy adaptive scheme is optimal since it chooses the top remaining eigenvalues in each iteration.

In Fig. 3.2 (bottom), we plot $\log ||P - \tilde{P}||_1$ for $k = 10$ (estimated by sampling), as well as the theoretical bounds from Sec. 3.1. The bounds track the actual variational error closely for both the MNIST and Abalone datasets. For the artificial dataset uniform sampling can do arbitrarily poorly, so we see looser bounds in this case. We note that the variational distance correlates strongly with the spectral norm error for each dataset.

3.1.5 RFF-approximated DPPs

The Nyström technique is, of course, not the only possible means of finding low-rank kernel approximations. One alternative for shift-invariant kernels is random Fourier features (RFF) [Rahimi and Recht, 2007]. RFFs map each item onto a random direction drawn from the Fourier transform of the kernel function; this results in a uniform approximation of the kernel matrix. In practice, however, reasonable RFF approximations seem to require a large number of random features, which can reduce the computational benefits of this technique.

While RFF approximations guarantee unbiased estimates of the elements of the kernel matrix, the approximation of the minors could be, in theory, highly biased especially for large subsets. For this reason, RFF approximations can potentially be a poor approximation for DPPs in cases where there are large correlation between items. Furthermore, the error matrix $E = L - \tilde{L}$ are not guaranteed to be positive semi-definite, making theoretical analysis much more difficult.

We instead performed empirical comparisons between the Nyström methods and random Fourier features (RFFs). We apply the RFF approximation method to the Abalone dataset. We use a Gaussian RBF kernel,

$$L_{ij} = \exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\sigma^2}) \qquad i, j = 1, \ldots, 1000 \ , \tag{3.26}$$

with σ^2 taking values 0.1,1, and 10. In this case, the Fourier transform of the kernel function, $p(\boldsymbol{\omega})$ is also a multivariate Gaussian.

In Fig. 3.3 we plot the empirically estimated $\log(\|\mathcal{P}^k - \tilde{\mathcal{P}}^k\|_1)$ for $k = 10$. While RFFs compare favorably to the uniform random sampling of landmarks, their performance is significantly worse than that of the adaptive Nyström methods, especially in the case where there are strong correlations between items ($\sigma^2 = 1$

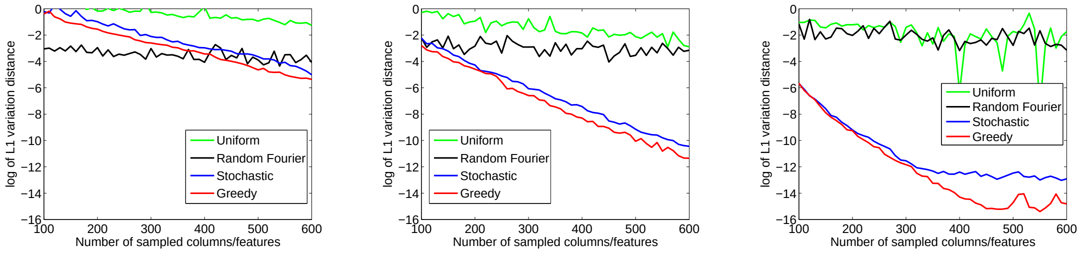


Figure 3.3: Error of Nyström and random Fourier features approximations: $\log(\|\mathcal{P} - \tilde{\mathcal{P}}\|_1)$ as a function of the number of landmarks sampled/random features used. From left to right, the values of σ^2 are 0.1, 1, and 10.

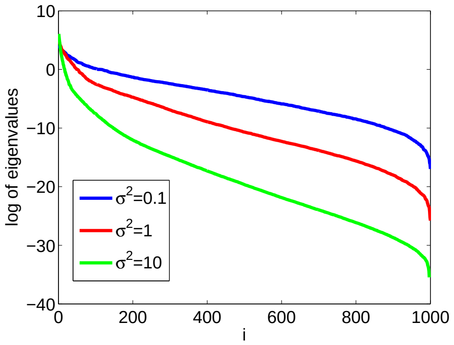


Figure 3.4: The log-eigenvalues of RBF kernel applied to the Abalone datset.

and 10). In the extreme case where there is little to no correlation, the Nyström methods suffer because a small sample of landmarks cannot reconstruct the other items accurately. Yang et al. [2012] have previously demonstrated that, in kernel learning tasks, the Nyström methods perform favorably compared to RFFs in cases where there are large eigengaps in the kernel matrix. The plot of the eigenvalues in Fig. 3.4 suggests that a similar result holds for approximating DPPs as well. In practice, for kernel learning tasks, the RFF approach typically requires more features than the number of landmarks needed for Nystroöm methods. However, due the fact that sampling from a DPP requires $O(r^3)$ time where r is the number of landmarks, we are constrained by the number of landmarks that can be used.

3.2 Experiments

Recall our discussion earlier in this chapter regarding the task of video summarization. Since videos typically contain tens of thousands of frames, if not more, selecting a diverse set of frames via DPP sampling is prohibitively slow. Applying the DPP sampling algorithm using low-rank approximation methods in this chapter, however, makes sampling a diverse subset selection task manageable. In Sec. 3.1, we have shown that the error of a low-rank approximated DPP can be bounded theoretically in terms of the variational distance. Furthermore, empirical analysis on small datasets seem to suggest agreement with the theoretical bound presented. This leads us to believe that sampling from a low-rank approximated DPP mantains the diversity of the resulting selection. In this section, we test this idea on a real life task of summarizing motion capture video.

In particular, we demonstrate the Nyström approximation on a motion summarization task that is too large to permit tractable inference in the original DPP. As input, we are given a series of motion capture recordings, each of which depicts human subjects performing motions related to a particular activity, such as dancing or playing basketball. In order to aid browsing and retrieval of these recordings in the future, we would like to choose, from each recording, a small number of frames that summarize its motions in a visually intuitive way. Since a good summary should contain a diverse but representative set of frames, a DPP is a natural model for this task.

We obtained test recordings from the CMU motion capture database[2], which offers motion captures of over 100 subjects performing a variety of actions. Each capture involves 31 sensors attached to the subject's body and sampled 120 times per second. For each of nine activity categories—basketball, boxing, dancing,

[2] http://mocap.cs.cmu.edu/

exercise, jumping, martial arts, playground, running, and soccer—we made a large input recording by concatenating all available captures in that category. On average, the resulting recordings are about $N = 24{,}000$ frames long (min 3,358; max 56,601). At this scale, storage of a full $N \times N$ DPP kernel matrix would be highly impractical (requiring up to 25GB of memory), and $O(N^3)$ SVD would be prohibitively expensive.

In order to model the summarization problem as a DPP, we designed a simple kernel to measure the similarity between pairs of poses recorded in different frames. We first computed the variance for the location of each sensor for each activity; this allowed us to tailor the kernel to the specific motion being summarized. For instance, we might expect a high variance for foot locations in dancing, and a relatively smaller variance in boxing. We then used these variance measurements to specify a Gaussian kernel over the position of each sensor, and finally combined the Gaussian kernels with a set of weights chosen manually to approximately reflect the importance of each sensor location to human judgments of pose similarity. Specifically, for poses $\mathcal{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \dots, \boldsymbol{a}_{31})$ and $\mathcal{B} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \dots, \boldsymbol{b}_{31})$, where $\boldsymbol{a}_1$ is the three dimensional location of the first sensor in pose $\mathcal{A}$, etc., the kernel value is given by

$$L(\mathcal{A}, \mathcal{B}) = \sum_{i=1}^{31} w_i \exp\left(-\frac{\|\boldsymbol{a}_i - \boldsymbol{b}_i\|_2^2}{2\sigma_i^2}\right) , \tag{3.27}$$

where σ_i^2 is the variance measured for sensor i, and $\boldsymbol{w} = (w_1, w_2, \dots, w_{31})$ is the importance weight vector. We chose a weight of 1 for the head, wrists, and ankles, a weight of 0.5 for the elbows and knees, and a weight of 0 for the remaining 22 sensors.

This kind of spatial kernel is natural for this task, where the items have inherent geometric relationships. However, because the feature representation is infinite-dimensional, it does not readily admit use of the dual methods of Kulesza and

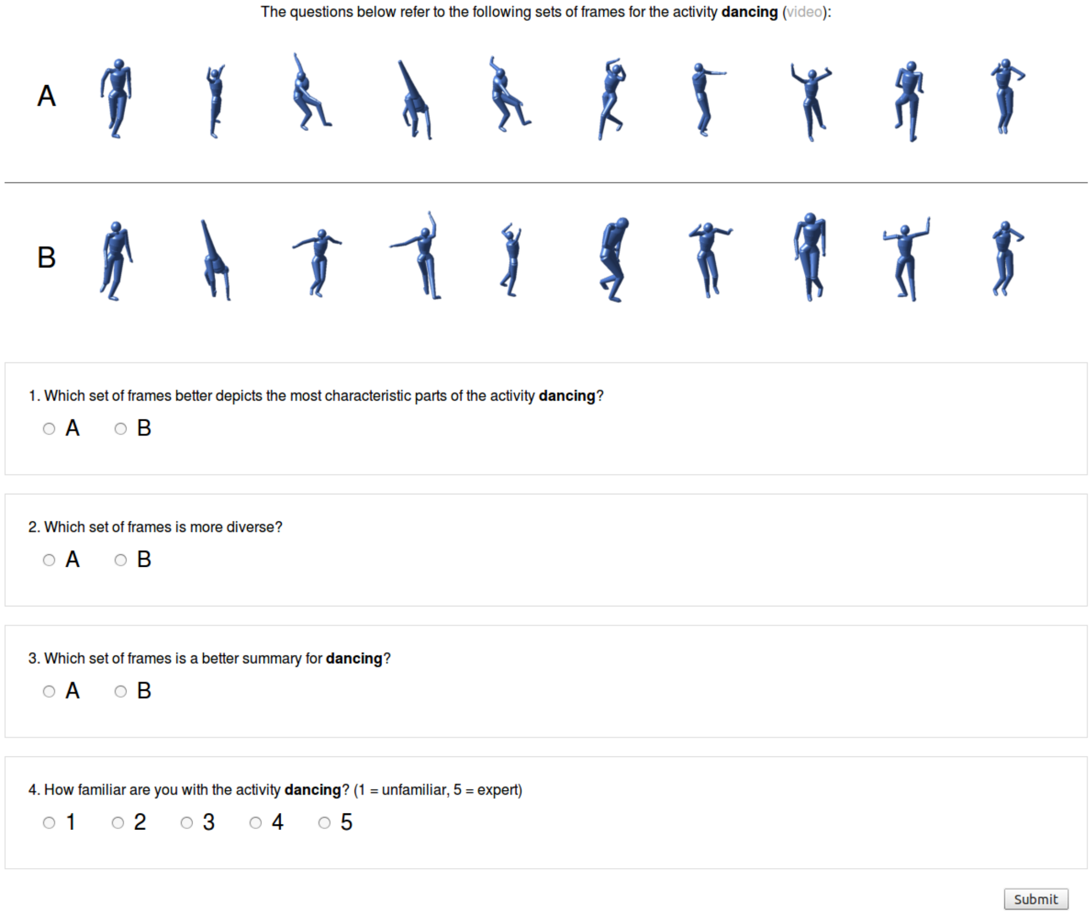
Taskar [2010]. Instead, we applied the stochastic adaptive Nyström approximation developed above, sampling a total of 200 landmark frames from each recording in 20 iterations (10 frames per iteration), bringing the intractable task of sampling from the high dimensional DPP down to an easily manageable size: sampling a set of ten summary frames from the longest recording took less than one second.

Of course, this speedup naturally comes at some approximation cost. In order to evaluate empirically whether the Nyström samples retained the advantages of the original DPP, which is too expensive for direct comparison, we performed a user study. Each subject in the study was shown, for each of the original nine recordings, a set of ten poses (rendered graphically) sampled from the approximated DPP model alongside a set of ten poses sampled uniformly at random (see Fig. 3.7). Fig. 3.6 shows motion capture summaries sampled from the Nyström-approximated kDPP (k=10). We asked the subjects to evaluate the two pose sets with respect to the motion capture recording, which was provided in the form of a rendered video. The subjects chose the set they felt better represented the characteristic poses from the video (quality), the set they felt was more diverse, and the set they felt made the better overall summary. The order of the two sets was randomized, and the samples were different for each user.

Fig. 3.5 shows a sample screen from our user study. Each subject completed four questions for each of the nine pairs of sets they saw (one pair for each of the nine activities). 18 subjects completed the study, for a total of 162 responses to each question. There was no significant correlation between a user's preference for the DPP set and their familiarity with the activity.

The results of the user study are shown in Table 3.1. Overall, the subjects felt that the samples from the Nyström-approximated DPP were significantly better on all three measures, $p < 0.001$. These results suggest that, while performing

DPP to select a diverse set of frames from a MoCap video is infeasible due to the large number of frames, sampling from an efficient low-rank approximated-DPP still result in a diverse collection. Furthermore, the study validates the notion that sampling a diverse collection of frames leads to better summaries of the activities potrayed in the MoCap recordings.



The questions below refer to the following sets of frames for the activity **dancing** (video):

1. Which set of frames better depicts the most characteristic parts of the activity **dancing**?

○ A ○ B

2. Which set of frames is more diverse?

○ A ○ B

3. Which set of frames is a better summary for **dancing**?

○ A ○ B

4. How familiar are you with the activity **dancing**? (1 = unfamiliar, 5 = expert)

○ 1 ○ 2 ○ 3 ○ 4 ○ 5

Figure 3.5: Sample screen from the user study.

3.3 Conclusion

Low-rank approximation of a kernel matrix is an appealing technique for managing the otherwise intractable task of sampling from high-dimensional DPPs. Given this

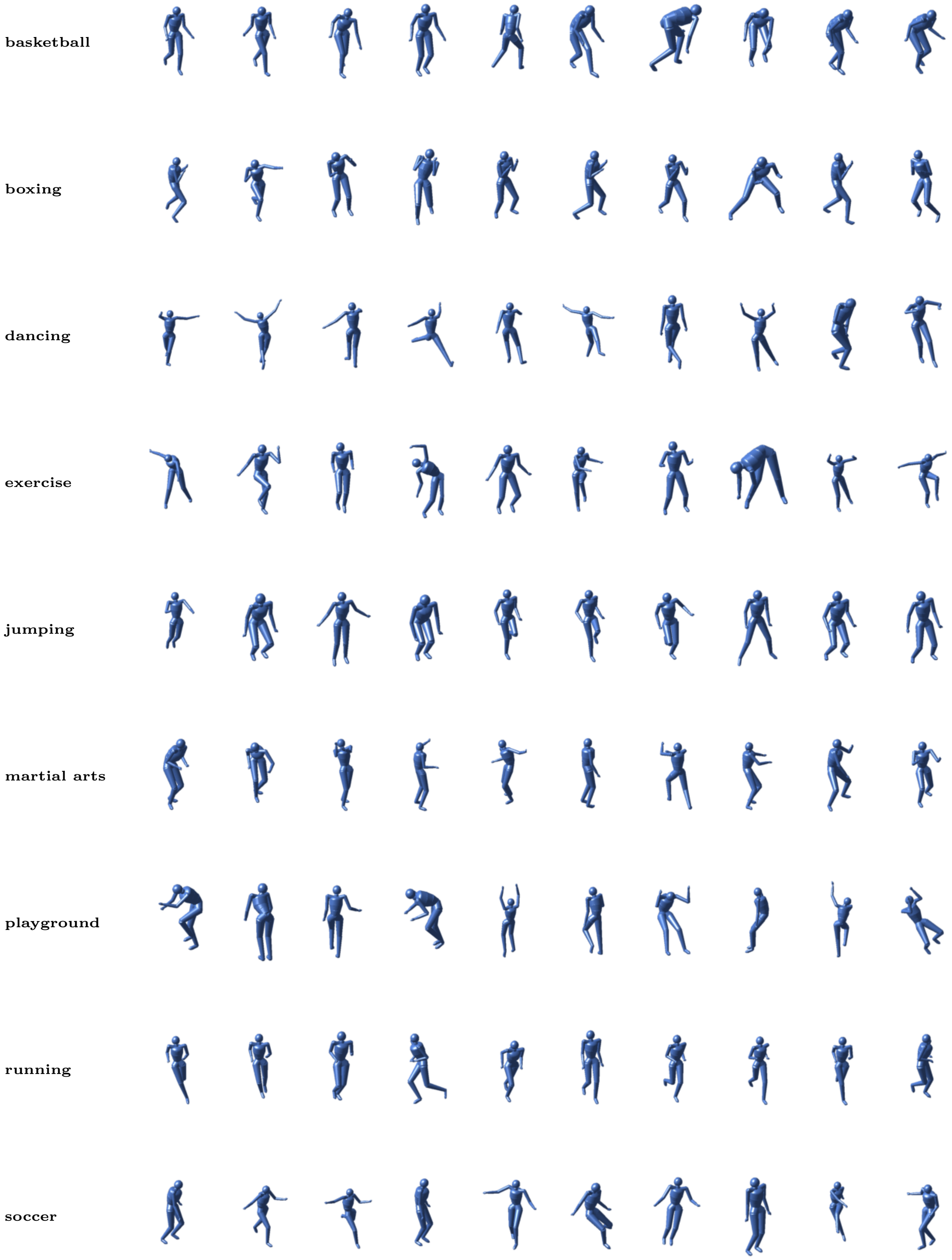


Figure 3.6: DPP samples ($k = 10$) for each activity.

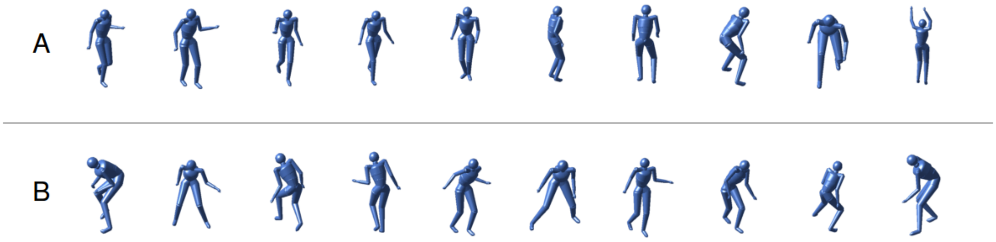


Figure 3.7: A sample pair of frame sets for the activity of basketball. The top set is chosen randomly, while the bottom is sampled from the Nyström-approximated DPP.

Evaluation measure	% DPP	% Random
Quality	66.7	33.3
Diversity	64.8	35.2
Overall	67.3	32.7

Table 3.1: The percentage of subjects choosing each method in a user study of motion capture summaries.

low-rank approximation, the unmanageable $O(N^3)$ task of sampling from DPPs on large N sets is reduced to $O(r^3)$, where r is the approximation rank. We showed that this appeal is theoretical as well as practical: we proved upper bounds for the variational error of low-rank-approximated DPPs for approximation methods that satisfy a few conditions (such as the Nyström method) and presented empirical results to validate them. We also demonstrated that Nyström-approximated DPPs can be usefully applied to the task of summarizing motion capture recordings.

Chapter 4

Inference for Continuous DPPs

Repulsive point processes, like *hard core processes* [Matérn, 1986, Daley and Vere-Jones, 2003] have a long history in the spatial statistics community, where considering continuous Ω is key. Many naturally occurring phenomena exhibit diversity—trees tend to grow in the least occupied space [Neeff et al., 2005], ant hill locations are over-dispersed relative to uniform placement [Bernstein and Gobbel, 1979] and the spatial distribution of nerve fibers is indicative of neuropathy, with hard-core processes providing a critical tool [Waller et al., 2011]. Repulsive processes on continuous spaces have garnered interest in machine learning as well, especially relating to generative mixture modeling [Zou and Adams, 2012, Petralia et al., 2012]. The main drawback for many repulsive point processes considered thus far, is the inefficient sampling algorithm and the lack of mathematical structure. For example, consider a well known example of a hard-core processes called the Matérn process [Matérn, 1986]. Here the sampling algorithm involves drawing a point one at a time, rejecting points that lie between a preset radius of a previously sampled point. Not only is this samping algorithm inefficient (since many points are potentially thrown away from the sample), the lack of mathematical properties associated with this process makes it difficult to incorporate it in many real world tasks. On the other hand, the computationally attractive properties of DPPs and their mathematical elegance as highlighted in Sec. 2.1 make them appealing to consider in these applications.

On the surface, it seems that the eigendecomposition and projection algorithm of Hough et al. [2006] for discrete DPPs, as highlighted in Sec. 2.1.2, would naturally extend to the continuous case. While this is true in a formal sense as L becomes an operator instead of a matrix, the key steps such as the eigendecomposition of the kernel and projection of points on subspaces spanned by *eigenfunctions* are computationally infeasible except in a few very limited cases where approximations can be made [Lavancier et al., 2012], as discussed in Section 2.2.2. The absence of a tractable DPP sampling algorithm for general kernels in continuous spaces has hindered progress in developing DPP-based models for repulsion.

In this chapter, we propose an efficient algorithm to sample from DPPs in continuous spaces using low-rank approximations of the kernel function. As in Chapter 3, we utilize the *dual representation* of the DPP (but here for continuous DPPs) and consider two low-rank approximation schemes: Nyström (highlighted in Sec. 2.3.1) and random Fourier features (highlighted in Sec. 2.3.2). For kDPPs, we also devise a Gibbs sampler that iteratively samples points in the k-set conditioned on all $k-1$ other points. The derivation relies on representing the conditional DPPs using the Schur complement of the kernel.

Our methods allow us to handle a broad range of typical kernels and continuous subspaces, provided certain simple integrals of the kernel function can be computed efficiently. Decomposing our kernel into *quality* and *similarity*:

$$L(\boldsymbol{x}, \boldsymbol{y}) = q(\boldsymbol{x}) k(\boldsymbol{x}, \boldsymbol{y}) q(\boldsymbol{y}) \tag{4.1}$$

where $q(\boldsymbol{x})$ represents the quality at point $\boldsymbol{x}$ and $k(\boldsymbol{x}, \boldsymbol{y})$ denotes the similarity between points $\boldsymbol{x}$ and $\boldsymbol{y}$, this includes, but is not limited to, all cases where the (i) spectral density of the quality and (ii) characteristic function of the similarity kernel can be computed efficiently. We provide a list of standard choices and their associated feasibilities for DPP sampling in Table 4.1. The list is by no means exhaustive, but is simply to provide some insight. We also elaborate upon some standard kernels in the following sections. Our methods scale well with dimension, in particular with complexity growing linearly in d.

Table 4.1: Examination of the feasibility of DPP sampling using Nyström, RFF approximations and Gibbs sampling for a few standard examples of quality functions q and similarity kernels k.

$q(x)$	$k(x,y)$	Method	
Gaussian, Laplacian	Gaussian, Laplacian	Nyström	✓
		RFF	✓
		Gibbs	✓
Gaussian, Laplacian	Cauchy	Nyström	?
		RFF	✓
		Gibbs	?
Cauchy	Gaussian, Laplacian	Nyström	?
		RFF	✓
		Gibbs	?
Cauchy	Cauchy	Nyström	?
		RFF	✓
		Gibbs	?
Gaussian, Laplacian	Linear, Polynomial	Nyström	✓
		RFF	X
		Gibbs	✓

We propose continuous DPP sampling algorithms based on low-rank kernel approximations in Sec. 4.1 and Gibbs sampling in Sec. 4.2. Theoretical and empirical analysis of the two schemes is provided in Sec. 4.3. Finally, we apply our methods to repulsive mixture modeling, repulsive latent social network modeling and human pose synthesis in Sec. 4.4.1, Sec. 4.4.2 and 4.4.3, respectively.

4.1 Sampling from a Low-Rank Continuous DPP

Recall that when Ω is discrete with cardinality N, the DPP sampling algorithm by Hough et al. [2006], as presented in Sec. 2.1.2, uses an eigendecomposition of the kernel matrix $L = \sum_{n=1}^{N} \lambda_n v_n v_n^\top$ and recursively samples points $\boldsymbol{x}_i$ as follows, resulting in a set $A \sim \mathrm{DPP}(L)$ with $A = \{\boldsymbol{x}_i\}$:

- Phase 1: Select eigenvector v_n with probability $\frac{\lambda_n}{\lambda_n+1}$. Let V be the selected eigenvectors ($k = |V|$).
- Phase 2: For $i = 1, \ldots, k$, sample points $\boldsymbol{x}_i \in \Omega$ sequentially with probability based on the projection of $\boldsymbol{x}_i$ onto the subspace spanned by V. Once $\boldsymbol{x}_i$ is sampled, update V by excluding the subspace spanned by the projection of $\boldsymbol{x}_i$ onto V.

When Ω is discrete, both steps are straightforward since the first phase involves eigendecomposing a kernel matrix and the second phase involves sampling from discrete probability distributions based on inner products between points and eigenvectors. Extending this algorithm to a continuous space was considered by Lavancier et al. [2012], but for a very limited set of kernels L and spaces Ω. We refer to Sec. 2.2.2 for details. For general L and Ω, we face difficulties in both phases. Extending Phase 1 to a continuous space requires knowledge of the eigendecomposition of the kernel function. When Ω is a compact rectangle in $\mathbb{R}^d$, Lavancier et al. [2012] suggest approximating the eigendecomposition using an orthonormal Fourier basis, as presented in Sec. 2.2.2.

Even if we are able to obtain the eigendecomposition of the kernel function (either directly or via approximations as considered in Lavancier et al. [2012] and in this section), we still need to implement Phase 2 of the sampling algorithm. Whereas the discrete case only requires sampling from a discrete probability function, here we have to sample from a probability density. When Ω is compact, Lavancier et al. [2012] suggest using a rejection sampler with a uniform proposal on Ω. The authors note that the acceptance rate of this rejection sampler decreases with the number of points sampled, making the method inefficient in sampling large sets from a DPP. In most other cases, implementing Phase 2 even via rejection sampling is infeasible since the target density is in general non-standard with unknown normalization. Furthermore, a generic proposal distribution can yield extremely low acceptance rates.

In summary, current algorithms can sample approximately from a continuous DPP only for translation-invariant kernels defined on a compact space. In this section, we propose a sampling algorithm that allows us to sample approximately from DPPs for a wide range of kernels L and spaces Ω.

Again considering Ω discrete with cardinality N, the sampling algorithm has complexity dominated by the eigendecomposition, $O(N^3)$. Again, as discussed in Sec. 2.1.5, if the kernel matrix L is low-rank, i.e. $L = B^\top B$, with B a $D \times N$ matrix and $D \ll N$, dual sampling allows us to sample efficiently from the DPP.

While the dependence on N in the dual is sharply reduced, in continuous spaces, N is infinite. In order to extend the algorithm, we must find efficient ways to compute C for Phase 1 and manipulate eigenfunctions implicitly for the projections in Phase 2. Generically, consider sampling from a DPP on a continuous space Ω with kernel $L(\mathbf{x}, \mathbf{y}) = \sum_{n=1}^{\infty} \lambda_n \phi_n(\mathbf{x}) \overline{\phi_n}(\mathbf{y})$,where λ_n and $\phi_n(\mathbf{x})$ are eigenvalues and eigenfunctions, and $\overline{\phi_n}(\mathbf{y})$ is the complex conjugate of $\phi_n(\mathbf{y})$. Assume that we can approximate L by a low-dimensional (generally complex-valued) mapping, $B(\mathbf{x}) : \Omega \mapsto \mathbb{C}^D$:

$$\tilde{L}(\mathbf{x}, \mathbf{y}) = B(\mathbf{x})^* B(\mathbf{y}) \text{ , where } B(\mathbf{x}) = [B_1(\mathbf{x}), \ldots, B_D(\mathbf{x})]^\top . \qquad (4.2)$$

Here, A^* denotes complex conjugate transpose of A. We consider two efficient low-rank approximation schemes in Sec. 4.1.2 and 4.1.1. Using such a low-rank representation, we propose an analog of the dual sampling algorithm for continuous spaces, described in Alg. 6. A similar algorithm provides samples from a k-DPP described is in Alg. 7.

Algorithm 6 Dual sampler for a low-rank continuous DPP

```
Input: $\tilde{L}(\mathbf{x},\mathbf{y}) = B(\mathbf{x})^* B(\mathbf{y})$,
        a rank-$D$ DPP kernel
PHASE 1
Compute $C = \int_\Omega B(\mathbf{x}) B(\mathbf{x})^* d\mathbf{x}$
Compute eigendecomp. $C = \sum_{k=1}^D \lambda_k \mathbf{v}_k \mathbf{v}_k^*$
$J \leftarrow \emptyset$
for $k = 1, \ldots, D$ do
    $J \leftarrow J \cup \{k\}$ with probability $\frac{\lambda_k}{\lambda_k + 1}$
$V \leftarrow \{\frac{v_k}{\sqrt{v_k^* C v_k}}\}_{k \in J}$
PHASE 2
$X \leftarrow \emptyset$
while $|V| > 0$ do
    Sample $\hat{\mathbf{x}}$ from $f(\mathbf{x}) = \frac{1}{|V|} \sum_{\mathbf{v} \in V} |\mathbf{v}^* B(\mathbf{x})|^2$
    $X \leftarrow X \cup \{\hat{\mathbf{x}}\}$
    Let $\mathbf{v}_0$ be a vector in $V$ such that $\mathbf{v}_0^* B(\hat{\mathbf{x}}) \neq 0$
    Update $V \leftarrow \{\mathbf{v} - \frac{\mathbf{v}^* B(\hat{\mathbf{x}})}{\mathbf{v}_0^* B(\hat{\mathbf{x}})} \mathbf{v}_0 \mid v \in V - \{v_0\}\}$
    Orthonormalize $V$ w.r.t. $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle = \mathbf{v}_1^* C \mathbf{v}_2$
Output: $X$
```

Algorithm 7 Dual sampler for a low-rank continuous kDPP

```
Input: $\tilde{L}(\mathbf{x},\mathbf{y}) = B(\mathbf{x})^* B(\mathbf{y})$,
        a rank-$D$ DPP kernel
PHASE 1
Compute $C = \int_\Omega B(\mathbf{x}) B(\mathbf{x})^* d\mathbf{x}$
$\{(\mathbf{v}_n, \lambda_n)\}_{n=1}^D \leftarrow$ eigendecomposition of $C$
for $n = D, \ldots, 1$ do
    if $u \sim U[0,1] < \lambda_n \frac{e_{k-1}^{n-1}}{e_k^n}$ then
        $J \leftarrow J \cup \{n\}$
        $k \leftarrow k - 1$
        if $k = 0$ then
            break
$V \leftarrow \{\frac{v_k}{\sqrt{v_k^* C v_k}}\}_{k \in J}$
PHASE 2
$X \leftarrow \emptyset$
while $|V| > 0$ do
    Sample $\hat{\mathbf{x}}$ from density $f(\mathbf{x}) = \frac{1}{|V|} \sum_{\mathbf{v} \in V} |\mathbf{v}^* B(\mathbf{x})|^2$
    $X \leftarrow X \cup \{\hat{\mathbf{x}}\}$
    Let $\mathbf{v}_0$ be a vector in $V$ such that $\mathbf{v}_0^* B(\hat{\mathbf{x}}) \neq 0$
    Update $V \leftarrow \{\mathbf{v} - \frac{\mathbf{v}^* B(\hat{\mathbf{x}})}{\mathbf{v}_0^* B(\hat{\mathbf{x}})} \mathbf{v}_0 \mid v \in V - \{v_0\}\}$
    Orthonormalize $V$ w.r.t. $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle = \mathbf{v}_1^* C \mathbf{v}_2$
Output: $X$
```

In this dual view, we still have the same two-phase structure, and must address two key challenges:

- Phase 1: Assuming a low-rank kernel function decomposition as in Eq. (4.2), we need to able to compute the dual kernel matrix, given by an integral:

$$C = \int_{\Omega} B(\mathbf{x})B(\mathbf{x})^* d\mathbf{x} \ . \tag{4.3}$$

- Phase 2: In general, sampling directly from the density $f(\mathbf{x})$ is difficult; instead, we can compute the cumulative distribution function (CDF) and sample $\mathbf{x}$ using the inverse CDF method Robert and Casella [2004]:

$$F(\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_d)) = \prod_{l=1}^{d} \int_{-\infty}^{\hat{x}_l} f(\mathbf{x}) 1_{\{x_l \in \Omega\}} dx_l. \tag{4.4}$$

Assuming (i) the kernel function $\tilde{L}$ is finite-rank and (ii) the terms C and $f(\mathbf{x})$ are computable, Alg. 6 provides exact samples from a DPP with kernel $\tilde{L}$. In what follows, approximations only arise from approximating general kernels L with low-rank kernels $\tilde{L}$. If given a finite-rank kernel L to begin with, the sampling procedure is exact.

