Active Positive Semidefinite Matrix Completion: Algorithms, Theory and Applications

Aniruddha Bhargava
University of Wisconsin- Madison

Ravi Ganti
Walmart Labs, San Bruno

Robert Nowak
University of Wisconsin - Madison

Abstract

In this paper we provide simple, computationally efficient, active algorithms for completion of symmetric positive semidefinite matrices. Our proposed algorithms are based on adaptive Nystrom sampling, and are allowed to actively query any element in the matrix, and obtain a possibly noisy estimate of the queried element. We establish sample complexity guarantees on the recovery of the matrix in the max-norm and in the process establish new theoretical results, potentially of independent interest, on adaptive Nystrom sampling. We demonstrate the efficacy of our algorithms on problems in multi-armed bandits and kernel dimensionality reduction.

1 Introduction

The problem of matrix completion is a fundamental problem in machine learning and data mining where one needs to estimate an unknown matrix using only a few entries from the matrix. This problem has seen an explosion in interest in recent years perhaps fueled by the famous Netflix prize challenge Bell and Koren (2007) which required predicting the missing entries of a large movie-user rating matrix. Candès and Recht (2009) showed that by solving an appropriate semidefinite programming problem it is possible to recover a low-rank matrix given a few entries at random. Many improvements have since been made both on the theoretical side (Keshavan et al., 2009; Foygel and Srebro, 2011) as well as on the algorithmic side (Tan et al., 2014; Vandereycken, 2013; Wen et al., 2012).

Very often in applications the matrix of interest has more structure than just low rank. One such structure is positive semi-definiteness which appears when dealing with covariance matrices in applications like PCA, and kernel matrices when dealing with kernel learning. *In this paper we study the problem of matrix completion of low-rank, symmetric positive semidefinite (PSD) matrices and provide simple, and computationally efficient algorithms that actively query a few elements of the matrix and output an estimate of the matrix that is provably close to the true PSD matrix.* More precisely, we are interested in algorithms that ouptput a matrix that is provably (ϵ, δ) close to the true underlying matrix in the max norm [1]. This means that if $\boldsymbol{L}$ is the true, underlying PSD matrix then we want our algorithms to output a matrix $\hat{\boldsymbol{L}}$ such that $\|\hat{\boldsymbol{L}} - \boldsymbol{L}\|_{\max} \leq \epsilon$, with probability at least $1 - \delta$. Our goal is strongly motivated by applications to certain multi-armed bandit problem where there are a large number of arms. In certain cases the losses of these arms can be arranged as a PSD matrix and finding the (ϵ, δ) best arm can be reduced to the above defined (ϵ, δ) PSD matrix completion (PSD-MC) problem. Our contributions are as follows.

Let $\boldsymbol{L}$ be a $K \times K$ rank r PSD matrix, which is apriori unknown. We propose two models for the PSD-MC problem. In both the models the algorithm has access to an oracle $\mathcal{O}$ which when queried with a pair-of-indices (i, j) obtains a response $y_{i,j}$. The main difference between these two oracle models is the power of the oracle. In the first model, which we call as a deterministic oracle model, the oracle is a powerful, deterministic, but expensive oracle where $y_{i,j} = L_{i,j}$. In the second model, called as the stochastic oracle model, we shall assume that all the elements of the matrix $\boldsymbol{L}$ are in $[0, 1]$, and we have access to a less powerful, but cheaper oracle, whose output $y_{i,j}$ is sampled from a Bernoulli distribution with parameter $\boldsymbol{L}_{i,j}$. These models are sketched in Figure (3.1). We propose algorithms for PSD-MC problem, under the above two models. Our algorithms, called MCANS, in the deterministic oracle model, and S-MCANS [2] in the stochastic oracle model are both based on the following key insight: In the case of PSD matrices it is possible to find linearly independent columns by using few, adaptively chosen queries. In the case of S-MCANS we use the above insight along with techniques from multi-armed bandits literature in order to tackle the randomness of the stochastic oracle.

Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54. Copyright 2017 by the author(s).

[1]The max norm of a matrix is the maximum of the absolute value of all the elements in a matrix

[2]MCANS stands for Matrix Completion via Adaptive Nystrom Sampling. S in S-MCANS stands for stochastic

We prove that the MCANS algorithm outputs a $(\epsilon = 0, \delta = 0)$ estimate of the matrix $\boldsymbol{L}$ (exact recovery) after making at most $K(r+1)$ queries. We are able to avoid logarithmic factors, and coherence assumptions that are typically found in the matrix completion literature. We also prove that the S-MCANS algorithm outputs $\hat{\boldsymbol{L}}$ that is (ϵ, δ) close to $\boldsymbol{L}$ using queries that is linear in K and a low-order polynomial in the rank r of matrix $\boldsymbol{L}$.

Motivated by problems in advertising and search, where users are presented multiple items, and the presence of a user click reflects positive feedback, we consider a natural MAB problem in Section (5) where each user is presented with two items each time. The user may click on any of these presented items, and the goal is to discover the best pair-of-items. We show how this MAB problem can be reduced to a PSD-MC problem and how MCANS and S-MCANS can be used to find an (ϵ, δ) optimal arm using far fewer queries than standard MAB algorithms would need. We demonstrate experimental results on a movielens dataset. We also demonstrate the efficacy of the MCANS algorithm in a kernel dimensionality reduction task, where only a part of the kernel matrix is available.

We believe that our work makes contributions of independent interest to the literature on matrix completion, and MAB in the following ways. First, the MCANS algorithm is a simple algorithm that has optimal sample complexity when dealing with noiseless, active, PSD-MC problem. Specifically Lemma 3.1 shows a fundamental property of PSD matrices that we have not encountered in previous work. Second, by using techniques common in the MAB literature in the design of S-MCANS we show how to design algorithms for PSD-MC which are robust to query noise. In contrast, algorithms such as Nystrom sampling assume that they can access the underlying matrix without any noise. Third, most MC literature deals with error guarantees in spectral norm. In contrast, motivated by applications, we provide guarantees in the max-norm, which requires new techniques. Finally, using the spectral structure to solve the MAB problem in Section (5) is a novel contribution to the multi-armed bandit literature.

Notation. Δ_r represents the r dimensional probability simplex. Matrices and vectors are represented in bold font. For a matrix $\boldsymbol{L}$, unless otherwise stated, the notation $\boldsymbol{L}_{i,j}$ represents (i,j) element of $\boldsymbol{L}$, and $\boldsymbol{L}_{i:j,k:l}$ is the submatrix consisting of rows $i, \ldots, j$ and columns $k, \ldots, l$. The matrix $\|\cdot\|_1$ and $\|\cdot\|_2$ norms are always operator norms. The matrix $\|\cdot\|_{\max}$ is the element wise infinity norm. Finally, let $\mathbb{1}$ be the all 1 column vector.

