Using The Matrix Ridge Approximation to Speedup Determinantal Point Processes Sampling Algorithms

Shusen Wang, Chao Zhang, Hui Qian*

College of Computer Science and Technology,
Zhejiang University, Hangzhou, China
{wss,zczju,qianhui}@zju.edu.cn

Zhihua Zhang

Key Laboratory of Shanghai Education Commission
for Intelligent Interaction and Cognitive Engineering,
Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, China
zhihua@sjtu.edu.cn

Abstract

Determinantal point process (DPP) is an important probabilistic model that has extensive applications in artificial intelligence. The exact sampling algorithm of DPP requires the full eigenvalue decomposition of the kernel matrix which has high time and space complexities. This prohibits the applications of DPP from large-scale datasets. Previous work has applied the Nyström method to speedup the sampling algorithm of DPP, and error bounds have been established for the approximation. In this paper we employ the matrix ridge approximation (MRA) to speedup the sampling algorithm of DPP, showing that our approach MRA-DPP has stronger error bound than the Nyström-DPP. In certain circumstances our MRA-DPP is provably exact, whereas the Nyström-DPP is far from the ground truth. Finally, experiments on several real-world datasets show that our MRA-DPP is more accurate than the other approximation approaches.

Introduction

The determinantal point process (DPP) is a probabilistic model that defines a distribution over 2^n subsets of an item set of size n. Given an item set $[n] \triangleq \{1, \cdots, n\}$ and an $n \times n$ symmetric positive semidefinite (SPSD) kernel matrix $\mathbf{K}$, the probability measure of a subset $\mathcal{S}$ ($\subset [n]$) is proportional to the determinant of the submatrix of $\mathbf{K}$ with rows and columns indexed by $\mathcal{S}$. DPP is originated from physics where DPP is used to capture the repulsion among particles; as a result of repulsion, DPP encourages *diversity*. Specifically, DPP assigns high probabilities to subsets containing dissimilar items (Kulesza and Taskar 2012).

The diversity property of DPP can be used to solve many real-world artificial intelligence problems. For example, when applied to the text summarization problem, DPP selects a subset of sentences covering distinct aspects of an article rather than sentences focusing on one specific issue (Kulesza and Taskar 2011b). For another example, DPP can be applied to information retrieval to make the search results more diverse (Kulesza and Taskar 2011a). Diversity can be also used as a filtering prior when applied to the pose estimation task (Kulesza and Taskar 2012; Affandi, Fox, and Taskar 2013; Affandi et al. 2013).

Although DPP is a distribution over 2^n subsets, DPP has a nice property that exact sampling can be done in polynomial-time. However, the sampling algorithm devised by Hough et al. (2006) requires the full eigenvalue decomposition of the $n \times n$ kernel matrix $\mathbf{K}$, costing time $\mathcal{O}(n^3)$ and space $\mathcal{O}(n^2)$. When the number of data instances is large, it is prohibitive to store $\mathbf{K}$ in RAM, not to mention the time expensive eigenvalue decomposition. Therefore the sampling algorithm is limited to small-scale data.

To speedup the DPP sampling algorithm, Kulesza and Taskar (2010) proposed to use the dual representation of DPP (Dual-DPP) for sampling. If $\mathbf{K}$ has a rank-d decomposition $\mathbf{K} = \mathbf{D}\mathbf{D}^T$ where $\mathbf{D}$ is an $n \times d$ matrix, then the dual-DPP-Sample algorithm only takes $\mathcal{O}(nd + d^3)$ time and $\mathcal{O}(nd)$ space. However, such an exact low-rank decomposition is in general unavailable except for some special kernel matrices, e.g., the linear kernel. When the kernel matrix $\mathbf{K}$ is not low rank, it is still possible to obtain an approximate low-rank decomposition such that $\|\mathbf{K} - \mathbf{D}\mathbf{D}^T\|$ is minimized. For example, Affandi et al. (2013) employed the Nyström method (Nyström 1930) to generate a fast low-rank decomposition.

The Nyström method is an efficient and effective low-rank matrix approximation approach widely studied in the literature (Nyström 1930; Drineas and Mahoney 2005; Kumar, Mohri, and Talwalkar 2012; Gittens and Mahoney 2013; Wang and Zhang 2013; 2014). The Nyström method can approximate any symmetric positive semidefinite (SPSD) matrix by a portion of its columns, and a rank-d Nyström approximation can be obtained in time $\mathcal{O}(d^3)$. The Nyström method can significantly speedup kernel methods that perform eigenvalue decomposition, e.g., spectral clustering (Fowlkes et al. 2004; Li et al. 2011) and kernel PCA (Zhang, Tsang, and Kwok 2008; Zhang and Kwok 2010; Talwalkar et al. 2013), and kernel methods that perform matrix inverse, e.g., Gaussian process regression (Williams and Seeger 2001), kernel SVM (Zhang, Tsang, and Kwok 2008; Yang et al. 2012), and kernel ridge regression (Cortes, Mohri, and Talwalkar 2010).

It is well known that the Nyström method can preserve

*Corresponding author.

top eigenvalues of SPSD matrices. However, the Nyström method, as well as any other low-rank approximation methods, does not preserve the small eigenvalues; it simply discards the bottom singular values. Consequently, when applied to approximating matrix determinants, the Nyström approximation can be extremely far from the truth, which was demonstrated by the three examples of (Affandi et al. 2013). Therefore the Nyström method and other low-rank approximation methods are not good choices for speeding-up DPP, and it is of great interest to find some fast matrix approximation methods that preserve large and small eigenvalues alike.

We notice that a recently proposed matrix approximation method called the *Matrix Ridge Approximation (MRA)* (Zhang 2014) preserves eigenvalues both large and small. Unlike the low-rank matrix decomposition $\mathbf{K} \approx \mathbf{D}\mathbf{D}^T$ generated by the Nyström method, MRA approximates $\mathbf{K}$ by $\mathbf{K} \approx \hat{\mathbf{A}}\hat{\mathbf{A}}^T + \hat{\delta}\mathbf{I}_n$, where $\hat{\delta}$ is the arithmetic mean of the bottom eigenvalues of $\mathbf{K}$ and helps to preserve the bottom eigenvalues. This motivates us to apply MRA to speedup DPP, targeting at higher accuracy than the Nyström-DPP method. We call our method *MRA-DPP*. In this paper we provide theoretical analysis, examples, and experiments to demonstrate that our MRA-DPP is much more accurate than the Nyström-DPP of (Affandi et al. 2013).