One could imagine approximating L as in Eq. (4.2) by simply truncating the eigendecomposition (either directly or using numerical approximations). However, this simple approximation for known decompositions does not necessarily yield a tractable sampler, because the products of eigenfunctions required in Eq. (4.3) might not be efficiently integrable. For our approximation algorithm to work, not only do we need methods that approximate the kernel function well, but also that enable us to solve Eq. (4.3) and (4.4) directly for many different kernel functions. We consider two such approaches that enable an efficient sampler for a wide range of kernels: Nyström and random Fourier features. The feasibility of these methods for a few standard examples of kernels are shown in Table 4.1.

4.1.1 Sampling from a Nyström-approximated DPP

One method of low-rank kernel approximation is the Nyström method (see Sec. 2.3.1). Recall that in this method, a small number of *landmarks* are selected as the basis for the low-rank approximation of the kernel $L(\boldsymbol{x}, \boldsymbol{y})$. In the continuous space, given $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_D$ *landmarks* sampled from Ω, we can approximate the kernel function and dual matrix as,

$$\tilde{L}_{Nys}(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{D} \sum_{k=1}^{D} W_{jk}^2 L(\mathbf{x}, \mathbf{z}_j) L(\mathbf{z}_k, \mathbf{y}),$$

$$C_{jk}^{Nys} = \sum_{n=1}^{D} \sum_{m=1}^{D} W_{jn} W_{mk} \int_{\Omega} L(\mathbf{z}_n, \mathbf{x}) L(\mathbf{x}, \mathbf{z}_m) d\mathbf{x},$$

where $W_{jk} = L(\mathbf{z}_j, \mathbf{z}_k)^{-1/2}$. Denoting $\mathbf{w}_j(\mathbf{v}) = \sum_{n=1}^{D} W_{jn} v_n$, the CDF of $f(\mathbf{x})$ in Alg. 6 is:

$$F_{Nys}(\hat{\mathbf{x}}) = \frac{1}{|V|} \sum_{\mathbf{v} \in V} \sum_{j=1}^{D} \sum_{k=1}^{D} \mathbf{w}_j(\mathbf{v}) \mathbf{w}_k(\mathbf{v}) \prod_{l=1}^{d} \int_{-\infty}^{\hat{x}_l} L(\mathbf{x}, \mathbf{z}_j) L(\mathbf{z}_k, \mathbf{x}) 1_{\{x_l \in \Omega\}} dx_l. \quad (4.5)$$

We consider a decomposition $L(\mathbf{x}, \mathbf{y}) = q(\mathbf{x}) k(\mathbf{x}, \mathbf{y}) q(\mathbf{y})$. Here, we provide the important example where $\Omega = \mathbb{R}^d$ and both $q(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{y})$ are Gaussians and also when $k(\mathbf{x}, \mathbf{y})$ is polynomial, a case that cannot be handled by RFF since it is not translationally invariant.

Example: Sampling from a Nyström-approximated DPP with Gaussian quality and similarity

Assuming

$$q(\mathbf{x}) = \exp\left\{-\frac{1}{2}(\mathbf{x}-\mathbf{a})^\top \Gamma^{-1}(\mathbf{x}-\mathbf{a})\right\},$$
$$k(\mathbf{x},\mathbf{y}) = \exp\left\{-\frac{1}{2}(\mathbf{x}-\mathbf{y})^\top \Sigma^{-1}(\mathbf{x}-\mathbf{y})\right\},$$

the approximated kernel is given by

$$\begin{aligned}\tilde{L}_{Nys}(\mathbf{x},\mathbf{y}) = &\sum_{j=1}^{D}\sum_{k=1}^{D} W_{jk}^2 q(\mathbf{x})q(\mathbf{z}_j)q(\mathbf{z}_k)q(\mathbf{y}) \\ &\times \exp\left\{-\frac{1}{2}(\mathbf{x}-\mathbf{z}_j)^\top\Sigma^{-1}(\mathbf{x}-\mathbf{z}_j) - \frac{1}{2}(\mathbf{y}-\mathbf{z}_k)^\top\Sigma^{-1}(\mathbf{y}-\mathbf{z}_k)\right\}.\end{aligned}$$

Let $\Sigma^{-1} = Q\Lambda Q^\top$ with $\Lambda = \mathrm{diag}(\frac{1}{\sigma_1^2},\ldots,\frac{1}{\sigma_D^2})$, $\Gamma^{-1} = R\Delta R^\top$ with $\Delta = \mathrm{diag}(\frac{1}{\delta_1^2},\ldots,\frac{1}{\delta_D^2})$ and $(\Sigma^{-1}+\Gamma^{-1}) = T\Theta T^\top$ with $\Theta = \mathrm{diag}(\frac{1}{\theta_1^2},\ldots,\frac{1}{\theta_D^2})$. Furthermore, let $\tilde{\mathbf{z}}_j = T^\top(\Gamma^{-1}+\Sigma^{-1})\Sigma^{-1}\mathbf{z}_j$, $\tilde{\mathbf{a}} = T^\top(\Gamma^{-1}+\Sigma^{-1})\Gamma^{-1}\mathbf{a}$ and $\mathbf{y} = T^\top\mathbf{x}$. Then, the elements of the dual matrix C^{Nys} are then given by

$$C_{jk}^{Nys} = \sum_{m-1}^{D}\sum_{n=1}^{D} W_{jn}W_{mk}A_{mn}\prod_{l=1}^{d}\sqrt{\pi\theta_l^2}.$$

where

$$\begin{aligned}A_{mn} = \exp\Bigg\{ &-\frac{1}{2}(\mathbf{z}_n-\mathbf{a})^\top\Gamma^{-1}(\mathbf{z}_n-\mathbf{a}) - \frac{1}{2}(\mathbf{z}_m-\mathbf{a})^\top\Gamma^{-1}(\mathbf{z}_m-\mathbf{a}) - \frac{1}{2}\mathbf{z}_m^\top\Sigma^{-1}\mathbf{z}_m \\ &-\frac{1}{2}\mathbf{z}_n^\top\Sigma^{-1}\mathbf{z}_n + (\Gamma^{-1}\mathbf{a}+\Sigma^{-1}\frac{(\mathbf{z}_m+\mathbf{z}_n)}{2})^\top(\Sigma^{-1}+\Gamma^{-1})^{-1}(\Gamma^{-1}\mathbf{a}+\Sigma^{-1}\frac{(\mathbf{z}_m+\mathbf{z}_n)}{2}) \\ &-\mathbf{a}^\top\Gamma^{-1}\mathbf{a}\Bigg\}.\end{aligned}$$

Finally, the CDF of $f(\mathbf{y})$ is given by

$$F_{Nys}(\mathbf{y}) = \frac{1}{|V|} \sum_{\mathbf{v} \in V} \sum_{j,k=1}^{D} \mathbf{w}_j(\mathbf{v}) \mathbf{w}_k(\mathbf{v}) A_{jk} \prod_{l=1}^{d} \frac{\sqrt{\pi \theta_l^2}}{2} \left[1 - \text{erf} \left(\frac{2\tilde{a}_l + \tilde{z}_{jl} + \tilde{z}_{kl} - 2y_l}{2\sqrt{\theta_l^2}} \right) \right].$$

Once samples $\mathbf{y}$ are obtained, we transform back to our original coordinate system by letting $\mathbf{x} = T\mathbf{y}$.

Example: Sampling from a Nyström-approximated DPP with Gaussian quality and polynomial similarity

For simplicity of exposition, we consider a linear similarity kernel and $d = 1$, although the result can straightforwardly be extended to higher order polynomials and dimensions d. Assuming $q(x) = \exp\{-\frac{x^2}{2\rho^2}\}$ and $k(x,y) = xy$, the approximated kernel is given by

$$\tilde{L}_{Nys}(x,y) = \sum_{j=1}^{D} \sum_{k=1}^{D} W_{jk}^2 \exp\left\{ -\frac{(x^2 + z_j^2 + z_k^2 + y^2)}{2\rho^2} \right\} (xz_j)(yz_k).$$

The elements of the dual matrix C^{Nys} are then given by

$$C_{jk}^{Nys} = \sum_{m-1}^{D} \sum_{n=1}^{D} W_{jn} W_{mk} \frac{z_m z_n}{2} \exp\{-\frac{z_m^2 + z_n^2}{2\rho^2}\} \sqrt{\pi} \rho^3.$$

The CDF is given by

$$\begin{aligned} F_{Nys}(y) = \frac{1}{|V|} \sum_{\mathbf{v} \in V} \sum_{j,k=1}^{D} \mathbf{w}_j(\mathbf{v}) \mathbf{w}_k(\mathbf{v}) \frac{z_j z_k}{2} \\ \times \exp\{-\frac{z_j^2 + z_k^2}{2\rho^2}\} \left[\frac{\sqrt{\pi}\rho^3}{4} \left[\text{erf}\left(\frac{y}{\sqrt{r}} \right) + 1 \right] - 2y e^{-\frac{y^2}{\rho^2}} \right]. \end{aligned}$$

4.1.2 Sampling from an RFF-approximated DPP

Another low-rank approximation method is the random Fourier features (RFF), as presented in Sec. 2.3.2. Here, we require the similarity kernel $k(\boldsymbol{x}, y)$ be translational invariant (ie. $k(\boldsymbol{x}, \boldsymbol{y}) = k(\boldsymbol{x} - \boldsymbol{y})$). In this method, a low-rank approximation of the kernel is obtained by representing each point $\boldsymbol{x}$ using features that are sampled from a random direction $\boldsymbol{\omega}$. In particular, frequencies are sampled independently from the Fourier transform of the kernel function, $\boldsymbol{\omega}_j \sim \mathcal{F}(k(\mathbf{x} - \mathbf{y}))$, and letting:

$$\tilde{k}(\mathbf{x} - \mathbf{y}) = \frac{1}{D} \sum_{j=1}^{D} \exp\{i \boldsymbol{\omega}_j^\top (\mathbf{x} - \mathbf{y})\} \ , \qquad \mathbf{x}, \mathbf{y} \in \Omega \ . \tag{4.6}$$

To apply RFFs, we factor L into a quality function q and similarity kernel k (i.e., $q(\mathbf{x}) = \sqrt{L(\mathbf{x}, \mathbf{x})}$):

$$L(\mathbf{x}, \mathbf{y}) = q(\mathbf{x}) k(\mathbf{x}, \mathbf{y}) q(\mathbf{y}) \ , \qquad \mathbf{x}, \mathbf{y} \in \Omega \ \text{ where } k(\mathbf{x}, \mathbf{x}) = 1. \tag{4.7}$$

The RFF approximation can be applied to cases where the similarity function has a known characteristic function, e.g., Gaussian, Laplacian and Cauchy. Using Eq. (4.6), we can approximate the similarity kernel function to obtain a low-rank kernel and dual matrix:

$$\begin{aligned} \tilde{L}_{RFF}(\mathbf{x}, \mathbf{y}) &= \frac{1}{D} \sum_{j=1}^{D} q(\mathbf{x}) \exp\{i \boldsymbol{\omega}_j^\top (\mathbf{x} - \mathbf{y})\} q(\mathbf{y}) \\ C_{jk}^{RFF} &= \frac{1}{D} \int_{\Omega} q^2(\mathbf{x}) \exp\{i(\boldsymbol{\omega}_j - \boldsymbol{\omega}_k)^\top \mathbf{x}\} d\mathbf{x}. \end{aligned}$$

The CDF of the sampling distribution $f(\mathbf{x})$ in Algorithm 6 is given by:

$$F_{RFF}(\hat{\mathbf{x}}) = \frac{1}{|V|} \sum_{\mathbf{v} \in V} \sum_{j=1}^{D} \sum_{k=1}^{D} v_j v_k^* \prod_{l=1}^{d} \int_{-\infty}^{\hat{x}_l} q^2(\mathbf{x}) \exp\{i(\boldsymbol{\omega}_j - \boldsymbol{\omega}_k)^\top \mathbf{x}\} 1_{\{x_l \in \Omega\}} dx_l.$$

where v_j denotes the jth element of vector $\mathbf{v}$. Note that equations C^{RFF} and F_{RFF} can be computed for many different combinations of Ω and $q(\mathbf{x})$. In fact, this method works for any combination of (i) translation-invariant similarity kernel k with known characteristic function and (ii) quality function q with known spectral density. The resulting kernel L need not be translation invariant. We illustrate this method by considering a common and important example where $\Omega = \mathbb{R}^d$, $q(\mathbf{x})$ is Gaussian, and $k(\mathbf{x}, \mathbf{y})$ is any kernel with known Fourier transform.

Example: Sampling from an RFF-approximated DPP with Gaussian quality

Assuming $q(\mathbf{x}) = \exp\left\{-\frac{1}{2}(\mathbf{x}-\mathbf{a})^\top \Gamma^{-1}(\mathbf{x}-\mathbf{a})\right\}$ and $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x}-\mathbf{y})$ is given by a translation-invariant kernel with known characteristic function. We start by sampling $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_D \sim \mathcal{F}(k(\mathbf{x}-\mathbf{y}))$. Note, for example, that the Fourier transform of a Gaussian kernel is a Gaussian while that of the Laplacian is Cauchy and vice versa. The approximated kernel is given by

$$\tilde{L}_{RFF} = q(\mathbf{x}) \left[\frac{1}{D} \sum_{j=1}^{D} \exp i\boldsymbol{\omega}_j{}^\top (\mathbf{x}-\mathbf{y}) \right] q(\mathbf{y}),$$

where $q(\mathbf{x}) = \exp\left\{-\frac{1}{2}(\mathbf{x}-\mathbf{a})^\top \Gamma^{-1}(\mathbf{x}-\mathbf{a})\right\}$.

The elements of the dual matrix C^{RFF} are then given by

$$C_{jk}^{RFF} = \frac{1}{D} \int_{\mathbb{R}^d} \exp\{-(\mathbf{x}-\mathbf{a})^\top \Gamma^{-1}(\mathbf{x}-\mathbf{a}) + i(\boldsymbol{\omega}_j - \boldsymbol{\omega}_k)^\top \mathbf{x}\} d\mathbf{x}.$$

Letting $R\Delta R^\top$ be the spectral decompostition of Γ^{-1} with $\Delta = \operatorname{diag}(\frac{1}{\delta_1^2}, \ldots, \frac{1}{\delta_D^2})$, $\tilde{\boldsymbol{\omega}}_j = R^\top \boldsymbol{\omega}_j, \tilde{\mathbf{a}} = R^\top \mathbf{a}$ and $\mathbf{y} = R^\top \mathbf{x}$, one can straightforwardly derive:

$$C_{jk}^{RFF} = \frac{1}{D} \prod_{l=1}^{d} \left[\sqrt{\pi \delta_l^2} \exp\left\{ -\frac{\delta_l^2 (\tilde{\boldsymbol{\omega}}_{jl} - \tilde{\boldsymbol{\omega}}_{jk})^2}{4} \right\} + i\tilde{a}_l (\tilde{\boldsymbol{\omega}}_{jl} - \tilde{\boldsymbol{\omega}}_{jk}) \right].$$

Likewise,

$$F_{RFF}(\mathbf{y}) = \frac{1}{D|V|} \sum_{\mathbf{v} \in V} \sum_{j=1}^{D} \sum_{k=1}^{D} \mathbf{v}^{(j)} \mathbf{v}^{(k)*} \prod_{l=1}^{d} g(\tilde{\boldsymbol{\omega}}_{jl}, \tilde{\boldsymbol{\omega}}_{kl}, \tilde{a}_l, \delta_l, y_l),$$

where

$$\begin{aligned} g(\tilde{\boldsymbol{\omega}}_{jl}, \tilde{\boldsymbol{\omega}}_{kl}, \tilde{\mathbf{a}}_l, \delta_l, y_l) = \frac{1}{2}\sqrt{\pi\delta_l^2} \exp\left\{-\frac{\delta_l^2(\tilde{\boldsymbol{\omega}}_{jl} - \tilde{\boldsymbol{\omega}}_{kl})^2}{4}\right\} \\ + i\tilde{a}_l \left(\tilde{\boldsymbol{\omega}}_{jl} - \tilde{\boldsymbol{\omega}}_{kl})(1 - \operatorname{erf}\left(\frac{i\sqrt{\delta_l^2}(\tilde{\boldsymbol{\omega}}_{jl} - \tilde{\boldsymbol{\omega}}_{kl})}{2} - \frac{y_l - \tilde{a}_l}{2\sqrt{\delta_l^2}}\right)\right)\right). \end{aligned}$$

Once samples $\mathbf{y}$ are obtained, we transform back into our original coordinate system by letting $\mathbf{x} = R\mathbf{y}$.

4.2 Gibbs Sampling

Instead of the low-rank sampling algorithms presented in Section 4.1, for a kDPP, we can also consider a Gibbs sampling scheme by iteratively sampling a point conditioned on the location of the other $k-1$ points being fixed. Although this one-at-a-time move is far from optimal, we will show that it yields a tractable full conditional distribution for a wide range of kernel functions. More importantly, theory suggests that, asymptotically, we get *exact* (though correlated) samples from the kDPP.

The probability of choosing a specific k point configuration is given by

$$p(\{\mathbf{x}_j\}_{j=1}^k) \propto \det(L_{\{\mathbf{x}_j\}_{j=1}^k}). \tag{4.8}$$

Denoting $J^{\backslash k} = \{\mathbf{x}_j\}_{j \neq k}$ and $M^{\backslash k} = L_{J^{\backslash k}}^{-1}$, the Schur's determinantal identity formula yields

$$\det(L_{\{\mathbf{x}_j\}_{j=1}^k}) = \det(L_{J\backslash k}) \left(L(\mathbf{x}_k, \mathbf{x}_k) - \sum_{i,j \neq k} M_{ij}^{\backslash k} L(\mathbf{x}_i, \mathbf{x}_k) L(\mathbf{x}_j, \mathbf{x}_k) \right). \tag{4.9}$$

Conditioning on the inclusion of the other $k-1$ points, and suppressing constants not dependent on $\mathbf{x}_k$ we can now write the conditional distribution as

$$p(\mathbf{x}_k | \{\mathbf{x}_j\}_{j \neq k}) \propto L(\mathbf{x}_k, \mathbf{x}_k) - \sum_{i,j \neq k} M_{ij}^{\backslash k} L(\mathbf{x}_i, \mathbf{x}_k) L(\mathbf{x}_j, \mathbf{x}_k). \tag{4.10}$$

Normalizing and integrating this density yields a full conditional CDF given by

$$F(\hat{\mathbf{x}}_l | \{\mathbf{x}_j\}_{j \neq k}) = \frac{\int_{-\infty}^{\hat{\mathbf{x}}_l} L(\mathbf{x}_l, \mathbf{x}_l) - \sum_{i,j \neq k} M_{ij}^{\backslash k} L(\mathbf{x}_i, \mathbf{x}_l) L(\mathbf{x}_j, \mathbf{x}_l) 1_{\{\mathbf{x}_l \in \Omega\}} d\mathbf{x}_l}{\int_{\Omega} L(\mathbf{x}, \mathbf{x}) - \sum_{i,j \neq k} M_{ij}^{\backslash k} L(\mathbf{x}_i, \mathbf{x}) L(\mathbf{x}_j, \mathbf{x}) d\mathbf{x}}. \tag{4.11}$$

In general, sampling directly from this full conditional is difficult. However, for a wide range of kernel functions, including those which can be handled by the Nyström approximation in Sec. 4.1.1, the CDF can be computed analytically and $\mathbf{x}_k$ can be sampled using the inverse CDF method.

As with any Gibbs sampling scheme, the mixing rate is dependent on the correlations between variables. In cases where the kernel introduces low repulsion we expect the Gibbs sampler to mix well, while in a high repulsion setting the sampler can mix slowly due to the strong dependencies between points and fact that we are only doing one-point-at-a-time moves. We explore the dependence of convergence on repulsion strength in the next section. Regardless, this sampler provides a nice tool in the kDPP setting.

To extend this approach to standard DPPs, we can first sample k (this assumes knowledge of the eigenvalues of L) and then apply the above method to get a sample. This is fairly inefficient if many samples are needed. A more involved but potentially efficient approach is to consider a birth-death sampling scheme where the size of the set can grow/shrink by 1 at every step. A paper that is concurent to our work, Decreusefond et al. [2013] considered this approach using *coupling from the past.* However, their method only allow sampling of DPPs with stationary and compact kernels, akin to those that can be handled by Lavancier et al. [2012].

We illustrate this method by considering the case where $\Omega = \mathbb{R}^d$ and $q(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{y})$ are Gaussians. We use this same Schur complement scheme for sampling from the full conditionals in the mixture model application of Sec. 4.4.1. A key advantage of this scheme for several types of kernels is that the complexity of sampling scales linearly with the number of dimensions d making it suitable in handling high-dimensional spaces.

Example: Sampling from a DPP with Gaussian quality and similarity using Gibbs sampling

For generic kernels $L(\mathbf{x}, \mathbf{y}) = q(\mathbf{x})k(\mathbf{x}, \mathbf{y})q(\mathbf{y})$, we can rewrite the CDF $\mathbf{x}_k$ given $\{\mathbf{x}_j\}_{j \neq k}$ for a k-DPP as:

$$F(\hat{\mathbf{x}}_k | \{\mathbf{x}_j\}_{j \neq k}) = \frac{\int_{-\infty}^{\hat{\mathbf{x}}_k} q(\mathbf{x}_k)^2 (1 - \sum_{i,j \neq k} M_{ij} q(\mathbf{x}_i) q(\mathbf{x}_j) k(\mathbf{x}_k, \mathbf{x}_i) k(\mathbf{x}_j, \mathbf{x}_k)) 1_{\{\mathbf{x}_k \in \Omega\}} d\mathbf{x}_k}{\int_\Omega q(\mathbf{x})^2 (1 - \sum_{i,j \neq k} M_{ij} q(\mathbf{x}_i) q(\mathbf{x}_j) k(\mathbf{x}, \mathbf{x}_i) k(\mathbf{x}_j, \mathbf{x})) d\mathbf{x}}.$$

Assuming that

$$\begin{aligned} q(\mathbf{x}) &= \exp\left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{a})^\top \Gamma^{-1} (\mathbf{x} - \mathbf{a}) \right\} \\ k(\mathbf{x}, \mathbf{y}) &= \exp\left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{y})^\top \Sigma^{-1} (\mathbf{x} - \mathbf{y}) \right\}, \end{aligned}$$

the integrals above can be solved to yield

$$F(\hat{\mathbf{x}}_k | \{\mathbf{x}_j\}_{j \neq k}) =$$

$$\frac{\prod_{l=1}^{d} \left[\frac{\sqrt{\pi\delta_l^2}}{2} \left[1 - \operatorname{erf}\left(\frac{2\tilde{a}_l - 2x_{kl}}{2\sqrt{\delta_l^2}} \right) \right] - \sum_{i,j \neq k} M_{ij} A_{ij} \frac{\sqrt{\pi\theta_l^2}}{2} \left[1 - \operatorname{erf}\left(\frac{2\tilde{a}_l + \tilde{z}_{il} + \tilde{z}_{jl} - 2x_{kl}}{2\sqrt{\theta_l^2}} \right) \right] \right]}{\prod_{l=1}^{d} \left[\sqrt{\pi\delta_l^2} - \sum_{i,j \neq k} W_{ij} A_{ij} \sqrt{\pi\theta_l^2} \right]} .$$

where $\tilde{\mathbf{a}}, \tilde{\mathbf{z}}, \delta_l, A_{ij}$ and θ_l are as given in the Nyström sampling for Gaussian quality and similarity kernel.

4.3 Analysis

4.3.1 Analysis of Low-Rank Approximation Sampling

Here we derive approximation bounds for the low-rank approximated DPPs in continuous space. These bounds depend heavily on the following established eigenvalue lemmas on compact, self-adjoint operators:

Lemma 4 *[Fan, 1951] Let $T = \tilde{T} + S$, where $T, \tilde{T}$ and S are all self-adjoint compact operators with non-negative eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots$, $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \ldots$, and $\xi_1 \geq \xi 2 \geq \ldots$, respectively. Then*

$$\lambda_n \leq \tilde{\lambda}_m + \xi_{n-m+1} \quad \textit{for} \quad m \leq n \ . \tag{4.12}$$

Lemma 5 *[Gohberg and Kreĭ] Let $T = \tilde{T} + S$, where $T, \tilde{T}$ and S are all self-adjoint compact operators with non-negative eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots$, $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \ldots$, and $\xi_1 \geq \xi 2 \geq \ldots$, respectively. Then*

$$\lambda_n \geq \tilde{\lambda}_n \quad \textit{for all} \quad n \ . \tag{4.13}$$

As with the analysis of low-rank approximated-DPPs in the discrete settings in Sec. 3.1, we analyze the error of the variational distance by bounding the error in distribution of each possible subsets. We establish the bounds by considering the effect of low-rank approximation on the eigenvalues of the kernel, akin to the analysis in the discrete case. In what follows, we derive similar results as in Sec. 3.1 using Lemmas 4 and 5 in place of Corrolaries 1 and 2. The intuition here is that many results for positive semidefinite matrices carries over for compact self-adjoint operators.

Theorem 3 *Let $\lambda_1 \geq \lambda_2 \geq \ldots$ be the eigenvalues of a symmetric, positive semidefinite Hilbert-Schmidt kernel, $L(\boldsymbol{x}, \boldsymbol{y})$ and let $\tilde{L}(\boldsymbol{x}, \boldsymbol{y})$ be its approximation, $|A| = k$, and L_A has eigenvalues $\lambda_1^A \geq \ldots \geq \lambda_k^A$ Further assume that $\tilde{L}(\mathbf{x}, \mathbf{y})$ and $E(\boldsymbol{x}, \boldsymbol{y}) = L(\boldsymbol{x}, \boldsymbol{y}) - \tilde{L}(\boldsymbol{x}, \boldsymbol{y})$ are both positive semidefinite kernels.*

$$\hat{\lambda}_i = \max\left\{0, \lambda_i - \|T - \tilde{T}\|\right\}, \; \hat{\lambda}_i^A = \max\left\{0, \lambda_i^A - \|T - \tilde{T}\|\right\} \;, \tag{4.14}$$

where T and $\tilde{T}$ are Hilbert-Schmidt operators associated with kernels L and $\tilde{L}$, respectively and $\|.\|$ denotes the spectral operator. Then,

$$|\mathcal{P}_A - \tilde{\mathcal{P}}_A| \leq \mathcal{P}_A \max\left\{\left[1 - \frac{\prod_{i=1}^k \hat{\lambda}_i^A}{\prod_{i=1}^k \lambda_i^A}\right], \left[\frac{\prod_{i=1}^\infty (1 + \lambda_i)}{\prod_{i=1}^\infty (1 + \hat{\lambda}_i)} - 1\right]\right\} \;. \tag{4.15}$$

Proof $T, \tilde{T}$ and S be the Hilbert-Schmidt operator associated with kernel function $L(\boldsymbol{x}, \boldsymbol{y}), \tilde{L}(\boldsymbol{x}, \boldsymbol{y})$ and $E(\boldsymbol{x}, \boldsymbol{y}) = L(\boldsymbol{x}, \boldsymbol{y}) - \tilde{L}(\boldsymbol{x}, \boldsymbol{y})$ respectively. Note that if $L(\boldsymbol{x}, \boldsymbol{y}), \tilde{L}(\mathbf{x}, \mathbf{y})$ and $E(\boldsymbol{x}, \boldsymbol{y})$ are all positive semidefinite kernels, then $T, \tilde{T}$ and S are all self-adjoint compact operators. Thus Lemmas 4 and 5 applies. Then using Lemmas 4 and 5 in place of Corrolaries 1 and 2, we can proof Theorem 3 in a similar manner as Theorem 1.

Theorem 4 *Under the conditions of Theorem 3,*

$$|\mathcal{P}_A^k - \tilde{\mathcal{P}}_A^k| \leq \mathcal{P}_A^k \max\left\{ \left[\frac{e_k(\lambda_1, \ldots, \lambda_N)}{e_k(\hat{\lambda}_1, \ldots, \hat{\lambda}_N)} - 1 \right], \left[1 - \frac{\prod_{i=1}^k \hat{\lambda}_i^A}{\prod_{i=1}^k \lambda_i^A} \right] \right\} .$$

Proof Using Lemmas 4 and 5 in place of Corrolaries 1 and 2, we can proof Theorem 4 in a similar manner as Theorem 2.

4.3.2 Empirical Study of Low-Rank Approximation Sampling

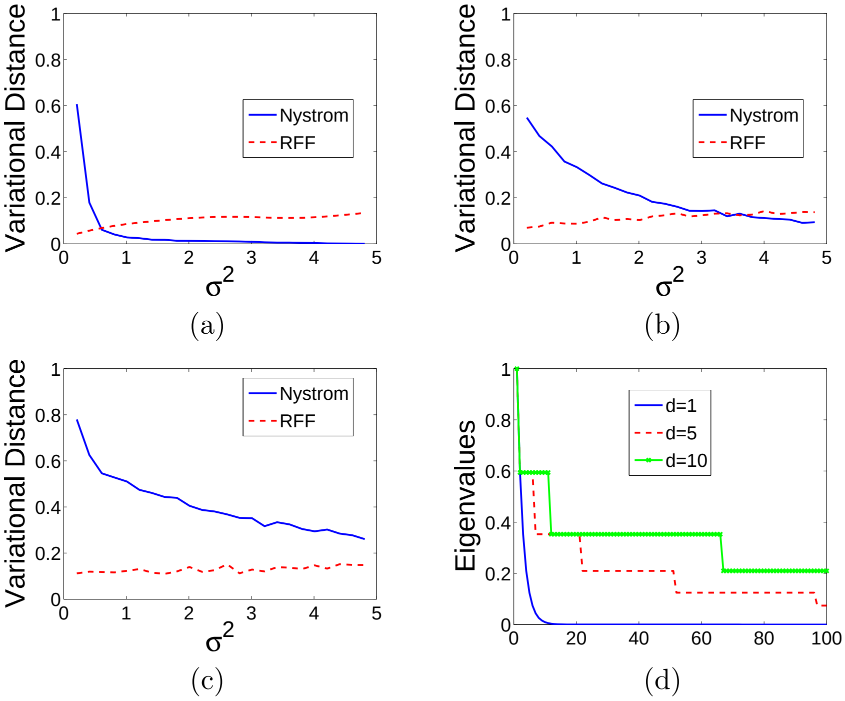
We also empirically evaluate the performance of the RFF and Nyström approximations. We compute the total variational distance

$$\|\mathcal{P}_L - \mathcal{P}_{\tilde{L}}\|_1 = \frac{1}{2} \sum_A |\mathcal{P}_L(A) - \mathcal{P}_{\tilde{L}}(A)| , \tag{4.16}$$

where $\mathcal{P}_L(A)$ denotes the probability of set X under a DPP with kernel L. One can show that the normalized density is $\mathcal{P}_L(A) = \frac{\det(L_A)}{\prod_{n=1}^{\infty}(1+\lambda_n(L))}$, which requires the eigenvalues of the kernel L. Thus, we restrict our analysis to the case where the quality function and similarity kernel are Gaussians with isotropic covariances $\Gamma = \text{diag}(\rho^2, \ldots, \rho^2)$ and $\Sigma = \text{diag}(\sigma^2, \ldots, \sigma^2)$, respectively, since the eigenvalues of the kernel is easily computable in this setting [Fasshauer and McCourt, 2012]. In this case, as presented in Sec. 2.2.1, letting $n = (n_1, \ldots, n_d)$ with $n_j \in \mathbb{Z}_+$, the eigenvalues (indexed by multi-index n) are given by:

$$\lambda_n = \prod_{j=1}^d \sqrt{\frac{\pi\rho^2}{\frac{\beta^2+1}{2} + \frac{1}{2\gamma}}} \left(\frac{1}{\gamma(\beta^2+1)+1} \right)^{n_j - 1} \tag{4.17}$$

where $\gamma = \frac{\sigma^2}{\rho^2}$ and $\beta = (1 + \frac{2}{\gamma})^{\frac{1}{4}}$. Since the eigenvalues are known in closed-form, we can estimate the total variation distance by sampling sets A from the approximated



(a)

(b)

(c)

(d)

Figure 4.1: Estimates of total variational distance for Nyström and RFF approximation methods to a DPP with Gaussian quality and similarity with covariances $\Gamma = \text{diag}(\rho^2, \ldots, \rho^2)$ and $\Sigma = \text{diag}(\sigma^2, \ldots, \sigma^2)$, respectively. (a)-(c) For dimensions d=1, 5 and 10, each plot considers $\rho^2 = 1$ and varies σ^2. (d) Eigenvalues for the Gaussian kernels with $\sigma^2 = \rho^2 = 1$ and varying dimension d.