2 Related Work

The problem of PSD-MC has been considered by many other authors (Bishop and Byron, 2014; Laurent and Varvitsiotis, 2014a,b). However, all of these papers consider the passive case, i.e. the entries of the matrix that have been revealed are not under their control. In contrast, we have an active setup, where we can decide which entries in the matrix to reveal. The Nystrom algorithm for approximation of low rank PSD matrices has been well studied both empirically and theoretically. Nyström methods typically choose random columns to approximate the original low-rank matrix (Gittens and Mahoney, 2013; Drineas and Mahoney, 2005). Adaptive schemes where the columns used for Nystrom approximation are chosen adaptively have also been considered in the literature. To the best of our knowledge these algorithms either need the knowledge of the full matrix (Deshpande et al., 2006) or have no provable theoretical guarantees (Kumar et al., 2012). Moreover, to the best of our knowledge all analysis of Nystrom approximation that has appeared in the literature assume that one can get error free values for entries in the matrix. Adaptive matrix completion algorithms have also been proposed and such algorithms have been shown to be less sensitive to the incoherence in the matrix (Krishnamurthy and Singh, 2013). The bandit problem that we study in the latter half of the paper is related to the problem of pure exploration in multi-armed bandits. In such pure exploration problems one is interested in designing algorithms with low, simple regret or designing algorithms with low (ϵ, δ) query complexity. Algorithms with small simple regret have been designed in the past (Audibert and Bubeck, 2010; Gabillon et al., 2011; Bubeck et al., 2013). Even-Dar et al. (2006) suggested the Successive Elimination (SE) and Median Elimination (ME) to find near optimal arms with provable sample complexity guarantees. These sample complexity guarantees typically scale linearly with the number of arms. In principal, one could naively reduce our problem to a pure exploration problem where we need to find an (ϵ, δ) good arm. However, such naive reductions ignore any dependency information among the arms. The S-MCANS algorithm that we design builds on the SE algorithm but crucially exploits the matrix structure to give much better algorithms than a naive reduction.

3 Algorithms in the deterministic oracle model

Our deterministic oracle model is shown in Figure (3.1) and assumes the existence of a powerful, deterministic oracle that returns queried entries of the unknown matrix accurately. Our algorithm in this model, called MCANS, is shown in Figure (3.2). It is an iterative algorithm that determines which columns of the matrix are independent. MCANS maintains a set of indices (denoted as $\mathcal{C}$ in the pseudo-code) corresponding to independent columns of matrix $\boldsymbol{L}$. Initially $\mathcal{C} = \{1\}$. MCANS then makes a single pass over the columns in $\boldsymbol{L}$ and checks if the current column is independent of the columns in $\boldsymbol{C}$. This check is done in line 5 of Figure (3.2) and most importantly requires *only the principal sub-matrix*, of $\boldsymbol{L}$, indexed by the set $\mathcal{C} \cup \{c\}$. If the column passes this test then all the elements in this column i whose values have not been queried

Model 3.1 Description of deterministic and stochastic oracle models

Figure (3.1)

```
1: while TRUE do
2:     Algorithm chooses a pair-of-indices $(i_t, j_t)$.
3:     Algorithm receives the response $y_t$ defined as follows
           $y_{t,\text{det}} = \boldsymbol{L}_{i_t,j_t}$ // if model is deterministic   (1)
           $y_{t,\text{stoc}} = \text{Bern}(\boldsymbol{L}_{i_t,j_t})$ // if model is stochastic   (2)
4:     Algorithm stops if it has found a good approximation to the unknown matrix $\boldsymbol{L}$.
5: end while
```

in the past are queried and the matrix $\hat{\boldsymbol{L}}$ is updated with these values. The test in line 5 is the column selection step of the MCANS algorithm and is justified by Lemma (3.1). Finally, once r independent columns have been chosen, we impute the matrix by using Nystrom extension. Nystrom based methods have been proposed in the past to handle large scale kernel matrices in the kernel based learning literature Drineas and Mahoney (2005); Kumar et al. (2012). The major difference between the above work and ours is that the column selection procedure in our algorithms is deterministic, whereas in Nystrom methods columns are chosen at random. Lemma (3.1) and Theorem (3.2) provide the proof-of-correctness and the sample complexity guarantees for Algorithm (3.2).

Algorithm 3.2 Matrix Completion via Adaptive Nystrom Sampling (MCANS)

```
Input: A deterministic oracle that takes a pair of indices $(i, j)$ and outputs $\boldsymbol{L}_{i,j}$.
Output: $\hat{\boldsymbol{L}}$
1: Choose the pairs $(j, 1)$ for $j = 1, 2, \ldots, K$ and set $\hat{\boldsymbol{L}}_{j,1} = \boldsymbol{L}_{j,1}$. Also set $\hat{\boldsymbol{L}}_{1,j} = \boldsymbol{L}_{j,1}$
2: $\mathcal{C} = \{1\}$ {Set of independent columns discovered till now}
3: for $(c = 2;\ c \leftarrow c + 1;\ c \leq K)$ do
4:     Query the oracle for $(c, c)$ and set $\hat{\boldsymbol{L}}_{c,c} \leftarrow \boldsymbol{L}_{c,c}$
5:     if $\sigma_{\min}\left(\hat{\boldsymbol{L}}_{\mathcal{C}\cup\{c\},\mathcal{C}\cup\{c\}}\right) > 0$ then
6:         $\mathcal{C} \leftarrow \mathcal{C} \cup \{c\}$
7:         Query $\mathcal{O}$ for the pairs $(\cdot, c)$ and set $\hat{\boldsymbol{L}}(\cdot, c) \leftarrow \boldsymbol{L}(\cdot, c)$ and by symmetry $\hat{\boldsymbol{L}}(c, \cdot) \leftarrow \boldsymbol{L}(\cdot, c)$.
8:     end if
9:     if $(|\mathcal{C}| = r)$ then
10:        break
11:    end if
12: end for
13: Let $\boldsymbol{C}$ denote the tall matrix comprised of the columns of $\boldsymbol{L}$ indexed by $\mathcal{C}$ and let $\boldsymbol{W}$ be the principle submatrix of $\boldsymbol{L}$ corresponding to the indices in $\mathcal{C}$. Then, construct the Nystrom extension $\widehat{\boldsymbol{L}} = \boldsymbol{C}\boldsymbol{W}^{-1}\boldsymbol{C}^\top$.
```

