EFFICIENT ALGORITHMS FOR COLLABORATIVE FILTERING

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

Raghunandan Hulikal Keshavan
August 2012

© 2012 by Raghunandan Hulikal Keshavan. All Rights Reserved.

Re-distributed by Stanford University under license with the author.

This work is licensed under a Creative Commons Attribution-Noncommercial 3.0 United States License.
http://creativecommons.org/licenses/by-nc/3.0/us/

This dissertation is online at: http://purl.stanford.edu/qz136dw4490

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Andrea Montanari, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Stephen Boyd, Co-Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Benjamin Van Roy

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Preface

Collaborative filtering is a novel statistical technique to obtain useful information or to make predictions based on data from multiple agents. A large number of such datasets are naturally represented in matrix form. Typically, there exists a matrix M from which we know a (typically sparse) subset of entries M_{ij} for (i, j) in some set E. The problem then is to predict/approximate the unseen entries. This framework of *matrix completion* is extremely general and applications include personalized recommendation systems, sensor positioning, link prediction and so on.

Low rank models have traditionally been used to learn useful information from such datasets. Low-dimensional representations simplify the description of the dataset and often yield predictive powers. As an added benefit, it is easier to store and retrieve low dimensional representations. Finally, many computationally intensive operations such as matrix multiplication and inversion are simplified with low dimensional representations.

Singular Value Decomposition (SVD) has traditionally been used to find the low-dimensional representation of a fully revealed matrix. There are numerous algorithms for computing the SVD of a matrix including several parallel implementations and implementations for sparse matrices. However, when the matrix is only partially observed, we show that SVD techniques are sub-optimal.

In this work, we will develop algorithms to learn a low rank model from a partially revealed matrix. These algorithms are computationally efficient and highly parallelizable. We will show that the proposed algorithms achieve a performance close to the fundamental limit in a number of scenarios. Finally, the algorithms achieve significantly better performance than the state-of-the-art algorithms on many real

collaborative filtering datasets.

Acknowledgments

I would like to begin by thanking my advisor Andrea Montanari for his continued support and guidance. He is indeed everything I could have ever hoped for in an advisor and more. Despite his busy schedule, he somehow always found time for us. We could just drop by his office and talk to him for hours. I will forever cherish those discussions.

We never had "group meetings" in our group. But the informal discussions at office more than made up for it. I would like to thank my office-mates Sewoong Oh, Yashodhan Kanoria, Jose Pereira, Morteza Ibrahimi, Adel Javanmard, Satish Korada, Mohsen Bayati and Yash Deshpande for the their company and help throughout my stay. Special thanks are due to Sewoong for being a great collaborator during the beginning years of my PhD.

Next I would like to thank my thesis committee members Stephen Boyd and Benjamin Van Roy. Thanks are also due to Vahab Mirrokni and Mayur Thakur for fruitful collaborations as part of my internship at Google. I would also like to thank all the professors who taught me so much throughout my graduate career.

This journey would have been difficult indeed without the help of some great friends. Thanks are due to Rangan, Krishna, Yashodhan, Venkateswaran, Aravindan, Arthi, Deepak Merugu, Srinidhi, Deepak Iyer, Nikhil, Mohan, Ananth, Srivats, Kumar, Achal, Chinmayee and many other friends for making my stay most enjoyable. Thanks also to the Asha crowd – Harendra, Pradeep J, Nikit, Venki, Arvind – for motivating me to work towards a good cause.

Most importantly, I would like to thank my parents and sisters for believing in me more than I did myself. None of this would have been possible without their affection

and support. Thank you.

Contents

Chapter 1

Introduction

Collaborative filtering is a novel statistical technique to obtain useful information or to make predictions based on data from multiple agents. Consider the example of a movie rental service. The dataset consists of ratings from users for a (typically sparse) subset of the movie collection. It is desirable to build a personalized *recommendation system* that uses this data to recommend new movies to users. Indeed, a popular movie rental service *Netflix* released such a dataset in 2006 and invited researchers to submit algorithms that predicted a pre-defined set of ratings. A substantial prize [3] was announced to the best performing algorithm, which clearly establishes the relevance of this problem to the industry.

Traditional approaches to this problem used only the data from a particular user while recommending a movie to that user, essentially building a separate recommender for each user in the system. A key point is that collaborative filtering techniques use all available information for each recommendation.

How is the data from user v relevant to the recommendations made to user u? Intuitively, the data from user v is relevant to the extent of the correlation between the users u and v. If the two users are known to rate movies very similarly, then clearly data from one user can be used while recommending movies to the other. Indeed, a number of *nearest neighbor* methods exploit this fact [12]. For each user u, identify a set of *neighbors* $N(u)$ who rate similarly to u. Further, for each $v \in N(u)$,

let s_{uv} be some similarity index. Then the predicted rating for movie m by user u is

$$\widehat{r}_{um} \leftarrow \frac{\sum_{v \in N(u,m)} s_{uv} r_{vm}}{\sum_{v \in N(u,m)} s_{uv}}$$

where $N(u,m) \subseteq N(u)$ is the set of neighbors of u that have rated movie m.

However this approach ignores the similarity between movies. Can we not use past ratings from user u to movies that are similar to m? To take an extreme example, the ratings by a user for the movie *Godfather II* is in most cases similar to her rating for the movie *Godfather I*. Indeed, a complementary approach involves finding *nearest neighbors* among movies. Let $N(m)$ be the set of neighbors of a movie m and for each $l \in N(m)$, let s_{ml} be the similarity index between movies m and l. Then the predicted rating for movie m by user u is

$$\widehat{r}_{um} \leftarrow \frac{\sum_{l \in N(m,u)} s_{ml} r_{ul}}{\sum_{l \in N(m,u)} s_{ml}}$$

where $N(m,u) \subset N(m)$ is the set of neighbors of m that the user u has rated. This approach is sometimes referred to as the item-oriented approach [69, 94].

There are several issues with the above approaches. For example, it is unclear how to choose the similarity indices. But one of the main issues with the above technique is that the *collaboration* is either among the movies or among the users but not simultaneously. Moreover, collaboration is limited to pair-wise similarities. To overcome some of these shortcomings, we will consider a *matrix completion* based approach to collaborative filtering.

A large number of datasets are naturally represented in matrix form. Typically, there exists a matrix M from which we know a (typically sparse) subset of entries M_{ij} for (i,j) in some set E. The problem then is to predict/approximate the unseen entries. We can see from the following examples that the framework of *matrix completion* is extremely general.

Consider, for example, the user-movie-rating dataset discussed above. We can represent the dataset as a rating matrix R where the r_{um} is the rating given by user u to movie m. Thus, each column corresponds to a movie and each row corresponds

to a user. Of course, we do not know the entire matrix, but only a (sparse) subset of it. And we are interested in predicting the unseen entries.

As another example, consider the following problem : we have a set of documents and we wish to categorize them based on their subject matter [31]. A common approach to this problem involves extracting the *keyword frequency* vector from each document, i.e a vector consisting of the number of occurrences of a set of keywords. Again, this dataset can be represented as a keyword-document-frequency matrix F where the entry F_{kd} is the frequency of keyword k in document d.

Finally, consider the sensor localization problem [34, 97, 84]. Here we have a (typically large) network of sensors. Each sensor can measure (approximately) its distance to other sensors within a certain radius. The problem is to reconstruct the topology of the network using this partial data. Consider the sensor-sensor-distance matrix D where the entry D_{ij} is the distance between the sensors i and j. As before, we have a sparse subset of entries from D and the problem, essentially, is to reconstruct the entire matrix D.

Predicting the missing entries of matrix is of course, impossible without further constraints on the entries of the matrix. Low rank models have traditionally been used [31, 100] to learn useful information from such datasets. Low-dimensional representations simplify the description of the dataset and often yield predictive powers. As an added benefit, it is easier to store and retrieve low dimensional representations. Finally, many computationally intensive operations such as matrix multiplication and inversion are simplified with low dimensional representations. A rank r representation of a matrix $M \in \mathbb{R}^{m \times n}$ can be written as

$$\sum_{i=1}^{r} \sigma_i u_i v_i^T$$

where $u_i \in \mathbb{R}^m$, $v_i \in \mathbb{R}^n$ and $u_i^T u_j = v_i^T v_j = \delta_{ij}$. The vectors u_i and v_i are referred to as the *principal components* of the low dimensional representation. The methods for obtaining and using the principal components is called *Principal Component Analysis* (PCA). PCA is widely used in machine learning and data mining applications. In the following, we discuss its application to a few of the problems mentioned above.

Principal Component Analysis has been used to study many user-movie-rating datasets [64]. By computing a low rank representation of the rating matrix R described above, we obtain say, r principal components. This representation yields a simpler and perhaps, more meaningful interpretation of the dataset. The principal components u_i should correspond to the different "kinds" (genres) of movies and the components v_i should record the affinity of a user towards said genre. Stacking all the principal component entries corresponding to a particular movie or a particular user, we obtain an r-vector p_m for each movie m and and r-vector q_u for each user u such that the predicted rating for movie m by user u is $\widehat{r}_{um} \leftarrow p_m^T q_u$. Now the entries in the vectors p_m correspond to the weight in the movie m of the principal genres and the entries in q_u correspond to the affinity of the user to the principal genres.

Latent Semantic Indexing (LSI) [31] is a technique based on PCA to identify the relationships between keywords and documents. Consider the document-keyword-frequency matrix F introduced above. By computing the principal components of the matrix F, we obtain an intuitive understanding of the concepts involved in the documents. Each principal component u_i can be thought of as a specific combination of keywords and corresponds to a concept. Then, the component v_i records the emphasis of the particular concept in each document. A small number of principal components reveal the most important concepts in the set of documents. Further, since the principal components v_i determine the applicability of the concepts to the documents, conceptually related documents are close to each other in this space.

There are of course, many more applications of PCA. Spectral clustering [83, 44] is used to cluster a set of objects based on a pre-defined similarity metric. This is done by first reducing the dimension by computing the principal components of the object-object-similarity matrix and then performing a standard clustering algorithm like K-means [52]. Principal Component Analysis techniques have also been used in a number of diverse areas such analysing gene-expression data [112] and control theory [80].

Computing the principal components of a fully revealed matrix is a well studied problem. The most commonly used technique is to compute the Singular Value Decomposition (SVD) of the matrix M. Now, the coefficients σ_i are the top r singular

values of M and the principal components are the singular vectors of M. Further, it is conventional to have $\|M\|_2 = \sigma_1 \geq \sigma_2 \geq \ldots \geq 0$. The entire matrix M can be rewritten as

$$M = \sum_{i=1}^{\min\{m,n\}} \sigma_i u_i v_i^T$$

In this setup, the low dimensional representation $M_r = \sum_{i=1}^{r} \sigma_i u_i v_i^T$ has several pleasing properties. For example, it is easy to see that M_r has the lowest spectral norm difference $\|M - X\|_2$ among all matrices X of rank at most r. It turns out that M_r also minimizes the Frobenius norm difference $\|M - X\|_F$ among all matrices X of rank at most r [102]. That is M_r is the solution of the optimization problem

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{ij} (M_{ij} - X_{ij})^2 \\ \text{subject to} \quad & \text{rank}(X) \leq r \end{aligned}$$

There are numerous algorithms for computing the SVD of a matrix including several parallel implementations [15] and implementations for sparse matrices [13]. There are also a number of techniques to compute only the first few singular values and the corresponding singular vectors. However in this manuscript, we are concerned, in part, with computing the principal components of the matrix M even when most of the entries of M are not seen.

Indeed in many practical applications of interest, entries are missing because of various reasons. For example, in the sensor localization problem, sensors have a limited range. They can measure their distances only to other sensors within a certain radius. This leads to missing entries in the sensor-sensor-distance matrix. In recommendation systems, the problem is interesting precisely because of missing entries. Moreover, a library like Netflix consists of orders of magnitude more movies than can be reasonably watched and rated by an individual. For instance, the dataset released by Netflix consists of about 17770 movies. An average individual has ratings for far fewer movies. Indeed, the average number of ratings for a user in the dataset was

about 200[1]. Hence most of the rating matrix is missing.

In some cases, the data acquisition process is expensive. This is the case, for example with Internet traffic management [115]. Here, the idea is to collect traffic data on the different links on a large network over time. This helps in monitoring the behavior of the network and are critical inputs to many network engineering tasks, such as trafc engineering, capacity planning and anomaly detection. Consider the link-time-traffic matrix where each entry corresponds to the traffic on a particular link at a particular time. Measuring link traffic is a difficult and time consuming process. A key idea is to measure this traffic on a different subset of links for each interval in time and use matrix completion techniques to approximate the remaining entries.

Data could be missing because of failures in the data acquisition process. For example, in applications involving microarrays [108] it is common to have failures because of insufficient resolution, corruption or sometimes even due to dust or scratches in the instruments involved. In structure-from-motion [26] applications, the goal is to infer the structure of an object from images of the object from multiple camera position. It is common to have occlusions or shadows in some of the frames, leading to missing data.

Missing data is a common and well studied problem in statistical analysis. Indeed a number of techniques have been proposed to overcome the problem of missing data [70]. A common approach is to impute the missing entries [47]. However, it is unclear how to do this systematically. One simple approach is to impute the missing entries with a default value like the mean entry or even zero. This is followed by a standard approach like the SVD. However, in most applications, this leads to poor results as we will discuss in Chapter 4.

In this work, we will develop efficient algorithms to learn a low rank model from a partially revealed matrix M. We are concerned with two broad aspects of the problem. On the one hand, we wish to learn the model from a small number of revealed entries. Indeed, in many applications, only a very sparse subset of the data

[1]We wish to note that many of the "users" in the dataset have more that 5000 ratings including a few that have rated almost the entire collection. We believe these to be automated bots or outliers. Hence the typical number of ratings per user is much smaller than the average.

matrix M is revealed. For instance, in the dataset released for the Netflix competition [3], only 1% of the entries were revealed. This metric is often referred to as the *sample complexity*. We will prove that, for a certain model of the matrix M and under certain (weak) conditions, the algorithms reconstruct (or well-approximate) the matrix M if the number of entries is larger than a certain threshold.

On the other hand, we are also interested in the algorithmic aspects of the problem. In this context, we provide efficient algorithms that can be applied to very large datasets. Further, we analyze the computational complexity of our algorithms and prove the rate of convergence for one of them, namely Alternating Least Squares (ALS). Moreover, Alternating Least Squares and Message Passing, a modification of ALS are both highly parallelizable and hence can be deployed on very large datasets.

In Chapter 2, we describe a very large scale application of Message Passing. We consider the problem of predicting missing links in a large network. For example, consider a *social network* where the links correspond to affiliation or friendship. Here, predicting missing links allows one to provide social recommendations. This problem can be cast as a matrix completion problem. We implement both Alternating Least Squares and Message Passing in a distributed framework called Pregel [74] and compare the results obtained by our algorithms with other standard algorithms.

Chapter 3 introduces the mathematical model for the matrix completion problem. There, we also detail some of the assumptions that are commonly used. In Chapter 4, we introduce an algorithm called OptSpace. We prove strong performance guarantees for the algorithm. As part of this, we generalize a celebrated result by Friedman, Kahn and Szemèradi [45] on the second eigenvalue of random graphs. In Section 4.3, we analyze a regularized version of OptSpace. This work has connections to the well studied problem [10, 96] of computing eigenvalues and eigenvectors of low-rank perturbations of random matrices.

In Chapter 5, we take a different approach to matrix completion. We consider a simple and intuitive algorithm, namely Alternating Least Squares and prove

performance guarantees. Further, we also show that the algorithm converges exponentially. This is the first rigorous analysis of such algorithms to the best of our knowledge. We then modify Alternating Least Squares into a Message Passing algorithm in Section 5.3. We demonstrate that Message Passing has much better convergence properties despite have the same computational complexity as Alternating Least Squares.