DPP and calculating the absolute difference between $\mathcal{P}_L(A)$ and $\mathcal{P}_{\tilde{L}}(A)$.

Fig. 4.1 displays estimates of the total variational distance for the RFF and Nyström approximations when $\rho^2 = 1$, varying σ^2 (the repulsion strength) and the dimension d. Note that the RFF method performs slightly worse as σ^2 increases and is rather invariant to d while the Nyström method performs much better for increasing σ^2 but worse for increasing d.

While this phenomenon seems perplexing at first, a study of the eigenvalues of the Gaussian kernel across dimensions sheds light on the rationale (see Fig. 4.1). Note that for fixed σ^2 and ρ^2, the decay of eigenvalues is slower in higher dimensions. It has been previously demonstrated that the Nyström method performs favorably in kernel learning tasks compared to RFF in cases where there is a large eigengap in the kernel matrix [Yang et al., 2012]. The plot of the eigenvalues seems to indicate the same phenomenon here. Furthermore, this result is consistent with the comparison of RFF to Nyström in approximating DPPs in the discrete Ω case in Chapter 3.

This behavior can also be explained by looking at the theory behind these two approximations. For the RFF, while the kernel approximation is guaranteed to be an unbiased estimate of the true kernel element-wise, the variance is fairly high [Rahimi and Recht, 2007]. In our case, we note that the RFF estimates of minors are biased because of non-linearity in matrix entries, overestimating probabilities for point configurations that are more spread out, which leads to samples that are overly-dispersed. For the Nyström method, on the other hand, the quality of the approximation depends on how well the landmarks cover Ω. In our experiments the landmarks are sampled i.i.d. from $q(\mathbf{x})$. When either the similarity bandwidth σ^2 is small or the dimension d is high, the effective distance between points increases, thereby decreasing the accuracy of the approximation.

4.3.3 Empirical Study of Gibbs Sampling

To assess the mixing rate of the Gibbs sampling scheme, we ran the Gibbs sampler to sample points from a 1-dimensional 15-DPP with uniform quality and Gaussian similarity kernels in the space $\Omega = [-\frac{1}{2}, \frac{1}{2}]$. We perform this sampling under two values of repulsion parameter, $\sigma^2 = 0.01$ (high repulsion) and $\sigma^2 = 0.001$ (low repulsion). We ran 100 Gibbs chains, each of length 3000, discard the first 1500 samples as burn-in and thin every 15 iterations which we call cycles. Each cycle represents a full resampling of the set, having cycled through the past 15 points.

Fig. 4.2 (a)-(b) shows a visualization of the 15 points of the 15-DPPs. As an ordered set, we see qualitatively that the locations of the points are highly correlated from cycle to cycle in the high repulsion Gibbs samples while less correlation is observed in the low-repulsion counterpart.

Quantitatively, we use two measures as a proxy to the mixing rate: the average movement of point from cycle to cycle and the effective sample size. The average movement, m, is simply defined as the average difference in distance between points from one cycle to another averaged over the cycles:

$$m = \frac{1}{T-1}\frac{1}{k}\sum_{t=1}^{T-1}\sum_{i=1}^{k}(x_i^{t+1} - x_i^t)^2, \tag{4.18}$$

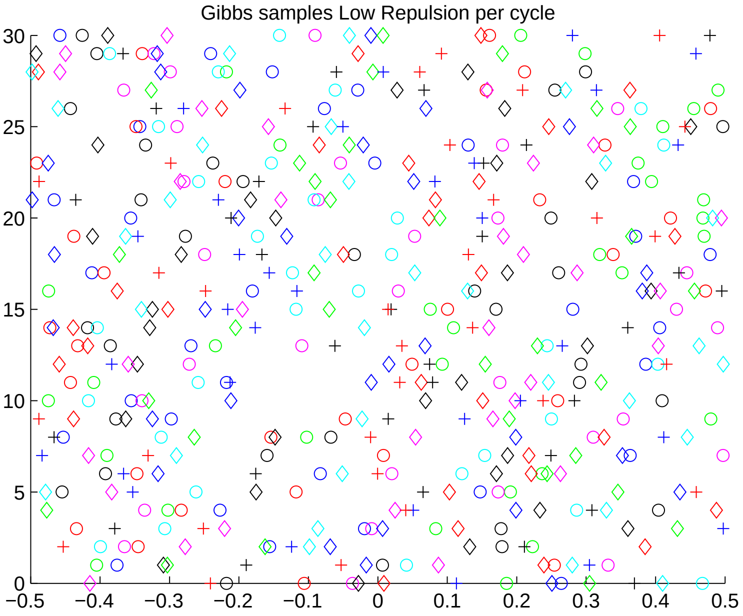
where T is the length of the chain after burn-in and thinning, k is the number of points and x_i^t is the coordinate of point x_i at cycle t. In our experiment, T and k are 100 and 15, respectively. When the Gibbs chain is mixing well, we expect the average movement to be high as this signals that the points are less correlated across cycles.

The effective sample size is a standard measure in assessing the mixing of a Gibbs chain. To compute this, we first compute the lag-s autocorrelation function of each point in the sampled sets. We then average the autocorrelation function at lag-s across the k points and denote this quantity $\bar{\rho}_s$. The effective sample size is then given by αT, where

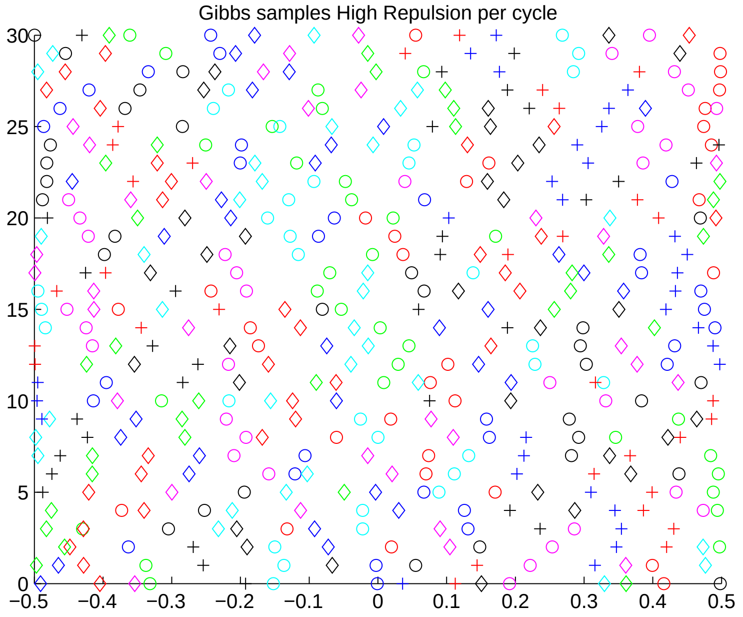
$$\alpha = \frac{1}{1 + 2\sum_{s=1}^{2\delta+1}\bar{\rho}_s}, \tag{4.19}$$

where δ is the smallest positive integer satisfying $\bar{\rho}_{2\delta} + \bar{\rho}_{2\delta+1} > 0$. In the case of i.i.d. samples, we expect α to be close to 1 while in cases where the mixing is bad, α will be much lower.

Table 4.2 shows the average values of m and α for our Gibbs samples with i.i.d. Nyström-approximated DPP samples serving as a benchmark. We see that



(a)



(b)

Figure 4.2: Visualization plots of location of 1-dimension DPP samples: (a)-(b) are samples from the Gibbs scheme in low repulsion and high repulsion settings, respectively.

in the low repulsion setting, the Gibbs chain mixes well with values close to the benchmarks while for the Gibbs sampler in the high repulsion setting, the values of m and α are much lower, indicating slow mixing.

	m	α
Gibbs High Repulsion	0.08 (0.07,0.08)	0.39 (0.31,0.45)
Gibbs Low Repulsion	0.1 (0.10,0.11)	0.92 (0.80,1)
Nyström High Repulsion	0.11 (0.1,0.11)	0.98 (0.82, 1)
Nyström Low Repulsion	0.11 (0.11,0.12)	0.98 (0.90, 1)

Table 4.2: The mean and 95% confidence interval for average movement, m and the effective sample size coefficient, α for Gibbs samples and i.i.d. Nyström samples in high and low repulsion settings.

4.4 Experiments

4.4.1 Repulsive priors for mixture models

Mixture models are used in a wide range of applications from clustering to density estimation. A common issue with such models, especially in density estimation tasks, is the introduction of redundant, overlapping components that increase the complexity and reduce interpretability of the resulting model. This phenomenon is especially prominent when the number of samples is small. In a Bayesian setting, a common fix to this problem is to consider a sparse Dirichlet prior on the mixture weights, which penalizes the addition of non-zero-weight components. However, such approaches run the risk of inaccuracies in the parameter estimates [Petralia et al., 2012]. Instead, Petralia et al. [2012] show that sampling the location parameters using repulsive priors leads to better separated clusters while maintaining the accuracy of the density estimate. They propose a class of repulsive priors that rely on explicitly defining a distance metric and the manner in which small distances are penalized. The resulting posterior computations can be fairly complex.

The theoretical properties of DPPs make them an appealing choice as a repulsive prior. In fact, Zou and Adams [2012] considered using DPPs as repulsive priors in latent variable models. However, in the absence of a feasible continuous DPP sampling algorithm, their method was restricted to performing MAP inference. Here we propose a fully generative probabilistic mixture model using a DPP prior for the location parameters, with a K-component model using a KDPP.

In the common case of mixtures of Gaussians (MoG), our posterior computations can be performed using Gibbs sampling with nearly the same simplicity of the standard case where the location parameters μ_k are assumed to be i.i.d.. In particular, with the exception of updating the location parameters $\{\mu_1, \ldots, \mu_K\}$, our sampling steps are identical to standard MoG Gibbs updates in the uncollapsed setting. For the location parameters, instead of sampling each μ_k independently from its conditional posterior, our full conditional depends upon the other locations $\mu_{\setminus k}$ as well. This full conditional has an interpretation as a single draw from a tilted 1-DPP. As such, we can employ the Gibbs sampling scheme of Sec. 4.2. We provide the details below.

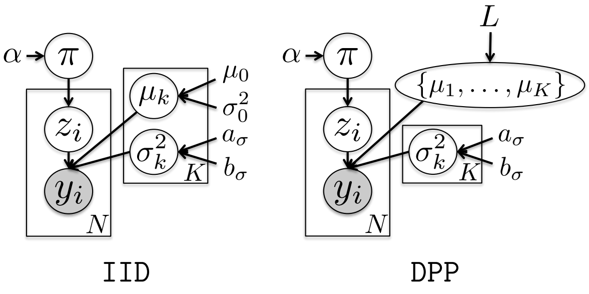


Figure 4.3: Graphical models for mixtures of Gaussians using IID and DPP priors on the location parameters.

Generative Model We consider a Bayesian mixture of Gaussians with either an independent normal (IID) or KDPP (DPP) prior on the location parameters.

In both cases, the K-component model with N observations is specified as:

$$\begin{aligned}
\pi \mid \alpha &\sim \mathrm{Dir}(\alpha, \dots, \alpha) \\
\sigma_k^2 \mid a_\sigma, b_\sigma &\sim \mathrm{IG}(a_\sigma, b_\sigma), \quad k = 1, \dots, K \\
\{\mu_1, \dots, \mu_K\} &\sim F \\
z_i \mid \pi &\sim \pi, \quad i = 1, \dots, N \\
y_i \mid \pi, \{\mu_k, \sigma_k^2\} &\sim N(\mu_{z_i}, \sigma_{z_i}^2), \quad i = 1, \dots, N.
\end{aligned} \tag{4.20}$$

Here, IG denotes the inverse gamma distribution and Dir a K-dimensional Dirichlet. For simplicity, we consider the univariate case here, though the multivariate case follows directly by considering an inverse Wishart prior in place of the inverse gamma and likewise modifying F accordingly. Such a multivariate case is examined in the *iris* classification example.

The difference between the models is in how the location parameters are specified. For the `IID` case, we simply have:

$$\mu_k \mid \mu_0, \sigma_0^2 \sim N(\mu_0, \sigma_0^2)\ . \tag{4.21}$$

For the `DPP` case, we jointly sample:

$$\{\mu_1, \dots, \mu_K\} \mid L \sim K\text{-DPP}(L). \tag{4.22}$$

We consider L decomposed into Gaussian quality and similarity terms:

$$L(\mu_m, \mu_n) = q(\mu_m) k(\mu_m, \mu_n) q(\mu_n), \tag{4.23}$$

Algorithm 8 Mixture of Gaussians sampler

```
Input: Previous mixture weights $\pi$, emission parameters $\{\mu_k, \sigma_k\}^2$.
for $i = 1, \dots, N$ do
    Sample cluster indicators $z_i \mid y_i, \{\mu_k, \sigma_k^2\}, \pi_k \propto \frac{1}{C_i} \sum_{k=1}^K \pi_k N(y_i; \mu_k, \sigma_k^2) \delta(z_i, k)$
Sample mixture weights $\pi \mid \{z_i\}, \alpha \sim \mathrm{Dir}(\alpha + N_1, \dots, \alpha + N_K)$
for $k = 1, \dots, K$ do
    Sample scale parameters $\sigma_k^2 \mid \{y_i : z_i = k\}, \mu_k, a_\sigma, b_\sigma$
    $\sim \mathrm{IG}\left(a_\sigma + \frac{N_k}{2}, b_\sigma + \frac{1}{2}\sum_{i:z_i=1}(y_i - \mu_k)^2\right)$
Sample location parameters $\{\mu_1, \dots, \mu_K\} \mid \{y_i\}, \{z_i\}, \{\sigma_k^2\} \sim F_{post}$
Output: New mixture weights $\pi$, emission parameters $\{\mu_k, \sigma_k^2\}$.
```

with

$$k(\mu_m, \mu_n) = \exp\left\{ -\frac{(\mu_m - \mu_n)^2}{\gamma_0^2} \right\}, \quad q(\mu_m) = N(\mu_0, 2\sigma_0^2). \tag{4.24}$$

Gibbs sampling For the uncollapsed setting, where mixture weights π and emission parameters $\{\mu_k, \sigma_k^2\}$ are sampled, Alg. 8 summarizes the Gibbs sampler for the finite mixture of Gaussians. We write the algorithm generically so that the overlap between `IID` and `DPP` is clear. In particular, the locations are sampled from F_{post}, which generically refers to the full conditional of the cluster means. For the `IID` case, we sample i.i.d. for each k from

$$\mu_k \mid \{y_i : z_i = k\}, \sigma_k^2, \mu_0, \sigma_0^2 \sim N\left(\hat{\mu}_k, \hat{\sigma}_k^2\right), \tag{4.25}$$

where $\hat{\mu}_k = \left(\frac{1}{\sigma_0^2} + \frac{N_k}{\sigma_k^2}\right)^{-1} \left(\frac{\mu_0}{\sigma_0^2} + \frac{1}{\sigma_k^2}\sum_{i:z_i=k} y_i\right)$ and $\hat{\sigma}_k^2 = \left(\frac{1}{\sigma_0^2} + \frac{N_k}{\sigma_k^2}\right)^{-1}$. Here, $N_k = |\{y_i : z_i = k\}|$, i.e., the cardinality of the set of observations assigned to cluster k.

For `DPP`, note that

$$p(\{\mu_j\}_{j=1}^k | \{y_i\}, \{z_i\}, \{\mu_k, \sigma_k^2\}) \propto \det(L_{\mu_1, \dots, \mu_k}) \prod_{j=1}^k \prod_{i:z_i=j} N(y_i; \mu_j, \sigma_j^2).$$

Unfortunately, this posterior distribution is not a k-DPP. However, fixing the rest of $k-1$ centroids, the full conditional of μ_k is (dropping constant terms that do not depend on μ_k)

$$p(\mu_k | \{y_i\}, \{z_i\}, \{\mu_j, \sigma_j^2\}_{j \neq k}, \sigma_k^2) \propto \det(L_{\mu_1,\ldots,\mu_k}) \prod_{i:z_i=k} N(y_i; \mu_k, \sigma_k^2). \tag{4.26}$$

As before, we can use Schur's determinantal equality [Schur, 1917] to get

$$\det(L_{\mu_1,\ldots,\mu_k}) \propto L(\mu_k, \mu_k) - \sum_{i,j \neq k} M_{ij}^{\setminus k} L(\mu_i, \mu_k) L(\mu_j, \mu_k) \tag{4.27}$$

$$= q^2(\mu_k) \left(1 - \sum_{i,j \neq k} M_{ij}^{\setminus k} q(\mu_i) k(\mu_i, \mu_k) k(\mu_j, \mu_k) q(\mu_j) \right). \tag{4.28}$$

Combining the previous two equations, we get the full conditional

$$p(\mu_k | \{y_i\}, \{z_i\}, \{\mu_j, \sigma_j^2\}_{j \neq k}, \sigma_k^2) \tag{4.29}$$

$$\propto q^2(\mu_k) \left(1 - \sum_{i,j \neq k} M_{ij}^{\setminus k} q(\mu_i) k(\mu_i, \mu_k) k(\mu_j, \mu_k) q(\mu_j) \right) \prod_{i:z_i=k} N(y_i; \mu_k, \sigma_k^2). \tag{4.30}$$

The CDF of the distribution above can be computed easily, since it only involves exponential quadratic forms. The inverse CDF method can then be used to obtain a sample from the above distribution. Note once again that $q^2(\mu_k) \prod_{i:z_i=k} N(y_i; \mu_k, \sigma_k^2)$ is defined to be exactly the same as the Gaussian distribution where μ_k would have been sampled from in the IID case. Thus the equation above gives a nice intuition on the conditional density of μ_k in the DPP setting: it is an exponentially tilted distribution in which $q^2(\mu_k) \prod_{i:z_i=k} N(y_i; \mu_k, \sigma_k^2)$ is corrected by a factor that depends on the location of the other centroids. In the case where all of the other centroids are far away from the cluster center $\hat{\mu}_k$, the correction factor is close to one and we would recover the density for the IID case.

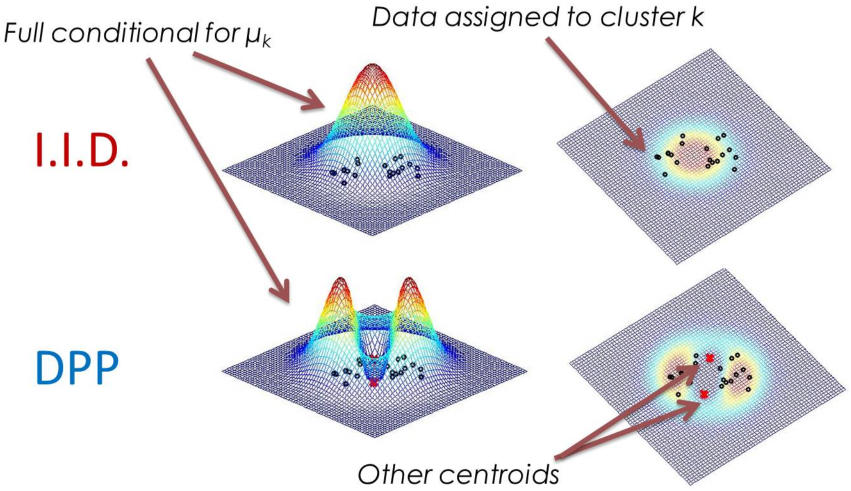


Figure 4.4: Comparison between the full conditional for μ_k using the IID and DPP models at a given iteration m of the sampler.

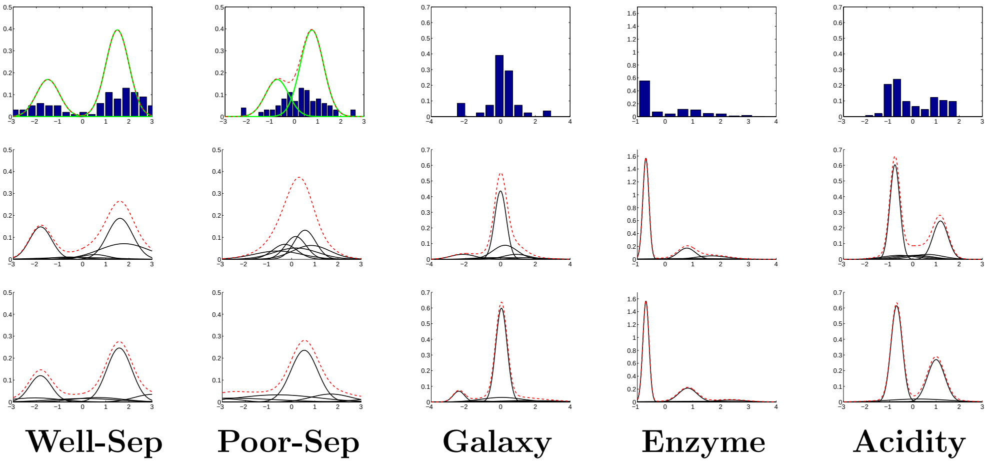
To get a sense of why the DPP leads to more diverse cluster centers than IID, consider the full conditional for μ_k at some iteration m of our sampler, as visualized in Fig. 4.4. We have some data points currently assigned to cluster k via cluster indicators $z_i = k$. The IID model assumes that μ_k is independent of the other μ_j's whereas the DPP conditions on the other cluster centers leading to a conditional distribution for μ_k that puts more mass on uncovered regions. In subsequent iterations, the data that had been assigned to cluster k but are not well covered by the sampled (and repulsed) μ_k will instead be assigned to one of the existing cluster centers that have mass near that data item. Such an alternative cluster exists, and is why μ_k was repulsed from that region, or will likely exist in future draws.

One attractive aspect of our DPP formulation is the fact that the sampling strategy maintains nearly the same simplicity as the standard IID sampler. This is in contrast, for example, to the repulsive mixture formulation of Petralia et al. [2012] which relied on slice sampling and draws from truncated normals, where the truncating region could only be computed in closed form for a restricted set of repulsive functions.

We assess the clustering and density estimation performance of the DPP-based model on both synthetic and real datasets. In each case, we run 10,000 Gibbs iterations, discard 5,000 as burn-in and thin the chain by 10. For our mixture of Gaussian experiments, we used an inverse-Wishart $\mathrm{IW}(\nu, \Psi)$ with $\nu = d + 1$ and $\Psi = I$, which corresponds to $a_\sigma = 2$ and $b_\sigma = 1$ for the inverse-Gamma in 1-dimension. Here, we use an inverse-Wishart specification such that $\Sigma \sim \mathrm{IW}(\nu, \Psi)$ has mean $E[\Sigma] = \frac{\Psi}{\nu - d + 1}$. The Dirichlet hyperparameters were set to $\alpha = \frac{1}{3}$, just as in Petralia et al. [2012]. For the location hyperparameters, in the `IID` case we set $\mu_0 = 0$ and $\sigma_0^2 = 1$. In the `DPP` case, we set $\mu_0 = 0$. After each Gibbs sampling iteration, we learn σ_0^2 and the repulsion parameter, ρ_0^2 by running an iteration of slice sampling discussed in Chapter 5.

We randomly permute the labels in each iteration to ensure balanced label switching. Draws are post-processed following the algorithm of Stephens [2000] to address the label switching issue.

Synthetic data To assess the role of the prior in a density estimation task, we generated a small sample of 100 observations from a mixture of two Gaussians. We consider two cases, the first with well-separated components and the second with poorly-separated components. We compare a mixture model with locations sampled i.i.d. (`IID`) to our DPP repulsive prior (`DPP`). In both cases, we set an upper bound of six mixture components. In Fig. 4.5, we see that both `IID` and `DPP` provide very similar density estimates. However, `IID` uses many large-mass components to describe the density. As a measure of simplicity of the resulting density description, we compute the average entropy of the posterior mixture membership distribution, which is a reasonable metric given the similarity of the



Well-Sep Poor-Sep Galaxy Enzyme Acidity

Figure 4.5: For each synthetic and real dataset: (top) histogram of data overlaid with actual Gaussian mixture generating the synthetic data, and posterior mean mixture model for (middle) IID and (bottom) DPP. Red dashed lines indicate resulting density estimate.

overall densities. Lower entropy indicates a more concise representation in an information-theoretic sense. We also assess the accuracy of the density estimate by computing both (i) Hamming distance error relative to true cluster labels and (ii) held-out log-likelihood on 100 observations. The results are summarized in Tables 4.3 and 4.4 . We see that DPP results in (i) significantly lower entropy, (ii) lower overall clustering error, and (iii) statistically indistinguishable held-out log-likelihood. This resulting lower entropy is especially true when the mixtures are poorly-separated since regular mixture models tend to infer overlapping clusters around the data. These results signify that we have a sparser representation with well-separated (interpretable) clusters while maintaining the accuracy of the density estimate.

Real data We also tested our DPP model on three real density estimation tasks considered in Richardson and Green [1997]: 82 measurements of velocity of galaxies diverging from our own (*galaxy*), acidity measurement of 155 lakes in Wisconsin (*acidity*), and the distribution of enzymatic activity in the blood of 245 individuals

Table 4.3: Mixture membership entropy and held-out log-likelihood for IID and DPP.

DATASET	ENTROPY		HELDOUT LOG-LIKE.	
	IID	DPP	IID	DPP
Well-separated	1.10 (0.3)	1.02 (0.2)	-169 (6)	-171(8)
Poorly-separated	1.45 (0.2)	0.75 (0.3)	-211(10)	-207(9)
Galaxy	0.91 (0.2)	0.70 (0.2)	-20(2)	-21(2)
Acidity	1.32 (0.1)	0.96 (0.1)	-49 (2)	-48(3)
Enzyme	1.01 (0.1)	0.95 (0.1)	-55(2)	-55(3)

(*enzyme*). We once again judge the complexity of the density estimates using the posterior mixture membership entropy as a proxy. To assess the accuracy of the density estimates, we performed 5-fold cross validation to estimate the predictive held-out log-likelihood. As with the synthetic data, we find that DPP visually results in better separated clusters (Fig. 4.5). The DPP entropy measure is also significantly lower for data that are not well separated (*acidity* and *galaxy*) while the differences in predictive log-likelihood estimates are not statistically significant (Table 4.3).

Finally, we consider a classification task based on the *iris* dataset: 150 observations from three iris species with four length measurements. For this dataset, there has been significant debate on the optimal number of clusters. While there are three species in the data, it is known that two have very low separation. Based on loss minimization, Sugar and James [2003], Wang [2010] concluded that the optimal number of clusters was two. Table 4.4 compares the classification error using DPP and IID when we assume for evaluation the real data has three or two classes (by collapsing two low-separation classes), but consider a model with a maximum of six components. While both methods perform similarly for three classes, DPP has significantly lower classification error under the assumption of two classes, since DPP places large posterior mass on only two mixture components. This result hints at the possibility of using the DPP mixture model as a model

Table 4.4: Classification error `IID` and `DPP`.

DATA	CLASSIFICATION ERROR	
	`IID`	`DPP`
Well-separated	0.19 (0.1)	0.15 (0.1)
Poorly-separated	0.51 (0.1)	0.39 (0.1)
Iris (3 cls)	0.43 (0.02)	0.43 (0.02)
Iris (2 cls)	0.23 (0.03)	0.17 (0.03)

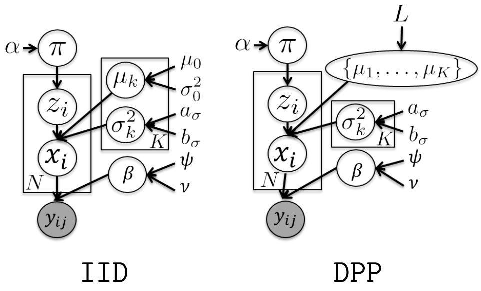
selection method.

4.4.2 Latent clustering of social networks

In social network models, there is often an interest in finding clusters of interacting actors when the number of groups in the data is unknown. Handcock et al. [2007] proposed a method in which ties between actors depends on their distances in an unobserved latent Euclidean space. The location of the actors in this latent space can be modeled by a mixture of distributions, each corresponding to a cluster. In particular, the K-component model with binary observations $y_{ij} \in \{0, 1\}$, denoting ties between N actors is specified as:

$$\begin{aligned}
\pi \mid \alpha &\sim \text{Dir}(\alpha, \dots, \alpha) \\
\sigma_k^2 \mid a_\sigma, b_\sigma &\sim \text{IG}(a_\sigma, b_\sigma), \quad k = 1, \dots, K \\
\{\mu_1, \dots, \mu_K\} &\sim F \\
z_i \mid \pi &\sim \pi, \quad i = 1, \dots, N \\
x_i \mid \pi, \{\mu_k, \sigma_k^2\} &\sim N(\mu_{z_i}, \sigma_{z_i}^2), \quad i = 1, \dots, N \\
\beta \mid \psi, \nu &\sim N(\psi, \nu) \\
y_{ij} \mid \beta, x_i, x_j &\sim \frac{e^{-\beta|x_i - x_j|}}{e^{-\beta|x_i - x_j| + 1}}.
\end{aligned} \tag{4.31}$$

Again, IG denotes the inverse-gamma distribution and Dir a K-dimensional Dirichlet. Here, $\{x_i\}$ denotes the location of the actors in the latent space and the ties, $\{y_{ij}\}$, between actors depends on their distances in this space. MCMC methods can then be used to estimate the mixture components. The number of groups can then be chosen by computing the BIC [Fraley and Raftery, 1998] and choosing the one with the highest value.



`IID` `DPP`

Figure 4.6: Graphical models for latent clustering of social network using `IID` and `DPP` priors on the location parameters.

In the case of the latent network clustering, posterior computations can be performed using Metropolis-Hastings and Gibbs sampling. With the exception of updating the location parameters $\{\mu_1, \ldots, \mu_K\}$, the steps are identical to the Metropolis-Hastings and Gibbs updates in Handcock et al. [2007]. Fig. 4.6 illustrates the associated graphical model. As in Sec. 4.4.1, instead of sampling each μ_k independently from a Gaussian posterior, we iteratively sample μ_k given the other locations $\mu_{\setminus k}$, variance, cluster indicators, and the location of the actors in the latent space. In each iteration, the hyperparameters for the DPP prior are inferred using the slice sampling method in Chapter 5.

We run our latent network clustering with DPP prior on the Sampson monk data [Sampson, 1969]. This data consists of 18 monks in an American monastery divided into three separate groups. Consistent with Handcock et al. [2007], we consider a monk having a tie to another monk if he ranked that monk in the top

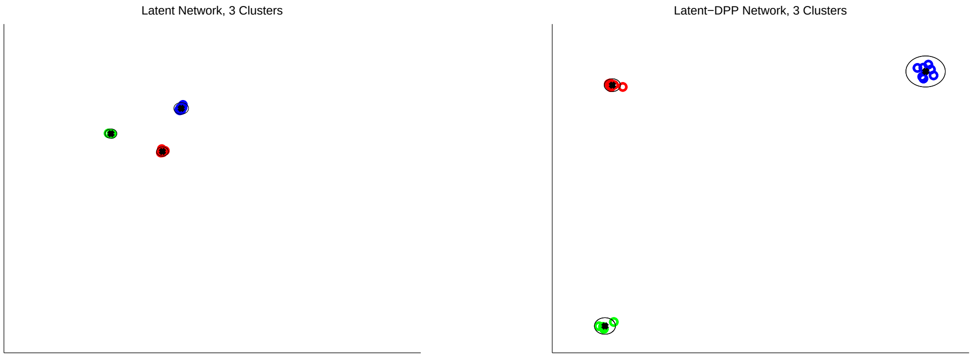


Figure 4.7: Social network clustering on Monk data with 3 clusters. Color indicates true clusters. Black cross indicates cluster center and black circle indicates the variance of the cluster.

three postive affect in of three interviews given over a 12-month period.

Figs. 4.7 and 4.8 show the plots of the posterior mean of the latent positions of the monks with I.I.D. and DPP priors under the assumptions of 3 and 5 mixture components. The color indicates the true labelling of the monk groups. Under the assumption of 3 mixture components, we see that with the DPP prior, the latent positions are farther separated than with the model with IID prior. In this case, however, both methods are able to accurately separate the three clusters with zero misclassification error as shown in Table 4.5. The real advantage of using the DPP prior is when the number of components is assumed to be higher than the actual number of groups. Under the assumption of 5 mixture components, we see that the model with DPP prior is still able to seperate the three groups. The model with the IID prior, on the other hand, is not able to do so since the latent positions often are assigned to different clusters in each iteration of the Gibbs sampling. As a result, the misclassification rate is much higher for the model with the IID prior. As before, this result hints at the possibility of using the DPP mixture model as a model selection method without the use of BIC or cross validation.