The remainder of this paper is organized as follows. We first define the notation used in this paper and then introduce DPP, the Nyström method, the Nyström-DPP, and the matrix ridge approximation. Then we describe our proposed MRA-DPP-Sample algorithm in Algorithm 1 and provide theoretical analysis in Theorems 1 and 2 and Corollary 3, showing that our approach is better than the Nyström-DPP. Finally, we provide empirical comparisons on several real-world datasets to demonstrate the superiority of MRA-DPP over the other approximate methods.

Notation and Preliminary

Given an $m \times n$ matrix $\mathbf{A}$, we let $\mathbf{a}^{(i)}$ be its i-th row, $\mathbf{a}_j$ be its j-th column, and a_{ij} be its (i,j)-th entry. For index sets $\mathcal{I} \subset [m]$ and $\mathcal{J} \subset [n]$, we let $\mathbf{A}^{(\mathcal{I})}$ be the rows of $\mathbf{A}$ indexed by $\mathcal{I}$, $\mathbf{A}_{\mathcal{J}}$ be the columns of $\mathbf{A}$ indexed by $\mathcal{J}$, and $\mathbf{A}_{\mathcal{J}}^{(\mathcal{I})}$ be the corresponding submatrix of $\mathbf{A}$. We let $\mathbf{I}_n$ be the $n \times n$ identity matrix.

The eigenvalue decomposition of an $n \times n$ real matrix $\mathbf{K}$ is defined by

$$\begin{aligned}\mathbf{K} &= \mathbf{U}_{\mathbf{K}}\mathbf{\Lambda}_{\mathbf{K}}\mathbf{U}_{\mathbf{K}}^T \\ &= [\mathbf{U}_{\mathbf{K},k}, \bar{\mathbf{U}}_{\mathbf{K},k}] \begin{bmatrix} \mathbf{\Lambda}_{\mathbf{K},k} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{\Lambda}}_{\mathbf{K},k} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{\mathbf{K},k}^T \\ \bar{\mathbf{U}}_{\mathbf{K},k}^T \end{bmatrix},\end{aligned}$$

where $\mathbf{U}_{\mathbf{K},k}$ $(n \times k)$ and $\mathbf{\Lambda}_{\mathbf{K},k}$ $(k \times k)$ correspond to the top k eigenvalues. We denote the i-th largest eigenvalue by $\lambda_i(\mathbf{K})$, which is the i-th diagonal entry of $\mathbf{\Lambda}_{\mathbf{K}}$. If $\mathbf{K}$ is SPSD, then $\lambda_i(\mathbf{K})$ equals to its i-th largest singular value $\sigma_i(\mathbf{K})$.

The singular value decomposition (SVD) of an $m \times n$ matrix costs time $\mathcal{O}(\min\{m^2n, mn^2\})$, and the eigenvalue decomposition of an $n \times n$ matrix costs time $\mathcal{O}(n^3)$. Although multiplying an $m \times n$ matrix by an $n \times p$ matrix runs in $\mathcal{O}(mnp)$ flops, the constant in the big-$\mathcal{O}$ notation is tremendously smaller than that of SVD and eigenvalue decomposition. Moreover, matrix multiplication can be implemented in a parallel fashion. So we instead denote the time complexity of matrix multiplication by $T_{\text{multiply}}(mnp)$, which is far less than $\mathcal{O}(mnp)$ in practice (Halko, Martinsson, and Tropp 2011).

Background

In this section we formally introduce DPPs, the Nyström method and Nyström-DPP, and MRA.

Determinantal Point Processes

Formally speaking, for an $n \times n$ SPSD kernel matrix $\mathbf{K}$, the probability measure of the standard DDP is defined by

$$P_{\mathbf{K}}(\mathcal{S}) = \frac{\det(\mathbf{K}_{\mathcal{S}}^{(\mathcal{S})})}{\det(\mathbf{K}+\mathbf{I}_n)} \quad \text{for all } \mathcal{S} \subset [n]. \tag{1}$$

The DPP satisfies $\sum_{\mathcal{S}\subset[n]} P_{\mathbf{K}}(\mathcal{S}) = 1$. Hough et al. (2006) provided an exact sampling algorithm for DPP which requires the full eigenvalue decomposition of $\mathbf{K}$, so learning DPP can be done in time $\mathcal{O}(n^3)$.

In applications where the cardinality of the sampled subsets is fixed, say k, we are interested in probability measures that only assign positive probability to subsets of cardinality k. This variant is called the kDPP (Kulesza and Taskar 2011a). The kDPP $P_{\mathbf{K}}^k$ is defined by

$$P_{\mathbf{K}}^k(\mathcal{S}) = \frac{\det(\mathbf{K}_{\mathcal{S}}^{(\mathcal{S})})}{\sum_{|\mathcal{I}|=k}\det(\mathbf{K}_{\mathcal{I}}^{(\mathcal{I})})}$$

for all sets $\mathcal{S} \subset [n]$ with cardinality k. The sampling algorithm of kDPP is similar to that of the standard DPP.

Let $\tilde{\mathbf{K}}$ be an arbitrary approximation of $\mathbf{K}$, Affandi et al. (2013) used the ℓ_1 variational distance between the DPP with kernel $\mathbf{K}$ and the DPP with kernel $\tilde{\mathbf{K}}$ as an approximation quality criterion. The ℓ_1 variational distance is defined by

$$\left\|P_{\mathbf{K}} - P_{\tilde{\mathbf{K}}}\right\|_1 \triangleq \frac{1}{2}\sum_{\mathcal{S}\subset[n]} \left|P_{\mathbf{K}}(\mathcal{S}) - P_{\tilde{\mathbf{K}}}(\mathcal{S})\right|. \tag{2}$$

The ℓ_1 variational distance of kDPP is similarly defined.

The Nyström Method and Nyström-DPP

Given an $n \times n$ SPSD kernel matrix $\mathbf{K}$, the Nyström method approximates $\mathbf{K}$ by a subset of its columns. There are various ways to choose columns of $\mathbf{K}$, e.g., uniform sampling, adaptive sampling (Boutsidis, Drineas, and Magdon-Ismail 2011; Deshpande et al. 2006), statistical leverage based sampling (Drineas, Mahoney, and Muthukrishnan 2008), volume sampling (Guruswami and Sinop 2012), etc. Let the selected columns be indexed by a set $\mathcal{J}$ ($\mathcal{J} \subset [n]$ and $|\mathcal{J}| = d \ll n$). The Nyström approximation is given by

$$\bar{\mathbf{K}} = \underbrace{\mathbf{K}_{\mathcal{J}}}_{n\times d}\underbrace{\left(\mathbf{K}_{\mathcal{J}}^{(\mathcal{J})}\right)^{\dagger}}_{d\times d}\underbrace{\left(\mathbf{K}_{\mathcal{J}}\right)^T}_{d\times n} = \mathbf{D}\mathbf{D}^T,$$

where $\mathbf{D} = \mathbf{K}_{\mathcal{J}}\big(\big(\mathbf{K}_{\mathcal{J}}^{(\mathcal{J})}\big)^{\dagger}\big)^{1/2} \in \mathbb{R}^{n\times d}$.

