MATRIX COMPLETION: FUNDAMENTAL LIMITS AND EFFICIENT ALGORITHMS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Sewoong Oh

December 2010

Preface

Low-rank models provide low-dimensional representations capturing the important aspects of data naturally described in matrix form. Examples range from users' preferences on movies to similarities between pairs of items. Naturally, the low-rank representations serve as a tool for data compression and efficient computation. But more importantly, they are used in data analysis to learn hidden structures of the data, which is at the heart of machine learning and data mining. Applications include latent semantic analysis, factor analysis, and clustering high-dimensional data.

While singular value decomposition provides an efficient way to construct a low-rank model of a fully observed data matrix, finding a low-rank model becomes challenging when we only have a partial knowledge about the matrix. When we observe a subset of the entries, we show that the singular value decomposition is sub-optimal and can be significantly improved upon.

In this work, we investigate the possibility of learning a low-rank model from a partial observation of a data matrix. We develop a novel and efficient algorithm that finds a near-optimal solution. Further, we prove that the proposed algorithm achieves performance close to the fundamental limit under a number of noise scenarios. This provides a solution to many practical problems including collaborative filtering and positioning.

Acknowledgement

During my years at Stanford, I was fortunate to have Andrea Montanari as my advisor. I learned everything I know about research from Andrea. Since the beginning of my PhD, I've continuously benefited from long meetings with Andrea, who was always ready to write down equations with me and to listen to the numerous talk rehearsals. I am grateful for his patience and support. Further, I learned from Andrea the joy of research. Everytime we discussed new ideas, I could see his enthusiasm and excitement. I am truly thankful to Andrea for sharing his passion and for showing me that research is fun, and I am proud and honored to be his student.

I also enjoyed the advice and guidance from a number of other mentors, who will have a lasting influence on my research direction and my life. Tom Cover has been a constant source of inspiration from the beginning of my graduate study. I am really fortunate to have him as a teacher and a mentor. I would like to thank Balaji Prabhakar for his support and encouragement. His door was always open when I needed help. I have also been fortunate to work with Rüdiger Urbanke. We had stimulating and interesting discussions when I started working on the positioning problem. I greatly benefited from Rüdiger's deep insights and positive energy. I am grateful to Emmanuel Candès for sharing his insights on compressed sensing and matrix completion. His work inspired me to start working on the topic of this thesis. Stephen Boyd taught me the beauty of convex optimization and guided me through the world of linear algebra and matrices. Abbas El Gamal, Tsachy Weissman, and Amin Saberi were always happy to answer my questions and provide advice.

I have also been fortunate to work with very talented friends here at Stanford and also at EPFL. I had a wonderful time working with Fernando, Farshid, Satish, Jose, and Amin. I also appreciate occasional discussions with Yashodhan, Mohsen, Morteza, Adel, Arian, and Yaniv. Raghunandan deserves special mention, not only for the fruitful collaborations and discussions, but also for being a careful editor. I have also enjoyed being with a number of good friends, and no acknowledgement would be complete without them. Some of these friends are Jungil, Taehoon, Changhan, Junkyu, Bora, Wanki, Jehyung, Yul, Jongsoo, Kahye, Tesup, Younggeun and many others.

Most importantly, I would like to thank my beloved family. Thank you Kyung Eun for your love, support, sacrifice and patience. You have been the best friend I could ever hope for, and this thesis would not have been possible without your support.

Contents

Chapter 1

Introduction

A wide range of datasets are naturally organized in matrix form. Consider the example of a movie-rental service collecting its customers' ratings on movies. This dataset can be represented as a movie-ratings matrix where each customer is represented by a row and each movie is represented by a column. Each entry of this matrix represents the preference of the associated customer for a particular movie. Similarly, the frequencies of words used in a collection of documents can be represented as a term-document matrix, where each entry corresponds to the number of times the associated term appears in the indicated document.

A low-rank model is often used to capture the important aspects of such a data matrix thus providing a low-dimensional representation. Traditionally, low-rank models have been used for data compression and efficient computation. Low-rank representations require less resources to be stored compared to the full matrix. Moreover, traditional computational tasks, such as matrix multiplication and inversion, can be performed more efficiently when working with the reduced representation.

The data analysis method that approximates a data matrix by a low-rank model is referred to as principal component analysis (PCA). The singular vectors of the data matrix corresponding to the leading singular values are often called the principal components. PCA provides the means to understand the hidden structures in the data, which is at the heart of machine learning and data mining. Examples include latent semantic indexing and spectral clustering.

Latent semantic indexing [29] is a technique for analyzing the relationships between a collection of documents and the terms they contain. A low-rank approximation of the term-document matrix allows us to identify a set of concepts reflecting the hidden relationships. Each principal component represents a concept. An entry of a principal component corresponding to a specific document gives the relation between the document and the concept represented by the principal component. Using a small number of principal components, each document can be represented by a low dimensional vector corresponding to the entries in the principal components. In this 'concept space', semantically related documents are placed near one another. This can be used to see how related two documents are in this concept space and to search for documents conceptually related to a given query.

In unsupervised learning, low-rank model is used for partitioning a collection of objects so that similar objects are placed in the same cluster. Given a notion of similarity between all pairs of objects, we can define a similarity matrix. Each entry of this matrix represents the similarity between two objects associated with its row and its column. After an appropriate scaling of each row and each column, we can use a small number of principal components to represent each object by a low dimensional vector corresponding to the entries in the principal components. Similar objects are placed near one another in this low dimensional space [75]. Once we have this reduced representation, we can apply standard clustering techniques such as k-means algorithm [65].

If all the entries of a data matrix are available, computing a low-rank model of the data matrix is easy. We denote the singular value decomposition (SVD) of the data matrix M with

$$M = \sum_i \sigma_i u_i v_i^T \ ,$$

where σ_i's are the singular values and u_i and v_i are the left and right singular vectors corresponding to the ith singular value. A common convention is to order the singular values in descending order, i.e., $\sigma_1 \geq \sigma_2 \geq \cdots > 0$. Then, a low-rank approximation $L = \sum_{i=1}^{r} \sigma_i u_i v_i^T$ minimizes the sum of squares of the residual errors among all matrices that have the same predefined rank. In other words, L is the optimal solution to the problem:

$$\begin{array}{ll} \text{minimize} & \sum_{i,j} (M_{ij} - X_{ij})^2 \\ \text{subject to} & \operatorname{rank}(X) \leq r \ . \end{array}$$

In many practical applications, we do not have all the entries of the data matrix available. Entries might be missing due to failure in the data acquisition process, or it might be too costly to measure all the entries. Consider an example of a partially revealed rank-1 matrix below.

$$\begin{bmatrix} ? & ? & -1 & ? & ? \\ ? & 1 & ? & ? & ? \\ 1 & 1 & -1 & 1 & -1 \\ 1 & ? & ? & ? & -1 \\ ? & ? & -1 & ? & ? \end{bmatrix}$$

A naive approach based on SVD is to first fill in the missing entries with the average value and then compute the SVD of the resulting full matrix. Then we can use the first singular value and the corresponding left and right singular vectors to compute the rank-1 approximation:

$$\begin{bmatrix} 0.24 & 0.20 & -0.24 & 0.17 & -0.24 \\ 0.20 & 0.16 & -0.20 & 0.14 & -0.20 \\ 1.09 & 0.89 & -1.09 & 0.75 & -1.09 \\ 0.48 & 0.39 & -0.48 & 0.33 & -0.48 \\ 0.24 & 0.20 & -0.24 & 0.17 & -0.24 \end{bmatrix}$$

This naive approach is sub-optimal and can be significantly improved upon. Since the original matrix has rank one, all the rows must span a 1-dimensional subspace. It is not difficult to see that all the rows with missing entries should be the same as the third row. Further, this defines a unique rank-1 model achieving zero error:

$$\begin{bmatrix} 1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

Matrix completion concerns this problem of predicting the missing entries in a partially observed data matrix by learning a low-rank model. The above example of completing a rank-1 matrix seems trivial, but the problem changes dramatically when we have matrices of higher ranks.

Many applications are captured within the framework of matrix completion. This could be used, for example, by a movie-rental service to provide personalized recommendations based on the past ratings provided by its customers. A movie-ratings matrix can be modelled as an approximately low-rank matrix: customers who agreed in the past tend to agree in the future and only a few factors determine a customer's taste in movies. Solving matrix completion provides predictions on the unobserved ratings, which in turn can be used to make customized recommendations. A well-known example of this *collaborative filtering* problem is the Netflix prize [59, 3].

Similarly, in a network of wireless devices, we could apply matrix completion schemes to find positions of the devices. If we know distances between all pairs of devices then we can reconstruct all the positions up to a rigid motion by applying a simple technique known as multidimensional scaling [26]. In practice, due to attenuation and interference, only measurements between close-by devices are available. This defines a partially revealed distance matrix, the rank of which is determined by the ambient dimension. To be precise, let each entry of a distance matrix be the squared Euclidean distance between two devices associated with the row and the column. Then, it is known that the rank of this distance matrix is at most five [93]. Hence, we can find the positions of the devices up to a rigid motion by first completing this distance matrix with a rank-5 matrix and then applying multidimensional scaling.

When it is costly to measure an entry of the data matrix, matrix completion schemes could be used to reduce the number of measurements significantly. One might be interested in monitoring temperature at hard-to-reach locations or traffic on all the links in a network [120]. Define a traffic matrix where each row represents each link and each column represents each time interval. Each entry represents the volume of traffic on a particular link at the associated time interval. Since monitoring all the links at all times is costly, we could instead measure a subset of the links at a time interval and measure a different subset of the links at the next time interval. Collecting all these partial traffic measurements over time, we can estimate the rest of the traffic matrix using matrix completion schemes.

Matrix completion schemes could also be used to predict missing data points possibly resulting from failures in the data acquisition process. Missing data points are common in applications with large datasets, such as microarray [111] and structure-from-motion [24]. Once all the missing values have been predicted, the complete dataset can be analyzed using standard statistical techniques. We are concerned with problems where most of the values are missing as well as problems where only a few values are missing. For instance, about 99% of the movie-ratings are missing in the Netflix prize dataset.

In this work, we study the problem of learning a low-rank model from a partially revealed matrix to predict the missing entries. We introduce an algorithmic solution to this matrix completion problem and analyze how well we can learn the low-rank model with this algorithm.

Chapter 2 introduces the novel and efficient algorithm for solving matrix completion problems. Assuming the samples are drawn uniformly at random and corrupted by noise, we provide strong performance guarantees. In the process, we obtain a generalization of a celebrated result by Friedman, Kahn, and Szemerédi [42], and Feige and Ofek [40] on the spectrum of sparse random graphs. The algorithm can easily deal with massive datasets with billions of entries.

The performance guarantees proved in Chapter 2 are quite general and hold for any noise. In Chapter 3 we consider the quality of our estimation and prove order-optimality under two important scenarios: the noiseless setting and the additive Gaussian noise setting. When we don't have any noise in the samples, if enough number of entries are sampled then we can correctly reconstruct a low-rank matrix. Candès and Tao [22] proved a lower bound on the necessary number of samples, below which no algorithm can guarantee correct reconstruction. With a small number of samples close to this lower bound, we show that our algorithm correctly recovers a low-rank matrix. In the case when we have i.i.d. Gaussian noise, Candès and Plan [19] and Negahban and Wainwright [74] proved lower bounds on the achievable estimation error. We show that our algorithm finds a near-optimal estimation with error close to these lower bounds. Further, we review related results in matrix completion and compare them to our performance guarantees. Then, we show numerical results on artificial and real datasets supporting the theoretical guarantees.

Chapter 4 deals with the problem of positioning wireless devices. We formulate this problem as a matrix completion problem, and discuss the main challenges in applying matrix completion schemes directly. As a first step to overcome these challenges, we focus on understanding and analyzing existing positioning algorithms and provide strong performance guarantees. Although this chapter is motivated by matrix completion, it is mostly self-contained.

Notations

Throughout this manuscript, we use the following notations. For a vector $x \in \mathbb{R}^n$, we use $\|x\| = (\sum_{i=1}^n x_i^2)^{1/2}$ for its Euclidean norm. For a matrix $X \in \mathbb{R}^{m\times n}$, we use X^T for the transpose of X, and $\mathrm{Tr}(X) \equiv \sum_i X_{ii}$ for the trace of X. $\|X\|_F$ denotes the Frobenius norm defined as $\|X\|_F \equiv \sqrt{\sum_{i,j} X_{ij}^2}$. We use $\|X\|_2$ to denote the operator norm (or the maximum singular value) defined as $\|X\|_2 \equiv \sup_{a\neq 0}\big\{\|Xa\|/\|a\|\big\}$. For a given positive integer n, we use $[n] = \{1, \dots, n\}$ to denote the set of first n integers. In the following, whenever we write that a property A holds *with high probability (w.h.p.)*, we mean that there exists a function $f(n)$ such that $\mathbb{P}(A) \geq 1 - f(n)$ and $f(n) \to 0$ as $n \to \infty$.

Chapter 2

Matrix Completion

In collaborative filtering, we want to predict the missing ratings of a partially revealed movie-ratings matrix in order to provide personalized recommendations [59]. In positioning, we want to find the missing distance measurements in a partially revealed distance matrix in order to find the positions of the wireless devices [97]. Both of these examples share a common fundamental challenge: How can we recover, or *complete*, a data matrix from a small number of revealed entries? Without further hypothesis on the structure of the data, we cannot hope to find the missing entries. A common hypothesis is that the matrix is well approximated by a low-rank matrix. Low-rank matrices are commonly used to model collaborative filtering datasets [59]. In the positioning problem, the rank of the data matrix is comparable with the ambient dimension which is three.

In Section 2.1, we start with a mathematical model for the problem of matrix completion which will be used throughout this chapter. We assume that the matrix to be reconstructed has low-rank and the entries are revealed uniformly at random. Some matrices are easier to *complete* than other matrices of the same rank, and 'incoherence' of a data matrix characterizes how difficult it is to complete the matrix from random samples. In Section 2.2, we provide a formal definition of this incoherence property which was introduced in [20]. Section 2.3 introduces an algorithm which efficiently finds provably near-optimal solutions to matrix completion problems. The proposed algorithm starts with 'trimming' the sampled data matrix to prevent overfitting and then computes the singular value decomposition to get an initial guess. Then, starting from this initial estimate, the algorithm iteratively performs gradient descent on a Cartesian product of two Grassmann manifolds. The role of the trimming step is further explained with an example in Section 2.4, and Section 2.5 discusses the geometry of the Grassmann manifold which will be useful in the proof of main results. The main result of this chapter is a near-optimal performance guarantee of the proposed algorithm and is presented in Section 2.6. The proof of this result is provided in Section 2.7. This chapter is based on joint work with Keshavan and Montanari [56, 57].

2.1 Mathematical model

This section provides a mathematical model and sets up notations that are going to be used throughout this chapter. We denote by M the unknown low-rank matrix that we want to reconstruct. In the context of collaborative filtering, the (i, j)th entry of M is the rating the user i would assign to the movie j. We assume that $M \in \mathbb{R}^{m\times n}$ has rank r which is much smaller than its dimensions m or n. Notice that we can always assume to have $m \geq n$, since, in the other case, we can instead consider the transpose of the matrix M. In the following, therefore, we define $\alpha \equiv m/n$ and assume $\alpha \geq 1$.

The singular value decomposition (SVD) of M can be expressed as

$$M = U\Sigma V^T\,,$$

where $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$ are orthonormal matrices corresponding to the left and right singular vectors. The singular values are collected in a diagonal matrix $\Sigma = \text{diag}(\Sigma_1, \dots, \Sigma_r)$, where Σ_i is the ith largest singular value of M , i.e., $\Sigma_1 \geq \Sigma_2 \geq \cdots \geq \Sigma_r > 0$.

To model the noise in the revealed entries, we assume that each entry of M is perturbed, thus producing an approximately low-rank matrix N. Let

$$N_{ij} = M_{ij} + Z_{ij}\,,$$

where the noise matrix Z will be assumed to be small in an appropriate sense. Notice that in a more practical case, the matrix to be reconstructed might not be exactly low-rank but only approximately low-rank. Then, the weaker spectral features can be modelled with Z and the same algorithm and our analysis applies to that case as well.

Out of the $m \times n$ entries of N, a subset $E \subseteq [m] \times [n]$ is revealed. Here and below $[k] = \{1, \dots, k\}$ denotes the set of first k integers. In the collaborative filtering dataset, E is the user/movie pairs for which a rating N_{ij} is available, and Z_{ij} models the noise in the sampled ratings. Let N^E denote the revealed matrix where all the missing entries are filled with 0's.

$$N^E_{ij} = \begin{cases} N_{ij} & \text{if } (i,j) \in E\,, \\ 0 & \text{otherwise.} \end{cases}$$

In the future, we use the terms 'observed', 'revealed', or 'sampled' matrix interchangeably to refer to this N^E. Notice that we can also write $N^E = M^E + Z^E$, where the superscript $(\cdot)^E$ of a matrix denotes that we set to zero all the missing entries (all the entries in the complement of E).

The set E is assumed to be uniformly random given its size $|E|$. From the random sample N^E, we want to find a low-rank estimation of M. The accuracy of our estimation depends on the structure of the original matrix M. We need the singular vectors U and V to be sufficiently "spread out" or "incoherent" in order to reconstruct M from a small number of samples. This notion is formalized by the *incoherence property* introduced by Candès and Recht [20], and defined in the following section.

2.2 Incoherence property

We need a simple characterization of M which conveys how difficult it is to recover the matrix from a small number of sampled entries. For instance, let e_i denote a standard basis vector which has 1 in the ith entry and 0's everywhere else. Let us take an extreme example where $M = e_i e_j^T$ and there is no noise, $Z = 0$. Then the original matrix M can be recovered exactly only if the (i, j)th entry is actually sampled, which happens with large probability only if we sample a significant fraction of the entries. Therefore, to complete a matrix from a small number of samples, we need the singular vectors to be sufficiently unstructured or incoherent in an appropriate sense.

Recall that Σ_i is the ith largest singular value of M, and u_i and v_i are the corresponding left and right singular vectors. We write $U = [u_1, u_2, \dots, u_r]$ and $V = [v_1, v_2, \dots, v_r]$. Then, a rank-$r$ matrix M has *incoherence property* with parameter μ if it satisfies the following properties:

A0. For all $i \in [m]$, $j \in [n]$, we have $\sum_{a\in[r]} U_{ia}^2 \le \mu r/m$, $\sum_{a\in[r]} V_{ja}^2 \le \mu r/n$.

A1. For all $i \in [m]$, $j \in [n]$, we have $|M_{ij}| \le \mu r^{1/2} \Sigma_1/\sqrt{mn}$.

We also refer to M as μ-incoherent. In particular, we say a family of matrices are *μ-incoherent* if the above conditions are uniformly satisfied. Further, with a slight abuse of terminology, we say a matrix or a family of matrices are *incoherent* if the conditions are satisfied with polylogarithmic μ, i.e., $\mu = O((\log n)^a)$.

This definition of incoherence was first introduced by Candès and Recht [20], and proved to be crucial in characterizing the accuracy of matrix completion [22]. The first assumption A0 coincides with the corresponding assumption in [20]. Further, [20] makes an assumption that $|\sum_{a\in[r]} U_{ia} V_{ja}| \le \mu r^{1/2}/\sqrt{mn}$. This is analogous to A1, although it does not coincide with it. The two versions of assumption A1 coincide in the case of equal singular values $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_r$. In the general case, they do not coincide but neither one implies the other. For instance, in the case the vectors $(U_{i1}, \dots, U_{ir})$ and $(V_{j1}, \dots, V_{jr})$ are collinear, our condition is weaker, and is implied by the assumption of [20].

For any M, we can always find a $\mu \in [1, m/\sqrt{r}]$ which satisfies the incoherence property. Let $U_\perp$ be an orthonormal basis spanning the subspace orthogonal to the span of U. Since $\|e_i^T U\|^2 \le \|e_i^T [U\ U_\perp]\|^2 = 1$, **A0** holds for any M with $\mu = m/r$. Since $\Sigma_1 \ge \max_{i,j} |M_{ij}|$, **A1** holds for any M with $\mu \le m/\sqrt{r}$. Hence for any M we can find a $\mu \le m/\sqrt{r}$ such that M is μ incoherent. Intuitively, μ is large if the the singular vectors of M are coherent with the standard basis. In the previous example where $M = e_i e_j^T$, μ is equal to m and $r = 1$.

A number of families of random matrices are incoherent. For instance, let $M = XY^T$ where $X \in \mathbb{R}^{m\times r}$ and $Y \in \mathbb{R}^{n\times r}$ are random matrices with i.i.d. bounded entries. Then, the incoherence condition is satisfied with $\mu = O(\min\{\sqrt{r}, \sqrt{\log n}\})$, with high probability. Also, if X and Y are drawn uniformly at random among all orthonormal matrices of dimensions $m \times r$ and $n \times r$, then $\mu = O(\log n)$ with high probability [20].

2.3 Algorithm

Consider an example where the noise is zero mean and each entry is sampled independently with probability $p = |E|/mn$. Then, given N^E, the following rescaling ensures that the sampled matrix has mean M.

$$\left(\frac{1}{p}\right) N_{ij}^E = \begin{cases} (1/p)(M_{ij} + Z_{ij}), & \text{with probability } p, \\ 0, & \text{otherwise.} \end{cases}$$

Indeed $\mathbb{E}[(1/p)N_{ij}^E] = M_{ij}$. Intuitively, the rescaling compensates for the smaller average size of the entries of N^E with respect to M. Achlioptas and McSherry [4] showed that this rescaled and sampled matrix has spectral features very close to M when enough entries are samples, namely $p \ge (8\log n)^4/n$. Based on the spectral properties of this matrix, a naive algorithm for estimating M from a random sampling N^E consists of the following projection operation.

Projection. Compute the singular value decomposition (SVD) of $(mn/|E|)N^E$ (with $\sigma_1 \ge \sigma_2 \ge \cdots \ge 0$)

$$\left(\frac{mn}{|E|}\right) N^E = \sum_{i=1}^{n} \sigma_i x_i y_i^T\ ,$$

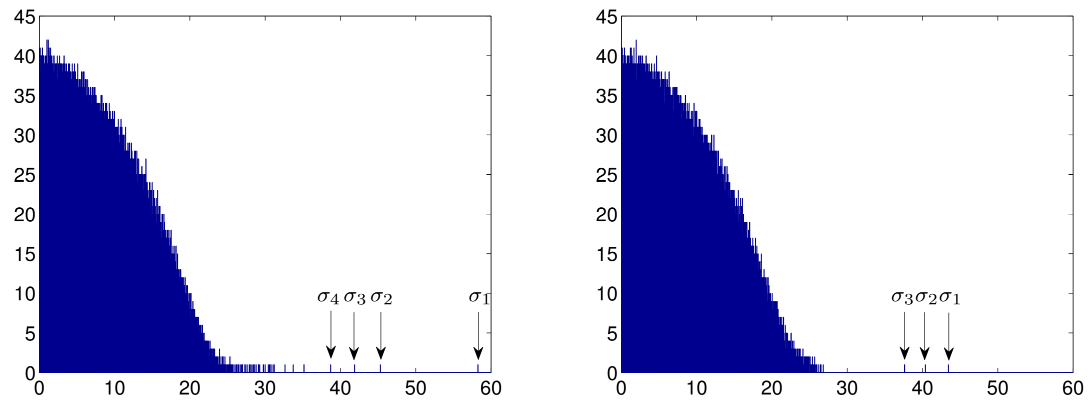


Figure 2.1: Histogram of the singular values of a partially revealed matrix M^E before trimming (left) and after trimming (right) for $10^4 \times 10^4$ random rank-3 matrix M with $|E|/mn = 0.003$ and $\Sigma = \mathrm{diag}(1, 1.1, 1.2)$. After trimming, the underlying rank-3 structure becomes clear. Here the number of revealed entries per row follows a heavy tail distribution with $\mathbb{P}\{N = k\} = \mathrm{const.}/k^3$.

and return the matrix

$$\mathcal{P}_r(N^E) = \sum_{i=1}^{r} \sigma_i x_i y_i^T ,$$

obtained by setting to 0 all but the r largest singular values. Notice that, apart from the rescaling factor $(mn/|E|)$, $\mathcal{P}_r(N^E)$ is the orthogonal projection of N^E onto the set of rank-r matrices.

It turns out that, if $|E| = \Theta(n)$, this algorithm performs very poorly. The reason is that the matrix N^E contains columns and rows with $\Theta(\log n/ \log\log n)$ non-zero (revealed) entries. The number of revealed entries in each row and each cloumn is distributed as i.i.d. binomial with bounded mean. It is known that among n i.i.d. binomial random variables with bounded mean the largest one is at least $C \log n/ \log\log n$ with high probability. We refer to Section 2.4 for the proof. These over-represented rows/columns alter the spectrum of N^E as will be discussed in Section 2.4. This motivates a preprocessing of the input data according to the following operation (hereafter the *degree* of a column or of a row is the number of its revealed entries).

Trimming. Set to zero all columns in N^E with degree larger that $2|E|/n$. Set to zero all rows with degree larger than $2|E|/m$. Let the matrix thus obtained be $\widetilde{N}^E$.

We can similarly define $\widetilde{M}^E$ from M and $\widetilde{Z}^E$ from Z, such that $\widetilde{N}^E = \widetilde{M}^E + \widetilde{Z}^E$. Figure 2.1 illustrates the role of trimming. We randomly generate a rank-3 matrix M. To emphasize the effect of trimming, we sample the entries from a heavy-tailed distribution. We let the degree of the ith row be a random variable X_i distributed as $\mathbb{P}\{X_i = k\} = C/k^3$ for $k \geq k_0$. The constants C and k_0 are chosen appropriately such that the probability sums to one and the average degree is 30. Then, given $\{X_i\}_{i\in[m]}$, the entries are sampled uniformly at random from each row. The histogram of the singular values of the resulting sampled matrix M^E is shown in the left figure. Most of the information about the original matrix M is captured in the singular values $\{\sigma_2(M^E), \sigma_3(M^E), \sigma_4(M^E)\}$ and the corresponding singular vectors, while the leading singular value $\sigma_1(M^E)$ is a 'spurious' one due to an over-represented row. After trimming (right figure), we can see a sharp separation between the leading three singular values revealing the structure of the original matrix M and the spurious ones. This effect becomes even more important when the number of revealed entries per row/column follows a heavy tail distribution, as is the case for real data.

After trimming and an appropriate rescaling, $(mn/|E|)\widetilde{N}^E$ has spectral properties close to the original matrix M. We can, therefore, apply the rank-r projection to this trimmed matrix to get a good estimation of the original matrix M. In the future, we refer to this procedure as trimming-plus-SVD. The operation of trimming seems counter-intuitive, because we are 'throwing out' valuable information. However, let us emphasize two facts here: (i) In the last step of our algorithm, to be explained in the following, the trimmed entries are actually incorporated in the cost function and hence the full information is exploited; (ii) Trimming is not the only way to treat over-represented rows/columns in N^E, and probably not the optimal one. Another popular way to balance out the rows and columns, for instance, is to rescale each row/column by the inverse of its degree. Similar rescaling techniques are often used in collaborative filtering [101, 59, 84, 103, 86]. We stick to trimming because we can prove it actually works. The role of trimming is crucial and is further explained in Section 2.4.

In terms of the above routines, our algorithm has the following structure. We call our algorithm OptSpace because we perform optimization over a lower dimensional space that can be described as the Cartesian product of two Grassmann manifolds. (We refer to Section 2.5 for background and references.)

```
OptSpace
Input: sampled matrix $N^E$, sample set $E$, target rank $r$
Output: estimated matrix $\widehat{M}$
1:  Trim $N^E$, and let $\widetilde{N}^E$ be the output;
2:  Project $\widetilde{N}^E$ to $X_0S_0Y_0^T = \mathcal{P}_r(\widetilde{N}^E)$;
3:  Clean residual errors by minimizing $F(X,Y)$
    using manifold optimization starting from $(X_0,Y_0)$.
```

The last step of the above algorithm allows to reduce (or eliminate) small discrepancies between $\mathcal{P}_r(\widetilde{N}^E)$ and M, and is described below. Note that the entries discarded in the trimming step are incorporated back into the algorithm in this cleaning step.

Cleaning. Various implementations are possible, but we found the following one particularly appealing. Given orthonormal matrices $X \in \mathbb{R}^{m\times r}$, $Y \in \mathbb{R}^{n\times r}$ with $X^TX = \mathbf{I}_r$ and $Y^TY = \mathbf{I}_r$ where $\mathbf{I}_r$ denotes the identity matrix, we define

$$\begin{aligned} F(X,Y) &\equiv \min_{S\in\mathbb{R}^{r\times r}} \mathcal{F}(X,Y,S)\,, \\ \mathcal{F}(X,Y,S) &\equiv \frac{1}{2}\sum_{(i,j)\in E}(N_{ij}-(XSY^T)_{ij})^2\,. \end{aligned} \tag{2.1}$$

In the cleaning step, we minimize $F(X,Y)$ locally starting from (X_0,Y_0), where X_0 and Y_0 are the orthogonal matrices resulting from the previous steps satisfying $X_0S_0Y_0^T = \mathcal{P}_r(\widetilde{N}^E)$.

Notice that $F(X,Y)$ is easy to evaluate since it is defined by minimizing the quadratic function $S \mapsto \mathcal{F}(X,Y,S)$ over the low-dimensional matrix S. Further, $F(X,Y)$ depends on X and Y only through their column spaces. In geometric terms, F is a function defined over the Cartesian product of two Grassmann manifolds. We refer to Section 2.5 for background and references on this manifold optimization step. Optimization over Grassmann manifolds is a well understood topic [36] and efficient algorithms (in particular Newton and conjugate gradient method) can be applied. To be definite, we propose using the gradient descent method with line search to minimize $F(X, Y)$. The gradient descent method is described in detail in Section 2.5.3.

The function $\mathcal{F}(X, Y, S)$ is a natural error function to minimize. Similar objective functions have been used for collaborative filtering in [101, 82, 84, 86]. The crucial differences in our algorithm and analysis are that, first, we introduce a novel trimming and projection step and, second, instead of minimizing $\mathcal{F}(X, Y, S)$ directly we minimize $F(X, Y)$ defined over the Cartesian product of two Grassmann manifolds. We refer to Section 3.4 for further comparisons.

The implementation proposed here implicitly assumes that the target rank r is known. In many practical applications, such as positioning [32, 97] or structure-from-motion [24, 109], the target rank is known in advance and is small. However, in some applications, such as collaborative filtering [82], the target rank is unknown. A very simple algorithm for estimating the rank of the matrix M from the revealed entries is introduced in Section 3.6.1. It is proved that the algorithm recovers the correct rank with high probability under the hypotheses of Theorem 2.6.2 in the noiseless case.

2.4 Role of trimming

The trimming step of OptSpace is somewhat counter-intuitive in that we seem to be wasting information. In this section, we want to clarify its role through a simple example. Assume, for the sake of simplicity, $M \in \mathbb{R}^{n \times n}$ to be a square all-ones matrix, the entries of M are sampled independently with probability p, and the samples are noiseless. Let $\epsilon = np$ denote the average number of samples per row/column. If $\epsilon \geq 15 \log n$, no row or column is over-represented with probability larger than $1 - 1/n^4$. Using the Chernoff bound, we get that each of the rows and the columns is over-represented with probability smaller than $(e/4)^{\epsilon}$. For $\epsilon \geq 15 \log n$ this is smaller than $1/(2n^5)$. Applying the union bound over all $2n$ rows and columns, the claim follows. Since we are interested in the regime where trimming is crucial, we assume ϵ to be $O(1)$.

The number of non-zero entries in a row is Binomial$(n, \epsilon/n)$ and is independent of other rows. There exists a constant C such that the row with the largest number of entries has more than $C \log n / \log\log n$ entries with probability larger than $1/n$. A simple proof of this claim is provided in [69] using the Poisson approximation. Let X_i's be independent Poisson random variables with mean ϵ. Also we let $p_k = \mathbb{P}(X_i \geq k)$. Then, $\mathbb{P}(\max_i X_i < k) = (1-p_k)^n \leq \exp\{-p_k n\}$. Note that $p_k \geq 1/(ek!)$. For $k = C \log n / \log\log n$ with small enough C and $\epsilon = O(1)$, it follows that $\exp\{-p_k n\} \leq 1/n$.

Let $i_{\max}$ be the index of this column, and consider the test vector $\underline{e}^{(i_{\max})}$ whose $i_{\max}$th entry is equal to one and all the others are equal to zero. By computing $\|M^E \underline{e}^{(i_{\max})}\|$, we conclude that the largest singular value of M^E, is at least $C\sqrt{\log n / \log\log n}$. In particular, this is very different from the largest singular value of $\mathbb{E}[M^E]$ which is ϵ. Approximating M with the projection $\mathcal{P}_r(M^E)$, therefore, leads to a large error.

Our first main result in Theorem 2.6.1 indeed confirms this expectation. In the process of showing that $\mathcal{P}_r(\widetilde{M}^E)$ is close to M, we require the leading singular values of $(1/p)\widetilde{M}^E$ to be close to the ones of M. This is only true if we assume trimming. Following arguments similar to the ones in the proof, it is indeed possible to show that the root mean squared error of the untrimmed estimation $\mathcal{P}_r(M^E)$ is increasing with n when ϵ is bounded.

$$\frac{1}{n}\left\|M - \mathcal{P}_r(M^E)\right\|_F \geq \frac{C'}{\epsilon}\Big(\frac{\log n}{\log\log n}\Big)^{1/2},$$

where $\|A\|_F = (\sum_{i,j} A_{ij}^2)^{1/2}$ denotes the Frobenius norm of a matrix. It is clear that this phenomenon is indeed general, as illustrated by Figure 2.1. Also, the phenomenon is more severe in real datasets where the number of revealed entries per row/column follows a heavy tail distribution. For instance, in the case of the Netflix prize dataset [3], the maxium degree of a movie is more than 200,000 while the average number of samples per movie is about 5,627 [9]. These over-represented movies alter the spectrum of the movie-ratings matrix significantly.

To summarize, Theorem 2.6.1 simply does not hold without trimming, or a similar procedure to normalize rows/columns of M^E. Trimming allows to overcome the above phenomenon by setting to 0 over-represented rows/columns. On the other hand, this makes the analysis technically non-trivial. Indeed, while M^E is a matrix with independent (but not identically distributed) entries, this is not the case for $\widetilde{M}^E$. As a consequence we cannot rely on standard concentration-of-measure results.

2.5 Manifold optimization

The function $F(X, Y)$ defined in (2.1) and to be minimized in the last part of OptSpace can naturally be viewed as defined on the Grassmann manifold. Here we recall from [36] a few important facts on the geometry of the Grassmann manifold and related optimization algorithms in Section 2.5.1. Then, we give the explicit formulae for the gradient in Section 2.5.2. In Section 2.5.3, we use these formulae to precisely describe the manifold optimization step.

2.5.1 Grassmann manifold

The orthogonal group $\mathsf{O}(r)$ consists of r-by-r orthogonal matrices and the orthonormal group $\mathsf{O}(n, r)$ consists of n-by-r orthonormal matrices. In formulae, $\mathsf{O}(r) = \{Q \in \mathbb{R}^{r\times r} : QQ^T = Q^TQ = \mathbf{I}_r\}$ and $\mathsf{O}(n, r) = \{A \in \mathbb{R}^{n\times r} : A^TA = \mathbf{I}_r\}$. The Grassmann manifold $\mathsf{G}(n, r)$ is defined as the set of all r-dimensional subspaces of $\mathbb{R}^n$. In other words, a point on the manifold is the equivalence class of an n-by-r orthonormal matrix $A \in \mathsf{O}(n, r)$

$$[A] = \{AQ : Q \in \mathsf{O}(r)\} .$$

Two n-by-r orthonormal matrices are equivalent if their columns span the same subspaces. Hence, a point in the Grassmann manifold is a specific subset of the orthogonal matrices, and the Grassmann manifold is the collection of all these subsets. Therefore the Grassmann manifold is also defined as the quotient $\mathsf{G}(n, r) =$

$\mathsf{O}(n)/(\mathsf{O}(r)\times\mathsf{O}(n-r))$, where the quotient $\mathsf{O}(n)/\mathsf{O}(n-r)$ is the set of all n-by-r orthonormal matrices. To represent a point in $\mathsf{G}(n,r)$, we will use an explicit representative A to represent the equivalence class $[A]$.