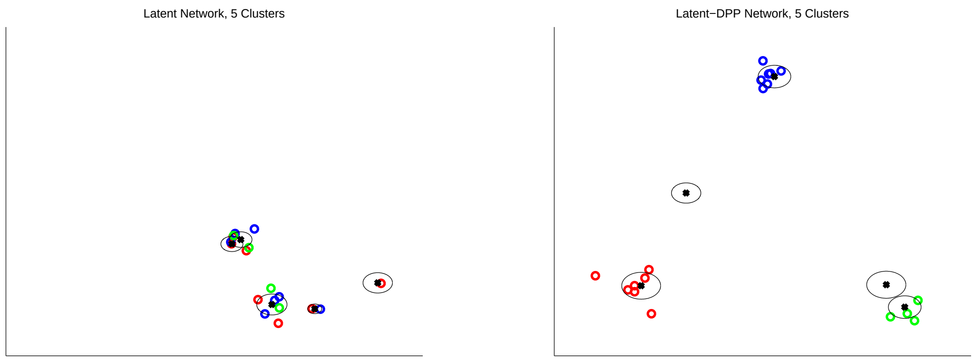


Figure 4.8: Social network clustering on Monk data with 5 clusters.Black cross indicates cluster center and black circle indicates the variance of the cluster.

Table 4.5: Median Misclassification rates for Monk Data. Brackets show the 25% and 75% quantile.

Method	IID(3 clust.)	DPP(3 clust.)	IID(5 clust.)	DPP(5 clust.)
Misclass	0 [0,0]	0 [0,0]	0.33 [0.28,0.33]	0.08 [0.08,1.5]

4.4.3 Generating diverse sample perturbations

We consider another possible application of continuous-space sampling. In many applications of inverse reinforcement learning or inverse optimal control, the learner is presented with control trajectories executed by an expert and tries to estimate a reward function that would approximately reproduce such policies [Abbeel and Ng, 2004]. In order to estimate the reward function, the learner needs to compare the rewards of a large set of trajectories (or all, if possible), which becomes intractable in high-dimensional spaces with complex non-linear dynamics. A typical approximation is to use a set of perturbed expert trajectories as a comparison set, where a good set of trajectories should cover as large a part of the space as possible.

We propose using DPPs to sample a large-coverage set of trajectories, in particular focusing on a human motion application where we assume a set of motion capture (MoCap) training data taken from the CMU database [CMU, 2009]. Here, our dimension d is 62, corresponding to a set of joint angle measurements. For a given activity, such as *dancing*, we aim to select a reference pose and synthesize a set of diverse, perturbed poses. To achieve this, we build a kernel with Gaussian quality and similarity using covariances estimated from the training data associated with the activity. In particular, we computed the covariance estimate from the training data, and set the similarity covariance parameter Σ equal to this estimate. We then take the quality covariance parameter to be $\Gamma = \frac{1}{2}\Sigma$. The Gaussian quality is centered about the selected reference pose and we synthesize new poses by sampling from our continuous DPP using the low-rank approximation scheme.

In Fig. 4.11, we show an example of such DPP-synthesized poses. For the activity *dance*, to quantitatively assess our performance in covering the activity space, we compute a *coverage rate* metric based on a random sample of 50 poses from a DPP. For each training MoCap frame, we compute whether the frame has a neighbor in the DPP sample within an ϵ-neighborhood in terms of the Euclidean distance. We compare our coverage to that of i.i.d. sampling from a multivariate Gaussian chosen to have variance matching our DPP sample. Despite favoring the i.i.d. case by inflating the variance to match the diverse DPP sample, the DPP poses still provide better average coverage over 100 runs. See Fig. 4.9 for an assessment of the coverage metric. A visualization of the samples is in the supplement. Note that the i.i.d. case requires on average $\epsilon = 253$ to cover all data whereas the DPP only requires $\epsilon = 82$. By $\epsilon = 40$, we cover over 90% of the data on average. Capturing the rare poses is extremely challenging with i.i.d. sampling, but the diversity encouraged by the DPP overcomes this issue.

In Fig. 4.10, we provide a visualization of poses sampled from the DPP relative to i.i.d. sampling of poses from a multivariate Gaussian. From these plots, we

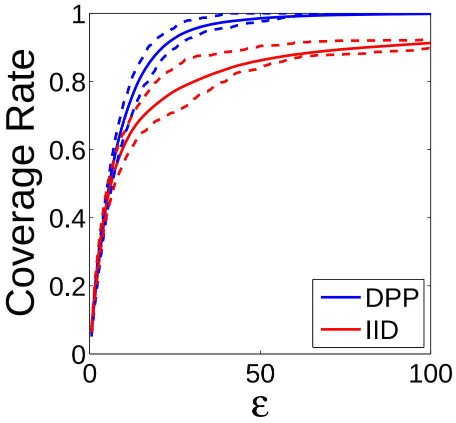
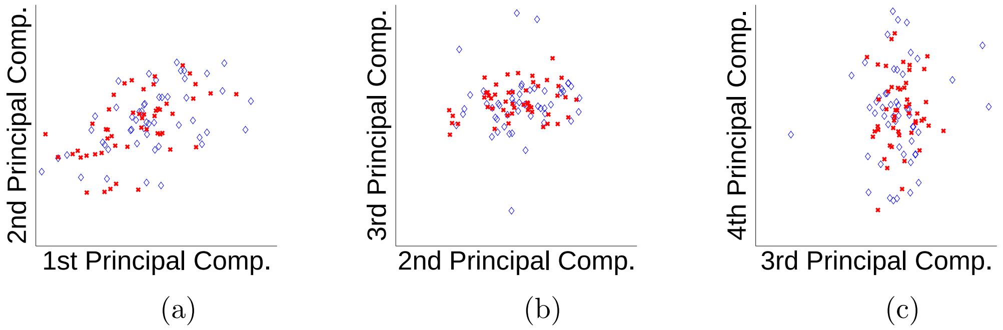


Figure 4.9: Fraction of data having a DPP/i.i.d. sample within an ϵ-neighborhood.

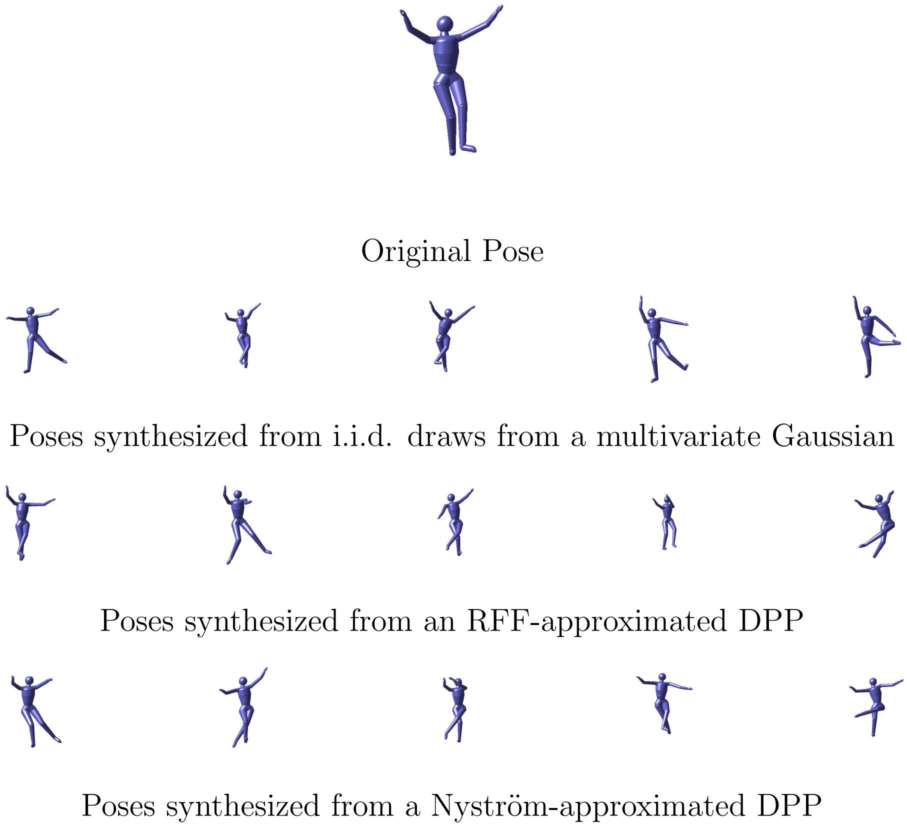


(a) (b) (c)

Figure 4.10: (a)-(c) DPP (blue) and i.i.d. multivariate Gaussian (red) samples projected onto the top 4 principal components of the *dance* data.

see how the sample of poses from the DPP covers a broader space, even when the covariance of the multivariate Gaussian is inflated to match that of the DPP. The reason for this broader coverage is the fact that the under the DPP, sampled poses repulse from regions already covered by other sampled poses.

Fig. 4.11 displays additional human poses that are drawn i.i.d. from a multivariate Gaussian, and compares to our DPP draws from both the RFF and Nyström approximations.



Original Pose

Poses synthesized from i.i.d. draws from a multivariate Gaussian

Poses synthesized from an RFF-approximated DPP

Poses synthesized from a Nyström-approximated DPP

Figure 4.11: Synthesizing perturbed human poses relative to an original pose by sampling (1) i.i.d. from a multivarite Gaussian versus (2) drawing a set from an RFF- or Nyström- approximated DPP with kernel based on MoCap data from the activity category. The Gaussian covariance is likewise formed from the activity data.

4.5 Conclusion

Motivated by the recent successes of DPP-based subset modeling in finite-set applications and the growing interest in repulsive processes on continuous spaces, we considered methods by which continuous-DPP sampling can be straightforwardly and efficiently approximated for a wide range of kernels. Our low-rank approach harnessed approximations provided by Nyström and random Fourier feature methods and then utilized a continuous dual DPP representation. The resulting approximate sampler garners the same efficiencies that led to the success of the DPP in the discrete case. For k-DPPs, we devised an exact Gibbs sampler that utilized the Schur complement representation. Finally, we demonstrated that continuous-DPP sampling is useful both for repulsive mixture modeling (which utilizes the Gibbs sampling scheme) and in synthesizing diverse human poses (which we demonstrated with the low-rank approximation method). As we saw in the MoCap example, we can handle high-dimensional spaces d, with our computations scaling just linearly with d. We believe this work opens up opportunities to use continuous DPPs as parts of many models.

Chapter 5

Large-Scale Learning of DPPs

While DPPs have many appealing properties, such as efficient sampling and marginal and conditional computation, 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 and Taskar, 2012a]. As discussed in Sec. 2.1.6, the issue arises from the fact that in seeking to maximize the log-likelihood of Eq. (2.1), the numerator yields a concave log-determinant term whereas the normalizer contributes a convex term, leading to a non-convex objective under various simplifying assumptions on the form of L.

In reference to Sec. 2.1.6, attempts to partially learn the kernel have been studied by, for example, learning the parametric form of the quality function $q(\boldsymbol{x})$ for fixed similarity $k(\boldsymbol{x}, \boldsymbol{y})$ [Kulesza and Taskar, 2011b], or learning a weighting on a fixed set of kernel experts [Kulesza and Taskar, 2011a]. So far, the only attempt to learn the parameters of the similarity kernel $k(\boldsymbol{x}, \boldsymbol{y})$ has used Nelder-Mead optimization [Lavancier et al., 2012], which lacks theoretical guarantees about convergence to a stationary point and is only exact in the discrete settings.

We consider parametric forms for the quality function $q(\boldsymbol{x})$ and similarity kernel $k(\boldsymbol{x}, \boldsymbol{y})$, as in Sec. 2.1.6 and propose Bayesian methods to learn the DPP kernel parameters Θ. In addition to capturing posterior uncertainty rather than a single point estimate, these methods can be easily modified to efficiently learn large-scale and continuous DPPs where the eigenstructures are either unknown or are inefficient to compute. In contrast, gradient ascent algorithms for maximum likelihood estimation (MLE) require kernels L that are differentiable with respect to Θ in the discrete Ω case. In the continuous Ω case, the eigenvalues must additionally have a known, differentiable functional form, which only occurs in limited scenarios.

We first explore likelihood maximization 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. 5.1, and explore modifications to accommodate learning large-scale and continuous DPPs. In Sec. 5.2, we derive a set of DPP moments assuming a known kernel eigenstructure and explore using these moments as a model-checking technique. In low-dimensional settings, we can use a method of moments approach to learn the kernel parameters via numerical techniques. Finally, we test our methods on both simulated and real-world data. Specifically, in Sec. 5.3 we use DPP learning to study the progression of diabetic neuropathy based on spatial distribution of nerve fibers and also to study human perception of diversity of images.

5.1 Learning Parametric DPPs

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

$$L(\mathbf{x}, \mathbf{y}; \Theta) = q(\mathbf{x}; \Theta) k(\mathbf{x}, \mathbf{y}; \Theta) q(\mathbf{y}; \Theta) \ , \tag{5.1}$$

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.

5.1.1 Learning using Optimization Methods

To learn the parameters Θ of a discrete DPP model, we can maximize the log-likelihood

$$\mathcal{L}(\Theta) = \sum_{t=1}^{T} \log\det(L_{A^t}(\Theta)) - T \log\det(L(\Theta) + I) \ . \tag{5.2}$$

Lavancier et al. [2012] suggest that the Nelder-Mead simplex algorithm [Nelder and Mead, 1965] can be used to maximize $\mathcal{L}(\Theta)$. This method is based on evaluating the objective function at the vertices of a simplex, then iteratively shrinking the simplex towards an optimal point. While this method is convenient since it does not require explicit knowledge of derivates of $\mathcal{L}(\Theta)$, it is regarded as a heuristic search method and is known for its failure to necessarily converge to a stationary point [McKinnon, 1998].

Gradient ascent and stochastic gradient ascent provide more attractive approaches because of their theoretical guarantees, but require knowledge of the gradient of $\mathcal{L}(\Theta)$:

$$\frac{d\mathcal{L}(\Theta)}{d\Theta} = \textstyle\sum_{t=1}^{T} \operatorname{tr}\left(L_{A^t}(\Theta)^{-1}\frac{dL_{A^t}(\Theta)}{d\Theta}\right) - T\operatorname{tr}\left((L(\Theta)+I)^{-1}\frac{dL(\Theta)}{d\Theta}\right) \ . \tag{5.3}$$

To find the MLE, we can perform gradient ascent

$$\Theta_i = \Theta_{i-1} + \eta \frac{d\mathcal{L}(\Theta)}{d\Theta} \ . \tag{5.4}$$

However, note that due to the non-convexity of our objective function, this can lead to convergence to local optima.

Below we provide a few examples. In the following examples, we denote $\boldsymbol{x}_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(d)})$, where d is the number of dimension.

Example: Gaussian Similarity with Uniform Quality

In this example, we consider the kernel

$$L(\Sigma) = \exp\{-(\boldsymbol{x}-\boldsymbol{y})^\top \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{y})\} \ .$$

In order to perform the gradient ascent algorithm in Eq. (5.4) we need to compute the gradient of the log-likelihood with respect to Σ.

Denote $G_{ij}^{(lm)} = L_{ij} \frac{(x_i^{(l)} - x_j^{(l)})(x_i^{(m)} - x_j^{(m)})}{2\Sigma_{lm}^2}$. Then,

$$\frac{d\mathcal{L}(\Sigma)}{d\Sigma_{lm}} = \sum_{t=1}^{T} \mathrm{tr}\left(L_{A^t}(\Sigma)^{-1} G_{A^t}^{(lm)}\right) - T\mathrm{tr}\left((L(\Sigma)+I)^{-1} G^{(lm)}\right)$$

Example: Gaussian Similarity with Gaussian Quality

Similarly, for the kernel

$$L(\Gamma, \Sigma) = \exp\{-\boldsymbol{x}^\top \Gamma^{-1} \boldsymbol{x} - (\boldsymbol{x}-\boldsymbol{y})^\top \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{y}) - \boldsymbol{y}^\top \Gamma^{-1} \boldsymbol{y}\},$$

the gradient log-likelihood with respect to $\Theta = (\Gamma, \Sigma)$ can easily be computed.

Denote $C_{ij}^{(lm)} = L_{ij} \frac{(x_i^{(l)} x_i^{(m)} + x_j^{(l)} x_j^{(m)})}{2\Gamma_{lm}^2}$ and $G_{ij}^{(lm)}$ as in previous example. Then,

$$\frac{d\mathcal{L}(\Gamma, \Sigma)}{d\Gamma_{lm}} = \sum_{t=1}^{T} \mathrm{tr}\left(L_{A^t}(\Sigma)^{-1} C_{A^t}^{(lm)}\right) - T\mathrm{tr}\left((L(\Sigma)+I)^{-1} C^{(lm)}\right)$$

and $\frac{dl(\Gamma,\Sigma)}{d\Sigma_{lm}}$ the same as the previous example.

Example: Polynomial Similarity with Uniform Quality

Finally we consider the polynomial similarity kernel

$$L(p, q) = \left(\boldsymbol{x}^\top \boldsymbol{y} + p\right)^q .$$

This kernel is important in the discrete setting for applications such as image search and text summarization where the similarity between items can be represented as the inner product of their feature vectors.

Denote $R_{ij} = qL_{ij}^{\frac{q-1}{q}}$ and $U_{ij} = L_{ij}\log(L_{ij}^{\frac{1}{q}})$. Then,

$$\frac{d\mathcal{L}(p,q)}{dp} = \sum_{t=1}^{T} \mathrm{tr}\left(L_{A^t}(p,q)^{-1}R_{A^t}\right) - T\mathrm{tr}\left((L(p,q)+I)^{-1}R)\right)$$

$$\frac{d\mathcal{L}(p,q)}{dq} = \sum_{t=1}^{T} \mathrm{tr}\left(L_{A^t}(p,q)^{-1}U_{A^t}\right) - T\mathrm{tr}\left((L(p,q)+I)^{-1}U)\right)$$

We note, however, that these methods are still susceptible to convergence to local optima due to the non-convex likelihood landscape. Furthermore, both these methods require that the likelihood is known exactly. When the number of base items N is large, computing the likelihood or its derivative will be inefficient. In Sec. 5.1.3, we will develop a method that requires only the upper and lower bounds on the likelihood.

The log likelihood of the k-DPP kernel parameter is

$$\mathcal{L}(\Theta) = \sum_{t=1}^{T} \log\det(L_{A^t}(\Theta)) - T\log\sum_{|B|=k}\det(L_B(\Theta)) \;, \tag{5.5}$$

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

In the case of continuous DPPs/kDPPs, once again, both methods require that the likelihood is known exactly. For kernel operators with infinite rank (such as the Gaussian), an explicit truncation has to be made since computing the exact likelihood is infeasible. Furthermore, gradient ascent can only be used in cases where the exact eigendecomposition of the kernel operator is known with a differentiable form for the eigenvalues (see Eq. (2.22)). This restricts the applicability of gradient-based likelihood maximization to a limited set of scenarios, such as a DPP with Gaussian quality function and similarity kernel. Even in these cases, an explicit truncation has to be made as well, resulting in an approximate gradient of $\mathcal{L}(\Theta)$. Unfortunately, such approximate gradients are not unbiased estimates of the true gradient, so the theory associated with attractive stochastic gradient based approaches does not hold. Table 5.1 summarizes the feasibility of using gradient-based methods to learn the parameters of DPP kernels.

Table 5.1: Examination of the feasibility of learning the parameters of DPP kernels using gradient-based methods.

DPP Setup	Eigendecomp.	Compute Normalizer	Gradient
Discrete(small N)	known	efficient	yes, for many kernels
Discrete(large N)	known	inefficient	approx. w/truncation
Cont.(Gaussian)	known	infeasible	approx. w/truncation
Cont.(most kernels)	unknown	infeasible	No

5.1.2 Bayesian Learning for Discrete DPPs

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

$$\mathbb{P}(\Theta|A^1,\dots,A^T) \propto \mathbb{P}(\Theta)\prod_{t=1}^{T}\frac{\det(L_{A^t}(\Theta))}{\det(L(\Theta)+I)} \tag{5.6}$$

for the DPP and, for the k-DPP,

$$\mathbb{P}(\Theta|A^1,\dots,A^T) \propto \mathbb{P}(\Theta)\prod_{t=1}^{T}\frac{\det(L_{A^t}(\Theta))}{e_k(\lambda_1(\Theta),\dots,\lambda_N(\Theta))}. \tag{5.7}$$

Here, $\mathbb{P}(\Theta)$ is the prior on Θ. Since neither Eq. (5.6) nor Eq. (5.7) yield a closed-form posterior, we resort to approximate techniques based on Markov chain Monte Carlo (MCMC) (see Sec. 2.4). We highlight two techniques: random-walk Metropolis-Hastings (MH) (see Sec. 2.4.1) and slice sampling (see Sec. 2.4.2), although other MCMC methods can be employed without loss of generality.

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

$$r = \left(\frac{\mathbb{P}(\hat{\Theta}|A^1,\dots,A^T)}{\mathbb{P}(\Theta_i|A^1,\dots,A^T)}\frac{f(\Theta_i|\hat{\Theta})}{f(\hat{\Theta}|\Theta_i)}\right). \tag{5.8}$$

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.

To avoid the need to tune the proposal distribution, we can instead use slice sampling [Neal, 2003], which performs a local search for an acceptable point while still satisfying detailed balance conditions. We refer to Section. 2.4.2 for further details.

Algorithm 9 Random-Walk Metropolis-Hastings

```
Input: Dimension: $D$, Starting point: $\Theta_0$, Prior distribution: $\mathbb{P}(\Theta)$, Proposal distribution $f(\hat{\Theta}|\Theta)$ with mean $\Theta$, Samples: $A^1, \dots, A^T]$.
$\Theta = \Theta_0$
for $i = 0 : (\tau - 1)$ do
  $\hat{\Theta} \sim f(\hat{\Theta}|\Theta_i)$
  $r = \left(\frac{\mathbb{P}(\hat{\Theta}|A^1,\dots,A^T)}{\mathbb{P}(\Theta_i|A^1,\dots,A^T)} \frac{f(\Theta_i|\hat{\Theta})}{f(\hat{\Theta}|\Theta_i)}\right)$ $r = \left(\frac{\prod_{t=1}^T \det(L_{X^t}(\hat{\Theta}))}{\prod_{t=1}^T \det(L_{X^t}(\Theta_i))} \big(\frac{\det(L(\Theta_i)+I)}{\det(L(\hat{\Theta})+I)}\big)^T \frac{f(\Theta_i|\hat{\Theta})}{f(\hat{\Theta}|\Theta_i)} \frac{p(\hat{\Theta};\Phi_0)}{p(\Theta_i;\Phi_0)}\right)$
  $u \sim$ Uniform[0,1]
  if $u < \min\{1, r\}$ then
    $\Theta_{i+1} = \hat{\Theta}$
Output: $\Theta_{0:\tau}$
```

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

$$q(\mathbf{x}_i) = \exp\left\{-\frac{1}{2}\mathbf{x}_i^\top \Gamma^{-1} \mathbf{x}_i\right\} \tag{5.9}$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^\top \Sigma^{-1} (\mathbf{x}_i - \mathbf{x}_j)\right\}, \tag{5.10}$$

where $\Gamma = \text{diag}(0.5, 0.5)$ and $\Sigma = \text{diag}(0.1, 0.2)$. We consider Ω to be a grid of 100 points evenly spaced in a 10×10 unit square and simulate 100 samples from a DPP with kernel as above. We then condition on these simulated data and perform posterior inference of the kernel parameters using MCMC. Fig. 5.1 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 and Rubin, 1992] on 5 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.

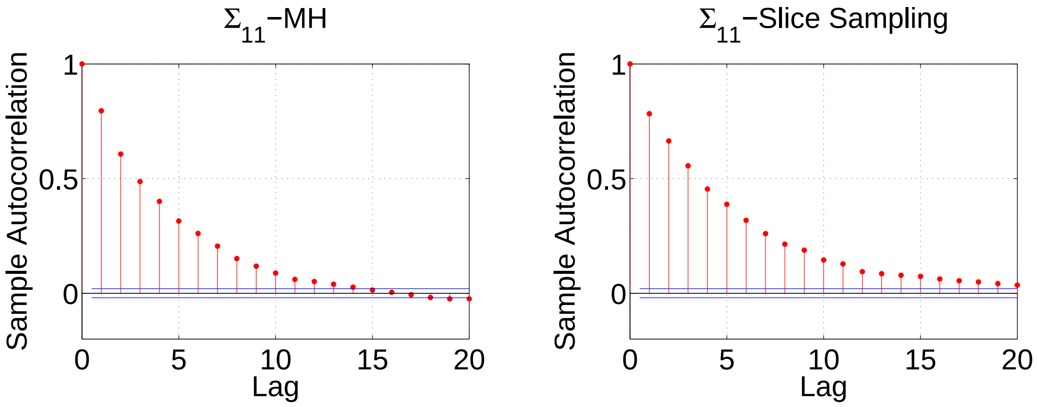


Figure 5.1: Sample autocorrelation function for posterior samples of the slowest mixing parameter of the kernel in Eq. (5.9) and Eq. (5.10) sampled using MH and slice sampling.

5.1.3 Bayesian Learning for Large-Scale Discrete and Continuous DPPs

In the large-scale discrete or continuous settings, 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. 5.1.1.

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. In particular, denote the upper and lower bounds by $\mathbb{P}^+(\Theta|A^1, \dots, A^T)$ and $\mathbb{P}^-(\Theta|A^1, \dots, A^T)$, respectively. In the random-walk MH algorithm we can

Algorithm 10 Random-Walk Metropolis-Hastings with Posterior Bounds

```
Input: Dimension: $D$, , Starting point: $\Theta_0$, Prior distribution: $\mathbb{P}(\Theta)$, Proposal distribution $f(\hat{\Theta}|\Theta)$ with mean $\Theta$, samples: $X = [X^1, \ldots, X^T]$.
$\Theta = \Theta_0$
for $i = 0 : \tau$ do
  $\hat{\Theta} \sim f(\hat{\Theta}|\Theta_i)$
  $r_+ = \infty, r_- = -\infty$
  $u \sim$ Uniform[0,1]
  while $u \in [r_-, r_+]$ do
    $r^+ = \left(\frac{\mathbb{P}^+(\hat{\Theta}|A^1,\ldots,A^T)}{\mathbb{P}^-(\Theta_i|A^1,\ldots,A^T)} \frac{f(\Theta_i|\hat{\Theta})}{f(\hat{\Theta}|\Theta_i)}\right)$
    $r^- = \left(\frac{\mathbb{P}^-(\hat{\Theta}|A^1,\ldots,A^T)}{\mathbb{P}^+(\Theta_i|A^1,\ldots,A^T)} \frac{f(\Theta_i|\hat{\Theta})}{f(\hat{\Theta}|\Theta_i)}\right)$
    Increase tightness on $\mathbb{P}^+$ and $\mathbb{P}^-$
  if $u < \min\{1, r^-\}$ then
    $\Theta_t = \hat{\Theta}$
Output: $\Theta_{0:\tau}$
```

then compute the upper and lower bounds on the acceptance ratio,

$$r^+ = \left(\frac{\mathbb{P}^+(\hat{\Theta}|A^1, \ldots, A^T)}{\mathbb{P}^-(\Theta_i|A^1, \ldots, A^T)} \frac{f(\Theta_i|\hat{\Theta})}{f(\hat{\Theta}|\Theta_i)}\right) \tag{5.11}$$

$$r^- = \left(\frac{\mathbb{P}^-(\hat{\Theta}|A^1, \ldots, A^T)}{\mathbb{P}^+(\Theta_i|A^1, \ldots, A^T)} \frac{f(\Theta_i|\hat{\Theta})}{f(\hat{\Theta}|\Theta_i)}\right) . \tag{5.12}$$

We can precompute the threshold $u \sim \text{Uniform}[0, 1]$, so we can still sometimes 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. 10.

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

$$\hat{y} \sim \text{Uniform}[0, \mathbb{P}^+(\Theta_i|A^1, \ldots, A^T)] \ . \tag{5.13}$$

We then decide whether $\hat{y} < \mathbb{P}^-(\Theta_i|A^1, \ldots, A^T)$, in which case we know $\hat{y} < \mathbb{P}(\Theta_i|A^1, \ldots, A^T)$ and we accept $\hat{y}$ as the slice and set $y = \hat{y}$. In the case where $\hat{y} \in (\mathbb{P}^-(\Theta_i|A^1, \ldots, A^T), \mathbb{P}^+(\Theta_i|A^1, \ldots, A^T))$, we keep increasing the tightness of our bounds until a decision can be made. If at any point $\hat{y}$ exceeds the newly computed $\mathbb{P}^+(\Theta_i|A^1, \ldots, A^T)$, we know that $\hat{y} > \mathbb{P}(\Theta_i|A^1, \ldots, A^T)$ 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 $y < \mathbb{P}^-(\Theta_e|A^1, \ldots, A^T)$ (endpoint is not in slice) or $y > \mathbb{P}^+(\Theta_e|A^1, \ldots, A^T)$ (endpoint is in slice). The tightness of the posterior bounds is increased until a decision can be made and the interval can be adjusted, if need be. After convergence of the interval, $\hat{\Theta}$ is generated uniformly over the interval and is likewise tested for acceptance. We illustrate this procedure in Fig. 5.2.

The lower and upper bounds of the posterior probability can in fact be incorporated in many MCMC-type algorithms. This provides a convenient and efficient way to garner posterior samples assuming that tightening the bounds can be done efficiently. In our case, the upper and lower bounds for the posterior probability can be computed either using eigenvalues truncation or low-rank approximations.

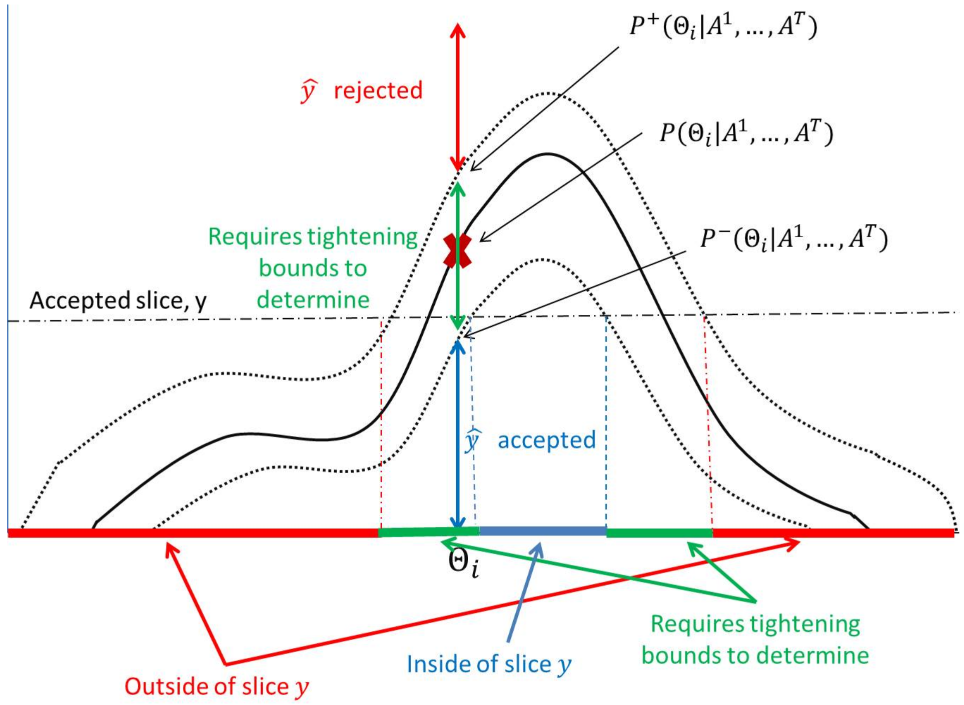

$P^{+}(\Theta_i|A^1,\ldots,A^T)$
$\hat{y}$ rejected
$P(\Theta_i|A^1,\ldots,A^T)$
Requires tightening bounds to determine
$P^{-}(\Theta_i|A^1,\ldots,A^T)$
Accepted slice, y
$\hat{y}$ accepted
Θ_i
Requires tightening bounds to determine
Inside of slice y
Outside of slice y

Figure 5.2: Illustration of slice sampling algorithm using posterior bounds. See Fig. 2.3 for the illustration for regular slice sampling method. In the first step, a candidate slice $\hat{y}$ is generated. $\hat{y}$ is rejected if it is above the upper posterior bound and rejected if it is below the lower posterior bound. If $\hat{y}$ is in between the bounds, then the bounds are tightened until a decision can be made. Once a slice, y is accepted, we need to sample new parameters inside the slice. To determine whether the endpoints of the interval or the new parameters are in the slice we decide that they are in the slice if the upper bound of posterior probability evaluated at the points are higher than the slice value and decide that they are outside of the slice if the lower bound of the posterior probability is lower than the slice value. Otherwise, we tighten the bounds until a decision can be made.