The decomposition above can be used to speedup eigenvalue decomposition as follows. We let $\mathbf{C} = \mathbf{D}^T\mathbf{D} \in \mathbb{R}^{d\times d}$ and compute the eigenvalue decomposition $\mathbf{C} = \mathbf{U}_{\mathbf{C}}\mathbf{\Lambda}_{\mathbf{C}}\mathbf{U}_{\mathbf{C}}^T$, then the eigenvalue decomposition of $\bar{\mathbf{K}}$ is

$$\bar{\mathbf{K}} \;=\; \big(\mathbf{D}\mathbf{U}_{\mathbf{C}}\mathbf{\Lambda}_{\mathbf{C}}^{-1/2}\big)\mathbf{\Lambda}_{\mathbf{C}}\big(\mathbf{D}\mathbf{U}_{\mathbf{C}}\mathbf{\Lambda}_{\mathbf{C}}^{-1/2}\big)^T.$$

Therefore it is feasible to make the sampling algorithm of DPP (DPP-Sample) more efficient by employing the Nyström method to speedup eigenvalue decomposition. Affandi et al. (2013) showed that Nyström-DPP is still efficient even for large-scale matrix where the exact eigenvalue decomposition is prohibitive.

Previous work (Gittens and Mahoney 2013) has shown that if sufficiently many columns are selected to construct the Nyström approximation, the incurred error $\|\mathbf{K} - \bar{\mathbf{K}}\|_F$ is small. However, small $\|\mathbf{K} - \bar{\mathbf{K}}\|_F$ does not implies small ℓ_1 variational distance $\|P_{\mathbf{K}} - P_{\bar{\mathbf{K}}}\|_1$. Affandi et al. (2013) presented three examples, showing that $\|P_{\mathbf{K}} - P_{\bar{\mathbf{K}}}\|_1$ may tend to some nonzero constants even when $\|\mathbf{K} - \bar{\mathbf{K}}\|_F \to 0$. It is because the Nyström method does not preserve the bottom eigenvalues, and small eigenvalues has big influence on the matrix determinant. Therefore, it is useful to find an approximation that preserves large and small eigenvalues alike.

The Matrix Ridge Approximation (MRA)

In one latest work, Zhang (2014) proposed a called matrix ridge approximation (MRA), which is able to preserve large and small eigenvalues of any SPSD matrix and is provably a tighter approximation than the truncated SVD.

Definition 1 (Zhang 2014). *The Matrix Ridge Approximation (MRA) of* $\mathbf{K}$ *is defined by*

$$\hat{\mathbf{K}} \;=\; \hat{\mathbf{A}}\hat{\mathbf{A}}^T + \hat{\delta}\mathbf{I}_n,$$

where

$$\begin{aligned}\hat{\mathbf{A}} \;&=\; \mathbf{U}_{\mathbf{K},d}\big(\mathbf{\Lambda}_{\mathbf{K},d} - \hat{\delta}\mathbf{I}_d\big)^{\frac{1}{2}}\mathbf{V},\\ \hat{\delta} \;&=\; \frac{1}{n-d}\sum_{i=d+1}^{n}\lambda_i(\mathbf{K}),\end{aligned}$$

and $\mathbf{V}$ *is an arbitrary* $d \times d$ *orthogonal matrix.*

MRA has a closed-form solution by performing the rank-d truncated eigenvalue decomposition. To avoid eigenvalue decomposition, Zhang (2014) provided an EM algorithm for computing $\hat{\mathbf{A}}$ and $\hat{\delta}$. The EM algorithm alternates the following steps until convergence:

$$\begin{aligned}\mathbf{A}_{(t+1)} \;&=\; \mathbf{K}\mathbf{A}_{(t)}\Big(\delta_{(t)}\mathbf{I}_d + \mathbf{\Sigma}_{(t)}^{-1}\mathbf{A}_{(t)}^T\mathbf{K}\mathbf{A}_{(t)}\Big)^{-1},\\ \delta_{(t+1)} \;&=\; \frac{1}{n}\Big(\mathrm{tr}(\mathbf{K}) - \mathrm{tr}\big(\mathbf{A}_{(t+1)}^T\mathbf{\Sigma}_{(t)}^{-1}\mathbf{A}_{(t)}^T\mathbf{K}\big)\Big),\\ \mathbf{\Sigma}_{(t+1)} \;&=\; \delta_{(t+1)}\mathbf{I}_d + \mathbf{A}_{(t+1)}^T\mathbf{A}_{(t+1)}.\end{aligned}$$

The algorithm costs time $\mathcal{O}(Td^3) + T_{\text{multiply}}(Tn^2d)$ where T is the maximum iterative number, and T is usually much smaller than n.

Algorithm 1 MRA-DPP-Sample.

Input: kernel matrix $\mathbf{K}$.
$\{\hat{\mathbf{A}}, \hat{\delta}\} \longleftarrow$ MRA of $\mathbf{K}$;
$\{(\hat{\lambda}_i, \hat{\mathbf{u}}_i)\}_{i=1}^d \longleftarrow$ eigenvalue decomposition of $\mathbf{A}^T\mathbf{A}$;
$\lambda_i \longleftarrow \hat{\lambda}_i + \hat{\delta}$ for $i = 1$ to d; $\quad \lambda_i \longleftarrow \hat{\delta}$ for $i = d$ to n;
$\mathbf{u}_i \longleftarrow \hat{\lambda}_i^{-1/2}\hat{\mathbf{A}}\hat{\mathbf{u}}_i$ for $i = 1$ to d;
$\mathcal{S} \longleftarrow \emptyset$;
$\mathcal{S} \longleftarrow \mathcal{S} \cup \{i\}$ with probability $\frac{\lambda_i}{\lambda_i+1}$ for $i = 1$ to n;
$\mathcal{V} \longleftarrow \{\mathbf{u}_i\}_{i\in\mathcal{S}\cap[d]}$;
$r \longleftarrow$ cardinality of the set $\mathcal{S} \cap \{d+1, \cdots, n\}$;
$\mathcal{W} \longleftarrow r$ arbitrary orthonormal bases that are orthogonal to $\mathbf{u}_1, \cdots, \mathbf{u}_d$; $\quad \mathcal{V} \longleftarrow \mathcal{V} \cup \mathcal{W}$;
$\mathcal{Y} \longleftarrow \emptyset$;
while $|\mathcal{V}| > 0$ **do**
  select $i \in [n]$ with probability $\frac{1}{|\mathcal{V}|}\sum_{\mathbf{u}\in\mathcal{V}}\big(\mathbf{u}^T\mathbf{e}_i\big)^2$;
  $\mathcal{Y} \longleftarrow \mathcal{Y} \cup i$;
  $\mathcal{V} \longleftarrow$ the orthonormal bases of the subspace of $\mathcal{V}$ orthogonal to $\mathbf{e}_i$;
end while
return $\mathcal{Y}$.