It is easy to see that $F(X,Y)$ depends on the matrices X and Y only through their equivalence classes $[X]$ and $[Y]$. We will therefore interpret it as a function defined on the manifold $\mathsf{M}(m,n)\equiv\mathsf{G}(m,r)\times\mathsf{G}(n,r)$:

$$\begin{aligned} F:\mathsf{M}(m,n) &\rightarrow \mathbb{R}\,,\\ ([X],[Y]) &\mapsto F(X,Y)\,. \end{aligned}$$

In the following, a point in this manifold will be represented as a pair $\mathbf{x}=(X,Y)$, with X an m-by-r orthogonal matrix and Y an n-by-r orthogonal matrix. Boldface symbols will be reserved for elements of $\mathsf{M}(m,n)$ or of its tangent space, and we shall use $\mathbf{u}=(U,V)$ for the point corresponding to the matrix $M=U\Sigma V^T$ to be reconstructed.

Given $\mathbf{x}=(X,Y)\in\mathsf{M}(m,n)$, the tangent space at $\mathbf{x}$ is denoted by $\mathsf{T}_{\mathbf{x}}$ and can be identified with the vector space of matrix pairs $\mathbf{w}=(W,Z)$, $W\in\mathbb{R}^{m\times r}$, $Z\in\mathbb{R}^{n\times r}$ such that $W^TX=Z^TY=0$. The 'canonical' Riemann metric on the Grassmann manifold corresponds to the usual scalar product $\langle W,W'\rangle\equiv\mathrm{Tr}(W^TW')$. The induced scalar product on $\mathsf{T}_{\mathbf{x}}$ and $\mathsf{M}(m,n)$ between $\mathbf{w}=(W,Z)$ and $\mathbf{w}'=(W',Z')$ is $\langle\mathbf{w},\mathbf{w}'\rangle=\langle W,W'\rangle+\langle Z,Z'\rangle$.

This metric induces a canonical notion of distance on $\mathsf{M}(m,n)$ which we denote by $d(\mathbf{x}_1,\mathbf{x}_2)$ (geodesic or arc-length distance). If $\mathbf{x}_1=(X_1,Y_1)$ and $\mathbf{x}_2=(X_2,Y_2)$ then

$$d(\mathbf{x}_1,\mathbf{x}_2)\equiv\sqrt{d(X_1,X_2)^2+d(Y_1,Y_2)^2}$$

where the arc-length distances $d(X_1,X_2)$ and $d(Y_1,Y_2)$ on the Grassmann manifold can be defined explicitly as follows. Let $\cos\theta=(\cos\theta_1,\ldots,\cos\theta_r)$ for $\theta_i\in[-\pi/2,\pi/2]$ be the singular values of $X_1^TX_2$. Then for $\theta=(\theta_1,\ldots,\theta_r)$

$$d(X_1,X_2)=\|\theta\|_2 \qquad \text{(geodesic or arc-length distance).}$$

The θ_i's are called the 'principal angles' between the subspaces spanned by the columns of X_1 and X_2. It is useful to introduce two equivalent notions of distance:

$$\begin{aligned} d_{\rm c}(X_1,X_2) &= \min_{Q_1,Q_2\in \mathsf{O}(r)} \|X_1Q_1 - X_2Q_2\|_F && \text{(chordal distance)},\\ d_{\rm p}(X_1,X_2) &= \frac{1}{\sqrt{2}}\|X_1X_1^T - X_2X_2^T\|_F && \text{(projection distance)}. \end{aligned}$$

Notice that $d(X_1,X_2)$, $d_{\rm c}(X_1,X_2)$ and $d_{\rm p}(X_1,X_2)$ do not depend on the specific representatives X_1 and X_2, but only on the equivalence classes $[X_1]$ and $[X_2]$. Distances on $\mathsf{M}(m,n)$ are defined through Pythagorean theorem, e.g. $d_{\rm c}(\mathbf{x}_1,\mathbf{x}_2) = \sqrt{d_{\rm c}(X_1,X_2)^2 + d_{\rm c}(Y_1,Y_2)^2}$.

Lemma 2.5.1. *The geodesic, chordal and projection distance are equivalent, namely*

$$\frac{1}{\pi}d(X_1,X_2) \le \frac{1}{\sqrt{2}}\,d_{\rm c}(X_1,X_2) \le d_{\rm p}(X_1,X_2) \le d_{\rm c}(X_1,X_2) \le d(X_1,X_2)\,.$$

Proof. Let $A(\mathrm{diag}(\cos\theta))B$ be the singular value decomposition of $X_1^TX_2$. Then, $\theta = (\theta_1,\dots,\theta_r)$ for $\theta_i\in[-\pi/2,\pi/2]$ are the principal angles between the planes spanned by the columns of X_1 and X_2. It is known that $d_{\rm c}(X_1,X_2) = \|2\sin(\theta/2)\|$ and $d_{\rm p}(X_1,X_2) = \|\sin\theta\|$. The first identity follows from

$$\begin{aligned} d_{\rm c}(X_1,X_2)^2 &= 2r - 2\max_{Q_1,Q_2\in\mathsf{O}(r)} \mathrm{Tr}(Q_1^TX_1^TX_2Q_2)\\ &= 2\sum_{i\in[r]}(1-\cos\theta_i) = \|2\sin(\theta/2)\|^2\,. \end{aligned}$$

The second identity follows from

$$\begin{aligned} d_{\rm p}(X_1,X_2)^2 &= r - \mathrm{Tr}(X_1X_1^TX_2X_2^T)\\ &= \sum_{i\in[r]}\big(1-(\cos\theta_i)^2\big) = \|\sin\theta\|^2\,. \end{aligned}$$

The thesis follows from the elementary inequalities

$$\frac{1}{\pi}z \le \sqrt{2}\,\sin(z/2) \le \sin z \le 2\sin(z/2) \le z\,,$$

valid for $z \in [0, \pi/2]$. □

An important remark is that geodesics with respect to the canonical Riemann metric admit an explicit and efficiently computable form. Given $\mathbf{u} \in \mathsf{M}(m,n)$, $\mathbf{w} \in \mathsf{T}_\mathbf{u}$ the corresponding geodesic is a curve $t \mapsto \mathbf{x}(t)$, with $\mathbf{x}(t) = \mathbf{u} + \mathbf{w}t + O(t^2)$ which minimizes the arc-length. If $\mathbf{u} = (U, V)$ and $\mathbf{w} = (W, Z)$ then $\mathbf{x}(t) = (X(t), Y(t))$ where $X(t)$ can be expressed in terms of the singular value decomposition $W = L\Theta R^T$ [36]:

$$X(t) = UR\cos(\Theta t)R^T + L\sin(\Theta t)R^T\,, \tag{2.2}$$

which can be evaluated in time of order $O(nr)$. Note that $X(t)$ satisfy $X(0) = U$, $\dot{X}(0) = W$, and $X(t)^T X(t) = \mathbf{I}_r$ where $\mathbf{I}_r$ denotes the identity matrix. An analogous expression holds for $Y(t)$.

2.5.2 Gradient

In the following, we give an explicit expression for the gradient of the objective function, which in turn will be used in the manifold optimization step. The gradient of the objective function $F(\cdot)$ at $\mathbf{x}$ is the pair of matrices $\operatorname{grad} F(\mathbf{x}) \in \mathsf{T}_\mathbf{x}$ such that, for any smooth curve $t \mapsto \mathbf{x}(t) \in \mathsf{M}(m,n)$ with $\mathbf{x}(t) = \mathbf{x} + \mathbf{w}\,t + O(t^2)$, one has

$$F(\mathbf{x}(t)) = F(\mathbf{x}) + \langle \operatorname{grad} F(\mathbf{x}), \mathbf{w}\rangle\, t + O(t^2)\,.$$

Let $\mathbf{I}$ denote the identity matrix. Then, the two components of the gradient of $F(\mathbf{x})$ on the Grassmann manifold are $\operatorname{grad} F(\mathbf{x})_X = (\mathbf{I} - XX^T)\nabla_X F(\mathbf{x})$ and $\operatorname{grad} F(\mathbf{x})_Y = (\mathbf{I} - YY^T)\nabla_Y F(\mathbf{x})$ [36], where

$$\big(\nabla_X F(\mathbf{x})\big)_{ij} = \frac{\partial F(\mathbf{x})}{\partial X_{ij}}\ ,\ \big(\nabla_Y F(\mathbf{x})\big)_{ij} = \frac{\partial F(\mathbf{x})}{\partial Y_{ij}}.$$

In order to write an explicit representation of the gradient, it is convenient to introduce the projector operator for sampling

$$\mathcal{P}_E(M)_{ij} = \begin{cases} M_{ij} & \text{if } (i,j) \in E, \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

The two components of the gradient are then

$$\begin{aligned} \operatorname{grad} F(\mathbf{x})_X &= \big(\mathbf{I} - XX^T\big)\mathcal{P}_E(XSY^T - N)YS^T\,, \\ \operatorname{grad} F(\mathbf{x})_Y &= \big(\mathbf{I} - YY^T\big)\mathcal{P}_E(XSY^T - N)^T XS\,, \end{aligned}$$

where S is the minimizer in (2.1). Since, $X^T\mathcal{P}_E(XSY^T - N)Y = 0$ for the minimizer S, the above formulae simplify to

$$\operatorname{grad} F(\mathbf{x})_X = \mathcal{P}_E(XSY^T - N)YS^T\,, \tag{2.4}$$
$$\operatorname{grad} F(\mathbf{x})_Y = \mathcal{P}_E(XSY^T - N)^T XS\,. \tag{2.5}$$

2.5.3 Gradient descent

Now we have all the tools necessary to fully specify the gradient descent method with line search. Let the SVD of the output from the projection step be $\mathcal{P}_r(\widetilde{N}^E) = X_0S_0Y_0^T$. In the manifold optimization step, we take as input the left and right factors denoted as $\mathbf{x}_0 = (X_0, Y_0)$, and minimize a regularized cost function

$$\begin{aligned} \widetilde{F}(X,Y) &= F(X,Y) + \rho\, G(X,Y) \\ &\equiv F(X,Y) + \rho \sum_{i=1}^{m} g\left(\frac{m\|X^{(i)}\|^2}{3\mu r}\right) + \rho \sum_{j=1}^{n} g\left(\frac{n\|Y^{(j)}\|^2}{3\mu r}\right)\,, \end{aligned} \tag{2.6}$$

where $X^{(i)}$ denotes the i-th column of X^T, and $Y^{(j)}$ the j-th column of Y^T. The role of the regularization is to force $\mathbf{x}$ to remain incoherent during the execution of the algorithm. To force (X, Y) to satisfy the incoherence property **A0** with 3μ, we define

$$g(z) = \begin{cases} 0 & \text{if } z < 1, \\ e^{(z-1)^2} - 1 & \text{if } z \geq 1. \end{cases}$$

Various choices of the regularization function would work as well, but we find this one particularly simple. Furthermore, the algorithm is quite insensitive to the regularization coefficient ρ, and various choices work well in practice. The analysis of the algorithm gives a whole range of correct values of ρ, and we can choose $\rho = \Theta(|E|^2)$. Let us stress that the regularization term is mainly introduced for our proof technique to work (and a broad family of functions g would work as well). In numerical experiments we did not find any performance loss in setting $\rho = 0$. Notice that $G(X, Y)$ is again naturally defined on the Grassmann manifold, i.e. $G(X, Y) = G(XQ, YQ')$ for any $Q, Q' \in \mathsf{O}(r)$.

```
Manifold Optimization
Input: sampled matrix $N^E$, sample set $E$, initial factors $\mathbf{x}_0$, $k_{\max}$, $\gamma$
Output: estimated matrix $\widehat{M}$
1:  For $k = 0, 1, \dots, k_{\max}$ do:
2:     Compute $S_k = \arg\min_S \mathcal{F}(\mathbf{x}_k, S)$;
3:     Compute $\mathbf{w}_k = \operatorname{grad} \widetilde{F}(\mathbf{x}_k)$;
4:     Let $t \mapsto \mathbf{x}_k(t)$ be the geodesic with $\mathbf{x}_k(t) = \mathbf{x}_k + \mathbf{w}_k t + O(t^2)$
       defined as in (2.2);
5:     Set $t = t_0$;
6:     While $\widetilde{F}(\mathbf{x}_k(t)) > \widetilde{F}(\mathbf{x}_k) - (1/2)t\langle \mathbf{w}_k, \mathbf{w}_k \rangle$ or $d(\mathbf{x}_k(t), \mathbf{x}_0) > \gamma$ do:
7:        $t \leftarrow t/2$;
8:     Set $\mathbf{x}_{k+1} = \mathbf{x}_k(t)$;
9:  End For;
10: Output $\widehat{M} = X_k S_k Y_k^T$, where $S_k$ is the minimizer.
```

Before passing to the main results, it is worth discussing a few important points concerning the gradient descent algorithm.

(*i*) In the above, γ must be set in such a way that $d(\mathbf{u}, \mathbf{x}_0) \leq \gamma$ and a simple value

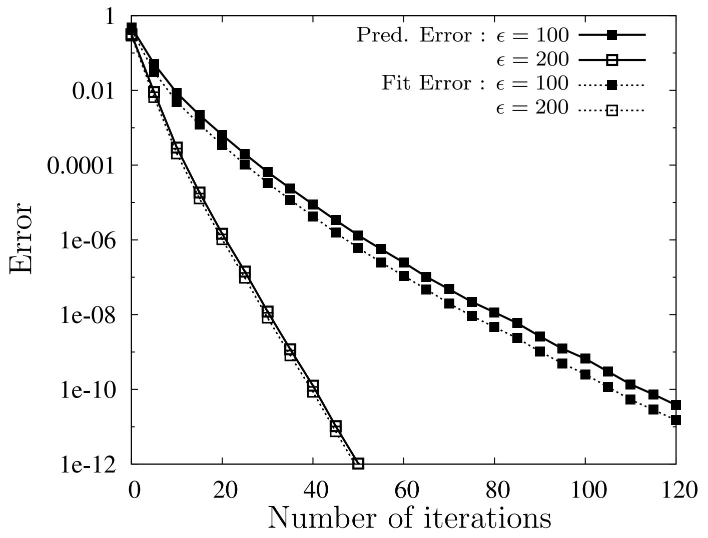


Figure 2.2: Prediction and fit errors as functions of the number of line minimizations for rank-10 random matrices of dimension 1000 $\times$ 1000. The average number of samples per row/column is ϵ for two values of ϵ: 100 and 200.

for γ is proposed in Section 2.7.3. The appropriate choice of γ might seem to pose a difficulty. In reality, this parameter is introduced only to simplify the proof. We will see that the constraint $d(\mathbf{x}_k(t), \mathbf{x}_0) \leq \gamma$ is, with high probability, never saturated.

(ii) Similarly, there is no need to know the actual value of μ in the regularization term. One can start with $\mu = 1$ and then repeat the optimization doubling it at each step. On the other hand, an algorithm for estimating the incoherence of an unknown matrix by sampling its columns is proposed in [70]. A similar approach can be extended to the case when we sample entries instead of columns.

(iii) The Hessian of $\widetilde{F}$ can be computed explicitly as well. This opens the way to quadratically convergent minimization algorithms (e.g. the Newton method). However, due to the increased complexity, such algorithms are not practical for large scale problems.

Figure 2.2 illustrates the rate of convergence of the manifold optimization step

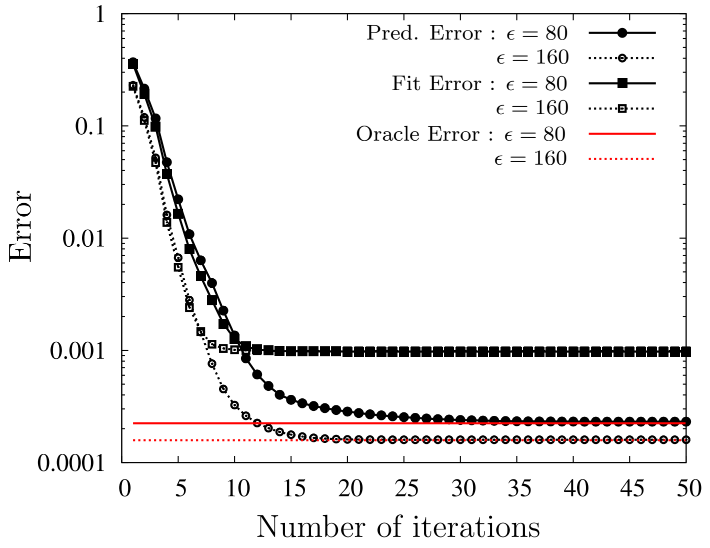


Figure 2.3: Prediction and fit errors as functions of the number of line minimizations for random rank-2 600×600 matrices. The average number of samples per row/column is ϵ: 80 and 160. The standard deviation of the i.i.d. Gaussian noise is 0.001. Prediction error of an oracle estimator (3.8) is also shown for comparison.

when there is no noise in the samples. We present the results of a numerical experiment for uniformly random matrices M. The fit error, $\|\mathcal{P}_E(\widehat{M}-M)\|_F/\sqrt{|E|}$ and the prediction error $\|\widehat{M}-M\|_F/n$, are plotted with respect to the number of iterations of the Manifold Optimization step and averaged over 10 instances. We can see that the prediction error decays exponentially with the number of iterations. Also, the prediction error is very close to the fit error, thus lending support to the validity of using the fit error as the stopping criterion.

Figure 2.3 illustrates the rate of convergence of the manifold optimization step in the presence of i.i.d. Gaussian noise. M is generated as in the previous example. A small Gaussian noise was used in order to trace the RMSE evolution over many iterations: $Z_{ij} \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = 0.001$. Each point in the figure is averaged over 20 random instances. The prediction error converges to that of an oracle estimator described in Section 3.2. The fit error is defined as $\|\mathcal{P}_E(\widehat{M}-N)\|_F/\sqrt{|E|}$, and can be easily evaluated since $N^E = \mathcal{P}_E(N)$ is always available at the estimator. The fit error decays exponentially with the number iterations until it reaches the standard deviation of the noise which is 0.001.

For simplicity, consider the case when all the entries are revealed. Since $N = M+Z$ and $\text{rank}(M) = r$,

$$\begin{aligned}\min_{A,\text{rank}(A)=r} \|N-A\|_F^2 &\geq \min_{A,\text{rank}(A)=2r} \|Z-A\|_F^2 \\ &\geq \|Z\|_F^2 - 2r\|Z\|_2^2 \,.\end{aligned}$$

From Theorem 2.6.3 we know that $\|Z\|_2^2 \leq C\sigma^2 n$ with high probability. Also, since $\mathbb{E}[\|Z\|_F^2] = \sigma^2 n^2$, we can apply the Chernoff bound to get $\|Z\|_F^2 \geq (1-\delta)\sigma^2 n^2$ with high probability for any positive $\delta > 0$. Then,

$$\min_{A,\text{rank}(A)=r} \frac{1}{n}\|N-A\|_F \geq \sigma\sqrt{1-\delta-\frac{Cr}{n}} \,.$$

For $r \ll n$, the fit error of the best rank-r estimation is lower bounded by a quantity close to $\sigma = 0.001$.

2.6 Main results

The two main results of this chapter are performance guarantees for the simple trimming-plus-SVD approach and the OptSpace algorithm. Since we are interested in very large datasets, we shall strive to prove performance guarantees that are asymptotically optimal for large m and n. However, our main results are completely non-asymptotic and provide concrete bounds for any m and n.

Trimming and rank-r projection can be done efficiently when we have N^E which is sparse. The following theorem establishes a bound on the quality of the estimation using this simple procedure.

Theorem 2.6.1. *Let M be a rank-r matrix of dimension $n\alpha \times n$ that satisfies $|M_{i,j}| \leq M_{\max}$ for all $i \in [m]$ and $j \in [n]$. Assume that the set of revealed entries $E \subseteq [n\alpha]\times[n]$ is uniformly random given its size $|E|$ and the sampled matrix is $N^E = M^E + Z^E$.*

Then there exists a numerical constant C such that

$$\frac{1}{n\sqrt{\alpha}}\Big\|M - \mathcal{P}_r(\widetilde{N}^E)\Big\|_{\rm F} \le C\, M_{\max}\left(\frac{n\, r\, \alpha^{3/2}}{|E|}\right)^{1/2} + 2\sqrt{2}\,\frac{n\sqrt{r\,\alpha}}{|E|}\big\|\widetilde{Z}^E\big\|_2\,, \qquad (2.7)$$

with probability larger than $1-1/n^3$.

Recall that $\|\cdot\|_F$ denotes the Frobenius norm and $\|\cdot\|_2$ denotes the operator norm or the largest singular value of a matrix. The architecture of the proof is described in Section 2.7.2. The factor $1/(n\sqrt{\alpha})$ corresponds to the usual normalization by the number of entries in the summation. Note that this bound does not depend on the incoherence of M.

Projection onto rank-r matrices through SVD is a pretty standard tool, and is used as the first analysis method for many practical problems. At a high-level, projection onto rank-r matrices can be interpreted as 'treat missing entries as zeros'. This theorem shows that this approach is reasonably robust if the number of observed entries is as large as the number of degrees of freedom (which is $(m+n)r-r^2$) times a large constant. The error bound in (2.7) is the sum of two contributions: the first term can be interpreted as an undersampling effect (error induced by missing entries) and the second as a noise effect. Let us stress that trimming is crucial for achieving this guarantee when the sample size is comparable with the degrees of freedom.

Notice that the top r singular values and singular vectors of the sparse matrix $\widetilde{N}^E$ can be computed efficiently using subspace iteration [11]. Each iteration requires $O(|E|r)$ operations. The subspace iteration converges exponentially when there is a gap between the rth singular value and the the $(r+1)$th one.

The second main result of this chapter provides a performance guarantee for OptSpace. By adding the 'cleaning' step, the performance improves systematically over the simple trimming-plus-SVD approach. The following theorem establishes that the 'cleaning' step eliminates all the effects from missing entries for large enough $|E|$. This theorem is order optimal in a number of important circumstances including the noiseless case and i.i.d. Gaussian noise case, and further comparisons are provided in Chapter 3. For the proof of this theorem we refer to Section 2.7.3.

Theorem 2.6.2. *Let M be a rank-r matrix of dimension $n\alpha \times n$ satisfying the incoherence property with parameter μ. Define Σ_k to be the k-th largest singular value of M and let $\kappa \equiv \Sigma_{\max}/\Sigma_{\min}$ where $\Sigma_{\max} \geq \Sigma_1 \geq \cdots \geq \Sigma_r \geq \Sigma_{\min}$. Assume that the set of revealed entries $E \subseteq [n\alpha] \times [n]$ is uniformly random given its size $|E|$. Let $\widehat{M}$ be the output of* OptSpace *on input $N^E = M^E + Z^E$. Then, there exists numerical constants C_1 and C_2 such that, if*

$$|E| \geq C_1\, n\, \mu\, r\, \alpha\, \kappa^2 \max\Big\{\log n\,,\, \mu r \sqrt{\alpha}\kappa^4\Big\}\,,$$

then, with probability larger than $1 - 1/n^3$,

$$\frac{1}{n\sqrt{\alpha}}\big\|M - \widehat{M}\big\|_F \leq C_2 \frac{n\,\kappa^2\,\sqrt{r\alpha}}{|E|}\|Z^E\|_2\,, \tag{2.8}$$

provided that the right-hand side is smaller than $\Sigma_{\min}/(n\sqrt{\alpha})$.

The above guarantee only holds 'up to numerical constants' independent of the matrix dimension or the incoherence parameter. One might wonder how good OptSpace is in practice. While a detailed study is presented in Section 3.5, in Figure 2.4, we present the results of a numerical experiment with randomly generated matrices. We generate $n \times r$ random matrices U and V with i.i.d. Gaussian entries distributed as $\mathcal{N}(0,1)$, and let $M = UV^T$ be the rank-r random matrix to be reconstructed. We plot the empirical *reconstruction rate* of OptSpace as a function of the average number of revealed entries per row. Here, the sampled entries are noiseless and we declare a matrix to be reconstructed if the relative error $\|M - \widehat{M}\|_F/\|M\|_F \leq 10^{-4}$. The *reconstruction rate* is the fraction of instances for which the matrix was reconstructed. For comparison, we also plot the fundamental limit obtained using the algorithm from [98] based on rigidity theory.

In Figure 2.5, the average root mean squared error RMSE $= \|M - \widehat{M}\|_F/\sqrt{mn}$ is shown for different sample size $|E|$ and various number of line minimization in the 'cleaning' step. The noise is i.i.d. Gaussian with $Z_{ij} \sim \mathcal{N}(0,1)$. We also plot an information theoretic lower bound described in Section 3.2. After 10 iterations, the RMSE achieved by OptSpace becomes indistinguishable from that of an oracle

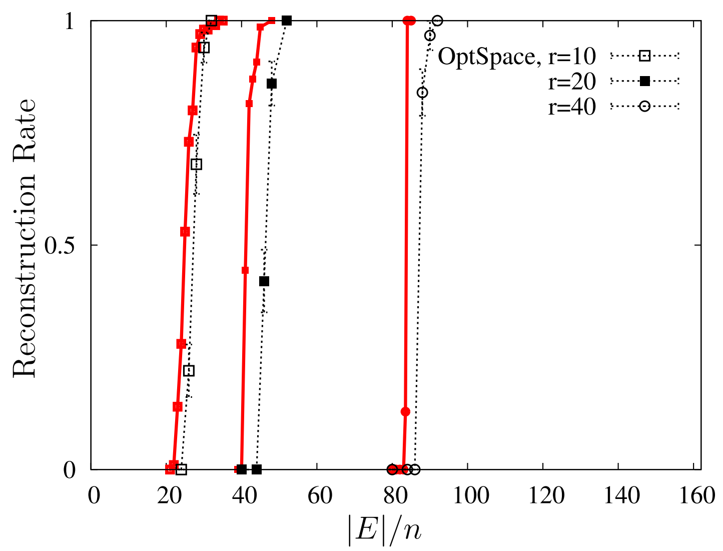


Figure 2.4: Reconstruction rates of OptSpace on 500×500 matrices. The solid curves are the upper bounds on the reconstruction rate for different ranks proved in [98].

estimator described in Section 3.2 for most values of $|E|$.

So far, we did not make any assumption about the noise matrix Z. Our results are completely general and provide concrete bounds for any Z. However, in order to make sense of the above results, it is convenient to consider a couple of simple models for the noise matrix. For each of these models, we provide a bound on $\|\widetilde{Z}^E\|_2$, which determines the accuracy of our estimation.

Independent entries model. We assume that Z_{ij}'s are i.i.d. random variables, with zero mean $\mathbb{E}\{Z_{ij}\} = 0$ and sub-Gaussian tails. The latter means that

$$\mathbb{P}\{|Z_{ij}| \geq z\} \leq 2\, e^{-\frac{z^2}{2\sigma^2}} \,,$$

for some constant σ^2 uniformly bounded in n.

Worst case model. In this model Z is arbitrary, but we have a uniform bound on the size of its entries: $|Z_{ij}| \leq Z_{\max}$.

The error bounds are determined by the operator norm $\|\widetilde{Z}^E\|_2$ and $\|Z^E\|_2$. The

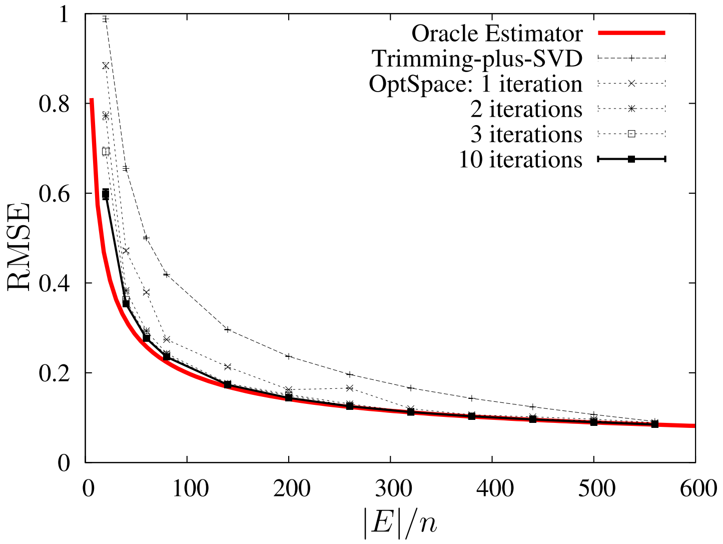


Figure 2.5: RMSE achieved by OptSpace in reconstructing a 600×600 rank-2 matrix using $|E|$ sampled entries with Gaussian noise for different number of line minimizations. RMSE achieved by an oracle estimator (3.8) is also shown for comparison.

next theorem provides a bound on these quantities under the two noise models.

Theorem 2.6.3. *If Z is a random matrix drawn according to the independent entries model, then for $|E| \geq n$ there is a constant C such that,*

$$\|\widetilde{Z}^E\|_2 \leq C\sigma \left(\frac{|E|\log n}{n}\right)^{1/2}, \tag{2.9}$$

with probability at least $1 - 1/n^4$. Further, if $|E| \geq n\alpha \log n$ then

$$\|\widetilde{Z}^E\|_2 \leq C\sigma \left(\frac{|E|}{n}\right)^{1/2}, \tag{2.10}$$

with probability at least $1 - 1/n^4$.

If Z is a matrix from the worst case model, then

$$\|\widetilde{Z}^E\|_2 \leq \frac{2|E|}{n\sqrt{\alpha}}\, Z_{\max}\,,$$

for any realization of E.

We showed that if $|E| \geq 15\alpha n \log n$ then no row or column is over-represented with high probability. It follows that in the regime of $|E|$ for which the conditions of Theorem 2.6.2 are satisfied, we have $Z^E = \widetilde{Z}^E$ and hence the bounds apply to $\|Z^E\|_2$ as well. Then, among other things, this result implies that for the independent entries model the right-hand side of our error estimate, (2.8), is with high probability, smaller than $\Sigma_{\min}/(n\sqrt{\alpha})$, if $|E| \geq C r \alpha^2 n^3 \kappa^4 (\sigma/\Sigma_{\min})^2$. For the worst case model, the same statement is true if $Z_{\max} \leq \Sigma_{\min}/C n \kappa^2 \sqrt{r\alpha}$.

2.7 Proofs

In this section, we provide the proofs of the main results, which rely on a number of key technical lemmas. For the sake of brevity, we refer to [56, 57] for the proofs of some of the lemmas.

2.7.1 Independent sampling model

Under the *uniform sampling model* assumed in Section 2.1, each subset E is chosen with probability $1/\binom{mn}{|E|}$ for a fixed $|E|$. Instead of proving that the main results hold under the uniform sampling model, we prove that the results hold under an *independent sampling model*, where each entry is independently sampled.

Define $\epsilon \equiv |E|/\sqrt{mn}$. In the case $m = n$, ϵ corresponds to the average number of revealed entries per row or per column. In practice, it is convenient to work with a model in which each entry is revealed independently with probability $\epsilon/\sqrt{mn}$. Elementary tail bounds on binomial random variables imply that, under the independent sampling model (also called Bernoulli model [20]), there exists a constant A such that, for all $\epsilon\sqrt{\alpha} \geq 1$

$$\mathbb{P}\Big\{|E| \in [n\epsilon\sqrt{\alpha} - A\sqrt{n \log n}, n\epsilon\sqrt{\alpha} + A\sqrt{n\log n}]\Big\} \geq 1 - \frac{1}{n^{10}}\,.$$

Since the success of the algorithm is a monotone function of $|E|$ (we can always 'throw away' entries) any guarantee proved within one model holds within the other model as well, if we allow for a vanishing shift in ϵ. This type of ensemble equivalence is standard and heavily used in random graph theory [63, 14].

2.7.2 Proof of Theorem 2.6.1

We want to prove that $\mathcal{P}_r(\widetilde{N}^E)$ is a good estimation of M with bounded error as in (2.7). Our key technical result is that the random matrix $(\alpha n^2/|E|)\widetilde{M}^E$ has spectral properties close to the original matrix M.

Lemma 2.7.1. *There exists a numerical constant $C > 0$ such that, with probability larger than $1 - 1/n^3$,*

$$\Big\|M - \frac{\alpha n^2}{|E|}\widetilde{M}^E\Big\|_2 \le C M_{\max}\, n\, \alpha\sqrt{\frac{n\sqrt{\alpha}}{|E|}}\ . \qquad (2.11)$$

This lemma generalizes a celebrated result on the second eigenvalue of a random graph by Friedman, Kahn, and Szemerédi [42], and Feige and Ofek [40]. The implication of this lemma is illustrated in Figure 2.1. The spectrum of $\widetilde{M}^E$ clearly reveals the rank-3 structure of M.

A main challenge in proving the above lemma is that the trimming step generates dependencies among the entries of $\widetilde{M}^E$. To address this challenge, we use techniques inspired by [42] and [40]. The proof of this lemma is somewhat cumbersome and, for the sake of brevity, we refer to [56] for the proof. Notice that (2.11) does not depend on the rank of M and, thus, generally holds for any M with any rank. Further, notice that (2.11) does not depend on the incoherence of M.

To bound $\|M - \mathcal{P}_r(\widetilde{N}^E)\|_F$, we start by noting that $M - \mathcal{P}_r(\widetilde{N}^E)$ is a random matrix with rank at most $2r$. For any matrix A of rank at most $2r$, we have $\|A\|_F \le \sqrt{2r}\|A\|_2$. Hence it is sufficient to bound the operator norm $\|M - \mathcal{P}_r(\widetilde{N}^E)\|_2$.

Define the singular values $\{\sigma_i\}$ and the singular vectors $\{x_i\}$ and $\{y_i\}$ as

$$\Big(\frac{\alpha n^2}{|E|}\Big)\widetilde{N}^E = \sum_{i\in[n]} \sigma_i x_i y_i^T\ ,$$

where $\sigma_1 \geq \cdots \geq \sigma_n$. Recall that $\widetilde{N}^E = \widetilde{M}^E + \widetilde{Z}^E$ and $\mathcal{P}_r(\widetilde{N}^E) = (\alpha n^2/|E|)\widetilde{N}^E - \sum_{i\geq r+1}\sigma_i x_i y_i^T$, whence

$$M - \mathcal{P}_r(\widetilde{N}^E) = \Big(M - \frac{\alpha n^2}{|E|}\widetilde{M}^E\Big) + \Big(\sum_{i\geq r+1}\sigma_i x_i y_i^T\Big) - \frac{\alpha n^2}{|E|}\widetilde{Z}^E \,.$$

Applying triangular inequality to the operator norm, we get

$$\|M - \mathcal{P}_r(\widetilde{N}^E)\|_2 \;\leq\; \Big\|M - \frac{\alpha n^2}{|E|}\widetilde{M}^E\Big\|_2 + \sigma_{r+1} + \Big\|\frac{\alpha n^2}{|E|}\widetilde{Z}^E\Big\|_2 \,. \tag{2.12}$$

Let $\tilde{\sigma}_k(\cdot)$ denote the kth largest singular value of a matrix. Then, applying Weyl's inequality [50] to bound $\sigma_{r+1} \equiv (\alpha n^2/|E|)\tilde{\sigma}_{r+1}(\widetilde{M}^E + \widetilde{Z}^E)$, we get

$$\begin{aligned}\sigma_{r+1} \;&\leq\; \tilde{\sigma}_1\Big(\frac{\alpha n^2}{|E|}\widetilde{M}^E - M\Big) + \tilde{\sigma}_1\Big(\frac{\alpha n^2}{|E|}\widetilde{Z}^E\Big) + \tilde{\sigma}_{r+1}(M)\\ &=\; \Big\|\frac{\alpha n^2}{|E|}\widetilde{M}^E - M\Big\|_2 + \Big\|\frac{\alpha n^2}{|E|}\widetilde{Z}^E\Big\|_2 \,,\end{aligned}$$

Substituting it into (2.12) and applying $\|M - \mathcal{P}_r(\widetilde{N}^E)\|_F \leq \sqrt{2r}\|M - \mathcal{P}_r(\widetilde{N}^E)\|_2$, we have

$$\|M - \mathcal{P}_r(\widetilde{N}^E)\|_F \;\leq\; 2\sqrt{2r}\Big\|M - \frac{\alpha n^2}{|E|}\widetilde{M}^E\Big\|_2 + 2\sqrt{2r}\Big\|\frac{\alpha n^2}{|E|}\widetilde{Z}^E\Big\|_2 \,.$$

Together with Lemma 2.7.1, this finishes the proof of Theorem 2.6.1.