Chapter 6 puts the above algorithms in their proper perspective. We begin by presenting fundamental limits to the matrix completion problem. Here, we draw upon the work of Candès and Tao [24] who proved fundamental limits on the noiseless matrix completion problem. For the noisy case Negahban and Wainwright [82] and Candès and Plan [20] proved lower bounds on the achievable error when the matrix entries are corrupted by Gaussian noise. We show that the algorithms presented are order-optimal in a number of such interesting scenarios. Finally, we present extensive numerical experiments using both synthetic and real datasets that clearly demonstrate the usefulness of our algorithms.

Notations

We use the following notations throughout this manuscript. Any vector v will be understood as a column vector and its transpose (a row vector) will be v^T. Given two vectors $u \in \mathbb{R}^m$, $v \in \mathbb{R}^n$, we denote their tensor product by $u \otimes v = uv^T \in \mathbb{R}^{m \times n}$. For $u, v \in \mathbb{R}^m$, we let $\langle u, v \rangle = u^T v$ denote their standard scalar product. The p-th norm, $p \in [1, \infty]$ of $v \in \mathbb{R}^m$ is denoted by $\|v\|_p$.

For a matrix $X \in \mathbb{R}^{m \times n}$, we use x_i to denote the i-th row of X (viewed as a column vector), and by X_i its i-th column. We use X^T for the transpose of X and $\mathrm{Tr}(X) = \sum_{ii} X_{ii}$ for the trace of X. Its singular values will be denoted by $\sigma_1(X) \geq \sigma_2(X) \geq \cdots \geq \sigma_{\min(m,n)}(X)$ and we will also write $\sigma_{\max}(X) \equiv \sigma_1(X)$ and $\sigma_{\min}(X) \equiv \sigma_{\min(m,n)}(X)$. The (p, q) norm of X is $\|X\|_{p,q} = \sup_{v \neq 0}(\|Xv\|_q / \|v\|_p)$, i.e. the operator norm of $X : \ell_q^n \to \ell_p^m$. We will often simplify this notation, and write $\|X\|_p = \|X\|_{p,p}$. $\|X\|_F$ will denote the Frobenius norm of X. Given two matrices $X, Y \in \mathbb{R}^{m \times n}$, we use $\langle X, Y \rangle = \sum_{ij} X_{ij} Y_{ij}$ to denote their inner product. $\mathbf{1}$ will denote the identity matrix when the dimension is unambiguous.

For a given positive integer n, we use $[n] = \{1, 2, \ldots, n\}$ 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 $P(A) \geq 1 - f(n)$ and $f(n) \to 0$ as $n \to \infty$.

Chapter 2

Applications

In recent years, the advent of the Internet has greatly eased the task of data acquisition in many areas. As a result, we see an explosion in the amount of data that is now being collected. It is now common to talk about datasets involving millions (and sometimes billions) of agents. For instance, the search engine *Google* is reported to receive 2 billion queries in a single day. As another example, the social networking website *Facebook* is reported to have 700 million users and most users input a considerable amount of data. Given this situation, it is imperative that we develop methods that scale to really large datasets if we are to make sense of vast amounts of information that is being thrown our way.

Let us take a moment to understand the scale of modern day datasets. Let us, for instance, consider the graph of users on the *Google+* social network. Here, the nodes correspond to the users. There is a (directed) edge from node u to node v if user u subscribes to user v. The version of the graph we used for our experiments had about 40 million nodes and about 1 billion edges. Simply describing the graph takes a few gigabytes of memory. Analyzing the graph for any useful information, then, is impossible on a single machine. This motives the use of distributed algorithms.

As we will demonstrate in Chapter 5, our algorithms Alternating Least Squares (ALS) and Message Passing (MP) are both highly parallelizable. In this Chapter we describe a large scale implementation of these algorithms using a framework called Pregel [74]. We then apply these algorithms to predict missing

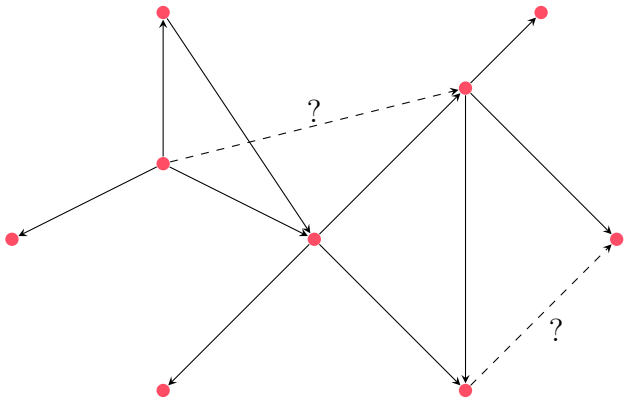

Figure 2.1: Can we predict the missing links?

links on several large datasets including the *Google+* graph described above. In Section 2.1 we set up the problem of *link prediction*. Section 2.2.1 briefly describes the implementation of Alternating Least Squares and Message Passing using Pregel. In Section 2.2.2, we describe a few benchmark algorithms for the link prediction problem. Finally, in Section 2.3, we describe the experimental setup and the results obtained. This chapter is based on joint work with Mayur, Montanari and Mirrokni [55, 57].

2.1 Link Prediction

A large number of datasets are naturally represented on graphs. Consider a (typically large) generic graph $G(V, E)$ like the one depicted in Figure 2.1. For instance, it could represent a social network like the one described above where the nodes correspond to users and edges correspond to, say, *friendship* when the edges are undirected or *subscription* when the edges are directed. Alternatively, it could represent a group of *terms* and their similarities. In this example, the nodes correspond to, say, English phrases and edges (usually weighted) correspond to similarities between phrases. Phrases that are synonymous with one another have large-weighted edges between them. Search engines typically append the incoming queries with several different approximately synonymous versions. A graph like the one described has obvious

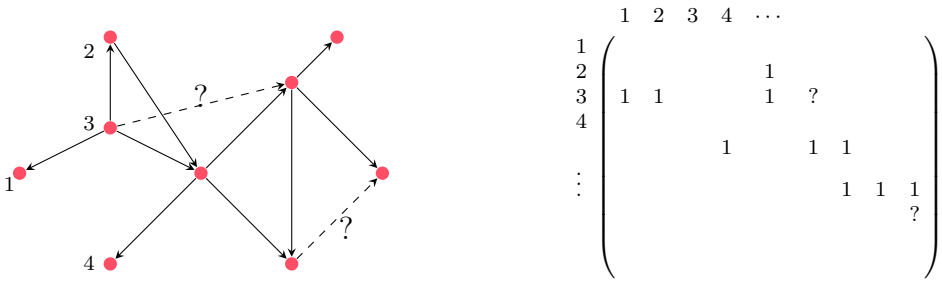

Figure 2.2: The graph G can be represented as an adjacency matrix A. Under this equivalence, the link prediction problem can be cast as a matrix completion problem.

applications in such query expansion techniques.

In this setting, a natural question to ask is the following : Is the observed graph "complete" or should there be more links among the nodes? If the graph is question is, for example, the phrase-similarity graph, then we wish our dataset to be comprehensive. Answering this question is a key step in the process. On the other hand, if we are concerned with a social graph, then answering the question and finding missing links will help build a recommendation system where we can suggest more friends to add or more people to subscribe to. This problem is referred to as the *link prediction* problem [68].

There is a natural way to express the link prediction problem as a matrix completion problem. This is done via the *adjacency matrix* of the graph. Given the graph $G(V, E)$, consider a matrix $A \in \mathbb{R}^{n \times n}$ where $n = |V|$ is the number of nodes in the graph. Further assume that the nodes are labeled $1, 2, \ldots, n$. For each edge $(i, j) \in E$, let $A_{ij} = 1$ and for each edge $(i, j) \notin E$, let $A_{ij} = 0$. The matrix A is referred to as the adjacency matrix corresponding to the graph G and captures the edge relations[1]. The relationship between the graph and the adjacency matrix is depicted in Figure 2.2.

However, our setup is slightly different to the scenario just described. Here, the absence of an edge could mean one of two things : it is either rightfully absent or it has yet to be observed by our data acquisition process. Hence, we set the A_{ij} to 1

[1]If the edges are weighted with weights w_{ij} for $(i, j) \in E$, then we set $A_{ij} = w_{ij}$ for such edges and $A_{ij} = 0$ otherwise

when the edge (i, j) is present. However, when the edge is absent, we simply assume that the entry A_{ij} is unrevealed. This leads to a natural *matrix completion* problem where we assume that the entries of A are partially observed and we wish to predict the unseen entries. In this chapter, we apply our algorithms to predict these missing entries – and the corresponding edges – and compare the performance to those of several standard algorithms.

2.2 Algorithms

In this Section, we first describe a few details implementation details for the Alternating Least Squares and the Message Passing algorithms of Chapter 5. We then describe a few other approaches like Personalized PageRank (PPR) and Contribution PageRank (CPR) to the link prediction problem.

2.2.1 Pregel Implementation

Distributed implementations of algorithms introduce several complexities compared to implementations on single machines. For example, communication between machines needs to be fast and robust. Network failures need to be resolved with retransmission of data. Further, it is not uncommon for individual machines to fail. It is therefore important to save checkpoints on machines and restart the algorithm from these checkpoints. In recent years, a number of *frameworks* have been developed to automate most of these tasks and to provide an abstract interface to the algorithm developer.

There are several such off-the-shelf frameworks available to implement algorithms in a distributed fashion. For our implementation we use a framework called Pregel [74]. Pregel is built upon the popular MapReduce framework [30]. In Pregel, the individual computing units are called *nodes*. Each node can save its own state variables. Computation proceeds in *iterations*. At each iteration, each node executes a user-defined *compute* function. Additionally, at the end of each iteration, nodes are allowed to send data to any of the other nodes. Given this setup, it is clear that

Pregel is well suited for implementing both the Alternating Least Squares and the Message Passing algorithms.

We implemented our algorithms on a cluster of 1000 computers. The computing nodes are identified with the nodes in the graph G. The compute functions simply execute the update equations for the algorithms (see Chapter 5 for details). For the Google+ graph described above, which was our largest graph, each iteration of the algorithm took about 15 minutes and the entire algorithm was completed within a couple of hours.

2.2.2 Related Algorithms

In this section we describe Personalized PageRank and Contribution PageRank and apply it to the link prediction problem. Given a graph and a re-start probability ϵ, consider a random walk on the graph starting uniformly at random at any node u. Further, let the walk be at node v at time t. With probability ϵ, the walk re-starts at a uniformly random node and with probability $1-\epsilon$, the walk moves to a neighbor of v uniformly at random. Given this setup, the PageRank [85] of a node i is the steady state probability p_i of finding the walk at i. PageRank of a node is a measure of its "importance" and has many applications including as a factor in Google search rankings.

Let us consider a slightly different random walk. Fix a particular node u. Let us start the random walk at u. Further at each restart (which happens at each iteration with probability ϵ), we reset the walk to u. The steady state probability of this random walk is called the Personalized PageRank with respect to u. Intuitively, the personalized pagerank of node v with respect to u is the probability of finding the walk at node v given that it was started at u. Personalized PageRank has been used in link prediction [68]. Here, to predict k links going out of node u, we return the k nodes with the highest personalized pagerank with respect to u and that is not already in the neighborhood of u.

A modification of Personalized PageRank, namely Contribution PageRank can also be used for link prediction. The contribution pagerank of node v with

respect to u is the personalized pagerank of v with respect to u in a modified graph G' where all the edges of G are reversed. Intuitively, the contribution pagerank of v with respect to u is the probability of the walk having started at v given that it was found to be at u.

Finally, we can also use a simple SVD for link prediction. Here, we compute the top singular values and the singular vectors of the complete adjacency matrix. We then predict edges which have a high weight in this low dimensional representation. One major drawback of SVD that we observed was with respect to scalability. We found that the SVD was infeasible for matrices with more than a few million rows and columns.

2.3 Experimental Evaluation

We tested our algorithms for their ability to predict missing links in large graphs. For our experiments, we used the following datasets. For each dataset we indicate the number of nodes n and the number of edges $|E|$ as the pair $(n, |E|)$.

1. (Small) The EU email communication network (265214, 420045) [67]
2. (Medium) The Google web graph (875713, 5105039) [67]
3. (Large) The LiveJournal social network (4847571, 68993773) [67]
4. (Very Large) The Google+ social graph: Our largest data set comes from the Google+ online social network. We took a subgraph of Google+ graph with 40 million nodes, and around 1 billion edges. In this graph, an edge between two nodes u and v corresponds to user u having user v in one of his/her Google+ circles.

We randomly divide each of the datasets above into two parts, the training set and the test set. We run our algorithms on the training set and evaluate its performance on the test set. Further, we experiment with different sizes of training vs. test sets. In the following, we present our results for training set:test set ratios of 1:3, 1:1 and 3:1. Throughout these experiments, we use a rank of 20 for all matrix factorization

Dataset	MP	ALS	SVD	PPR	CPR
Email-EU	15.14	11.1	20.21	18.97	5.61
Google-Web	40.74	40.84	39.70	46.51	39.01
Livejournal	24.16	24.45	-	32.26	26.04
Google+	21.68	21.61	-	23.3	18.3

Table 2.1: Link prediction results for the different algorithms (training set:test set ratio of 1:3). The value reported is the percentage of all test edges predicted.

Dataset	MP	ALS	SVD	PPR	CPR
Email-EU	20.49	16.84	26.39	24.55	8.96
Google-Web	55.22	55.35	53.23	52.00	40.66
Livejournal	28.53	29.11	-	35.66	28.70
Google+	31.74	31.73	-	31.08	25.65

Table 2.2: Link prediction results for the different algorithms (training set:test set ratio of 1:1). The value reported is the percentage of all test edges predicted.

algorithms. For each of these datasets, we tune the value of the regularization parameter λ by cross validation and choose the value that achieves the best test set performance

We report the percentage of test set edges correctly predicted by each algorithm in Tables 2.1, 2.2 and 2.3. We see that the Message Passing algorithm achieves performance comparable to the state-of-the-art algorithm. When the test set and the training set sizes are the same, it correctly predicts about 30% of the test set edges.

Dataset	MP	ALS	SVD	PPR	CPR
Email-EU	22.20	19.10	26.36	26.51	11.39
Google-Web	53.46	53.80	51.57	48.56	38.24
Livejournal	24.22	24.81	-	30.34	25.14
Google+	7	6.8	-	6.5	6.5

Table 2.3: Link prediction results for the different algorithms (training set:test set ratio of 3:1). The value reported is the percentage of all test edges predicted.

Chapter 3

Matrix Completion

A number of collaborative filtering problems can be cast as the problem of finding missing entries in a data matrix. In *link prediction*, we want to find the missing entries in the adjacency matrix. In (movie) recommendation systems, we want to find missing ratings in the user-movie-rating matrix. Similarly, in positioning, we want to find the missing distances in the sensor-sensor-distance matrix. We consider these problems under the general framework of the *matrix completion* problem : How best can we predict/approximate a matrix from a small subset of revealed entries? It is clear that prediction is impossible without further constraints on the entries of the matrix. In this work, we use the commonly used [63] constraint that the underlying matrix has a small rank.

In Section 3.1, we introduce the mathematical model for the matrix completion problem. In addition to the low-rank condition, we will need further assumptions on the structure of the revealed set and the factors of the underlying matrix. We introduce these in Section 3.2. The model and the assumptions encompass much of the work in this manuscript.