Lemma 3.1. *Let $\boldsymbol{L}$ be any PSD matrix of size K. Given a subset $\mathcal{C} \subset \{1, 2, \ldots, K\}$, the columns of the matrix $\boldsymbol{L}$ indexed by the set $\mathcal{C}$ are independent iff the principal submatrix $\boldsymbol{L}_{\mathcal{C},\mathcal{C}}$ is non-degenerate, equivalently iff, $\lambda_{\min}(\boldsymbol{L}_{\mathcal{C},\mathcal{C}}) > 0$.*

Proof. Suppose $\boldsymbol{L}_{\cdot,\mathcal{C}}$ is degenerate. The there exists $\boldsymbol{x} \neq 0$ with $\boldsymbol{x}_j = 0\ \forall j \in \mathcal{C}$ such that $\boldsymbol{L}_{\cdot,\mathcal{C}}\boldsymbol{x} = \boldsymbol{L}_{\mathcal{C},\mathcal{C}}\boldsymbol{x}_{\mathcal{C}} = 0$. Therefore $x_{\mathcal{C}}^\top \boldsymbol{L}_{\mathcal{C},\mathcal{C}}\boldsymbol{x}_{\mathcal{C}} = 0$ showing $\boldsymbol{L}_{\mathcal{C},\mathcal{C}}$ is degenerate.

Now assume that $\boldsymbol{L}_{\mathcal{C},\mathcal{C}}$ is degenerate. Then $\boldsymbol{z}$ such that $\boldsymbol{z}^\top \boldsymbol{L}_{\mathcal{C},\mathcal{C}}\boldsymbol{z} = 0$. Now notice that setting $x_i = 0,\ i \notin \mathcal{C}$ and $\boldsymbol{x}_i = \boldsymbol{z}_i,\ i \in \mathcal{C}$, $\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} = \boldsymbol{z}^\top \boldsymbol{L}_{\mathcal{C},\mathcal{C}}\boldsymbol{z} = 0$. Therefore, $\boldsymbol{x}$ is a minimizer of the quadratic $\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x}$. This satisfies the property that its gradient vanishes. i.e. $\boldsymbol{L}\boldsymbol{x} = 0$. Therefore, $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{L}_{\cdot,\mathcal{C}}\boldsymbol{z} = 0$. Therefore $\boldsymbol{L}_{\cdot,\mathcal{C}}$ is degenerate [3] □

Theorem 3.2. *If $\boldsymbol{L} \in \mathbb{R}^{K\times K}$ is an PSD matrix of rank r, then the matrix $\hat{\boldsymbol{L}}$ output by the MCANS algorithm* (3.2) *satisfies $\hat{\boldsymbol{L}} = \boldsymbol{L}$. Moreover, the number of oracle calls made by MCANS is at most $K(r + 1)$. The sampling algorithm* (3.2) *requires: $K + \ldots + (K - (r - 1)) + (K - r) \leq (r + 1)K$ samples from the matrix $\boldsymbol{L}$.*

Note that the sample complexity of the MCANS algorithm is better than typical sample complexity results for LRMC and Nystrom methods. We managed to avoid factors logarithmic in dimension and rank that appear in LRMC and Nystrom methods (Gittens and Mahoney, 2013), as well as incoherence factors that are typically found in LRMC results (Candès and Recht, 2009). Also, our algorithm is purely deterministic, whereas LRMC uses randomly drawn samples from a matrix. In fact, this careful, deterministic choice of entries of the matrix is what helps us do better than LRMC.

Moreover, MCANS algorithm is optimal in a min-max sense. This is because any PSD matrix of size K and rank r is characterized via its singular value decomposition by Kr degrees of freedom. Hence, any algorithm for completion of an PSD matrix would need to see at least Kr entries. As shown in theorem (3.2) the MCANS algorithm makes at most $K(r + 1)$ queries and hence is order optimal.

The MCANS algorithm needs the rank r as an input. However, the MCANS algorithm can be made to work even if r is unknown by simply removing the condition on line 9 in the MCANS algorithm. In this case, once r independent columns have been found, all future checks on the if statement in line 5 of MCANS will fail, and the algorithm eventually exits the for loop. Even in this case the sample complexity guarantees in Theorem (3.2) hold. Finally, if the matrix is not exactly rank r but can be approximated by a matrix of rank r, then we might be able to modify

[3] Proof of Theorem (3.2) is in the appendix.

MCANS to output the best rank r approximation, by modifying line 5 to use an appropriate $\sigma_{\text{thresh}} > 0$. We leave this modification to future work.

4 Algorithms in the stochastic oracle model

For the stochastic model considered in this paper we shall propose an algorithm, called S-MCANS, which is a stochastic version of MCANS. Like MCANS, the stochastic version discovers a set of independent columns iteratively and then uses the Nyström extension to impute the matrix. Figure (4.1) provides a pseudo-code of the S-MCANS algorithm.

S-MCANS like the MCANS algorithm repeatedly performs column selection steps to select a column of the matrix $\boldsymbol{L}$ that is linearly independent of the previously selected columns, and then uses these selected columns to impute the matrix via a Nystrom extension. In the case of deterministic models, due to the presence of a deterministic oracle, the column selection step is pretty straight-forward and requires calculating the smallest singular-value of certain principal sub-matrices. In contrast, for stochastic models the stochastic oracle outputs a Bernoulli random variable $\mathrm{Bern}(\boldsymbol{L}_{i,j})$ when queried with the indices (i, j). This makes the column selection step much harder. We resort to the successive elimination algorithm (shown in Fig (4.2)) where principal sub-matrices are repeatedly sampled to estimate the smallest singular-values for those matrices. The principal sub-matrix that has the largest smallest singular-value determines which column is selected in the column selection step.