In fact, MRA can be solved much more efficiently by randomized SVD (Halko, Martinsson, and Tropp 2011). By using randomized SVD to compute $\hat{\mathbf{A}}$ and $\hat{\delta}$, MRA can be solved within arbitrary approximation accuracy in time $\mathcal{O}(nd^2) + T_{\text{multiply}}(n^2d)$ and space $\mathcal{O}(nd)$. Therefore, MRA by randomized SVD still works efficiently when n is large, and MRA can be potentially applied to big data problems. Though currently MRA by randomized SVD has no theoretical guarantee when applied to speedup DPP, we still recommend the readers to compute MRA by randomized SVD, which is faster than the EM algorithm.

Methodology

As is discussed in the previous section, a high quality approximation of DPP should preserve both of the large and the small eigenvalues of the kernel matrix. It is thus very intuitive to use the MRA method to approximate the kernel matrix of DPP. Following the DPP-Sample algorithm of Hough et al. (2006), we derive a sampling algorithm for MRA-DPP and describe it in Algorithm 1. We also provide theoretical analysis for MRA-DPP, showing that MRA-DPP is more accurate than the Nyström-DPP of (Affandi et al. 2013). The error bound of kDPP can be obtained in a similar way as the work of (Affandi et al. 2013), so we leave it out in this paper.

The MRA-DPP-Sample algorithm (Algorithm 1) is derived as follows. Let $\hat{\mathbf{K}} = \hat{\mathbf{A}}\hat{\mathbf{A}}^T + \hat{\delta}\mathbf{I}_n$ be the MRA of the kernel matrix $\mathbf{K}$, and let $\hat{\lambda}_i$ and $\hat{\mathbf{u}}_i$ be the i-th top eigenvalue and eigenvector of the $d \times d$ matrix $\hat{\mathbf{A}}^T\hat{\mathbf{A}}$. It is easily verified that $\lambda_i = \hat{\lambda}_i + \hat{\delta}$ and $\mathbf{u}_i = \hat{\lambda}_i^{-1/2}\hat{\mathbf{A}}\hat{\mathbf{u}}_i$ are the i-th ($i \in [d]$) top eigenvalue and eigenvector of $\hat{\mathbf{K}}$. The $d+1$ to n eigenvalues of $\hat{\mathbf{K}}$ are all δ, and the eigenvectors are arbitrary orthonormal bases of the subspace orthogonal to $\{\mathbf{u}_i\}_{i=1}^d$. With the full eigenvalue decomposition of $\hat{\mathbf{K}}$ at hand, the MRA-DPP-Sample algorithm is immediately obtained fol-

lowing the DPP-Sample algorithm of (Hough et al. 2006).

We analyze the approximation quality of MRA-DPP in Theorem 1, and then in the remainder of this section we discuss the error bound given in the theorem.

Theorem 1. *Given an $n \times n$ SPSD matrix $\mathbf{K}$, we let $\hat{\mathbf{K}} = \hat{\mathbf{A}}\hat{\mathbf{A}}^T + \hat{\delta}\mathbf{I}_n$ be an size-d MRA of $\mathbf{K}$ defined in Definition 1. For an index set $\mathcal{S} \subset [n]$, the relative set-wise error satisfies that*

$$\left|P_{\mathbf{K}}(\mathcal{S}) - P_{\hat{\mathbf{K}}}(\mathcal{S})\right| \,/\, P_{\mathbf{K}}(\mathcal{S}) \leq \left|1 - \frac{\prod_{i=d+1}^{n}\left(1+\lambda_i(\mathbf{K})\right)}{(1+\hat{\delta})^{n-d}} \left[\prod_{i=1}^{|\mathcal{S}|} \frac{\lambda_i\big(\hat{\mathbf{K}}_{\mathcal{S}}^{(\mathcal{S})}\big)}{\lambda_i\big(\mathbf{K}_{\mathcal{S}}^{(\mathcal{S})}\big)}\right]\right|.$$

We note that $1+\hat{\delta}$ is the arithmetic mean of $1 + \lambda_{d+1}(\mathbf{K}), \cdots, 1+\lambda_n(\mathbf{K})$, while $\big(\prod_{i=d+1}^{n}(1+\lambda_i(\mathbf{K}))\big)^{\frac{1}{n-d}}$ is the geometric mean. When the bottom $n-d$ eigenvalues $\lambda_{d+1}(\mathbf{K}), \cdots, \lambda_n(\mathbf{K})$ are small, $1+\lambda_{d+1}(\mathbf{K}), \cdots, 1+\lambda_n(\mathbf{K})$ have small variance, and thus the arithmetic mean and the geometric mean approach each other (Aldaz 2012). Therefore, when the d to n eigenvalues of $\mathbf{K}$ are small, the term $\frac{\prod_{i=d+1}^{n}(1+\lambda_i(\mathbf{K}))}{(1+\hat{\delta})^{n-d}}$ approaches 1.

We bound the difference between $\lambda_i\big(\hat{\mathbf{K}}_{\mathcal{S}}^{(\mathcal{S})}\big)$ and $\lambda_i\big(\mathbf{K}_{\mathcal{S}}^{(\mathcal{S})}\big)$ in the following theorem. We will discuss in Remark 1 that our theoretical result is stronger than that of (Affandi et al. 2013).

Theorem 2. *Given an $n \times n$ SPSD matrix $\mathbf{K}$, let $\hat{\mathbf{K}}$ be a size-d MRA of $\mathbf{K}$. Then for any index set $\mathcal{S} \subset [n]$, the eigenvalues of any submatrices of $\mathbf{K}$ and $\hat{\mathbf{K}}$ indexed by $\mathcal{S}$ satisfy that*

$$\left|\lambda_i\big(\mathbf{K}_{\mathcal{S}}^{(\mathcal{S})}\big) - \lambda_i\big(\hat{\mathbf{K}}_{\mathcal{S}}^{(\mathcal{S})}\big)\right| \leq \max\left\{\lambda_{d+1}(\mathbf{K}) - \hat{\delta},\ \hat{\delta} - \lambda_n(\mathbf{K})\right\},$$

where $\hat{\delta}$ is defined in Definition 1.