3.1 The Model

We let N denote an $m \times n$ matrix which is 'approximately' low rank, that is

$$N = M + W = U\Sigma V^T + W\,. \tag{3.1}$$

where U has dimensions $m \times r$, V has dimensions $n \times r$ and Σ has dimensions $r \times r$. Further, we can assume that $U^T U = \mathbf{1}$ and $V^T V = \mathbf{1}$. Without loss of generality, we assume that $m \geq n$ and let α denote the aspect ratio $\alpha \equiv m/n$. M has rank r and W can be thought of as 'noise,' or 'unexplained contributions' to N. In order to be able to explore various possibilities, we assume that W is random, but not necessarily with vanishing expectation. As a special case, W can be equal to its expectation, i.e. a deterministic perturbation.

Out of the $m \times n$ entries of N, a subset $E \subseteq [m] \times [n]$ is observed. We let $\mathcal{P}_E(N)$ be the $m \times n$ matrix that contains the observed entries of N, and is filled with 0's in the other positions

$$\mathcal{P}_E(N)_{ij} = \begin{cases} N_{ij} & \text{if } (i,j) \in E\,, \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

More generally, this equation defines the projection operator $\mathcal{P}_E : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$. The *matrix completion* problem amounts to reconstructing or approximating the low rank matrix M from the observations $\mathcal{P}_E(N)$. We sometimes use N^E to denote $\mathcal{P}_E(N)$.

3.2 Assumptions

Sampling Model

For our analytical results, we assume that E is uniformly distributed among all sets of size $|E|$. Define $\epsilon \equiv |E|/\sqrt{mn}$. In the case $m = n$, ϵ corresponds to the average number of revealed entries per row or 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 model) there exists a constant A such that, for all $\epsilon\sqrt{\alpha} \geq 1$

$$\mathbb{P}\Big\{|E| \in [n\epsilon\sqrt{\alpha} - A\sqrt{n\log n}, n\epsilon\sqrt{\alpha} + A\sqrt{n\log n}]\Big\} \geq 1 - \frac{1}{n^{10}}\,. \tag{3.3}$$

Since the success of an 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 [71, 17].

Other sampling models have been considered in the literature. In sensor localization [28, 34, 84], entries are revealed with a probability that is inversely proportional to the size of the entry. For fast computation of approximate SVD, [4] presents an algorithm where the matrix entries are sampled proportional to their absolute value. Finally, Dhillon et. al. [77] advocate the use of power-law distributed samples for collaborative filtering applications. However, we follow the independent entries model described above because of two reasons : it is more tractable for analysis and it is of more general interest than the other models.

Incoherence property

Before we introduce the notion of incoherence, let us consider an extreme example for the matrix M. Let $M \in \mathbb{R}^{n\times n} = e_1 e_1^T$, i.e M consists of zeros everywhere except M_{11} which is 1. It is not hard to see that reconstructing M from a randomly sampled subset is impossible if the subset does not contain M_{11}. But the probability of this event is vanishingly small unless $|E| = \Omega(n^2)$. Since we are mostly concerned with subsets with size $|E| = O(n\log n)$, we introduce the following notion of incoherence to prohibit such matrices from the problem space.

The matrices U, V and Σ will be said to be (μ_0, μ_1)-*incoherent* if they satisfy the following properties:

A0. For all $i \in [m]$, $j \in [n]$, we have $\|u_i\|_2^2 \leq \mu_0 r/m$, $\|v_j\|_2^2 \leq \mu_0 r/n$.

A1. For all $i \in [m]$, $j \in [n]$, we have $|M_{ij}| \leq \mu_1 \Sigma_1 r^{1/2}/\sqrt{mn}$.

Apart from a difference in normalization, the first assumption A0 coincides with the corresponding assumption in [21]. Further [21] makes an assumption that $|\sum_{k=1}^{r} U_{i,k}V_{j,k}| \leq \mu_1 r^{1/2}$. This is analogous to A1 (called also there 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, if the vectors $(U_{i,1}, \ldots, U_{i,r})$ and $(V_{j,1}, \ldots, V_{j,r})$ are collinear, our condition is weaker, and is implied by the assumption of [21].

It is easy to see that $\mu_0 \in [1, m/r]$ since $\|u_i\|_2^2 \leq \|U\|_2^2 = 1$ for all $i \in [m]$ and $\|v_j\|_2^2 \leq \|V\|_2^2 = 1$ for all $j \in [n]$. Similarly $\mu_1 \in [1, m/\sqrt{r}]$ since $\Sigma_1 \geq |M_{ij}|$ for all $(i, j) \in [m] \times [n]$. However in this manuscript, we will chiefly be concerned with matrices with μ_0 and μ_1 that are polylogarithmic in n and we refer to such matrices as *incoherent*. In many of our results, we will only need assumption A0. In those cases, we refer to μ_0 as the incoherence parameter and denote it simply as μ. However, when both assumptions A0 and A1 are applied, we define $\mu \equiv \max\{\mu_0, \mu_1\}$ as the incoherence parameter.

The incoherence condition is satisfied with high probability if $M = XY^T$ where the entries of $X \in \mathbb{R}^{m\times r}$ and $Y \in \mathbb{R}^{n\times r}$ are i.i.d. zero mean uniformly bounded random variables, with incoherence parameter scaling as $O(\min\{r, \log n\})^{1/2}$ (this follows from a standard Chernoff bound argument). It is also satisfied with high probability if $M = U\Sigma V^T$ with U and V uniformly random matrices with $U^TU = \mathbf{1}$ and $V^TV = \mathbf{1}$, with incoherence parameter scaling as $O(r \log n)$.

Chapter 4

Optspace

Spectral techniques have long been used in machine learning, statistics and signal processing [83, 8, 31]. Given a matrix M, the top singular vectors of M 'explain' most of M. This intuition is quantified by the following interpretation of singular value decomposition : for any r, the rank r matrix defined by the top r singular vectors (and the associated singular values) forms the closest rank r approximation to M in terms of the Frobenius norm.

In this chapter, we will explore the use of spectral techniques for *matrix completion*. We first need to extend the idea of singular value decomposition to a subset of a matrix. This is commonly done by 'imputing' the unseen entries by a default value, like 0. As we shall see in Theorem 4.2.1, taking the first r singular vectors of this imputed matrix, already provides a reasonable estimate of M. We will then use gradient descent on a non-convex cost function to refine this estimate. Theorem 4.2.2 bounds the error achieved by the refined estimate. We refer to the entire algorithm as OptSpace.

The rest of the chapter is organized as follows. Section 4.1 describes the details of OptSpace. In Section 4.2, we discuss the analytical results concerning OptSpace. For proofs of these results, we refer to [59, 60]. In Section 4.3, we explore the effect of regularization on the performance of OptSpace. It will be shown that regularization significantly improves the performance of the spectral method. Indeed, this improvement is quantified by the analytical results described in Section 4.3.1. We

defer the proofs of the these results to Section 4.3.2. This chapter is based on joint work with Montanari and Oh [59, 60, 56].

4.1 Algorithm

Consider the sampled matrix N^E defined in Section 3.1. Discounting noise for the moment, we see that N^E_{ij} is a random variable with an expected value of pN_{ij} where $p \equiv |E|/mn$ is the probability of the entry being revealed. This suggests that

$$\frac{1}{p}N^E = \frac{1}{p}\left(M^E + W^E\right)$$

is a proxy for the original matrix M. Following this intuition, we can compute the singular value decomposition of $(mn/|E|)N^E$ as

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

and return the "best" rank r approximation

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

Achlioptas and McSherry showed [4] that this provides a close approximation to the original matrix M if $|E| \geq 8n(\log n)^4$. It turns out that, if $|E| = O(n)$, this algorithm performs very poorly. The reason is that the matrix N^E contains columns and rows with $O(\log n/\log\log n)$ non-zero (revealed) entries [59]. Indeed this can be shown to hold with a probability larger than $1 - 1/n$ using a simple Poisson approximation [78]. These over-represented rows/columns alter the spectrum of N^E as illustrated in the left panel of Figure 4.1 where the rank-3 structure of the underlying matrix is buried under the structure induced by the sampling process. This motivates a particular 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).

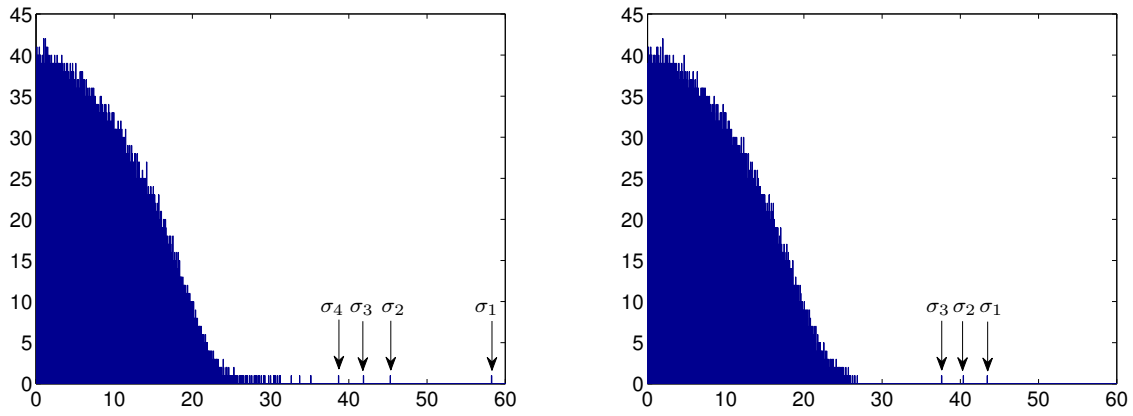

Figure 4.1: Histogram of the singular values of a partially revealed matrix M^E before trimming (left) and after trimming (right) for $10^4 \times 10^4$ random rank-3 matrix M with $\epsilon = 30$ and $\Sigma = \mathrm{diag}(1, 1.1, 1.2)$. After trimming the underlying rank-3 structure becomes clear. Here the number of revealed entries per row follows a heavy tail distribution with $\mathbb{P}\{N = k\} = \mathrm{const.}/k^3$.

Trimming

To avoid the problems associated with over-represented rows/columns, we propose a simple preprocessing routine. Set to 0 all columns in N^E with degree larger than $2|E|/n$. Set to 0 all rows with degree larger than $2|E|/m$. Let the matrix thus obtained be $\widetilde{N}^E$. Figure 4.1 shows the singular value distributions of N^E and $\widetilde{N}^E$ for a random rank-3 matrix M. The surprise is that trimming (which amounts to throwing out information) makes the underlying rank-3 structure much more apparent. 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.

The routine suggested above might appear counter-intuitive since we seem to be "throwing" away valuable data. However, as shown in [59], the naïve projection algorithm yields provably bad performance. Note that we re-incorporate the trimmed data in the final step of the algorithm. So, trimming is used only for the projection step of the algorithm. Further, trimming is just one way to overcome over-representation. We use it because it is analytical tractable. There are potentially many different pre-processing routines that achieve similar results. Indeed, similar techniques have

been commonly used in collaborative filtering applications [63, 91, 93, 100, 103]. For a detailed description of the role of trimming, we refer to [59].

Projection

Projection consists of computing the singular value decomposition of the trimmed matrix and considering only the top r components, i.e we compute $\mathcal{P}_r(\widetilde{N}^E)$. The algorithm at this point consists of a pre-processing step followed by a projection to the space of rank r matrices. Theorem 4.2.1 analyzes the output of the projection step. In particular, it bounds the deviation of $\mathcal{P}_r(\widetilde{N}^E)$ from M in terms of the root mean squared error. This theorem is comparable to the results of Achlioptas and McSherry [4] with the added novelty of trimming which allows us to eliminate the condition on $|E|$ which was necessary there.

Gradient Descent

The last step of the algorithm consists of gradient descent on a non-convex cost function. This allows to reduce (or eliminate) small discrepancies between $\mathcal{P}_r(\widetilde{N}^E)$ and M. Given $X \in \mathbb{R}^{m\times r}$, $Y \in \mathbb{R}^{n\times r}$ with $X^TX = \mathbf{1}$ and $Y^TY = \mathbf{1}$, we define

$$F(X,Y) \equiv \min_{S\in\mathbb{R}^{r\times r}} \mathcal{F}(X,Y,S)\,, \tag{4.1}$$

$$\mathcal{F}(X,Y,S) \equiv \frac{1}{2}\sum_{(i,j)\in E} (M_{ij} - (XSY^T)_{ij})^2\,. \tag{4.2}$$

We now minimize $F(X,Y)$ locally starting from (X_0,Y_0) where X_0 and Y_0 are orthonormal matrices obtained in the previous step, i.e $X_0S_0Y_0^T = \mathcal{P}_r(\widetilde{N}^E)$.

Notice that it is easy to evaluate $F(X,Y)$ since it is defined by minimizing the quadratic function $S \mapsto \mathcal{F}(X,Y,S)$ over the low-dimensional matrix S. Further it depends on the matrices 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 4.1.1 for background and references). Optimization over

OptSpace
Input: matrix N^E, rank r
Output: estimate $\widehat{M}$
1: **Trimming :** Trim N^E, and let $\widetilde{N}^E$ be the output;
2: **Projection :** Project $\widetilde{N}^E$ to $\mathcal{P}_r(\widetilde{N}^E)$;
3: **Manifold Optimization :**
Clean residual errors by minimizing the discrepancy $F(X, Y)$.

Figure 4.2: A description of the OptSpace algorithm

Grassmann manifolds is a well understood topic [38] and efficient algorithms (in particular Newton and conjugate gradient) can be applied. To be definite, we assume that gradient descent with line search is used to minimize $F(X, Y)$.

The function $\mathcal{F}(X, Y, S)$ is a natural risk function to optimize over. Similar objective functions have been used in collaborative filtering in [89, 91, 93, 100, 103]. Indeed, a similar objective function will be the subject of our study in Chapter 5. The crucial differences of our algorithm with respect to other implementations are : (i) We start our gradient descent at the projection point. In contrast, most implementations have randomized initializations. (ii) We minimize $F(X, Y)$ instead of $\mathcal{F}(X, Y, S)$ over Grassmann manifolds. In terms of the above routines, the OptSpace algorithm has been summarized in Figure 4.2.

The implementation proposed here implicitly assumes that the rank r is known. This is indeed that case in many practical applications like positioning [34, 97] or structure from motion [26, 106] where the rank is known and is small. Moreover, a very simple algorithm for estimating the rank of the matrix M from the revealed entries was introduced in [61]. It was proved there that the algorithm recovers the correct rank with high probability under the hypotheses of Theorem 4.2.2 in the noiseless case.

4.1.1 Manifold Optimization

The function $F(X, Y)$ defined in Eq. (4.1) and to be minimized in the last part of the algorithm can naturally be viewed as defined on a Grassmann manifold. Here we recall from [38] a few important facts on the geometry of the Grassmann manifold and related optimization algorithms in the next section. In the following section, we compute the gradients and describe the manifold optimization step.

Geometry of the Grassmann manifold

Denote by $\mathsf{O}(d)$ the orthogonal group of $d \times d$ matrices. The Grassmann manifold is defined as the quotient $\mathsf{G}(n, r) \simeq \mathsf{O}(n)/\mathsf{O}(r) \times \mathsf{O}(n-r)$. In other words, a point on the manifold is the equivalence class of an $n \times r$ orthogonal matrix A

$$[A] = \{AQ : Q \in \mathsf{O}(r)\}\,. \tag{4.3}$$

For consistency with the rest of the paper, we will assume the normalization $A^T A = \mathbf{1}$ where $\mathbf{1}$ denotes the identity matrix. To represent a point in $\mathsf{G}(n, r)$, we will use an explicit representative of this form. More abstractly, $\mathsf{G}(n, r)$ is the manifold of r-dimensional subspaces of $\mathbb{R}^n$.