Given a set $\mathcal{C}$, define $\boldsymbol{C}$ to be a $K \times r$ matrix corresponding to the columns of $\boldsymbol{L}$ indexed by $\mathcal{C}$ and define $\boldsymbol{W}$ to be the $r \times r$ principal submatrix of $\boldsymbol{L}$ corresponding to indices in $\mathcal{C}$. S-MCANS constructs estimators $\widehat{\boldsymbol{C}}, \widehat{\boldsymbol{W}}$ of $\boldsymbol{C}, \boldsymbol{W}$ respectively by repeatedly sampling independent entries of $\boldsymbol{C}, \boldsymbol{W}$ (which are Bernoulli) for each index and averaging these entries. The sampling is such that each entry of the matrix $\boldsymbol{C}$ is sampled at least m_1 times and each entry of the matrix $\boldsymbol{W}$ is sampled at least m_2 times, where

$$m_1 = 100 C_1(\boldsymbol{W}, \boldsymbol{C}) \log(2Kr/\delta) \max\left(\frac{r^{5/2}}{\epsilon}, \frac{r^2}{\epsilon^2}\right) \quad (3)$$

$$m_2 = 200 C_2(\boldsymbol{W}, \boldsymbol{C}) \log(2r/\delta) \max\left(\frac{r^3}{\epsilon}, \frac{r^5}{\epsilon^2}\right) \quad (4)$$

and C_1, C_2 are problem dependent constants defined as:

$$\begin{aligned} C_1(\boldsymbol{W}, \boldsymbol{C}) = \max\big(&\|\boldsymbol{W}^{-1}\boldsymbol{C}^\top\|_{\max}, \|\boldsymbol{W}^{-1}\boldsymbol{C}^\top\|_{\max}^2 \\ &\|\boldsymbol{W}^{-1}\|_{\max}, \|\boldsymbol{C}\boldsymbol{W}^{-1}\|_1^2, \\ &\|\boldsymbol{W}^{-1}\|_2 \|\boldsymbol{W}^{-1}\|_{\max}\big) \end{aligned} \quad (5)$$

$$\begin{aligned} C_2(\boldsymbol{W}, \boldsymbol{C}) = \max\big(&\|\boldsymbol{W}^{-1}\|_2^2 \|\boldsymbol{W}^{-1}\|_{\max}^2, \\ &\|\boldsymbol{W}^{-1}\|_2 \|\boldsymbol{W}^{-1}\|_{\max}, \\ &\|\boldsymbol{W}^{-1}\|_2, \|\boldsymbol{W}^{-1}\|_2^2\big) \end{aligned} \quad (6)$$

S-MCANS then returns the Nyström extension constructed using matrices $\widehat{\boldsymbol{C}}, \widehat{\boldsymbol{W}}$.

Algorithm 4.1 Stochastic Matrix Completion via Adaptive Nystrom Sampling (S-MCANS)

```
Input: $\epsilon > 0, \delta > 0$ and a stochastic oracle $\mathcal{O}$ that when
    queried with indices $(i, j)$ outputs a Bernoulli random
    variable $\mathrm{Bern}(L_{i,j})$
Output: A PSD matrix $\hat{\boldsymbol{L}}$, which is an approximation to
    the unknown matrix $\boldsymbol{L}$, such that with probability at
    least $1 - \delta$, all the elements of $\hat{\boldsymbol{L}}$ are within $\epsilon$ of the
    elements of $\boldsymbol{L}$.
1: $\mathcal{C} \leftarrow \{1\}$.
2: $\mathcal{I} \leftarrow \{2, 3, \dots, K\}$.
3: for ($t = 2; t \leftarrow t + 1; t \leq r$) do
4:     Define, $\tilde{\mathcal{C}_i} = \mathcal{C} \bigcup \{i\}, \forall i \in \mathcal{I}$.
5:     Run the successive elimination algorithm 4.2 on matrices $\boldsymbol{L}_{\tilde{\mathcal{C}}_i, \tilde{\mathcal{C}}_i}$, $i \in \mathcal{I}$, with given $\delta \leftarrow \frac{\delta}{2r}$ to get $i_t^\star$.
6:     $\mathcal{C} \leftarrow \mathcal{C} \bigcup \{i_t^\star\}; \mathcal{I} \leftarrow \mathcal{I} \setminus \{i_t^\star\}$.
7: end for
8: Obtain estimators $\widehat{\boldsymbol{C}}, \widehat{\boldsymbol{W}}$ of $\boldsymbol{C}, \boldsymbol{W}$ by repeatedly sampling and averaging entries. Calculate the Nystrom extension $\widehat{\boldsymbol{L}} = \widehat{\boldsymbol{C}}\widehat{\boldsymbol{W}}^{-1}\widehat{\boldsymbol{C}}^\top$.
```

4.1 Sample complexity of the S-MCANS algorithm

As can be seen from the S-MCANS algorithm, samples are consumed both in the successive elimination steps (step 5 of S-MCANS) as well as during the construction of the Nyström extension. We analyze both these steps next.

Sample complexity analysis of successive elimination. Before we provide a sample complexity analysis of the S-MCANS algorithm, we need a bound on the spectral norm of random matrices with 0 mean where each element is sampled possibly different number of times. This bound plays a key role in correctness of the successive elimination algorithm. The proof of this bound follows from matrix Bernstein inequality. We relegate the proof to the appendix due to lack of space.

Lemma 4.1. *Let $\hat{\boldsymbol{P}}$ be a $p \times p$ random matrix that is constructed as follows. For each index (i, j), set $\hat{\boldsymbol{P}}_{i,j} = \frac{H_{i,j}}{n_{i,j}}$, where $H_{i,j}$ is an independent random variable drawn from the distribution* $\mathrm{Binomial}(n_{i,j}, p_{i,j})$. *Then, $||\hat{\boldsymbol{P}} - \boldsymbol{P}||_2 \leq \frac{2\log(2p/\delta)}{3 \min_{i,j} n_{i,j}} + \sqrt{\frac{\log(2p/\delta)}{2} \sum_{i,j} \frac{1}{n_{i,j}}}$. Furthermore, if we denote by Δ the R.H.S. in the above bound, then $|\sigma_{\min}(\hat{\boldsymbol{P}}) - \sigma_{\min}(\boldsymbol{P})| \leq \Delta$.*

Lemma 4.2. *The successive elimination algorithm shown in Figure* (4.2) *on m square matrices of size $\boldsymbol{A}_1, \dots, \boldsymbol{A}_m$ each of size $p \times p$ outputs an index $i_\star$ such that, with probability at least $1 - \delta$, the matrix $\boldsymbol{A}_{i_\star}$ has the largest minimum singular value among all the input matrices. Let, $\Delta_{k,p} := \max_{j=1,\dots,m} \sigma_{\min}(\boldsymbol{A}_j) - \sigma_{\min}(\boldsymbol{A}_k)$. Then num-*