2.7.3 Proof of Theorem 2.6.2

Recall that the cost function is defined over the Riemannian manifold $\mathsf{M}(m,n) \equiv \mathsf{G}(m,r) \times \mathsf{G}(n,r)$. The definitions given in Section 2.5.1 will be heavily used in this section. We first present a few preliminary remarks that are useful, then we present two key lemmas which are crucial in proving Theorem 2.6.2. In the proof, we will use C, C', etc. to denote general numerical constants, and C_1, C_2, etc. to denote specific numerical constants.

Preliminary remarks and auxiliary lemmas

The next remark, together with Theorem 2.6.1, shows that we can get an initial estimate (X_0, Y_0) close to the correct solution (U, V) with trimming-plus-SVD approach. Let $\mathbf{I}_r$ denote an $r \times r$ identity matrix.

Remark 2.7.2. *Let $U, X \in \mathbb{R}^{m\times r}$ with $U^TU = X^TX = \mathbf{I}_r$, $V, Y \in \mathbb{R}^{n\times r}$ with $V^TV = Y^TY = \mathbf{I}_r$, and $M = U\Sigma V^T$, $\widehat{M} = XSY^T$ for $\Sigma = \mathrm{diag}(\Sigma_1, \dots, \Sigma_r)$ and $S \in \mathbb{R}^{r\times r}$. If $\Sigma_1, \dots, \Sigma_r \geq \Sigma_{\min}$, then*

$$d(U, X) \leq \frac{\pi}{\sqrt{2}\Sigma_{\min}} \|M - \widehat{M}\|_F \quad , \qquad d(V, Y) \leq \frac{\pi}{\sqrt{2}\Sigma_{\min}} \|M - \widehat{M}\|_F \, .$$

Proof of Remark 2.7.2. We start by observing that $d(V, Y) \leq \pi d_{\mathrm{p}}(V, Y)$ from Lemma 2.5.1 and

$$d_{\mathrm{p}}(V, Y) = \frac{1}{\sqrt{2}} \min_{A\in\mathbb{R}^{r\times r}} \|V - YA\|_F \, . \tag{2.13}$$

Indeed the minimization on the right hand side can be performed explicitly (as $\|V - YA\|_F^2$ is a quadratic function of A) and the minimum is achieved at $A = Y^TV$. This implies (2.13) from the definition of $d_{\mathrm{p}}(\cdot)$.

To prove the remark, take $A = S^TX^TU\Sigma^{-1}$. Then

$$\begin{aligned} \|V - YA\|_F &= \sup_{B, \|B\|_F \leq 1} \langle B, (V - YA)\rangle \\ &= \sup_{B, \|B\|_F \leq 1} \langle B^T, \Sigma^{-1}U^T(U\Sigma V^T - XSY^T)\rangle \\ &= \sup_{B, \|B\|_F \leq 1} \langle U\Sigma^{-1}B^T, (M - \widehat{M})\rangle \\ &\leq \sup_{B, \|B\|_F \leq 1} \|U\Sigma^{-1}B^T\|_F \, \|M - \widehat{M}\|_F \, . \end{aligned} \tag{2.14}$$

On the other hand

$$\|U\Sigma^{-1}B^T\|_F^2 = \mathrm{Tr}(B\Sigma^{-1}U^TU\Sigma^{-1}B^T) = \mathrm{Tr}(B^TB\Sigma^{-2}) \leq \Sigma_{\min}^{-2}\|B\|_F^2 \, ,$$

whereby the last inequality follows from the fact that Σ is a diagonal matrix. Together with (2.13) and (2.14), this implies our claim. $\square$

The following remark provides a bound on $\|Z^E\|_2$. Let $\epsilon \equiv |E|/\sqrt{mn}$.

Remark 2.7.3. *By assumption that the right-hand side of* (2.8) *is smaller than* $\Sigma_{\min}/(n\sqrt{\alpha})$, *the following is true:*

$$\|Z^E\|_2 \le \frac{\epsilon \Sigma_{\min}}{C_2 \kappa^2 n \sqrt{\alpha r}} \, , \tag{2.15}$$

where C_2 can be made as large as we want by modifying the constant appearing in the statement of Theorem 2.6.2.

The main idea of the proof of Theorem 2.6.2 is that when $\mathcal{P}_r(\widetilde{N}^E)$ is close to M the cost function is well approximated by a quadratic function plus some noise term. The proof consists in controlling the behavior of $F(\cdot)$ in a neighborhood of $\mathbf{u} = (U, V)$ (the point corresponding to the matrix M to be reconstructed). The proof is based on the following two lemmas. Given S achieving the minimum in (2.1), it is also convenient to introduce the notations

$$d_-(\mathbf{x}, \mathbf{u}) \equiv \frac{1}{n\sqrt{\alpha}} \sqrt{\Sigma_{\min}^2 d(\mathbf{x}, \mathbf{u})^2 + \|S - \Sigma\|_F^2} \, ,$$
$$d_+(\mathbf{x}, \mathbf{u}) \equiv \frac{1}{n\sqrt{\alpha}} \sqrt{\Sigma_{\max}^2 d(\mathbf{x}, \mathbf{u})^2 + \|S - \Sigma\|_F^2} \, ,$$

and let $\mathcal{K}(\mu)$ be the set of matrix couples $(X, Y) \in \mathbb{R}^{m \times r} \times \mathbb{R}^{n \times r}$ such that $\|X^{(i)}\|^2 \le \mu r/m$, $\|Y^{(j)}\|^2 \le \mu r/n$ for all i, j, where $X^{(i)}$ is the ith row of X and $Y^{(j)}$ is the jth row of Y.

Lemma 2.7.4. *There exists numerical constants C_1, C_3, C_4, C_5 such that the following happens. Assume $\epsilon \ge C_1 \mu r \sqrt{\alpha}\, \max\{\log n \, ; \, \mu r \sqrt{\alpha} \kappa^4\}$ and $\delta \le 1/(C_3 \kappa)$. Then,*

$$F(\mathbf{x}) - F(\mathbf{u}) \;\ge\; C_4 \epsilon n \sqrt{\alpha} d_-(\mathbf{x}, \mathbf{u})^2 - C_4 n \sqrt{r\alpha} \|Z^E\|_2 d_+(\mathbf{x}, \mathbf{u}) \, , \tag{2.16}$$
$$F(\mathbf{x}) - F(\mathbf{u}) \;\le\; \frac{C_5 \epsilon}{n\sqrt{\alpha}} \Sigma_{\max}^2 \, d(\mathbf{x}, \mathbf{u})^2 + C_5 n \sqrt{r\alpha} \|Z^E\|_2 d_+(\mathbf{x}, \mathbf{u}) \, , \tag{2.17}$$

for all $\mathbf{x} \in \mathsf{M}(m,n) \cap \mathcal{K}(4\mu)$ *such that* $d(\mathbf{x},\mathbf{u}) \le \delta$, *with probability at least* $1 - 1/n^4$. *Here* $S \in \mathbb{R}^{r\times r}$ *is the matrix realizing the minimum in* (2.1).

For the proof of this lemma we refer to [57]. The next corollary follows from the above lemma.

Corollary 2.7.5. *There exists a constant* C *such that, under the hypotheses of Lemma 2.7.4,*

$$\|S - \Sigma\|_F \le C\Sigma_{\max} d(\mathbf{x},\mathbf{u}) + C\frac{n\sqrt{r\alpha}}{\epsilon}\,\|Z^E\|_2\,.$$

Further, for an appropriate choice of the constants in Lemma 2.7.4, we have

$$\begin{aligned}\sigma_{\max}(S) &\le 2\Sigma_{\max} + C\frac{n\sqrt{r\alpha}}{\epsilon}\,\|Z^E\|_2\,, &(2.18)\\ \sigma_{\min}(S) &\ge \frac{1}{2}\Sigma_{\min} - C\frac{n\sqrt{r\alpha}}{\epsilon}\,\|Z^E\|_2\,. &(2.19)\end{aligned}$$

Proof of Corollary 2.7.5. By putting together (2.16) and (2.17) and using the definitions of $d_+(\mathbf{x},\mathbf{u})$ and $d_-(\mathbf{x},\mathbf{u})$, we get

$$\begin{aligned}\|S-\Sigma\|_F^2 \;\le\; & \frac{(C_4+C_5)}{C_4}\Sigma_{\max}^2 d(\mathbf{x},\mathbf{u})^2\\ & + \frac{(C_4+C_5)n\sqrt{\alpha r}}{C_4\epsilon}\|Z^E\|_2\sqrt{\Sigma_{\max}^2 d(\mathbf{x},\mathbf{u})^2 + \|S-\Sigma\|_F^2}\,.\end{aligned}$$

To prove the bound on $\|S-\Sigma\|_F$, let $x \equiv \|S-\Sigma\|_F$, $a^2 \equiv \big((C_4+C_5)/C_4\big)\Sigma_{\max}^2 d(\mathbf{x},\mathbf{u})^2$, and $b \equiv \big((C_4+C_5)n\sqrt{\alpha r}/C_4\epsilon\big)\|Z^E\|_2$. The above inequality then takes the form

$$x^2 \le a^2 + b\sqrt{x^2+a^2} \le a^2 + ab + bx\,,$$

which implies our claim $x \le a+b$.

The singular value bounds (2.18) and (2.19) follow by triangular inequality. For instance

$$\sigma_{\min}(S) \ge \Sigma_{\min} - C\Sigma_{\max} d(\mathbf{x},\mathbf{u}) - C\frac{n\sqrt{\alpha r}}{\epsilon}\|Z^E\|_2\,.$$

which implies the inequality (2.19) for $d(\mathbf{x},\mathbf{u}) \le \delta = \Sigma_{\min}/C_3\Sigma_{\max}$ and C_3 large enough. An analogous argument proves (2.18). □

Further, using Corollary 2.7.5 in (2.16) and (2.17), we get the following corollary after some algebra. Define

$$\delta_{0,-} \equiv C\frac{n\kappa\sqrt{r\alpha}}{\epsilon\Sigma_{\min}}\|Z^E\|_2\,, \qquad \delta_{0,+} \equiv C\frac{n\kappa\sqrt{r\alpha}}{\epsilon\Sigma_{\max}}\|Z^E\|_2\,.$$

We can assume $\delta_{0,-} \le \delta/20$ and $\delta_{0,+} \le \sqrt{C_4/C_5}(\delta/(20\kappa))$ by substituting (2.15) with large enough C_2.

Corollary 2.7.6. *There exists a constant C such that, under the hypotheses of Lemma 2.7.4,*

$$\begin{aligned} F(\mathbf{x}) - F(\mathbf{u}) &\ge C_4\frac{\epsilon\Sigma_{\min}^2}{n\sqrt{\alpha}}\left\{d(\mathbf{x},\mathbf{u})^2 - \delta_{0,-}^2\right\}\,, &(2.20)\\ F(\mathbf{x}) - F(\mathbf{u}) &\le C_5\frac{\epsilon\Sigma_{\max}^2}{n\sqrt{\alpha}}\left\{d(\mathbf{x},\mathbf{u})^2 + \delta_{0,+}^2\right\}\,. &(2.21)\end{aligned}$$

The next lemma shows that, in a neighborhood of $\mathbf{u}$, local minimizers satisfying $\operatorname{grad}\widetilde{F}(\mathbf{x}) = 0$ are all close to $\mathbf{u}$: $d(\mathbf{x},\mathbf{u}) \le (C_7 n\kappa\sqrt{r\alpha}/(\epsilon\Sigma_{\min}))\|Z^E\|_2$.

Lemma 2.7.7. *There exists numerical constants C_1, C_3, C_6, C_7 such that the following happens. Assume $\epsilon \ge C_1\mu r\sqrt{\alpha}\,\kappa^2\max\{\log n\,;\,\mu r\sqrt{\alpha}\kappa^4\}$ and $\delta \le 1/(C_3\kappa)$. Then,*

$$\|\operatorname{grad}\widetilde{F}(\mathbf{x})\|^2 \ge C_6\frac{\epsilon^2\Sigma_{\min}^4}{n\alpha}\left[d(\mathbf{x},\mathbf{u}) - C_7\frac{n\kappa\sqrt{r\alpha}}{\epsilon\Sigma_{\min}}\|Z^E\|_2\right]_+^2\,, \qquad (2.22)$$

for all $\mathbf{x} \in \mathsf{M}(m,n)\cap\mathcal{K}(4\mu)$ such that $d(\mathbf{x},\mathbf{u}) \le \delta$, with probability at least $1 - 1/n^4$. (Here $[a]_+ \equiv \max(a,0)$.)

For the proof of this lemma we refer to [57]. We can now turn to the proof of Theorem 2.6.2.

Proof of Theorem 2.6.2

Let $\delta = 1/(C_3\kappa)$ with C_3 large enough so that the hypotheses of Lemmas 2.7.4 and 2.7.7 are verified. The strategy for proving Theorem 2.6.2 is as follows. Call $\{\mathbf{x}_k\}_{k\geq 0}$ the sequence of pairs $(X_k, Y_k) \in \mathsf{M}(m,n)$ generated by k iterations of gradient descent. First, we claim that we get a good starting point $\mathbf{x}_0$ with distance at most $\delta/10$ from $\mathbf{u}$. Further, with this starting point, we can show that all the $\mathbf{x}_k$'s satisfy $\mathbf{x}_k \in \mathcal{K}(4\mu)$ and $d(\mathbf{u}, \mathbf{x}_k) \leq \delta/10$. The proof of these claims are provided later in this section.

These claims imply that the lemmas 2.7.4 and 2.7.7 are satisfied for all $\mathbf{x}_k$ and, in particular, the sequence $\{\mathbf{x}_k\}$ converges to

$$\Omega = \big\{\mathbf{x} \in \mathcal{K}(4\mu_0) \cap \mathsf{M}(m,n)\;:\; d(\mathbf{x},\mathbf{u}) \leq \delta\,, \mathrm{grad}\,\widetilde{F}(\mathbf{x}) = 0\,\big\}\,.$$

By Lemma 2.7.7, for any $\mathbf{x} \in \Omega$ we have

$$d(\mathbf{x},\mathbf{u}) \leq C_7 \frac{n\kappa\sqrt{\alpha r}}{\epsilon \Sigma_{\min}} \|Z^E\|_2\,. \tag{2.23}$$

By triangular inequality,

$$\begin{aligned}
\|M - XSY^T\|_F^2 &\leq 3\|X(S-\Sigma)Y^T\|_F^2 + 3\|X\Sigma(Y-V)^T\|_F^2 + 3\|(X-U)\Sigma V^T\|_F^2 \\
&\leq 3\|S-\Sigma\|_F^2 + 3\Sigma_{\max}^2(\|X-U\|_F^2 + \|Y-V\|_F^2) \\
&\leq Cn^2\alpha d_+(\mathbf{x},\mathbf{u})^2\,,
\end{aligned}$$

Using Corollary 2.7.5, we have

$$\begin{aligned}
d_+(\mathbf{x},\mathbf{u}) &\leq \frac{1}{n\sqrt{\alpha}}\Big(\Sigma_{\max} d(\mathbf{x},\mathbf{u}) + \|S-\Sigma\|_F\Big) \\
&\leq \frac{C}{n\sqrt{\alpha}}\Sigma_{\max} d(\mathbf{x},\mathbf{u}) + \frac{C\sqrt{r}}{\epsilon}\|Z^E\|_2\,.
\end{aligned}$$

Substituting (2.23) in the above series of inequalities, this implies for all $\mathbf{x} \in \Omega$,

$$\frac{1}{n\sqrt{\alpha}}\|M - XSY^T\|_F \leq C\frac{\kappa^2\sqrt{r}}{\epsilon}\|Z^E\|_2\,,$$

which proves Theorem 2.6.2.

Now we are left to prove the following three claims:

1. $d(\mathbf{x}_0, \mathbf{u}) \le \delta/10$.

   First, we apply Theorem 2.6.1 to show that we get a good starting point $\mathbf{x}_0$ with distance at most $\delta/10$ from $\mathbf{u}$. From Remark 2.7.2 we have $d(\mathbf{u}, \mathbf{x}_0) \le (\pi/\Sigma_{\min})\|M - X_0 S_0 Y_0^T\|_F$, where S_0 is the minimizer in (2.1). Together with Theorem 2.6.1 this implies

$$d(\mathbf{x}_0, \mathbf{u}) \le \frac{C M_{\max}\, n\, \alpha}{\Sigma_{\min}} \left(\frac{r}{\epsilon}\right)^{1/2} + \frac{C'\, n\sqrt{\alpha r}}{\epsilon \Sigma_{\min}} \, \|\widetilde{Z}^E\|_2 \, .$$

   Since $\epsilon \ge C_1 \alpha \mu^2 r^2 \kappa^4$ as per our assumptions and $M_{\max} \le \mu\sqrt{r}\Sigma_{\max}/(n\sqrt{\alpha})$ for incoherent M, the first term in the above bound is upper bounded by $C/(C_1^{1/2}\kappa)$. For a given C_3, we can choose large enough C_1 such that $C/(C_1^{1/2}\kappa) \le 1/(20C_3\kappa)$. Using (2.15), with large enough constant C_2, the second term in the above bound is also upper bounded by $1/(20C_3\kappa)$. This proves that if $\epsilon \ge C_1\alpha\mu^2 r^2 \kappa^4$ then for $\delta = 1/(C_3\kappa)$,

$$d(\mathbf{u}, \mathbf{x}_0) \le \frac{\delta}{10} \, .$$

2. $\mathbf{x}_k \in \mathcal{K}(4\mu)$ for all k.

   First we notice that we can assume $\mathbf{x}_0 \in \mathcal{K}(3\mu)$. Indeed, if this does not hold, we can 'rescale' those rows of X_0, Y_0 that violate the constraint. The following remark shows that such rescaling is possible. For the proof of this remark we refer to [56].

   Remark 2.7.8. *Let $U, X \in \mathbb{R}^{n\times r}$ with $U^T U = X^T X = \mathbf{I}_r$ and $U \in \mathcal{K}(\mu)$ and $d(X, U) \le \delta \le \frac{1}{16}$. Then there exists $X' \in \mathbb{R}^{n\times r}$ such that $X'^T X' = \mathbf{I}_r$, $X' \in \mathcal{K}(3\mu)$ and $d(X', U) \le 4\delta$. Further, such an X' can be computed from X in a time of $O(nr^2)$.*

   Since $\mathbf{x}_0 \in \mathcal{K}(3\mu)$, $\widetilde{F}(\mathbf{x}_0) = F(\mathbf{x}_0) \le 4C_5\epsilon^2\Sigma_{\max}^2\delta^2/(100n\sqrt{\alpha})$, which follows from (2.21) after some algebra. On the other hand $\widetilde{F}(\mathbf{x}) \ge \rho(e^{1/9} - 1)$ for

$\mathbf{x} \notin \mathcal{K}(4\mu)$. Since $\widetilde{F}(\mathbf{x}_k)$ is a non-increasing sequence, the thesis follows provided we take $\rho \geq C\epsilon^2\Sigma_{\min}^2/(n\sqrt{\alpha})$.

3. $d(\mathbf{x}_k, \mathbf{u}) \leq \delta/10$ for all k.

   Since $\epsilon \geq C_1\alpha\mu^2r^2\kappa^6$ with large enough C_1 as per our assumptions in Theorem 2.6.2, we have $d(\mathbf{x}_0, \mathbf{u})^2 \leq (C_4/C_5\kappa^2)(\delta/20)^2$. Also assuming (2.15) with large enough C_2, we have $\delta_{0,-} \leq \delta/20$ and $\delta_{0,+} \leq \sqrt{C_4/C_5}(\delta/20\kappa)$. Then, by (2.21),

$$F(\mathbf{x}_0) \leq F(\mathbf{u}) + C_4\epsilon\Sigma_{\min}^2 \frac{2\delta^2}{400n\sqrt{\alpha}} \, .$$

   Now, set the algorithm parameter to $\gamma = \delta/2$ such that $d(\mathbf{x}_k, \mathbf{u}) \leq \delta$. Using (2.20), for all $\mathbf{x}_k$ such that $d(\mathbf{x}_k, \mathbf{u}) \in [\delta/10, \delta]$, we have

$$F(\mathbf{x}) \geq F(\mathbf{u}) + C_4\epsilon\Sigma_{\min}^2 \frac{3\delta^2}{400n\sqrt{\alpha}} \, .$$

   Hence, for all $\mathbf{x}_k$ such that $d(\mathbf{x}_k, \mathbf{u}) \in [\delta/10, \delta]$, we have $\widetilde{F}(\mathbf{x}) \geq F(\mathbf{x}) \geq F(\mathbf{x}_0)$. This contradicts the monotonicity of $\widetilde{F}(\mathbf{x})$, and thus proves the claim.

2.7.4 Proof of Theorem 2.6.3

Proof. (*Independent entries model*) We start with a claim that for any sampling set E, we have

$$\|\widetilde{Z}^E\|_2 \leq \|Z^E\|_2 \, .$$

It is, therefore, enough to show that the bounds hold for $\|Z^E\|_2$, in which case the independence of the entries can be exploited. To prove this claim, let x^* and y^* be m and n dimensional vectors, respectively, achieving the optimum in $\max_{\|x\|\leq 1, \|y\|\leq 1}\{x^T\widetilde{Z}^Ey\}$. Equivalently, we have $\|\widetilde{Z}^E\|_2 = x^{*T}\widetilde{Z}^Ey^*$. Recall that, as a result of the trimming step, all the entries in trimmed rows and columns of $\widetilde{Z}^E$ are set to zero. Then, there is no gain in maximizing $x^T\widetilde{Z}^Ey$ to have a non-zero entry x_i^* if ith row is trimmed. Analogous result holds for y^*. It follows that $x^{*T}\widetilde{Z}^Ey^* = x^{*T}Z^Ey^* \leq \|Z^E\|_2$.

Now we are left to bound $\|Z^E\|_2$. The strategy is to bound the expectation

$\mathbb{E}\big[\|Z^E\|_2\big]$ and use the concentration inequality for Lipschitz functions on i.i.d. Gaussian random variables.

Note that $\|\cdot\|_2$ is a Lipschitz function with a Lipschitz constant 1. Indeed, for any M and M', $\big|\|M'\|_2 - \|M\|_2\big| \le \|M' - M\|_2 \le \|M' - M\|_F$, where the first inequality follows from triangular inequality and the second inequality follows from the fact that $\|\cdot\|_F^2$ is the sum of the squared singular values. To bound the probability of a large deviation, we use Talagrand's result on concentration inequality for Lipschitz functions on i.i.d. sub-Gaussian random variables [104]. We have independent random variables $\{Z^E_{ij}\}_{i\in[m],j\in[n]}$ with zero mean and sub-Gaussian tails with parameter σ^2. For a 1-Lipschitz function $\|\cdot\|_2$, it follows that

$$\mathbb{P}\big(\|Z^E\|_2 - \mathbb{E}[\|Z^E\|_2] > t\big) \le \exp\Big\{-\frac{t^2}{2\sigma^2}\Big\}\,.$$

Setting $t = \sqrt{8\sigma^2\log n}$, this implies that $\|Z^E\|_2 \le \mathbb{E}\big[\|Z\|_2\big] + \sqrt{8\sigma^2\log n}$ with probability larger than $1 - 1/n^4$. Now, we are left to bound the expectation $\mathbb{E}\big[\|Z^E\|_2\big]$ using the following lemma.

Lemma 2.7.9 (Expectation of the spectral norm). *There exists a numerical constant $C > 0$ such that, if $|E| \ge n$ then*

$$\mathbb{E}\big[\|Z^E\|_2\big] \le C\sigma\sqrt{\frac{|E|\log n}{n}}\,.$$

Further, if $|E| \ge \alpha n\log n$, then

$$\mathbb{E}\big[\|Z^E\|_2\big] \le C\sigma\sqrt{\frac{|E|}{n}}\,.$$

The proof of this lemma uses a result from [92] on expected norm of a random matrix. Together with Talagrand's inequality, this proves the desired thesis. □

Proof of the Lemma 2.7.9. Our strategy to bound the expectation $\mathbb{E}\big[\|Z^E\|_2\big]$ is to apply the result from [92] on expected norm of random matrices with i.i.d. symmetric random entries. First, we symmetrize the possibly asymmetric random variables Z^E_{ij}. Let Z'_{ij}'s be independent copies of Z_{ij}'s, and ξ_{ij}'s be independent Bernoulli random variables such that $\xi_{ij} = +1$ with probability $1/2$ and $\xi_{ij} = -1$ with probability $1/2$. Then, by convexity of $\mathbb{E}\big[\|Z^E - Z'^E\|_2 | Z'^E\big]$ and Jensen's inequality,

$$\mathbb{E}\big[\|Z^E\|_2\big] \le \mathbb{E}\big[\|Z^E - Z'^E\|_2\big] = \mathbb{E}\big[\|\{\xi_{ij}(Z^E_{ij} - Z'^E_{ij})\}\|_2\big] \le 2\mathbb{E}\big[\|\{\xi_{ij}Z^E_{ij}\}\|_2\big] \ ,$$

where $\{\xi_{ij}Z^E_{ij}\}$ denotes an $m \times n$ matrix with entry $\xi_{ij}Z^E_{ij}$ in position (i,j). Thus, it is enough to consider symmetric random variables Z_{ij}'s.

To this end, we apply the following bound on expected norm of random matrices with i.i.d. symmetric random entries [92, Theorem 1.1].

$$\mathbb{E}\big[\|Z^E\|_2\big] \le C\Big(\mathbb{E}\big[\max_{i\in[m]} \|Z^E_{i\bullet}\|\big] + \mathbb{E}\big[\max_{j\in[n]} \|Z^E_{\bullet j}\|\big]\Big) \ , \tag{2.24}$$

for a general numerical constant C where $Z^E_{i\bullet}$ and $Z^E_{\bullet j}$ denote the ith row and jth column of Z^E respectively. For a positive parameter β, which will be specified later, the following is true.

$$\mathbb{E}\big[\max_j \|Z^E_{\bullet j}\|^2\big] \le \beta\sigma^2\frac{|E|}{n} + \int_0^\infty \mathbb{P}\Big(\max_j \|Z^E_{\bullet j}\|^2 \ge \beta\sigma^2\frac{|E|}{n} + z\Big)\,\mathrm{d}z\,. \tag{2.25}$$

To bound the second term, we want to apply union bound on each of the n columns. Further, we claim

$$\mathbb{P}\Big(\sum_{k=1}^m (Z^E_{kj})^2 \ge \beta\sigma^2\frac{|E|}{n} + z\Big) \le \exp\Big\{-\frac{3}{8}\Big((\beta-3)\frac{|E|}{n} + \frac{z}{\sigma^2}\Big)\Big\}\,. \tag{2.26}$$

To prove the above result, we apply Chernoff bound on the sum of independent random variables. Recall that $Z^E_{kj} = \tilde{\xi}_{kj}Z_{kj}$ where $\tilde{\xi}$'s are independent Bernoulli random variables such that $\tilde{\xi} = 1$ with probability $|E|/mn$ and zero with probability

$1-|E|/mn$. Then, for the choice of $\lambda = 3/(8\sigma^2) < 1/(2\sigma^2)$,

$$\begin{aligned}
\mathbb{E}\Big[\exp\Big(\lambda\sum_{k=1}^{m}(\tilde{\xi}_{kj}Z_{kj})^2\Big)\Big] &= \Big(1-\frac{|E|}{mn}+\frac{|E|}{mn}\mathbb{E}[e^{\lambda Z_{kj}^2}]\Big)^m \\
&\le \Big(1-\frac{|E|}{mn}+\frac{|E|}{mn\sqrt{(1-2\sigma^2\lambda)}}\Big)^m \\
&= \exp\Big\{m\log\Big(1+\frac{|E|}{mn}\Big)\Big\} \\
&\le \exp\Big\{\frac{|E|}{n}\Big\}\,,
\end{aligned}$$

where the first inequality follows from the definition of Z_{kj} as a zero mean random variable with sub-Gaussian tail and the second inequality follows from $\log(1+x)\le x$. By applying Chernoff bound, (2.26) follows. Note that an analogous result holds for the Euclidean norm on the rows $\|Z^E_{i\bullet}\|^2$.

Substituting (2.26) and $\mathbb{P}\big(\max_j\|Z^E_{\bullet j}\|^2 \ge z\big) \le m\,\mathbb{P}\big(\|Z^E_{\bullet j}\|^2\ge z\big)$ in (2.25), we get

$$\mathbb{E}\big[\max_j\|Z^E_{\bullet j}\|^2\big] \le \beta\sigma^2\frac{|E|}{n}+\frac{8\sigma^2 m}{3}e^{-(\beta-3)\frac{3|E|}{8n}}\,. \tag{2.27}$$

The second term can be made arbitrarily small by taking $\beta = C\log n$ with large enough C. Since $\mathbb{E}\big[\max_j\|Z^E_{\bullet j}\|\big] \le \sqrt{\mathbb{E}\big[\max_j\|Z^E_{\bullet j}\|^2\big]}$, applying (2.27) with $\beta = C\log n$ in (2.24) gives

$$\mathbb{E}\big[\|Z^E\|_2\big] \le C\sigma\sqrt{\frac{|E|\log n}{n}}\,.$$

Together with (2.24), this proves the desired thesis for $|E|\ge n$.

In the case when $|E|\ge \alpha n\log n$, we can get a tighter bound by similar analysis. Since $|E|\ge\alpha n\log n$, the second term in (2.27) can be made arbitrarily small with a large constant β. Hence, applying (2.27) with $\beta = C$ in (2.24), we get

$$\mathbb{E}\big[\|Z^E\|_2\big] \le C\sigma\sqrt{\frac{|E|}{n}}\,.$$

This finishes the proof of the lemma.

□

Proof. (*Worst Case Model*) Let D be the $m \times n$ all-ones matrix. Then for any matrix Z from the *worst case model*, we have $\|\widetilde{Z}^E\|_2 \leq Z_{\max}\|\widetilde{D}^E\|_2$, since $x^T\widetilde{Z}^E y \leq \sum_{i,j} Z_{\max}|x_i|\widetilde{D}^E_{ij}|y_j|$, which follows from the fact that Z_{ij}'s are uniformly bounded. Further, $\widetilde{D}^E$ is an adjacency matrix of a corresponding bipartite graph with bounded degrees. Then, for any choice of E the following is true for all positive integers k:

$$\|\widetilde{D}^E\|_2^{2k} \leq \max_{x,\|x\|=1} \left| x^T((\widetilde{D}^E)^T\widetilde{D}^E)^k x \right| \leq \mathrm{Tr}\Big(\big((\widetilde{D}^E)^T\widetilde{D}^E\big)^k \Big) \,.$$

Now consider the bipartite graph with adjacency matrix $\widetilde{D}^E$. Then, $\mathrm{Tr}\Big(\big((\widetilde{D}^E)^T\widetilde{D}^E\big)^k \Big)$ is the sum of the number of paths that begin and end at node i for every $i \in [n]$ and has length $2k$. Since this graph has degree bounded by $2|E|/\sqrt{mn}$, we get

$$\|\widetilde{D}^E\|_2^{2k} \leq n\Big(\frac{2|E|}{\sqrt{mn}}\Big)^{2k} \,.$$

Taking k large, we get the desired thesis. □

Chapter 3

Comparisons and Extensions

In the previous chapter, we introduced an efficient matrix completion algorithm–OptSpace–and provided performance guarantees. One might wonder how good these guarantees are and whether this algorithm is also suitable for practical use. To answer these questions, we first present fundamental limits which coincide with the error bounds achieved with OptSpace up to numerical constants under mild assumptions. This proves that OptSpace achieves order-optimal performance under the noiseless setting (Section 3.1) and under the Gaussian noise setting (Section 3.2). We also review recent results in matrix completion and compare them to the performance guarantees of OptSpace. Next, to provide a new perspective on matrix completion, we compare OptSpace with a line of work in numerical computations, namely sampling-based algorithms for low-rank approximations (Section 3.3). In the past two years, a number of novel and efficient matrix completion algortihms have been developed. In Section 3.4, we review a subset of these recent developments, a few of which are numerically compared on both synthetic and real datasets in Section 3.5. Section 3.6 discusses two important extensions to make OptSpace more suitable in practice. Finally, we discuss some open problems and future research directions in Section 3.7. This chapter is based on joint work with Keshavan and Montanari [55, 56, 57, 58].

$$\begin{bmatrix} a & b \\ c & ? \end{bmatrix}$$

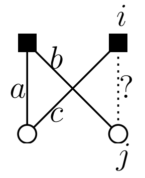


Figure 3.1: 2×2 minor with one unique missing entry

3.1 Noiseless scenario

When the sampled entries are revealed without any noise, Theorem 2.6.2 states that uniform random samples of size $O(r\mu n \max\{r\mu, \log n\})$ are sufficient to recover a rank-r μ-incoherent $m \times n$ matrix. In this section, we first present the fundamental limit on the number of samples that are necessary for exact recovery. Then, we compare the error bound in Theorem 2.6.2 to the fundamental limit and analogous results using a different algorithm, namely nuclear norm minimization.

3.1.1 Fundamental limit

We first investigate how many samples are necessary for exact recovery. As a warm-up example, consider a rank-1 $m \times n$ matrix M to be recovered from a random sample of entries. Assume that we know 3 entries of the matrix M that belong to the same 2×2 minor. Explicitly, the entries a,b, and c are known as in Figure 3.1. Unless $a = 0$, the fourth entry is uniquely determined by bc/a. The case $a = 0$ can be treated separately, but for the sake of simplicity we shall assume that all the entries of M are non-zero.

This observation suggests a simple matrix completion algorithm: Recursively look for a 2×2 minor with a unique unknown entry and complete it according to the above rule. This algorithm has a nice graph-theoretic interpretation. Let R and C denote the set of vertices corresponding to the rows and the columns of M. Consider the bipartite graph $G = (R, C, E)$ with vertices corresponding to the row and columns of M and edges to the observed entries. If a 2×2 minor has a unique unknown entry M_{ij}, it means that the corresponding vertices $i \in R$, $j \in C$ are connected by a length-3 path in G. Hence the algorithm recursively adds edges to G connecting distance-3 vertices.

After at most $O(n^2)$ operations the process described halts on a graph that is a disjoint union of cliques, corresponding to the connected components in G. Each edge corresponds to a correctly predicted matrix entry. If the graph is connected then there is a unique rank-1 matrix matching the sampled entries, and the above recursion recovers M exactly. If it is disconnected then multiple rank-1 matrices match the observations and no algorithm can distinguish the correct M. Clearly, in the large n limit only the components with $\Theta(n)$ vertices matter (as they have $\Theta(n^2)$ edges). Recall that E is the set of indices of the sampled entries and is drawn uniformly at random given sample size $|E|$. In the following, we let $\alpha \equiv (m/n)$ and $\epsilon \equiv |E|/\sqrt{mn}$. It is a fundamental result in random graph theory [38] that there is no connected component with $\Theta(n)$ vertices for $\epsilon \le 1$. For $\epsilon > 1$ there is one such component involving approximately $n\xi_\epsilon$ vertices in R and $m\zeta_\epsilon$ vertices in C, where $(\xi_\epsilon, \zeta_\epsilon)$ is the largest solution of

$$\xi = 1 - e^{-\epsilon\sqrt{\alpha}\zeta}\,, \qquad \zeta = 1 - e^{-\frac{\epsilon}{\sqrt{\alpha}}\xi}\,.$$

Applying the recursive completion algorithm, matrix element corresponding to vertex pairs in distinct components are predicted to vanish. Therefore, for a rank-1 matrix M, there is an algorithm with $O(n^2)$ complexity which achieves the following with high probability. We use a parameterization of $\epsilon = \sqrt{\alpha}(\log m + w)$.

(i) If $w \to \infty$ as $n \to \infty$, then M is recovered exactly.

(ii) If $w \to -\infty$ as $n \to \infty$, then there exists multiple rank-1 matrices matching the sampled entries, thus no algorithm can recover the original matrix M exactly.

(iii) The root mean squared error is bounded by

$$\frac{1}{\sqrt{mn}}\|M - \widehat{M}\|_F \le \sqrt{1 - \xi_\epsilon\zeta_\epsilon}\, M_{\max} + O\Big(\sqrt{\frac{\log n}{n}}\Big)\,,$$

where $M_{\max} \equiv \max_{i,j}\{|M_{ij}|\}$ [55].