Remark 1. *The result in Theorem 2 is stronger than the corresponding results of (Affandi et al. 2013) because*

$$\begin{aligned}\max\left\{\lambda_{d+1}(\mathbf{K}) - \hat{\delta},\ \hat{\delta} - \lambda_n(\mathbf{K})\right\} &\leq \lambda_{d+1}(\mathbf{K}) \\ &\leq \|\mathbf{K} - \bar{\mathbf{K}}\|_2,\end{aligned}$$

where $\bar{\mathbf{K}}$ is the size-d Nyström approximation of $\mathbf{K}$. The first inequality holds if and only if $\lambda_{d+1}(\mathbf{K}) = \cdots = \lambda_n(\mathbf{K}) = 0$. Thus in general settings the error bound of our method is strictly better than that of (Affandi et al. 2013). The inequality means that using MRA to approximate DPP is more accurate than using the truncated SVD and the Nyström method.

As a special case, when the bottom $n-d$ eigenvalues of $\mathbf{K}$ have zero variance, the error incurred by MRA-DPP is zero. Under the same condition, the error incurred by the truncated SVD or the Nyström approximation is in general nonzero (see Examples 1 and 2 of Affandi et al. 2013).

Corollary 3. *Let $\hat{\mathbf{K}}$ be the size-d MRA of $\mathbf{K}$. When $\lambda_{d+1}(\mathbf{K}) = \cdots = \lambda_n(\mathbf{K})$, the ℓ_1 variational distance between the DPP with kernel $\mathbf{K}$ and the DPP with kernel $\hat{\mathbf{K}}$ is zero. That is, $\left\|P_{\mathbf{K}} - P_{\hat{\mathbf{K}}}\right\|_1 = 0$.*

Experiments

We conduct experiments on several real-world datasets from UCI (Frank and Asuncion 2010) and Statlog (Michie, Spiegelhalter, and Taylor 1994) to evaluate the kernel approximation methods for DPP. For each dataset, we generate a radial basis function (RBF) kernel $\mathbf{K}$ defined by $k_{ij} = \exp(-\frac{1}{2\alpha}\|\mathbf{x}_i - \mathbf{x}_j\|_2^2)$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are data instances, and α is the scaling parameter defining the scale of the kernel matrix. We set $\alpha = 0.2$, 0.5, and 1 in our experiments. All of the compared methods are implemented in MATLAB and run on a PC with Intel Core i5 CPU, 8GB RAM, and Windows 7 64bit system.

To make the experiments simple, we compare the performance of the approximation methods for kDPP. For a dataset of n instances, there are $\binom{n}{k}$ subsets of size k, so it is prohibitive to directly compute the ℓ_1 variational distance

$$\left\|P_{\mathbf{K}}^k - P_{\tilde{\mathbf{K}}}^k\right\|_1 \triangleq \frac{1}{2}\sum_{\mathcal{S}\subset[n],|\mathcal{S}|=k}\left|P_{\mathbf{K}}^k(\mathcal{S}) - P_{\tilde{\mathbf{K}}}^k(\mathcal{S})\right|,$$

where $\tilde{\mathbf{K}}$ is an arbitrary approximation of $\mathbf{K}$. We instead evaluate the approximation accuracy by the *empirical ℓ_1 variational distance* which is defined as follows. We generate m sets $\mathcal{S}_1, \cdots, \mathcal{S}_m \subset [n]$ uniformly at random, each of cardinality k, and define

$$\tilde{P}_{\mathbf{K}}^k(\mathcal{S}_i) = \frac{\det(\mathbf{K}_{\mathcal{S}_i}^{(\mathcal{S}_i)})}{\sum_{j=1}^{m}\det(\mathbf{K}_{\mathcal{S}_j}^{(\mathcal{S}_j)})};$$

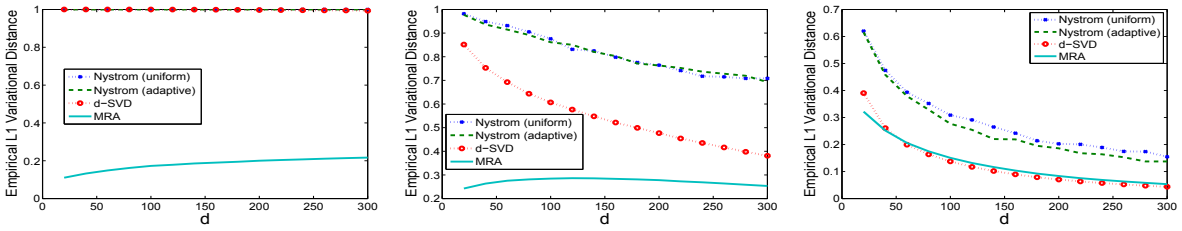
the empirical ℓ_1 variational distance is defined by

$$\left\|\tilde{P}_{\mathbf{K}}^k - \tilde{P}_{\tilde{\mathbf{K}}}^k\right\|_1 \triangleq \frac{1}{2}\sum_{i=1}^{m}\left|\tilde{P}_{\mathbf{K}}^k(\mathcal{S}_i) - \tilde{P}_{\tilde{\mathbf{K}}}^k(\mathcal{S}_i)\right|.$$

We set $k = 10$ and $m = 10^6$ in our experiments.

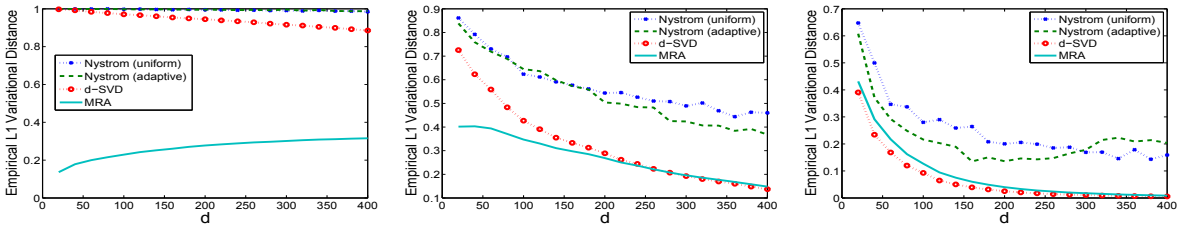
We mainly compare our MRA-DPP with the Nyström-DPP of (Affandi et al. 2013). The MRA is computed by the EM algorithm of (Zhang 2014). The Nyström approximation is constructed by uniform sampling or the adaptive sampling algorithm of (Wang and Zhang 2013). Since the uniform sampling and the adaptive sampling algorithms are both randomized, we run each algorithm ten times and use the sampled columns that achieve the minimal Frobenius norm error $\|\mathbf{K} - \bar{\mathbf{K}}\|_F$ where $\bar{\mathbf{K}}$ is the Nyström approximation. We also employ the truncated SVD for comparison, although it is impractical in real-world applications. We let d be the size of MRA or the rank of the Nyström approximation or the truncated SVD; we range d and report the empirical ℓ_1 distance in Figures 1–6.