Exact Bayesian Learning with Eigenvalue Truncation

The upper and lower bounds for the posterior probability can be computed based on the truncation of the kernel eigenvalues and can be arbitrarily tightened by including more terms in the truncation. In the discrete DPP/k-DPP settings, the eigenvalues can be efficiently computed to a specified point using methods such as power law iterations. In the continuous setting, explicit truncation can be done when the kernel is Gaussian,

$$L(\boldsymbol{x}, \boldsymbol{y}) = \alpha \exp\left\{ -\frac{1}{2}\boldsymbol{x}^\top \Gamma^{-1} \boldsymbol{x} - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{y})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{y}) - \frac{1}{2}\boldsymbol{y}^\top \Gamma^{-1}\boldsymbol{y} \right\}, \quad (5.14)$$

$\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, since the eigenvalues are explicitly known [Fasshauer and McCourt, 2012].

Explicit forms for the posterior probability bounds of Θ for DPPs and k-DPPs as a function of the eigenvalue truncations follow from Prop. 1 and 2 combined with Eqs. (5.6) and (5.7), respectively.

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

$$\prod_{n=1}^{M}(1+\lambda_n) \leq \prod_{n=1}^{\infty}(1+\lambda_n) \quad (5.15)$$

and

$$\prod_{n=1}^{\infty}(1+\lambda_n) \leq \exp\left\{ \operatorname{tr}(L) - \sum_{n=1}^{M}\lambda_n \right\} \left[\prod_{n=1}^{M}(1+\lambda_n) \right]. \quad (5.16)$$

Proof The first inequality is trivial since the eigenvalues $\lambda_{1:\infty}$ are all nonnegative.

To prove the second inequality, we use the AM-GM inequality: For any nonnegative numbers, $\gamma_1, ..., \gamma_M$, $(\prod_{n=1}^{M}\gamma_n)^{\frac{1}{M}} \leq \sum_{n=1}^{M}\frac{\gamma_n}{M}$.

Let $\Lambda_M = \sum_{n=M+1}^{\infty} \lambda_n$ and $\gamma_n = 1 + \lambda_n$. Then,

$$\begin{aligned}
\prod_{n=1}^{\infty}(1+\lambda_n) &= \prod_{n=1}^{\infty} \gamma_n = (\prod_{n=1}^{M} \gamma_n)(\prod_{n=M+1}^{\infty} \gamma_n) \\
&= (\prod_{n=1}^{M} \gamma_n)(\lim_{l\to\infty} \prod_{n=M+1}^{M+l} \gamma_n) \\
&\leq (\prod_{n=1}^{M} \gamma_n)(\lim_{l\to\infty} (\sum_{n=M+1}^{M+l} \frac{\gamma_n}{l})^l) \\
&\leq (\prod_{n=1}^{M}(1+\lambda_n)) \exp(\Lambda_M) \ .
\end{aligned}$$

Proposition 2 *Let $\lambda_{1:\infty}$ be the eigenvalues of kernel L. Then*

$$e_k(\lambda_{1:M}) \leq e_k(\lambda_{1:\infty}) \tag{5.17}$$

and

$$e_k(\lambda_{1:\infty}) \leq \sum_{j=0}^{k} \frac{(\mathrm{tr}(L) - \sum_{n=1}^{M} \lambda_n)^j}{j!} e_{k-j}(\lambda_{1:M}) \ . \tag{5.18}$$

Proof Let $e_k(\lambda_{1:m})$ be the kth elementary symmetric function: $e_k(\lambda_{1:m}) = \sum_{J\subseteq\{1,\ldots,m\},|J|=k} \prod_{j\in J} \lambda_j$.

Trivially, we have a lower bound since the eigenvalues $\lambda_{1:\infty}$ are non-negative: $e_k(\lambda_{1:m}) \leq e_k(\lambda_{1:n}) \quad \text{for } m \leq n$.

For the upper bound we can use the Schur-concavity of elementary symmetric functions for non-negative arguments [Guan, 2006].Thus for $\bar{\lambda}_{1:N} \prec \lambda_{1:N}$:

$$\sum_{i=1}^{k} \bar{\lambda}_n \leq \sum_{n=1}^{k} \lambda_n \quad \text{for } k = 1, \ldots, N-1 \tag{5.19}$$

and

$$\sum_{n=1}^{N} \bar{\lambda}_n = \sum_{n=1}^{N} \lambda_n \ , \tag{5.20}$$

we have $e_k(\bar{\lambda}_{1:N}) \geq e_k(\lambda_{1:N})$.

Now let $\Lambda_M = \sum_{n=M+1}^{\infty} \lambda_n$ and $\Lambda_M^N = \sum_{n=M+1}^{N} \lambda_n$. We consider

$$\bar{\lambda}_{1:N}^{(M)} \equiv \left(\lambda_1, \ldots, \lambda_M, \frac{\Lambda_M^N}{N-M}, \ldots, \frac{\Lambda_M^N}{N-M} \right) . \tag{5.21}$$

Note that $\bar{\lambda}_{1:N}^{(M)} \prec \lambda_{1:N}$ and so $e_k(\bar{\lambda}_{1:N}^{(M)}) \geq e_k(\lambda_{1:N})$ for $M < N$.

We now compute $e_k(\bar{\lambda}_{1:N}^{(M)})$. Note that for $e_k(\bar{\lambda}_{1:N}^{(M)})$, the terms in the sum are products of k factors, each containing some of the $\lambda_{1:M}$ factors and some of the $\frac{\Lambda_M^N}{N-M}$ factors. The sum of the terms that have j factors of type $\frac{\Lambda_M^N}{N-M}$ is $\binom{N-M}{j} \left(\frac{\Lambda_M^N}{N-M}\right)^j e_{k-j}(\Lambda(m))$, so we have:

$$e_k(\bar{\lambda}_{1:N}^{(M)}) = \sum_{j=0}^{k} \binom{N-M}{j} \left(\frac{\Lambda_M^N}{N-M} \right)^j e_{k-j}(\lambda_{1:M}) \ .$$

Using $\binom{N-M}{j} \leq \frac{(N-M)^j}{j!}$, we get

$$e_k(\bar{\lambda}_{1:N}^{(M)}) = \sum_{j=0}^{k} \left(\frac{(\Lambda_M^N)^j}{j!} \right) e_{k-j}(\lambda_{1:M}) \ .$$

Letting $N \to \infty$, we get out upper bound

$$e_k(\lambda_{1:\infty}) \leq \sum_{j=0}^{k} \left(\frac{(\Lambda_M)^j}{j!} \right) e_{k-j}(\lambda_{1:M}) \quad \text{for } m \leq n.$$

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

The corresponding bounds for a 3600 × 3600 discrete Gaussian kernel example are shown in Fig. 5.3.

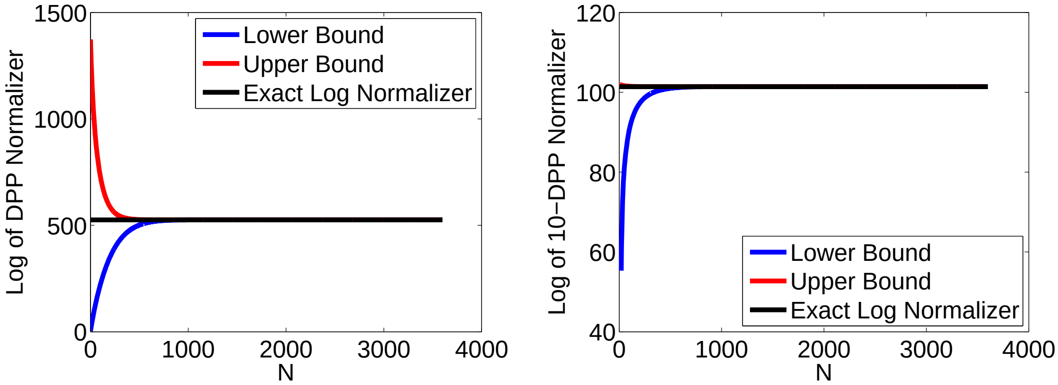


Figure 5.3: Normalizer bounds for a discrete DPP (*left*) and a 10-DPP (*right*) with Gaussian quality and similarity as in Eqs. (5.9) and (5.10) and Ω a grid of 3600 points.

Exact Bayesian Learning with Low-Rank Approximations

The algorithms developed in this section relies heavily on the ability to not only calculate the upper and lower bounds on the posterior probability but to arbitrarily tighten those bounds as well. For many continuous DPP kernels, the exact eigendecomposition is unknown. While approximate eigenvalues can be computed, for example using Fourier bases as illustrated in Sec. 2.2.1, this will only lead to an approximate posterior computation. Furthermore, the associated approximation bounds cannot be arbitrarily tightened.

Here we propose using low-rank approximations to sample from the exact posterior distribution. The advantage we have here is that we can harness the theory and algorithms we developed in Chapter 4 to provide bounds on the posterior probability that can be arbitrarily tightened by increasing the rank of the approximations. Thus we can exact samples from the posterior even in cases where no known algorithm can even attempt to learn kernel parameters due to the infeasibility of computing the exact posterior probability or likelihood.

We will utilize the same Bayesian methodologies described in the previous section. However, since a low-rank approximation gives us approximated eigenvalues, the bounds in Props. 1 and 2 no longer hold. Instead, we provide the bounds below by finding the upper and lower bounds on the approximated eigenvalues.

Proposition 3 *Let $\lambda_{1:\infty}$ be the eigenvalues of kernel L and $\tilde{\lambda}_{1:r}$ be the eigenvalues of its rank-r approximated kernel, $\tilde{L}$. Further assume that $\tilde{L}(\boldsymbol{x}, \boldsymbol{y})$ and $E(\boldsymbol{x}, \boldsymbol{y}) = L(\boldsymbol{x}, \boldsymbol{y}) - \tilde{L}(\boldsymbol{x}, \boldsymbol{y})$ are both positive semidefinite kernels, then*

$$\prod_{n=1}^{r}(1+\tilde{\lambda}_n) \leq \prod_{n=1}^{\infty}(1+\lambda_n) \tag{5.22}$$

and

$$\prod_{n=1}^{\infty}(1+\lambda_n) \leq \left[\prod_{n=1}^{r}(1+\tilde{\lambda}_n + (\operatorname{tr}(L) - \operatorname{tr}(\tilde{L})))\right] \exp\left\{\operatorname{tr}(L) - \operatorname{tr}(\tilde{L})\right\}. \tag{5.23}$$

Proof The first inequality follows from Lemma 5. For the second inequality, we have

$$\lambda_n \leq \tilde{\lambda}_n + \|L - \tilde{L}\| \leq \tilde{\lambda}_n + \operatorname{tr}(L) - \operatorname{tr}(\tilde{L}), \tag{5.24}$$

from Lemma 4. Let $\Lambda_r = \sum_{n=r+1}^{\infty} \lambda_n$ and $\gamma_n = 1 + \lambda_n$. Then,

$$\begin{aligned}
\prod_{n=1}^{\infty}(1+\lambda_n) &= \prod_{n=1}^{\infty}\gamma_n = (\prod_{n=1}^{r}\gamma_n)(\prod_{n=r+1}^{\infty}\gamma_n) \\
&= (\prod_{n=1}^{r}\gamma_n)(\lim_{l\to\infty}\prod_{n=r+1}^{r+l}\gamma_n) \\
&\leq (\prod_{n=1}^{r}\gamma_n)(\lim_{l\to\infty}(\sum_{n=r+1}^{r+l}\frac{\gamma_n}{l})^l) \\
&\leq (\prod_{n=1}^{r}(1+\lambda_n))\exp(\Lambda_r) \\
&\leq \left[\prod_{n=1}^{r}(1+\tilde{\lambda}_n + (\operatorname{tr}(L) - \operatorname{tr}(\tilde{L})))\right] \exp\left\{\operatorname{tr}(L) - \operatorname{tr}(\tilde{L})\right\}.
\end{aligned}$$

Proposition 4 *Let $\lambda_{1:\infty}$ be the eigenvalues of kernel L and $\tilde{\lambda}_{1:r}$ be the eigenvalues* *of its rank-r approximated kernel, $\tilde{L}$. Further assume that $\tilde{L}(\boldsymbol{x},\boldsymbol{y})$ and $E(\boldsymbol{x},\boldsymbol{y}) = L(\boldsymbol{x},\boldsymbol{y}) - \tilde{L}(\boldsymbol{x},\boldsymbol{y})$ are both positive semidefinite kernels. Let*

$$\hat{\lambda}_n = \tilde{\lambda}_n + \mathrm{tr}(L) - \mathrm{tr}(\tilde{L})$$

for $n = 1, 2, \dots$ (we make the convention that $\tilde{\lambda}_n = 0$ for $n > r$). Then

$$e_k(\lambda_{1:\infty}) \geq e_k(\tilde{\lambda}_{1:r}) \tag{5.25}$$

and

$$e_k(\lambda_{1:\infty}) \leq e_k(\hat{\lambda}_{1:\infty}). \tag{5.26}$$

Proof By Lemmas 4 and 5,

$$\tilde{\lambda}_n \leq \lambda_n \leq \hat{\lambda}_n. \tag{5.27}$$

for $n = 1, 2, \dots$. Then

$$e_k(\tilde{\lambda}_{1:r}) = \sum_{|J|=k} \prod_{n\in J} \tilde{\lambda}_n \leq \sum_{|J|=k} \prod_{n\in J} \lambda_n = e_k(\lambda_{1:\infty}) \tag{5.28}$$

and

$$e_k(\lambda_{1:\infty}) = \sum_{|J|=k} \prod_{n\in J} \lambda_n \leq \sum_{|J|=k} \prod_{n\in J} \hat{\lambda}_n = e_k(\hat{\lambda}_{1:\infty}) \ . \tag{5.29}$$

Once again, $\mathrm{tr}(L)$ in the bounds can be easily computed: $\int_\Omega L(\mathbf{x},\mathbf{x})d\mathbf{x}$. For Nyström approximation, $\mathrm{tr}(\tilde{L}) = \sum_{i=1}^r \sum_{j=1}^r \sum_{k=1}^r W_{ji} W_{ik} \int_\Omega L(\boldsymbol{x},\boldsymbol{z}_j)L(\boldsymbol{z}_k,\boldsymbol{x})d\boldsymbol{x}$. While this looks daunting at first, is nothing more than the trace of the dual matrix computed in Sec. 4.1.1. Furthermore, note that we can make these bounds arbitrarily tight by sampling more landmarks since $\tilde{\lambda}_n \to \lambda_n$ for all n and $\mathrm{tr}(\tilde{L}) \to \mathrm{tr}(L)$ as $r \to \infty$. Table 5.2 summarizes the feasibility of using MCMC-based methods to sample from exact posteriors of the parameters of DPP kernels.

Table 5.2: Examination of the feasibility of sampling from exact posteriors of the parameters of DPP kernels using MCMC-based methods.

DPP Setup	Eigendecomp.	Compute Normalizer	MCMC
Discrete(small N)	known	efficient	yes
Discrete(large N)	known	inefficient	yes w/eigenval. truncation
Cont.(Gaussian)	known	infeasible	yes w/eigenval. truncation
Cont.(most kernels)	unknown	infeasible	yes w/low-rank. approx.

5.2 Method of Moments

Convergence and mixing of MCMC samplers can be challenging to assess. Although generic techniques such as Gelman-Rubin diagnostics [Gelman and Rubin, 1992] are applicable, we additionally provide a set of tools more directly tailored to the DPP by deriving a set of theoretical moments. When performing posterior inference of kernel parameters, we can check whether the moments of our data match the theoretical moments given by the posterior samples. This can be done in cases where the eigenstructure is fully known.

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:

$$K = L(I + L)^{-1} . \tag{5.30}$$

The mth moment can then be calculated via

$$\mathbb{E}[\mathbf{x}^m] = \sum_{i=1}^{N} \mathbf{x}_i^m K(\mathbf{x}_i, \mathbf{x}_i) \ . \tag{5.31}$$

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

$$\mathbb{E}[\mathbf{x}^m] = \int_{\Omega} \sum_{n=1}^{\infty} \frac{\lambda_n}{\lambda_n + 1} \mathbf{x}^m \phi_n(\mathbf{x})^2 d\mathbf{x} \ . \tag{5.32}$$

Note that this generally cannot be evaluated in closed form since the eigendecompositions of most kernel operators are not known. However, in certain cases where the eigenfunctions are known analytically, the moments can be directly computed. For a kernel defined by Gaussian quality and similarity the eigendecomposition can be performed using Hermite polynomials. We provide the details below.

Example: Moments for Continuous DPP with Gaussian Quality and Similarity

Let

$$q(\mathbf{x}) = \sqrt{\alpha} \prod_{d=1}^{D} \frac{1}{\sqrt{\pi \rho_d}} \exp\left\{ -\frac{x_d^2}{2\rho_d} \right\} \tag{5.33}$$

and

$$k(\mathbf{x}, \mathbf{y}) = \prod_{d=1}^{D} \exp\left\{ -\frac{(x_d - y_d)^2}{2\sigma_d} \right\}, \mathbf{x}, \mathbf{y} \in \mathbb{R}^D. \tag{5.34}$$

In this case, the eigenvalues and eigenvectors of the operator L are given by [Fasshauer and McCourt, 2012]:

$$\lambda_{\mathbf{n}} = \alpha \prod_{d=1}^{D} \sqrt{\frac{1}{\frac{\beta_d^2+1}{2} + \frac{1}{2\gamma_d}}} \left(\frac{1}{\gamma_d(\beta_d^2+1)+1}\right)^{n_d-1} , \tag{5.35}$$

and

$$\phi_{\mathbf{n}}(\boldsymbol{x}) = \prod_{d=1}^{D} \left(\frac{1}{\pi\rho_d^2}\right)^{\frac{1}{4}} \sqrt{\frac{\beta_d}{2^{n_d-1}\Gamma(n_d)}} \exp\left\{-\frac{\beta_d^2 x^2}{2\rho_d^2}\right\} H_{n_d-1}\left(\frac{\beta_d x_d}{\sqrt{\rho_d^2}}\right) , \tag{5.36}$$

where $\gamma_d = \frac{\sigma_d}{\rho_d}$, $\beta_d = (1+\frac{2}{\gamma_d})^{\frac{1}{4}}$ and $\mathbf{n} = (n_1, n_2, \ldots, n_D)$ is a multi index.

In the case of DPPs (as opposed to k-DPPs), we can use the number of items as an estimate of the 0th moment. The 0th moment is given by $\sum_{\mathbf{n}=1} \frac{\lambda_{\mathbf{n}}}{1+\lambda_{\mathbf{n}}}$. Denote $\mathbf{x} = (x_1, x_2, \ldots, x_d)$. For higher moments, note that

$$\begin{aligned} E[x_j^m] &= \textstyle\int_{\mathbb{R}} \sum_{n=1}^{\infty} \frac{\lambda_n}{\lambda_n+1} x_j^m \phi_n(\mathbf{x})^2 dx_j \\ &= \textstyle\sum_{n=1}^{\infty} \frac{\lambda_n}{\lambda_n+1} \int_{\mathbb{R}} x_j^m \phi_n(\mathbf{x})^2 dx_j \ . \end{aligned}$$

Using the results of moment integrals involving a product of two Hermite polynomials [Paris, 2010], we get that

$$E[x_j^m] = \int_{\mathbb{R}^d} \sum_{\mathbf{n}}^{\infty} \frac{\lambda_{\mathbf{n}}}{\lambda_{\mathbf{n}}+1} (\frac{\sqrt{\rho_j}}{\sqrt{2}\beta_j})^m \wp_{\frac{m}{2}}(n_j - 1) \ , \tag{5.37}$$

for m even and 0 otherwise. The polynomial $\wp_{\frac{m}{2}}(n_j-1)$ is given in Eq. (4.8) in Paris [2010]. For example, the second and fourth moments are given by

(i) $E[x_j^2] = \sum_{\mathbf{n}}^{\infty} \frac{\lambda_{\mathbf{n}}}{\lambda_{\mathbf{n}}+1} (\frac{\sqrt{\rho_j}}{\sqrt{2}\beta_j})^2 (2n_j - 1)$,

(ii) $E[x_j^4] = \sum_{\mathbf{n}}^{\infty} \frac{\lambda_{\mathbf{n}}}{\lambda_{\mathbf{n}}+1} (\frac{\sqrt{\rho_j}}{\sqrt{2}\beta_j})^4 3(2n_j^2 - 2n_j + 1)$.

Unfortunately, the method of moments can be challenging to use for direct parameter learning since Eqs. (5.31) and (5.32) are not analytically available in most cases. In low dimensions, these quantities can be estimated numerically, but it remains an open question as to how these moments should be estimated for large-scale problems.

5.3 Experiments

5.3.1 Simulations

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

$$q(\mathbf{x}) = \sqrt{\alpha} \prod_{d=1}^{D} \frac{1}{\sqrt{\pi \rho_d}} \exp\left\{ -\frac{x_d^2}{2\rho_d} \right\} \tag{5.38}$$

$$k(\mathbf{x}, \mathbf{y}) = \prod_{d=1}^{D} \exp\left\{ -\frac{(x_d - y_d)^2}{2\sigma_d} \right\}, \mathbf{x}, \mathbf{y} \in \mathbb{R}^D. \tag{5.39}$$

Here, $\Theta = \{\alpha, \rho_d, \sigma_d\}$ and the eigenvalues of the operator $L(\Theta)$ are given in Sec. 5.2.

Furthermore, the trace of $L(\Theta)$ can be easily computed as

$$\operatorname{tr}(L(\Theta)) = \int_{\mathbb{R}^d} \alpha \prod_{d=1}^{D} \frac{1}{\pi \rho_d} \exp\left\{ -\frac{x_d^2}{2\rho_d} \right\} d\mathbf{x} = \alpha \ . \tag{5.40}$$

Thus, upper and lower bounds for the likelihood can be explicitly calculated and our proposed Bayesian learning algorithms are applicable.

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 in Sec. 4.2. We used weakly informative inverse gamma priors on σ,ρ and α. In particular, we used the same priors for all three parameters

$$\mathbb{P}(\alpha) = \mathbb{P}(\rho) = \mathbb{P}(\sigma) = \text{Inv-Gamma}(0.001, 0.001)\ . \tag{5.41}$$

We then learn the parameters using hyperrectangle slice sampling.

We tweak the (α, ρ, σ) used for simulation so that we have the following three 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. 5.4 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 the cases, the zeroth and second moment estimated from the posterior samples are in the neighborhood of the corresponding empirical moments as shown in Fig. 5.5.

This leads us to believe that the posterior is broad in cases where 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. Note that using a 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, one standard quantity used to measure dispersion is the Ripley K-function [Ripley, 1977]. Here, instead,

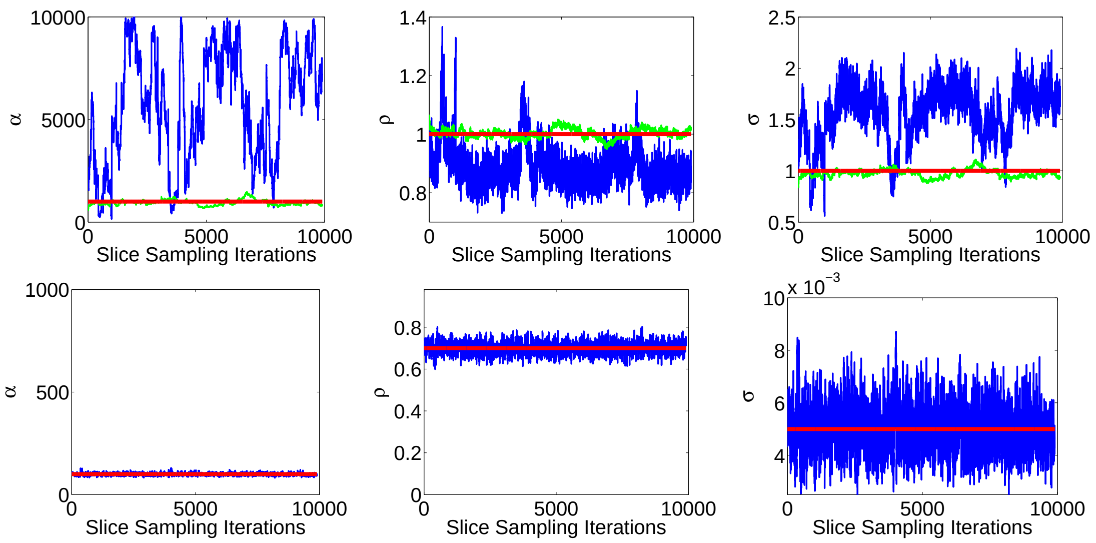


Figure 5.4: Posterior samples for a continuous DPP with Gaussian quality and similarity. The first 3 columns show posterior samples for (from right to left) α, ρ and σ. The top row are samples from Scenario (i) (blue) and Scenario (ii) (green) while the second row are samples from Scenario (iii), plotted with the same relative scales to the other scenarios. Red lines indicate the true parameter values that generated the data.

we would like to use the learned parameters of the DPP to measure overdispersion as repulsion. An important characteristic of a measure of repulsion is that it should be invariant to scaling. In the case for Gaussian quality and similarity kernel, 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 measure of repulsion is $\gamma_i = \sigma_i/\rho_i$.

5.3.2 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 appear to become more clustered as diabetes progresses. Waller et al. [2011] previously analyzed this phenomena based on 6 thigh nerve fiber samples. These samples were collected from 5 diabetic patients at different stages

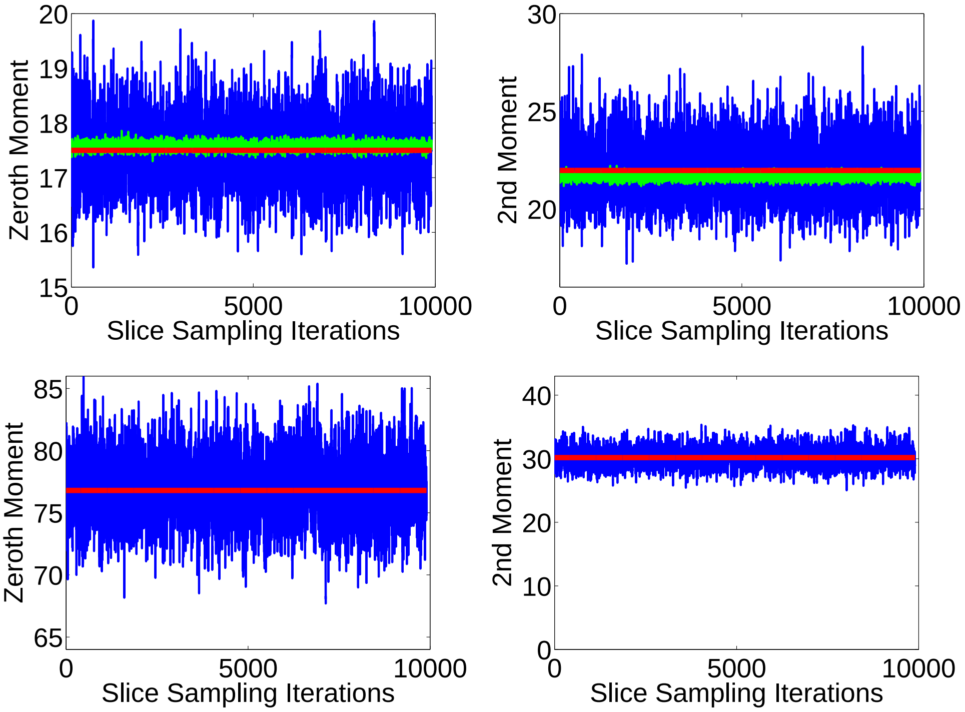


Figure 5.5: Moment estimates under the parameters learned for a continuous DPP with Gaussian quality and similarity. The two columns show the zero-th and second moment estimates. The top row are samples from Scenario (i) (blue) and Scenario (ii) (green) while the second row are samples from Scenario (iii), plotted with the same relative scales to the other scenarios. Red lines indicate the empirical moments.

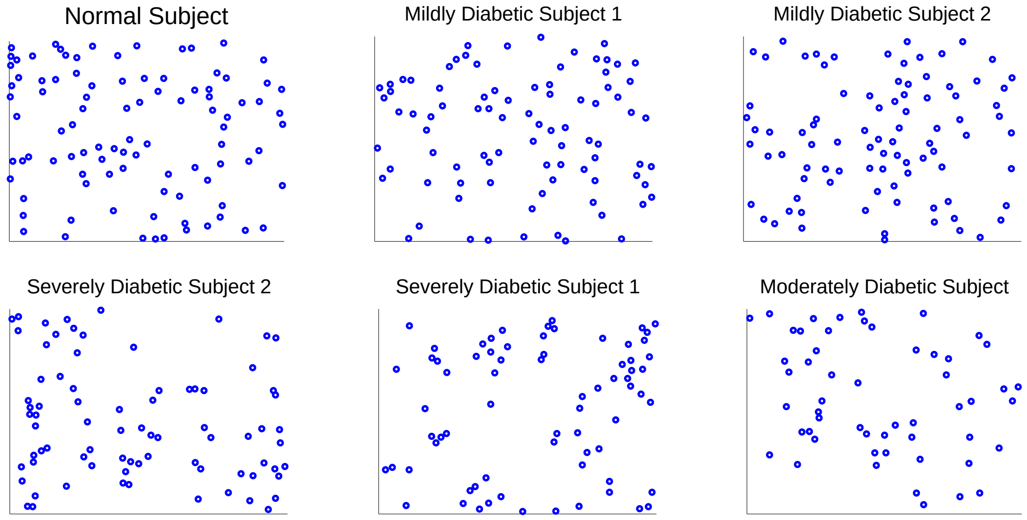


Figure 5.6: Nerve fiber samples. Clockwise: (i) Normal subject, (ii) Mildly Diabetic Subject 1, (iii) Mildly Diabetic Subject 2,(iv) Moderately Diabetic subject, (v) Severely Diabetic Subject 1 and (vi) Severely Diabetic Subject 2.

of diabetic neuropathy and one healthy subject. On average, there are 79 points in each sample (see Fig. 5.6). Waller et al. [2011] analyzed the Ripley K-function and found that the difference between the healthy and severely diabetic samples to be 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. 5.3.1). In the second, we perform naive Bayes classification on the two classes. Typically, we would train the classifier on a small subset of the data and test the classifier on the rest of the data. Here, however, since we only have six samples, 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

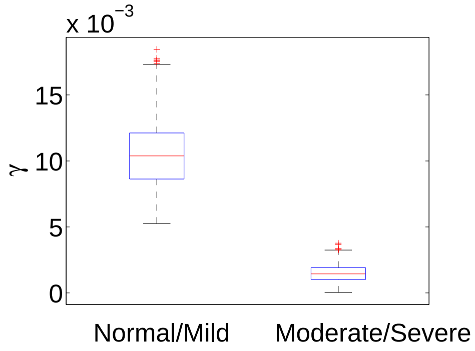


Figure 5.7: The repulsion measure, γ under the two learned DPP classes: Normal/Mildly Diabetic (left box) and Moderately/Severely Diabetic (right box)

.

all six samples.

We model our data using a 2-dimensional continuous DPP with Gaussian quality and similarity as in Eqs. (5.38) and (5.39). 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):

$$\mathbb{P}(\alpha) = \mathbb{P}(\rho) = \mathbb{P}(\sigma) = \text{Inv-Gamma}(0.001, 0.001) \tag{5.42}$$

and learn the parameters using slice sampling with eigenvalue bounds as outlined in Sec. 5.1.3. The results shown in Fig. 5.7 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 as shown in Fig. 5.8. While the results are preliminary, being based on only six observations, they show promise for this task.