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

$$F : \mathsf{M}(m, n) \to \mathbb{R}\,, \tag{4.4}$$

$$([X], [Y]) \mapsto F(X, Y)\,. \tag{4.5}$$

In the following, a point in this manifold will be represented as a pair $\mathbf{x} = (X, Y)$, with X an $m \times r$ orthogonal matrix and Y an $n \times r$ 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 $\mathbf{g} = (G, H)$, $G \in \mathbb{R}^{m\times r}$, $H \in \mathbb{R}^{n\times r}$ such that $G^T X = H^T Y = 0$. The 'canonical' Riemann metric on the Grassmann manifold corresponds to the usual scalar product $\langle G, G' \rangle \equiv \mathrm{Tr}(G^T G')$. The induced scalar product on $\mathsf{T}_{\mathbf{x}}$ between $\mathbf{g} = (G, H)$ and $\mathbf{g}' = (G', H')$ is $\langle \mathbf{g}, \mathbf{g}' \rangle = \langle G, G' \rangle + \langle H, H' \rangle$.

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

$$d(\mathbf{x}_1, \mathbf{x}_2) \equiv \sqrt{d(X_1, X_2)^2 + d(Y_1, Y_2)^2} \tag{4.6}$$

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

$$d(X_1, X_2) = \|\theta\|_2 \,. \tag{4.7}$$

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:

$$d_{\mathrm{c}}(X_1, X_2) = \min_{Q_1, Q_2 \in \mathsf{O}(r)} \|X_1 Q_1 - X_2 Q_2\|_F \qquad \text{(chordal distance)}, \tag{4.8}$$

$$d_{\mathrm{p}}(X_1, X_2) = \frac{1}{\sqrt{2}} \|X_1 X_1^T - X_2 X_2^T\|_F \qquad \text{(projection distance)}. \tag{4.9}$$

Notice that d_{c} and d_{p} do not depend on the specific representatives X_1, 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. $d_{\mathrm{c}}(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{d_{\mathrm{c}}(X_1, X_2)^2 + d_{\mathrm{c}}(Y_1, Y_2)^2}$.

Remark 4.1.1. *The geodesic, chordal and projection distance are equivalent, namely*

$$\frac{1}{\pi} d(X_1, X_2) \le \frac{1}{\sqrt{2}} d_{\mathrm{c}}(X_1, X_2) \le d_{\mathrm{p}}(X_1, X_2) \le d_{\mathrm{c}}(X_1, X_2) \le d(X_1, X_2) \,. \tag{4.10}$$

For a proof of this fact, we refer to [59].

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 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 $W = L\Theta R^T$ [38]:

$$X(t) = UR\cos(\Theta t)R^T + L\sin(\Theta t)R^T\,, \tag{4.11}$$

which can be evaluated in time of order $O(nr)$. An analogous expression holds for $Y(t)$.

Gradient Descent

The gradient of F at $\mathbf{x}$ is the vector $\operatorname{grad} F(\mathbf{x}) \in \mathsf{T}_{\mathbf{x}}$ 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

$$F(\mathbf{x}(t)) = F(\mathbf{x}) + \langle \operatorname{grad} F(\mathbf{x}), \mathbf{w}\rangle\, t + O(t^2)\,. \tag{4.12}$$

The two components of the gradient are then

$$\operatorname{grad} F(\mathbf{x})_X = \mathcal{P}_E(XSY^T - M)YS^T - XQ_X\,, \tag{4.13}$$

$$\operatorname{grad} F(\mathbf{x})_Y = \mathcal{P}_E(XSY^T - M)^T XS - YQ_Y\,, \tag{4.14}$$

where S is the minimizer in (4.1) and $Q_X, Q_Y \in \mathbb{R}^{r\times r}$ are determined by the condition $\operatorname{grad} F(\mathbf{x}) \in \mathsf{T}_{\mathbf{x}}$. This yields

$$Q_X = \frac{1}{m}X^T\mathcal{P}_E(XSY^T - M)YS^T\,, \tag{4.15}$$

$$Q_Y = \frac{1}{n}Y^T\mathcal{P}_E(XSY^T - M)^T XS\,. \tag{4.16}$$

However, since S is the minimizer of (4.1), we have that $Q_X = 0$ and $Q_Y = 0$. At this point the gradient descent algorithm is fully specified. It takes as input the factors of

$\mathsf{T}_r(\widetilde{M}^E)$, to be denoted as $\mathbf{x}_0 = (X_0, Y_0)$, and minimizes a regularized cost function

$$\begin{aligned} \widetilde{F}(X,Y) &= F(X,Y) + \rho\, G(X,Y) && (4.17)\\ &\equiv F(X,Y) + \rho \sum_{i=1}^{m} G_1\left(\frac{\|X^{(i)}\|^2}{3\mu_0 r}\right) + \rho \sum_{j=1}^{n} G_1\left(\frac{\|Y^{(j)}\|^2}{3\mu_0 r}\right), && (4.18)\end{aligned}$$

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.

$$G_1(z) = \begin{cases} 0 & \text{if } z \le 1, \\ e^{(z-1)^2} - 1 & \text{if } z \ge 1. \end{cases} \tag{4.19}$$

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. We will set $\rho = n\epsilon$ Let us stress that the regularization term is mainly introduced for our proof technique to work (and a broad family of functions 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 $Q, Q' \in \mathsf{O}(r)$. Let

$$\mathcal{K}(\mu') \equiv \left\{(X,Y) \text{ such that } \|X^{(i)}\|^2 \le \mu' r,\ \|Y^{(j)}\|^2 \le \mu' r \text{ for all } i \in [m],\ j \in [n]\right\} \tag{4.20}$$

We have $G(X,Y) = 0$ on $\mathcal{K}(3\mu_0)$. Notice that $\mathbf{u} \in \mathcal{K}(\mu_0)$ by the incoherence property. Also, by the following remark [59], we can assume that $\mathbf{x}_0 \in \mathcal{K}(3\mu_0)$.

Remark 4.1.2. *Let $U, X \in \mathbb{R}^{n\times r}$ with $U^TU = X^TX = n\mathbf{1}$ and $U \in \mathcal{K}(\mu_0)$ and $d(X,U) \le \delta \le \frac{1}{16}$. Then there exists $X'' \in \mathbb{R}^{n\times r}$ such that $X''^TX'' = n\mathbf{1}$, $X'' \in \mathcal{K}(3\mu_0)$ and $d(X'',U) \le 4\delta$. Further, such an X'' can be computed in a time of $O(nr^2)$.*

The manifold optimization procedure is detailed in Figure 4.3. In this context, γ must be set in such a way that $d(\mathbf{u}, \mathbf{x}_0) \le \gamma$. The next remark determines the correct scale.

MANIFOLD OPTIMIZATION

Input : matrix M^E, factors $\mathbf{x}_0$
Output : estimate $\widehat{M}$
1: For $k = 0, 1, \dots$ do:
2: Compute $\mathbf{w}_k = \operatorname{grad} \widetilde{F}(\mathbf{x}_k)$;
3: Let $t \mapsto \mathbf{x}_k(t)$ be the geodesic with $\mathbf{x}_k(t) = \mathbf{x}_k + \mathbf{w}_k t + O(t^2)$;
4: Minimize $t \mapsto \widetilde{F}(\mathbf{x}_k(t))$ for $t \geq 0$, subject to $d(\mathbf{x}_k(t), \mathbf{x}_0) \leq \gamma$;
5: Set $\mathbf{x}_{k+1} = \mathbf{x}_k(t_k)$ where t_k is the minimum location;
6: End For.
7: Output $\widehat{M} = X_k S_k Y_k^T$ where S_k is the minimizer.

Figure 4.3: The manifold optimization step of OPTSPACE

Remark 4.1.3. *Let $U, X \in \mathbb{R}^{m\times r}$ with $U^T U = X^T X = m\mathbf{1}$, $V, Y \in \mathbb{R}^{n\times r}$ with $V^T V = Y^T Y = n\mathbf{1}$, and $M = U\Sigma V^T$, $\widehat{M} = XSY^T$ for $\Sigma = \operatorname{diag}(\Sigma_1, \dots, \Sigma_r)$ and $S \in \mathbb{R}^{r\times r}$. If $\Sigma_1, \dots, \Sigma_r \geq \Sigma_{\min}$, then*

$$d_{\mathrm{p}}(U, X) \leq \frac{1}{\sqrt{2\alpha} n \Sigma_{\min}} \|M - \widehat{M}\|_F \quad , \quad d_{\mathrm{p}}(V, Y) \leq \frac{1}{\sqrt{2\alpha} n \Sigma_{\min}} \|M - \widehat{M}\|_F \tag{4.21}$$

As a consequence of this remark and Theorem 4.2.1, we can assume that $d(\mathbf{u}, \mathbf{x}_0) \leq C(\frac{\Sigma_{\max}}{\Sigma_{\min}}) \frac{\mu_1 r\sqrt{\alpha}}{\sqrt{\epsilon}}$. We shall then set $\gamma = C'(\frac{\Sigma_{\max}}{\Sigma_{\min}}) \frac{\mu_1 r\sqrt{\alpha}}{\sqrt{\epsilon}}$ (the value of C' is set in the course of the proof). A few remarks regarding the manifold optimization routine are in order.

(*i*) 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*) The line minimization instruction 4 (which might appear complex to implement) can be replaced by a standard step selection procedure, such as the one in [6]. Such routines are standard practice in gradient descent algorithms.

(*iii*) There is no need to know the actual value of μ_0 in the regularization term. One can start with $\mu_0 = 1$ and then repeat the optimization doubling it at each step. One the other hand, an algorithm for estimating the incoherence parameter was

proposed in [72] in the context of column sampling. A similar approach could be developed for the uniform sampling of entries.

(*iv*) The Hessian of F can be computed explicitly as well. This opens the way to quadratically convergent minimization algorithms (e.g. the Newton method). However, the computational complexity of such a procedure might limit its applicability.

Figure 4.4 illustrates the effectiveness of the manifold optimization step. Here, we present the results of our simulations with 1000×1000 matrices of rank 10. We generate the matrix as $M = UV^T$ where $U_{ij}, V_{ij} \approx \mathsf{N}(0,1)$. We plot both the fit error $\|\mathcal{P}_E(M - \widehat{M})\|_F/\sqrt{|E|}$ and the prediction error $\|M - \widehat{M}\|_F/n$ as a function of the number of iterations of gradient descent for two different revealed set sizes $|E| = 10^5$ and $|E| = 2 \cdot 10^5$. We see that the prediction error decays exponentially with the number of iterations. Further, the prediction error is close to the fit error, thus validating our use of the fit error as a stopping criterion. We refer to Chapter 6 for more extensive simulations demonstrating the performance of OptSpace on simulated and real datasets.

4.2 Main Results

We characterize the performance of OptSpace based on the following analytical results. Since we are interested in 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 bounds for any m and n.

We begin by analyzing the "Projection" step of OptSpace. There are efficient algorithms for finding the rank-r projection of a sparse matrix and the complexity of the whole procedure is $O(|E|r \log n)$. Our first result bounds the estimation error for this simple procedure.

Theorem 4.2.1. *Let M be a rank r matrix of dimensions $n\alpha \times n$ that satisfies $|M_{i,j}| \le M_{\max}$ for all $(i,j) \in [n\alpha] \times [n]$. Assume that the revealed set $E \subset [n\alpha] \times [n]$*

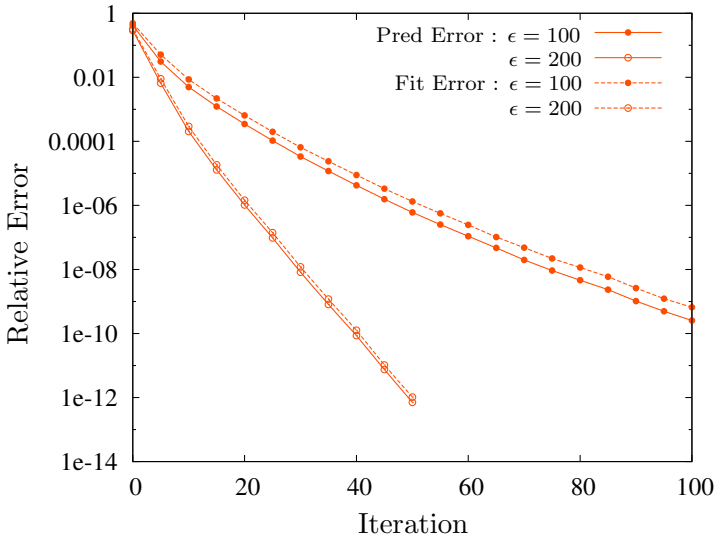

Figure 4.4: Empirical demonstration of the rate of convergence of MANIFOLD OPTIMIZATION. We plot the fit error and the prediction error as a function of the number of iterations of gradient descent. Simulations with $m = n = 1000$ and $r = 10$ and $\epsilon = |E|/n = 100$ and 200.

is uniformly random given the size $|E|$. *Then there is a constant* C *such that with probability larger than* $1 - 1/n^3$

$$\frac{1}{\sqrt{mn}}\|M - \mathcal{P}_r(\widetilde{N}^E)\|_{\mathrm{F}} \le C\, M_{\max} \left(\frac{nr\alpha^{3/2}}{|E|}\right)^{1/2} + 2\sqrt{2}\frac{n\sqrt{r\alpha}}{|E|}\|\widetilde{W}^E\|_2 \qquad (4.22)$$

Recall that $\|\cdot\|_F$ denotes the Frobenius norm and $\|\cdot\|_2$ denotes the operator norm. Further, the left hand side is simply the root mean squared error of the estimation. Observe that we do not need any of the incoherence assumptions for this result.

Projection is a standard procedure employed for dimensionality reduction. Indeed, procedures similar to our Projection step have been widely used in learning. The above result is a rigorous analysis of this intuitive step. The error bound achieved above consists of two terms. The first term corresponds to the missing entries. Indeed, the bound is non-trivial as soon as the number of revealed entries is larger, in order, than the number of degrees of freedom $O(nr)$. The second term in the error bound corresponds to the noise W.

The second main result provides performance guarantees for the entire algorithm, i.e the output of the manifold optimization step. This theorem is order-optimal in a number of important circumstances including the noiseless case (bounded r and μ) and the case of i.i.d Gaussian noise. We refer to Chapter 6 for further comparisons.

Theorem 4.2.2. *Let M be a rank r matrix of dimensions $n\alpha \times n$ satisfying the incoherence conditions with parameter μ. Let $\Sigma_{\min} = \Sigma_1 \leq \cdots \leq \Sigma_r = \Sigma_{\max}$ be singular values of M and define $\kappa \equiv \Sigma_{\max}/\Sigma_{\min}$. Assume that the revealed set $E \subset [n\alpha] \times [n]$ is uniformly random given the size $|E|$. Let $\widehat{M}$ be the output of* OptSpace *given the input $N^E = M^E + W^E$. Then there exists numerical constants C and C' such that if*

$$|E| \;\geq\; Cn\mu r\alpha\kappa^2 \max\left\{\log n\,;\, \mu r\sqrt{\alpha}\kappa^4\right\}\;,$$

then, with probability at least $1 - 1/n^3$,

$$\frac{1}{\sqrt{mn}}\,\|M - \widehat{M}\|_F \leq C'\,\frac{n\kappa^2\sqrt{r\alpha}}{|E|}\|W^E\|_2 \tag{4.23}$$

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

As was noted before, the cleaning step essentially eliminates the term corresponding to the missing entries in Theorem 4.2.1. In Figure 4.5, we demonstrate the performance of OptSpace in practice. Here, we plot the *reconstruction rate*, the empirical probability of recovery the matrix M with a tolerance of 10^{-4}, i.e $\|M - \widehat{M}\|_F/\|M\|_F \leq 10^{-4}$ as a function of the sampling probability $p = |E|/mn$. The matrices M are generated as UV^T with $U_{ij}, V_{ij} \approx \mathsf{N}(0,1)$. We conduct the experiments with $m = n = 500$ and for ranks 10, 20 and 40. For comparison, we have also plotted the upper bound on reconstruction error computed using rigidity theory [98]. It is clear from the plots that the OptSpace performs close to the fundamental limit even in practice. Results of more extensive simulations are presented in Section 6.3.