We can see from the experiment results that our MRA-DPP is much more accurate than the Nyström-DPP in all experiments. Especially, when the scaling parameter α is set as a small value, say $\alpha = 0.2$, the ℓ_1 variational distances corresponding to the Nyström method and the truncated SVD are near 1; that is, the two low-rank approximation methods are ineffective approximations of DPP. This is because when the scaling parameter α takes a small value, the bottom eigenvalues are not sufficiently small, and discarding these



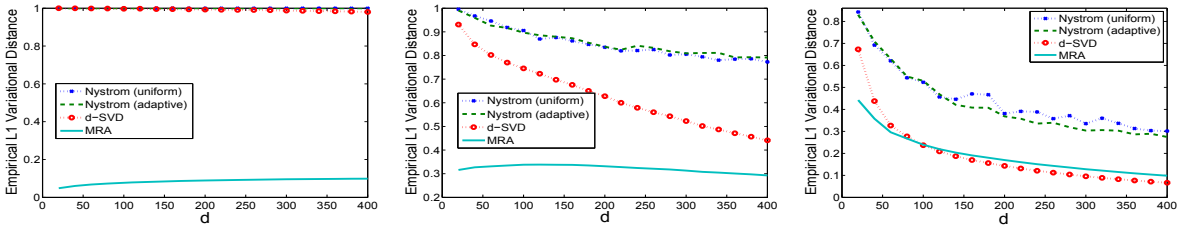
(a) RBF kernel with $\sigma = 0.2$. (b) RBF kernel with $\sigma = 0.5$. (c) RBF kernel with $\sigma = 1$.

Figure 1: Results on the Letters dataset ($5,000$ instances, 16 attributes, Statlog).



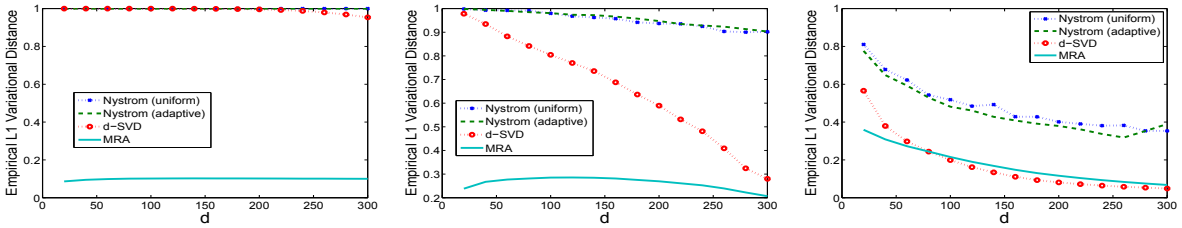
(a) RBF kernel with $\sigma = 0.2$. (b) RBF kernel with $\sigma = 0.5$. (c) RBF kernel with $\sigma = 1$.

Figure 2: Results on the Wine Quality dataset ($4,898$ instances, 12 attributes, UCI).



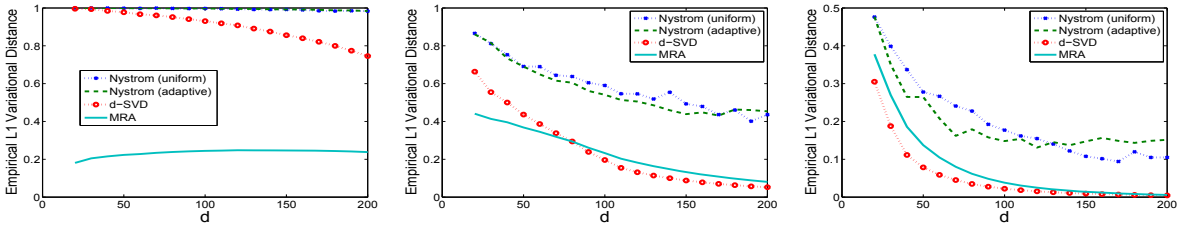
(a) RBF kernel with $\sigma = 0.2$. (b) RBF kernel with $\sigma = 0.5$. (c) RBF kernel with $\sigma = 1$.

Figure 3: Results on the Satimage dataset ($4,435$ instances, 36 attributes, Statlog).



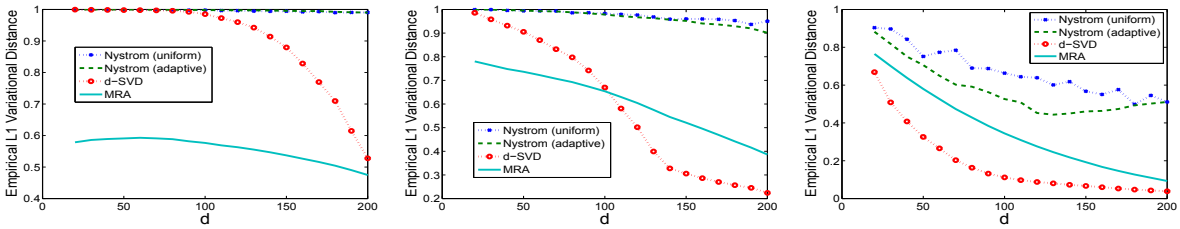
(a) RBF kernel with $\sigma = 0.2$. (b) RBF kernel with $\sigma = 0.5$. (c) RBF kernel with $\sigma = 1$.

Figure 4: Results on the German dataset ($1,000$ instances, 24 attributes, Statlog).



(a) RBF kernel with $\sigma = 0.2$. (b) RBF kernel with $\sigma = 0.5$. (c) RBF kernel with $\sigma = 1$.

Figure 5: Results on the Diabetes dataset (768 instances, 8 attributes, UCI).



(a) RBF kernel with $\sigma = 0.2$. (b) RBF kernel with $\sigma = 0.5$. (c) RBF kernel with $\sigma = 1$.

Figure 6: Results on the Breast-Cancer dataset (683 instances, 10 attributes, UCI).

bottom eigenvalues has big influence on the matrix determinants. Since the low-rank approximation methods, i.e., the Nyström method and the truncated SVD, discard the bottom eigenvalues, they have very low performance on the RBF kernel with small scaling parameter α. In contrast, the error of our MRA-DPP is small in all cases because MRA preserves eigenvalues both big and small.