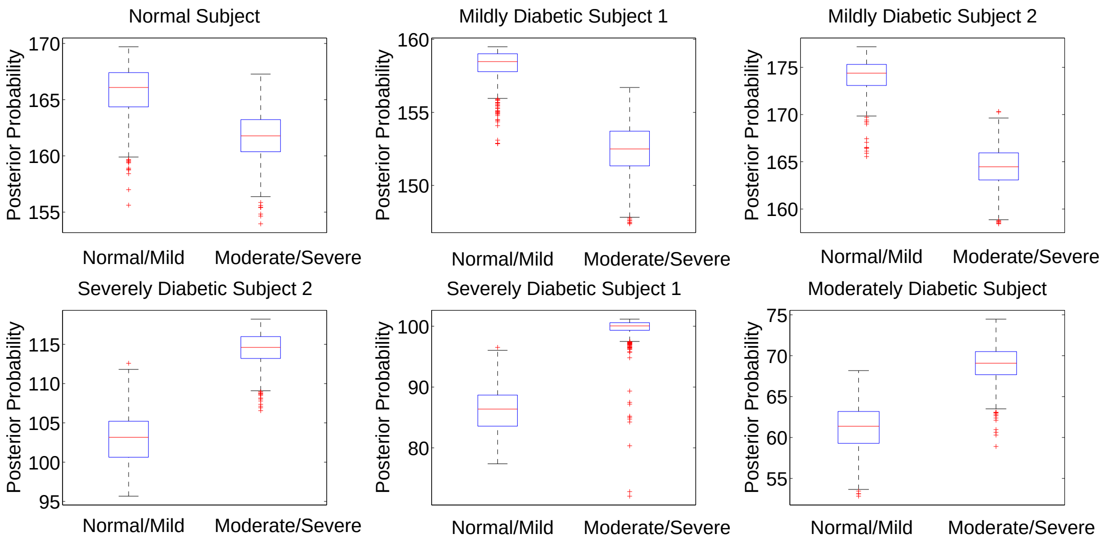


Figure 5.8: The leave-one out log-likelihood of each sample under the two learned DPP classes: Normal/Mildly Diabetic (left box) and Moderately/Severely Diabetic (right box).

5.3.3 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 and 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, human 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 and 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 (explained later). 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 spread evenly across the different subcategories.

We aim to study how the human annotated sets differ from the top six Google results. As in Kulesza and Taskar [2011a], we extracted three types of features from the images—color features, SIFT descriptors [Vedaldi and Fulkerson, 2010, Lowe, 1999] and GIST descriptors [Oliva and Torralba, 2006] below:

Color: Each pixel is assigned a coordinate in three-dimensional Lab color space. The colors are then sorted into axis-aligned bins, producing a histogram of either 8 (denoted color8) or 64 (denoted color64) dimensions.

SIFT: The images are processed to obtain sets of 128-dimensional SIFT descriptors. These descriptors are commonly used in object recognition to identify objects in images and are invariant to scaling, orientation and minor distortions. The descriptors for a given category are combined, subsampled to set of 25,000, and then clustered using k-means into either 256 (denoted SIFT256) or 512 (denoted SIFT512) clusters. The feature vector for an image is the normalized histogram of the nearest clusters to the descriptors in the image.

GIST: The images are processed to obtain 960-dimensional GIST feature vectors that is commonly used to describe scene structure.

We also extracted the features above from the center of the images, defined as the centered rectangle with dimensions half those of the original image. This yields a total of 10 different feature vectors. Since we are only concerned with the diversity of the images, we ensure that the quality across the images are uniform by normalizing each feature vector such that their L_2 norm equals to 1. We then combine the feature vectors into 3 types of features- color, SIFT and GIST.

We denote these features for image i as $f_i^{\textsf{color}}$, $f_i^{\textsf{SIFT}}$, and $f_i^{\textsf{GIST}}$, respectively. For each subcategory, we model our data as a discrete 6-DPP on $\Omega_{\textsf{subcat}}$ with kernel

$$L_{i,j}^{\textsf{subcat}} = \exp\left\{ -\sum_{\textsf{feat}} \frac{\|f_i^{\textsf{feat}} - f_j^{\textsf{feat}}\|_2^2}{\sigma_{\textsf{feat}}^{\textsf{cat}}} \right\} \tag{5.43}$$

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

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 and Taskar, 2012a] that fixes the five images from the partial results set and only considers the probability of adding the human annotated image. In general, given a partial set of observations A and k-DPP kernel L, we can define the conditional k-DPP probability of choosing a set B given the inclusion of set A (with $|A| + |B| = k$)as

$$\mathbb{P}_L^k(Y = A \cup B | A \in Y) \propto \det(L_B^A) \tag{5.44}$$

with

$$L^A = \left(\left[((L + I_{A^c})^{-1}\right]_{A^c}\right)^{-1} - I \tag{5.45}$$

where I_{A^c} denotes the identity matrix with 0 for diagonal corresponding to elements in A. Here, following the $N \times N$ inversion, the matrix is restricted to rows and columns indexed by elements not in A, then inverted again. The normalizer is given by Kulesza and Taskar [2012a].

$$\sum_{|Y'|=k-|A|} \det(L^{A}_{Y'}) \tag{5.46}$$

In our experiment, our samples can be seperated into the partial result sets and human annotations,

$$X^{t}_{\text{DPP+human}} = (A^t, b^t) \tag{5.47}$$

where A^t is the partial result sets and b^t is the human annotated result, we model the data from the conditional 6-DPP $L^{subcat}(b^t|A^t)$. In this case, the likelihood is given by

$$L^i(\Theta^{cat}) = \frac{\det(L^{i\ A^t}_{b_t}(\Theta^{cat}))}{\sum_{i=1}^{N} L^{i\ A^t}_{x_i}(\Theta^{cat})} \tag{5.48}$$

for each subcategory, i. That is, for each subcategory, i, we compute $L^i(\Theta^{cat})$ and use Eq. (5.45) to compute the conditional kernel.

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

$$\mathbb{P}(\sigma^{\mathsf{cat}}_{\mathsf{feat}}) = \text{Inv-Gamma}(0.001, 0.001)\ . \tag{5.49}$$

Fig. 5.9 shows a comparison between $\sigma^{\mathsf{cat}}_{\mathsf{feat}}$ 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 in 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 important for diversity by human annotators, while the Google search engine puts a much lower weight on them. 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.

5.3.4 Learning Mixture Model Parameters

In Secs. 4.4.1 and 4.4.2, we introduced repulsive mixture modeling and repulsive latent social clustering by putting a kDPP prior on the location parameters, $\{\mu_k\}$ and performing Gibbs sampling to learn the mixture model. In many cases, a priori, we have no information about the clusters variances and the amount of repulsion that exists in the data. Instead of setting the hyperparameters to arbitrary values, it is desirable to develop a more robust way to handle this.

We do this by learning the parameters in each iteration of the Gibbs sampling using slice sampling as proposed in Sec. 5.1.3. In experiments in Secs. 4.4.1 and 4.4.2, we find that performing the slice sampling learning of the parameters in each Gibbs iteration leads to results that are at least as good as setting $\sigma^2 = 1$ and $\rho^2 = 1$. Furthermore, this method is now robust to the scaling of the data as the slice sampler will generate parameters that adapt to the amount of variance

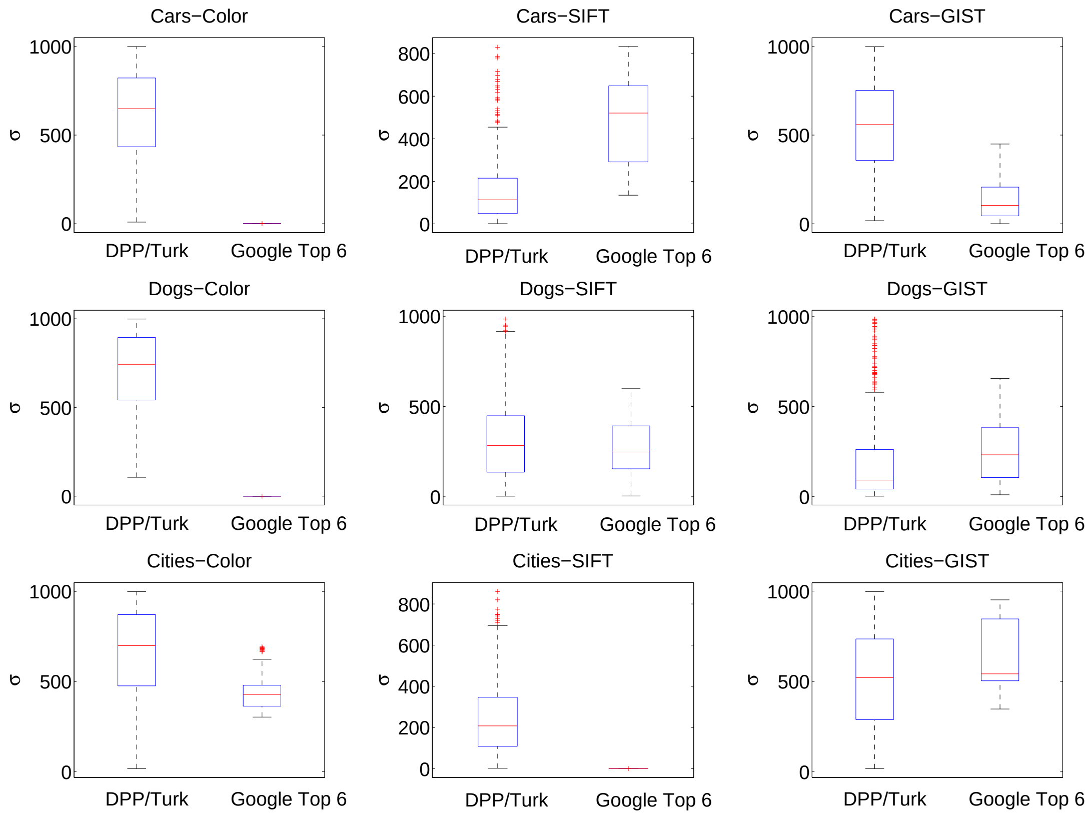


Figure 5.9: For the image diversity experiment, boxplots of posterior samples of (from left to right) $\sigma^{\mathsf{cat}}_{\mathsf{color}}$, $\sigma^{\mathsf{cat}}_{\mathsf{SIFT}}$ and $\sigma^{\mathsf{cat}}_{\mathsf{GIST}}$. Each plot shows results for human annotated sets (left) versus Google Top 6 (right). Categories from top to bottom: (a)cars, (b)dogs and (c)cities

and repulsion among clusters.

5.4 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 due to the fact that the likelihood is non-convex and that the likelihood and it's gradient are either not known or not computationally feasible in many scenarios. We proposed Bayesian approaches using MCMC, in particular, for inferring these parameters. In addition to being more robust and providing a characterization of the posterior uncertainty, these algorithms can be modified to deal with large-scale and continuous DPPs. We also showed how our posterior samples can be evaluated using moment matching as a model-checking method. Finally we demonstrated the utility of learning DPP parameters in studying diabetic neuropathy and evaluating human perception of diversity in images. We also illustrated that we can perform full Bayesian inference in mixture models by combining the continuous DPP sampling algorithm and the learning of the kernel parameters.

Chapter 6

Markov DPPs

While a discrete DPP is useful in selecting *diverse* subcollections, there are many applications that require these subsets to be diverse not just individually but also through time. For example, we might use a DPP to display a set of news headlines that are relevant to a user's interests while maintaining diversity, covering a variety of topics. Suppose further that we are asked to sequentially select *multiple* diverse sets of items, for example, displaying new headlines day-by-day. In this case we want the subsets to be diverse across time, offering headlines today that are unlike the ones shown yesterday. In this chapter, we construct a *Markov* DPP (M-DPP) that models a *sequence* of random sets $\{\boldsymbol{Y}_t\}$. The proposed M-DPP defines a stationary process that maintains DPP margins. Crucially, the induced union process $\boldsymbol{Z}_t \equiv \boldsymbol{Y}_t \cup \boldsymbol{Y}_{t-1} \cup \cdots \cup \boldsymbol{Y}_{t-p}$ is also marginally DPP-distributed. Jointly, these properties imply that the sequence of random sets are encouraged to be diverse both at a given time step as well as across consecutive time steps. Figs. 6.1 and 6.2 illustrate the comparison between a sequence of independent samples from a DPP and a sequence sampled from a first order M-DPP which introduces diversity across two consecutive time-steps. This process can also be extended to a higher order such that more than two consecutive subsets are jointly diverse.

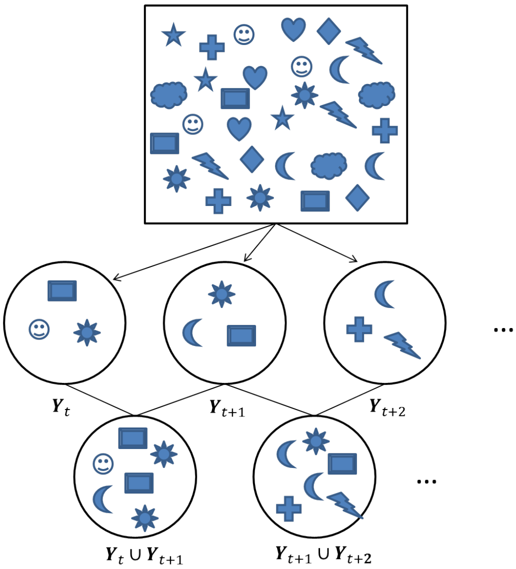


Figure 6.1: Sequence of samples drawn independently from a DPP. While DPP points are diverse within each time steps, the union of subsets across consecutive time-steps are not.

Fig. 6.3 illustrates the case for order 2. Actual samples from a 1-dimensional DPP and M-DPP with Gaussian kernels are shown in Fig. 6.4.

Our specific construction of the M-DPP yields an exact sampling procedure that can be performed in polynomial time. Additionally, we explore a method for incrementally learning the quality of each item in the base set $\mathcal{Y}$ based on externally provided preferences. In particular, a decomposition of the DPP kernel matrix has an interpretation as defining the quality of each item and pairwise similarities between items. Our incremental learning procedure assumes a well-defined similarity metric and aims to learn features of items that a user deems as preferable. These features are used to define the quality scores for each item. The M-DPP aids in the exploration of items of interest to the user by providing sequentially diverse results.

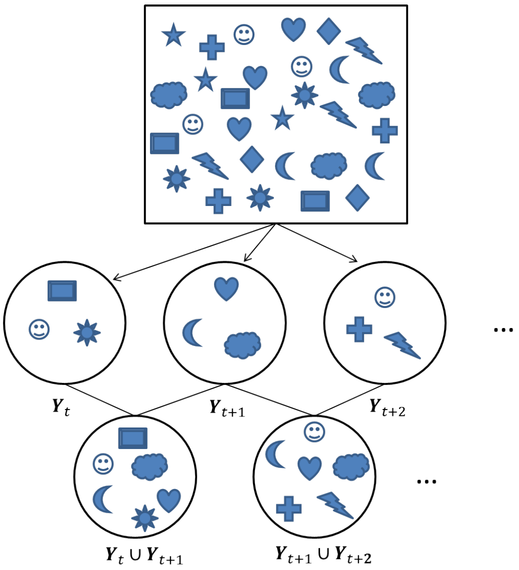


Figure 6.2: Sequence of samples drawn from a first order Markov-DPP. Not only are the M-DPP samples diverse within each time steps but the union of two consecutive subsets are jointly diverse as well.

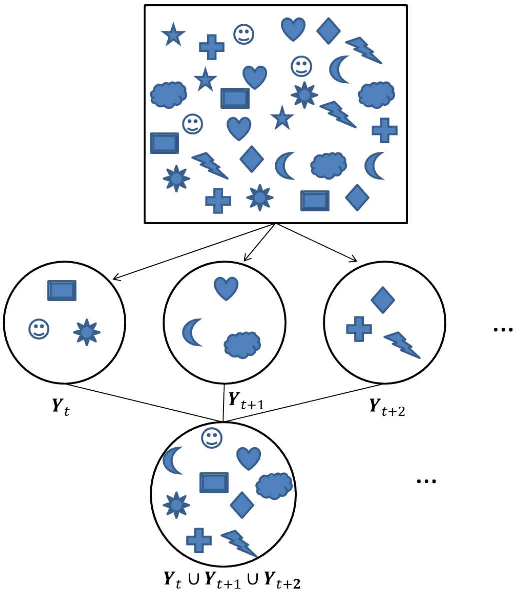


Figure 6.3: Sequence of samples drawn from a second order Markov-DPP. Not only are the second order M-DPP samples diverse within each time steps but the union of any of the three consecutive subsets are jointly diverse as well.

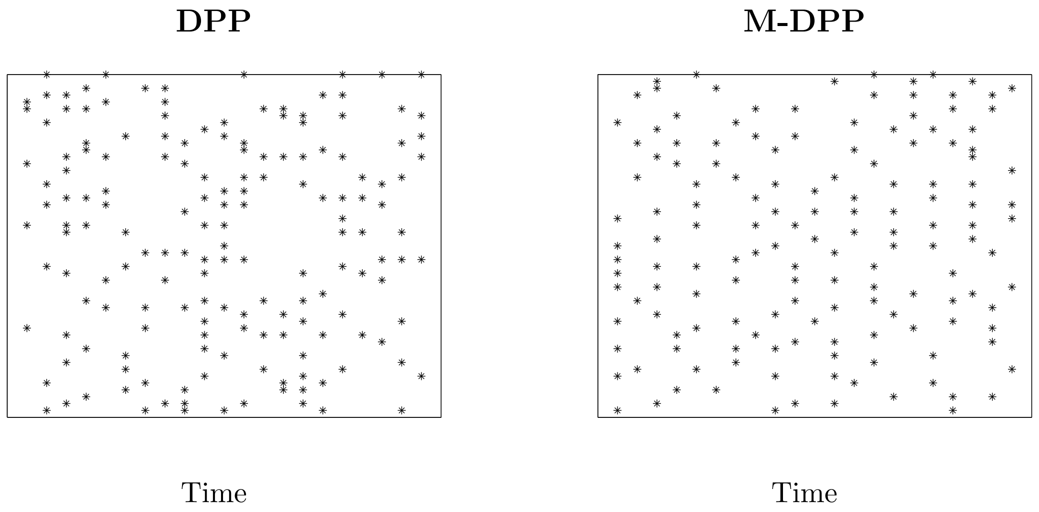


Figure 6.4: A set of points on a line (y axis) drawn from a DPP independently over time (left) and from a M-DPP (right). While DPP points are diverse only within time steps (columns), M-DPP points are also diverse across time steps.

6.1 Markov DPPs (M-DPPs)

In certain applications, such as in the task of displaying news headlines, our goal is not only to generate a diverse collection of items at one time point, but also to generate collections of items at subsequent time points that are both highly relevant and dissimilar to the previous collection. To address these goals, we introduce the Markov determinantal point process (M-DPP), which emphasizes both marginal and conditional diversity of selected items. Harnessing the *quality* and *similarity* interpretation of the DPP in Eq. (2.11), the M-DPP provides a dynamic way of selecting high quality and diverse collections of items as a temporal process. In this section, we first explore the easier construction of a first order M-DPP/M-kDPP including their sampling algorithms. We then extend this construction to M-DPP/M-kDPP of higher order such that longer sequences of consecutive subsets are ensured to be jointly diverse.

6.1.1 First Order M-DPPs

We constructively define a first-order, discrete-time Markov point process on $\mathcal{Y}$ by specifying a Markov transition distribution (and initial distribution). Throughout, we use the notation $\{\boldsymbol{Y}_t\}$, $\boldsymbol{Y}_t \subseteq \mathcal{Y}$, to represent a sequence of sets following a M-DPP. We consider two such constructions: one based on marginal kernels, and the other on L-ensembles. Both yield equivalent stationary processes with DPP margins. Additionally, and quite intuitively, the induced union process $\{\boldsymbol{Z}_t \equiv \mathbf{Y}_t \cup \mathbf{Y}_{t-1}\}$ has DPP margins with a closely related kernel. Combining these two properties, we conclude that the constructed M-DPPs yield a sequence of sets $\{\boldsymbol{Y}_t\}$ that are diverse at any time t and across time steps $t, t-1$.

Marginal construction. Let K be a marginal kernel with $K \prec \frac{1}{2}I$ (that is, all the eigenvalues are non-negative and less than $\frac{1}{2}$). Define $\mathcal{P}(\boldsymbol{Y}_1 \supseteq A) = \det(K_A)$ and

$$\mathcal{P}(\boldsymbol{Y}_t \supseteq B | \boldsymbol{Y}_{t-1} \supseteq A) = \frac{\det(K_{A\cup B})}{\det(K_A)} \ , \tag{6.1}$$

where $A \cap B = \emptyset$. Throughout, we adopt the implicit constraint that $\mathbf{Y}_t \cap \mathbf{Y}_{t-1} = \emptyset$. We have immediately the joint probability

$$\mathcal{P}(\boldsymbol{Y}_2 \supseteq B, \boldsymbol{Y}_1 \supseteq A) = \det(K_{A\cup B}) \ , \tag{6.2}$$

and therefore

$$\mathcal{P}(\boldsymbol{Y}_2 \supseteq B) = \mathcal{P}(\boldsymbol{Y}_2 \supseteq B, \boldsymbol{Y}_1 \supseteq \emptyset) = \det(K_B) \ . \tag{6.3}$$

Inductively, the process is stationary and marginally DPP.

Finally, we have the union of consecutive sets:

$$\begin{aligned}
&\mathcal{P}(\boldsymbol{Z}_t \equiv \boldsymbol{Y}_t \cup \boldsymbol{Y}_{t-1} \supseteq C) \\
&\quad = \sum_{A \subseteq C} \mathcal{P}(\boldsymbol{Y}_t \supseteq C \setminus A, \boldsymbol{Y}_{t-1} \supseteq A) = \sum_{A \subseteq C} \det(K_C) \\
&\quad = 2^{|C|} \det(K_C) = \det((2K)_C) \; . \qquad (6.4)
\end{aligned}$$

That is, $\boldsymbol{Z}_t$ is marginally distributed as a DPP with marginal kernel $2K$. Since a randomly sampled subset of a DPP-distributed set also follows a DPP, marginally we can imagine this process as sampling $\boldsymbol{Z}_t$ and then splitting its elements randomly into two sets, $\boldsymbol{Y}_{t-1}$ and $\boldsymbol{Y}_t$.

L-ensemble construction. The above is appealingly simple, but the marginal form of the conditional in Eq. (6.1) is not particularly conducive to a sequential sampling process. Instead, we can rewrite everything as L-ensembles. Assume that at the first time step $\mathcal{P}(\boldsymbol{Y}_1 = Y_1) = \frac{\det(L_{Y_1})}{\det(L+I)}$ and define the transition distribution as

$$\mathcal{P}(\boldsymbol{Y}_t = Y_t | \boldsymbol{Y}_{t-1} = Y_{t-1}) = \frac{\det(M_{Y_t \cup Y_{t-1}})}{\det(M + I_{\mathcal{Y} \setminus Y_{t-1}})} \; , \qquad (6.5)$$

for $M = L(I-L)^{-1}$. Note that the transition distribution is essentially a conditional DPP with L-ensemble kernel M (Eq. (2.5)). M is well-defined as long as $L \prec I$, which is equivalent to $K \prec \frac{1}{2}I$, as in the marginal construction.

Now we have the joint probability

$$\mathcal{P}(\boldsymbol{Y}_2 = Y_2, \boldsymbol{Y}_1 = Y_1) = \frac{\det(M_{Y_1 \cup Y_2})}{\det(M + I_{\mathcal{Y} \setminus Y_1})} \frac{\det(L_{Y_1})}{\det(L + I)} \; . \qquad (6.6)$$

Using the fact that $\det(M + I_{\mathcal{Y}\setminus Y_1})/\det(M + I) = \det(L_{Y_1})$,

$$\mathcal{P}(\boldsymbol{Y}_2 = Y_2, \boldsymbol{Y}_1 = Y_1) = \frac{1}{\det(L + I)} \frac{\det(M_{Y_1 \cup Y_2})}{\det(M + I)} \ . \tag{6.7}$$

Therefore, marginally,

$$\begin{aligned} \mathcal{P}(\boldsymbol{Y}_2 = Y_2) &= \sum_{Y_1 \subseteq \mathcal{Y}} \frac{1}{\det(L + I)} \frac{\det(M_{Y_1 \cup Y_2})}{\det(M + I)} \\ &= \sum_{(Y_1 \cup Y_2) \supseteq Y_2} \frac{1}{\det(L + I)} \frac{\det(M_{Y_1 \cup Y_2})}{\det(M + I)} \\ &= \frac{\det(M + I_{\mathcal{Y}\setminus Y_2})}{\det(L + I)\det(M + I)} = \frac{\det(L_{Y_2})}{\det(L + I)} \ . \end{aligned} \tag{6.8}$$

Here, we used $\sum_{B \supseteq A} \det(M_B) = \det(M + I_{\mathcal{Y}\setminus A})$, which is immediately derived from Eq. (2.5). By induction, we conclude

$$\mathcal{P}(\boldsymbol{Y}_t = Y_t) = \frac{\det(L_{Y_t})}{\det(L + I)} \ . \tag{6.9}$$

Thus, our construction yields a stationary process with $\boldsymbol{Y}_t$ marginally distributed as a DPP with L-ensemble kernel L.

One can likewise analyze the margin of the induced union process $\{\boldsymbol{Z}_t \equiv$

$\boldsymbol{Y}_t \cup \boldsymbol{Y}_{t-1}\}$:

$$\begin{aligned}
&\mathcal{P}(\boldsymbol{Z}_t \equiv \boldsymbol{Y}_t \cup \boldsymbol{Y}_{t-1} = C) \\
&\quad = \sum_{A \subseteq C} \mathcal{P}(\boldsymbol{Y}_t = C \setminus A, \boldsymbol{Y}_{t-1} = A) \\
&\quad = \sum_{A \subseteq C} \frac{1}{\det(L+I)} \frac{\det(M_C)}{\det(M+I)} \\
&\quad = \frac{2^{|C|}}{\det(L+I)} \frac{\det(M_C)}{\det(M+I)} && (6.10) \\
&\quad = \frac{1}{\det(L+I)} \frac{\det((2M)_C)}{\det(M+I)} \ . && (6.11)
\end{aligned}$$

Noting that

$$\begin{aligned}
&\det(M+I)\det(L+I) = \det((M+I)(L+I)) \\
&\quad = \det((M+I)(I-(M+I)^{-1}+I)) \\
&\quad = \det(2M+2I-I) = \det(2M+I) \ , && (6.12)
\end{aligned}$$

we conclude

$$\mathcal{P}(\boldsymbol{Z}_t \equiv \boldsymbol{Y}_t \cup \boldsymbol{Y}_{t-1} = C) = \frac{\det((2M)_C)}{\det(2M+I)} \ . \tag{6.13}$$

We have shown that $\boldsymbol{Z}_t$ is marginally distributed as a DPP with L-ensemble kernel $2M$. The corresponding marginal kernel is

$$\begin{aligned}
2M(2M+I)^{-1} &= 2L(I-L)^{-1}\left[(L+I)(I-L)^{-1}\right]^{-1} \\
&= 2L(L+I)^{-1} = 2K \ . && (6.14)
\end{aligned}$$

Thus, we have reproduced the same characterization of $\boldsymbol{Z}_t$ as in Eq. (6.4) for the marginal kernel construction.

To summarize the marginal properties of the M-DPP, using the notation $\boldsymbol{Y} \sim L, K$ to denote that $\boldsymbol{Y}$ is from a DPP with L-ensemble kernel L and marginal kernel K, we have:

$$\boldsymbol{Y}_t \sim L, K \tag{6.15}$$

$$\boldsymbol{Z}_t \sim 2L(I-L)^{-1}, 2K \ . \tag{6.16}$$

6.1.2 First Order Markov kDPPs

One can also construct a first order *Markov kDPP* (M-kDPP). Although we define a stationary process, our construction does not yield $\boldsymbol{Y}_t$ marginally kDPP. Instead, it yields what we will call a thinned-kDPP which encourages the sets to be diverse as well. However, the M-kDPP ensures that $\boldsymbol{Z}_t \equiv \boldsymbol{Y}_t \cup \boldsymbol{Y}_{t-1}$ follows a $2k$DPP. Since $\boldsymbol{Z}_t$ is encouraged to be diverse, the subsets $\boldsymbol{Y}_t$ and $\boldsymbol{Y}_{t-1}$ will likewise be diverse despite not following a kDPP themselves. Fig. 6.5 illustrates the process of drawing from a thinned-kDPP.

We start by defining the margin and transition distributions:

$$\mathcal{P}(\boldsymbol{Y}_{t-1} = Y_{t-1}) = \frac{\sum_{|A|=k} \det(L_{Y_{t-1} \cup A})}{\binom{2k}{k} \sum_{|B|=2k} \det(L_B)} \tag{6.17}$$

$$\mathcal{P}(\boldsymbol{Y}_t = Y_t | \boldsymbol{Y}_{t-1} = Y_{t-1}) = \frac{\det(L_{Y_{t-1} \cup Y_t})}{\sum_{|A|=k} \det(L_{Y_{t-1} \cup A})} \ , \tag{6.18}$$

where A and Y_t are disjoint from Y_{t-1}. Then, jointly

$$\mathcal{P}(\boldsymbol{Y}_t = Y_t, \boldsymbol{Y}_{t-1} = Y_{t-1}) = \frac{\det(L_{Y_{t-1} \cup Y_t})}{\binom{2k}{k} \sum_{|B|=2k} \det(L_B)} \ , \tag{6.19}$$

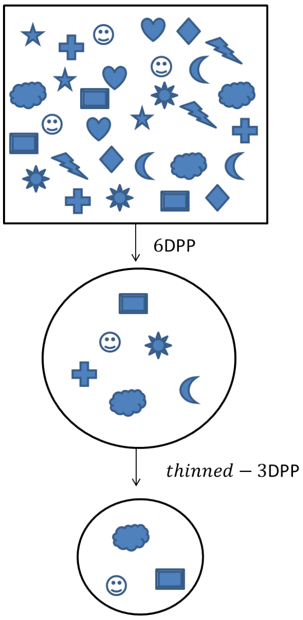


Figure 6.5: Drawing a *thinned* − kDPP sample. First, samples are drawn from a $2k$DPP. Then the set is thinned by randomly selecting half of the points in the set.

from which we confirm the stationarity of the process:

$$\mathcal{P}(\mathbf{Y}_t = Y_t) = \frac{\sum_{|Y_{t-1}|=k} \det(L_{Y_t \cup Y_{t-1}})}{\binom{2k}{k} \sum_{|B|=2k} \det(L_B)} \ . \tag{6.20}$$

The implied union process has margins

$$\begin{aligned}
&\mathcal{P}(\boldsymbol{Z}_t \equiv \mathbf{Y}_t \cup \mathbf{Y}_{t-1} = C) \\
&\quad = \sum_{A \subseteq C, |A|=k} \mathcal{P}(\mathbf{Y}_t = C \setminus A, \mathbf{Y}_{t-1} = A) \\
&\quad = \sum_{A \subseteq C, |A|=k} \frac{\det(L_C)}{\binom{2k}{k} \sum_{|B|=2k} \det(L_B)} \\
&\quad = \frac{\det(L_C)}{\sum_{|B|=2k} \det(L_B)} \ ,
\end{aligned} \tag{6.21}$$

which is a $2k$DPP with L-ensemble kernel L.

Algorithm 11 Sampling from a Markov DPP

```
Input: matrix $L$
$M \leftarrow L(I-L)^{-1}$
$Y_1 \leftarrow$ DPP-SAMPLE($L$) using Alg. 1
for $t = 2, \ldots, T$ do
    $L^{(t)} \leftarrow \left((M + I_{\mathcal{Y}\setminus Y_{t-1}})^{-1}_{\mathcal{Y}\setminus Y_{t-1}}\right)^{-1} - I$
    $Y_t \leftarrow$ DPP-SAMPLE($L^{(t)}$) using Alg. 1
Output: $\{Y_t\}$
```

6.1.3 Sampling from First Order M-DPPs and M-kDPPs

In the previous subsections we showed how our constructions of M-(k)DPPs lead to DPP (and DPP-like) marginals for $\{\boldsymbol{Y}_t\}$ and the union process $\{\boldsymbol{Z}_t\}$. These connections to DPPs give us valuable intuition about the diversity induced both within and across time steps. They serve another purpose as well: since DPPs and kDPPs can be sampled in polynomial time, we can leverage existing algorithms to efficiently sample from M-DPPs and M-kDPPs.