Algorithm 4.2 Successive elimination on principal submatrices

```
Input: Square matrices $A_1, \dots, A_m$ of size $p \times p$, which share the same $p-1 \times p-1$ left principal submatrix; a failure probability $\delta > 0$; and a stochastic oracle $\mathcal{O}$
Output: An index
1:  Set $t = 1$, and $\mathcal{S} = \{1, 2, \dots, m\}$ (Here $m = K - \tau + 1$ where $\tau$ is the iteration number in MCANS, when successive elimination is invoked).
2:  Sample each entry of the input matrices once.
3:  while $|\mathcal{S}| > 1$ do
4:      Set $\delta_t = \frac{6\delta}{\pi^2 m t^2}$
5:      Let $\widehat{\sigma}^{\max} = \max_{k \in \mathcal{S}} \sigma_{\min}(\widehat{A}_k)$ and let $k_\star$ be the index that attains argmax.
6:      For each $k \in \mathcal{S}$, define $\alpha_{t,k} = \frac{2\log(2p/\delta_t)}{3 \min_{i,j} n_{i,j}(\widehat{A}_k)} + \sqrt{\frac{\log(2p/\delta_t)}{2} \sum_{i,j} \frac{1}{n_{i,j}(\widehat{A}_k)}}$
7:      For each index $k \in \mathcal{S}$, if $\widehat{\sigma}^{\max} - \widehat{\sigma}_{\min}(\widehat{A}_k) \geq \alpha_{t,k^\star} + \alpha_{t,k}$ then do $\mathcal{S} \leftarrow \mathcal{S} \setminus \{k\}$.
8:      $t \leftarrow t + 1$
9:      Sample each entry of the matrices indexed by the indices in $\mathcal{S}$ once.
10: end while
11: Output $k$, where $k \in \mathcal{S}$.
```

ber of queries to the stochastic oracle are

$$\sum_{k=2}^{m} O\left(p^3 \log(2p\pi^2 m^2/3\Delta_{k,p}^2\delta)/\Delta_{k,p}^2\right) + O\left(p^4 \max_k \log(2p\pi^2 m^2/3\Delta_{k,p}^2\delta)/\Delta_{k,p}^2\right) \quad (7)$$

Sample complexity analysis of Nystrom extension. The following theorem tells us how many calls to a stochastic oracle are needed in order to guarantee that the Nystrom extension obtained by using matrices $\widehat{C}, \widehat{W}$ is accurate with high probability. The proof has been relegated to the appendix.

Theorem 4.3. *Consider the matrix $\widehat{C}\widehat{W}^{-1}\widehat{C}^\top$ which is the Nystrom extension constructed in step 10 of the S-MCANS algorithm. Given any $\delta \in (0,1)$, with probability at least $1-\delta$, $\left\|CW^{-1}C^\top - \widehat{C}\widehat{W}^{-1}\widehat{C}^\top\right\|_{\max} \leq \epsilon$ after making a total of $Krm_1 + r^2 m_2$ number of oracle calls to a stochastic oracle, where m_1, m_2 are given in equations* (3), (4).

The following corollary follows directly from theorem (4.3), and lemma (4.2).

Corollary 4.4. *The S-MCANS algorithm outputs an (ϵ, δ) good arm after making at most*

$$Krm_1 + r^2 m_2 + \sum_{p=1}^{r} \sum_{k=2}^{K-r} \tilde{O}\left(\frac{p^3}{\Delta_{k,p}^2} + p^4 \max_k \frac{1}{\Delta_{k,p}^2}\right)$$

number of calls to a stochastic oracle, where $\tilde{O}$ hides factors that are logarithmic in $K, r, \frac{1}{\delta}, 1/\Delta_{k,p}$, and m_1, m_2 are given in equations (3), (4).

In principal, precise values of m_1, m_2 given in equations (3), (4) are application dependent, and often unknown apriori. If, for a given PSD matrix L, and for all possible choices of submatrices C of L, which admit an invertible principal $r \times r$ sub-matrix W, the terms involved in Equation (3), (4) can be upper bounded by a universal constant $\theta(L)$, then one can use $\theta(L)$ instead of the terms $C_1(W, C), C_2(W, C)$ in the expressions for m_1, m_2 in equations (3), (4). For our experiments, we assume that we are given some sampling budget B that we can use to query elements of the matrix L, and once we run out of this budget we stop and report the necessary error metrics. As we see MCANS and S-MCANS allow us to properly allocate our budget to obtain good estimates of the matrix L.

5 Applications to multi-armed bandits

We shall now look at a multi-armed bandit (MAB) problem where there are a large number of arms and show how this MAB problem can be reduced to a PSD-MC problem. To motivate the MAB problem consider the following example: Suppose an advertising engine wants to show different advertisements to users. Each incoming user belongs to one of r different unknown sub-populations. Each sub-population may have different taste in advertisements. For example, if there are $r = 3$ sub-populations, then sub-population P_1 may like advertisements about vacation rentals, while P_2 may like advertisements about car rentals and population P_3 may like advertisements about motorcycles. Suppose, the advertising company has a constraint that it can show only two advertisements each time to a random, unknown incoming user. The question of interest is what would be a good pair of advertisements to show to a random incoming user in order to maximize click probability?

Such problems and more can be cast in a MAB framework, where the MAB algorithm actively elicits response from users on different pairs of advertisements. In Figure (5.1) we sketch the two models for the above mentioned advertising problem. In both the models, there are K ads in total, and in each round t, we choose a pair of ads and receive a reward which is a function of the pair. Let, Z_t be a multinomial random variable defined by a probability vector $\boldsymbol{p} \in \Delta_r$, whose output space is the set $\{1, 2, \dots, r\}$. Let $\boldsymbol{u}_{Z_t}$ be a reward vector in $[0,1]^K$ indexed by Z_t. On displaying the pair of ads (i_t, j_t) in round t the algorithm receives a scalar reward y_t. This reward is large if either of the ads in the chosen pair is "good". For both the models we are interested in designing algorithms that discover an (ϵ, δ) best pair of ads using as few trials as possible, i.e. algorithms which can output, with probability at least $1-\delta$, a pair of ads that is ϵ close to the best pair of ads in terms of the expected reward of the pair. The difference between the models is whether the reward is stochastic or deterministic. In the deterministic model y_t is deterministic and is

Model 5.1 Description of our proposed models

```
1: while TRUE do
2:    In the case of stochastic model, nature chooses $Z_t \sim \mathrm{Mult}(\boldsymbol{p})$, but does not reveal it to the algorithm.
3:    Algorithm chooses a pair of items $(i_t, j_t)$.
4:    Algorithm receives the reward $y_t$ defined as follows:
      If the model is deterministic
          $y_{t,\text{det}} = 1 - \mathbb{E}_{Z_t \sim \boldsymbol{p}}(1-\boldsymbol{u}_{Z_t}(i_t))(1-\boldsymbol{u}_{Z_t}(j_t))$   (8)
      If the model is stochastic
          $y_{t,\text{stoc}} = \max\{y_{i_t}, y_{j_t}\}$   (9)
          $y_{i_t} \sim \mathrm{Bern}(\boldsymbol{u}_{Z_t}(i_t))$   (10)
          $y_{j_t} \sim \mathrm{Bern}(\boldsymbol{u}_{Z_t}(j_t))$   (11)
5:    Algorithm stops if it has found a certifiable $(\epsilon, \delta)$ optimal pair of items.
6: end while
```

given by Equation (8), whereas in the stochastic model y_t is a random variable that depends on the random variable Z_t as well as additional external randomness. However, a common aspect of both these models is that the expected reward associated with the pair of choices (i_t, j_t) in round t is the same and is equal to the expression given in Equation (8). It is clear from Figure (5.1) that the optimal pair of ads satisfies the equation

$$(i_\star, j_\star) = \arg\min\nolimits_{i,j} \mathbb{E}_{Z_t \sim \boldsymbol{p}}(1-\boldsymbol{u}_{Z_t}(i))(1-\boldsymbol{u}_{Z_t}(j)). \tag{12}$$