In some sets of experiments, the performance of MRA-DPP is not monotonically getting better in d. This counter-intuitive phenomenon may be explained as follows. As d grows, the whole MRA matrix is getting closer to the original matrix, but many submatrices of the MRA may be getting farther from the ground truth. Consequently, the ℓ_1 variational distance may become worse.

Proof of Theorems

Proof of Theorem 1

Proof. The relative set-wise error is

$$\frac{\left|P_{\mathbf{K}}(\mathcal{S}) - P_{\hat{\mathbf{K}}}(\mathcal{S})\right|}{P_{\mathbf{K}}(\mathcal{S})} = \left|1 - \frac{\det\left(\hat{\mathbf{K}}_{\mathcal{S}}^{(\mathcal{S})}\right)\det\left(\mathbf{K}+\mathbf{I}_n\right)}{\det\left(\hat{\mathbf{K}}+\mathbf{I}_n\right)\det\left(\mathbf{K}_{\mathcal{S}}^{(\mathcal{S})}\right)}\right|$$

$$= \left|1 - \left[\prod_{i=1}^{n}\frac{1+\lambda_i(\mathbf{K})}{1+\lambda_i(\hat{\mathbf{K}})}\right]\left[\prod_{i=1}^{|\mathcal{S}|}\frac{\lambda_i\left(\hat{\mathbf{K}}_{\mathcal{S}}^{(\mathcal{S})}\right)}{\lambda_i\left(\mathbf{K}_{\mathcal{S}}^{(\mathcal{S})}\right)}\right]\right|$$

$$= \left|1 - \frac{\prod_{i=d+1}^{n}\left(1+\lambda_i(\mathbf{K})\right)}{(1+\hat{\delta})^{n-d}}\left[\prod_{i=1}^{|\mathcal{S}|}\frac{\lambda_i\left(\hat{\mathbf{K}}_{\mathcal{S}}^{(\mathcal{S})}\right)}{\lambda_i\left(\mathbf{K}_{\mathcal{S}}^{(\mathcal{S})}\right)}\right]\right|$$

where the last equality follows by Definition 1. □

Proof of Theorem 2

Proof. The eigenvalue decomposition of the SPSD kernel matrix $\mathbf{K}$ is

$$\begin{aligned}\mathbf{K} &= \mathbf{U}_{\mathbf{K}}(\mathbf{\Lambda}_{\mathbf{K}} - \hat{\delta}\mathbf{I}_n)\mathbf{U}_{\mathbf{K}}^T + \hat{\delta}\mathbf{I}_n \\ &= \mathbf{U}_{\mathbf{K},d}(\mathbf{\Lambda}_{\mathbf{K},d} - \hat{\delta}\mathbf{I}_d)\mathbf{U}_{\mathbf{K},d}^T + \hat{\delta}\mathbf{I}_n \\ &\quad + \bar{\mathbf{U}}_{\mathbf{K},d}(\bar{\mathbf{\Lambda}}_{\mathbf{K},d} - \hat{\delta}\mathbf{I}_{n-d})\bar{\mathbf{U}}_{\mathbf{K},d}^T\end{aligned}$$

and thus the submatrix of $\mathbf{K}$ indexed by $\mathcal{S}$ ($|\mathcal{S}| = k$) is

$$\begin{aligned}\mathbf{K}_{\mathcal{S}}^{(\mathcal{S})} = \mathbf{U}_{\mathbf{K},d}^{(\mathcal{S})}\left(\mathbf{\Lambda}_{\mathbf{K},d} - \hat{\delta}\mathbf{I}_d\right)\left(\mathbf{U}_{\mathbf{K},d}^{(\mathcal{S})}\right)^T + \hat{\delta}\mathbf{I}_k \\ + \bar{\mathbf{U}}_{\mathbf{K},d}^{(\mathcal{S})}\left(\bar{\mathbf{\Lambda}}_{\mathbf{K},d} - \hat{\delta}\mathbf{I}_{n-d}\right)\left(\bar{\mathbf{U}}_{\mathbf{K},d}^{(\mathcal{S})}\right)^T.\end{aligned}$$

The submatrix of $\hat{\mathbf{K}}$ (MRA of $\mathbf{K}$) indexed by $\mathcal{S}$ is

$$\hat{\mathbf{K}}_{\mathcal{S}}^{(\mathcal{S})} = \mathbf{U}_{\mathbf{K},d}^{(\mathcal{S})}(\mathbf{\Lambda}_{\mathbf{K},c} - \hat{\delta}\mathbf{I}_{n-d})(\mathbf{U}_{\mathbf{K},d}^{(\mathcal{S})})^T + \hat{\delta}\mathbf{I}_k.$$

We can see that the difference between $\mathbf{K}_{\mathcal{S}}^{(\mathcal{S})}$ and $\hat{\mathbf{K}}_{\mathcal{S}}^{(\mathcal{S})}$ is

$$\mathbf{K}_{\mathcal{S}}^{(\mathcal{S})} - \hat{\mathbf{K}}_{\mathcal{S}}^{(\mathcal{S})} = \bar{\mathbf{U}}_{\mathbf{K},d}^{(\mathcal{S})}\left(\bar{\mathbf{\Lambda}}_{\mathbf{K},d} - \hat{\delta}\mathbf{I}_{n-d}\right)\left(\bar{\mathbf{U}}_{\mathbf{K},d}^{(\mathcal{S})}\right)^T.$$

Finally we have that

$$\begin{aligned}\left|\lambda_i\left(\mathbf{K}_{\mathcal{S}}^{(\mathcal{S})}\right) - \lambda_i\left(\hat{\mathbf{K}}_{\mathcal{S}}^{(\mathcal{S})}\right)\right| &= \left|\sigma_i\left(\mathbf{K}_{\mathcal{S}}^{(\mathcal{S})}\right) - \sigma_i\left(\hat{\mathbf{K}}_{\mathcal{S}}^{(\mathcal{S})}\right)\right| \\ &\le \sigma_1\left(\mathbf{K}_{\mathcal{S}}^{(\mathcal{S})} - \hat{\mathbf{K}}_{\mathcal{S}}^{(\mathcal{S})}\right) \\ &= \sigma_1\left(\bar{\mathbf{U}}_{\mathbf{K},d}^{(\mathcal{S})}\left(\bar{\mathbf{\Lambda}}_{\mathbf{K},d} - \hat{\delta}\mathbf{I}_{n-d}\right)\left(\bar{\mathbf{U}}_{\mathbf{K},d}^{(\mathcal{S})}\right)^T\right) \\ &\le \sigma_1\left(\bar{\mathbf{U}}_{\mathbf{K},d}\left(\bar{\mathbf{\Lambda}}_{\mathbf{K},d} - \hat{\delta}\mathbf{I}_{n-d}\right)\bar{\mathbf{U}}_{\mathbf{K},d}^T\right) \\ &= \lambda_1^{\frac{1}{2}}\left(\left(\bar{\mathbf{\Lambda}}_{\mathbf{K},d} - \hat{\delta}\mathbf{I}_{n-d}\right)^2\right) \\ &= \max\left\{\lambda_{d+1}(\mathbf{K}) - \hat{\delta},\ \hat{\delta} - \lambda_n(\mathbf{K})\right\}.\end{aligned}$$