When there is an isolated node in $G = (R, C, E)$ there are multiple rank-1 matrices matching all the observed entries. The property (ii) follows from the fact that if $w \to -\infty$ then there exists isolated nodes in R. The number of isolated nodes in R is asymptotically a Poisson random variable with mean e^{-w}. Hence

$$\mathbb{P}(\exists \text{ an isolated node in } R) \to 1 - e^{-e^{-w}} , \tag{3.1}$$

which converges to one. In the $G(n, p)$ model, where there are n nodes and each pair of edges is connected independently with probability p, it is known that the probability a random graph is connected is about the probability that it has no isolated node [15]. Together with (3.1), the analysis can be generalized to bipartite random graphs to prove the property (i).

The property (iii) implies that arbitrarily small root mean squared error is achieved with a large enough constant ϵ, or equivalently $|E| \geq C(m+n)$. A very simple extension of this algorithm to general rank-r matrices is introduced and analyzed in [68] under the assumption that all the r-by-r minors of U and V are full-rank (here U and V are the left and right factors resulting from the SVD of $M = USV^T$).

Another argument to understand the $\log n$ factor in the necessary condition (ii) is via coupon collector's problem. As an example, assume that we do not observe any samples in a particular row or column. Then there is no hope for correctly reconstructing that row or column. Hence, for exact recovery of M, it is necessary to sample every row and every column. Under the uniform sampling assumption, this happens with high probability only if $|E| \geq Cn \log n$. This is the famous coupon collector's problem where we have n coupons (or columns) that we want to collect and each time we collect one of those n coupons uniformly at random. It is well known that we need on the order of $n \log n$ samples to sample all the rows.

Graph connectivity and coupon collector's problem provide necessary conditions for the uniqueness of the solution. For a general rank $r > 1$, a stronger necessary condition is proved assuming incoherence by Candès and Tao [22]. Under mild assumptions on n, if

$$|E| < (3/4)\, n\, r\, \mu \log n \ , \tag{3.2}$$

there exists a μ-incoherent matrix M such that there are multiple rank-r and μ-incoherent matrices matching the samples in the set E, with probability larger than 1/2 [22, Theorem 1.7]. The definition of incoherence condition is provided in Section 2.2. In particular, this implies that if all we know about M is the rank r and the incoherence μ, then no algorithm can guarantee to succeed with less than $(3/4)nr\mu \log n$ random samples. The proof essentially uses the coupon collector's argument, but with a block diagonal structure on M. To get the factor $r\mu$ in the right-hand side of the bound, each block is constructed with size $\lfloor n/(r\mu)\rfloor \times \lfloor n/(r\mu)\rfloor$.

For a non-random sampling, a different approach based on rigidity theory gives a necessary condition for exact matrix completion. In [98], Singer and Cucuringu introduced an algorithm that checks if there exists multiple rank-r matrices that are identical in the given sampled set E. If this algorithm detects that E is not rigid, then there are multiple rank-r matrices that are identical on E. Therefore no algorithm can guarantee to succeed if all we know about M is that it has rank-r. This algorithmic bound is presented for comparison in Section 3.5.

3.1.2 Nuclear norm minimization

Assuming there is no noise in the samples, we want to compare the performance guarantees of Theorem 2.6.2 with analogous guarantees for a different approach, i.e., nuclear norm minimization. Recall the projector operator for sampling $\mathcal{P}_E(\cdot)$ defined in (2.3).

$$\mathcal{P}_E(M)_{ij} = \begin{cases} M_{ij} & \text{if } (i,j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

When there is no noise in the samples and the sample size is large enough, we can try to recover M by solving the following optimization problem.

$$\begin{aligned} \text{minimize} \quad & \text{rank}(X) \\ \text{subject to} \quad & \mathcal{P}_E(X) = \mathcal{P}_E(M)\ , \end{aligned} \tag{3.3}$$

where X is the variable matrix and rank$(\cdot)$ is the rank of a matrix. When M is a unique rank-r matrix satisfying the condition, solving this problem recovers M. However, this is an NP-hard problem and any known algorithm requires time doubly exponential in the problem dimensions [25].

We can use the nuclear norm minimization heuristic, introduced by Fazel [39], which consists solving a convex relaxation of (3.3), namely

$$\begin{aligned} \text{minimize} \quad & \|X\|_* \\ \text{subject to} \quad & \mathcal{P}_E(X) = \mathcal{P}_E(M)\ , \end{aligned} \tag{3.4}$$

where $\|X\|_*$ denotes the nuclear norm of X, i.e the sum of its singular values. The nuclear norm is also known as the Ky-Fan n-norm or the trace norm, and is commonly used as a convex relaxation of the rank$(\cdot)$ of a matrix. It has also been shown that this problem can be formulated as a Semidefinite Program (SDP) which in turn can be solved using off-the-shelf solvers with time complexity $O(n^4)$.

Nuclear norm minimization has a close relationship with *compressed sensing* [30, 21]. At the heart of compressed sensing is the problem of finding the sparsest vector satisfying a set of affine constraints, i.e.,

$$\begin{aligned} \text{minimize} \quad & \|x\|_{\ell_0} \\ \text{subject to} \quad & Ax = b\ , \end{aligned} \tag{3.5}$$

where $\|\cdot\|_{\ell_0}$ denotes the sparsity, the number of nonzero entries. To tackle this

NP-hard problem, a common heuristic is to instead solve a convex relaxation:

$$\begin{aligned} \text{minimize} \quad & \|x\|_{\ell_1} \\ \text{subject to} \quad & Ax = b \ , \end{aligned} \tag{3.6}$$

where $\|\cdot\|_{\ell_1}$ denotes the sum of absolute values of all the entries. ℓ_1 norm is the tightest convex envelop of ℓ_0 norm and the ℓ_1 minimization is known to provide an optimal solution to (3.5) under appropriate conditions on the measurement matrix A.

In case of the rank minimization problem (3.3), the rank function counts the number of nonzero singular values whereas, in the nuclear norm minimization (3.4), the nuclear norm sums the magnitude of singular values. Hence, the rank$(\cdot)$ of a matrix is analogous to the ℓ_0 norm of a vector, and the nuclear norm $\|\cdot\|_*$ corresponds to the ℓ_1 norm. There are close connections between compressed sensing and rank minimization, and a number of matrix completion algorithms are in fact based on compressed sensing. Some of these algorithms are described in Section 3.4. Further connections between rank minimization and compressed sensing are investigated in [81], which also provides performance guarantees under appropriate conditions on the constraints.

In a breakthrough paper [20], Candès and Recht analyzed the nuclear norm minimization (3.4) in the context of matrix completion. A significant theoretical challenge here is that the condition that is proven to be quite effective for compressed sensing, namely the restricted isometry property, is not satisfied in the matrix completion setting. To address this challenge, Candès and Recht introduced the incoherence property which is a less restrictive condition and has proved to be very useful. Under the simple noiseless setting and assuming the incoherence property and uniform sampling, they proved that there exists a numerical constant C such that if $|E| \geq Cn^{6/5}\mu r \log n$, then solving (3.4) recovers M correctly with high probability. Here, μ is the incoherence parameter of the original matrix M. For further details on incoherence we refer to Section 2.2.

The sufficient condition was later tightened with contributions from [22, 46, 80].

A major breakthrough came from the powerful ideas introduced by Gross [47] in the context of quantum-state tomography. In particular, using the clever 'golfing scheme', the most recent analysis shows that solving (3.4) recovers M correctly, with high probability, if

$$|E| \geq Cn\,\mu\,r\,(\log n)^2\;. \tag{3.7}$$

For comparison, assume that there is no noise in the samples. Then the bound in Theorem 2.6.2 guarantees exact recovery with OptSpace when

$$|E| \geq Cn\,\mu\,r\,\kappa^2(\log n + \mu\,r\,\kappa^4)\;.$$

This improves over (3.7), when we have $\mu r = O((\log n)^2)$ and bounded 'condition number': $\kappa \equiv (\Sigma_{\max}/\Sigma_{\min}) = O(1)$. Note that in many practical applications, such as positioning or structure-from-motion [24, 97], r is known and is small (indeed it is comparable with the ambient dimension that is 3). Further, comparing with the lower bound in (3.2), both nuclear norm minimization and OptSpace achieve near-optimal performance up to polylogarithmic factors. Theorem 2.6.2 is also suboptimal in the following regimes:

(i) Large rank. The necessary number of samples for exact recovery scale linearly in the rank rather than quadratically as in (3.2). As should be clear from Figure 2.4 (and from more extensive simulations in Section 3.5) this appears to be a weakness of our proof technique rather than of the algorithm.

(ii) Large 'condition number' $\kappa = (\Sigma_{\max}/\Sigma_{\min})$. Indeed our bound depends explicitly on this quantity, while this is not the case in (3.7). This appears to be indeed a limitation of the singular value decomposition step. However, in Section 3.6.2, we discusses a simple modification of OptSpace, and shows empirically that we can overcome this problem.

3.2 Noisy scenario

For a more general setting where we have noisy samples, we start with an example of an oracle estimator introduced by Candès and Plan and follow closely the analysis of the estimation error in [19].

Assume that we have an oracle giving us information about U, where U is the orthonormal basis spanning the column space of M. With this precious information, the oracle estimator finds an estimation by projecting the observations onto the subspace spanned by U or equivalently by solving the following optimization problem. For simplicity assume M to be a square matrix of dimensions $n \times n$. Let $\mathcal{P}_U = UU^T$ denote the self-adjoint projection operator onto the subspace spanned by U.

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|\mathcal{P}_E(X-N)\|_F^2 \\ \text{subject to} & X \in T_U \ , \end{array}$$

where $T_U = \{UY^T \,|\, Y \in \mathbb{R}^{n\times r}\}$ is a rn dimensional subspace in $\mathbb{R}^{n\times n}$. Recall that $\mathcal{P}_E$ is the self-adjoint sampling operator defined in (2.3), and let $T_E \equiv \{X \,|\, X_{ij} = 0 \text{ for } (i,j) \notin E\}$ be a $|E|$ dimensional subspace in $\mathbb{R}^{n\times n}$. Then the least-squares solution to the above problem is given by $\widehat{M}_U = (\mathcal{A}^*\mathcal{A})^{-1}\mathcal{A}^*(N^E)$, where

$$\begin{array}{rcl} \mathcal{A} : \mathbb{R}^{n\times n} & \to & T_E \ , \\ X & \mapsto & \mathcal{P}_E\mathcal{P}_U(X) \ , \end{array}$$

and $\mathcal{A}^*$ is the adjoint of the linear operator $\mathcal{A}$. Under the hypotheses of Theorem 2.6.2, $\mathcal{A}^*\mathcal{A} = \mathcal{P}_U\mathcal{P}_E\mathcal{P}_U$ is invertible and the inverse is well defined. Since, $(\mathcal{A}^*\mathcal{A})^{-1}\mathcal{A}^*(N^E) = M + (\mathcal{A}^*\mathcal{A})^{-1}\mathcal{A}^*(Z)$,

$$\|M - \widehat{M}_U\|_F = \|(\mathcal{A}^*\mathcal{A})^{-1}\mathcal{A}^*(Z)\|_F \ .$$

In the special case when we have i.i.d. Gaussian noise with variance σ_Z^2, the expectation $\mathbb{E}[\|M - \widehat{M}_U\|_F^2] = \sigma_Z^2 \mathrm{Tr}((\mathcal{A}^*\mathcal{A})^{-1})$. Under the hypotheses of Theorem 2.6.2, all the rn eigenvalues of $\mathcal{A}^*\mathcal{A}$ concentrate around $|E|/n^2$. This fact is proved later in this section. Therefore, $\mathrm{Tr}((\mathcal{A}^*\mathcal{A})^{-1}) \simeq rn^3/|E|$. Then,

$$\mathbb{E}[\|M - \widehat{M}_U\|_F^2] \simeq \sigma_Z^2 \frac{n^3 r}{|E|} ,$$

where $\simeq$ indicates that the left-hand side concentrates around the right-hand side.

Now consider another oracle estimator with information on $T_{U,V} = \{UY^T + XV^T \,|\, X \in \mathbb{R}^{n\times r}, Y \in \mathbb{R}^{n\times r}\}$, which is a $2nr - r^2$ dimensional subspace in $\mathbb{R}^{n\times n}$. Here, $V \in \mathbb{R}^{n\times r}$ is the orthonormal basis for the space spanned by the rows of M. Again with this precious information, the oracle estimator finds an estimation $\widehat{M}_{\text{oracle}} = (\widetilde{\mathcal{A}}^*\widetilde{\mathcal{A}})^{-1}\widetilde{\mathcal{A}}^*(N^E)$, where

$$\begin{aligned} \widetilde{\mathcal{A}} : \mathbb{R}^{n\times n} &\to T_E \,, \\ X &\mapsto \mathcal{P}_E\mathcal{P}_{U,V}(X) \,, \end{aligned}$$

and $\mathcal{P}_{U,V}(A) = UU^TA + AVV^T - UU^TAVV^T$ is the projection onto $T_{U,V}$.

Under the hypotheses of Theorem 2.6.2, $\widetilde{\mathcal{A}}^*\widetilde{\mathcal{A}} = \mathcal{P}_{U,V}\mathcal{P}_E\mathcal{P}_{U,V}$ is invertible, and $2nr - r^2$ eigenvalues of $\widetilde{\mathcal{A}}^*\widetilde{\mathcal{A}}$ concentrates around $|E|/n^2$, whence $\mathrm{Tr}((\widetilde{\mathcal{A}}^*\widetilde{\mathcal{A}})^{-1}) \simeq (2nr - r^2)n^2/|E|$ [20, Theorem 4.1]. Notice that, by definition, $\lambda_{\max}(\mathcal{A}^*\mathcal{A}) \le \lambda_{\max}(\widetilde{\mathcal{A}}^*\widetilde{\mathcal{A}})$ and $\lambda_{\min}(\mathcal{A}^*\mathcal{A}) \ge \lambda_{\min}(\widetilde{\mathcal{A}}^*\widetilde{\mathcal{A}})$. This confirms the previous claim that the eigenvalues of $\mathcal{A}^*\mathcal{A}$ also concentrate around $|E|/n^2$.

In the special case when we have i.i.d. Gaussian noise with variance σ_Z^2, following the similar analysis, we have

$$\mathbb{E}[\|M - \widehat{M}_{\text{oracle}}\|_F]^2 \simeq \sigma_Z^2 \frac{2n^3 r}{|E|} , \tag{3.8}$$

for $r \ll n$ where $\simeq$ indicates that the left-hand side concentrates around the right-hand side. Notice that $\widehat{M}_{\text{oracle}}$ is highly likely to have rank $2r$, and the information on r, if available, is not being used.

A lower bound on the estimation error with i.i.d. Gaussian noise is proved using minimax argument in [74]. Consider a family of matrices

$$\mathcal{M}(r,C) = \Big\{ M \in \mathbb{R}^{n\times n} \,\Big|\, \text{rank}(M) = r,\ \text{incoherent with } \mu = C\sqrt{\log n} \Big\}\ ,$$

corresponding to rank-r matrices with logarithmic incoherence. Using an information theoretic argument, it is proved that there exists universal constants $C, C' > 0$ such that

$$\inf_{\widehat{M}} \sup_{M\in\mathcal{M}(r,C)} \mathbb{E}[\|M - \widehat{M}\|_F^2] \geq C'\sigma_Z^2 \frac{n^3 r}{|E|}\ , \tag{3.9}$$

where the infimum is taken over all estimators $\widehat{M}$ that are measureable functions of the $|E|$ samples. For better readability, we presented the result with a slightly modified version of $\mathcal{M}(r,C)$ from the original version in [74]. However, the same lower bound holds in this case, since the modified set includes the original one.

The error achieved by the nuclear norm minimization in the noisy case was analyzed by Candès and Plan in [19]. Note that the hard constraint in (3.4) must be relaxed when dealing with noisy samples, which gives

$$\begin{aligned} \text{minimize} \quad & \|X\|_* \\ \text{subject to} \quad & \|\mathcal{P}_E(X - N)\|_F \leq \delta\ , \end{aligned} \tag{3.10}$$

where δ is the estimated noise power and $\|X\|_*$ denotes the nuclear norm of X, i.e the sum of its singular values.. They show that when $|E| \geq Cn\,\mu\,r\,(\log n)^2$, the solution of the nuclear norm minimization above achieves

$$\frac{1}{\sqrt{mn}}\|M - \widehat{M}_{\text{SDP}}\|_F \leq 7\sqrt{\frac{n}{|E|}}\|Z^E\|_F + \frac{2}{n\sqrt{\alpha}}\|Z^E\|_F\ . \tag{3.11}$$

In [19], the constant in front of the first term is slightly smaller than 7, but in any case larger than $4\sqrt{2}$.

Theorem 2.6.2 improves over this result in several respects: (i) We do not have the second term on the right-hand side of (3.11), that actually increases with the number of observed entries; (ii) Our error decreases as $n/|E|$ rather than $(n/|E|)^{1/2}$;

(*iii*) The noise enters Theorem 2.6.2 through the operator norm $\|Z^E\|_2$ instead of its Frobenius norm $\|Z^E\|_F \geq \|Z^E\|_2$. For E uniformly random, one expects $\|Z^E\|_F$ to be roughly of order $\|Z^E\|_2\sqrt{n}$. For instance, within the independent entries model with bounded variance σ_Z, $\|Z^E\|_F = \Theta(\sigma_Z\sqrt{|E|})$ while $\|Z^E\|_2$ is of order $\sigma_Z\sqrt{|E|/n}$ (up to logarithmic terms).

For the sake of comparison, suppose we have i.i.d. Gaussian noise with variance σ_Z^2, and recall that M is a rank-r matrix of dimension $m \times n$. In this case the oracle estimator yields (for $r = o(n)$)

$$\frac{1}{\sqrt{mn}}\,\|M - \widehat{M}_{\text{oracle}}\|_F \simeq \sigma_Z\sqrt{\frac{(m+n)r}{|E|}}\,.$$

The bound (3.11) on the semidefinite programming approach yields

$$\frac{1}{\sqrt{mn}}\,\|M - \widehat{M}_{\text{SDP}}\|_F \leq \sigma_Z\left(7\sqrt{n} + 2\sqrt{\frac{|E|}{mn}}\right).$$

More recently, Negahban and Wainwright in [74] analyzed (3.10) with a 'spikiness' condition, instead of the incoherence condition, and showed that, with high probability, the solution to (3.10) achieves

$$\frac{1}{\sqrt{mn}}\,\|M - \widehat{M}_{\text{SDP}}\|_F \leq \sigma_Z\,\tilde{\mu}\sqrt{\frac{Crn\log n}{|E|}}\,,$$

where the spikiness $\tilde{\mu} \equiv \sqrt{mn}M_{\max}/\|M\|_F$ and assuming $\sigma_z^2 \geq 1/(mn)$. The spikiness $\tilde{\mu}$ is not too large for incoherent matrices since, due to the incoherence condition **A1**, we have $\tilde{\mu} \leq \mu$.

Finally, using Theorems 2.6.2 and 2.6.3 we deduce that OptSpace achieves

$$\frac{1}{\sqrt{mn}}\,\|M - \widehat{M}_{\text{OptSpace}}\|_F \leq \sigma_Z\sqrt{\frac{C\,nr}{|E|}}\,,$$

where the constant C depends on $\alpha = m/n$ and $\kappa = \sigma_1(M)/\sigma_r(M)$. When α and κ are bounded and we have i.i.d. Gaussian noise with small enough σ_Z, we have matching lower and upper bound up to a constant and OptSpace provably achieves order-optimal performance.

3.3 Sampling-based low-rank approximation

Low-rank approximation of matrices plays an important role in numerical analysis, data mining, control and a number of other areas. Singular Value Decomposition (SVD) is widely used to compute the optimal low-rank approximation minimizing the mean squared error. However, the computational complexity of computing SVD of a dense matrix with dimensions $n \times n$ is $O(n^3)$ which, in large-scale applications, becomes infeasible. Particular applications of interest include spectral clustering [75, 41] applied to image segmentation, manifold learning [95, 105], and low-rank approximation of kernel matrices [114, 118]. To segment an image with 1000×1000 pixels requires computing SVD of a dense $10^6 \times 10^6$ affinity matrix. Learning a high-dimensional manifold with 41368 samples (e.g., the CMU PIE face dataset [96]) requires computing SVD of a dense 41368×41368 matrix. In these large-scale applications with dense matrices, SVD is computationally infeasible. Even storing the matrix is not possible, since, for instance, the affinity matrix of a 1000×1000 image requires storing a 8 TB matrix. But, these applications, and many more interesting ones, require only a few, say r, top singular values and the corresponding singular vectors. Iterative methods, such as power iteration, still suffer from superlinear complexity requiring $O(n^2 r)$ operations.

A popular strategy is to compute, instead of the optimal low-rank approximation, a near-optimal approximation. To this end, sampling based methods provide powerful alternatives. Computation is performed on a small number of entries, hence reducing the number of operations. Further, it is not necessary to store the full matrix, requiring less storage. In this section, we review the recent developments on sampling-based algorithms, and show that our main result in Theorem 2.6.1, along with the novel trimming step introduced in Section 2.3, improves over the state-of-the-art algorithms.

Sparsification based approaches are efficient ways to compute near optimal low-rank approximations. Given an input matrix M, we can first sparsify it to get a sparse matrix S, then use orthogonal iteration to compute S_r, the best rank-r approximation of S. If $(M-S)$ has weak spectral features, then S_r will be a good rank-r approximation of M. Sparsifying M speeds up the matrix vector multiplication in orthogonal iterations, while reducing the memory required to store the input data.

The operator norm $\|M-S\|_2$ bounds how much we lose from sparsification. For simplicity, assume M to be an $n \times n$ rank-r matrix, although all the arguments can be easily generalized to a full-rank matrix M. Then, the approximation error, in terms the Frobenius norm, is bounded by $\|M-S_r\|_F \leq 2\sqrt{2r}\|M-S\|_2$ [4, Lemma 1.1]. The main challenge is to find a matrix S that is sparse and has small operator norm of the difference matrix $\|M-S\|_2$. Perhaps surprisingly, a simple uniform sampling generates a good sparsification of M. Further, adaptive non-uniform sampling can reduce the approximation error.

Under the uniform sampling

$$S_{ij} = \begin{cases} \frac{1}{p}M_{ij} & \text{with probability } p \ , \\ 0 & \text{with probability } 1-p \ . \end{cases}$$

A direct application of a recent result in random matrix theory [110], an improvement over the ideas of Ahlswede and Winter [5], gives the following bound on $\|M-S\|_2$. An analogous bound was proved in [20, Theorem 6.3], which is only weaker by a constant factor. We refer to the end of this section for the proof.

Remark 3.3.1. *Assuming* $p \geq (\log n)/n$, *with probability larger than* $1-1/n^3$,

$$\|M-S\|_2 \leq 5M_{\max}\sqrt{\frac{n\log n}{p}} \ ,$$

where $M_{\max} \equiv \max_{i,j}|M_{ij}|$.

Achlioptas and McSherry in [4] proved a tighter bound by eliminating the logarithmic factor but holding only for $p \geq (8\log n)^4/n$. Namely, they show that, with probability larger than $1 - \exp\big\{ - 19(\log n)^4\big\}$,

$$\|M - S\|_2 \le 4M_{\max}\sqrt{\frac{n}{p}}\ . \tag{3.12}$$

Our trimming-plus-SVD approach improves over both of these results by eliminating the logarithmic factor in the error bound as well as conditions on p. As explained in detail in Section 2.4, 'trimming' is crucial especially when p is small. Let $\widetilde{S}$ be the sparsification obtained from a random sampling followed by trimming. Theorem 2.6.1, and in particular Lemma 2.7.1, implies that, with probability larger than $1 - 1/n^3$, there exists a constant C such that

$$\|M - \widetilde{S}\|_2 \le CM_{\max}\sqrt{\frac{n}{p}}\ .$$

We get the same bound as (3.12) up to a constant without the extra condition on p.

Sparsification need not be bound to uniform sampling. We can get sparser matrices (hence reducing the number of operations) without increasing the error bound, if we allow non-uniform sampling. This is proved in [4] with adaptive sampling, sampling entries with probability depending on the magnitude of the entries. Let pn^2 be, in a similar way, the average number of samples under the adaptive sampling with

$$S_{ij} = \begin{cases} \frac{1}{p_{i,j}}M_{ij} & \text{with probability } p_{i,j}\ , \\ 0 & \text{with probability } 1 - p_{i,j}\ , \end{cases} \tag{3.13}$$

where $p_{i,j} = \max\big\{\, c(M_{ij}/M_{\max})^2\,,\ \sqrt{c(M_{ij}/M_{\max})^2(8\log n)^4/n}\,\big\}$. Then, it is proved in [4] that with probability larger than $1 - \exp\{-19(\log n)^4\}$,

$$\|M - S\|_2 \le 4\sqrt{2}\|M\|_F\sqrt{\frac{1}{pn}}\ . \tag{3.14}$$

The constant in the right-hand side is in fact slightly smaller than $4\sqrt{2}$ in [4], but in any case larger than 4. For comparison with (3.12), notice that $1 \le (\|M\|_F/M_{\max}) \le n$, and, intuitively, $\|M\|_F/M_{\max}$ is larger for an incoherent matrix M than for a coherent M (We refer to Section 2.2 for the definition of incoherence). Therefore, the gain in the performance guarantees with adaptive sampling is at best constant when M is incoherent, but significant when M is coherent.

While searching for a fast algorithm for semidefinite programming, Arora et al. [6] developed a modified adaptive sampling scheme with a comparable error bound. Again, let pn^2 be the average number of samples under the adaptive sampling defined in (3.13) with $p_{i,j} = \min\{1\,,\, c\sqrt{n}|M_{ij}|\}$. Using the discretization technique from [40] and the Chernoff-Hoeffding bound, it is proved in [6] that there exists constants C and C' such that, with probability larger than $1 - \exp\{-C'n\}$,

$$\|M - S\|_2 \leq \frac{C\sum_{i,j}|M_{ij}|}{pn^{3/2}}\ .$$

Compared to (3.14) this gives a tighter error bound when $\sum_{i,j}|M_{ij}| \lesssim \sqrt{n}\|M\|_F$.

In conclusion, finding the best sparsification is a very difficult problem. We need to solve

$$\begin{aligned} \text{minimize}\ \ & \|M - S\|_2 \\ \text{subject to}\ \ & \|S\|_{\ell_0} \leq pn\ , \end{aligned}$$

where the ℓ_0 norm $\|\cdot\|_{\ell_0}$ of a matrix denotes the number of non-zero entries. As we saw in this section, sampling based algorithms can be thought of as efficient heuristics for solving this problem, but several questions remain open. It is an interesting research direction to understand the fundamental limit on how good an approximation we can get with sparsification and to devise efficient heuristics.

In an alternative approach, there is a copious line of work on sampling rows and columns of M, instead of sampling entries, for fast low-rank approximation. Popular column-sampling algorithms include fast Monte Carlo algorithms, known as ConstantTimeSVD or LinearTimeSVD [31, 43, 34], Nyström-based method [114], and CUR decomposition method [33]. The relationship between these algorithms are presented in [35] in detail.

Proof of Remark 3.3.1. The remark follows from a direct application of the generalization of Bernstein's inequality to sum of independent random matrices. Let $X_{i,j} \equiv (S_{ij} - M_{ij})e_i e_j^*$ be a random matrix with a single non-zero entry in position (i,j), such that $S - M = \sum_{i,j} X_{i,j}$. Note that for all (i,j), $\mathbb{E}[X_{i,j}] = 0$ and $\|X_{i,j}\|_2 \leq M_{\max}/p$. Since $\mathbb{E}[X_{i,j}^2] = ((1-p)/p)M_{\max}^2$, the total variance is bounded by $\|\sum_{i,j}\mathbb{E}[X_{i,j}^2]\|_2 \leq nM_{\max}^2/p$. Theorem 2.10 in [110] can be easily extended to non-symmetric random matrices $X_{i,j}$ to give

$$\mathbb{P}\left(\Big\|\sum_{i,j} X_{i,j}\Big\|_2 \geq aM_{\max}\right) \leq 2n\exp\left\{\frac{-a^2}{\frac{2n}{p} + \frac{2a}{3p}}\right\} .$$

Assuming $p \geq \log n/n$, set $a = 5\sqrt{n\log n/p}$ to prove the desired claim. □

3.4 Efficient algorithms

One challenge in solving nuclear norm minimization (3.4) using SDP is that the computational complexity, although polynomial in n, is still quite large: of the order $O(n^4)$. This must be compared with the most complex component of OptSpace which is the SVD in step 2 with complexity $O(|E|r\log n)$. More recently, a number of innovative and efficient algorithms were introduced to solve variations of the nuclear norm minimization.

3.4.1 Compressed sensing based algorithms

We review recent developments in efficient algorithms for matrix completion, mostly based on compressed sensing approaches. Numerical simulation results from a few of these algorithms are compared in Section 3.5.

Cai, Candès and Shen [17] proposed an efficient first-order procedure called Singular Value Thresholding (SVT) to solve the nuclear norm minimization in (3.4) directly. Trimming-plus-SVD in OptSpace is akin to a single step of this procedure, with the important novelty of the trimming step that improves its performance significantly.

One problem with (3.4) is that it does not generalize to the case when we have noise in the samples. The hard constraint in (3.4) can be relaxed, resulting in the problem

$$\text{minimize} \quad \lambda\|X\|_* + \frac{1}{2}\|\mathcal{P}_E(X-N)\|_F^2 \ . \tag{3.15}$$

Here, $\mathcal{P}_E(N)$ is the sampled matrix with noise. A number of efficient first-order procedures for solving (3.15) were recently developed. Fixed Point Continuation with Approximate SVD (FPCA) [64] is based on the Bregman iterative regularization algorithm [117]. Accelerated Proximal Gradient (APG) algorithm [108] is based on the fast iterative shrinkage-thresholding algorithm [8]. Soft-Impute and Hard-Impute iteratively replaces the missing entries with those obtained from thresholded SVD [66].

When we have a good estimate of the rank r, the matrix completion problem can be recast into the following rank-r approximation problem.

$$\begin{aligned} \text{minimize} \quad & \|\mathcal{P}_E(X-N)\|_F \\ \text{subject to} \quad & \text{rank}(X) \leq r \ , \end{aligned} \tag{3.16}$$

To find an approximate solution to (3.16), Lee and Bresler introduced Atomic Decomposition for Minimum Rank Approximation (ADMiRA) [62] inspired by CoSaMP [73]. Singular Value Projection (SVP) [67], is a different approach to solving (3.16) based on the Iterative Hard Thresholding (IHT) algorithm [13].

Another popular approach is to minimize (3.17) over X and Y alternatively. This procedure is known as the alternate minimization method or the nonlinear (block) Gauss-Seidel (GS) scheme or the block coordinate descent method.

$$\text{minimize} \quad \frac{1}{2}\|\mathcal{P}_E(XY^T-N)\|_F^2 \ , \tag{3.17}$$

where $X \in \mathbb{R}^{m\times r}$ and $Y \in \mathbb{R}^{n\times r}$ and r is an estimated target rank. Alternate minimization methods for the matrix completion problem have been studied in [10, 59, 49, 116]. Subspace Evolution and Transfer (SET) [28] is a manifold optimization approach to (3.17) with an extra condition that X lies on the Grassmann manifold. This is similar to the last step of OptSpace, but with a novel transfer step which seeks to overcome getting blocked by 'barriers' in the objective function.

In [113], a slightly modified problem is solved with the Low-rank Matrix Fitting algorithm (LMaFit).

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|XY^T - Z\|_F^2 \;, \\ \text{subject to} \quad & \mathcal{P}_E(Z) = \mathcal{P}_E(N) \;, \end{aligned} \tag{3.18}$$

where $X \in \mathbb{R}^{m\times r}$, $Y \in \mathbb{R}^{n\times r}$ and $Z \in \mathbb{R}^{m\times n}$ are the variables we minimize over. The hope here is that it is much faster to solve (3.18) compared to (3.4). However, the non-convexity in the model may prevent one from getting a global solution. Wen et al. [113] provide convincing evidence showing that LMaFit is empirically as reliable as other nuclear norm minimization approaches.

3.4.2 Collaborative filtering algorithms

Local optimization techniques such as gradient descent methods or coordinate descent methods have been intensively used in collaborative filtering applications to solve (3.17) and its variants. Weighted Low Rank Approximation (WLRA), introduced in an early work by Srebro and Jaakkola [101], uses gradient descent methods to minimize a slightly modified objective function $F(X) = \min_Y \|\mathcal{P}_E(XY^T) - \mathcal{P}_E(N)\|_F^2$.

The use of such an objective function was justified in [84] using a probabilistic model. Let N_{ij} represent the rating of user i for movie j. Also, let $X \in \mathbb{R}^{m\times r}$ and $Y \in \mathbb{R}^{n\times r}$ be user and movie features with the column vectors X_i and Y_j representing the features of the user i and movie j. Assume that the observed ratings are scalar product of the feature vectors corrupted by additive i.i.d. Gaussian noise such that $N_{ij} = X_i^T Y_j + Z_{ij}$, where Z_{ij}'s are i.i.d. Gaussian noise distributed as $\mathcal{N}(0, \sigma_Z^2)$. The user and movie features are also assumed to have i.i.d. zero-mean Gaussian priors, namely $X_{ij} \sim \mathcal{N}(0, \sigma^2)$ and $Y_{ij} \sim \mathcal{N}(0, \sigma^2)$. Under this model, the (negative) log-posterior naturally leads to the use of quadratic regularization in the factors.

$$\text{minimize} \quad \frac{1}{2}\|\mathcal{P}_E(XY^T - N)\|_F^2 + \lambda\|X\|_F^2 + \lambda\|Y\|_F^2 \,. \tag{3.19}$$

Again, gradient descent in the factors was used to perform the optimization. Also, [84] introduced a logistic mapping between the low-rank matrix and the recorded ratings. Recently, this line of work was pushed further by Salakhutdinov and Srebro in [86] using a non-uniform quadratic regularization in the factors. Based on the observation that in real datasets the samples are often drawn according to non-uniform distributions, they suggest weighting the nuclear norm by the frequency of rows and columns. Then, the objective function is

$$\text{minimize} \quad \frac{1}{2}\|\mathcal{P}_E(XY^T - N)\|_F^2 + \sum_{i=1}^{m} \lambda p_i \|X_i\|^2 + \sum_{j=1}^{n} \lambda q_j \|Y_j\|^2 \,,$$

where p_i is the number of samples in row i and q_j is the number of samples in column j. A version of stochastic gradient descent is used to optimize this objective function.

The relationship between the non-convex objective function (3.19) and convex relaxation in (3.15) was further investigated in [102, 82, 81]. The basic relation is provided by the identity

$$\|A\|_* = \frac{1}{2} \min_{A=XY^T} \left\{ \|X\|_F^2 + \|Y\|_F^2 \right\} , \tag{3.20}$$

where $\|A\|_*$ denotes the nuclear norm of the matrix A. In other words, adding a regularization term that is quadratic in the factors (which is used in much of the literature reviewed in this section) is equivalent to adding a nuclear norm regularization of A. In view of the identity (3.20) it might be possible to use the results in Section 2.6 to prove stronger guarantees on the nuclear norm minimization approach. Unfortunately this implication is not immediate. Indeed in OptSpace we assume that the correct rank r is known, while on the other hand we do not use a quadratic regularization in the factors. In Section 3.6.1 we present a procedure that estimates the rank from the data and is provably successful under the hypotheses of Theorem 2.6.2 assuming noiseless samples. Trying to establish such an implication, and clarifying the relation between the two approaches is nevertheless a promising research direction.

In collaborative filtering applications, matrix completion was also studied from a graphical models perspective in [85], which introduced an approach to prediction based on Restricted Boltzmann Machines (RBM). Exact learning of the model parameters is intractable for such models. The authors studied the performance of *contrastive divergence*, which uses an approximate gradient of the likelihood function for local optimization. Based on empirical evidence, it was argued that RBM's have several advantages over spectral methods for collaborative filtering. More recently, [7] used a graphical model to characterize the probability distribution underlying the collaborative filtering dataset. A message passing algorithm, dubbed IMP, is introduced to infer the underlying distribution from the observed entries.

3.5 Numerical comparisons

In this section, we apply OptSpace to both synthetic and real data to compare its performance to that of the other competing algorithms. First, we present numerical comparisons using randomly generated matrices. Then, we present results on real data from publicly available collaborative filtering datasets [3, 2, 1].

We implemented OptSpace in C and MATLAB, and the code is publicly available[1]. The comparisons were performed using a 3.4 GHz Desktop computer with 4 GB RAM with the C implementation. For efficient singular value decomposition of sparse matrices, we used a modification of SVDLIBC[2], which is based on SVDPACKC.