A naive way to solve this problem is to treat this problem as a best-arm identification problem in stochastic multi-armed bandits where there are $\Theta(K^2)$ arms each corresponding to a pair of items. One could now run a Successive Elimination (SE) algorithm or a Median Elimination algorithm on these $\Theta(K^2)$ pairs Even-Dar et al. (2006) to find an (ϵ, δ) optimal pair. The sample complexity of the SE or ME algorithms on these $\Theta(K^2)$ pairs would be roughly $\tilde{\mathcal{O}}(\frac{K^2}{\epsilon^2})$ [4]. In the advertising application that we mentioned before and other applications K can be very large, and therefore the sample complexity of such naive algorithms can be very large. However, these simple reductions throw away information between different pairs of items and hence are sub-optimal. We next show that via a simple reduction it is possible to convert this MAB problem to a PSD-MC problem.

[4]The $\tilde{\mathcal{O}}$ notation hides logarithmic dependence on $\frac{1}{\delta}, K, \frac{1}{\delta}$

5.1 Reduction from MAB to PSD matrix completion

Since, we are interested in returning an (ϵ, δ) optimal pair of ads it is enough if the pair returned by our algorithm attains an objective function value that is at most ϵ more than the optimal value of the objective function shown in equation (12), with probability at least $1-\delta$. Let $\boldsymbol{p} \in \Delta_r$, and let the reward matrix $\boldsymbol{R} \in \mathbb{R}^{K \times K}$ be such that its $(i,j)^{\text{th}}$ entry is the expected reward obtained using the pair of ads (i,j). Then from equation (12) we know that the $(i,j)^{\text{th}}$ element of matrix $\boldsymbol{R}$ has the form

$$\begin{aligned}\boldsymbol{R}_{i,j} &= 1 - \mathbb{E}_{Z_t \sim \boldsymbol{p}}(1-\boldsymbol{u}_{Z_j}(i))(1-\boldsymbol{u}_{Z_j}(j)) \\ &= 1 - \sum_{k=1}^{r} \boldsymbol{p}_k (1-\boldsymbol{u}_k(i))(1-\boldsymbol{u}_k(j)) \qquad (13)\end{aligned}$$

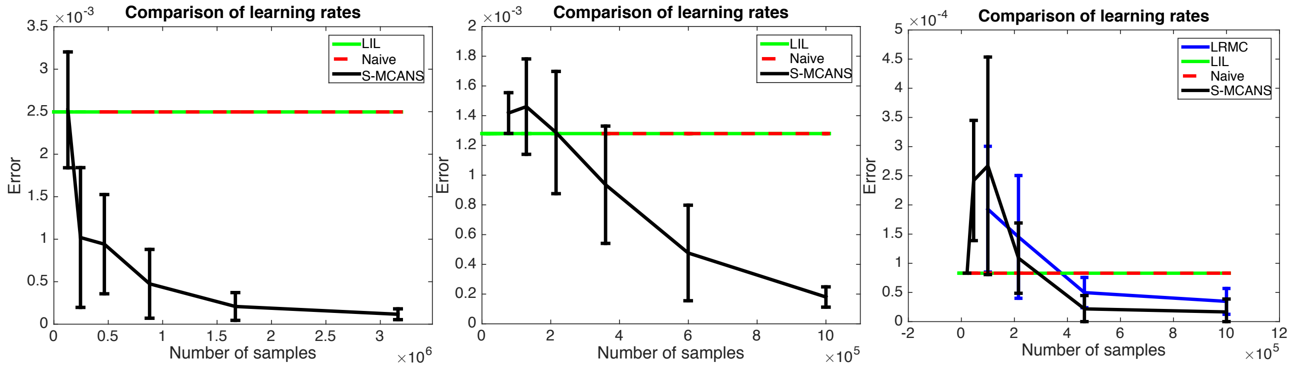
$$\boldsymbol{R} = \mathbb{1}\mathbb{1}^\top - \underbrace{\sum_{k=1}^{r} \boldsymbol{p}_k (\mathbb{1}-\boldsymbol{u}_k)(\mathbb{1}-\boldsymbol{u}_k)^\top}_{\boldsymbol{L}}. \tag{14}$$

It is enough to find an entry in the matrix $\boldsymbol{L}$ that is ϵ close to the smallest entry in the matrix $\boldsymbol{L}$ with probability at least $1-\delta$. In order to do this it is enough to estimate the matrix $\boldsymbol{L}$ using repeated trials and then use the pair-of-indices corresponding to the smallest entry as an (ϵ, δ) optimal pair. In order to do this we exploit the structural properties of matrix $\boldsymbol{L}$. From equation (14) it is clear that the matrix $\boldsymbol{L}$ can be written as a sum of r rank-1 matrices. Hence $\mathrm{rank}(\boldsymbol{L}) \le r$. Furthermore, since these rank-1 matrices are all positive semi-definite and $\boldsymbol{L}$ is a convex combination of such, we can conclude that $\boldsymbol{L} \succeq 0$. We have proved the following proposition:

Proposition 5.1. *The matrix* $\boldsymbol{L}$ *shown in equation* (14) *satisfies the following two properties: (i)* $\mathrm{rank}(\boldsymbol{L}) \le r$ *(ii)* $\boldsymbol{L} \succeq 0$.

The above property immediately implies that we can treat the MAB problem as a MC-PSD problem.

Proposition 5.2. *The* (ϵ, δ) *optimal pair for the MAB problem shown in model* (5.1) *with deterministic rewards can be reduced to a PSD-MC problem with a deterministic oracle. Using the MCANS algorithm we can obtain a* $(0, 0)$ *optimal arm using less than* $(r+1)K$ *queries. Similarly, the* (ϵ, δ) *optimal pair for the MAB problem shown in Figure* (5.1)*, under the stochastic model can be reduced to a PSD-MC problem with a stochastic oracle. Using the S-MCANS algorithm we can obtain an* (ϵ, δ) *optimal pair-of-arms using number of trials equal to the quantity shown in Corollary* (4.4).