Here the first equality follows from that $\mathbf{K}_{\mathcal{S}}^{(\mathcal{S})}$ and $\tilde{\mathbf{K}}_{\mathcal{S}}^{(\mathcal{S})}$ are SPSD; the first inequality follows from the singular value inequality in (Horn and Johnson 1991, Theorem 3.3.16); the second inequality follows from the singular value inequality in (Horn and Johnson 1991, Theorem 3.3.1); the last equality follows from that $\lambda_{d+1}(\mathbf{K}) \ge \hat{\delta} \ge \lambda_n(\mathbf{K})$. □

Conclusions

In this paper we have proposed to apply the matrix ridge approximation (MRA) to speedup the determinantal point processes (DPPs) and provided theoretical analysis for the approximation performance. Our proposed MRA-DPP is superior over the Nyström-DPP both theoretically and empirically. We have shown theoretically that the error bound of MRA-DPP is stronger than Nyström-DPP. The experiments on several real-world datasets have shown that MRA is much more accurate than Nyström when applied to approximate DPP. Especially, when the spectrum of the kernel matrix decays slowly, MRA achieves much higher accuracy than the Nyström method and even the truncated SVD.

Acknowledgement

Shusen Wang is supported by Microsoft Research Asia Fellowship 2013 and the Scholarship Award for Excellent Doctoral Student granted by Chinese Ministry of Education. Hui Qian is supported by the National Natural Science Foundation of China (No. 61272303) and the National Program on Key Basic Research Project of China (973 Program, No. 2010CB327903). Zhihua Zhang is supported by the National Natural Science Foundation of China (No. 61070239).

References

Affandi, R. H.; Kulesza, A.; Fox, E. B.; and Taskar, B. 2013. Nyström approximation for large-scale determinantal processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Affandi, R. H.; Fox, E.; and Taskar, B. 2013. Approximate inference in continuous determinantal processes. In *Advances in Neural Information Processing Systems (NIPS)*.

Aldaz, J. M. 2012. Sharp bounds for the difference between the arithmetic and geometric means. *Archiv der Mathematik* 99(4):393–399.

Boutsidis, C.; Drineas, P.; and Magdon-Ismail, M. 2011. Near optimal column-based matrix reconstruction. In *Annual Symposium on Foundations of Computer Science (FOCS)*.

Cortes, C.; Mohri, M.; and Talwalkar, A. 2010. On the impact of kernel approximation on learning accuracy. In *Conference on Artificial Intelligence and Statistics (AISTATS)*.

Deshpande, A.; Rademacher, L.; Vempala, S.; and Wang, G. 2006. Matrix approximation and projective clustering via volume sampling. *Theory of Computing* 2(2006):225–247.

Drineas, P., and Mahoney, M. W. 2005. On the Nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research* 6:2153–2175.

Drineas, P.; Mahoney, M. W.; and Muthukrishnan, S. 2008. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications* 30(2):844–881.

Fowlkes, C.; Belongie, S.; Chung, F.; and Malik, J. 2004. Spectral grouping using the Nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(2):214–225.

Frank, A., and Asuncion, A. 2010. UCI machine learning repository.

Gittens, A., and Mahoney, M. W. 2013. Revisiting the nyström method for improved large-scale machine learning. In *International Conference on Machine Learning (ICML)*.

Guruswami, V., and Sinop, A. K. 2012. Optimal column-based low-rank matrix reconstruction. In *the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*.

Halko, N.; Martinsson, P.-G.; and Tropp, J. A. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review* 53(2):217–288.

Horn, R. A., and Johnson, C. R. 1991. Topics in matrix analysis. *Cambridge University Presss, Cambridge*.

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

Kulesza, A., and Taskar, B. 2010. Structured determinantal point processes. In *Advances in neural information processing systems (NIPS)*.

Kulesza, A., and Taskar, B. 2011a. k-dpps: Fixed-size determinantal point processes. In *International Conference on Machine Learning (ICML)*.

Kulesza, A., and Taskar, B. 2011b. Learning determinantal point processes. In *Proceedings of Conference on Uncertainty (UAI)*.

Kulesza, A., and Taskar, B. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning* 5(2-3):123–286.

Kumar, S.; Mohri, M.; and Talwalkar, A. 2012. Sampling methods for the Nyström method. *Journal of Machine Learning Research* 13:981–1006.

Li, M.; Lian, X.-C.; Kwok, J. T.; and Lu, B.-L. 2011. Time and space efficient spectral clustering via column sampling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Michie, D.; Spiegelhalter, D. J.; and Taylor, C. C. 1994. *Machine Learning, Neural and Statistical Classification*. Prentice Hall.

Nyström, E. J. 1930. Über die praktische auflösung von integralgleichungen mit anwendungen auf randwertaufgaben. *Acta Mathematica* 54(1):185–204.

Talwalkar, A.; Kumar, S.; Mohri, M.; and Rowley, H. 2013. Large-scale svd and manifold learning. *Journal of Machine Learning Research* 14:3129–3152.

Wang, S., and Zhang, Z. 2013. Improving CUR matrix decomposition and the Nyström approximation via adaptive sampling. *Journal of Machine Learning Research* 14:2729–2769.

Wang, S., and Zhang, Z. 2014. Efficient algorithms and error analysis for the modified Nyström method. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Williams, C., and Seeger, M. 2001. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*.

Yang, T.; Li, Y.-F.; Mahdavi, M.; Jin, R.; and Zhou, Z.-H. 2012. Nyström method vs random fourier features: A theoretical and empirical comparison. In *Advances in Neural Information Processing Systems (NIPS)*.

Zhang, K., and Kwok, J. T. 2010. Clustered Nyström method for large scale manifold learning and dimension reduction. *IEEE Transactions on Neural Networks* 21(10):1576–1587.

Zhang, K.; Tsang, I. W.; and Kwok, J. T. 2008. Improved Nyström low-rank approximation and error analysis. In *International Conference on Machine Learning (ICML)*.

Zhang, Z. 2014. The matrix ridge approximation: algorithms and applications. *Machine Learning* 1–32.