A key observation concerning the results presented is that the dependence on noise in Theorems 4.2.1 and 4.2.2 is only through the operator norm of $\widetilde{W}^E$. This

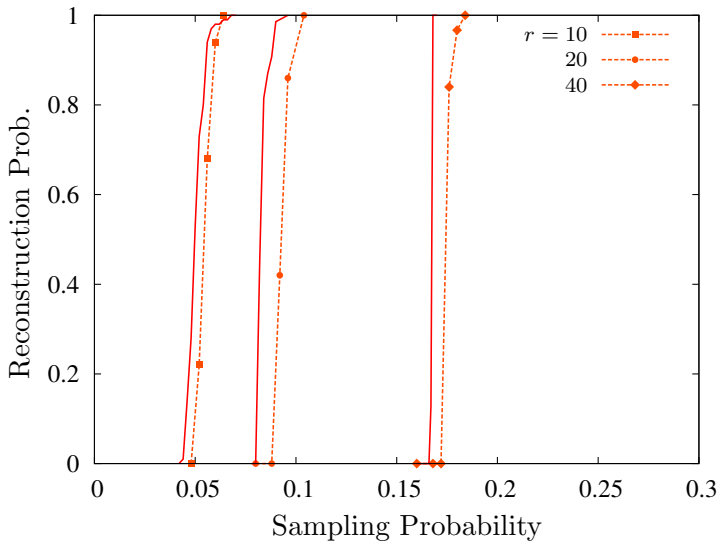

Figure 4.5: Empirical reconstruction probability for OptSpace as a function of the sampling probability $|E|/n^2$. Simulations with $m = n = 500$ and $r = 10, 20, 40$. The plain red curve is the fundamental limit computed using rigidity theory [98].

is of significance since in many cases of interest (when the noise is not spectrally concentrated, for eg. i.i.d sub-Gaussian entries), the operator norm of $\|\widetilde{W}^E\|_2$ is much smaller than the *noise intensity*, typically measured by the Frobenius norm $\|\widetilde{W}^W\|_F$. In order to gain more intuition about the results, it is instructive to consider a couple of simple models for the noise matrix W:

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

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

for some bounded constant σ^2.

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

The basic parameter entering our main results is the operator norm of $\widetilde{W}^E$, which is bounded as follows.

Theorem 4.2.3. *If W is a random matrix drawn according to the independent entries model, then there is a constant C such that,*

$$\|\widetilde{W}^E\|_2 \leq C\sigma \left(\frac{\sqrt{\alpha}|E| \log |E|}{n} \right)^{1/2} ,$$

with probability at least $1 - 1/n^3$.

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

$$\|\widetilde{W}^E\|_2 \leq \frac{2|E|}{n\sqrt{\alpha}} W_{\max} , \tag{4.24}$$

for any realization of E.

Note that for $|E| = \Omega(n \log n)$, no row or column is over-represented with high probability. It follows that in the regime of $|E|$ for which the conditions of Theorem 4.2.2 are satisfied, we have $W^E = \widetilde{W}^E$ and hence the bounds apply to $\|W^E\|_2$ as well.

Figure 4.6 studies the performance of OptSpace in the presence of noise. The matrix M is generated as before but with $m = n = 600$, rank $r = 2$ and noise variance $\sigma^2 = 1$. We plot the average root mean squared error $\|M - \widehat{M}\|_F/n$ as a function of the sampling probability $p = |E|/n^2$. For comparison, we have also plotted the *Oracle Bound* proved in [20] (see Section 6.2.1 for more details). We see that the projection step of OptSpace gives a reasonable estimate of M, especially when the size of the revealed set is large. Further, OptSpace converges in about 10 iterations and is essentially indistinguishable from the lower bound after a certain threshold on p.

4.3 The Role of Regularization

In many of the applications that we are interested in, the noise level is much larger than the underlying signal. In such cases, OptSpace tends to overfit the model. The situation is illustrated by an example in Figure 4.7. Here, we plot the relative error ($\|M - \widehat{M}\|_F/\|M\|_F$) achieved by OptSpace as a function of the number of iterations of gradient descent for different signal to noise ratios ($\|M^E\|_F/\|W^E\|_F$).

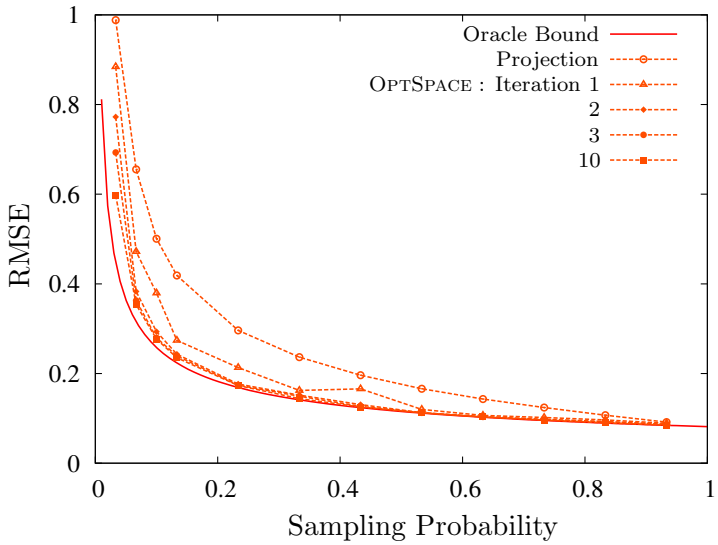

Figure 4.6: Average RMSE $\|M - \widehat{M}\|_F/n$ as a function of the sampling probability $|E|/n^2$. Simulations with $m = n = 600$, $r = 2$ and $\sigma^2 = 1$. The plain red curve is the *Oracle Bound* from [20]. The "Projection" step of OptSpace provides a reasonable estimate of $\widehat{M}$. OptSpace converges in about 10 iterations of gradient descent and achieves a performance close to the lower bound.

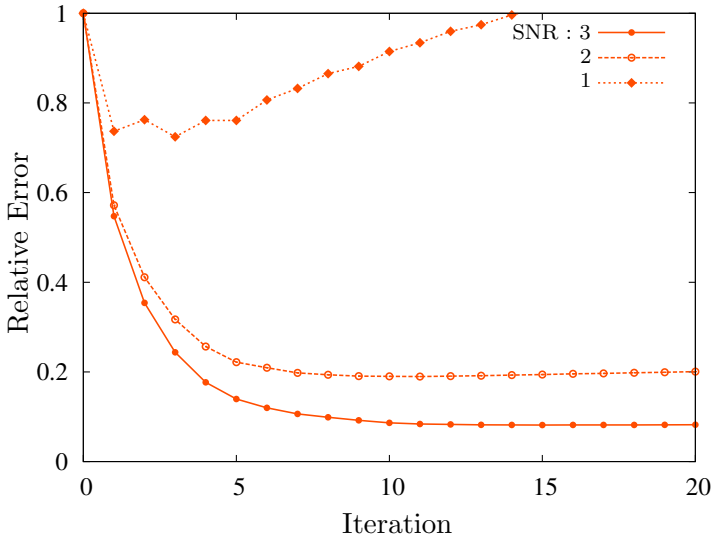

Figure 4.7: The relative error achieved by OptSpace as a function of the number of iterations of gradient descent for different SNR values. OptSpace tends to overfit the model when the noise levels are high.

As can be seen from the plot, OptSpace works well when the signal to noise ratio is large. However, for small signal to noise ratios, it overfits the model resulting in poor performance on the test set.

In order to tackle this problem, we use the following *regularized* cost function well suited for spectral reconstruction methods.

$$\mathcal{F}_E(X, Y; S) \equiv \frac{1}{2}\|\mathcal{P}_E(N - XSY^T)\|_F^2 + \frac{1}{2}\lambda \|S\|_F^2 . \tag{4.25}$$

Here, $\lambda > 0$ is a regularization parameter, with λ large corresponding to more constrained fitting.

Algorithm

Using the regularized cost function, the new algorithm is a straightforward extension of OptSpace. A key observation is that the following modified cost function can be

OptSpace(λ)

Input: matrix N^E, rank r
Output: estimate $\widehat{M}$
1: **Trimming :** Trim N^E, and let $\widetilde{N}^E$ be the output;
2: **Projection :**
Minimize $\widehat{\mathcal{F}}_E(X, Y; S)$ using SVD. Let X_0, Y_0, S_0 be the output;
3: **Manifold Optimization :**
Minimize $\mathcal{F}_E(X, Y; S)$ by gradient descent using X_0, Y_0, S_0 as initial condition.

Figure 4.8: A description of the modified OptSpace algorithm

minimized by singular value decomposition (see Section 4.3.2):

$$\widehat{\mathcal{F}}_E(X, Y; S) \equiv \frac{1}{2}\|\mathcal{P}_E(N) - XSY^T\|_F^2 + \frac{1}{2}\lambda\,\|S\|_F^2\,. \tag{4.26}$$

Note that the "Projection" step of OptSpace described in Figure 4.2 corresponds to minimizing $\widehat{\mathcal{F}}_E(X, Y; S)$ with $\lambda = 0$. It turns out that the idea of projection can still be used even when $\lambda \neq 0$ to minimize $\widehat{\mathcal{F}}_E(X, Y; S)$. The modified OptSpace algorithm is given in Figure 4.8. We study this algorithm in regimes where the trimming step is superfluous and we will not discuss it further here. Our main analytical result is a sharp characterization of the mean square error after step 2.

Regularization techniques have been widely used in a number of machine learning applications including collaborative filtering [91, 93, 100, 101, 90]. For instance, one important component of many algorithms competing for the Netflix challenge [3], consisted in minimizing the cost function

$$\mathcal{H}_E(X, Y; S) \equiv \frac{1}{2}\|\mathcal{P}_E(N - \widetilde{X}\widetilde{Y}^T)\|_F^2 + \frac{1}{2}\lambda\,\|\widetilde{X}\|_F^2 + \frac{1}{2}\lambda\,\|\widetilde{Y}\|_F^2\,. \tag{4.27}$$

(this is also known as *maximum margin matrix factorization* [101, 90]). Here the minimization variables are $\widetilde{X} \in \mathbb{R}^{m\times r}$, $\widetilde{Y} \in \mathbb{R}^{n\times r}$. Unlike in OptSpace, these matrices are not constrained to be orthogonal, and as a consequence the problem becomes significantly more degenerate. Notice that, in our approach, the orthogonality constraint fixes the norms $\|X\|_F$, $\|Y\|_F$. This motivates the use of $\|S\|_F^2$ as a regularization term.

Convex relaxations of the matrix completion using nuclear norm minimization were studied in [21, 20]. As emphasized by Mazumder, Hastie and Tibshirani [75], such nuclear norms relaxations can be viewed as spectral regularizations of a least squares problem.

4.3.1 Main Results

We characterize the performance of the "Projection" step of the modified OptSpace algorithm by means of the following theorem. Here and below, the limit $n \to \infty$ is understood to be taken with $m/n \to \alpha \in (0, \infty)$.

Theorem 4.3.1. *Assume $|M_{ij}| \le M_{\max}$, W_{ij} to be i.i.d. random variables with mean 0 variance $\sigma^2/\sqrt{mn}$ and $\mathbb{E}\{W_{ij}^4\} \le C/n^2$, and that for each entry (i,j), N_{ij} is observed (i.e. $(i,j) \in E$) independently with probability p. Finally let $\widehat{M} = X_0 S_0 Y_0^T$ be the rank r matrix reconstructed by step 2 of of* OptSpace*, for the optimal choice of λ. Then, almost surely for $n \to \infty$*

$$\frac{\|\widehat{M} - M\|_F^2}{\|M\|_F^2} = 1 - \frac{\left\{ \sum_{k=1}^r \Sigma_k^2 \left(1 - \frac{\sigma^4}{p^2 \Sigma_k^4}\right)_+ \right\}^2}{\left\{ \sum_{k=1}^r \Sigma_k^2 \right\} \left\{ \sum_{k=1}^r \Sigma_k^2 \left(1 + \frac{\sqrt{\alpha}\sigma^2}{p\Sigma_k^2}\right)\left(1 + \frac{\sigma^2}{p\Sigma_k^2\sqrt{\alpha}}\right) \right\}} + o_n(1)\,.$$

This theorem predicts a sharp phase transition: if $\sigma^2/p < \Sigma_1$, we can successfully extract information on M, from the observations N^E. If on the other hand $\sigma^2/p \ge \Sigma_1$, the observations are essentially useless in reconstructing M. To the best of our knowledge, this is the first sharp phase transition result for low rank approximation.

Further, note that the error predicted is sharp in the sense that it does not depend on unknown constants. This is in contrast with typical results in the literature. Let us consider a simple case for the matrix M. Let $m = n$ and let the rank $r = 1$ with $\Sigma_1 = 1$. In this case, the Theorem predicts the following result.

$$\frac{\|M - \widehat{M}\|_F}{\|M\|_F} = 1 - \left(1 - \frac{\sigma^2}{p}\right)_+^2 + o_n(1) \tag{4.28}$$

One way of understanding this result is by observing the distribution of singular

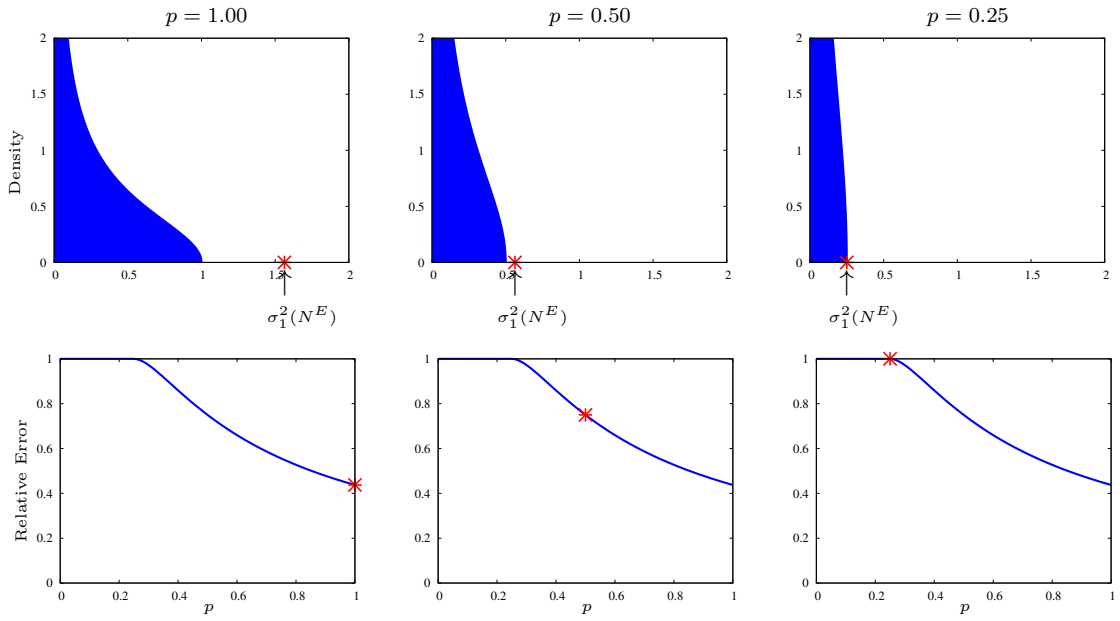

Figure 4.9: A demonstration of the effect of Theorem 4.3.1 for $r = 1$, $\Sigma_1 = 1$ and $\sigma = 0.5$. The top panel shows the distribution of the singular values of N^E for $p = 1, 0.5$ and 0.25. In each case, we have marked the position of the top singular value $\sigma_1(N^E)$. The bottom panel, illustrates the curve corresponding to the leading term in (4.28), namely, $1 - \left(1 - \frac{\sigma^2}{p}\right)_+^2$ as a function of p. The points corresponding to $p = 1.0, 0.5$ and 0.25 respectively are marked. At the threshold $p = \sigma^2$, we achieve the trivial relative error of 1.