(a) ML-100K; $K = 800, r = 2$ (b) ML-100K; $K = 800, r = 4$ (c) ML-1M; $K = 200, r = 2$

Figure 1: Error of various algorithms with increasing budget. The error is defined as $\boldsymbol{L}_{\hat{i},\hat{j}} - \boldsymbol{L}_{i_\star,j_\star}$ where $(\hat{i},\hat{j})$ is a pair of optimal choices as estimated by each algorithm. Note that the Naive and LIL' UCB have similar performances and do not improve with budget and both are outperformed by S-MCANS. This is because both Naive and LIL' UCB have few samples that they can use on each pair of movies. All experiments were repeated 10 times.

5.2 Related work to multi-armed bandits

Bandit problems where multiple actions are selected have also been considered in the past usually in the context of computational advertising (Kale et al., 2010), information retrieval Radlinski et al. (2008), Yue and Guestrin (2011), resource allocation Streeter and Golovin (2009). A major difference between the above mentioned works and our work is that our feedback and reward model is different and that we are not interested in cumulative regret guarantees but rather in finding a good pair of arms as quickly as possible. Furthermore our linear-algebraic approach to the problem is very different from the approaches taken in the previous papers. Finally we would like to mention that our model shown in Figure (5.1) on the surface bears resemblance to dueling bandit problems (Yue et al., 2012). However, in duleing bandits two arms are compared which is not the case in the bandit problem that we study. A more thorough literature survey has been relegated to the appendix due to lack of space.

6 Experiments

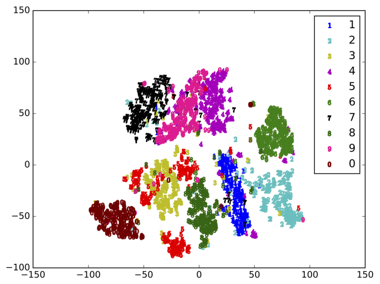
In this section we demonstrate experiments to show the efficacy of our proposed algorithms: MCANS and S-MCANS.

6.1 Movie reommendation as a MAB problem

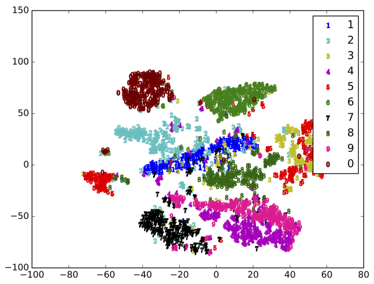
We describe a multi-armed bandit task where the target is to recommend a good pair of movies to users.

Experimental setup. We used the Movie Lens datasets (Harper and Konstan, 2015), namely ML-100K, ML-1M. This dataset contains incomplete movie ratings provided by users for different movies. We pre-process this dataset to make it suitable for a bandit experiment as follows: We use this incomplete user-movie ratings dataset as an input to an LRMC solver called OptSpace. The complete ratings obtained from an LRMC solver are then thresholded to obtain binary values. More precisely, all ratings of at least 3 are set to 1 and ratings less than 3 are set to 0. All the users are assigned to different sub-populations, based on some attribute of the user. For example in Figures (1a), (1b) the gender attribute is used, to create 2 sub-populations and in Figure (1b) occupation of the user is used to define the resulting 4 sub-populations. In the final step we averaged the binary ratings of all users in a certain population to get the probability that a random user from a given sub-population likes a certain movie. This gets us matrices $\boldsymbol{R}$ and $\boldsymbol{L} = 1 - \boldsymbol{R}$. In the experiments we provide the different algorithms with increasing budget and measure the error of each algorithm in finding the best pair of movies. The algorithms that we use for comparison are Naive, LiL'UCB (Jamieson et al., 2014) and LRMC using OptSpace (Keshavan et al., 2009). The naive algorithm uniformly distributes the given budget equally among all the $K(K + 1)/2$ pairs of movies. LIL applies the LiL' UCB algorithm treating each pair of movies as an arm in a stochastic multi-armed bandit game. All algorithms can access entries of the matrix $\boldsymbol{L}$ via noisy queries of the form (i, j) and obtain a Bernoulli outcome with probability $\boldsymbol{L}_{i,j}$. No other information such as sub-populations are available to any of the algorithms. The setup faithfully imitates the stochastic oracle model shown in Figure (5.1).

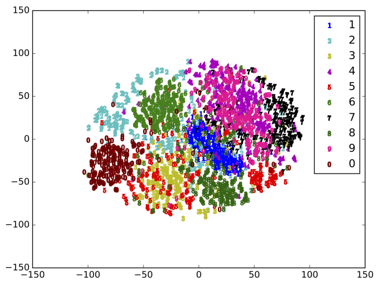
As can be seen from the figures (1) the Naive and LIL'UCB algorithms have similar performance on all the datasets. On the ML-100K datasets LIL'UCB quickly finds a good pair of movies but fails to improve with an increase in the budget. To see why, observe that there are about 32×10^4 pairs of movies. The maximum budget here is on the order of 10^6. Therefore, Naive samples each of those pairs on an average at most four times. Since many entries in the matrix are of the order of 10^{-4}, Naive algorithm a lot of sees 0's when sampling. The same thing happens with the LIL'UCB algorithm too; very few samples are avail-



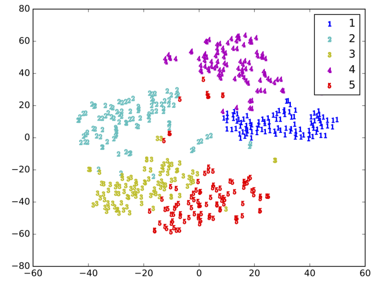
(a) t-SNE with the full kernel matrix .



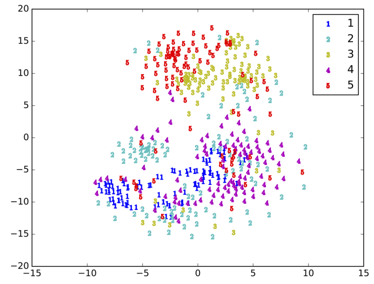
(b) t-SNE + MCANS, $B = 250K$



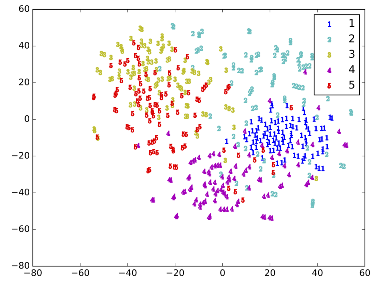
(c) t-SNE + LRMC, $B = 250K$



(d) t-SNE with the full kernel matrix.