3.5.1 Synthetic datasets

First, we consider the noiseless scenario and compare the rate of exact reconstructions. Random matrices are generated as $M = UV^T$. Here, U and V are $n \times r$ matrices where each entry is i.i.d. with standard Gaussian distribution $\mathcal{N}(0, 1)$, unless specified otherwise. Then, each entry is revealed independently with probability $p = \epsilon/n$, so

[1]available at http://www.stanford.edu/∼raghuram/optspace

[2]available at http://tedlab.mit.edu/∼dr/SVDLIBC/

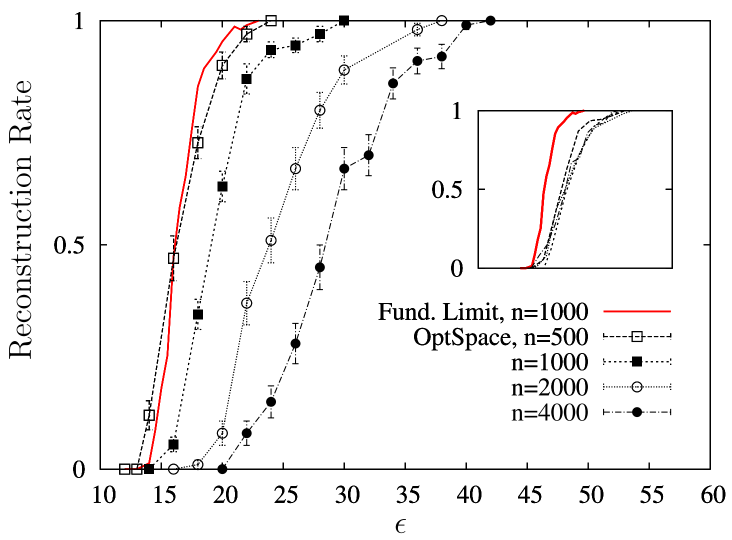


Figure 3.2: Reconstruction rates of rank-4 matrices using OptSpace. The solid curve is the fundamental limit proved in [98]. In the inset, the same data are plotted vs. $|E|/(n(\log n)^2)$

that on average ϵn entries are revealed. Numerical results, and also the analysis, show that there is no notable difference if we choose the revealed set of entries E uniformly at random given $|E| = n\epsilon$.

Exact matrix completion

We first study the *reconstruction rate*, the fraction of instances for which the matrix is reconstructed correctly. We declare a matrix to be reconstructed if $\|M - \widehat{M}\|_F / \|M\|_F \leq 10^{-4}$. Figure 3.2 shows the reconstruction rate of OptSpace as a function of $\epsilon = |E|/n$ for fixed $r = 4$. As predicted by Theorem 2.6.2, the reconstruction rate is close to one when $|E| \geq C\mu n \log n$. Note that the incoherence parameter μ scales like $\log n$ in this example. Figure 3.3 illustrates how the reconstruction rate of OptSpace compares with those of the competing algorithms: SVT[3], FPCA[4], and ADMiRA described in Section 3.4 [17, 64, 62]. Different algorithms have different thresholds, the value of ϵ where the reconstruction rate has a sharp transition. For

[3]available at http://svt.caltech.edu
[4]available at http://www.columbia.edu/~sm2756/FPCA.htm

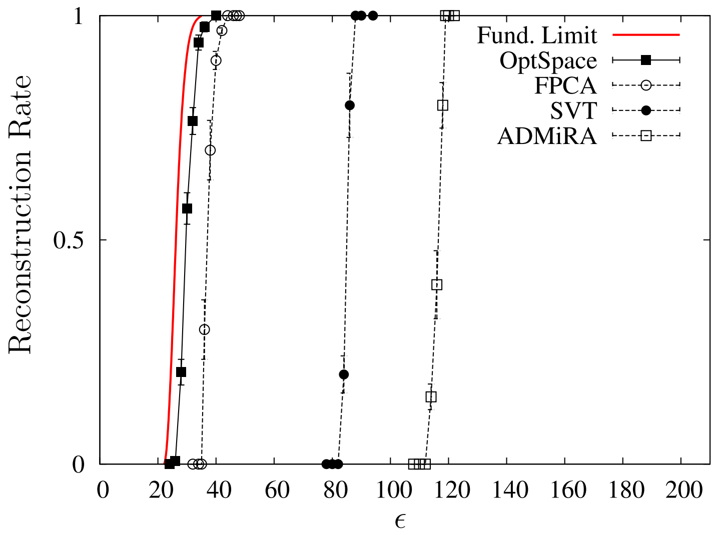


Figure 3.3: Reconstruction rates of rank-10 matrices of dimension 1000×1000. The leftmost solid curve is the fundamental limit proved in [98].

$r = 10$ and $n = 1000$, the figure shows that OptSpace has a lower threshold compared to the other competing algorithms, and this is consistent for various values of n and r as illustrated in the table below. Furthermore, the threshold of OptSpace is very close to the bound proved in [98], below which no algorithm can correctly recover M.

In the following Tables 3.1 and 3.2, we further compare the resulting relative root mean squared error $\|M - \widehat{M}\|_F / \|M\|_F$ of different algorithms. Table 3.1 presents results for smaller values of ϵ and hence for *hard* problems, whereas Table 3.2 is for larger values of $\epsilon = |E|/n$ which are relatively *easy*. Note that the values of ϵ used in Table 3.1 all correspond to $|E| \leq 2.6(2nr - r^2)$, which is slightly larger than the degrees of freedom. We ran into *Out of Memory* problems for the FPCA algorithm for $n \geq 20000$ and hence the results are omitted. On the easy problems, all the algorithms achieves similar performances, whereas on the hard problems, OptSpace outperforms other algorithms on most of instances.

Matrix completion from noisy entries

In the noisy scenario, M is generated as above and samples are corrupted by added noise so that $N_{ij} = M_{ij} + Z_{ij}$. Again, each entry is revealed independently with

n	r	ϵ	OptSpace		SVT		FPCA		ADMiRA	
			rel. error	time(s)	rel. err.	time(s)	rel. err.	time(s)	rel. err.	time(s)
1000	10	50	1.95×10^{-5}	33	3.42×10^{-1}	734	6.04×10^{-4}	65	4.41×10^{-1}	8
	50	200	1.28×10^{-5}	235	2.54×10^{-1}	11769	1.07×10^{-5}	83	3.54×10^{-1}	141
	100	400	9.22×10^{-6}	837	7.99×10^{-2}	27276	3.86×10^{-6}	165	1.28×10^{-1}	1767
5000	10	50	7.27×10^{-5}	338	5.34×10^{-1}	476	9.99×10^{-1}	1776	5.13×10^{-1}	77
	50	200	1.47×10^{-5}	1930	4.87×10^{-1}	36022	2.17×10^{-2}	2757	5.36×10^{-1}	358
	100	400	1.38×10^{-5}	6794	4.12×10^{-1}	249330	2.49×10^{-5}	3942	4.84×10^{-1}	36266
10000	10	50	1.91×10^{-5}	725	6.33×10^{-1}	647	9.99×10^{-1}	9947	6.19×10^{-1}	129
	50	200	5.02×10^{-6}	3032	5.50×10^{-1}	18558	9.97×10^{-1}	14048	5.79×10^{-1}	11278
	100	400	1.33×10^{-5}	18928	4.84×10^{-1}	169578	8.59×10^{-3}	18448	5.30×10^{-1}	67880
20000	10	50	1.95×10^{-2}	2589	7.30×10^{-1}	1475	–	–	7.20×10^{-1}	286
	50	200	1.49×10^{-5}	10364	6.30×10^{-1}	14588	–	–	6.04×10^{-1}	29323
30000	10	50	1.62×10^{-2}	5767	7.74×10^{-1}	2437	–	–	7.43×10^{-1}	308

Table 3.1: Relative errors on *hard* problem instances without noise.

n	r	ϵ	OptSpace		SVT		FPCA		ADMiRA	
			rel. error	time(s)	rel. err.	time(s)	rel. err.	time(s)	rel. err.	time(s)
1000	10	120	1.18×10^{-5}	28	1.68×10^{-5}	40	5.20×10^{-5}	18	9.09×10^{-4}	52
	50	390	9.26×10^{-6}	212	1.62×10^{-5}	247	3.53×10^{-6}	106	3.62×10^{-5}	701
	100	570	1.49×10^{-5}	723	1.71×10^{-5}	694	1.92×10^{-6}	160	1.88×10^{-5}	2319
5000	10	120	1.51×10^{-5}	252	1.76×10^{-5}	112	1.69×10^{-4}	1083	4.68×10^{-2}	198
	50	500	1.16×10^{-5}	850	1.62×10^{-5}	1312	5.99×10^{-5}	1005	7.42×10^{-3}	92751
	100	800	8.39×10^{-6}	3714	1.73×10^{-5}	5432	3.32×10^{-5}	1953	4.42×10^{-2}	634028
10000	10	120	7.64×10^{-6}	632	1.75×10^{-5}	221	9.95×10^{-1}	13288	1.22×10^{-1}	442
	50	500	1.19×10^{-5}	2585	1.63×10^{-5}	2872	9.51×10^{-5}	7337	2.58×10^{-2}	186591
	100	800	1.46×10^{-5}	8514	1.76×10^{-5}	10962	6.90×10^{-5}	9426	9.66×10^{-2}	755082
20000	10	120	1.59×10^{-5}	1121	1.76×10^{-5}	461	–	–	3.04×10^{-1}	181
	50	500	9.77×10^{-6}	4473	1.64×10^{-5}	6014	–	–	4.33×10^{-2}	346651
30000	10	120	1.56×10^{-5}	1925	1.80×10^{-5}	838	–	–	4.19×10^{-1}	71

Table 3.2: Relative errors on *easy* problem instances without noise.

a probability ϵ/n. As a performance measure, we use the root mean squared error defined as

$$\text{RMSE} = \frac{1}{n}\left(\sum_{i,j}(M_{ij} - \widehat{M}_{ij})^2\right)^{1/2} .$$

First, in the standard scenario, Z_{ij}'s are distributed as i.i.d. Gaussian random variables. In the following series of examples, we illustrate how the performances change under different noise models. Since ADMiRA and OptSpace require a target rank, we estimate the rank using the algorithm described in Section 3.6.1. For FPCA we set $\mu = \sqrt{2\epsilon}\sigma$, where σ^2 is the variance of the noise. A convincing argument for this choice of μ is given in [19].

Standard noise scenario

We refer to the example used in [19] as the standard noise scenario. In this example, $M = UV^T$ where U and V are $500 \times r$ matrices with each entry being sampled independently from a standard Gaussian distribution $\mathcal{N}(0, 1)$. Further, the noise Z_{ij}'s are i.i.d. standard Gaussian $\mathcal{N}(0, 1)$ unless specified otherwise. We also refer to Section 3.2 for more details on the nuclear norm minimization and the oracle estimator.

Figure 3.4 compares the average RMSE as a function of ϵ, and Figure 3.5 compares the average RMSE as a function of the rank. After a few iterations, Optspace has a smaller root mean square error than the nuclear norm minimization approach (FPCA) for most instances, and in about 10 iterations it becomes indistinguishable from the oracle estimation error in (3.8), which is $\sqrt{(2nr - r^2)/|E|}$.

Figure 3.6 compares the average RMSE as a function of the noise power. The results are presented in terms of the *noise ratio* defined as in [17].

$$\text{NR} \equiv \left(\frac{\mathbb{E}[\|\mathcal{P}_E(Z)\|_F^2]}{\mathbb{E}[\|\mathcal{P}_E(M)\|_F^2]}\right)^{1/2} , \tag{3.21}$$

The RMSE achieved using FPCA does not increase with the noise ratio in the large

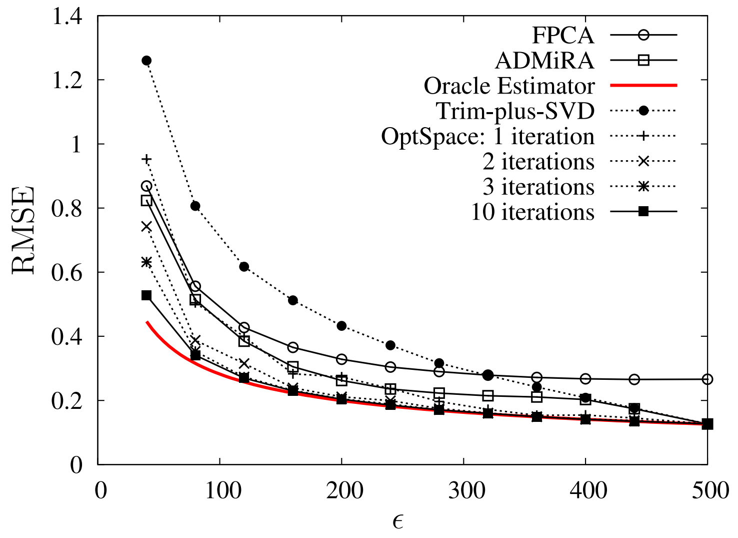


Figure 3.4: RMSE under additive Gaussian noise as a function of ϵ for fixed $n = 500$ and $r = 4$.

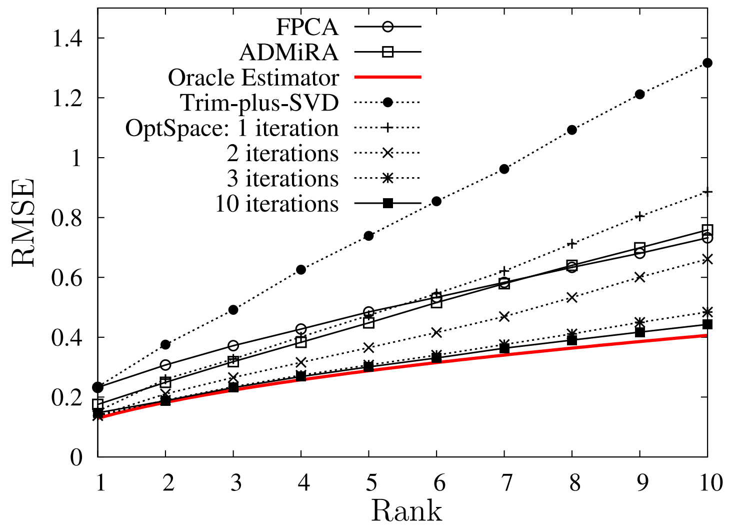


Figure 3.5: RMSE under additive Gaussian noise as a function of r for fixed $n = 500$ and $\epsilon = 120$.

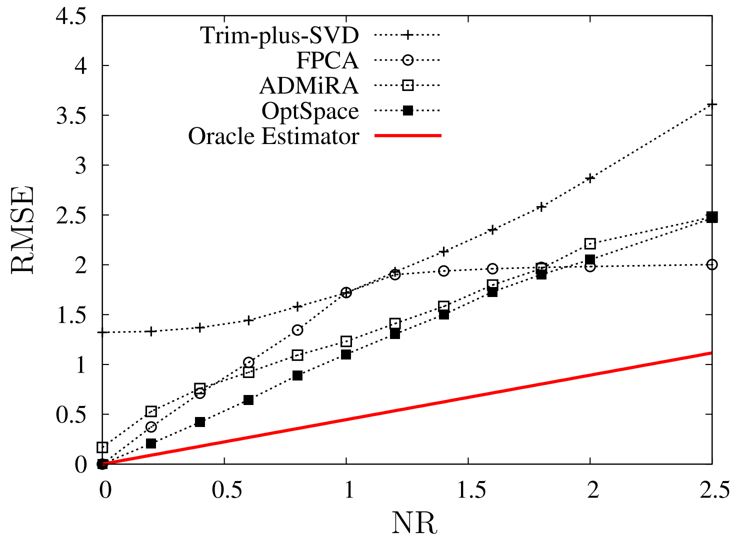


Figure 3.6: RMSE under additive Gaussian noise as a function of the noise ratio NR for fixed $n = 500$, $\epsilon = 40$ and $r = 4$.

noise regime. The reason is not that the estimates are good, but rather the estimation becomes effectively a zero matrix and the resulting RMSE is close to $\sqrt{\mathbb{E}[\|M\|_F^2/n^2]}$, which is 2, regardless of the noise power. Table 3.3 illustrate how the relative error $\|M - \widehat{M}\|_F/\|M\|_F$ changes with respect to the noise power for fixed $n = 1000$.

In the standard scenario, we made the following three assumptions on the noise matrix Z: (i) Z is independent of M; (ii) Z_{ij}'s are mutually independent; (iii) Z_{ij}'s follow the Gaussian distribution. The matrix completion algorithms described in Section 3.4 are designed to be especially effective under this standard scenario for the following two reasons. First, the algorithms minimize the squared error, which is well suited for the Gaussian noise. Second, the independence of Z_{ij}'s ensure that the spectral norm of the noise matrix, which is the key parameter determining the quality of the approximation in Theorem 2.6.2, is not large. In the following, we fix $n = 500$ and $r = 4$, and study how the performance changes when we change the noise model.

NR	r	ϵ	OptSpace		SVT		FPCA		ADMiRA	
			rel. error	time(s)	rel. err.	time(s)	rel. err.	time(s)	rel. err.	time(s)
	10	120	4.47×10^{-3}	24	7.8×10^{-3}	11	5.48×10^{-3}	99	2.01×10^{-2}	35
10^{-2}	50	390	5.49×10^{-3}	149	9.5×10^{-3}	88	7.18×10^{-3}	805	1.83×10^{-2}	391
	100	570	6.39×10^{-3}	489	1.13×10^{-2}	216	1.08×10^{-2}	1111	1.63×10^{-2}	1424
	10	120	4.50×10^{-2}	23	0.72×10^{-1}	4	6.04×10^{-2}	140	1.18×10^{-1}	12
10^{-1}	50	390	5.52×10^{-2}	147	0.89×10^{-1}	33	7.77×10^{-1}	827	1.20×10^{-1}	139
	100	570	6.38×10^{-2}	484	1.01×10^{-1}	85	1.13×10^{-1}	1140	1.15×10^{-1}	572
	10	120	4.86×10^{-1}	31	0.52	1	5.96×10^{-1}	141	5.38×10^{-1}	3
1	50	390	6.33×10^{-1}	153	0.63	8	9.54×10^{-1}	1088	5.92×10^{-1}	47
	100	570	1.68	107	0.69	35	1.19	1582	6.69×10^{-1}	181

Table 3.3: Relative errors under additive Gaussian noise.

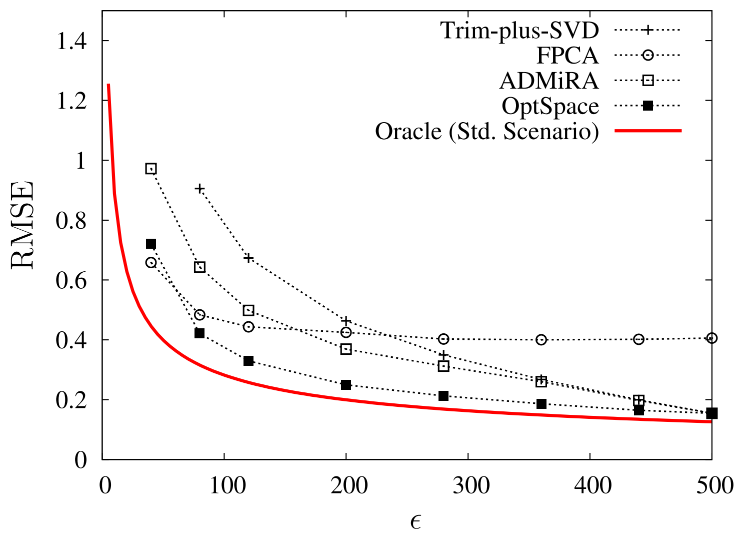
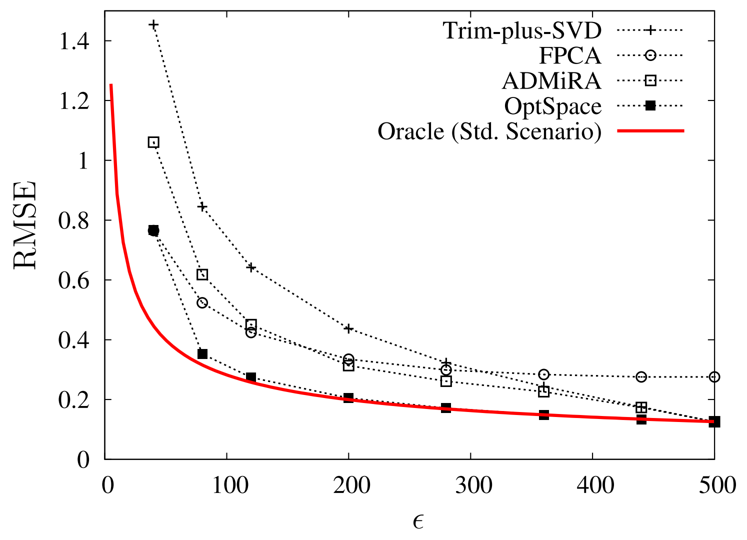


Figure 3.7: RMSE under the multiplicative noise model.

Multiplicative Gaussian noise

In sensor network localization [112], where the entries of M correspond to the squared pairwise distances between sensors, the observation noise is oftentimes modelled as multiplicative Gaussian noise. In formulae, $Z_{ij} = \xi_{ij} M_{ij}$, where ξ_{ij}'s are i.i.d. Gaussian random variables with zero mean. In this example, the variance of ξ_{ij}'s is chosen to be $1/r$ with r denoting the rank of M, so that the resulting noise ratio is $\text{NR} = 1/2$ for consistency with the standard scenario.

Figure 3.7 shows the RMSE with respect to ϵ under multiplicative Gaussian noise. The RMSE of the trimming-plus-SVD for $\epsilon = 40$ is larger than 1.5 and is omitted in the figure. The bottommost line corresponds to the oracle performance under the standard scenario, and is displayed here, and all of the following figures, to serve as a reference for comparison. The error is larger compared to the standard scenario in Figure 3.4. It is more difficult to separate the noise from the original matrix, since the noise is now correlated with M.



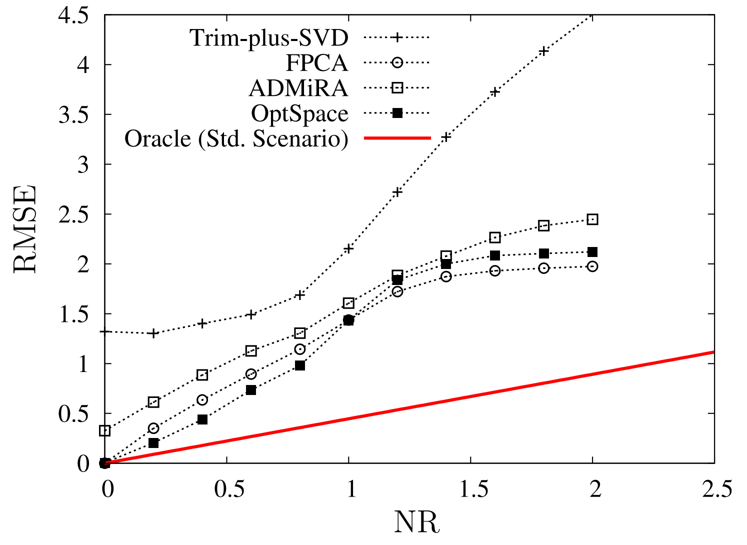


Figure 3.8: RMSE for fixed noise ratio NR $= 1/2$ (top) and for fixed $\epsilon = 40$ (bottom) with outliers as in (3.22).

Outliers

In structure from motion applications [24], each entry of M corresponds to the position of a point of interest in the 2-dimensional images captured by cameras in different angles and locations. However, due to failures in the feature extraction algorithm, some of the observed positions are corrupted by large noise whereas most of the observations are noise free. To account for such outliers, we use the following model.

$$Z_{ij} = \begin{cases} a & \text{with probability } 1/200 \ , \\ -a & \text{w.p. } 1/200 \ , \\ 0 & \text{w.p. } 99/100 \, . \end{cases} \tag{3.22}$$

The value of a is chosen according to the target noise ratio $\text{NR} = a/20$. The noise is i.i.d. but the distribution is non-Gaussian.

Figure 3.8 shows the performance of the algorithms with respect to ϵ and the noise ratio NR with outliers. The performance degrades for non-Gaussian noise when the number of samples is small. This problem of separating sparse noise from low-rank matrices, also known as *robust principal component analysis*, is an interesting research topic and is investigated in a number of recent papers [23, 115, 18, 121].

Quantization noise

Quantization is ubiquitous in practical applications. To model regular quantization, we choose a parameter a and quantize the matrix entries to the nearest value in $\{\ldots, -a/2, a/2, 3a/2, 5a/2, \ldots\}$. The parameter a is chosen such that the resulting noise ratio is $1/2$. Quantization noise is deterministic and completely depends on the matrix entries M_{ij}, whereas in the multiplicative noise model the noise was still random. Hence, the error is expected to be larger with quantization. Figure 3.9 shows the performance against ϵ under the quantization. Compared to Figure 3.7, for the same value of $\text{NR} = 1/2$, quantization is much more detrimental than the multiplicative noise as expected.

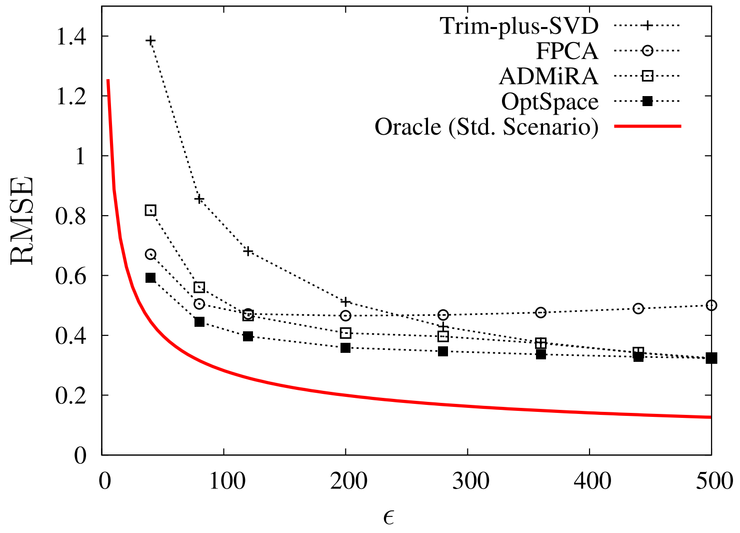


Figure 3.9: RMSE with quantization.

3.5.2 Real datasets

In this section, we compare the error on collaborative filtering datasets: the Jester joke dataset [1] and the Movielens dataset [2].

The Jester joke dataset contains 4.1×10^6 ratings for 100 jokes from 73,421 users [5]. Since the number of users is large compared to the number of jokes, we select $n_u \in \{100, 1000, 2000, 4000\}$ users at random for comparison purposes. We follow the examples used in [64]. We randomly choose two ratings for each user as a test set T, and we compare the Normalized Mean Absolute Error (NMAE) in this test set. The NMAE is commonly used in collaborative filtering as an performance metric, and in this section we provide the estimation NMAE so that it is easy to compare with results using other approaches. The Mean Absolute Error (MAE) is defined as in [64, 45].

$$\text{MAE} = \frac{1}{|T|} \sum_{(u,i)\in T} |M_{ui} - \widehat{M}_{ui}| \ ,$$

[5]The dataset is available at http://eigentaste.berkeley.edu/dataset/

where M_{ui} is the original rating and $\widehat{M}_{ui}$ is the predicted rating for user u and item i. The Normalized Mean Absolute Error (NMAE) is defined as

$$\mathrm{NMAE} = \frac{\mathrm{MAE}}{M_{\max} - M_{\min}} \,,$$

where $M_{\max}$ and $M_{\min}$ are upper and lower bounds for the ratings. Jester joke ratings are in $[-10, 10]$ so that $M_{\max} - M_{\min} = 20$.

In this section, we use an extension of OptSpace which we call Incremental Optspace. Incremental Optspace iteratively performs OptSpace, using the previous estimation as a initial guess for the next iteration and incrementing the target rank by one in each iteration. While this requires more computations, it gives a better estimation when the original matrix to be reconstructed is ill-conditioned. The description of the algorithm and more numerical simulation results are presented in Section 3.6.2.

The numerical results for the Jester joke dataset are presented in the first four rows of Table 3.4. In the table, rank indicates the rank used for estimation. To get an idea of how good the predictions are, consider the case where each missing entry is predicted with a random number drawn uniformly at random in $[-10, 10]$ and the actual rating is also a random number with same distribution. After a simple computation, we can see that the resulting NMAE of the random prediction is 0.333. If we estimate each missing entry with zero, the resulting NMAE is 0.25. As another comparison, for the same dataset with $n_u = 18000$, a nearest neighbor algorithm and Eigentaste both yield NMAE of 0.187 [45].

The NMAE of Incremental Optspace is lower than these algorithms even for $n_u = 100$ and tends to decrease with n_u. Numerical simulation results on the Movielens dataset are shown in the last row. The dataset contains $100,000$ ratings for $1,682$ movies from 943 users.[6] We use $80,000$ randomly chosen ratings to estimate the $20,000$ ratings in the test set, which is called *u1.base* and *u1.test*, respectively, in the movielens dataset.

Next, to get some insight on the structure of the real data, we compute the

[6]The dataset is available at http://www.grouplens.org/node/73

n_u	m	samples	rank	Incremental OptSpace		FPCA		ADMiRA	
				NMAE	time(s)	NMAE	time(s)	NMAE	time(s)
100	100	7484	2	0.17674	0.1	0.20386	25	0.18194	0.3
1000	100	73626	9	0.15835	11	0.16114	111	0.16194	0.5
2000	100	146700	9	0.15747	26	0.16101	243	0.16286	0.9
4000	100	290473	9	0.15918	56	0.16291	512	0.16317	2
943	1682	80000	10	0.18638	213	0.19018	753	0.24276	5

Table 3.4: Numerical results on the Jester joke dataset with the number of jokes m fixed at 100 (top four rows), and for the Movielens dataset with 943 users and 1682 movies (last row).

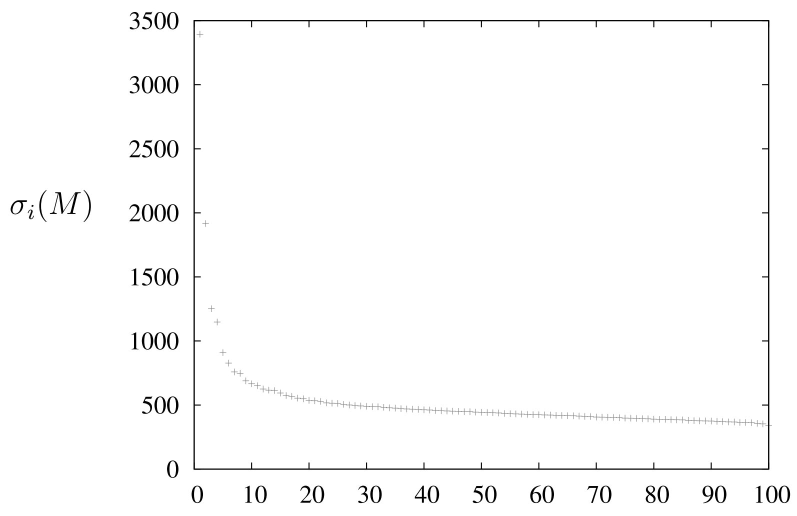


Figure 3.10: The singular values of the complete sub matrix in the Jester joke dataset in the decreasing order.

spectrum of a complete sub matrix where all the entries are known. With the Jester joke dataset, we delete all users containing missing entries, and generate a complete matrix M with $14,116$ users and 100 jokes. The empirical distribution of the singular values of M is shown in Figure 3.10. The first two singular values are dominant, but the distribution has a heavy tail. Computing $\sum_{i=1}^{n}(\sigma_i(M)/\sigma_1(M))$, which can be thought of as a *soft* rank, gives 15.5. This heavy tail in the singular value distribution is one of the difficulties in dealing with collaborative filtering datasets. The other aspect is that the samples are not drawn uniformly at random as commonly assumed in matrix completion. A typical user tends to watch and rate the movie she prefers.

Finally, we test the incoherence assumption for the Netflix dataset [3] in Figure 3.11. For the definition of the incoherence property we refer to Section 2.2. To check if the first condition **A0** holds with small μ for the Netflix movie ratings matrix, we run OptSpace on the Netflix dataset and plot the cumulative sum of the sorted row norms of the left and right factors defined as follows. Let the output of OptSpace be $\widehat{M} = XSY^T$, where $X \in \mathbb{R}^{m\times r}$ and $Y \in \mathbb{R}^{n\times r}$ are orthonormal. The number of users m is equal to $480,189$ and the number of movies n is $17,770$. For the target rank we use $r = 5$. Let $x_i = (m/r)\|X^{(i)}\|^2$ and $y_i = (n/r)\|Y^{(i)}\|^2$ denote the rescaled norm of the ith rows of X and Y respectively. Define a permutation $\pi_l : [m] \to [m]$ which sorts x_i's in a non-decreasing order such that $x_{\pi_l(1)} \le x_{\pi_l(2)} \le \ldots \le x_{\pi_l(m)}$. We use the standard combinatorics notation $[k] = \{1, 2, \ldots, k\}$ for an integer k. Similarly, we can define $\pi_r : [n] \to [n]$ for y_i's.

In Figure 3.11, we plot $\sum_{i=1}^{k} x_{\pi_l(i)}$ and $\sum_{i=1}^{k} y_{\pi_r(i)}$ against k. For comparison, we also plot the corresponding results for a randomly generated matrix X_G. Generate $U \in \mathbb{R}^{m\times r}$ by sampling its entries U_{ij} independently and distributed as $\mathcal{N}(0,1)$ and let X_G be the orthonormal basis spanning the column space of U. For a given matrix, if **A0** holds with a small μ close to one then the corresponding curve would be close to a straight line. The curvature in the plots is indicative of the disparity among the row weights of the factors. In this example, the randomly generated matrix is incoherent with $\mu = 7.2$, whereas the Netflix dataset is incoherent with $\mu = 25.8$. We can say that a randomly generated matrix is more incoherent than the Netflix dataset.

3.6 Extensions

There are two potential drawbacks in using OptSpace in practice. OptSpace requires a priori knowledge of the target rank, and in practice given rank r it is sensitive to the condition number, i.e., $\sigma_1(M)/\sigma_r(M)$. We provide algorithmic solutions, which make OptSpace more robust for practical use.

3.6.1 Rank estimation

Although in many applications such as positioning [97] or structure-from-motion [24] the target rank is known in advance, in other applications like collaborative filtering [44, 100], we need a good estimate of the rank. We propose a very simple scheme to estimate the target rank r from observation M^E with provable performance guarantees.