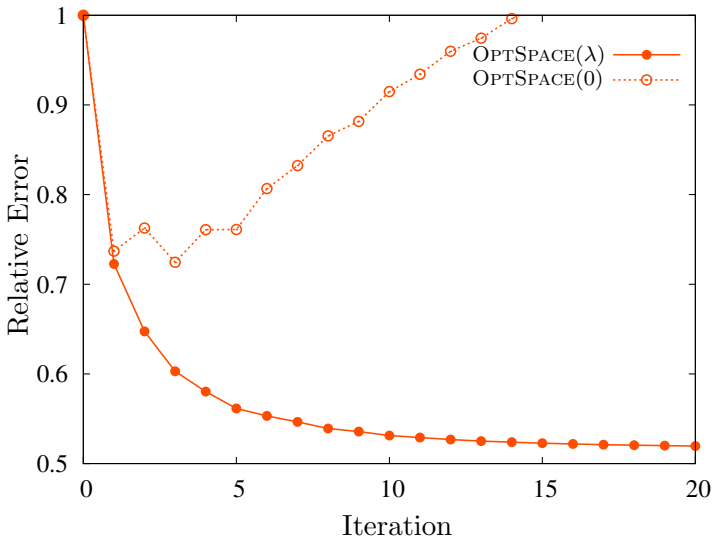

Figure 4.10: The relative error achieved by OptSpace as a function of the number of iterations of gradient descent for signal to noise ratio $\|M^E\|_F/\|W^E\|_F = 1$. OptSpace without regularization tends to overfit the model when the noise levels are high. Regularized OptSpace overcomes this problem. The regularization coefficient was chosen by cross validation.

values of N^E. In Figure 4.9, we have plotted the distribution of the singular values of N^E (top panel) for $p = 1, 0.5$ and 0.25. Here, we have taken $\sigma = 0.5$. In each case, we have marked the position of the top singular value $\sigma_1(N^E)$. In the bottom panel, we have plotted the curve corresponding to the leading term in (4.28), namely, $1 - \left(1 - \frac{\sigma^2}{p}\right)_+^2$ as a function of p. Again, we have marked the point on the curve corresponding to $p = 1.0, 0.5$ and 0.25 respectively. We see that as p decreases, the top singular value of N^E, corresponding to the "signal" edges closer to the rest of the distribution corresponding to the "noise". Consequently, the relative error achieved increases. Finally, when $p = 0.25$, the top singular value $\sigma_1(N^E)$ coincides with the rest of the distribution. This corresponds to the threshold discussed above. At this threshold, we achieve the trivial relative error of 1.

Let us test the validity of our initial motivation for regularizing OptSpace. In Figure 4.10, we revisit the example of Figure 4.7. We show the results of OptSpace for the case of signal to noise ratio $\|M^E\|_F/\|W^E\|_F = 1$. However, using the modified

OptSpace algorithm with a regularization coefficient λ chosen by cross validation, overcomes the problem of over-fitting. We refer to Section 6.3 for more extensive experiments and comparisons.

The proof of Theorem 4.3.1 relies on characterizing the singular values of $N^E = M^E + W^E$. Characterizing the eigen vectors and eigen values of random matrices and their low rank perturbations is a well studied topic [10, 96]. Consequently the proof draws upon some of these results. The next section contains a detailed description of the proof of Theorem 4.3.1. An important byproduct of the proof is that it provides a rule for choosing the regularization parameter λ, in the large system limit.

4.3.2 Proofs

The proof of Theorem 4.3.1 is based on the following three steps: (i) Obtain an explicit expression for the root mean square error in terms of right and left singular vectors of N; (ii) Estimate the effect of the noise W on the right and left singular vectors; (iii) Estimate the effect of zeroed entries. Step (ii) builds on recent estimates on the eigenvectors of large covariance matrices. In step (iii) we use the results of [59].

Step (i) is contained in Proposition 4.3.2 and is based on a simple linear algebra calculation. In Section 4.3.2, we analyze Step (ii) and Section 4.3.2 contains Step (iii)

Proposition 4.3.2. *Let $X_0 \in \mathbb{R}^{m\times r}$ and $Y_0 \in \mathbb{R}^{m\times r}$ be the matrices whose columns are the first r, right and left singular vectors of N^E. Let $\widehat{M}(\lambda) = X_0 S_0(\lambda) Y_0^T$ be the rank r matrix reconstructed by step 2 of* OptSpace*, with regularization parameter λ. Then there exists $\lambda_* > 0$ such that*

$$\|M - \widehat{M}(\lambda_*)\|_F^2 = \|\Sigma\|_F^2 - \left(\frac{\langle X_0^T(U\Sigma V^T)Y_0\,,\, X_0^T N^E Y_0\rangle}{\|X_0 N^E Y_0\|_F}\right)^2 .$$

Proof. We first show that $(X_0, Y_0; S_0 \equiv \frac{X_0^T N^E Y_0}{(1+\lambda)})$ minimizes $\widehat{\mathcal{F}}_E(X, Y; S)$. For any

triplet (X, Y, S),

$$\begin{aligned}
\widehat{\mathcal{F}}_E(X,Y;S) &= \frac{1}{2}\|\mathcal{P}_E(N) - XSY^T\|_F^2 + \frac{1}{2}\lambda\,\|S\|_F^2 \\
&= \frac{1}{2}\|N^E\|_F^2 + \frac{(1+\lambda)}{2}\|S\|_F^2 - \langle X^T N^E Y\,,\, S\rangle \\
&\geq \frac{1}{2}\|N^E\|_F^2 + \frac{(1+\lambda)}{2}\|S\|_F^2 - \|X^T N^E Y\|_F\|S\|_F \\
&\geq \frac{1}{2}\|N^E\|_F^2 + \frac{(1+\lambda)}{2}\|S\|_F^2 - \|X_0^T N^E Y_0\|_F\|S\|_F \\
&\geq \frac{1}{2}\|N^E\|_F^2 + \frac{(1+\lambda)}{2}\|S_0\|_F^2 - \|X_0^T N^E Y_0\|_F\|S_0\|_F \\
&= \widehat{\mathcal{F}}_E(X_0,Y_0;S_0).
\end{aligned}$$

If $\widehat{M}(\lambda) = X_0 S_0(\lambda) Y_0^T$ is the rank r matrix reconstructed by Step 2 of OptSpace, then

$$\begin{aligned}
\|M - \widehat{M}(\lambda)\|_F^2 &= \|U\Sigma V^T - X_0 S_0 Y_0^T\|_F^2 \\
&= \left(\|\Sigma\|_F^2 + \frac{\|X_0^T N^E Y_0\|_F^2}{(1+\lambda)^2} - 2\frac{\langle X_0^T(U\Sigma V^T)Y_0\,,\, X_0^T N^E Y_0\rangle}{(1+\lambda)}\right).
\end{aligned}$$

Optimizing the MSE over λ, we get $\lambda_* = \frac{\|X_0^T N^E Y_0\|_F^2}{\langle X_0^T(U\Sigma V^T)Y_0\,,\, X_0^T N^E Y_0\rangle} - 1$ and hence,

$$\|M - \widehat{M}(\lambda_*)\|_F^2 = \|\Sigma\|_F^2 - \left(\frac{\langle X_0^T(U\Sigma V^T)Y_0\,,\, X_0^T N^E Y_0\rangle}{\|X_0^T N^E Y_0\|_F}\right)^2$$

which completes the proof. □

The effect of noise

In order to isolate the effect of noise, we consider the matrix $\widehat{N} = p\,U\Sigma V^T + W^E$. Throughout this section we assume that the hypotheses of Theorem 4.3.1 hold.

Lemma 4.3.3. *Let $(z_{1,n}, \ldots, z_{r,n})$ be the r largest singular values of $\widehat{N}$. Then, as*

$n, m \to \infty$, $z_{i,n} \to z_i$ almost surely, where, for $\Sigma_i^2 > \sigma^2/p$,

$$z_i = p\Sigma_i \left\{ \left(\frac{\sigma^2}{p\Sigma_i^2} + \frac{1}{\sqrt{\alpha}} \right) \left(\frac{\sigma^2}{p\Sigma_i^2} + \sqrt{\alpha} \right) \right\}^{\frac{1}{2}} , \tag{4.29}$$

and $z_i = \sigma\sqrt{p\alpha^{-1/2}}(1+\sqrt{\alpha})$ for $\Sigma_i^2 \le \sigma^2/p$.

Further, let $X \in \mathbb{R}^{m\times r}$ and $Y \in \mathbb{R}^{n\times r}$ be the matrices whose columns are the first r, right and left, singular vectors of $\widehat{N}$. Then there exists a sequence of $r\times r$ orthogonal matrices Q_n such that, almost surely $\|U^T X - AQ_n\|_F \to 0$, $\|V^T Y - BQ_n\|_F \to 0$ with $A = \mathrm{diag}(a_1, \dots, a_r)$, $B = \mathrm{diag}(b_1, \dots, b_r)$ and

$$a_i^2 = \Big(1 - \frac{\sigma^4}{p^2\Sigma_i^4}\Big)\Big(1 + \frac{\sqrt{\alpha}\sigma^2}{p\Sigma_i^2}\Big)^{-1}, \qquad b_i^2 = \Big(1 - \frac{\sigma^4}{p^2\Sigma_i^4}\Big)\Big(1 + \frac{\sigma^2}{p\sqrt{\alpha}\Sigma_i^2}\Big)^{-1}, \tag{4.30}$$

for $\Sigma_i^2 > \sigma^2/p$, while $a_i = b_i = 0$ otherwise.

Proof. Notice that W^E is an $m \times n$ matrix with i.i.d. entries with variance $\sigma^2 p$ and fourth moment bounded by C/n^2. It is therefore sufficient to prove our claim for $p = 1$ and then rescale Σ by p and σ by $\sqrt{p}$. We will also assume that, without loss of generality, $m \ge n$.

Let $\widehat{Z}$ be an $r \times r$ diagonal matrix containing the eigenvalues $(z_{n,1}, \dots, z_{n,r})$. The eigenvalue equations read

$$U\beta_y + WY - X\widehat{Z} = 0\,, \tag{4.31}$$

$$V\beta_x + W^T X - Y\widehat{Z} = 0\,. \tag{4.32}$$

where we defined $\beta_x \equiv \Sigma\, U^T X$, $\beta_y \equiv \Sigma\, V^T Y \in \mathbb{R}^{r\times r}$. By singular value decomposition we can write $W = L\,\mathrm{diag}(w_1, w_2, \dots w_n) R^T$, with $L^T L = I_{m\times m}$, $R^T R = I_{n\times n}$.

Let u_i^T, x_i^T, v_i^T, $y_i^T \in \mathbb{R}^r$ be the i-th row of -respectively- $L^T U$, $L^T X$, $R^T V$, $R^T Y$.

In this basis equations (4.31) and (4.32) read

$$\begin{aligned} u_i^T\beta_y + w_i y_i^T - x_i^T\widehat{Z} &= 0\,, \qquad i\in[n]\,,\\ u_i^T\beta_y - x_i^T\widehat{Z} &= 0\,, \qquad i\in[m]\backslash[n]\,,\\ v_i^T\beta_x + w_i x_i^T - y_i^T\widehat{Z} &= 0\,, \qquad i\in[n]\,. \end{aligned}$$

These can be solved to get

$$\begin{aligned} x_i^T &= (u_i^T\beta_y\widehat{Z} + w_i v_i^T\beta_x)(Z^2 - w_i^2)^{-1}\,, \qquad i\in[n]\,,\\ x_i^T &= u_i^T\beta_y\widehat{Z}^{-1}\,, \qquad i\in[m]\backslash[n]\,,\\ y_i^T &= (v_i^T\beta_x\widehat{Z} + w_i u_i^T\beta_y)(\widehat{Z}^2 - w_i^2)^{-1}\,, \qquad i\in[n]. \end{aligned} \tag{4.33}$$

By definition $\Sigma^{-1}\beta_x = \sum_{i=1}^m u_i x_i^T$, and $\Sigma^{-1}\beta_y = \sum_{i=1}^n v_i y_i^T$, whence

$$\Sigma^{-1}\beta_x = \sum_{i=1}^n u_i(u_i^T\beta_y\widehat{Z} + w_i v_i^T\beta_x)(\widehat{Z}^2 - w_i^2)^{-1} + \sum_{i=n+1}^m u_i u_i^T\beta_y\widehat{Z}^{-1}\,, \tag{4.34}$$

$$\Sigma^{-1}\beta_y = \sum_{i=1}^n v_i(v_i^T\beta_x\widehat{Z} + w_i u_i^T\beta_y)(\widehat{Z}^2 - w_i^2)^{-1}. \tag{4.35}$$

Let $\lambda = w_i^2\alpha^{-1/2}/\sigma^2$. Then, it is a well known fact [96] that as $n\to\infty$ the empirical law of the λ_i's converges weakly almost surely to the Marcenko-Pastur law, with density $\rho(\lambda) = \alpha\sqrt{(\lambda - c_-^2)(c_+^2 - \lambda)}/(2\pi\lambda)$, with $c_\pm = 1\pm\alpha^{-1/2}$.

A priori, it is not clear that the sequence $(\beta_x, \beta_y, \widehat{Z})$ –dependent on n– converges. However, it is immediate to show that the sequence is tight, and hence we can restrict ourselves to a subsequence $\Xi\equiv\{n_i\}_{i\in\mathbb{N}}$ along which a limit exists. Eventually we will show that the limit does not depend on the subsequence, apart, possibly, from the rotation Q_n. Hence we shall denote the subsequential limit, by an abuse of notation, as (β_x, β_y, Z). Consider now such a convergent subsequence.

Let β_x^k and β_y^k denote the k^{th} columns of β_x and β_y respectively and Z_k denote $\widehat{Z}_{kk}$. We further define the following notations.

$$\begin{aligned} S_n(z) &= \Sigma^{-1} - \sum_{i=1}^{n} \frac{w_i u_i v_i^T}{z^2 - w_i^2} \\ P_n(z)\Lambda_n(z)P_n(z)^T &= \sum_{i=1}^{m} \frac{z u_i u_i^T}{z^2 - w_i^2} \\ Q_n(z)\Phi_n(z)Q_n(z)^T &= \sum_{i=1}^{n} \frac{z v_i v_i^T}{z^2 - w_i^2} \end{aligned}$$

where we set $w_i = 0$ for $i > n$.

Using these, the equations (4.35) read,

$$\begin{aligned} F_n(Z_k)\tilde{\beta_x}^k &= \tilde{\beta_y}^k \\ F_n^T(Z_k)\tilde{\beta_y}^k &= \tilde{\beta_x}^k \end{aligned} \tag{4.36}$$

where $F_n = \Lambda_n^{-1/2} P_n^T S_n Q_n \Phi^{-1/2}$, $\tilde{\beta_y}^k = \Lambda_n^{1/2} P_n^T \beta_y^k$ and $\tilde{\beta_x}^k = \Phi_n^{1/2} Q_n^T \beta_x^k$.

Hence $z_{n,k}$ are such that a singular value of F_n is 1. It is also easy to characterize the singular values of $F_n(z)$ as $n \to \infty$ for $z > z_c = \alpha^{1/2}\sigma^2 c_+(\alpha)^2$.