To sample from M-DPPs, we will proceed sequentially, first sampling $\boldsymbol{Y}_1$ from the initial distribution and then repeatedly selecting $\boldsymbol{Y}_t$ from the transition distribution given $\boldsymbol{Y}_{t-1}$. The initial distribution is a DPP with L-ensemble kernel L and can therefore be sampled directly using Alg. 1. As we have shown, the transition distribution Eq. (6.5) is a conditional DPP with L-ensemble kernel $M = L(I-L)^{-1}$; using Eq. (2.7), the L-ensemble kernel for $\boldsymbol{Y}_t$ given $\boldsymbol{Y}_{t-1} = Y_{t-1}$ can be written as

$$L^{(t)} = \left((M + I_{\mathcal{Y}\setminus Y_{t-1}})^{-1}_{\mathcal{Y}\setminus Y_{t-1}}\right)^{-1} - I \ . \tag{6.22}$$

Thus we can sample simply and efficiently from a M-DPP using Alg. 11. The runtime is $O(TN^3 + TNk_{\max}^3)$, where $k_{\max}$ is the maximum number of items chosen at a single time step. Note that for constant $k_{\max}$ this is the same runtime as a Kalman filter with a state vector of size N.

Algorithm 12 Sampling from a Markov kDPP

```
**Input:** matrix $L$, size $k$
$Z_1 \leftarrow k\textsc{DPP-sample}(L, 2k)$ using Alg. 2
$Y_1 \leftarrow$ random half of $Z_1$
**for** $t = 2, \ldots, T$ **do**
    $L^{(t)} \leftarrow \left((L + I_{\mathcal{Y}\setminus Y_{t-1}})^{-1}_{\mathcal{Y}\setminus Y_{t-1}}\right)^{-1} - I$
    $Y_t \leftarrow k\textsc{DPP-sample}(L^{(t)}, k)$ using Alg. 2
**Output:** $\{Y_t\}$
```

To sample from M-kDPP, we can now use Alg. 12 to perform sequential sampling for a M-kDPP. At first glance, the initial distribution (which is not a kDPP) seems difficult to sample; however it can be obtained by harnessing the union process form of Eq. (6.42) and first sampling a $2k$DPP with L-ensemble kernel L and then throwing away half of the resulting items at random. For this reason, we call this process *thinned-kDPP*. Transitionally, we have a conditional kDPP whose kernel can be computed as in Eq. (6.22). Alg. 12 summarizes the M-kDPP sampling process, which runs in time $O(TN^3 + TNk^3)$.

6.1.4 Higher Order M-DPPs

While we have shown that the first order M-DPP subsets are diverse at subsequent time steps, this does not necessarily imply diversity at longer intervals. In fact, it is possible for realizations to have oscillations, where groups of high-quality items recur every two (or more) time steps. While our experiments (Sec. 6.3) suggest that first order M-DPP oscillations do not arise in the task we study, it is useful to construct higher order M-DPPs/M-kDPPs such that subsets are diverse across a longer consecutive time steps. Here we extend the construction of the first order M-DPPs to a higher order process.

Here we constructively define a pth-order, discrete-time autoregressive point process on $\mathcal{Y}$ by specifying a Markov transition distribution (and initial distribution).

We once again consider two such constructions: one based on marginal kernels, and the other on L-ensembles. Both yield equivalent stationary processes with DPP margins. Furthermore, the induced union process $\{\boldsymbol{Z}_t \equiv \boldsymbol{Y}_t \cup \boldsymbol{Y}_{t-1} \cup \cdots \cup \boldsymbol{Y}_{t-p}\}$ has DPP margins with a closely related kernel. We can conclude that the constructed M-DPPs yield a sequence of sets $\{\boldsymbol{Y}_t\}$ that are diverse at any time t and across time steps $t, t-1, \ldots, t-p$.

Marginal construction. Let K be a marginal kernel with $K \prec \frac{1}{p+1} I$. Define $\mathcal{P}(\boldsymbol{Y}_1 \supseteq A) = \det(K_A)$ and

$$\mathcal{P}(\boldsymbol{Y}_t \supseteq A_t | \boldsymbol{Y}_{t-1} \supseteq A_{t-1}, \ldots, \boldsymbol{Y}_{t-l} \supseteq A_{t-l}) = \frac{\det(K_{A_t \cup A_{t-1} \cup \cdots \cup A_{t-l}})}{\det(K_{A_{t-1} \cup A_{t-2} \cup \cdots \cup A_{t-l}})}, \qquad (6.23)$$

for $l = 1, \ldots, p$.

Once again, we assume that $\boldsymbol{Y}_t \cap \boldsymbol{Y}_{t-1} \cap \ldots \cap \boldsymbol{Y}_{t-p} = \emptyset$. We have immediately the joint probability

$$\mathcal{P}(\boldsymbol{Y}_t \supseteq A_t, \boldsymbol{Y}_{t-1} \supseteq A_{t-1}, \ldots, \boldsymbol{Y}_{t-l} \supseteq A_{t-l}) = \det(K_{A_t \cup A_{t-1} \cup \cdots \cup A_{t-l}}), \qquad (6.24)$$

for $l = 1, \ldots, p$. and therefore

$$\mathcal{P}(\boldsymbol{Y}_t \supseteq A_t) = \mathcal{P}(\boldsymbol{Y}_t \supseteq A_t, \boldsymbol{Y}_{t-1} \supseteq \emptyset, \ldots, \boldsymbol{Y}_{t-l} \supseteq \emptyset) = \det(K_{A_t}) \qquad (6.25)$$

Inductively then, the process is stationary and marginally DPP. Finally, we have the union of consecutive sets: for $l = 1, 2, \dots, p$,

$$\begin{aligned}
&\mathcal{P}(\boldsymbol{Z}_t \equiv \boldsymbol{Y}_t \cup \boldsymbol{Y}_{t-1} \cup \cdots \cup \boldsymbol{Y}_{t-l} \supseteq C_l) \\
&\quad = \sum_{C_1 \subseteq C_2} \sum_{C_2 \subseteq C_3} \cdots \sum_{C_{l-1} \subseteq C_l} \mathcal{P}(\boldsymbol{Y}_t \supseteq C_l \setminus C_{l-1}, \boldsymbol{Y}_{t-1} \supseteq C_{l-1} \setminus C_{l-2}, \cdots, \boldsymbol{Y}_{t-1} \supseteq C_1) \\
&\quad = \sum_{C_1 \subseteq C_2} \sum_{C_2 \subseteq C_3} \cdots \sum_{C_{l-1} \subseteq C_l} \det(K_{C_l}) \\
&\quad = (l+1)^{|C|} \det(K_{C_l}) = \det\left(((l+1)K)_{C_l}\right) \ . \qquad (6.26)
\end{aligned}$$

where we used the binomial expansion identity, $\sum_{i=0}^{N} \binom{N}{i} m^i = (m+1)^N$ for any integer m.

Thus the union process of the $l+1$ consecutive sets (with $l \leq p$) is marginally distributed as a DPP with marginal kernel $(l+1)K$. Note that we recover the first order M-DPPs as a special case when $p = 1$.

L-ensemble construction. Once again, we can rewrite everything as L-ensembles. Assume that at the first time step $\mathcal{P}(\boldsymbol{Y}_1 = Y_1) = \frac{\det(L_{Y_1})}{\det(L+I)}$ and define the transition distribution as

$$\mathcal{P}(\boldsymbol{Y}_t = A_t | \boldsymbol{Y}_{t-1} = A_{t-1}, \dots, \boldsymbol{Y}_{t-l} = A_{t-l}) = \frac{\det(M^{(l)}_{A_t \cup A_{t-1} \cup \cdots \cup A_{t-l}})}{\det(M^{(l)} + I_{\mathcal{Y} \setminus (A_{t-1} \cup \cdots \cup A_{t-l})})}, \quad (6.27)$$

where $M^{(l)} = L(I - lL)^{-1}$ for $l = 1, \dots, p$.

Note here that the transition distribution is a conditional DPP with L-ensemble kernel $M^{(l)}$ (Eq. (2.5)). M is well-defined as long as $L \prec \frac{1}{p} I$, which is equivalent to $K \prec \frac{1}{p+1} I$, as in the marginal construction.

For $t = 1, \ldots, p+1$, we now we have the joint probability

$$\begin{aligned} &\mathcal{P}(\boldsymbol{Y}_t = A_t, \boldsymbol{Y}_{t-1} = A_{t-1}, \ldots, \boldsymbol{Y}_1 = A_1) \\ &= \left[\prod_{i=1}^{t-1} \frac{\det(M^{(i)}_{A_{i+1} \cup \cdots \cup A_1})}{\det(M^{(i)} + I_{\mathcal{Y} \setminus (A_i \cup \cdots \cup A_1)})} \right] \frac{\det(L_{A_1})}{\det(L+I)}, \end{aligned} \tag{6.28}$$

Using the fact that $\det(M^{(l)} + I_{\mathcal{Y} \setminus Y}) / \det(M^{(l)} + I) = \det(M^{(l-1)}_Y)$ for $l = 2, \ldots, p$ and $\det(M^{(1)} + I_{\mathcal{Y} \setminus Y}) / \det(M^{(1)} + I) = \det(L_Y)$,

$$\begin{aligned} &\mathcal{P}(\boldsymbol{Y}_t = A_t, \boldsymbol{Y}_{t-1} = A_{t-1}, \ldots, \boldsymbol{Y}_1 = A_1) \\ &= \left[\prod_{i=1}^{t-1} \frac{1}{\det(M^{(i)} + I)} \right] \frac{\det(M^{(t-1)}_{A_t \cup \cdots \cup A_1})}{\det(L+I)}, \end{aligned} \tag{6.29}$$

Using the fact that $\sum_{B \supseteq A} \det(M^{(i)}_B) = \det(M^{(i)} + I_{\mathcal{Y} \setminus A})$ from Eq. (2.5), marginally we have

$$\begin{aligned} &\mathcal{P}(\boldsymbol{Y}_t = A_t) \\ &= \left[\prod_{i=1}^{t-1} \frac{1}{\det(M^{(i)} + I)} \right] \sum_{(A_1 \cup A_2 \cup \ldots \cup A_t) \supseteq (A_1 \cup A_2 \cup \ldots \cup A_{t-1})} \cdots \sum_{(A_1 \cup A_2) \supseteq A_1} \frac{\det(M^{(t-1)}_{A_t \cup \cdots \cup A_1})}{\det(L+I)} \\ &= \frac{\det(L_{Y_t})}{\det(L+I)} \ . \end{aligned} \tag{6.30}$$

By induction, we conclude that $\boldsymbol{Y}_t$ is marginally distributed a a DPP with kernel L for all t.

One can likewise analyze the margin of the induced union process $\{\boldsymbol{Z}_t \equiv$

$\boldsymbol{Y}_t \cup \boldsymbol{Y}_{t-1} \cup \boldsymbol{Y}_1\}$ for $t = 2, \ldots, p+1$:

$$\begin{aligned}
&\mathcal{P}(\boldsymbol{Z}_t \equiv \boldsymbol{Y}_t \cup \boldsymbol{Y}_{t-1} \cup \boldsymbol{Y}_1 = C_t) \\
&\quad = \sum_{C_1 \subseteq C_2} \sum_{C_2 \subseteq C_3} \cdots \sum_{C_{t-1} \subseteq C_t} \mathcal{P}(\boldsymbol{Y}_t = C_t \setminus C_{t-1}, \boldsymbol{Y}_{t-1} = C_{t-1} \setminus C_{t-2}, \cdots, \boldsymbol{Y}_1 = C_1) \\
&\quad = \sum_{C_1 \subseteq C_2} \sum_{C_2 \subseteq C_3} \cdots \sum_{C_{l-1} \subseteq C_l} \left[\prod_{i=1}^{t-1} \frac{1}{\det(M^{(i)} + I)} \right] \frac{\det(M^{(t-1)}_{C_t})}{\det(L+I)} \\
&\quad = t^{|C_t|} \left[\prod_{i=1}^{t-1} \frac{1}{\det(M^{(i)} + I)} \right] \frac{\det(M^{(t-1)}_{C_t})}{\det(L+I)}, \qquad (6.31)
\end{aligned}$$

where we once again used the binomial expansion identity, $\sum_{i=0}^{N} \binom{N}{i} m^i = (m+1)^N$ for any integer m. Noting that

$$\begin{aligned}
\left[\prod_{i=1}^{t-1} \det(M^{(i)} + I) \right] \det(L+I) &= \left[\prod_{i=1}^{t-1} \det(L(I - iL)^{-1} + I) \right] \det(L+I) \\
&= \det\left(tM^{(t-1)} + I\right), \qquad (6.32)
\end{aligned}$$

by induction, we conclude that for $l = 1, \ldots, p$:

$$\mathcal{P}(\boldsymbol{Z}_t \equiv \boldsymbol{Y}_t \cup \boldsymbol{Y}_{t-1} \cup \boldsymbol{Y}_{t-l} = C) = \frac{\det\left(((l+1)M^{(l)})_C\right)}{\det\left((l+1)M^{(l)} + I\right)} \ . \qquad (6.33)$$

We have shown that the union of $l+1$ consecutive sets (for $l = 1, \ldots, p$) is marginally distributed as a DPP with L-ensemble kernel $(l+1)M^{(l)}$. The corresponding marginal kernel is

$$\begin{aligned}
(l+1)M^{(l)} \left[(l+1)M^{(l)} + I\right]^{-1} &= (l+1)L(I - lL)^{-1} \left[(l+1)L(I - lL)^{-1} + I\right]^{-1} \\
&= (l+1)L(L+I)^{-1} = (l+1)K \ . \qquad (6.34)
\end{aligned}$$

recovering our marginal kernel construction.

For $l = 1, \ldots, p$, we can summarize the process as follows:

$$\boldsymbol{Y}_t \sim L, K, \tag{6.35}$$

$$\boldsymbol{Y}_t \cup \boldsymbol{Y}_{t-l} \cup \cdots \cup \boldsymbol{Y}_{t-l} \sim (l+1)L(I - lL)^{-1}, (l+1)K. \tag{6.36}$$

In particular,

$$\boldsymbol{Y}_t \cup \boldsymbol{Y}_{t-1} \cup \cdots \cup \boldsymbol{Y}_{t-p} \sim (p+1)L(I - pL)^{-1}, (p+1)K. \tag{6.37}$$

6.1.5 Higher Order M-kDPP

We now consider a pth order M-kDPP. For $l = 1, \ldots, p+1$, we start by defining the initial and transition distributions:

$$\mathcal{P}(\boldsymbol{Y}_1 = A_1) = \frac{\sum_{|A|=pk} \det(L_{A_1 \cup A})}{\binom{(p+1)k}{k} \sum_{|B|=(p+1)k} \det(L_B)} \tag{6.38}$$

$$\begin{aligned}&\mathcal{P}(\boldsymbol{Y}_t = A_t | \boldsymbol{Y}_{t-1} = A_{t-1}, \ldots, \boldsymbol{Y}_{t-l} = A_{t-l}) \\ &\quad = \frac{\sum_{|A|=(p-l)k} \det(L_{A_t \cup \cdots \cup A_{t-l} \cup A})}{\binom{(p-l+1)k}{k} \sum_{|B|=(p-l+1)k} \det(L_{A_{t-1} \cup \cdots \cup A_{t-l} \cup B})},\end{aligned} \tag{6.39}$$

where A and A_i, $i = t, t-1, \ldots, t-l$ are all disjoint. Then, jointly for $l = 1, \ldots, p+1$,

$$\mathcal{P}(\boldsymbol{Y}_l = A_l, \boldsymbol{Y}_{l-1} = A_{l-1}, \ldots, \boldsymbol{Y}_1 = A_1) = \frac{\sum_{|A|=(p-l)k} \det(L_{A_t \cup \cdots \cup A_{t-l} \cup A})}{\left[\prod_{i=0}^{l} \binom{(p-i+1)k}{k}\right] \sum_{|B|=(p+1)k} \det(L_B)}, \tag{6.40}$$

from which we confirm the stationarity of the process:

$$\mathcal{P}(\boldsymbol{Y}_l = A_l) = \frac{\sum_{|A|=pk} \det(L_{A_t \cup A})}{\binom{(p+1)k}{k} \sum_{|B|=(p+1)k} \det(L_B)} \ . \tag{6.41}$$

By induction, the above marginal holds for all t. The implied union process has margins

$$\begin{aligned}
&\mathcal{P}(\boldsymbol{Z}_t \equiv \boldsymbol{Y}_t \cup \boldsymbol{Y}_{t-1} \cup \boldsymbol{Y}_{t-l} = C_{l+1}) \\
&= \sum_{C_1 \subseteq C_2, |C_1|=k} \sum_{C_2 \subseteq C_3, |C_2|=2k} \cdots \sum_{C_l \subseteq C_{l+1}, |C_l|=lk} \mathcal{P}(\boldsymbol{Y}_t = C_{l+1} \setminus C_l, \ldots, \boldsymbol{Y}_{t-l} = C_1) \\
&= \sum_{C_1 \subseteq C_2, |C_1|=k} \sum_{C_2 \subseteq C_3, |C_2|=2k} \cdots \sum_{C_l \subseteq C_{l+1}, |C_l|=lk} \frac{\sum_{|A|=(p-l)k} \det(L_{C_{l+1}})}{\left[\prod_{i=0}^{l} \binom{(p-i+1)k}{k}\right] \sum_{|B|=(p+1)k} \det(L_B)} \\
&= \frac{\sum_{|A|=(p-l)k} \det(L_{C_{l+1} \cup A})}{\binom{(p-l+1)k}{k} \sum_{|B|=(p+1)k} \det(L_B)} \ . \qquad (6.42)
\end{aligned}$$

In particular,

$$\mathcal{P}(\boldsymbol{Z}_t \equiv \boldsymbol{Y}_t \cup \boldsymbol{Y}_{t-1} \cup \boldsymbol{Y}_{t-p} = C) = \frac{\det(L_C)}{\sum_{|B|=(p+1)k} \det(L_B)} \ , \qquad (6.43)$$

which is a k-DPP. So in summary, marginally, $\boldsymbol{Y}_t$ is stationary, distributed as a *thinned-kDPP*.

The union process of $l+1$ consecutive sets (with $l = 1, \ldots, p$) is also stationary, distributed as another *thinned-kDPP*.

Finally, the union process of $p+1$ consecutive sets is stationary, distributed as a $(p+1)k$DPP.

6.1.6 Sampling from Higher Order M-DPPs and M-kDPPs

To sample from a higher order MDPP, we first sample, we first sample Y_1 from a DPP with kernel L. We then sequentially select Y_t given the previous samples using

$$L^{(t)} = \left((M^{(t-1)} + I_{\mathcal{Y} \setminus Y_{t-1} \cup \cdots \cup Y_1})^{-1}_{\mathcal{Y} \setminus Y_{t-1} \cup \cdots \cup Y_1}\right)^{-1} - I \ . \qquad (6.44)$$

for $t = 2, \ldots, p+1$. From then onwards, we sample the subsequent sets using the kernel

$$L^{(t)} = \left((M^{(p)} + I_{\mathcal{Y} \setminus Y_{t-1} \cup \cdots \cup Y_{t-p}})^{-1}_{\mathcal{Y} \setminus Y_{t-1} \cup \cdots \cup Y_{t-p}} \right)^{-1} - I \ . \tag{6.45}$$

We highlight this in Alg. 13.

To sample from a pth order Markov k-DPP, we first draw from the initial distribution by sampling from a $(p+1)k$-DPP with kernel L and throwing away pk of the resulting items at random. Subsequently, for $t = 2, \ldots, p+1$ we sample from a $(p-t+2)k$DPP with the conditional kernel given by

$$L^{(t)} = \left((L + I_{\mathcal{Y} \setminus Y_{t-1} \cup \ldots \cup Y_1})^{-1}_{\mathcal{Y} \setminus Y_{t-1} \cup \ldots \cup Y_1} \right)^{-1} - I \ . \tag{6.46}$$

and we randomly throw away $(p-t+1)k$ of the resulting samples at random. From then on, we sample from a k-DPP with conditional kernel

$$L^{(t)} = \left((L + I_{\mathcal{Y} \setminus Y_{t-1} \cup \cdots \cup Y_{t-p}})^{-1}_{\mathcal{Y} \setminus Y_{t-1} \cup \cdots \cup Y_{t-p}} \right)^{-1} - I \ . \tag{6.47}$$

We highlight this in Alg. 14.

6.2 Learning User Preferences

A broad class of problems suited to M-(k)DPP modeling are also applications in which we would like to learn preferences from a user over time. Recall the news headlines scenario. Here, the goal is to present articles on a daily basis that are both relevant to the user's interests and also non-redundant. With feedback from a user in the form of click-through behavior, we can attempt to simultaneously learn features of the articles that the user regards as preferable. While the diversity

Algorithm 13 Sampling from a pth Order Markov DPP

```
Input: matrix $L$
$Y_1 \leftarrow$ DPP-SAMPLE($L$) using Alg. 1
for $t = 2, \ldots, p+1$ do
    $M^{(t-1)} \leftarrow L(I - (t-1)L)^{-1}$
    $L^{(t)} \leftarrow \Big((M + I_{\mathcal{Y}\setminus Y_{t-1}\cup,\ldots\cup Y_1})^{-1}_{\mathcal{Y}\setminus Y_{t-1}\cup\ldots\cup Y_1}\Big)^{-1} - I$
    $Y_t \leftarrow$ DPP-SAMPLE($L^{(t)}$) using Alg. 1
$M^{(p)} \leftarrow L(I - pL)^{-1}$
for $t = p+2, \ldots, T$ do
    $L^{(t)} \leftarrow \Big((M^{(p)} + I_{\mathcal{Y}\setminus Y_{t-1}\cup,\ldots\cup Y_{t-p}})^{-1}_{\mathcal{Y}\setminus Y_{t-1}\cup\ldots\cup Y_{t-p}}\Big)^{-1} - I$
    $Y_t \leftarrow$ DPP-SAMPLE($L^{(t)}$) using Alg. 1
Output: $\{Y_t\}$
```

Algorithm 14 Sampling from a Higher Order Markov kDPP

```
Input: matrix $L$, size $k$
$Z_1 \leftarrow k$DPP-SAMPLE($L, (p+1)k$) using Alg. 2
$Y_1 \leftarrow$ random $k$ of $Z_1$
for $t = 2, \ldots, p+1$ do
    $L^{(t)} \leftarrow \Big((L + I_{\mathcal{Y}\setminus Y_{t-1}\cup,\ldots\cup Y_1})^{-1}_{\mathcal{Y}\setminus Y_{t-1}\cup,\ldots\cup Y_1}\Big)^{-1} - I$
    $Z_t \leftarrow k$DPP-SAMPLE($L^{(t)}, (p-t+2)k$) using Alg. 2
    $Y_t \leftarrow$ random $k$ of $Z_t$
for $t = p+2, \ldots, T$ do
    $L^{(t)} \leftarrow \Big((L + I_{\mathcal{Y}\setminus Y_{t-1}\cup,\ldots\cup Y_{t-p}})^{-1}_{\mathcal{Y}\setminus Y_{t-1}\cup,\ldots\cup Y_{t-p}}\Big)^{-1} - I$
    $Z_t \leftarrow k$DPP-SAMPLE($L^{(t)}, k$) using Alg. 2
Output: $\{Y_t\}$
```

offered by a M-DPP is intrinsically valuable for this task, e.g., to keep the user from getting bored, in the context of learning it also has an important secondary benefit: it promotes exploration of the preference space.

Consider the following simple learning setup. At each time step t, the algorithm shows the user a set of k items drawn from some base set $\mathcal{Y}_t$, for instance, articles from the day's news. The user then provides feedback by identifying each shown item as either preferred or not preferred, perhaps by clicking on the preferred ones. The algorithm then incorporates this feedback and proceeds to the next round. The learner has two goals. First, as often as possible at least some of the items shown to the user should be preferred. Second, over the long term, many different items preferred by the user should be shown. In other words, the algorithm should not focus on a small set of preferred items.

Perhaps the most important consideration in this framework is balancing showing articles that the user is known to like (*exploitation*) against showing a variety of articles so as to discover new topics in which the user is also interested (*exploration*). Neither extreme is likely to be successful. However, using the L-ensemble kernel decomposition in Eq. (2.11), a DPP seeks to propose sets of items that are simultaneously high quality and diverse. The M-DPP takes this a step further and encourages diversity from step to step while maintaining DPP margins, exposing the user to an even greater variety of items without significantly sacrificing quality. Thus, we might expect that M-(k)DPPs can be used to enable fast and successful learning in this setting.

The tradeoff between exploration and exploitation is a fundamental issue for interactive learning, and has received extensive treatment in the literature on multi-armed bandits. However, our setup is relatively unusual for two reasons. First, we show multiple items per time step, sometimes called the *multiple plays* setting [Anantharam et al., 1987]. Second, we use feature vectors to describe the items we choose, allowing us to generalize to unseen items (e.g., new articles); this is a special case of *contextual bandits* [Langford and Zhang, 2007]. Each of these scenarios has received some attention on its own, but it is only in combination that a notion of diversity becomes relevant, since we have both the need to select multiple items as well as a basis for relating them. This combination has been considered recently by Yue and Guestrin [2011], who showed an algorithm that yields bounded regret under the assumption that the reward function is submodular. Here, on the other hand, our goal is primarily to illustrate the empirical effects on learning when the items shown at each time step are sampled from a M-DPP. To that end, we propose a very simple quality learning algorithm that appears to work well in practice. Whether formal regret guarantees can be established for learning with M-DPPs is an open question for future work.

6.2.1 Setup

To naturally accommodate user feedback and transfer knowledge across items, we will consider algorithms that learn a log-linear quality model assigning item i the score

$$q_i = \exp(\theta^\top f_i) \ , \tag{6.48}$$

where $f_i \in \mathbb{R}^m$ is a known feature vector for item i and $\theta \in \mathbb{R}^m$ is the parameter vector to be learned. Learning iterates between two distinct steps: (1) sampling articles according to the current quality scores and (2) using user feedback to revise the quality scores via updates to θ.

Let $\theta^{(t)}$ denote the parameter vector prior to time step t and let $q_i^{(t)}$ denote the corresponding quality scores for the items $i \in \mathcal{Y}_t$. We initialize $\theta^{(1)} = \mathbf{0}$ so that $q_i^{(1)} = 1$ for all $i \in \mathcal{Y}_1$. (At this point we are effectively in a purely exploratory mode.) Denote the items preferred by the user at iteration t by $\{a_i^{(t)}\}_{i=1}^{R_t}$, and the non-preferred items by $\{b_i^{(t)}\}_{i=1}^{S_t}$. Inspired by standard online algorithms, we define the parameter update rule as follows:

$$\theta^{(t+1)} \leftarrow \theta^{(t)} + \eta \left(\frac{1}{R_t} \sum_{i=1}^{R_t} f_{a_i^{(t)}} - \frac{1}{S_t} \sum_{i=1}^{S_t} f_{b_i^{(t)}} \right) \tag{6.49}$$

That is, we add to θ the average features of the preferred items, and subtract from θ the average features of non-preferred items. This increases the quality of the former and decreases the quality of the latter. η is a learning rate hyperparameter. We can then proceed to the next time step, computing the new quality scores $q_i^{(t+1)} = \exp(\theta^{(t+1)\top} f_i)$ for each $i \in \mathcal{Y}_{t+1}$.

The updated quality scores are then used to select subsequent items to be shown to the user. In order to separate the challenges of learning the quality scores, which is not our primary interest, from the benefits of incorporating the M-DPP, we consider five sampling methods:

- **Uniform.** We ignore the quality scores and choose k items uniformly at random without replacement.

- **Weighted.** We draw k items with probabilities proportional to their quality scores without replacement.

- ***k*DPP.** We sample the set of items from a kDPP with L-ensemble kernel L given by the decomposition in Eq. (2.11), where ϕ is fixed in advance and q_i are the current quality scores.

- ***k*DPP + heuristic (threshold).** We sample the set of items from a kDPP after removing articles whose similarity to the previously selected articles exceeds a predetermined threshold. At threshold > 1, the heuristic is equivalent to the kDPP.

- **M-*k*DPP.** We sample the set of items from the first order M-kDPP transition distribution given the items selected at the previous time step. The L-ensemble transition kernel is as in Eq. (6.22), with L defined as for the kDPP.

The learning algorithm is summarized in Alg. 15.

Algorithm 15 Interactive learning of quality scores

```
Input: learning rate $\eta$
$\theta^{(1)} \leftarrow \mathbf{0}$
for $t = 1, 2, \ldots$ do
  $q_i^{(t)} \leftarrow \exp(\theta^{(t)\top} f_i) \quad \forall i \in \mathcal{Y}_t$
  Select items to display given $q_i^{(t)}$
    (using one of the methods described in Sec. 6.2.1)
  Receive user feedback $\{a_i^{(t)}\}_{i=1}^{R_t}$ and $\{b_i^{(t)}\}_{i=1}^{S_t}$
  $\theta^{(t+1)} \leftarrow \theta^{(t)} + \eta \left( \frac{1}{R_t} \sum_{i=1}^{R_t} f_{a_i^{(t)}} - \frac{1}{S_t} \sum_{i=1}^{S_t} f_{b_i^{(t)}} \right)$
```

6.2.2 Likelihood Based Alternatives

Instead of the additive learning rule proposed above, one could instead take advantage of the probabilistic nature of the M-DPP and perform likelihood-based learning, which has associated theoretical guarantees. In particular, based on a sequence of user feedback, we could solve for the penalized DPP maximum likelihood estimate of $q^{(t)} = [q_1^{(t)}, \ldots, q_N^{(t)}]$ as:

$$\arg\max_q \prod_{t=1}^{t} \mathcal{P}_q(\{a_i^{(t)}\} \subseteq \mathbf{Y}_t, \{b_i^{(t)}\} \cap \mathbf{Y}_t = \emptyset) + \lambda ||q||_2, \tag{6.50}$$

where $\mathcal{P}_q$ is a DPP with L-ensemble kernel defined by quality scores q and λ is a regularization parameter. We have

$$\begin{aligned} &\mathcal{P}_q(\{a_i^{(t)}\} \subseteq \mathbf{Y}_t, \{b_i^{(t)}\} \cap \mathbf{Y}_t = \emptyset) \\ &\quad = \Big(1 - \mathcal{P}_q(\{b_i^{(t)}\} \subseteq \mathbf{Y}_t \mid \{a_i^{(t)}\} \subseteq \mathbf{Y}_t)\Big) \cdot \mathcal{P}_q(\{a_i^{(t)}\} \subseteq \mathbf{Y}_t), \end{aligned} \tag{6.51}$$

which has computable terms in a DPP given the quality scores q. The M-DPP has obvious extensions. However, in both cases the objective function is not convex so computations are intensive and only converge to local maxima. Due to its simplicity and good performance in practice (see Sec. 6.3.2), we use the heuristic algorithm described previously for illustrating the behavior of the M-DPP.

6.3 Experiments

We study the performance of the M-kDPP for selecting daily news items from a selection of over 35,000 New York Times newswire articles obtained between January and June of 2005 as part of the Gigaword corpus [Graff and Cieri, 2009]. On each day of a given week, we display 10 articles from a base set of the roughly 1400 articles written that week. This process is repeated for each of the 26 weeks in our dataset. The goal is to choose a collection of articles that is high quality but also diverse, both marginally and between time steps.

To examine performance in the absence of confounding issues of quality learning, we first consider a scenario in which the quality scores are fixed. Here, we measure both the diversity and quality of articles chosen each day by the different methods. We then turn to quality learning based on user feedback to examine how the properties of the M-kDPP influence the discovery of a user's preferences.

6.3.1 Fixed Quality

Similarity To generate similarity features ϕ_i, we first compute standard normalized term frequency - inverse document frequency (tf-idf) vectors for document i. For each term, t, in the corpus, we computed the term frequency:

$$TF_i(t) = \frac{\text{Number of times term } t \text{ appears in document } i}{\text{Number of terms in document } i} \tag{6.52}$$

and the inverse document frequency:

$$IDF_i(t) = \log\left(\frac{\text{Total number of documents corpus}}{\text{Number of documents with term } t \text{ in it}}\right). \tag{6.53}$$

The element t of the tf-idf vectors for document i is then given by

$$TF\text{-}IDF_i(t) = TF_i(t) \times IDF_i(t). \tag{6.54}$$

The idf scores are computed across all 26 weeks worth of articles. We then compute the cosine similarity between all pairs of articles. Due to the sparsity of the tf-idf vectors, these similarity scores tend to be quite low, leading to poor diversity if used directly as a kernel matrix. Instead, we let the similarity features be given by binary vectors where the jth coordinate of ϕ_i is 1 if article j is among the 150 nearest neighbors of article i in that week based on our cosine distance metric, and 0 otherwise.