(e) t-SNE + MCANS, $B = 8K$



(f) t-SNE + LRMC, $B = 8K$

Figure 2: $2-$ dimensional visualization obtained by using a partially observed RBF kernel matrix with the t-SNE algorithm for kernel dimensionality reduction. The budget B specifies how many entries of the RBF kernel matrix the algorithms are allowed to query. The kernel matrix obtained using MCANS and LRMC are then fed into t-SNE. The first row shows resuls on the MNIST2500 dataset and the second row shows results on USPS500 dataset, obtained by subsampling digits $1 - 5$ of the USPS dataset. Figures (2a),(2d) shows the result of KDR when the entire kernel matrix is observed. The MNIST2500 dataset is available at `https://lvdmaaten.github.io/tsne/`

able for each pair to improve its confidence bounds. This explains why the Naive and LIL' UCB algorithm have such poor and similar performances. In contrast, S-MCANS focuses most of the budget on a few select pairs and infers the value of other pairs via Nystrom extensio. This is why, in our experiments we see that S-MCANS finds good pair of movies quickly and finds even better pairs with increasing budget, outperforming all the other algorithms. S-MCANS is also better than LRMC, because we specifically exploit the SPSD structure in our matrix $\boldsymbol{L}$, which enables us to do better. We would like to mention that on ML-100K dataset the performance of LRMC was much inferior and this result and more results are in the appendix.

6.2 Kernel dimensionality reduction under budget

Kernel based dimensionality reduction (KDR) is a suite of powerful non-linear dimensionality reduction techniques which all use a kernel matrix in order to perform dimensionality reduction. Given a collection of points residing in a d dimensional space where d is very large most KDR based techniques require constructing a kernel matrix between all pairs of points. A popular kernel matrix used in KDR is an RBF kernel matrix, obtained using all pairwise distances. Calculating all pairwise distances takes $O(K^2 d)$ time which can be large when d is very high. Hence, we need algorithms that can use only a few pairwise distance measurements and use the incomplete kernel matrix to perform dimensionality reduction. Given a budget of $B = O(Kr)$, where r is the approximate rank of the kernel matrix, we expect MCANS to construct a good approximation of the underlying kernel matrix. This matrix is in turn used for KDR. In the experiments shown in this section, we want to investigate how the estimate of the kernel matrix provided by MCANS and LRMC effect KDR. In order to do this we use as our true kernel matrix $\boldsymbol{L}$ a matrix obtained by applying the RBF kernel to all pairs of points. All algorithms are assumed to have an access to a deterministic, oracle that can query at the most B entries of $\boldsymbol{L}$. We compare MCANS with LRMC using SoftImpute (Mazumder et al., 2010) as implemented in the python package fancyimpute. For the LRMC implementation we sample B indices randomly from the upper triangle of the kernel matrix $\boldsymbol{L}$, and use these sampled values in the corresponding lower triangle too. The completed matrices are then used in t-SNE (Maaten and Hinton, 2008) to visualize the USPS digits dataset and the MNIST2500 dataset. As can be seen in Figure (2), t-SNE with MCANS generates clusters which are comparable in quality to the ones obtained using full kernel matrix. However, the LRMC algorithm when used with t-SNE output poor quality clusters.

7 Conclusions

In this paper we proposed theoretically sound active algorithms for the problem of positive semi-definite matrix completion in the presence of deterministic and stochastic oracles and applications shown. In the future we will look at applications to graphical models and kernel machines.

8 Acknowledgements

AB, and RG made equal contributions to the paper.

References

J.-Y. Audibert and S. Bubeck. Best arm identification in multi-armed bandits. In *COLT*, 2010.

R. M. Bell and Y. Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

W. E. Bishop and M. Y. Byron. Deterministic symmetric positive semidefinite matrix completion. In *Advances in Neural Information Processing Systems*, pages 2762–2770, 2014.

S. Bubeck, T. Wang, and N. Viswanathan. Multiple identifications in multi-armed bandits. In *ICML*, 2013.

E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *FOCM*, 2009.

A. Deshpande, L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via volume sampling. In *SODA*, 2006.

P. Drineas and M. W. Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *JMLR*, 2005.

E. Even-Dar, S. Mannor, and Y. Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *JMLR*, 2006.

R. Foygel and N. Srebro. Concentration-based guarantees for low-rank matrix reconstruction. In *COLT*, pages 315–340, 2011.

V. Gabillon, M. Ghavamzadeh, A. Lazaric, and S. Bubeck. Multi-bandit best arm identification. In *NIPS*, 2011.

A. Gittens and M. Mahoney. Revisiting the nystrom method for improved large-scale machine learning. In *ICML*, pages 567–575, 2013.

F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19, 2015.

K. Jamieson, M. Malloy, R. Nowak, and S. Bubeck. lil'ucb: An optimal exploration algorithm for multi-armed bandits. *COLT*, 2014.

S. Kale, L. Reyzin, and R. E. Schapire. Non-stochastic bandit slate problems. In *NIPS*, 2010.

R. Keshavan, A. Montanari, and S. Oh. Matrix completion from noisy entries. In *Advances in Neural Information Processing Systems*, pages 952–960, 2009.

A. Krishnamurthy and A. Singh. Low-rank matrix and tensor completion via adaptive sampling. In *Advances in Neural Information Processing Systems*, pages 836–844, 2013.

S. Kumar, M. Mohri, and A. Talwalkar. Sampling methods for the nyström method. *JMLR*, 2012.

M. Laurent and A. Varvitsiotis. A new graph parameter related to bounded rank positive semidefinite matrix completions. *Mathematical Programming*, 145(1-2):291–325, 2014a.

M. Laurent and A. Varvitsiotis. Positive semidefinite matrix completion, universal rigidity and the strong arnold property. *Linear Algebra and its Applications*, 452:292–317, 2014b.

L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

R. Mazumder, T. Hastie, and R. Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, 11(Aug):2287–2322, 2010.

F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multi-armed bandits. In *ICML*. ACM, 2008.

M. Streeter and D. Golovin. An online algorithm for maximizing submodular functions. In *NIPS*, 2009.

M. Tan, I. W. Tsang, L. Wang, B. Vandereycken, and S. J. Pan. Riemannian pursuit for big matrix recovery. In *ICML*, pages 1539–1547, 2014.

B. Vandereycken. Low-rank matrix completion by riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013.

Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 2012.

Y. Yue and C. Guestrin. Linear submodular bandits and their application to diversified retrieval. In *NIPS*, pages 2483–2491, 2011.

Y. Yue, J. Broder, R. Kleinberg, and T. Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.