Remark 4.3.4. *Let $\sigma_n^k(z)$ be the k^{th} singular value of $F_n(z)$. Then, with high probability, as $n \to \infty$ and for $z > z_c$,*

$$\sigma_n^k(z) \to \left\{ \Sigma_k^2 \Big(z \int (z^2 - \alpha^{1/2}\sigma^2\lambda)^{-1} \rho(\lambda) \mathrm{d}\lambda + (\alpha - 1) z^{-1} \Big) \Big(z \int (z^2 - \alpha^{1/2}\sigma^2\lambda)^{-1} \, \rho(\lambda) \, \mathrm{d}\lambda \Big) \right\}^{-1/2}$$

Proof. Since almost surely as $n \to \infty$, $w_i^2 < \alpha^{1/2}\sigma^2 c_+(\alpha)^2 + \delta/2$ for all i, for all purposes the summands on the rhs of Eqs. (4.34), (4.35) can be replaced by uniformly continuous, bounded functions of the limiting eigenvalues λ_i. Further, each entry of u_i (resp. v_i) is just a single coordinate of the left (right) singular vectors of the random

matrix W. Using Theorem 1 in [10], it follows that w.h.p

$$\begin{aligned} P_n(z)\Lambda_n(z)P_n(z)^T &= \sum_{i=1}^{m} \frac{z u_i u_i^T}{z^2 - w_i^2} \\ &\to \int (Z^2 - \alpha^{1/2}\sigma^2\lambda)^{-1}\rho(\lambda)\mathrm{d}\lambda + (\alpha - 1)Z^{-1} \\ Q_n(z)\Phi_n(z)Q_n(z)^T &= \sum_{i=1}^{n} \frac{z v_i v_i^T}{n^2 z^2 - w_i^2} \\ &\to \int (Z^2 - \alpha^{1/2}\sigma^2\lambda)^{-1}\rho(\lambda)\mathrm{d}\lambda \end{aligned}$$

We now show that w.h.p $\left(\sum_{i=1}^{n} \frac{w_i u_i v_i^T}{z^2 - w_i^2}\right)_{kl} \to 0$. Note that this implies the required result.

$$\begin{aligned} \left(\sum_{i=1}^{n} \frac{w_i u_i v_i^T}{z^2 - w_i^2}\right)_{kl} &= \sum_{i=1}^{n} \frac{\bar{u}_k^T l_i w_i r_i^T \bar{v}_l}{z^2 - w_i^2} \\ &= \sum_{j=0}^{\infty} \frac{\bar{u}_k^T W (W^T W)^j \bar{v}_l}{z^{2j+2}} \\ &\equiv \sum_{j=0}^{\infty} a_j \end{aligned}$$

Let $w_{\max}$ be the largest singular value of W. Note that $|a_j| \leq C(w_{\max}/z)^{2j}$ and $z > z_c$. Therefore for n large enough $(w_{\max}/z) < 1$. Hence, it is enough to show that whp $a_j \to 0$ as $n \to \infty$. We do this by computing $\mathbb{E}[a_j^4]$.

$$\mathbb{E}[a_j^4] = \sum \mathbb{E}[\Pi_{i=1}^{4} \bar{u}_{kp_i}(W(W^T W)^j)_{p_i q_i} \bar{v}_{l q_i}]/(z)^{8j+8}$$

where the summation is over indices $p_i \in [n]$ and $q_i \in [m]$. The most general case is when none of the p_i and q_i are equal to one another. In this case,

$$\mathbb{E}[\Pi_{i=1}^{4}(W(W^TW))_{p_iq_i}] \leq C_1 n^{4j-4}\mathbb{E}[W_1 1^{8j+4}] \leq Cn^{4j-4+4j+2}$$

and hence

$$\begin{aligned}\sum \mathbb{E}[\Pi_{i=1}^{4}\bar{u}_{kp_i}(W(W^TW)^j)_{p_iq_i}\bar{v}_{lq_i}]/(z)^{8j+8} &\leq \sum C\Pi_{i=1}^{4}|\bar{u}_{kp_i}\bar{v}_{lq_i}|/n^{10} \\ &\leq C(\sum_{i=1}^{n}\bar{u}_{ki})^4(\sum_{i=1}^{n}\bar{v}_{li})^4/n^{10} \\ &\leq C/n^2\end{aligned}$$

where the first two sums above is only over values such that none of p_i and q_i are equal. The other cases are handled similarly. We finally obtain $\mathbb{E}[a_j^4] \leq C/n^2$ for some constant C. Hence by Borel Cantelli Lemma, we have that whp $a_j \to 0$ as $n \to \infty$.

□

Define sets $\mathcal{I} = \{k \in [r] : \Sigma_k^2 > \sigma^2\}$ and $\mathcal{J} = [r]\backslash\mathcal{I}$. Since $\sigma_n^k(z)$ is monotonic in $z > z_c$ and roots of $\sigma^k(z) = 1$ exists in $z > z_c$ for $k \in \mathcal{I}$, we have that $z_{nk} \to \bar{z}_k$ for $k \in \mathcal{I}$ where $\bar{z}_k$ is the root of $\sigma^k(z) = 1$ in $z > z_c$. Further, for $k \in \mathcal{J}$, no root of $\sigma^k(z) = 1$ exists in $z > z_c$ and hence limsup $z_{nk} \leq z_c$. But we know that liminf $z_{nk} \geq z_c$. Combining these, we get that the eigenvalues are uniquely determined (independent of the subsequence) and given by Eq. (4.29).

In order to determine β_x and β_y first observe that, since $I_{r\times r} = Y^TY = \sum_{i=1}^{n} y_i y_i^T$, we have, using Eq. (4.33)

$$I_{r\times r} = \sum_{i=1}^{n}(\widehat{Z}^2 - w_i^2)^{-1}(\widehat{Z}\beta_x^T v_i + w_i\beta_y^T u_i)(v_i^T\beta_x\widehat{Z} + w_i u_i^T\beta_y)(\widehat{Z}^2 - w_i^2)^{-1}\,.$$

We find it useful to define the following matrices.

$$\begin{aligned}
A_1(z_1,z_2) &= \sum_{i=1}^{n} \frac{z_1 z_2 v_i v_i^T}{(z_1^2-w_i^2)(z_2^2-w_i^2)} \\
A_2(z_1,z_2) &= \sum_{i=1}^{n} \frac{w_i z_2 u_i v_i^T}{(z_1^2-w_i^2)(z_2^2-w_i^2)} \\
A_3(z_1,z_2) &= \sum_{i=1}^{n} \frac{w_i z_1 v_i u_i^T}{(z_1^2-w_i^2)(z_2^2-w_i^2)} \\
A_4(z_1,z_2) &= \sum_{i=1}^{n} \frac{w_i^2 u_i u_i^T}{(z_1^2-w_i^2)(z_2^2-w_i^2)}
\end{aligned}$$

Using these, the above equations can be written as

$$\delta_{kl} = (\beta_x^k)^T A_1(z_k,z_l)\beta_x^l + (\beta_y^k)^T A_2(z_k,z_l)\beta_x^l + (\beta_x^k)^T A_3(z_k,z_l)\beta_y^l + (\beta_y^k)^T A_4(z_k,z_l)\beta_y^l$$

Further, using $\beta_y^k = P_n \Lambda_n^{-1} P_n^T S_n \beta_x^k \equiv G_n(z_k)\beta_x^k$,

$$\begin{aligned}
\delta_{kl} &= (\beta_x^k)^T A_1(z_k,z_l)\beta_x^l + (\beta_x^k)^T G(z_k)^T A_2(z_k,z_l)\beta_x^l \\
&\quad + (\beta_x^k)^T A_3(z_k,z_l)G(z_l)\beta_y^l + (\beta_y^k)^T G(z_k)^T A_4(z_k,z_l)G(z_l)\beta_y^l
\end{aligned}$$

where we have dropped the n for notational convenience.

We also know that (through calculations similar to those in the proof of Remark 4.3.4) as $n \to \infty$ and for $z_1, z_2 > z_c$,

$$\begin{aligned}
A_1(z_1,z_2) &\to \int \frac{z_1 z_2}{(z_1^2-\alpha^{1/2}\sigma^2\lambda)(z_2^2-\alpha^{1/2}\sigma^2\lambda)}\rho(\lambda)\mathrm{d}\lambda \equiv f_1(z_1,z_2), \\
A_4(z_1,z_2) &\to \int \frac{\alpha^{1/2}\sigma^2\lambda}{(z_1^2-\alpha^{1/2}\sigma^2\lambda)(z_2^2-\alpha^{1/2}\sigma^2\lambda)}\rho(\lambda)\mathrm{d}\lambda
\end{aligned}$$

and hence

$$\begin{aligned} G(z_1)^T A_4(z_1,z_2)G(z_2) &\to \Sigma^{-2} g(z_1)g(z_2) \int \frac{\alpha^{1/2}\sigma^2\lambda}{(z_1^2-\alpha^{1/2}\sigma^2\lambda)(z_2^2-\alpha^{1/2}\sigma^2\lambda)}\rho(\lambda)\mathrm{d}\lambda \\ &\equiv f_2(z_1,z_2), \end{aligned}$$

$A_2(z_1,z_2) \to 0$, and $A_3(z_1,z_2) \to 0$. Here $g(z) = \left(z\int(z^2-\alpha^{1/2}\sigma^2\lambda)^{-1}\mathrm{d}\lambda + (\alpha-1)z^{-1}\right)^{-1}$.

First consider $k,l \in \mathcal{I}$. Also, let B_{kl} denote $(\beta_x^k)^T\beta_x^l$. Then,

$$\delta_{kl} = (f_1(z_k,z_l)+f_2(z_k,z_l))B_{kl} + E_n$$

where $|E_n| \le |e_n|(B_{kk}B_{ll})^{1/2}$ and $|e_n| \to 0$ as $n \to \infty$. Thus $B_{kk} \to (f_1+f_2)^{-1}$. Substituting the value of z_k and solving, we get that $B_{kk} \to \Sigma_k^2 a_k^2$. Further, since $f_1(z_k,z_l), f_2(z_k,z_l)$ are between $f_1(z_k,z_k), f_2(z_k,z_k)$ and $f_1(z_l,z_l), f_2(z_l,z_l)$, we get that $B_{kl} \to 0$ as $n \to \infty$ for $k \neq l$.

Now consider $k \in \mathcal{J}$.

$$\begin{aligned} 1 &= \sum_{i=1}^{n}\left(\frac{z_c(v_i^T\beta_x^k)+w_i(u_i^T\beta_x^k)}{z_c^2-w_i^2}\right)^2 \\ &\ge \sum_{i=1}^{n}\left(\frac{z_c(v_i^T\beta_x^k)+w_i(u_i^T\beta_x^k)}{(z_c+\delta)^2-w_i^2}\right)^2 \end{aligned}$$

for some $\delta > 0$. Therefore,

$$1 \ge \left((\frac{z_c}{z_c+\delta})^2 f_1(z_c+\delta,z_c+\delta) + f_2(z_c+\delta,z_c+\delta)\right)B_{kk} + E_n$$

where $|E_n| \le |e_n|B_{kk}$ and $|e_n| \to 0$ as $n \to \infty$. Further, for any $\epsilon > 0$, there is a $\delta > 0$ such that

$$\left((\frac{z_c}{z_c+\delta})^2 f_1(z_c+\delta,z_c+\delta) + f_2(z_c+\delta,z_c+\delta)\right) > 1/\epsilon$$

and hence $0 \le$ limsup $B_{kk} \le \epsilon$ for any $\epsilon > 0$ which implies $B_{kk} \to 0$. Further,

$B_{kl} \leq (B_{kk}B_{ll})^{1/2}$ implies that $B_{kl} \to 0$.

Combining the above results we get that in the limit $\beta_x^T \beta_x = \mathrm{diag}(\Sigma_1^2 a_1^2, \dots \Sigma_r^2 a_r^2)$ and using equations (4.36), we get that $\beta_y^T \beta_y = \mathrm{diag}(\Sigma_1^2 b_1^2, \dots \Sigma_r^2 b_r^2)$.

Further from equations (4.36), β_x and β_y are block diagonal with blocks in correspondence with the degeneracy pattern of Σ.Since $\beta_x^T \beta_x = C_x$ and $\beta_y^T \beta_y = C_y$ are diagonal, with the same degeneracy pattern, it follows that, inside each block of size d, each of β_x and β_y is proportional to a $d \times d$ orthogonal matrix. Therefore $\beta_x = \Sigma A Q_s$, $\beta_y = \Sigma B Q'_s$, for some orthogonal matrices Q_s, Q'_s. Also, using equation (4.36) one can prove that $Q_s = Q'_s$.

Notice, by the above argument A, B are uniquely fixed by our construction. On the other hand Q_s might depend on the subsequence Ξ. Since our statement allows for a sequence of rotations Q_n, that depend on n, the eventual subsequence dependence of Q_s can be factored out. □

It is useful to point out a straightforward consequence of the above.

Corollary 4.3.5. *There exists a sequence of orthogonal matrices $Q_n \in \mathbb{R}^{r\times r}$ such that, almost surely,*

$$\lim_{n\to\infty} \left|\left| X^T U \Sigma V^T Y - Q_n D Q_n^T \right|\right|_F \;=\; 0\,, \tag{4.37}$$

with $D = \mathrm{diag}(\Sigma_1 a_1 b_1, \dots, \Sigma_r a_r b_r)$.

The effect of missing entries

The proof of Theorem 4.3.1 is completed by establishing a relation between the singular vectors X_0, Y_0 of N^E and the singular vectors X and Y of $\widehat{N}$.

Lemma 4.3.6. *Let $k \leq r$ be the largest integer such that $\Sigma_1 \geq \cdots \geq \Sigma_k > \sigma^2/p$, and denote by $X_0^{(k)}$, $Y_0^{(k)}$, $X^{(k)}$, and $Y^{(k)}$ the matrices containing the first k columns of X_0, Y_0, X, and Y, respectively. Let $X_0^{(k)} = X^{(k)} S_x + X_\perp^{(k)}$, $Y_0^{(k)} = Y^{(k)} S_y + Y_\perp^{(k)}$ where $(X_\perp^{(k)})^T X^{(k)} = 0$, $(Y_\perp^{(k)})^T Y^{(k)} = 0$ and $S_x, S_y \in \mathbb{R}^{r\times r}$. Then there exists a numerical*

constant $C = C(\Sigma_i, \sigma^2, \alpha, M_{\max})$*, such that, with high probability,*

$$\|X_\perp^{(k)}\|_F^2, \|Y_\perp^{(k)}\|_F^2 \le Cr\sqrt{\frac{1}{n}}\,, \tag{4.38}$$

with probability approaching 1 *as* $n \to \infty$*.*

Proof. We will prove our claim for the right singular vector Y, since the left case is completely analogous. Further we will drop the superscript k to lighten the notation.

We start by noticing that $\|N^E Y_0\|_F^2 = \sum_{a=1}^k (z_{a,n})^2$, where $nz_{a,n}$ are the singular values of N^E. Using Lemma 3.2 in [59] which bounds $\|M^E - pM\|_2 = \|N^E - \widehat{N}\|_2$, we get

$$\|N^E Y_0\|_F^2 \;\ge\; \sum_{a=1}^{k} (z_{a,n} - CM_{\max}\sqrt{pn})^2\ . \tag{4.39}$$

On the other hand $\|N^E Y_0\|_F \le \|\widehat{N}Y_0\|_F + \|N^E - \widehat{N}\|_2 \|Y_0\|_F$. Further by letting $S_y = L_y \Theta_y R_y^T$, for L_y, R_y orthogonal matrices, we get $\|\widehat{N}Y_0\|_F^2 = \|\widehat{N}YL_y\Theta_y\|_F^2 + \|\widehat{N}Y_\perp\|_F^2$. Since $Y_0^T Y_0 = I_{k\times k}$, we have $I_{k\times k} = R_y\Theta_y^T\Theta_y R_y^T + Y_\perp^T Y_\perp$, and therefore

$$\begin{aligned}
\|\widehat{N}Y_0\|_F^2 &= \|\widehat{N}YL_y\|_F^2 - \|\widehat{N}YL_yR_y^TY_\perp^T\|_F^2 + \|\widehat{N}Y_\perp\|_F^2 \\
&\le \sum_{a=1}^{k} z_{a,n}^2 - z_{k,n}^2\|Y_\perp\|_F^2 \\
&\quad + p\sigma^2\alpha^{-1/2}(c_+(\alpha)+\delta)\|Y_\perp\|_F^2 \\
&= n^2\sum_{a=1}^{k} z_{a,n}^2 - n^2 e_y\|Y_\perp\|_F^2\,,
\end{aligned}$$

where $e_y \equiv z_{k,n}^2 - p\sigma^2\alpha^{-1/2}(c_+(\alpha)+\delta)$, and used the inequality $\|\widehat{N}Y_\perp\|_F^2 \le n^2 p\sigma^2\alpha^{-1/2}(c_+(\alpha)+\delta)\|Y_\perp\|_F^2$ which holds for all $\delta > 0$ asymptotically almost surely as $n \to \infty$ (by an immediate generalization of Lemma 4.3.3). It is simple to check that $\Sigma_k \ge \sigma^2/p$ implies $e_y > 0$.