Quality In the fixed scenario, we need a way to assign quality scores to articles. A natural approach is to score articles based on their proximity to the other articles; this way, an article that is close to many others (as measured by cosine similarity) is considered to be of high quality (i.e. articles in popular topics). In this data set, for example, we find that there is a large cluster of articles that talk about *politics* and articles that fall under this topic generally have much higher quality than articles that talk about, say, *food*. To model this, we compute quality scores as $q_i = \exp(\alpha d_i)$, where d_i is the sum of the cosine similarities between article i and all other articles in our collection and α is a hyperparameter that determines the dynamic range. We chose $\alpha = 5$ for our data set, although a range of values gave qualitatively similar results.

For each method, we sample sets of articles on a daily basis for each of the 26 weeks. To measure diversity within a time step, we compute the average cosine similarity between articles chosen on a given day. We then subtract the result from 1 so that larger values correspond to greater diversity. Diversity between

Table 6.1: Average Diversity and Quality of Selected Articles

Method	Marginal diversity	1-step diversity	2-step diversity	Quality
M-kDPP	0.899	0.849	0.843	0.654
k-DPP	0.896	0.786	0.779	0.668
k-DPP + heuristic (0.4)	0.904	0.849	0.804	0.651
k-DPP + heuristic (0.2)	0.946	0.891	0.889	0.587
Weighted Rand.	0.750	0.681	0.677	0.756
Uniform Rand.	0.975	0.949	0.947	0.457

time steps is obtained by measuring the average cosine similarity between each article at time t and the single most similar article at time $t+1$ (or $t+2$ for 2-step diversity), and again subtracting the result from 1. We also report the average quality score of the articles chosen across all 182 days. All measures are averaged over 100 random runs; statistical significance is computed by bootstrapping.

Table 6.3.2 displays the results for all methods. The M-kDPP shows a marked increase in between-step diversity, on average, compared to the kDPP and weighted random sampling. All of the differences are significant at 99% confidence. The average marginal diversities for the M-kDPP and kDPP are statistically significantly higher than for weighted random sampling, but are not statistically significantly different from each other. This is to be expected since, as we have seen in Sec. 6.1, the marginal distribution for the M-kDPP does not greatly differ from the kDPP process. On the other hand, the uniform sampling shows much higher diversity than the other methods, which can be attributed to the fact that it is a purely exploratory method that ignores the quality of the articles it chooses.

Table 6.3.2 also shows the average quality of the selected articles. The weighted random sampling chooses, on average, higher quality articles compared to the rest of the methods since it does not have to balance issues of diversity within the set. The kDPP on average chooses slightly higher quality articles than the M-kDPP,

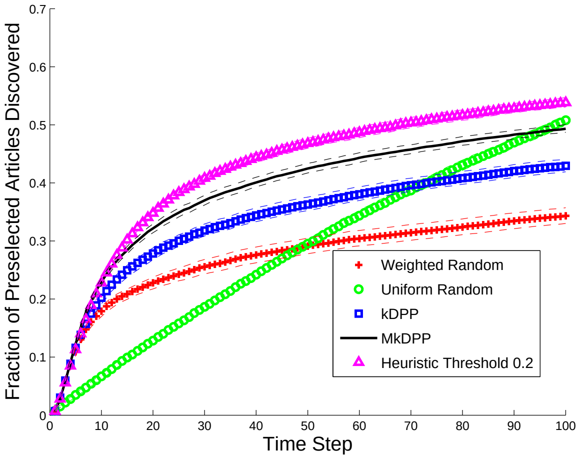


Figure 6.6: Performance of the methods at recovering the preselected preferred articles. Solid lines indicate the mean over 100 random runs, and dashed lines indicate the corresponding confidence intervals, computed by bootstrapping.

perhaps due to the additional between-step diversity sought by the M-kDPP; however, the difference is not statistically significant. It is evident from Table 6.3.2 that the M-kDPP achieves a balance between the diversity of the articles it chooses (both marginally and across time steps) and their quality.

As for the kDPP + heuristic baseline, our experiments show that by tuning the threshold carefully we can mimic the performance of the M-kDPP, but without the associated probabilistic interpretation and theoretical properties. When the threshold is too low, quality degrades significantly.

6.3.2 Learning Preferences

We also study the performance of the M-kDPP when learning from user feedback, as outlined in Sec. 6.2. For simplicity, we use only a week's worth of news articles (1427 articles). To create feature vectors, we first generate topics by running LDA on the entire corpus [Blei et al., 2003]. We then manually label the most prevalent 10 topics as *finance*, *health*, *politics*, *world news*, *baseball*, *football*, *arts*, *technology*, *entertainment*, and *justice*, and associate each article with its LDA-inferred mixture of these topics (a 10-dimensional feature vector f_i). We define a synthetic user by a sparse topic preference vector (0.7 for *finance*, 0.2 for *world news*, 0.1 for *politics*, and 0 for all other topics), and preselect as "preferred" the 200 articles whose feature vectors f_i maximize the dot product with the user preference vector.

Similar to our previous experiment, we define the similarity features between articles to be binary vectors based on 50 nearest neighbors using the tf-idf cosine distances. The quality is defined as in Sec. 6.2, $q_i^{(t)} = \exp(\theta^{(t)\top} f_i)$, where f_i is the feature vector of article i (based on the mixture of topics) normalized to sum to 1. We set the learning rate $\eta = 2$; however, varying η did not change the qualitative behavior of each method, only the time scale at which these behaviors became noticeable. We also note that although we base the similarity on 50 nearest neighbors, the results were not sensitive to the size of this neighborhood.

The goal of this experiment is to illustrate how the different methods balance between exploring the space of all articles to discover the 200 preselected articles (*recall*) and exploiting a learned set of features to keep showing preferred articles (*precision*). On one end of the spectrum, uniform sampling simply explores the space of articles without taking advantage of the user feedback, leading to high recall and low precision. On the other end, the weighted random sampling fully exploits the learned preference in selecting articles, but does not have a mechanism to encourage exploration. We demonstrate that the M-kDPP balances these two extremes, taking advantage of the user feedback while also exploring diverse articles.

Results We use each method to select 10 articles per day over a period of 100 days, using the current quality scores $q^{(t)}$ on each day t. We measure recall by keeping track of the fraction of preselected preferred articles (out of the 200 total) that have been displayed so far. We also compute, out of the 10 articles shown on

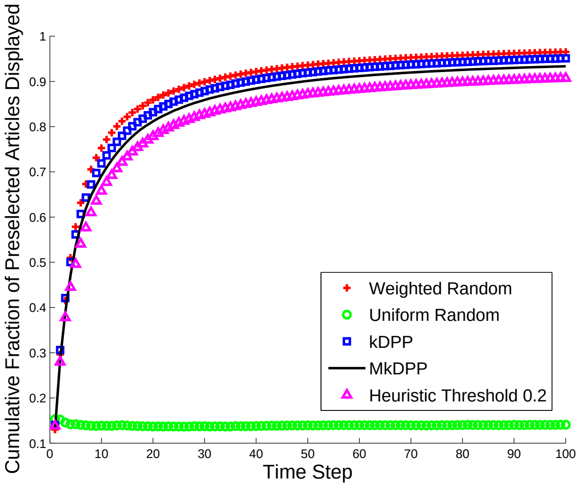


Figure 6.7: Cumulative fraction of preferred articles displayed to the user.

a given day, the fraction that are preferred. This serves as a measure of precision. All measures are averaged over 100 random runs.

Figure 6.6 shows the recall performance of the methods we tested. Uniform sampling discovers the articles at a somewhat linear rate of about 5% per day; given a larger base set relative to the size of the preferred set, however, we would expect a slower rate of discovery. The methods that incorporate user feedback discover a larger set of preferred articles more rapidly by harnessing learned features of the user's interests. The M-kDPP dominates both the kDPP and weighted random sampling in this metric since it encourages exploration by introducing both marginal and between-step diversity of displayed articles. In contrast, the kDPP does not penalize repeating similar marginally diverse sets and the weighted random sampling does not have any explicit mechanism for exploration. It takes uniform random sampling nearly 100 time steps to discover the same number of unique preferred articles as the M-kDPP. For the sake of clarity, we omit the results of kDPP + heuristic with threshold 0.4 since they are not statistically significantly different from the M-kDPP. Figure 6.8 shows the headlines of the articles shown

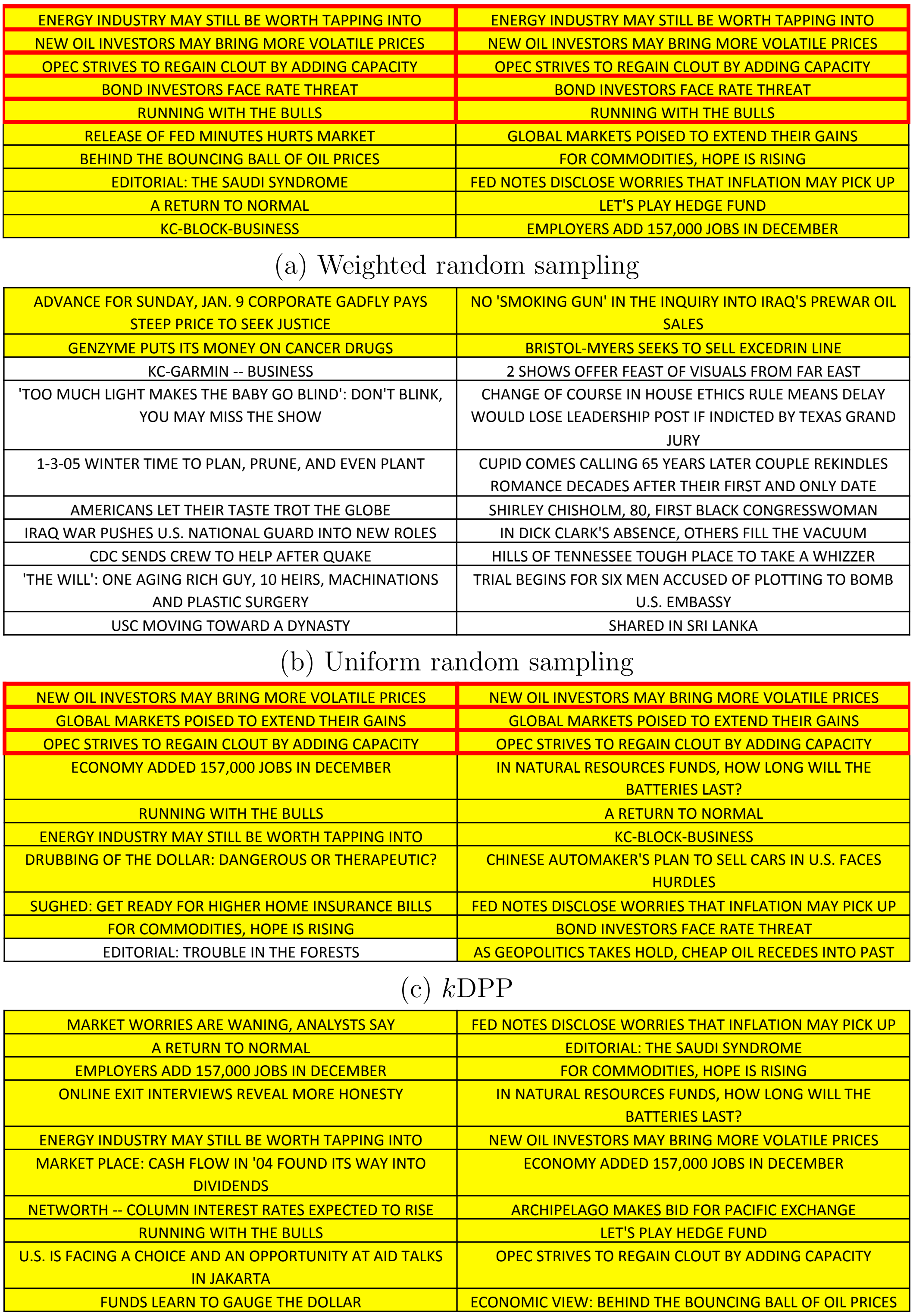
Method	Avg. marginal diversity	Between-step diversity
M-kDPP	0.766	0.738
k-DPP	0.753	0.512
Weighted Rand.	0.716	0.362
Uniform Rand.	0.975	0.964

Table 6.2: Diversity of the articles chosen on days 99 and 100.

on days 99 and 100 by all four methods during a single randomly selected trial. Yellow highlighting indicates the article headlines that are in the preselected pool of preferred articles, while the red borders indicate articles that appear on two consecutive days.

We observe that weighted random sampling has high precision, with all articles displayed in both days coming from the preselected pool. Notice, however, that five articles are repeated within the the two days. Compare this with uniform sampling, which did not display any articles twice in the two days, but only manages to display two preselected articles each day. The kDPP has high precision, but again suffers from a similar repeating of articles as seen in the weighted random sampling. The M-kDPP on the other hand does a better job overall, having a high precision for the two days without any repeat articles. Table 6.2 shows the marginal diversity and the between-step diversity on those two days. We observe M-kDPP chooses more diverse articles compared to the kDPP and weighted random sampling.

Fig. 6.7 shows the cumulative fraction of displayed articles that were preferred, reflecting precision. All methods besides uniform sampling quickly achieve high precision. Weighted random sampling displays the largest number of preferred articles per day, almost always having precision of at least 0.9. However, as we have observed, this large precision is at the cost of lower recall. In particular, weighted random sampling quickly homes in on features related to a small subset



ENERGY INDUSTRY MAY STILL BE WORTH TAPPING INTO	ENERGY INDUSTRY MAY STILL BE WORTH TAPPING INTO
NEW OIL INVESTORS MAY BRING MORE VOLATILE PRICES	NEW OIL INVESTORS MAY BRING MORE VOLATILE PRICES
OPEC STRIVES TO REGAIN CLOUT BY ADDING CAPACITY	OPEC STRIVES TO REGAIN CLOUT BY ADDING CAPACITY
BOND INVESTORS FACE RATE THREAT	BOND INVESTORS FACE RATE THREAT
RUNNING WITH THE BULLS	RUNNING WITH THE BULLS
RELEASE OF FED MINUTES HURTS MARKET	GLOBAL MARKETS POISED TO EXTEND THEIR GAINS
BEHIND THE BOUNCING BALL OF OIL PRICES	FOR COMMODITIES, HOPE IS RISING
EDITORIAL: THE SAUDI SYNDROME	FED NOTES DISCLOSE WORRIES THAT INFLATION MAY PICK UP
A RETURN TO NORMAL	LET'S PLAY HEDGE FUND
KC-BLOCK-BUSINESS	EMPLOYERS ADD 157,000 JOBS IN DECEMBER

(a) Weighted random sampling

ADVANCE FOR SUNDAY, JAN. 9 CORPORATE GADFLY PAYS STEEP PRICE TO SEEK JUSTICE	NO 'SMOKING GUN' IN THE INQUIRY INTO IRAQ'S PREWAR OIL SALES
GENZYME PUTS ITS MONEY ON CANCER DRUGS	BRISTOL-MYERS SEEKS TO SELL EXCEDRIN LINE
KC-GARMIN -- BUSINESS	2 SHOWS OFFER FEAST OF VISUALS FROM FAR EAST
'TOO MUCH LIGHT MAKES THE BABY GO BLIND': DON'T BLINK, YOU MAY MISS THE SHOW	CHANGE OF COURSE IN HOUSE ETHICS RULE MEANS DELAY WOULD LOSE LEADERSHIP POST IF INDICTED BY TEXAS GRAND JURY
1-3-05 WINTER TIME TO PLAN, PRUNE, AND EVEN PLANT	CUPID COMES CALLING 65 YEARS LATER COUPLE REKINDLES ROMANCE DECADES AFTER THEIR FIRST AND ONLY DATE
AMERICANS LET THEIR TASTE TROT THE GLOBE	SHIRLEY CHISHOLM, 80, FIRST BLACK CONGRESSWOMAN
IRAQ WAR PUSHES U.S. NATIONAL GUARD INTO NEW ROLES	IN DICK CLARK'S ABSENCE, OTHERS FILL THE VACUUM
CDC SENDS CREW TO HELP AFTER QUAKE	HILLS OF TENNESSEE TOUGH PLACE TO TAKE A WHIZZER
'THE WILL': ONE AGING RICH GUY, 10 HEIRS, MACHINATIONS AND PLASTIC SURGERY	TRIAL BEGINS FOR SIX MEN ACCUSED OF PLOTTING TO BOMB U.S. EMBASSY
USC MOVING TOWARD A DYNASTY	SHARED IN SRI LANKA

(b) Uniform random sampling

NEW OIL INVESTORS MAY BRING MORE VOLATILE PRICES	NEW OIL INVESTORS MAY BRING MORE VOLATILE PRICES
GLOBAL MARKETS POISED TO EXTEND THEIR GAINS	GLOBAL MARKETS POISED TO EXTEND THEIR GAINS
OPEC STRIVES TO REGAIN CLOUT BY ADDING CAPACITY	OPEC STRIVES TO REGAIN CLOUT BY ADDING CAPACITY
ECONOMY ADDED 157,000 JOBS IN DECEMBER	IN NATURAL RESOURCES FUNDS, HOW LONG WILL THE BATTERIES LAST?
RUNNING WITH THE BULLS	A RETURN TO NORMAL
ENERGY INDUSTRY MAY STILL BE WORTH TAPPING INTO	KC-BLOCK-BUSINESS
DRUBBING OF THE DOLLAR: DANGEROUS OR THERAPEUTIC?	CHINESE AUTOMAKER'S PLAN TO SELL CARS IN U.S. FACES HURDLES
SUGHED: GET READY FOR HIGHER HOME INSURANCE BILLS	FED NOTES DISCLOSE WORRIES THAT INFLATION MAY PICK UP
FOR COMMODITIES, HOPE IS RISING	BOND INVESTORS FACE RATE THREAT
EDITORIAL: TROUBLE IN THE FORESTS	AS GEOPOLITICS TAKES HOLD, CHEAP OIL RECEDES INTO PAST

(c) kDPP

MARKET WORRIES ARE WANING, ANALYSTS SAY	FED NOTES DISCLOSE WORRIES THAT INFLATION MAY PICK UP
A RETURN TO NORMAL	EDITORIAL: THE SAUDI SYNDROME
EMPLOYERS ADD 157,000 JOBS IN DECEMBER	FOR COMMODITIES, HOPE IS RISING
ONLINE EXIT INTERVIEWS REVEAL MORE HONESTY	IN NATURAL RESOURCES FUNDS, HOW LONG WILL THE BATTERIES LAST?
ENERGY INDUSTRY MAY STILL BE WORTH TAPPING INTO	NEW OIL INVESTORS MAY BRING MORE VOLATILE PRICES
MARKET PLACE: CASH FLOW IN '04 FOUND ITS WAY INTO DIVIDENDS	ECONOMY ADDED 157,000 JOBS IN DECEMBER
NETWORTH -- COLUMN INTEREST RATES EXPECTED TO RISE	ARCHIPELAGO MAKES BID FOR PACIFIC EXCHANGE
RUNNING WITH THE BULLS	LET'S PLAY HEDGE FUND
U.S. IS FACING A CHOICE AND AN OPPORTUNITY AT AID TALKS IN JAKARTA	OPEC STRIVES TO REGAIN CLOUT BY ADDING CAPACITY
FUNDS LEARN TO GAUGE THE DOLLAR	ECONOMIC VIEW: BEHIND THE BOUNCING BALL OF OIL PRICES

(d) M-kDPP

Figure 6.8: Headlines selected on days 99 (left) and 100 (right). Yellow highlighting indicate articles from the preselected pool of preferred articles and red boxes outline repeated articles chosen on both days.

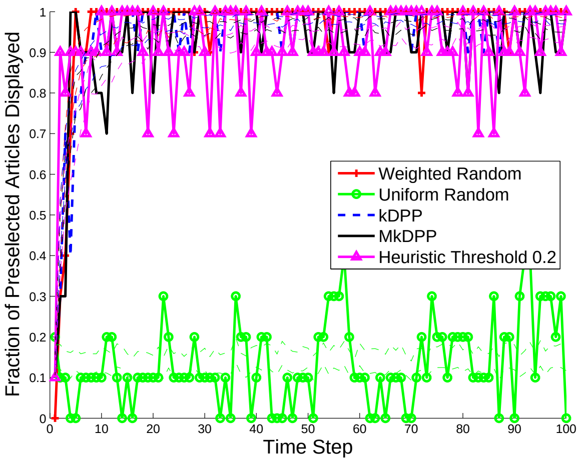


Figure 6.9: The fraction of preferred articles displayed to user at each time step for a single trial.

of preferred articles, thereby increasing the probability of them being repeatedly selected with no force to counteract this behavior. As expected, by only requiring marginal diversity, the kDPP achieves slightly higher precision than the M-kDPP on average (both typically above 0.8), but again at the cost of reduced exploration. Overall, the differences in precision between these methods are not large. In many applications, having 8 out of 10 results preferred may be more than sufficient. For a single trial, Fig. 6.9 shows the fraction of articles displayed to the user at each time step that are in the preselected pool.

Finally, to examine the balance between exploration and exploitation, we compute a metric based on the idea of marginally decreasing utility. Under this metric, at every time step, the user experiences a utility of 1 for each preferred article shown for the first time. If a previously displayed preferred article is once again chosen, the user gets a utility of $\frac{1}{l+1}$ where l is the number of times that article has appeared in the past. The underlying assumption is that a user benefits from seeing preferred articles, but in decreasing amounts as the same articles are

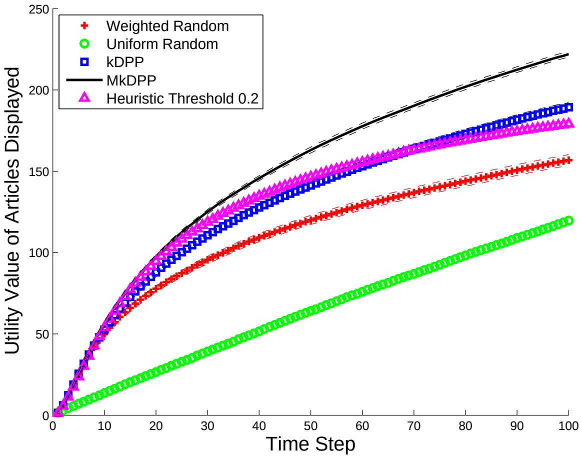


Figure 6.10: Performance measured by a marginally decreasing utility function.

repeatedly displayed. Figure 6.10 shows the performance of the methods under this utility metric; the M-kDPP scores highest.

6.4 Conclusion

We introduced the Markov DPP, a combinatorial process for modeling diverse sequences of subsets. By establishing the theoretical properties of this process, such as stationary DPP margins and a DPP union process, we showed how our construction yields sets that are diverse at each time step as well as diverse jointly for p consecutive time-steps, making it appropriate for interactive tasks like news recommendation. Additionally, by explicitly connecting with DPPs, further properties of M-DPPs are straightforwardly derived, such as the marginal and conditional expected set cardinality.

We showed how to efficiently sample from a M-DPP, and found empirically that the model achieves an improved balance between diversity and quality compared to baseline methods. We also studied the effects of the M-DPP on learning, finding significant improvements in recall at minimal cost to precision for a news task with user feedback.

Chapter 7

Conclusion

We have presented a variety of algorithms, extensions and theoretical results for determinantal point processes in both discrete and continuous spaces. We believe that our contributions will enable DPPs to be used in practice for many real-world machine learning and statistical applications. In Chapter 3, we introduced low-rank approximation algorithms for sampling large-scale discrete DPPs and provided variational error bounds associated with our methods. We then extended this algorithm to the continuous case in Chapter 4, enabling sampling to be performed for a wide range of kernels that previous algorithms can't handle. We provided theoretical bounds in this case as well. In addition, for a fixed-sized kDPPs, we presented a Gibbs sampling-type algorithm that can be executed efficiently using Schur's determinantal formula. We then proposed Bayesian algorithms to learn the kernel parameters of DPPs in Chapter 5. We showed how these algorithms can be modified to enable the efficient learning of large-scale discrete and continuous DPPs, even in cases where the likelihood cannot be computed exactly but rather, can be only be bounded arbitrarily tight. In Chapter 6, we also introduced our construction of the *Markov*-DPP/kDPP that defines a stationary process that mantains individual diversity as well as diversity throughout time. Throughout this thesis, we illustrated these methodologies with real-world applications such as motion capture video summarization, repulsive mixture modelling, latent clustering of social network, human motion synthesis, diabetic neuropathy based on nerve fibers clustering, exploration of human judgement of image diversity and news articles retrieval.

7.1 Future Work

We briefly suggests a few threads for future work.

Improved Low-Rank Approximation Bounds. In this thesis, we provided analysis of variational error by bounding the error in probability distribution for each possible subset. These bounds rely heavily on the condition that the resulting error kernel, $E(\boldsymbol{x}, \boldsymbol{y}) = L(\boldsymbol{x}, \boldsymbol{y}) - \tilde{L}(\boldsymbol{x}, \boldsymbol{y})$ is positive semi-definite. While many approximation techniques such as Nyström and power iteration methods satisfy this condition, others such as random Fourier features approximation do not. Future work includes developing bounds that do not assume this condition. Furthermore, the bounds we presented can possibly be improved by considering the properties of the kernels beyond their eigenvalues. For example, the full characteristics of their eigenstructure, including properties such as their coherence, can be possibly be harnessed to provide tighter bounds. It is not clear at this point how they can be incorporated and this leaves an interesting avenue for future research.

Hierarchical and Multiresolution DPP. Thus far, we considered models that separate the points into repulsed clusters using the repulsive mixture models. An interesting thread of research is in considering a hierarchical DPPs such that diverse data points can be sampled from a repulsed set of clusters. An example of a real life application is sampling of a diverse set of news articles generated from a diverse set of topics. Another possible thread is in considering multiresolution models where points cluster if they are within some range ϵ between each other but repulse when they are far apart.

Attractive Processes While DPPs are useful in modeling repulsive processes, another class of processes called *Permanental Point Processes* (PPPs) can be used to model attractive processes. So far, however, little is known about them besides their definition as being distributed proportional to the permanent of the submatrix of a kernel:

$$\mathcal{P}(A) \propto \mathrm{per}(L_A). \tag{7.1}$$

An interesting thread of future research is in developing theories and algorithms for PPPs and perhaps combine them with DPPs to build more complex models.

Time-Varying Kernel. We presented our construction of Markov DPPs to provide subset selection methods that select diverse sets across time. In our construction. however, we assumed that the kernel is time-invariant. In more realistic settings, such as in news retrieval, items stream in and out of the base collection (think in terms of news articles collection updated daily). An interesting extension would be in modeling a temporal extension that allows for this evolution in the kernel items/parameters.

Other Applications. We presented a myriad of different real world machine learning tasks in this thesis. We believe that our algorithms can be applied to problems beyond the scope of tasks done in this thesis. For example, incorporating DPPs into statistical methods such as *particle filters* and other MCMC-type algorithms where diversity is key should be explored in the future. Other machine learning tasks such as active learning would also benefit from the applications of DPPs as well. In short, we believe that there are a wide variety of interesting applications that have not previously been studied that can be enabled by our extension to large-scale, continuous and temporal DPPs.

Bibliography

P. Abbeel and A.Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proc. ICML*, 2004.

V. Anantharam, P. Varaiya, and J. Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part i: Iid rewards. *Automatic Control, IEEE Transactions on*, 32(11):968–976, 1987.

N.F. Arcolano. *Approximation of Positive Semidefinite Matrices Using the Nyström Method*. PhD thesis, Harvard University, 2011.

R. A. Bernstein and M. Gobbel. Partitioning of space in communities of ants. *Journal of Animal Ecology*, 48(3):931–942, 1979.

R. Bhatia. *Matrix analysis*, volume 169. Springer Verlag, 1997.

D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

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

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

E. Candes and J. Romberg. Sparsity and incoherence in compressive sampling. *Inverse problems*, 23(3):969, 2007.

CMU. Carnegie Mellon University graphics lab motion capture database. *http://mocap.cs.cmu.edu/*, 2009.

D.J. Daley and D. Vere-Jones. *An introduction to the theory of point processes: Volume I: Elementary theory and methods*. Springer, 2003.

Laurent Decreusefond, Ian Flint, and Kah Choon Low. Perfect simulation of determinantal point processes. *arXiv preprint arXiv:1311.1027*, 2013.

A. Deshpande, L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via volume sampling. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1117–1126. ACM, 2006.

P. Drineas and M.W. Mahoney. On the Nyström method for approximating a Gram matrix for improved kernel-based learning. *The Journal of Machine Learning Research*, 6:2153–2175, 2005.

Ky Fan. Maximum properties and inequalities for the eigenvalues of completely continuous operators. *Proceedings of the National Academy of Sciences of the United States of America*, 37(11):760, 1951.

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

Chris Fraley and Adrian E Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal*, 41(8):578–588, 1998.

A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of the ACM (JACM)*, 51(6):1025–1041, 2004.

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

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

Israel Gohberg and Kreĭ. *Introduction to the theory of linear nonselfadjoint operators.*

D. Graff and C. Cieri. English Gigaword, 2009.

K. Guan. Schur-convexity of the complete elementary symmetric function. *Journal of Inequalities and Applications*, 2006(1):67624, 2006.

Mark S Handcock, Adrian E Raftery, and Jeremy M Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170(2):301–354, 2007.

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

R. Jin, T. Yang, and M. Mahdavi. Improved bound for the Nyström's method and its application to kernel classification. *arXiv preprint arXiv:1111.2262*, 2011.

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

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

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

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

A. Kulesza and B. Taskar. Determinantal point processes for machine learning. *arXiv preprint arXiv:1207.6083*, 2012b.

S. Kumar, M. Mohri, and A. Talwalkar. Sampling methods for the Nyström method. *The Journal of Machine Learning Research*, 98888:981–1006, 2012.

J. Langford and T. Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. *Advances in Neural Information Processing Systems*, 20, 2007.

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

D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

O. Macchi. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, pages 83–122, 1975.

B. Matérn. *Spatial variation.* Springer-Verlag, 1986.

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

R. M. Neal. Slice sampling. *Annals of statistics*, pages 705–741, 2003.

T. Neeff, G. S. Biging, L. V. Dutra, C. C. Freitas, and J. R. Dos Santos. Markov point processes for modeling of spatial forest patterns in Amazonia derived from interferometric height. *Remote Sensing of Environment*, 97(4):484–494, 2005.

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

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

R.B Paris. Asymptotics of integrals of hermite polynomials. *Appl. Math. Sci. 4*, pages 3043–3056, 2010.

F. Petralia, V. Rao, and D. Dunson. Repulsive mixtures. In *NIPS*, 2012.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. *NIPS*, 2007.

S. Richardson and P. J. Green. On Bayesian analysis of mixtures with an unknown number of components (with discussion). *JRSS:B*, 59(4):731–792, 1997.

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

C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2nd edition, 2004.

Samuel F Sampson. *Crisis in a cloister*. PhD thesis, Ph. D. Thesis. Cornell University, Ithaca, 1969.

J Schur. Über potenzreihen, die im innern des einheitskreises beschränkt sind. *Journal für die reine und angewandte Mathematik*, 147:205–232, 1917.

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

M. Stephens. Dealing with label switching in mixture models. *JRSS:B*, 62(4): 795–809, 2000.

C.A. Sugar and G.M. James. Finding the number of clusters in a dataset: An information-theoretic approach. *JASA*, 98(463):750–763, 2003.

A. Talwalkar and A. Rostamizadeh. Matrix coherence and the Nyström method. *arXiv preprint arXiv:1004.2008*, 2010.

A. Vedaldi and B. Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the international conference on Multimedia*, pages 1469–1472. ACM, 2010.

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

J. Wang. Consistent selection of the number of clusters via crossvalidation. *Biometrika*, 97(4):893–904, 2010.

C.K.I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. *NIPS*, 2000.

T. Yang, Y.-F. Li, M. Mahdavi, R. Jin, and Z.-H. Zhou. Nyström method vs random fourier features: A theoretical and empirical comparison. *NIPS*, 2012.

Yisong Yue and Carlos Guestrin. Linear submodular bandits and their application to diversified retrieval. In *In Advances in Neural Information Processing Systems (NIPS)*, Granada, Spain, December 2011.

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