Let σ_i be the ith largest singular value of the trimmed matrix $\widetilde{M}^E$ and define $\epsilon \equiv |E|/\sqrt{mn}$. 'Trimming' is explained in Section 2.3. Then, the following cost

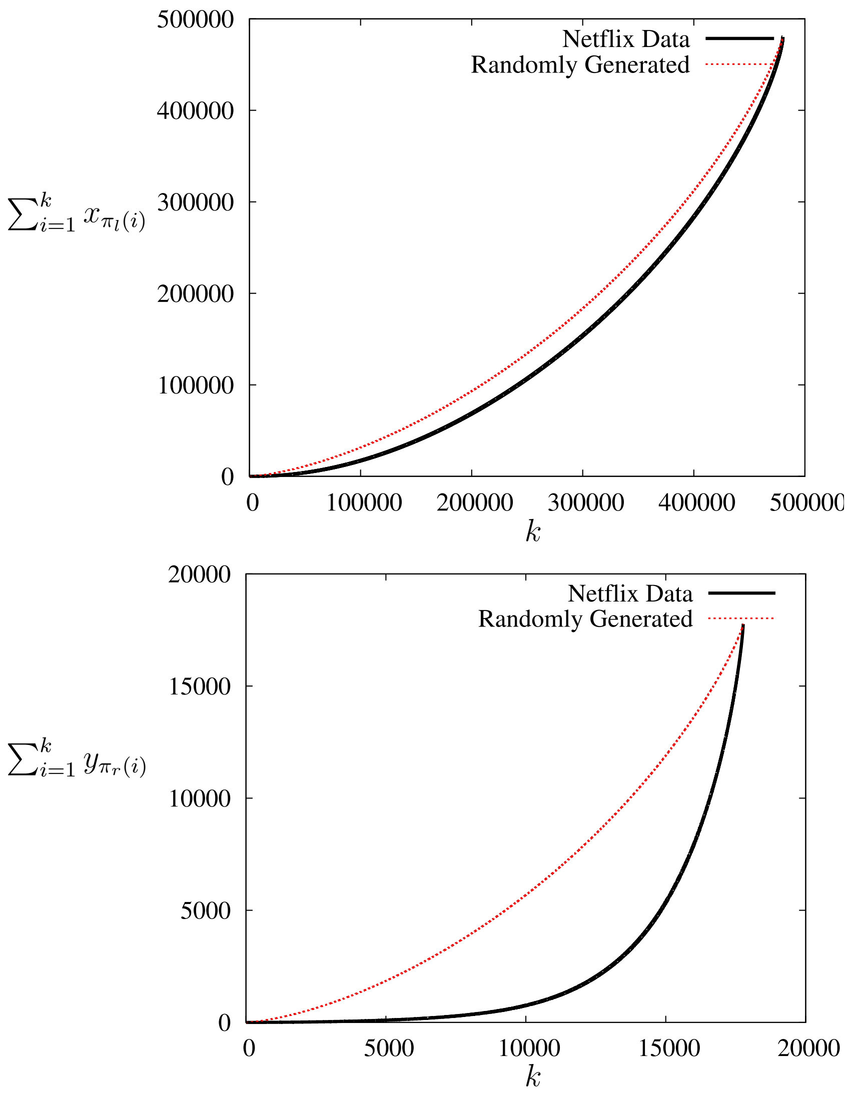


Figure 3.11: Cumulative sum of the sorted row norms of the factor corresponding to users (top) and movies (bottom).

function is defined in terms of the singular values.

$$R(i) = \frac{\sigma_{i+1} + \sigma_1\sqrt{i/\epsilon}}{\sigma_i} \ .$$

Based on the above definition, Rank Estimation consists of two steps:

```
RANK ESTIMATION
Input: trimmed observation matrix $\widetilde{M}^E$
Output: estimated rank $\hat{r}$
1:  Compute singular values $\{\sigma_i\}$ of $\widetilde{M}^E$;
2:  Find the index $i$ that minimizes $R(i)$, and let $\hat{r}$ be the minimizer.
```

The idea is that, if enough entries of M are revealed then there is a clear separation between the first r singular values, which reveal the structure of the matrix M to be reconstructed, and the spurious ones. This follows from Lemma 2.7.1 and is illustrated in Figure 2.1. The following remark shows that this simple procedure is guaranteed to reconstruct the correct rank r, with high probability, when $|E|$ is large enough. In particular, in the noiseless setting, we are guaranteed to recover the correct rank when the hypotheses in Theorem 2.6.2 are satisfied. Hence, we combine this algorithm with OptSpace to get the same guarantees without any prior knowledge of the rank r. In practice, we do not need to compute all $\min\{m, n\}$ singular values of $\widetilde{M}^E$, since we often have a bound on the maximum target rank we would like to use.

Remark 3.6.1. *Assume M to be a rank-r $m \times n$ matrix and $\kappa = O(1)$ for $\kappa = \sigma_1(M)/\sigma_r(M)$, and E is uniformly random given its size $|E|$. Then there exists a constant $C(\kappa, \alpha)$ such that, if $|E| > C(\kappa, \alpha)nr$, then* Rank Estimation *correctly estimates the rank r from $\widetilde{M}^E$, with high probability.*

Proof. Let Σ_i be the ith largest singular value of the matrix M. From Section 2.7.2 we know that there exists a numerical constant $C > 0$ such that, with high probability

$$\left|\frac{\sigma_q}{\epsilon} - \frac{\Sigma_q}{\sqrt{mn}}\right| \le CM_{\max}\left(\frac{\alpha}{\epsilon}\right)^{1/2} \ ,$$

where it is understood that $\Sigma_q = 0$ for $q > r$, and $M_{\max} = \max_{i,j}\{|M_{ij}|\}$.

Applying this result to the cost function $R(i)$, we get the following bounds.

$$\begin{aligned}
R(r) &\leq \frac{CM_{\max}\sqrt{\alpha\epsilon} + \left(\frac{\Sigma_1}{\sqrt{mn}} + CM_{\max}\sqrt{\frac{\alpha}{\epsilon}}\right)\sqrt{r\epsilon}}{\frac{\epsilon\Sigma_r}{\sqrt{mn}} - CM_{\max}\sqrt{\alpha\epsilon}} \;, \\
R(i) &\geq \frac{\frac{\epsilon\Sigma_{i+1}}{\sqrt{mn}} - CM_{\max}\sqrt{\alpha\epsilon}}{\frac{\epsilon\Sigma_i}{\sqrt{mn}} + CM_{\max}\sqrt{\alpha\epsilon}} \;, && \forall\, i < r \;, \\
R(i) &\geq \frac{\left(\frac{\Sigma_1}{\sqrt{mn}} - CM_{\max}\sqrt{\frac{\alpha}{\epsilon}}\right)\sqrt{r\epsilon}}{CM_{\max}\sqrt{\alpha\epsilon}} \;, && \forall\, i > r \;.
\end{aligned}$$

Let, $\kappa = \Sigma_1/\Sigma_r$ and $\xi = (M_{\max}\sqrt{\alpha})/(\Sigma_1\sqrt{r})$. After some calculus, we get that for

$$\epsilon > C\, r\, \max\left\{\, \xi^2 \,,\, \xi^4\kappa^2 \,,\, \kappa^4 \,\right\} \;,$$

we have the desired inequality: $R(r) < R(i)$ for all $i \neq r$. This proves the remark. $\Box$

3.6.2 Incremental OptSpace

In this section, we introduce a modification to the OptSpace algorithm, which has substantially better performance in the case when the matrix M to be reconstructed is ill-conditioned. The condition number of a rank-r matrix M is defined as $\kappa \equiv \sigma_1(M)/\sigma_r(M)$, where $\sigma_i(M)$ is the ith singular value. When M has high condition number, then the simple trimming-plus-SVD step is unable to track the relatively smaller singular values. As a result, the greedy gradient descent iterations get stuck in a local minima which only captures the structure corresponding to the largest singular value.

One heuristic to compensate for this lost signal is to iteratively infer the structure corresponding to the largest singular values and subtract their effect from the observations. This allows for those weaker features to stand out. We start by first finding the first singular value and the corresponding singular vectors, and incrementally search for the next ones.

```
Incremental OptSpace
Input: observation matrix $M^E$, observed set $E$, $\rho_{\max}$
Output: estimation $\widehat{M}$
1:  Trim $M^E$, and let $\widetilde{M}^E$ be the output;
2:  Set $\widehat{M}^{(0)} = 0$;
3:  For $\rho = 1, \dots, \rho_{\max}$ do:
4:    Compute the largest singular vectors $(u, v)$ of $\widetilde{M}^E - \widehat{M}^{(\rho-1)}$;
5:    Set $X_0^{(\rho)}$ and $Y_0^{(\rho)}$ to be the orthonormal basis
        spanning $[X^{(\rho-1)}; v;]$ and $[Y^{(\rho-1)}; u;]$;
6:    Minimize $\widetilde{F}(X, Y)$ through Manifold Optimization
        with initial condition $(X_0^{(\rho)}, Y_0^{(\rho)})$;
7:    Set $\widehat{M}^{(\rho)} = X^{(\rho)} S^{(\rho)} Y^{(\rho)T}$;
8:  End for
9:  Return $\widehat{M}^{(\rho)}$.
```

Figure 3.12 illustrates how the RMSE defined as $(1/n)\|M - \widehat{M}\|_F$ degrades when the matrix M is ill-conditioned. We generate as $M = \sqrt{4/166}\, U \operatorname{diag}([1, 4, 7, 10])\, V^T$, where U and V have i.i.d. Gaussian $\mathcal{N}(0, 1)$ entries. The resulting matrix has $\Sigma_1/\Sigma_4 \simeq 10$, and the normalization constant $\sqrt{4/166}$ is chosen such that $\mathbb{E}[\|M\|_F^2]$ is the same as in other examples in Section 3.5. Incremental OptSpace achieves a smaller error for most values of ϵ compared to other competing algorithms including OptSpace. In Table 3.5, we also compare the relative error $\|M - \widehat{M}\|_F / \|M\|_F$ for different values of the condition number κ in the noiseless scenario. Table 3.5 shows that Incremental OptSpace improves significantly over OptSpace and achieves performance comparable to those of the other algorithms.

3.7 Discussions and open questions

Let us consider the noiseless setting for the sake of simplicity, since the same arguments hold in the noisy case as well. There are two important ingredients in proving our main result in Theorem 2.6.2: (i) Simple trimming-plus-SVD step can produce a

κ	r	OptSpace		Inc. OptSpace		SVT		FPCA		ADMiRA	
		rel. error	time	rel. error	time(s)	rel. err.	time(s)	rel. err.	time(s)	rel. err.	time(s)
	10	8.56×10^{-6}	20	8.66×10^{-6}	19	1.70×10^{-5}	55	5.32×10^{-5}	22	1.57×10^{-5}	242
1	50	1.16×10^{-5}	78	1.09×10^{-5}	832	1.64×10^{-5}	628	2.97×10^{-6}	115	1.60×10^{-5}	1252
	100	7.05×10^{-6}	401	7.37×10^{-6}	4605	1.78×10^{-5}	2574	1.88×10^{-6}	174	1.67×10^{-5}	3454
	10	1.08×10^{-1}	124	1.53×10^{-5}	70	1.53×10^{-5}	72	5.53×10^{-5}	21	1.56×10^{-5}	234
5	50	1.10×10^{-1}	1591	1.30×10^{-5}	921	1.46×10^{-5}	639	1.08×10^{-5}	145	1.61×10^{-5}	1221
	100	1.24×10^{-1}	5004	1.41×10^{-5}	5863	1.54×10^{-5}	1541	4.38×10^{-6}	664	1.68×10^{-5}	3450
	10	1.09×10^{-1}	112	2.00×10^{-1}	238	1.47×10^{-5}	127	5.22×10^{-5}	21	1.55×10^{-5}	243
10	50	1.04×10^{-1}	1410	1.32×10^{-5}	1593	1.36×10^{-5}	1018	1.42×10^{-5}	270	1.61×10^{-5}	1206
	100	1.10×10^{-1}	4569	1.36×10^{-5}	9550	1.41×10^{-5}	2473	4.54×10^{-6}	996	1.65×10^{-5}	3426

Table 3.5: Numerical results for different condition numbers κ. ϵ depends only on r and is the same as used in Table 3.3.

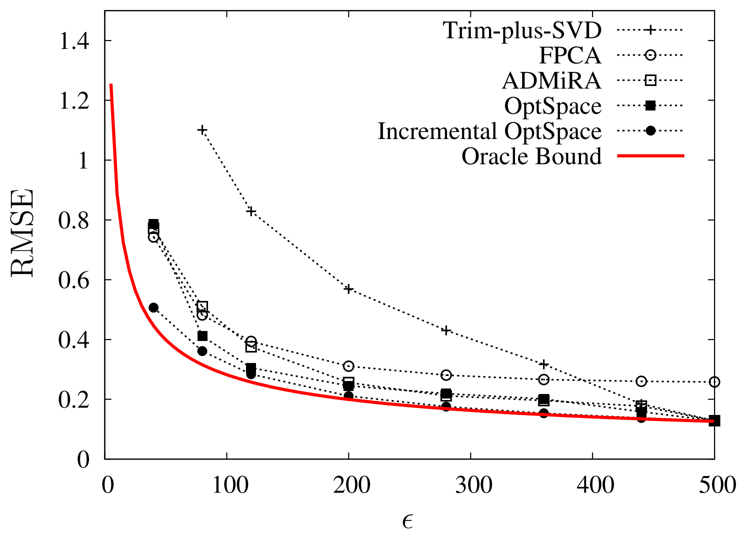


Figure 3.12: RMSE with ill-conditioned matrices and i.i.d. Gaussian noise.

good initial estimate; (ii) The objective function $F(\cdot)$ defined in (2.1) that we want to minimize is well-behaved in a neighborhood close to the global minimizer M. Thus, once we have a good initial estimate, greedy minimization converges to the correct solution M.

The objective function $F(\cdot)$ is a non-convex function with many local minima. Hence, a greedy minimization scheme gets stuck in a local minimum of $F(\cdot)$ and the performance varies depending on where we start the greedy minimization. Therefore analyzing such a non-convex optimization problem is a priori a difficult task. However, once we are guaranteed an initial estimate close to the global minimizer M, greedy minimization converges to the correct solution M, and we can prove strong performance guarantees. This idea of combining a simple initial estimation step and a following greedy optimization step could be applied to other problems with non-convex objective functions beyond matrix completion. And the techniques used in our work for analyzing such a two-step algorithm could also be applied to other problems of interest.

In the following, we discuss open problems in matrix completion which could be interesting future research directions.

Characterizing computation-accuracy tradeoff

Finding a low-rank approximation of a large data matrix is a common computational task in many important applications including spectral clustering [75, 41], manifold learning [95, 105], and low-rank approximation of kernel matrices [114, 118]. It is known that the SVD of a data matrix provides the best low-rank approximation. Let the SVD of the data matrix be $M = \sum_i \sigma_i u_i v_i^T$, where σ_i's are the singular values and u_i's and v_i's are the left and right singular vectors. The singular values are ordered in a decreasing order, i.e., $\sigma_1 \geq \sigma_2 \geq \cdots > 0$. Then, the low-rank approximation based on SVD is $L = \sum_{i=1}^r \sigma_i u_i v_i^T$, and L has the minimum approximation error $\sum_{i,j}(M_{ij} - L_{ij})^2$ among all matrices of rank r.

Traditional schemes to compute the leading singular values and the singular vectors involve applying the data matrix multiple times, which can be time consuming when the data matrix is large and dense. In Section 3.3, we discussed the possibility of using matrix completion schemes as efficient alternatives for finding low-rank approximations. Matrix completion schemes deal with a sparsified version of the data matrix, thus reducing the number of operations significantly, at the cost of increased approximation error. The trade-off between the number of samples $|E|$ and the approximation error $\sum_{i,j}(M_{ij} - \widehat{M}_{ij})^2$ achieved using OptSpace is characterized in Theorem 2.6.2. However, we do not have a bound on the total number of computations necessary to produce an estimate achieving the performance guaranteed in the theorem. Namely, we don't have a bound on how many iterations in the gradient descent step we need.

Let $\widehat{M}^{(t)}$ be the low-rank estimation we get after t iterations of the gradient descent step. If we can characterize the approximation error $\sum_{i,j}(M_{ij} - \widehat{M}^{(t)}_{ij})^2$ in terms of the sample size $|E|$ and the number of iterations t, then we can customize OptSpace for fast low-rank approximation. For instance, consider an example where we are given a target accuracy and we want to produce an estimate within that target accuracy with small number of computations. If we can characterize the computation-accuracy tradeoff of OptSpace, we can use it to choose the number of samples to use and the number of gradient descent iterations to perform.

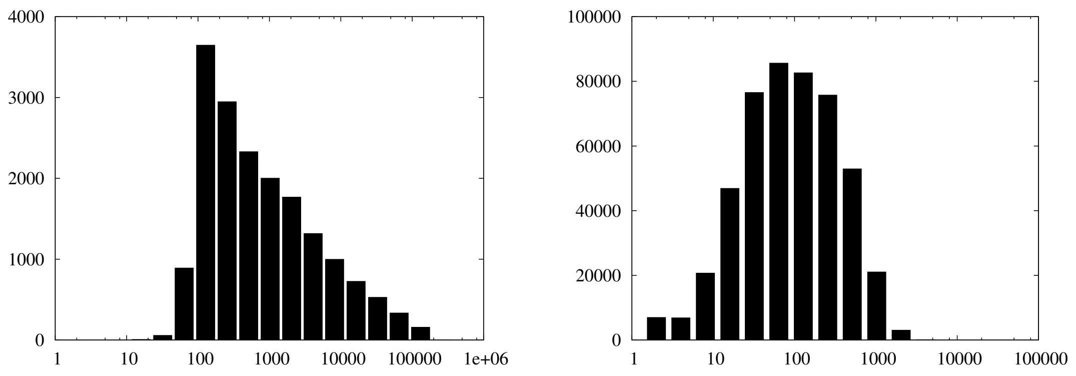


Figure 3.13: Histogram of number of ratings per movie (left) and per user (right) in the Netflix prize dataset [3]. Average number of samples per movie is about $5,575$ and per user is about 206. Maximum number of samples per movie is $227,715$ and per user is $17,651$.

General sampling model

The model we study assumes the set of sampled entries E to be uniformly random. Under this model, we expect to see binomial distribution of the number of samples per row/column. However, in practical applications like collaborative filtering, we observe heavy-tailed distributions indicating that the sample set E is far from uniformly random. This is apparent in the Netflix prize dataset [3] (Figure 3.13). Under the uniform sampling model, the probability of seeing a movie with $227,715$ ratings is close to zero. Recently, Salakhutdinov and Srebro in [86] introduced a generalization of nuclear norm regularization to deal with collaborative filtering datasets with non-uniform sampling. The idea is to use weighted nuclear norm regularization with weights determined by the number of samples in each row/column. It is a practically important challenge to extend OptSpace, perhaps using a similar idea as in [86], when dealing with datasets under non-uniform sampling.

General low-dimensional model

The low-rank model is used in many applications to capture the important information about data in matrix form. However, in practical applications like collaborative filtering, the data matrix might not be well approximated by a low-rank matrix. This is illustrated in Figure 3.10 that shows the distribution of singular values of Jester joke dataset [1].

Although a low-rank matrix might not be a good model for such datasets, the data might still be determined by low-dimensional features. Let $u_i \in \mathbb{R}^r$ be a low-dimensional vector representing customer i and $v_j \in \mathbb{R}^r$ be a low-dimensional vector representing movie j. Then, there might be a function $h : \mathbb{R}^r \times \mathbb{R}^r \to \mathbb{R}$ that explains the ratings, namely $h(u_i, v_j) = M_{ij}$. It is an interesting problem to find such a function $h(\cdot)$ for datasets of interest and develop schemes to estimate the feature vectors from a small number of entries $\{M_{ij}\}_{(i,j)\in E}$.

Chapter 4

Positioning

This chapter treats the problem of positioning wireless devices from pairwise distance measurements. We first show that such a positioning problem can be viewed as a matrix completion problem (Section 4.1). However, direct application of matrix completion algorithms fails in common positioning settings. We first discuss the challenges in applying matrix completion approaches to positioning problems. Then, as a first step to overcome these challenges, we focus on understanding how existing positioning algorithms work. To this end, the main result of this chapter provides performance guarantees of two popular positioning algorithms. In Section 4.2, we analyze a centralized positioning algorithm known as MDS-MAP introduced in [93]. In a more practical setting, we might not have a central infrastructure and only local communication might be available. In section 4.3, therefore, we assume a decentralized setting, and analyze a decentralized algorithm called Hop-TERRAIN introduced in [87]. This chapter is based on joint work with Karbasi and Montanari [78, 54].

4.1 Introduction

The problem of determining the physical locations of wireless devices has recently gained much interest, mainly due to the increasing number of applications from wireless ad-hoc sensor networks to location based services for mobile devices. Localization plays an important role in many applications, when a large number of wireless devices, which we call here nodes, are randomly and densely placed in a region. The geographic information can be used in communication protocols for routing in wireless ad-hoc networks [60]. There are abundant applications for location-based services such as personal navigation [91], provided that real-time and low-cost indoor positioning is possible using accessible mobile devices like smart phones.

Although Global Positioning System (GPS) has been widely used for positioning, it adds considerable cost to the system. Further, it does not work in indoor environments. As an alternative, we want to use basic local information such as connectivity (which nodes are within communication range of each other) or local distances (pairwise distances between neighboring nodes).

Two common techniques for obtaining the local distance and/or connectivity information are Received Signal Strength Indicator (RSSI) and Time Difference of Arrival (TDoA). RSSI is a measurement of the ratio of the power present in a received radio signal and a reference power. Signal power at the receiving end is inversely proportional to the square of the distance between the receiver and the transmitter. Hence, RSSI could potentially be used to estimate the distance and it is common to assume the use of RSSI in distance measurements. However, experimental results indicate that the accuracy of RSSI is limited [37]. TDoA technique uses the time difference between the receipt of two different signals with different velocities, for instance ultrasound and radio frequency signals [79, 89]. The time difference is proportional to the distance between the receiver and the transmitter, and given the velocity of the signals the distance can be estimated from the time difference. These techniques can be used, independently or together, for distance estimation. In an alternative approach, Angle of Arrival (AoA) can also be used to infer the positions of sensors [76]. Once a node has angle of arrival information to three other nodes with known positions, we can perform triangulation to locate the wireless node. To measure the angle of arrival, an antenna array is required at each wireless node.

In the following section, we first discuss centralized positioning algorithms which use the connectivity information or the local distance measurements. Then, we discuss positioning algorithms which can be used in the decentralized setting.

4.1.1 Centralized positioning

In the centralized setting, we assume that there is a central processor with access to all the distance measurements. A centralized algorithm is used to estimate the positions of the nodes from the pairwise distance measurements. Note that we can only determine the positions up to a rigid transformation (rotation and translation), since pairwise distances are invariant under a rigid transformation. Further, due to signal attenuation, we assume that only distance measurements between nodes within some radio range R are available. The corresponding network is modeled as a fixed radius random geometric graph. In some cases, when we have special nodes with known positions in some global coordinate system, we can hope to uniquely determine the node positions. These special nodes are called *anchors*, landmarks [76] or seeds [83].

In Section 4.2, we are interested in the theoretical guarantees associated with the performance of centralized positioning algorithms. Such analytical bounds on the performance of localization algorithms can provide answers to practical questions: for example, how large should the radio range be in order to get the error within a threshold? With this motivation in mind, we provide a bound on the performance of the popular MDS-MAP [93] algorithm when applied to centralized positioning using only the connectivity information.

In the case when all pairwise distances are known, the positions can be exactly determined, up to a rigid motion, using a classical method known as multidimensional scaling (MDS) [16, 26]. We refer to Section 4.2.2 for further details. However, when most pairwise distances are missing, the problem is much more challenging. One approach is to first estimate the missing distance measurements and then apply MDS to the estimated distance measurements [106, 107, 93, 94]. MDS-MAP [93] uses the shortest paths between all pairs of nodes to approximate the missing distance measurements. In Section 4.2, we describe the MDS-MAP algorithm in detail, and provide a performance guarantee for this algorithm.

Various approaches have been proposed to solve the problem of positioning from partial distance measurements with noise. One line of work deals with solving a convex relaxation to find the node positions using semidefinite programming (SDP)

[12, 112]. When the distances are measured exactly without any noise, then SDP guarantees exact reconstruction of the node positions if the underlying graph is universally rigid [99]. However, when the graph is only globally rigid or when there is a small noise in the measurements, there is no performance guarantee. Another line of work deals with 'stitching' together local structures, including Locally Rigid Embedding (LRE) [97], eigenvector synchronization of locally rigid subgraphs [27], and As-rigid-As-Possible algorithm [119]. The performances of these practical algorithms are measured through simulations and there is little work on the theoretical analysis to support the results.

Dineas et al. used a matrix completion technique to reconstruct the distance matrix and prove an upper bound on the resulting estimation error [32]. However, the analysis is based on a number of strong assumptions. First, it is assumed that even far away sensors have non-zero probability of detecting their distances. Second, the algorithm explicitly requires the knowledge of detection probabilities between all pairs. Third, it is assumed that the average degree of the network (i.e., the average number of nodes detected by each node) grows linearly with the number of nodes in the network. In the following section, we further discuss how positioning can be viewed as a matrix completion problem and what the main challenges are.

4.1.2 Positioning as matrix completion

As will be explained in Section 4.2.2, if we have the complete and exact matrix of squared distances $D \in \mathbb{R}^{n\times n}$, we can use Multidimensional Scaling (MDS) to determine the exact positions of the nodes, up to a rigid motion. We define $D_{i,j} \equiv \|x_i - x_j\|^2$, where $x_i \in \mathbb{R}^d$ is the d-dimensional position of node $i \in [n]$. It is, therefore, enough to determine D in order to solve the positioning problem. Further, matrix D has low rank. By definition,

$$D = a\mathbb{1}_n^T + \mathbb{1}_n a^T - 2XX^T \ ,$$

where $a \in \mathbb{R}^n$ is a vector with $a_i = \|x_i\|^2$, $\mathbb{1}_n \in \mathbb{R}^n$ is an all-ones vector, and $X \in \mathbb{R}^{n\times d}$ is the position matrix with i-th row equal to x_i. D is a sum of two rank-1 matrices and a rank-d matrix. Hence, the rank of D is at most $d+2$, which is much smaller than n.

We can view positioning as a matrix completion problem, where we want to find a low-rank matrix D from a small number of sampled entries corrupted by noise. However, matrix completion algorithms cannot be directly applied to the problem of positioning, because the assumptions made in the matrix completion setting are not always satisfied. We made two main assumptions: (i) The low-rank matrix has small incoherence parameter μ; (ii) The entries are sampled uniformly at random given the sample size $|E|$. The following remark shows that the first assumption on the incoherence of D is still satisfied in a common positioning setting. We refer to Section 2.2 for the definition of incoherence.

Remark 4.1.1. *Assume n nodes are positioned independently and uniformly at random in a d-dimensional hypercube $[-1,1]^d$. Then the matrix D of squared distances is incoherent with parameter $\mu = 4d$, with high probability.*

Note that a typical value of d is two or three, so μ is quite small compared to n, and therefore assumptioin (i) is satisfied. However, assumption (ii), which is on sampling, is violated. It is very common and more practical to assume that the distances between close-by sensors are more likely to be measured. This introduces bias in the samples, since we are sampling entries of D which have smaller values. Hence, none of the analysis for matrix completion applies in the positioning setting. In Section 4.2, we describe and analyze a simple positioning algorithm based on the low-rank structure of D, and discuss how we can combine the idea of this algorithm and OptSpace to apply matrix completion to solve positioning problems.

Proof of Remark 4.1.1. We start with the condition **A0**. Let $D = \widetilde{X} S \widetilde{X}^T$, where $\widetilde{X} \in \mathbb{R}^{n\times(d+2)}$ is the matrix whose i-th row is the column vector $\widetilde{X}_i \equiv (1, x_{i,1}, \ldots, x_{i,d}, \|x_i\|^2)^T$

and

$$S \equiv \begin{pmatrix} 0 & 0 & 1 \\ 0 & -2\,\mathbf{I}_d & 0 \\ 1 & 0 & 0 \end{pmatrix} . \tag{4.1}$$

With probability one, the columns of $\widetilde{X}$ are linearly independent and therefore there exists a matrix $A \in \mathbb{R}^{(d+2)\times(d+2)}$ such that $\widetilde{X} = VA^T$ with $V^TV = \mathbf{I}_{(d+2)}$. The SVD of M then reads $M = U\Sigma U^T$ with $\Sigma = Q^TA^TSAQ$ and $U = VQ$ for some orthogonal matrix $Q \in \mathbb{R}^{(d+2)\times(d+2)}$. It is enough to show that

$$\sup_{1\le i\le n} \|V_i\|^2 \le \frac{\mu(d+2)}{n} ,$$

where $V_i = A^{-1}\widetilde{X}_i$ is the i-th row of V. Since $\|V_i\|^2 \le \sigma_{\min}(A)^{-2}\|\widetilde{X}_i\|^2$ and $\|\widetilde{X}_i\|^2 = 1 + \|x_i\|^2 + \|x_i\|^4 \le d\,(d+2)$, it is sufficient to bound from below $\sigma_{\min}(A)^2$. Since V is orthonormal,

$$AA^T = \widetilde{X}^T\widetilde{X} = \sum_{i=1}^n \widetilde{X}_i\widetilde{X}_i^T .$$

It is easy to compute the expectation of this matrix over the distribution of node positions $\{x_i\}$,

$$\mathbb{E}\{AA^T\} = n \begin{pmatrix} 1 & 0 & d/3 \\ 0 & (1/3)\mathbf{I}_d & 0 \\ d/3 & 0 & (d/3)^2 + 4\,d/45 \end{pmatrix} . \tag{4.2}$$

Therefore $\lambda_{\min}(\mathbb{E}\{AA^T\}) = n/3$. Using the fact that $\lambda_{\min}(\,\cdot\,)$ is a Lipschitz continuous function of its argument, together with the Chernoff bound for large deviation of sums of i.i.d. random variables, we get that $\lambda_{\min}(AA^T) \le n/4$ with high probability. This implies that $\sup_{1\le i\le n} \|V_i\|^2 \le (4d(d+2))/n$ as claimed.

Next, consider the condition **A1**. We need to show that $\sup_{i,j\in[n]} |D_{ij}| \le \Sigma_1 \mu\sqrt{d+2}/n$. Since $x_i, x_j \in [-1,1]^d$, we have $|D_{ij}| = \|x_i - x_j\|^2 \le 4d$. Further, the above argument implies that $\Sigma_1 = \|A^T S A\|_2 \ge \lambda_{\max}(AA^T)\sigma_{\min}(S) = \lambda_{\max}(AA^T)$.

Using (4.2), we get $\lambda_{\max}(\mathbb{E}\{AA^T\}) \ge n(1+d^2/9)$. Again using the standard Chernoff bound for i.i.d. random variables, we get $\lambda_{\max}(AA^T) \ge n$ with high probability. We get therefore $\Sigma_1 \ge n$ with the same probability, and $\sup_{i,j\in[n]} |D_{ij}|/\Sigma_1 \le 4d/n$, which proves our claim. $\square$

4.1.3 Decentralized positioning

One consequence of the ad-hoc nature of the underlying networks is the lack of a central infrastructure. In the centralized setting, we assumed that a central processor has access to all the pairwise distance measurements. However, centralized algorithms typically have low energy efficiency and low scalability due to dependency on a central processor and excessive communication overload. In Section 4.3, we analyze the performance of a distributed positioning algorithm. We show that the HOP-TERRAIN algorithm introduced in [87] achieves a bounded error when only local connectivity information is given.

Table 4.1: Distributed localization algorithm classification [61]

Phase	Robust positioning [87]	Ad-hoc positioning [77]	N-hop multilat. [90]
1. Distance	DV-HOP	Euclidean	Sum-dist
2. Position	Lateration	Lateration	Min-max
3. Refinement	Yes	No	Yes

Recently, a number of decentralized localization algorithms have been proposed [87, 90, 77, 72, 83]. Langendoen and Reijers [61] identified a common three-phase structure of three popular distributed positioning algorithms, namely robust positioning [87], ad-hoc positioning [77] and N-hop multilateration [90] algorithms. Table

4.1 illustrates the structure of these algorithms. In the first phase, nodes share information to collectively determine the distances from each of the nodes to a number of anchors. Anchors are special nodes with a priori knowledge of their own position in some global coordinate system. In the second phase, nodes determine their position based on the estimated distances provided by the first phase. In the last phase, the initial estimated positions are iteratively refined. It is empirically demonstrated that these simple three-phase distributed sensor localization algorithms are robust and energy-efficient [61]. However, depending on which method is used in each phase there are different tradeoffs between localization accuracy, computation complexity and power requirements. In [72], a distributed algorithm based on the gradient desent method was proposed, which is similar to ad-hoc positioning [77] but uses a different method for estimating the average distance per hop.

Another distributed approach introduced in [52] is to pose the localization problem as an inference problem on a graphical model and solve it using nonparametric belief propagation. It is a naturally distributed procedure and produces both an estimate of node positions and a representation of the location uncertainties. The estimated uncertainty may subsequently be used to determine the reliability of each node's location estimate.

4.2 Centralized algorithm

In a centralized setting, we assume that a central processor has access to all the proximity measurements available. Then we can apply a centralized algorithm to find the positions of the nodes that best matches the proximity measurements. In this section, we provide a performance bound for a popular centralized positioning algorithm known as MDS-MAP [93], when only the connectivity information is available as in the *connectivity-based model* described in the next section. Further, the algorithm can be readily applied to the *range-based model* without any modification.

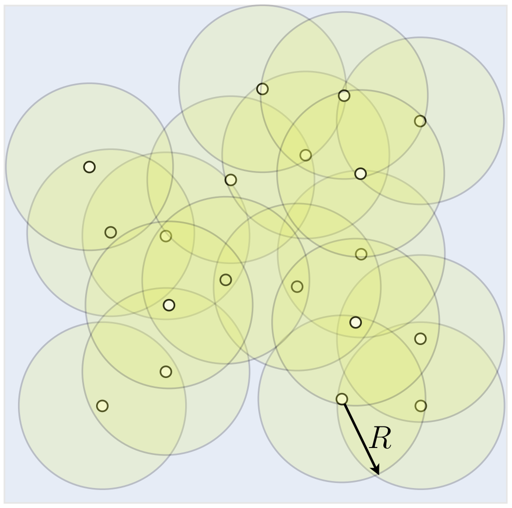


Figure 4.1: An example of random nodes with radio range R. A node is connected to its neighbors within distance R.

4.2.1 Model

We assume that the nodes are placed uniformly at random in a bounded convex domain in $\mathbb{R}^d$. To be definite, we consider the random geometric graph $G(n, r) = (V, E, P)$, where V is a set of n nodes that are distributed uniformly at random in a d-dimensional hypercube $[0, 1]^d$. $E \subseteq V \times V$ is a set of undirected edges that connect pairs of nodes that are within a fixed radio range R, and $P : E \to \mathbb{R}^+$ is a non-negative real-valued weight of the edges. The weight $P_{i,j}$, which we call the proximity measurement, corresponds to the measured distance between nodes i and j possibly corrupted by noise and quantization. Although we assume the space to be $[0, 1]^d$ hypercube for the sake of simplicity, our analysis easily generalizes to any bounded convex set, in which case the nodes are distributed according to the homogeneous Poisson process model with density $\rho = n$. This is characterized by the probability that there are exactly k nodes appearing in any region with volume A : $\mathbb{P}(k_A = k) = \frac{(\rho A)^k}{k!} e^{-\rho A}$.

Let x_i denote the position of node i in the d-dimensional space and R be the radius of detection or the radio range. We assume that if two nodes i and j are within a given radio range R, then a proximity measurement $P_{i,j}$ between those two nodes is available. Figure 4.1 shows an example of the random nodes in a unit square with radio range R. This corresponds to the disc model, a common random geometric graph model, where

$$(i, j) \in E, \text{ if and only if } d_{i,j} \leq R \ ,$$

where $d_{i,j} = \|x_i - x_j\|$. To each edge $(i, j) \in E$, we associate the proximity measurement $P_{i,j}$ between sensors i and j, which is a function of the distance $d_{i,j}$ and random noise. In an ideal case where our measurements are exact, we have $P_{i,j} = d_{i,j}$.

Localization algorithms are typically categorized into two categories depending on the available proximity information. In the first one, approximate measurements of the distances between neighboring sensors are available, perhaps with limited accuracy. This is referred to as *range-based localization* or range-aware model. Common technologies to obtain the distance measurements include Time of arrival (ToA), Time Difference of Arrival (TDoA), Angle of Arrival (AoA), and Received Signal Strength Indicator (RSSI). Under the *range-based* model, it is assumed that the distance measurements are available but may be corrupted by noise or have limited accuracy. In formulae,

$$P_{i,j} = \begin{cases} [d_{i,j} + z_{i,j}]_+ & \text{if } (i, j) \in E, \\ * & \text{otherwise,} \end{cases}$$

where $z_{i,j}$ models the measurement noise, possibly a function of the distance $d_{i,j}$, and $[a]_+ \equiv \max\{0, a\}$. Here a $*$ denotes that we do not have any proximity measurement but we know that $d_{i,j} > R$. Common examples are the additive Gaussian noise model, where the $z_{i,j}$'s are i.i.d. Gaussian random variables, and the multiplicative noise model, where $P_{i,j} = [(1+N_{i,j})d_{i,j}]_+$, for independent Gaussian random variables $N_{i,j}$'s.

Depending on the applications in which the location information is going to be used, the tolerance for error in the estimation may vary. Acknowledging the fact that the cost of hardware required to measure the distance in the range-based model may be inappropriate, we assume a simpler *range-free localization* or connectivity based model. Throughout this chapter, we assume that only connectivity information is available at the nodes. Formally,

$$P_{i,j} = \begin{cases} R & \text{if } (i,j) \in E, \\ * & \text{otherwise,} \end{cases} \tag{4.3}$$

where a $*$ denotes that the proximity measurement is missing due to $d_{i,j} > R$. Hence, a node can only detect which other nodes are within radio range R, and no other information is available. This can be considered as an extreme case quantization, where we only get one bit of information for each pair of nodes.

In the following, let D denote the $n \times n$ matrix of squared pairwise distances where $D_{ij} = d_{i,j}^2$. We use X to denote the $n \times d$ matrix of node positions where the ith row corresponds to x_i. Given the graph $G(n, r) = (V, E, P)$, the goal of positioning is to find a d-dimensional embedding of the nodes that best fits the measured proximity between all the pairs in E. The notion of embedding, or configuration, will be made clear in the following sections.

4.2.2 Multidimensional scaling

Multidimensional scaling (MDS) refers to a set of statistical techniques used in finding the configuration of objects in a low dimensional space such that the measured pairwise distances are preserved [26, 16]. It is often used for visual representation of the proximities between a set of items. For example, given a matrix of perceived similarities or dissimilarities between n items, MDS geometrically places each of those items in a low dimensional space such that the items that are similar are placed close to each other. Formally, MDS finds a lower dimensional embedding $\hat{x}_i$'s that minimize the *stress* defined as

$$\text{stress} \equiv \sqrt{\frac{\sum_{i \neq j} \left(f(d_{i,j}) - \hat{d}_{i,j}\right)^2}{\sum_{i \neq j} \hat{d}_{i,j}^2}} \,,$$

where $d_{i,j}$ is the input similarity or dissimilarity, $\hat{d}_{i,j} = \|\hat{x}_i - \hat{x}_j\|$ is the Euclidean distance in the lower dimensional embedding, and $f(\cdot)$ is some function on the input data. When MDS perfectly embeds the input data, $f(d_{i,j}) = \hat{d}_{i,j}$ and the stress is zero.

In this chapter we use what is called the classical metric MDS (We refer to [26, 16] for the definition of other types of MDS algorithms, for instance nonmetric MDS, replicated MDS, and weighted MDS). In classical metric MDS, $f(\cdot)$ is the identity function and the input dissimilarities correspond to the Euclidean distances such that $d_{i,j} = \|x_i - x_j\|$ for some lower dimensional embedding $\{x_i\}$. Further, when all the dissimilarities (or pairwise distances) are measured without error, the following spectral method correctly recovers the lower dimensional embedding up to a rigid motion.

```
Spectral method for classical metric MDS

Input: dimension $d$, input matrix $A$
Output: estimated positions $\mathrm{MDS}_d(A)$
1:  Compute $B = (-1/2)LAL$, where $L = \mathbb{I}_n - (1/n)\mathbb{1}_n\mathbb{1}_n^T$;
2:  Compute the the singular value decomposition of $B$, $B = U\Sigma U^T$;
3:  Compute the best rank-$d$ approximation $U_d\Sigma_d U_d^T$ by taking
    the $d$ largest singular values and corresponding singular vectors;
4:  Return $\mathrm{MDS}_d(A) \equiv U_d\Sigma_d^{1/2}$.
```

This algorithm has been commonly used in positioning applications [32, 93] and in the future whenever we say MDS we refer to the above algorithm. Let L be the $n \times n$ symmetric matrix $L = \mathbb{I}_n - (1/n)\mathbb{1}_n\mathbb{1}_n^T$, where $\mathbb{1}_n \in \mathbb{R}^n$ is the all ones vector and $\mathbb{I}_n$ is the $n \times n$ identity matrix. L projects n-dimensional vectors onto the $(n-1)$-dimensional subspace orthogonal to $\mathbb{1}_n$. Let $\mathrm{MDS}_d(D)$ denote the $n \times d$ matrix returned by MDS when applied to the matrix D of exact squared distances. Let $(-1/2)LDL = U\Sigma U^T$ be the singular value decomposition (SVD) of the symmetric matrix $(-1/2)LDL$. It is proved below that this is a positive semidefinite matrix. Then, we can define

$$\mathrm{MDS}_d(D) \equiv U_d\Sigma_d^{1/2} ,$$

where U_d denotes the $n \times d$ left singular matrix corresponding to the d largest singular values and Σ_d denotes the $d \times d$ diagonal matrix with d largest singular values in the diagonal. Note that since the columns of $(-1/2)LDL$ are orthogonal to $\mathbb{1}_n$ by construction, it follow that $L \cdot \text{MDS}_d(D) = \text{MDS}_d(D)$.

It can be easily shown that when MDS is applied to the matrix D of squared distances without noise, the configuration of sensors are exactly recovered [26, 32]. This follows from rewriting the distance matrix as $D = \mathbb{1}a^T + a\mathbb{1}^T - 2XX^T$, which implies that

$$-\frac{1}{2}LDL = LXX^TL \ , \tag{4.4}$$

where $a_i = \|x_i\|^2$ and i-th row of $X \in \mathbb{R}^{n\times d}$ is the position of the node i. Since LXX^TL has rank at most d, $\text{MDS}_d(D) = LXQ$ for some orthogonal matrix $Q \in \mathbb{R}^{d\times d}$ satisfying $Q^TQ = QQ^T = \mathbb{I}_d$. Note that we only get the configuration and not the absolute positions, in the sense that $\text{MDS}_d(D)$ is one version of infinitely many solutions that matches the distance measurements D.

Intuitively, it is clear that the pairwise distances are invariant to a rigid transformation (a combination of a rotation and a translation) of the matrix X of node positions. There are, therefore, multiple ways to position the nodes which result in the same pairwise distances. For future use, we introduce a formal definition of rigid transformation and related terms.

Denote by $\mathsf{O}(d)$ the orthogonal group of $d \times d$ matrices. A set of node positions $Y \in \mathbb{R}^{n\times d}$ is a *rigid transformation* of $X \in \mathbb{R}^{n\times d}$, if there exists a d-dimensional shift vector $s \in \mathbb{R}^d$ and an orthogonal matrix $Q \in \mathsf{O}(d)$ such that

$$Y = XQ + \mathbb{1}_n s^T \ .$$

Y should be interpreted as a result of first rotating the node positions by Q and then shifting all the node positions by s. Similarly, when we say two position matrices X and Y are *equal up to a rigid transformation*, we mean that there exists a rotation Q and a shift s such that $Y = XQ + \mathbb{1}_n s^T$. Also, we say a function $f(\cdot)$ is *invariant under rigid transformations* if for all X and Y that are equal up to a rigid transformation we have $f(X) = f(Y)$. Under these definitions, it is clear that D is invariant under rigid transformations.

In many applications of interest, for instance geographic routing, it is enough to find the node positions up to a rigid motion. The task of finding a *relative map* is to find one configuration of sensors that have the same pairwise distances as the original configuration. The task of finding an *absolute map* is to determine the absolute geographic coordinates of all sensors. This typically requires special nodes, called anchors, with known positions in a given coordinate system. In this section, we are interested in finding a configuration that best fits the proximity measurements and providing an analytical bound on the resulting error. The problem of finding an absolute map will be discussed in Section 4.3.

4.2.3 Algorithm

While MDS works perfectly when D is available, in practice not all proximity measurements are available because of the limited radio range R and the measurements are further corrupted by noise. In this section, we describe the MDS-MAP algorithm introduced in [93], where we apply the shortest paths algorithm on the graph $G = (V, E, P)$ to get an estimate of the unknown proximity measurements. Note that the idea of combining shortest paths with multidimensional scaling has also been used extensively in the context of low-dimensional embedding in [106, 107].

Based on MDS, MDS-MAP consists of two steps :

```
MDS-MAP [93]
Input: dimension $d$, graph $G = (V, E, P)$
Output: estimation $\widehat{X}$
1: Compute the shortest paths, and let $\widehat{D}$ be the matrix of
   squared distances along the shortest paths;
2: Apply MDS to $\widehat{D}$, and let $\widehat{X}$ be the output.
```

The original MDS-MAP algorithm introduced in [93] has an additional post-processing step to fit the given configuration to an absolute map using a small number (typically $d+1$) of anchors. However, the focus of this paper is to give a bound on the error between the relative map found by the algorithm and the correct configuration. Hence, the last step, which does not change the configuration, is omitted here.

Shortest paths. The shortest path between nodes i and j in a graph $G = (V, E, P)$ is defined as a path between two nodes such that the sum of the proximity measures of its constituent edges is minimized. Let $\hat{d}_{i,j}$ be the computed shortest path between nodes i and j. Since we assume that only the connectivity information is available as in (4.3), the shortest path is equivalent to the minimum number of hops scaled by R. Then, we define $\widehat{D} \in \mathbb{R}^{n\times n}$ as the squared distance along the shortest paths.

$$\widehat{D}_{ij} = \begin{cases} 0 & \text{if } i = j \ , \\ \hat{d}_{i,j}^2 & \text{otherwise} \ . \end{cases}$$

Note that $\widehat{D}$ is well defined only if the graph G is connected. If G is not connected, there are multiple configurations resulting in the same observed proximity measures and global localization is not possible. In the unit square, assuming sensor positions are drawn uniformly at random as defined in the previous section, the graph is connected, with high probability if $\pi R^2 > (\log n + c_n)/n$ for $c_n \to \infty$ [48]. A similar condition can be derived for generic d-dimensions as $C_d R^d > (\log n + c_n)/n$, where $C_d \leq \pi$ is a constant that depends on d. Hence, in the following, we are especially interested in the regime where $R = (\alpha \log n/n)^{1/d}$ for some positive constant α.

As will be shown in Lemma 4.2.3, the key observation in this shortest paths step is that the estimation $\hat{d}_{i,j}$ is guaranteed to be arbitrarily close to the correct distance $d_{i,j}$ for an appropriate radio range R and a large enough n. Moreover, the all-pairs shortest paths problem has an efficient algorithm whose complexity is $O(n^2 \log n + n|E|)$ [53]. For $R = (\alpha \log n/n)^{1/d}$ with constant α, the graph is sparse with $|E| = O(n \log n)$, whence the complexity is $O(n^2 \log n)$.

Multidimensional scaling. In step 2, we apply the MDS algorithm to $\widehat{D}$ to get a good estimate of X. As explained in Section 4.2.2, we compute $\widehat{X} = \text{MDS}_d(\widehat{D})$.

The idea is that if the shortest paths algorithm gives a good estimate, then the resulting error between X and $\widehat{X}$ is small. The main step in MDS is singular value decomposition of a dense matrix $\widehat{D}$, which has a complexity of $O(n^3)$.

4.2.4 Main result

The main result of this section establishes that MDS-MAP achieves an arbitrarily small error with a radio range $R = (\alpha \log n/n)^{1/d}$ with a large enough constant α, when we have only the connectivity information under the *range-free model.*

Let $\widehat{X} \in \mathbb{R}^{n\times d}$ denote an estimation of X. To compare $\widehat{X}$ and X, we first need to introduce a distance metric which is invariant under a rigid transformation. Define $L \equiv \mathbb{I}_n - (1/n)\mathbb{1}_n\mathbb{1}_n^T$ as in the MDS algorithm. L is a projection operator which eliminates the contribution of the translation, in the sense that $LX = L(X + \mathbb{1}s^T)$ for all $s \in R^d$. Note that L has the following nice properties : (i) LXX^TL is invariant under rigid transformation; (ii) $LXX^TL = L\widehat{X}\widehat{X}^TL$ implies that X and $\widehat{X}$ are equal up to a rigid transformation. This naturally leads to the following distance metric:

$$d_{\text{pos}}(X, \widehat{X}) = \frac{1}{n}\big\|LXX^TL - L\widehat{X}\widehat{X}^TL\big\|_F \,, \tag{4.5}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Notice that the factor $(1/n)$ corresponds to the usual normalization by the number of entries in the summation. Indeed this distance is invariant to rigid transformations of X and $\widehat{X}$. Furthermore, $d_{\text{pos}}(X, \widehat{X}) = 0$ implies that X and $\widehat{X}$ are equal up to a rigid transformation. With this metric, our main result establishes an upper bound on the resulting error. Define

$$R_0 \equiv \frac{8\,d^2}{\sqrt{10}}\left(\frac{\log n}{n}\right)^{1/d} \,. \tag{4.6}$$

Theorem 4.2.1. *Assume n nodes are distributed uniformly at random in the $[0,1]^d$ hypercube, for a bounded dimension $d = O(1)$. For a given radio range R, we are given the connectivity information of the nodes according to the range-free model. Then, with high probability, the distance between the estimate $\widehat{X}$ produced by* MDS-MAP *and the correct position matrix X is bounded by*

$$d_{\text{pos}}(X, \widehat{X}) \le \frac{R_0}{R} + O(R) \; ,$$

for $R > R_0$, where $d_{\text{pos}}(\cdot)$ is defined in (4.5) *and R_0 in* (4.6).

A proof is provided in Section 4.2.5. As described in the previous section, we are interested in the regime where $R = (\alpha \log n/n)^{1/d}$ for some constant α. Given any small positive constant δ, this implies that MDS-MAP is guaranteed to produce estimated positions that satisfy $d_{\text{pos}}(X, \widehat{X}) \le \delta$ with a large enough constant α and a large enough n.

Using the above theorem, we can further show that there is a linear transformation $S \in \mathbb{R}^{d\times d}$, such that when applied to the estimations, we get a similar bound in the Frobenius norm of the error in the positions. The proof of this corollary is presented in Section 4.2.5.

Corollary 4.2.2. *Under the hypotheses of Theorem 4.2.1 , with high probability,*

$$\min_{S\in\mathbb{R}^{d\times d}} \frac{1}{\sqrt{n}} \|LX - L\widehat{X}S\|_F \le \sqrt{6}\frac{R_0}{R} + O(R) \; .$$

Now that we have a bound on the estimation error using MDS-MAP algorithm, we can apply the two-step idea of OptSpace to positioning. Namely, in the first step, we apply MDS-MAP to get an initial estimate of the sensor positions $\widehat{X}$. Then, we can apply greedy algorithm to minimize the residual errors. Figure 4.2 compares the average error of this OptSpace inspired algorithm with that of MDS-MAP. In this example, we revealed the exact distances as in the range-based model. For OptSpace, we saw that we can prove strong performance guarantees, provided that we get an arbitrarily close initial estimate using a simple spectral method. The same idea applies here in the positioning setting, since we proved that we get an arbitrarily close initial estimate using a simple MDS-MAP algorithm. However, analyzing the greedy minimization in the positioning setting is more challenging, and it is an interesting research direction to analyze the performance of this modified OptSpace for positioning.

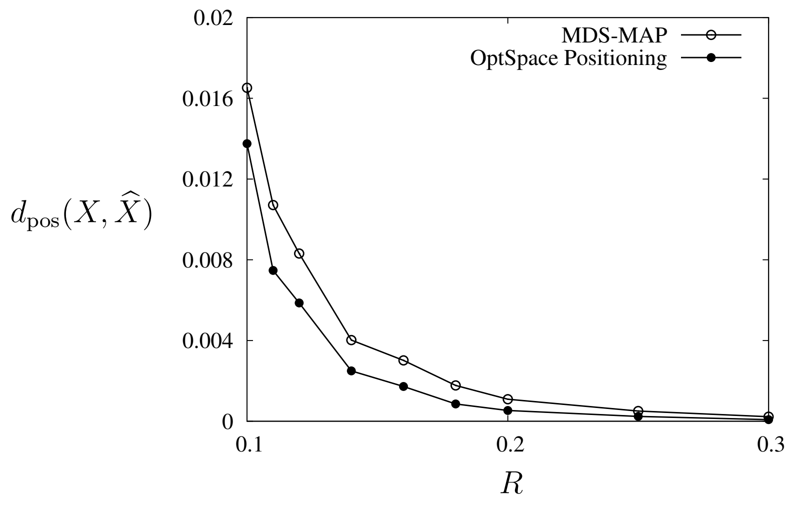


Figure 4.2: Average error with MDS-MAP and the OptSpace inspired positioning algorithm under the range-based model without noise.

4.2.5 Proof

Proof of Theorem 4.2.1

We start by bounding the distance $d_{\text{pos}}(X, \widehat{X})$, as defined in (4.5), in terms of D and $\widehat{D}$. Since $L(XX^T - \widehat{X}\widehat{X}^T)L$ has rank at most $2d$, it follows that

$$\|L(XX^T - \widehat{X}\widehat{X}^T)L\|_F \le \sqrt{2d}\|L(XX^T - \widehat{X}\widehat{X}^T)L\|_2 \; ,$$

where we used the fact that, for any rank $2d$ matrix A, we have $\|A\|_F \le \sqrt{2d}\|A\|_2$.

It is enough to bound the operator norm of $L(XX^T - \widehat{X}\widehat{X}^T)L$. To this end we use

$$\Big\| -\frac{1}{2}L\widehat{D}L - \widehat{X}\widehat{X}^T\Big\|_2 \le \Big\| -\frac{1}{2}L\widehat{D}L - A\Big\|_2 \; , \tag{4.7}$$

for any rank-d matrix A. From the definition of $\widehat{X} = \text{MDS}_d(\widehat{D})$ in Section 4.2.2, we know that $\widehat{X}\widehat{X}^T$ is the best rank-d approximation to $-(1/2)L\widehat{D}L$. Hence, $\widehat{X}\widehat{X}^T$

minimizes $\| - (1/2)L\widehat{D}L - A\|_2$ for any rank-d matrix A. It follows that

$$\begin{aligned}
&\|L(XX^T - \widehat{X}\widehat{X}^T)L\|_2 \\
&\overset{(a)}{\leq} \left\|L(XX^T + \frac{1}{2}\widehat{D})L\right\|_2 + \left\| - \frac{1}{2}L\widehat{D}L - \widehat{X}\widehat{X}^T\right\|_2 \\
&\overset{(b)}{\leq} \frac{1}{2}\|L(-D + \widehat{D})L\|_2 + \frac{1}{2}\|L(-\widehat{D} + D)L\|_2 \\
&\overset{(c)}{\leq} \|\widehat{D} - D\|_2 \ , \qquad (4.8)
\end{aligned}$$

where (a) follows from the triangular inequality and the fact that $\widehat{X} = L\widehat{X}$ as shown in Section 4.2.2. In (b) we used (4.4) and (4.7). Since the rank of $-(1/2)LDL$ is d, the second term in (b) follows by setting $A = -(1/2)LDL$. The inequality (c) follows trivially from the observation that $\|L\|_2 = 1$.

Now it is sufficient to bound $\|\widehat{D} - D\|_2$, using the following key result on the shortest paths. The main idea is that, for nodes with uniformly random positions, shortest paths provide arbitrarily close estimates of the correct distances for an appropriate radio range R. Define

$$\widetilde{R}(\beta) \equiv 8\sqrt{d}\left(\frac{(1+\beta)\log n}{n}\right)^{\frac{1}{d}} \ . \qquad (4.9)$$

For simplicity we denote $\widetilde{R}(0)$ by $\widetilde{R}_0$.

Lemma 4.2.3 (Bound on the length of the shortest path)**.** *Under the hypotheses of Theorem 4.2.1, w.h.p., the shortest paths between all pairs of nodes in the graph $G(V, E, P)$ are uniformly bounded:*

$$\hat{d}^2_{i,j} \leq \left(1 + \frac{\widetilde{R}_0}{R}\right)d^2_{i,j} + O(R) \ ,$$

for $R > R_0$, where $\widetilde{R}_0$ is defined as in (4.9) *and R_0 as in* (4.6)*.*

The proof of this lemma is given later in this section. Define an error matrix $Z = \widehat{D} - D$. Then by Lemma 4.2.3, Z is element-wise bounded by $0 \leq Z \leq (\widetilde{R}_0/R)D + O(R)$. Let u and v be the left and right singular vectors of Z respectively. Then by

Perron-Frobenius theorem, the entries of u and v are non-negative. It follow that

$$\begin{aligned} \|\widehat{D}-D\|_2 &= u^TZv \\ &\leq \frac{\widetilde{R}_0}{R}u^TDv + O(Rn) \\ &\leq \frac{\widetilde{R}_0}{R}\|D\|_2 + O(Rn)\ . \end{aligned} \tag{4.10}$$

The first inequality follows from the element-wise bound on Z and the non-negativity of u and v. It is simple to show that $\|D\|_2 = \Theta(n)$. Typically, we are interested in the region where $R = o(1)$, which implies that the first term in (4.10) dominates the error.

Now we are left with the task of bounding $\|D\|_2$. Using the element-wise bound on D, $0 \leq D_{i,j} \leq d$, we can prove a slightly loose bound : $\|D\|_2 \leq dn$. We can also prove a simple lower bound for $\|D\|_2 \geq dn/6$, with high probability. This follows from $\|D\|_2 \geq (1/n)\mathbb{1}^TD\mathbb{1}$ and the fact that right hand side concentrates around $dn/6$. Hence, the upper and lower bounds only differ in the constant. We can use concentration of measure inequalities to show a slightly tighter upper bound with a smaller constant in the following lemma which is proved in [78].

Lemma 4.2.4. *With high probability,*

$$\|D\|_2 \ \leq \ \frac{d}{\sqrt{20}}\,n + o(n)\ .$$

Applying Lemma 4.2.4 to (4.10) and substituting $\|\widehat{D}-D\|_2$ in (4.8), we get the desired bound :

$$d_{\text{pos}}(X,\widehat{X}) \leq \frac{d^{3/2}\,\widetilde{R}_0}{\sqrt{10}\,R} + O(R)\ ,$$

and this finishes the proof of Theorem 4.2.1.

□

Proof of the bound on the length of the shortest path

Proof. (Proof of Lemma 4.2.3) We prove a slightly more general version of Lemma 4.2.3. We will show that under the hypotheses of Theorem 4.2.1, for any $\beta \geq 0$ and all sensors $i \neq j$, there exists a constant C such that, with probability larger than $1 - \frac{Cn^{-\beta}}{(1+\beta)\log n}$, the shortest paths between all the pairs of nodes are uniformly bounded by

$$\hat{d}_{i,j}^2 \leq \Big(1 + \frac{\widetilde{R}(\beta)}{R}\Big) d_{i,j}^2 + O(R) \ , \tag{4.11}$$

for $R > R_0$, where $\widetilde{R}(\beta)$ is defined as in (4.9) and R_0 as in (4.6). Especially, Lemma 4.2.3 follows if we set $\beta = 0$.

We start by applying the bin-covering technique to the random geometric points in $[0,1]^d$ in a similar way as in [71]. We cover the space $[0,1]^d$ with a set of non-overlapping hypercubes whose edge lengths are δ. Thus there are total $\lceil 1/\delta \rceil^d$ bins, each of volume δ^d. In formula, bin $(i_1, \ldots, i_d)$ is the hypercube $[(i_1-1)\delta, i_1\delta) \times \cdots \times [(i_d-1)\delta, i_d\delta)$, for $i_k \in \{1, \ldots, \lceil 1/\delta \rceil\}$ and $k \in \{1, \ldots, d\}$. When n nodes are deployed in $[0,1]^d$ uniformly at random, we say a bin is empty if there is no node inside the bin. We want to ensure that, with high probability, there are no empty bins.

For a given δ, define a parameter

$$\beta \equiv (\delta^d \, n / \log n) - 1 \ .$$

Since a bin is empty with probability $(1-\delta^d)^n$, we apply union bound over all the $\lceil 1/\delta \rceil^d$ bins to get,

$$\begin{aligned} \mathbb{P}(\text{no bins are empty}) &\geq 1 - \Big\lceil \frac{1}{\delta} \Big\rceil^d (1-\delta^d)^n \\ &\geq 1 - \frac{C\,n}{(1+\beta)\log n}\Big(1 - \frac{(1+\beta)\log n}{n}\Big)^n & (4.12) \\ &\geq 1 - \frac{C\,n^{-\beta}}{(1+\beta)\log n} \ , & (4.13) \end{aligned}$$

where in (4.12) we used the fact that there exists a constant C such that $\lceil 1/\delta \rceil^d \leq$

C/δ^d, and in (4.13) we used $(1-1/n)^n \le e^{-1}$, which is true for any positive n.

Assuming that no bins are empty with high probability, we first show that the length of the shortest path is bounded by a function $F(d_{i,j})$ that only depends on the distance between the two nodes. Let $\hat{d}_{i,j}$ denote the shortest path between nodes i and j and $d_{i,j}$ denote the Euclidean distance. Define a function $F : \mathbb{R}^+ \to \mathbb{R}^+$ as

$$F(y) = \begin{cases} R & \text{if } y \le R \ , \\ (k+1)R & \text{if } y \in \mathcal{L}_k \ \text{ for } k \in \{1,2,\dots\} \ , \end{cases}$$

where $\mathcal{L}_k$ denotes the interval $(R+(k-1)(R-2\sqrt{d}\delta)\delta, R+k(R-2\sqrt{d}\delta)]$.

We will show, by induction, that for all pairs (i,j),

$$\hat{d}_{i,j} \le F(d_{i,j}) \ . \tag{4.14}$$

First, in the case when $d_{i,j} \le R$, by definition of *range-free model*, nodes i and j are connected by an edge in the graph G, whence $\hat{d}_{i,j} = R$.

Next, assume that the bound in (4.14) is true for all pairs (l,m) with $d_{l,m} \le R+(k-1)(r-2\sqrt{d})\delta$. We consider two nodes i and j at distance $d_{i,j} \in \mathcal{L}_k$. Consider a line segment $l_{i,j}$ in a d-dimensional space with one end at x_i and the other at x_j, where x_i and x_j denote the positions of nodes i and j respectively. Let p be the point in the line $l_{i,j}$ which is at distance $r-\sqrt{d}\delta$ from x_i. Then, we focus on the bin that contains p. By the assumption that no bin is empty, we know that we can always find at least one node in the bin. Let k denote any one of those nodes in the bin. Then we use following inequality which is true for all nodes (i,k,j).

$$\hat{d}_{i,j} \le \hat{d}_{i,k} + \hat{d}_{k,j} \ .$$

Each of these two terms can be bounded using the triangular inequality. To bound the first term, note that two nodes in the same bin are at most distance $\sqrt{d}\delta$ apart. Since p and x_k are in the same bin and p is at distance $R-\sqrt{d}\delta$ from node x_i by construction, we know that $d_{i,k} \le \|x_i - p\| + \|p - x_k\| \le R$, whence $\hat{d}_{i,k} = R$. Analogously for the second term, $d_{k,j} \le \|x_j - p\| + \|p - x_k\| \le R+(k-1)(R-2\sqrt{d})\delta$,

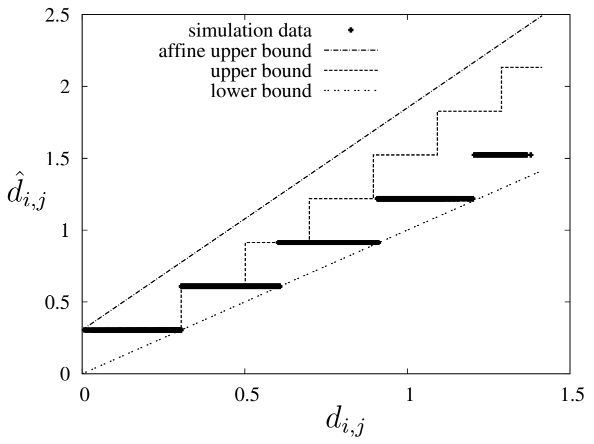


Figure 4.3: Comparison of upper and lower bounds of shortest paths $\{\hat{d}_{i,j}\}$ with respect to the correct distance $\{d_{i,j}\}$ computed for $n = 6000$ sensors in 2-dimensional square $[0,1]^2$ under *connectivity based model.*

which implies that $\hat{d}_{k,j} \leq F(d_{k,j}) = kR$. Hence, we proved that if (4.14) is true for pairs (i,j) such that $d_{i,j} \leq R + (k-1)(R - 2\sqrt{d})\delta$, then $\hat{d}_{i,j} \leq (k+1)R$ for pairs (i,j) such that $d_{i,j} \in \mathcal{L}_k$. By induction, this proves that the bound in (4.14) is true for all pairs (i,j), provided that the graph is connected.

Let $\mu = (R/2\sqrt{d})\Big(n/((1+\beta)\log n)\Big)^{1/d}$. Then, the function $F(y)$ can be easily upper bounded by an affine function $F_a(y) = (1 + 1/(\mu - 1))y + R$. Hence we have the following bound on the length of the shortest path between two nodes i and j.

$$\hat{d}_{i,j} \leq \left(1 + \frac{1}{\frac{R}{2\sqrt{d}}\left(\frac{n}{(1+\beta)\log n}\right)^{1/d} - 1}\right) d_{i,j} + R \; . \tag{4.15}$$

Figure 4.3 illustrates the comparison of the upper bounds $F(d_{i,j})$ and $F_a(d_{i,j})$, and the trivial lower bound $\hat{d}_{i,j} \geq d_{i,j}$ in a simulation with parameters $d = 2$, $n = 6000$ and $R = \sqrt{64 \log n/n}$. The simulation data shows the distribution of shortest paths between all pairs of nodes with respect to the actual pairwise distances, which confirms that shortest paths lie between the analytical upper and lower bounds. While the gap between the upper and lower bound is seemingly large, in the regime where

$R = \alpha\sqrt{\log n/n}$ with a constant α, the vertical gap R vanishes as n goes to infinity and the slope of the affine upper bound can be made arbitrarily small by increasing the radio range R or equivalently taking a large enough α. The bound on the squared shortest paths $\hat{d}^2_{i,j}$ can be derived from the bound on the shortest paths in (4.15) after some calculus.

$$\begin{aligned}
\hat{d}^2_{i,j} &\leq \left\{\frac{\mu}{\mu-1}d_{i,j} + R\right\}^2 && (4.16)\\
&= \frac{\mu^2}{(\mu-1)^2}d^2_{i,j} + R^2 + 2\frac{\mu}{\mu-1}d_{i,j}\,R && (4.17)\\
&= \left(1+\frac{2\mu-1}{(\mu-1)^2}\right)d^2_{i,j} + O(R) && (4.18)\\
&\leq \left(1+\frac{4}{\mu}\right)d^2_{i,j} + O(R)\,. && (4.19)
\end{aligned}$$

where in (4.18) we used the fact that $(\mu/(\mu-1))d_{i,j} = O(1)$, which follows from the assumptions $(R/4\sqrt{d})^d > (1+\beta)\log n/n$ and $d = O(1)$. In (4.19), we used the inequality $(2\mu-1)/(\mu-1)^2 \leq 4/\mu$, which is true for $\mu \geq 2+\sqrt{3}$. Substituting μ in the formula, this finishes the proof of the desired bound in (4.11). □

Proof of Corollary 4.2.2

Proof. Let the SVD of LX be $LX = U\Sigma V^T$, where $U^TU = \mathbf{I}_d$, $VV^T = V^TV = \mathbf{I}_d$ and $\mathbf{I}_d$ denotes the $d\times d$ identity matrix. Let $\langle X, Y\rangle = \mathrm{Tr}(X^TY)$ denote the standard scalar product. Then, for $S = \widehat{X}^TLU\Sigma^{-1}V^T$,

$$\begin{aligned}
\|LX - L\widehat{X}S\|_F &= \sup_{B\in\mathbb{R}^{n\times d}, \|B\|_F\leq 1} \langle B\,,\, LX - L\widehat{X}S\rangle\\
&= \sup_{B\in\mathbb{R}^{n\times d}, \|B\|_F\leq 1} \langle B\,,\, (LXV\Sigma U^T - L\widehat{X}\widehat{X}^TL)U\Sigma^{-1}V^T\rangle\\
&= \sup_{B\in\mathbb{R}^{n\times d}, \|B\|_F\leq 1} \langle BV\Sigma^{-1}U^T\,,\, LXX^TL - L\widehat{X}\widehat{X}^TL\rangle\\
&\leq \sup_{B\in\mathbb{R}^{n\times d}, \|B\|_F\leq 1} \|BV\Sigma^{-1}U^T\|_F\|LXX^TL - L\widehat{X}\widehat{X}^TL\|_F\,.
\end{aligned}$$

On the other hand, $\|BV\Sigma^{-1}U^T\|_F^2 = \mathrm{Tr}(BV\Sigma^{-2}V^TB) \leq \Sigma_{\min}^{-2}\|B\|_F^2$, where $\Sigma_{\min}$ is the smallest singular value of LX.

To show that $\Sigma_{\min} \geq \sqrt{n/6}$, we use concentration of measure on the singular values of LXX^TL for uniformly placed nodes. Note that $\mathbb{E}[LXX^TL] = (n/12)\mathbf{I}_d$, and singular values are Lipschitz functions of the entries. Using concentration of measure result for Lipschitz functions on bounded independent random variables, it follows that with high probability the smallest singular value of LX is larger than $\sqrt{n/6}$. This finishes the proof of the corollary. □

4.3 Decentralized algorithm

In a decentralized approach to positioning, we assume that only local information and local communication is available. Each node is aware of only who its neighbors are within its radio range. Further, each node is allowed to communicate only with its neighbors. Then, each node shares messages with its neighbors to estimate its position. In this section, we provide a performance bound for a popular decentralized positioning algorithm known as Hop-TERRAIN [87].

4.3.1 Model

The model we assume in this decentralized setting is identical to the centralized setting, except for the addition of anchor nodes. As before, we assume that we have no fine control over the placement of the nodes which we call here the *unknown nodes* (e.g., the nodes are dropped from an airplane). Formally, n unknown nodes are distributed uniformly at random in the d-dimensional hypercube $[0,1]^d$. Additionally, we assume that there are m special nodes, which we call *anchors*, which are equipped with the capability to find their own positions in some global coordinate (e.g., the anchors are equipped with a global positioning device). The anchors are assumed to be distributed uniformly at random in $[0,1]^d$ as well. If we have some control over the positions of the anchors, we show that we get similar error bound with smaller number of anchors ($m = d+1$) compared to when the anchors are randomly placed.

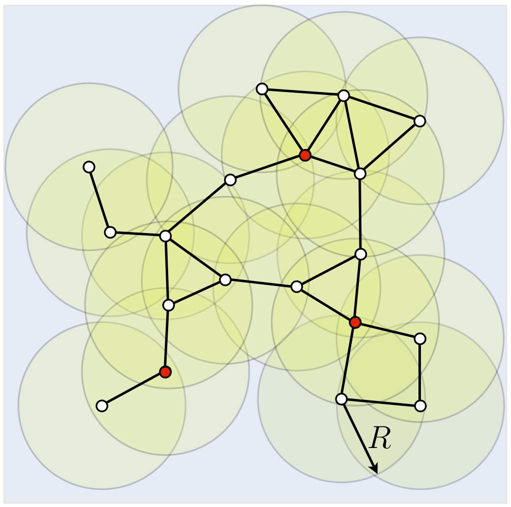


Figure 4.4: Example of a random geometric graph with radio range R. Anchors are denoted by solid circles (red). A node is connected to all other nodes that are within distance R.

Let $V_a = \{1, \dots, m\}$ denote the set of m vertices corresponding to the anchors and $V_u = \{m + 1, \dots, m + n\}$ the set of n vertices corresponding to the unknown nodes. We consider the random geometric graph model $G(m + n, r) = (V, E, P)$ where $V = V_u \cup V_a$, $E \subseteq V \times V$ is a set of undirected edges that connect pairs of nodes which are close to each other, and $P : E \to \mathbb{R}^+$ is a non-negative real-valued function. The weight function P corresponds to the proximity measurements. Let $x_a \in \mathbb{R}^d$ denote the position of anchor a for $a \in \{1, \cdots, m\}$ and $x_i \in \mathbb{R}^d$ denote the position of unknown node i for $i \in \{m + 1, \cdots, m + n\}$. We assume that only the anchors have a priori information about their own positions. Again, we assume the disk model where nodes i and j are connected if they are within a radio range R. More formally,

$$(i, j) \in E, \text{ if and only if } d_{i,j} \leq R \, .$$

Figure 4.4 shows an example of the random nodes and anchors with radio range R.

We assume the *range-free model* where we only have the connectivity information. The proximity measurements tell us which nodes are within the radio range R, but we have no information on how close they are. Formally,

$$P_{i,j} = \begin{cases} R & \text{if } (i,j) \in E, \\ * & \text{otherwise,} \end{cases}$$

where a $*$ denotes that $d_{i,j} > R$. We refer to Section 4.2.1 for the definitions and comparison of the range-free model and the range-based model.

4.3.2 Algorithm

Based on the robust positioning algorithm introduced in [87], the decentralized sensor localization algorithm we consider in this section consists of two steps :

```
HOP-TERRAIN [87]
1:  Each unknown node $i$ computes the shortest paths
    $\{\hat{d}_{i,a} : a \in V_a\}$ between itself and the anchors;
2:  Each unknown node $i$ derives an estimated position $\hat{x}_i$ using multilateration.
```

According to the three phase classification presented in Table 4.1, this is closely related to the first two phases of the robust positioning algorithm. The robust positioning algorithm uses a slightly different method for computing the shortest paths, which is compared in detail later in this section. Throughout this section, we refer to the above algorithm as HOP-TERRAIN, which is the first two steps of the robust positioning algorithm in [87].

Decentralized shortest paths. The goal of the first step is for each of the unknown nodes to estimate the distances between itself and the anchors. These approximate distances will be used in the next multilateration step to derive estimated positions. The shortest path between an unknown node i and an anchor a in the graph G provides an estimate for the Euclidean distance $d_{i,a} = \|x_i - x_a\|$, and for a carefully chosen radio range R this shortest path estimation is close to the actual distance as shown in Lemma 4.2.3.

Formally, the shortest path between an unknown node i and an anchor a in the graph $G = (V, E, P)$ is defined as a path between two nodes such that the sum of the proximity measures of its constituent edges is minimized. We denote by $\hat{d}_{i,a}$ the computed shortest path and this provides the initial estimate for the distance between the node i and the anchor a. When only the connectivity information is available and the corresponding graph $G = (V, E, P)$ is defined as in the *range-free model*, the shortest path $\hat{d}_{i,a}$ is equivalent to the minimum number of hops between a node i and an anchor a multiplied by the radio range R.

In order to find the minimum number of hops from an unknown node $i \in V_u$ to an anchor $a \in V_a$ in a distributed way, we use a method similar to DV-HOP[77]. Each unknown node maintains a table $\{x_a, h_a\}$ which is initially empty, where $x_a \in \mathbb{R}^d$ refers to the position of the anchor a and h_a to the number of hops from the unknown node to the anchor a. First, each of the anchors initiate a broadcast containing its known location and a hop count of 1. All of the one-hop neighbors surrounding the anchor, on receiving this broadcast, record the anchor's position and a hop count of 1, and then broadcast the anchor's known position and a hop count of 2. From then on, whenever a node receives a broadcast, it does one of the two things. If the broadcast refers to an anchor that is already in the record and the hop count is larger than or equal to what is recorded, then the node does nothing. Otherwise, if the broadcast refers to an anchor that is new or has a hop count that is smaller, the node updates its table with this new information on its memory and broadcast the new information after incrementing the hop count by one. When every node has computed the hop count to all the anchors, the number of hops is multiplied by the radio range R to estimate the distances between the node and the anchors. Note that to start multilateration, not all the hop counts to all the anchors are necessary. A node can start triangulation as soon as it has estimated distances to $d + 1$ anchors. There is an obvious trade-off between number of communications and performance.

The above step of computing the minimum number of hops is the same decentralized algorithm as described in DV-HOP. However, the main difference is that instead of multiplying the number of hops by a fixed radio range R, in DV-HOP, the number of hops is multiplied by an average hop distance. The average hop distance

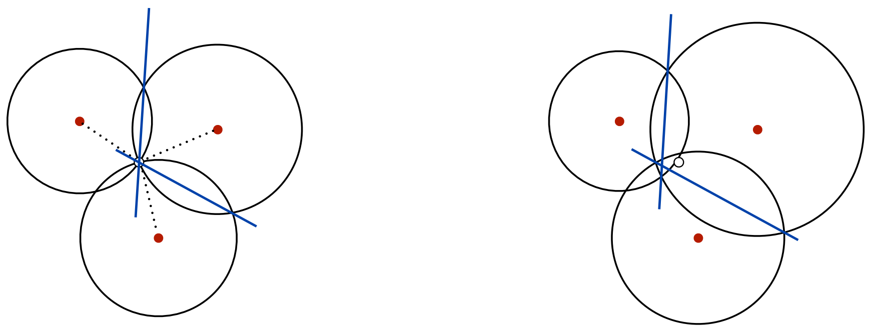


Figure 4.5: Multilateration with exact distance measurements (left) and with approximate distance measurements (right). Three solid circles denote the anchors (red) and the white circle denotes the unknown node we want to locate. The intersection of the lines (blue) corresponds to the solution of multilateration.

is computed from the known pairwise distances between anchors and the number of hops between the anchors. While numerical simulations show that the average hop distance provides a better estimate, the difference between the computed average hop distance and the radio range R becomes negligible as n grows large.

As explained in Section 4.2.3 the graph is connected with high probability if $C_d R^d > (\log n + c_n)/n$ where $C_d \leq \pi$ is a constant that depends on d. Hence, we focus in the regime where the average number of connected neighbors is slowly increasing with n, namely, $R = \alpha(\log n/n)^{1/d}$ for some large enough constant α. As shown in Lemma 4.2.3, the key observation of the shortest paths step is that the estimation is guaranteed to be arbitrarily close to the correct distance for a properly chosen radio range $R = \alpha(\log n/n)^{1/d}$ and large enough n.

Multilateration. In the second step, each unknown node $i \in V_u$ uses a set of estimated distances $\{\hat{d}_{i,a} : a \in V_a\}$ together with the known positions of the anchors to estimate its position. The resulting estimation is denoted by $\hat{x}_i$. For each node, this step consists of solving a single instance of a least squares problem ($Ax = b$) and this process is known as Multilateration [88, 61].

Let us consider a single unknown node i. The position vector x_i and the anchor positions $\{x_a : a \in [m]\}$ satisfy the following series of equations:

$$\begin{aligned} \|x_1 - x_i\|^2 &= d_{i,1}^2 \,, \\ &\vdots \\ \|x_m - x_i\|^2 &= d_{i,m}^2 \,. \end{aligned}$$

If the exact distances are known, we can simply find the intersection of m circles centered at the anchors with radius $d_{i,a}$. This is illustrated in Figure 4.5. There is a simpler way to find the intersection by linearizing the equations. This set of quadratic equations can be linearized by subtracting each line from the next line.

$$\begin{aligned} \|x_2\|^2 - \|x_1\|^2 + 2(x_1 - x_2)^T x_i = d_{i,2}^2 - d_{i,1}^2 \,, \\ \vdots \\ \|x_m\|^2 - \|x_{m-1}\|^2 + 2(x_{m-1} - x_m)^T x_i = d_{i,m}^2 - d_{i,m-1}^2 \,. \end{aligned}$$

By reordering the terms, we get a series of linear equations for node i in the form $A\,x_i = b_0^{(i)}$, for $A \in \mathbb{R}^{(m-1)\times d}$ and $b_0^{(i)} \in \mathbb{R}^{m-1}$ defined as

$$\begin{aligned} A &\equiv \begin{bmatrix} 2(x_1 - x_2)^T \\ \vdots \\ 2(x_{m-1} - x_m)^T \end{bmatrix} \,, \\ b_0^{(i)} &\equiv \begin{bmatrix} \|x_1\|^2 - \|x_2\|^2 + d_{i,2}^2 - d_{i,1}^2 \\ \vdots \\ \|x_{m-1}\|^2 - \|x_m\|^2 + d_{i,m}^2 - d_{i,m-1}^2 \end{bmatrix} \,. \end{aligned}$$

Again, if we have all the exact distance measurements and assuming A is full rank, we can solve the above system of linear equations using the least squares method to find the correct position of the unknown node i. Figure 4.5 illustrates that the intersection of the lines, which correspond to each of the linear equations, is equivalent to finding the intersection of the circles.

Note that the matrix A does not depend on the particular unknown node i and all the entries are known exactly to all the nodes after the decentralized shortest paths step. However, the vector $b_0^{(i)}$ is not available at node i, since $d_{i,a}$'s are not known. Hence we use an estimation $b^{(i)}$, which is defined from $b_0^{(i)}$ by replacing $d_{i,a}$ by $\hat{d}_{i,a}$ everywhere. Then, we can find find the optimal estimation $\hat{x}_i$ that minimizes the mean squared error by using a standard least squares method.

$$\hat{x}_i = (A^T A)^{-1} A^T b^{(i)} \; . \tag{4.20}$$

Note that the inverse is well defined when A has full-rank, which happens with high probability if $m \geq d+1$ anchors are placed uniformly at random. Figure 4.5 illustrates how the estimation changes when we have inaccurate distance measurements.

For bounded $d = O(1)$, a single least squares has complexity $O(m)$, and applying it n times results in the overall complexity of $O(n\,m)$. No communication between the nodes is necessary for this step.

4.3.3 Main result

The main result of this section establishes an error bound for the HOP-TERRAIN algorithm described in the previous section. In particular, we show that although we only have the connectivity information, as in the range-free model, an arbitrarily small error can be achieved for a radio range $R = \alpha(\log n/n)^{1/d}$ with a large enough constant α. Define

$$R_1 \equiv 16\, d^{3/2} \left(\frac{\log n}{n}\right)^{\frac{1}{d}} \; , \tag{4.21}$$

Theorem 4.3.1. *Assume n sensors and m anchors are distributed uniformly at random in the $[0,1]^d$ hypercube for a bounded dimension $d = O(1)$. For a given radio range $R > R_1$ and the number of anchors $m = \Omega(\log n)$, the following is true with high probability. For all unknown nodes $i \in V_u$, the Euclidean distance between the* *estimate $\hat{x}_i$ given by* HOP-TERRAIN *and the correct position x_i is bounded by*

$$\|x_i - \hat{x}_i\| \leq \frac{R_1}{R} + O(R) ,$$

where R_1 is defined in (4.21).

A proof is provided in Section 4.3.4. In the regime where $R = o(1)$, the error is dominated by the first term, which is inversely proportional to the radio range R. As described in the previous section, we are interested in the regime where $R = \alpha(\log n/n)^{1/d}$ for some constant α. Given a small positive constant δ, the above theorem implies that HOP-TERRAIN is guaranteed to give estimations with error less than δ for all the nodes, given a large enough constant α and large enough n.

When the positions of the anchors are chosen randomly, it is possible that, in the multilateration step, we get an ill-conditioned matrix $A^T A$, resulting in an large estimation error. This happens, for instance, if any three anchors fall close to a line. To prevent this we require $\Omega(\log n)$ anchors in Theorem 4.3.1. However, it is reasonable to assume that the system designer has some control over where the anchors are placed. The next result shows that when the positions of anchors are properly chosen, only $d+1$ anchors suffice to get a similar error bound. Note that this is the minimum number of anchors necessary for global positioning. For the sake of simplicity, we assume that one anchor is placed at the origin and d anchors are placed at positions corresponding to d-dimensional standard basis vectors. Namely, the position of the $d+1$ anchors are $\{[0,\ldots,0]$, $[1,0,\ldots,0]$, $[0,1,0,\ldots,0]$, $[0,\ldots,0,1]$ $\}$. An example in two dimensions is shown in Figure 4.6.

Theorem 4.3.2. *Assume that n sensors are distributed uniformly at random in the $[0,1]^d$ hypercube for a bounded dimension $d = O(1)$. Also, assume that there are $d+1$ anchors, one of which is placed at the origin and the d remaining anchors are placed at the standard basis in d dimensions. For a given radio range $R > R_1$, the following is true with high probability. For all unknown nodes $i \in V_u$, the Euclidean distance between the estimate $\hat{x}_i$ given by* HOP-TERRAIN *and the correct position*

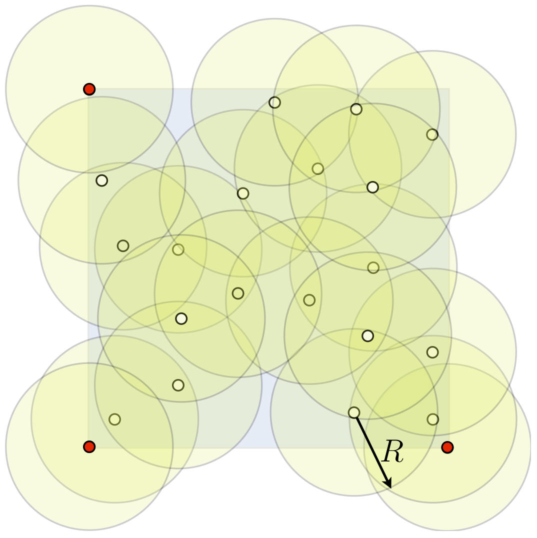


Figure 4.6: Three anchors (solid circles) in fixed positions in a standard unit square.

x_i is bounded by

$$\|x_i - \hat{x}_i\| \le \frac{d^{3/2}}{4}\frac{R_1}{R} + O(R) ,$$

where R_1 is defined in (4.21).

The proof of this theorem closely follows that of Theorem 4.3.1, and is provided in Section 4.3.4. However, there is nothing particular about the positions of the anchors in the standard basis. Any $d+1$ anchors in general positions will give a similar error bound. Only the constant factor in the bound depends on the anchor positions.

When $R = \alpha(\log n/n)^{1/d}$ for some positive parameter α, the error bound gives $\|x_i - \hat{x}_i\| \le C_1/\alpha + C_2\alpha\sqrt{\log n/n}$ for some numerical constants C_1 and C_2. The first term is inversely proportional to α and is independent of n, where as the second term is linearly dependent on α and vanishes as n grows large. This is illustrated in Figure 4.7, which shows results of numerical simulations with n sensors randomly distributed in the 2-dimensional unit square. Each point in the figure shows the root mean squared error, $\{(1/n)\sum_{i=1}^{n}\|x_i - \hat{x}_i\|^2\}^{1/2}$, averaged over 100 random instances. In the first figure, where we have only the connectivity information, we can see the

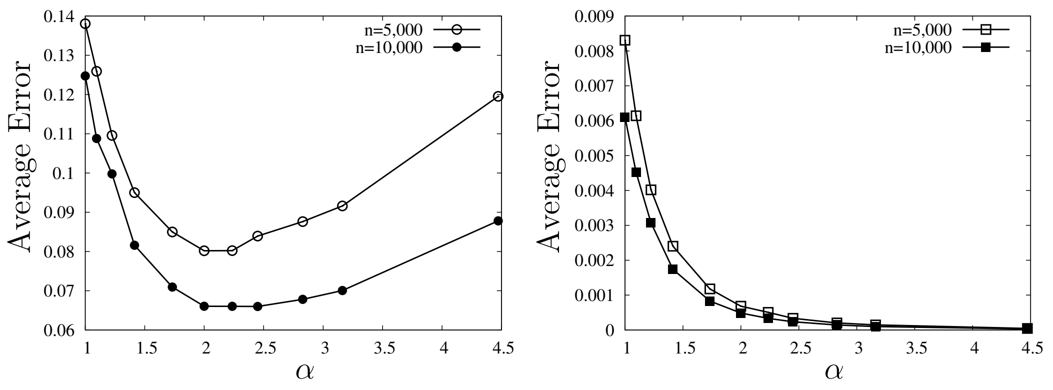


Figure 4.7: Average error between the correct position $\{x_i\}$ and estimation $\{\hat{x}_i\}$ using Hop-TERRAIN is plotted as a function of α, for $R = \alpha\sqrt{\log n/n}$. n sensors are randomly placed in the unit square under *range-free model* (left) and *range-based model* (right).

two contributions: the linear term which depends on n and the $1/\alpha$ term which is less sensitive to n. In the second figure, where we have the exact distance measurements without noise, we can see that the linear term almost vanishes even for $n = 5000$, and the overall error is much smaller.

A network of $n = 200$ nodes randomly placed in the unit square is shown in Figure 4.8. Three anchors in fixed positions are displayed by solid circles (blue). In this experiment the distance measurements are from the range-based model with multiplicative noise, where $P_{i,j} = [(1 + N_{i,j})d_{i,j}]_+$ for i.i.d. Gaussian $N_{i,j}$ with zero mean and variance 0.1. The noisy distance measurement is revealed only between the nodes within radio range $R = \sqrt{0.8 \log n/n}$. On the right, the estimated positions using Hop-TERRAIN is shown. The circles represent the correct positions and the solid lines represent the differences between the estimates and the correct positions. The average error in this example is 0.075.

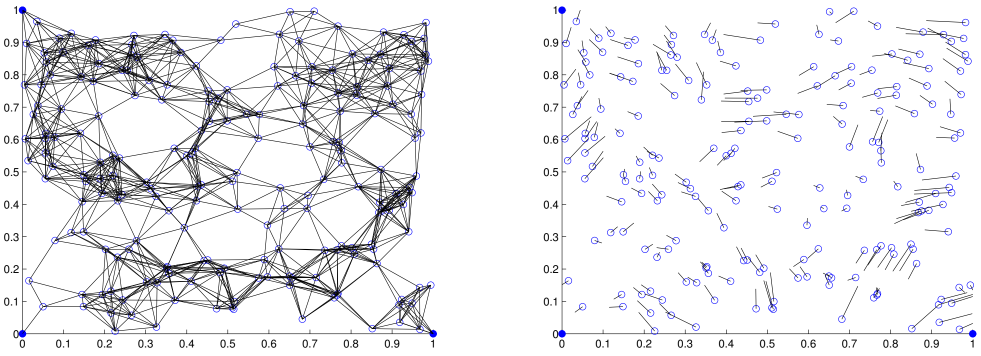


Figure 4.8: 200 nodes randomly placed in the unit square and three anchors (solid circles) in fixed positions. The radio range is $R = \sqrt{0.8 \cdot \log n/n}$. The estimation using Hop-TERRAIN is shown on the right. Each line segment shows how the estimtion differs from the original position (white circles).

4.3.4 Proof

Proof of Theorems 4.3.1 and 4.3.2

For an unknown node i, the estimate $\hat{x}_i$ is given in (4.20).

$$\begin{aligned} \|x_i - \hat{x}_i\| &= \|(A^TA)^{-1}A^Tb_0^{(i)} - (A^TA)^{-1}A^Tb^{(i)}\| \\ &\leq \|(A^TA)^{-1}A^T\|_2\|b_0^{(i)} - b^{(i)}\| \ , \end{aligned}$$

First, to bound $\|b_0^{(i)} - b^{(i)}\|$, we use the following key result on the shortest paths. By Lemma 4.2.3, for all i and j, with high probability,

$$\hat{d}_{i,j}^2 \leq \Big(1 + \frac{\widetilde{R}_0}{R}\Big) d_{i,j}^2 + O(R) \ .$$

where $\widetilde{R}_0$ is defined in (4.9). Since $d_{i,j}^2 \leq d$ for all i and j, we have

$$\begin{aligned}\|b_0^{(i)} - b^{(i)}\| &= \Big(\sum_{k=1}^{m-1} \big(d_{k,2}^2 - d_{k,1}^2 - \hat{d}_{k,2}^2 + \hat{d}_{k,1}^2\big)^2\Big)^{1/2} \\ &\leq \sqrt{m-1}\frac{\widetilde{R}_0}{R}\, d + O(\sqrt{m}R) ,\end{aligned}$$

Next, to bound $\|(A^TA)^{-1}A^T\|_2$, we use the following lemma, and this finishes the proofs of Theorems 4.3.1 and 4.3.2.

Lemma 4.3.3. *Under the hypotheses of Theorem 4.3.1, with high probability,*

$$\|(A^TA)^{-1}A^T\|_2 \leq \frac{2}{\sqrt{m-1}} .$$

Under the hypotheses of Theorem 4.3.2,

$$\|(A^TA)^{-1}A^T\|_2 \leq \frac{d}{2} .$$

Proof of Lemma 4.3.3

Proof. By using the singular value decomposition of a full-rank and tall $(m-1) \times d$ matrix A, we know that it can be written as $A = U\Sigma V^T$ where U is an orthogonal matrix, V is a unitary matrix and Σ is a diagonal matrix. Then,

$$(A^TA)^{-1}A = U\Sigma^{-1}V^T.$$

Hence,

$$||(A^TA)^{-1}A||_2 = \frac{1}{\sigma_{min}(A)}, \tag{4.22}$$

where $\sigma_{min}(A)$ is the smallest singular value of A. This means that in order to upper bound $||(A^TA)^{-1}A||_2$ we need to lower bound the smallest singular value of A.

Deterministic Model. By putting the sensors in the mentioned positions, the $d \times d$ matrix A will be Toeplitz and have the following form.

$$A = 2 \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & -1 \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix}.$$

We can easily find the inverse of matrix A.

$$A^{-1} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 1 & \cdots & 1 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 1 \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix}.$$

Since $\sigma_{\min}(A) = \sigma_{\max}(A^{-1})^{-1}$, we need to calculate the largest singular value of

$$A^{-1}\left(A^{-1}\right)^T = \frac{1}{4} \begin{bmatrix} d & d-1 & d-2 & \cdots & 1 \\ d-1 & d-1 & d-2 & \cdots & 1 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 2 & \cdots & 2 & 2 & 1 \\ 1 & \cdots & 1 & 1 & 1 \end{bmatrix}.$$

By the Gershgorin circle theorem [51], we know that $\sigma_{\max}\left(A^{-1}\left(A^{-1}\right)^T\right) \le d^2/4$. It follows that

$$\|(A^T A)^{-1} A\|_2 \le \frac{d}{2}.$$

Random Model. Let $B \equiv A^T A$. The diagonal entries of B are

$$b_{i,i} = 4 \sum_{k=1}^{m-1} (x_{k,i} - x_{k+1,i})^2, \tag{4.23}$$

for $1 \leq i \leq d$ and the off-diagonal entries are denoted by

$$b_{i,j} = 4\sum_{k=1}^{m-1}(x_{k,i} - x_{k+1,i})(x_{k,j} - x_{k+1,j}), \tag{4.24}$$

for $1 \leq i \neq j \leq d$ where $x_{k,i}$ is the i-th element of vector x_k. Using the Gershgorin circle theorem [51] we can find a lower bound on the minimum eigenvalue of B.

$$\lambda_{\min}(B) \geq \min_i(b_{i,i} - R_i),$$

where $R_i = \sum_{j\neq i}|b_{i,j}|$. We can show that, with high probability, $b_{i,i} \geq (2/3)(m-1) - 4m^{1/2+\epsilon}$ and $|b_{i,j}| \leq 16m^{1/2+\epsilon}$ for any positive constant ϵ. We refer to [54] for the proof of this claim. Then for large enough m, we have $\lambda_{\min}(B) \geq (1/4)(m-1)$. This implies that $\|(A^TA)^{-1}A\|_2 \leq 2/\sqrt{m-1}$

□

Bibliography

[1] Jester jokes. http://eigentaste.berkeley.edu.

[2] Movielens. http://www.movielens.org.

[3] Netflix prize. http://www.netflixprize.com.

[4] D. Achlioptas and F. McSherry. Fast computation of low-rank matrix approximations. *J. ACM*, 54(2), April 2007.

[5] R. Ahlswede and A. Winter. Strong converse for identification via quantum channels. *IEEE Trans. Inform. Theory*, 48(3):569–579, March 2002.

[6] S. Arora, E. Hazan, and S. Kale. A fast random sampling algorithm for sparsifying matrices. In *APPROX-RANDOM*, pages 272–279, 2006.

[7] H. D. Pfister B. Kim, A. Yedla. IMP: A message-passing algorithmfor matrix completion. `arXiv:1007.0481`, 2010.

[8] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202, 2009.

[9] R. M. Bell and Y. Koren. Lessons from the Netflix prize challenge. *SIGKDD Explor. Newsl.*, 9:75–79, December 2007.

[10] R. M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *ICDM '07: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 43–52, Washington, DC, USA, 2007. IEEE Computer Society.

[11] M. W. Berry. Large scale sparse singular value computations. *International Journal of Supercomputer Applications*, 6:13–49, 1992.

[12] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *IPSN '04: Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 46–54, New York, NY, USA, 2004. ACM.

[13] T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2009. `arXiv:0805.0510`.

[14] B. Bollobás. *Graph Theory: An Introductory Course*. Springer-Verlag, 1979.

[15] B. Bollobas. *Random Graphs*. Cambridge University Press, January 2001.

[16] I. Borg and P. J. F. Groenen. *Modern multidimensional scaling: theory and applications*. Springer, 1997.

[17] J-F Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.*, 20:1956–1982, March 2010.

[18] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? `arXiv:0912.3599`, 2010.

[19] E. J. Candès and Y. Plan. Matrix completion with noise. In *Proceedings of the IEEE*, volume 98, pages 925–936, June 2010.

[20] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundation of computational mathematics*, 9(6):717–772, February 2009.

[21] E. J. Candès, J. K. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. on Inform. Theory*, 52:489– 509, 2006.

[22] E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. `arXiv:0903.1476`, 2009.

[23] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. `arXiv:0906.2220`, 2009.

[24] P. Chen and D. Suter. Recovering the missing components in a large noisy low-rank matrix: application to SFM. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(8):1051–1063, Aug. 2004.

[25] A. L. Chistov and D. Grigoriev. Complexity of quantifier elimination in the theory of algebraically closed fields. In *Proceedings of the Mathematical Foundations of Computer Science 1984*, pages 17–31, London, UK, 1984. Springer-Verlag.

[26] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling.* Chapman & Hall, 2001.

[27] M. Cucuringu, Y. Lipman, and A. Singer. Sensor network localization by eigenvector synchronization over the euclidean group. http://www.math.princeton.edu/∼amits/publications/sensors_final.pdf.

[28] W. Dai and O. Milenkovic. SET: an algorithm for consistent matrix completion. `arXiv:0909.2705`, 2009.

[29] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[30] D. L. Donoho. Compressed Sensing. *IEEE Trans. on Inform. Theory*, 52:1289–1306, 2006.

[31] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering in large graphs and matrices. In *SODA '99: Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 291–299, Philadelphia, PA, USA, 1999. Society for Industrial and Applied Mathematics.

[32] P. Drineas, A. Javed, M. Magdon-Ismail, G. Pandurangant, R. Virrankoski, and A. Savvides. Distance matrix reconstruction from incomplete distance information for sensor network localization. In *Proceedings of Sensor and Ad-Hoc Communications and Networks Conference (SECON)*, volume 2, pages 536–544, Sept. 2006.

[33] P. Drineas and R. Kannan. Pass efficient algorithms for approximating large matrices. In *SODA '03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 223–232, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics.

[34] P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix. *SIAM J. Comput.*, 36(1), 2006.

[35] P. Drineas and M. W. Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *J. Mach. Learn. Res.*, 6:2153–2175, 2005.

[36] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matr. Anal. Appl.*, 20:303–353, 1999.

[37] E. Elnahrawy, X. Li, and R. P. Martin. The limits of localization using signal strength: a comparative study. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 406–414, 2004.

[38] P. Erdős and A Rényi. On the evolution of random graphs. In *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, pages 17–61, 1960.

[39] M. Fazel. *Matrix Rank Minimization with Applications.* PhD thesis, Stanford University, 2002.

[40] U. Feige and E. Ofek. Spectral techniques applied to sparse random graphs. *Random Struct. Algorithms*, 27(2):251–275, 2005.

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

[42] J. Friedman, J. Kahn, and E. Szemerédi. On the second eigenvalue in random regular graphs. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, pages 587–598, Seattle, Washington, USA, may 1989. ACM.

[43] A. Frieze, R. Kannan, and S. Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *J. ACM*, 51(6):1025–1041, 2004.

[44] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992.

[45] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval Journal*, 4(2):133–151, July 2001.

[46] D. Gross. Recovering low-rank matrices from few coefficients in any basis. `arXiv:0910.1879`, 2009.

[47] D. Gross, Y. Liu, S. T. Flammia, S. Becker, and J. Eisert. Quantum state tomography via compressed sensing. `arXiv:0909.3304`, 2009.

[48] P. Gupta and P.R. Kumar. Critical power for asymptotic connectivity. In *Proceedings of the 37th IEEE Conference on Decision and Control*, volume 1, pages 1106–1110 vol.1, 1998.

[49] J. P. Haldar and D. Hernando. Rank-constrained solutions to linear matrix equations using powerfactorization. *IEEE Signal Processing Letters*, 16:584 – 587, March 2009.

[50] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, 1990.

[51] R. A. Horn and C. R. Johnson. *Topics in matrix analysis.* Cambridge University Press, 1991.

[52] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky. Nonparametric belief propagation for self-calibration in sensor networks. In *IPSN '04: Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 225–233, New York, NY, USA, 2004. ACM.

[53] D. B. Johnson. Efficient algorithms for shortest paths in sparse networks. *J. ACM*, 24(1):1–13, 1977.

[54] A. Karbasi and S. Oh. Distributed sensor network localization from local connectivity: Performance analysis for the HOP-TERRAIN algorithm. In *ACM SIGMETRICS*, June 2010.

[55] R. H. Keshavan, A. Montanari, and S. Oh. Learning low rank matrices from $O(n)$ entries. In *Proc. of the Allerton Conf. on Commun., Control and Computing*, September 2008.

[56] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *IEEE Trans. Inform. Theory*, 56(6):2980–2998, June 2010.

[57] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from noisy entries. *J. Mach. Learn. Res.*, 11:2057–2078, July 2010.

[58] R. H. Keshavan and S. Oh. Optspace: A gradient descent algorithm on the grassman manifold for matrix completion. `arXiv:0910.5260`, 2009.

[59] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, New York, NY, USA, 2008. ACM.

[60] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric ad-hoc routing: of theory and practice. In *PODC '03: Proceedings of the twenty-second annual symposium on Principles of distributed computing*, pages 63–72, New York, NY, USA, 2003. ACM.

[61] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Comput. Netw.*, 43(4):499–518, 2003.

[62] K. Lee and Y. Bresler. Admira: Atomic decomposition for minimum rank approximation. `arXiv:0905.0044`, 2009.

[63] T. Luczak. On the equivalence of two basic models of random graphs. In *Random Graphs '87: Proceedings of the 3rd International Seminar on Random Graphs and Probabilistic Methods in Combinatorics*, pages 151–157, 1990.

[64] S. Ma, D. Goldfarb, and L. Chen. Fixed point and Bregman iterative methods for matrix rank minimization. `arXiv:0905.1643`, 2009.

[65] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[66] R. Mazumder, T. Hastie, and R. Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. `http://www-stat.stanford.edu/~hastie/Papers/SVD_JMLR.pdf` , 2009.

[67] R. Meka, P. Jain, and I. S. Dhillon. Guaranteed rank minimization via singular value projection. `arXiv:0909.5457`, 2009.

[68] R. Meka, P. Jain, and I. S. Dhillon. Matrix completion from power-law distributed samples. In *Advances in Neural Information Processing Systems*, 2009.

[69] M. D. Mitzenmacher. *The Power of Two Choices in Randomized Load Balancing.* PhD thesis, University of California, Berkeley, 1996.

[70] M. Mohri and A. Talwalkar. On the estimation of coherence. `arXiv:1009.0861`, 2010.

[71] S. Muthukrishnan and G. Pandurangan. The bin-covering technique for thresholding random geometric graph properties. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 989–998, Philadelphia, PA, USA, 2005.

[72] R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *IPSN '03: Proceedings of the 2nd international conference on Information processing in sensor networks*, pages 333–348, 2003.

[73] D. Needell and J. A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, Apr 2008.

[74] S. Negahban and M. J. Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. `arXiv:1009.2118`, 2010.

[75] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856. MIT Press, 2001.

[76] D. Niculescu and B. Nath. Ad hoc positioning system (APS) using AOA. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1734 – 1743 vol.3, march 2003.

[77] D. Niculescu and B. Nath. DV based positioning in ad hoc networks. *Journal of Telecommunication Systems*, 22:267–280, 2003.

[78] S. Oh, A. Karbasi, and A. Montanari. Sensor network localization from local connectivity : performance analysis for the MDS-MAP algorithm. In *Proc. of the IEEE Inform. Theory Workshop*, January 2010.

[79] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43, New York, NY, USA, 2000. ACM.

[80] B. Recht. A simpler approach to matrix completion. `arXiv:0910.0651`, 2009.

[81] B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum rank solutions of matrix equations via nuclear norm minimization. `arXiv:0706.4138`, 2007.

[82] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 713–719, New York, NY, USA, 2005. ACM.

[83] M. Rudafshani and S. Datta. Localization in wireless sensor networks. In *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks*, pages 51–60, New York, NY, USA, 2007. ACM.

[84] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

[85] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the International Conference on Machine Learning*, volume 24, pages 791–798, 2007.

[86] R. Salakhutdinov and N. Srebro. Collaborative Filtering in a Non-Uniform World: Learning with the Weighted Trace Norm. `arXiv:1002.2780`, February 2010.

[87] C. Savarese, K. Langendoen, and J. Rabaey. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *USENIX Technical Annual Conference*, pages 317–328, Monterey, CA, June 2002.

[88] C. Savarese, J. Rabaey, and J. Beutel. Locationing in distributed ad-hoc wireless sensor networks. In *in ICASSP*, pages 2037–2040, 2001.

[89] A. Savvides, C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179, New York, NY, USA, 2001. ACM.

[90] A Savvides, H Park, and M. Srivastava. The n-hop multilateration primitive for node localization problems. *Mob. Netw. Appl.*, 8(4):443–451, 2003.

[91] J. Schiller and A. Voisard. *Location Based Services.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[92] Y. Seginer. The expected norm of random matrices. *Combinatorics, Probability and Computing*, 9:149–166, 2000.

[93] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz. Localization from mere connectivity. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 201–212, New York, NY, USA, 2003. ACM.

[94] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz. Localization from connectivity in sensor networks. *IEEE Trans. Parallel Distrib. Syst.*, 15(11):961–974, 2004.

[95] V. De Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems 15*, volume 15, pages 705–712, 2003.

[96] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1615–1618, 2003.

[97] A. Singer. A remark on global positioning from local distances. *Proceedings of the National Academy of Sciences*, 105(28):9507–9511, 2008.

[98] A. Singer and M. Cucuringu. Uniqueness of low-rank matrix completion by rigidity theory. *SIAM. J. Matrix Anal. and Appl.*, 31:1621–1641, February 2010.

[99] A. M. So and Y. Ye. Theory of semidefinite programming for sensor network localization. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 405–414, Philadelphia, PA, USA, 2005.

[100] N. Srebro. *Learning with Matrix Factorizations.* PhD thesis, MIT, 2004. http://ttic.uchicago.edu/∼nati/Publications/thesis.pdf.

[101] N. Srebro and T. Jaakkola. Weighted low-rank approximations. In *20th International Conference on Machine Learning*, pages 720–727. AAAI Press, 2003.

[102] N. Srebro, J. D. M. Rennie, and T. S. Jaakola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems 17*, pages 1329–1336. MIT Press, 2005.

[103] G. Takács, I. Pilászy, B. Németh, and D. Tikk. Scalable collaborative filtering approaches for large recommender systems. *J. Mach. Learn. Res.*, 10:623–656, 2009.

[104] M. Talagrand. A new look at independence. *The Annals of Probability*, 24(1):1–34, 1996.

[105] A. Talwalkar, S. Kumar, and H. Rowley. Large-scale manifold learning. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008*, pages 1–8, August 2008.

[106] J. B. Tenenbaum. Mapping a manifold of perceptual observations. In *Advances in Neural Information Processing Systems 10*, pages 682–688. MIT Press, 1998.

[107] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.

[108] K. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. http://www.math.nus.edu.sg/∼matys, 2009.

[109] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. Comput. Vision*, 9(2):137–154, 1992.

[110] J. A. Tropp. User-friendly tail bounds for matrix martingales. `arXiv:1004:4389`, 2010.

[111] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics (Oxford, England)*, 17(6):520–525, June 2001.

[112] Z. Wang, S. Zheng, S. Boyd, and Y. Ye. Further relaxations of the SDP approach to sensor network localization. *Siam Journal on Optimization*, 19:655–673, 2008.

[113] Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. Technical report, Rice University, 2010.

[114] C. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.

[115] J. Wright, A. Ganesh, S. Rao, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices. `arXiv:0905.0233`, 2009.

[116] J. Yang and X. Yuan. An inexact alternating direction method for trace norm regularized least squares problem. Technical report, Dept. of Mathematics, Nanjing University, 2010.

[117] W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing. *SIAM J. Img. Sci.*, 1(1):143–168, 2008.

[118] K. Zhang, I. W. Tsang, and J. T. Kwok. Improved Nyström low-rank approximation and error analysis. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 1232–1239, New York, NY, USA, 2008. ACM.

[119] L. Zhang, L. Liu, C. Gotsman, and S. J. Gortler. An as-rigid-as-possible approach to sensor network localization. *ACM Trans. Sen. Netw.*, 6(4):1–21, 2010.

[120] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu. Spatio-temporal compressive sensing and internet traffic matrices. *SIGCOMM Comput. Commun. Rev.*, 39:267–278, August 2009.

[121] Z. Zhou, J. Wright, X. Li, E. J. Candès, and Y. Ma. Stable principal component pursuit. In *Proc. of the IEEE Int. Symposium on Inform. Theory*, July 2010.