Using triangular inequality, Lemma 3.2 in [59], we get

$$\begin{aligned}\|NY_0\|_F^2 &\leq \sum_{a=1}^{r} z_{a,n}^2 - e_y\|Y_\perp\|_F^2 + Cnp\alpha^{3/2}M_{\max}^2 r \\ &\quad +2C\sqrt{np}\alpha^{3/4}M_{\max}\sqrt{r}\|z\|\,,\end{aligned}$$

which, combined with equation (4.39), implies the thesis. □

Proof of Theorem 4.3.1. We now turn to upper bounding the right hand side of Eq. (4.28). Let k be defined as in the last lemma. Notice that by Lemma 4.3.3, $X^T(U\Sigma V^T)Y$ is well approximated by $(X^{(k)})^T(U\Sigma V^T)Y^{(k)}$. Analogously, it can be proved that $X_0^T(U\Sigma V^T)Y_0$ is well approximated by $(X_0^{(k)})^T(U\Sigma V^T)Y_0^{(k)}$. Due to space limitations, we will omit this technical step and thus focus here on the case $k = r$ (equivalently, neglect the error incurred by this approximation).

Using Lemma 4.3.6 to bound the contribution of $X_\perp, Y_\perp$, we have

$$\begin{aligned}&\langle X_0^T(U\Sigma V^T)Y_0\,,\, X_0^T N^E Y_0\rangle \\ &= \langle S_x^T X^T(U\Sigma V^T)Y S_y\,,\, X_0^T N^E Y_0\rangle(1+o_n(1)) \\ &= \langle X^T(U\Sigma V^T)Y\,,\, S_x^T X_0^T N^E Y_0 S_y\rangle(1+o_n(1))\,. \end{aligned} \tag{4.40}$$

Further $X_0^T N^E Y_0 = X_0^T\widehat{N}Y_0 + X_0^T(N^E - \widehat{N})Y_0$ and, using once more the bound in Lemma 3.2 of [59], that implies $|X_0^T(N^E - \widehat{N})Y_0| \leq Cr\sqrt{nrp}$, we get

$$\begin{aligned}S_x^T X_0^T N^E Y_0 S_y &= L_x\Theta_x^2 L_x^T X^T\widehat{N}Y R_y\Theta_y^2 R_y^T + E_1 \\ &= Z + E_2\,,\end{aligned}$$

where we recall that Z is the diagonal matrix with entries given by the singular values of $\widehat{N}$, and $\|E_1\|_F^2, \|E_2\|_F^2 \leq C(p,r)\sqrt{n}$. Using this estimate in Eq. (4.40), together with the result in Lemma 4.3.3, we finally get

$$\frac{\langle X_0^T(U\Sigma V^T)Y_0\,,\, X_0^T N^E Y_0\rangle}{\sqrt{mn}\|X_0^T N^E Y_0\|_F^2} \geq \frac{\sum_{k=1}^r \Sigma_k a_k b_k z_k}{\sqrt{\alpha}\|z\|} - o_n(1)\,,$$

which implies the thesis after simple algebraic manipulations □

Chapter 5

Alternating Least Squares

Spectral techniques such as those discussed in Chapter 4 have been the subject of numerous studies [4, 8, 59], and are relatively well understood. In particular [82, 62] shows that the estimation error achieved by OptSpace is minimax optimal up to a multiplicative constant over the class of matrices with bounded rank and bounded entries. Despite the theoretical understanding, spectral methods are considered sub-optimal in practice, and only marginally useful. A more powerful idea is to minimize the empirical risk

$$\mathcal{R}_E(X, Y) \equiv \sum_{(i,j)\in E} \left(XY^T - N\right)_{ij}^2 . \tag{5.1}$$

over factors $X \in \mathbb{R}^{m\times r}$, $Y \in \mathbb{R}^{n\times r}$. Indeed this approach was pursued in a number of papers that developed greedy optimization strategies and various improvements on the risk function [100, 99, 101, 90, 89, 91, 103, 93, 51]. Most importantly, minimization of the nonconvex risk function (5.1) is at the core of award-winning collaborative filtering methods [12, 63].

Unfortunately our theoretical understanding of such optimization-based methods is far inferior to the one of spectral methods. In this chapter, we introduce a simple algorithm called Alternating Least Squares that alternately minimizes the risk function $\mathcal{R}_E(X, Y)$ over X and Y and provide the first rigorous analysis for this

problem. We show that the estimation error achieved by Alternating Least Squares is close to optimal. Further, we show that the convergence is exponential in the number of iterations.

Although alternate minimization shows remarkable convergence properties, we develop a second iterative algorithm based on message passing updates. This is achieved by a simple modification of the alternating least squares method. In Chapter 6, we carry out extensive numerical simulations on synthetic and real collaborative filtering datasets, and demonstrate that message passing converges faster than alternate minimization. Indeed, within our experience, this is the fastest available algorithm for low rank matrix completion with collaborative filtering datasets.

The rest of the chapter is organized as follows. In Section 5.1, we describe the details of the algorithm. In Section 5.2, we present analytical results showing the convergence of the algorithm. The proofs of these results are described in Section 5.4. Finally, in Section 5.3, we introduce the Message Passing algorithm as a modification to Alternating Least Squares and discuss some implementation ideas for the new algorithm.

5.1 Algorithm

We begin by a description of the Alternate Least Squares algorithm. Alternating least squares was first developed in [12, 51], without however establishing performance guarantees. Indeed, it is a most natural approach and is widely employed in practice. The basic idea is to iteratively minimize the risk Eq. (5.1) over X and over Y.

Rigorous analysis of iterative algorithms on random structures is particularly challenging because the randomness is unchanged across iterations, and hence the resulting random process is non-Markov. The standard way to deal with this problem is to prove convergence under specific deterministic conditions, and then show that those conditions hold with high probability. The resulting analysis is not always tight and –for the present problem– it appears extremely difficult to define appropriate deterministic conditions on the set E of observed entries.

We propose here a a modification of the algorithm that opens the way to rigorous analysis, and appears to be of independent interest. At each iteration, instead of making use of all the available data, we sample a random subset of the data (i.e. a subset of the entries E). This subset is never used in the following iterations, thus making iterations statistically independent. This trick is somewhat analogous to the one of [49] although the objective is quite different (in [49] a similar idea is developed to construct a dual witness).

First of all, we partition the set of observed entries E into $2t_{\max}+1$ subsets of roughly equal size. This can be done by assigning to every $(i,j) \in E$, a uniformly random label in $[2t_{\max}+1]$ and collecting all (i,j) with the same label into a set. We denote these subsets by E^0, and by E_1^t, E_2^t for $t = 1, 2, \dots, t_{\max}$. Here $t_{\max}$ is the maximum number of alternating minimization iterations run in the algorithm. Each iteration comprises a minimization over X and a minimization over Y. This modification greatly simplifies the algorithm properties, since each iteration uses fresh independent information.

A key point is that we will be able to prove exponentially fast convergence of alternate minimization to the target performances. As a consequence, it is sufficient to consider $t_{\max} = C \log n$ for a suitable constant C. In other words, using a different subset of the data at each iteration implies only a modest (logarithmic) 'thinning' of the data.

In order to initialize the alternate least squares procedure, we use the data in the set E^0 and perform a singular value decomposition (SVD) of the matrix $\mathcal{P}_{E^0}(N)$. Denote by $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(m,n)}$ the singular values of $\mathcal{P}_{E^0}(N)$ and by $X_i \in \mathbb{R}^m$, $Y_i \in \mathbb{R}^n$,$i \in \{1, \dots, \min(m,n)\}$ the corresponding singular vectors. We then have

$$\mathcal{P}_{E^0}(N) = \sum_{i=1}^{\min\{m,n\}} \sigma_i \, X_i Y_i^T.$$

We use this decomposition to initialize the alternate least squares iteration. Namely, let $X^{(t)} = [X_1^{(t)}|\cdots|X_r^{(t)}]$, $Y^{(t)} = [Y_1^{(t)}|\cdots|Y_r^{(t)}]$, be the estimates of the right and left

ALTERNATING LEAST SQUARES

Input : set E, matrix $\mathcal{P}_E(N)$, number of iterations $t_{\max}$, rank r
Output : estimation $\widehat{M}$

0: Partition $E = E^0 \cup \mathbb{E}_1^1 \cup E_2^1 \cup \cdots \cup \mathbb{E}_1^{t_{\max}} \cup E_2^{t_{\max}}$

1: Compute the SVD $\mathcal{P}_{E^0}(N) = \sum_{i=1}^{\min(m,n)} \sigma_i X_i Y_i^T$
Initialize $(X^{(0)}, Y^{(0)})$ by letting, for $i \in [r]$
$X_i^{(0)} = \frac{mn}{|E^0|}\sqrt{\sigma_i} x_i$ and $Y_i^{(0)} = \frac{mn}{|E|}\sqrt{\sigma_i} y_i$

2: **for** $t = 1, 2, \ldots t_{\max}$ **do** :
$X^{(t)} = \operatorname{argmin}_{X \in \mathbb{R}^{n\times r}} \|\mathcal{P}_{E_1^t}(XY^{(t-1)T} - N)\|_F^2$
$Y^{(t)} = \operatorname{argmin}_{Y \in \mathbb{R}^{m\times r}} \|\mathcal{P}_{E_1^t}(X^{(t)}Y^T - N)\|_F^2$
end for

3: Output $X^{(t_{\max})}Y^{(t_{\max})T}$

Figure 5.1: A summary of the ALTERNATING LEAST SQUARES algorithm

factors after t iterations of alternate minimization. Then, we set, for $i \in [r]$,

$$X_i^{(0)} = \frac{mn}{|E^0|}\sqrt{\sigma_i}\, X_i\,, \qquad Y_i^{(0)} = \frac{mn}{|E^0|}\sqrt{\sigma_i}\, Y_i\,.$$

Figure 5.1 below summarizes the algorithm. Each iteration of the algorithm amounts to solving two least squares problems of dimension mr and nr. Of course, this can be generally achieved in time $O((mr)^2, (nr)^2)$. In the present case, the least square problem is separable and hence the complexity is reduced to $O(|E|r^2)$. In order to see this, we will denote by $x_1^{(t)}, \ldots, x_m^{(t)} \in \mathbb{R}^r$ the rows of $X^{(t)}$ and by $y_1^{(t)}, \ldots, y_m^{(t)} \in \mathbb{R}^r$ the rows of $Y^{(t)}$ (always seen as column vectors). It is easy to see that the minimizations in step 2 above are solved by letting, for all $i \in [n]$,

$$x_i^{(t)} = \Big(\sum_{j\in\partial i} y_j^{(t-1)} \otimes y_j^{(t-1)}\Big)^{-1}\Big(\sum_{j\in\partial i} N_{ij} y_j^{(t-1)}\Big)\,, \tag{5.2}$$

$$y_j^{(t)} = \Big(\sum_{l\in\partial j} x_l^{(t)} \otimes x_l^{(t)}\Big)^{-1}\Big(\sum_{l\in\partial j} N_{lj} x_l^{(t)}\Big)\,, \tag{5.3}$$

Here, for $i \in [m]$, we let $\partial i = \{j \in [n] \ : \ (i,j) \in E_1^t\}$, and, for $j \in [m]$, $\partial j = \{i \in$

$[m] : (i,j) \in E_2^t\}$. Further, we used the notation $a \otimes b \equiv ab^T$.

Summarizing, the algorithm presented here differs from standard implementations of alternate least squares in two ways. First, the initialization is obtained from the singular value decomposition of $\mathcal{P}_{E^0}(N)$, with E^0 a subset of the observed entries. Second, at each iteration a distinct subset E^t of entries is used. These modifications open the way to our rigorous analysis. However, they are also of independent interest from a computational point of view:

1. The initialization through SVD can accelerate convergence (and highly optimized algorithms exist for SVD).

2. Using a sparsified subset of entries per iteration does reduce the per-iteration complexity.

5.2 Main Results

In this section, we formally state our rigorous results for the Alternating Least Squares algorithm. Let us begin by recalling the model definition. We let N denote an $m \times n$ matrix which is 'approximately' low rank, that is

$$N = M + W = U\Sigma V^T + W\,.$$

where U has dimensions $m \times r$, V has dimensions $n \times r$ and Σ has dimensions $r \times r$. In this chapter, we subsume Σ into either U or V and hence, we will work with the model

$$N = M + W = UV^T + W\,. \tag{5.4}$$

For $X, U \in \mathbb{R}^{n\times r}$ and $\gamma \in \mathbb{R}_+$, let us introduce the following notion of distance

$$\begin{aligned} d_\gamma(X,U) &\equiv \min_{\alpha\in\mathbb{R}^{r\times r}} \left\{\|U - X\alpha\|_{2,2} + \gamma \left\|U - X\alpha^T\right\|_{2,\infty}\right\} \\ &= \min_{\alpha\in\mathbb{R}^{r\times r}} \left\{\|U - X\alpha\|_{2,2} + \gamma \max_{i\in[n]} \|u_i - \alpha x_i\|_2\right\}. \end{aligned} \tag{5.5}$$

Notice that $d_\gamma(\,\cdot\,,\,\cdot\,)$ is not symmetric in its arguments, and hence is not an actual distance. Also, it depends on X only through the linear subspace of $\mathbb{R}^n$ spanned by its columns.

5.2.1 Assumptions

We will assume that the low rank matrix UV^T is incoherent, along the lines of Section 3.2, and further that it has bounded condition number. However since we use a slightly different model here, the parameters involved are different from those described before.

As for the perturbation W we generically describe it as the sum of a deterministic component plus terms that are i.i.d. across entries. Correlations between distinct entries can be implicitly absorbed in the deterministic component. Our assumption will be quantified by five parameters that will enter explicitly in our performances estimates:

$$\mu\,, \quad \kappa\,, \quad \|\overline{W}\|_2\,, \quad \theta\,, \quad \omega\,.$$

These are formally defined as follows:

Incoherence We will define the incoherence parameter μ_0 as

$$\begin{aligned} \mu_0 &\equiv \max\left\{ \frac{\|U\|_{2,\infty}^2}{r\|U\|_2^2}, \frac{\|V\|_{2,\infty}^2}{r\|V\|_2^2} \right\} \\ &= \max_{i\in[m],j\in[n]} \left\{ \frac{m\|u_i\|_2^2}{r\|U\|_2^2}\,,\ \frac{n\|v_j\|_2^2}{r\|V\|_2^2} \right\}. \end{aligned}$$

Our results are most useful when $\mu_0 = O(\log n)$. Note that we do not need the second incoherence assumption A1. We will state our results in terms of $\mu = \max\{\mu_0\,, \log m, \log n\}$. It is easy to show that the incoherence parameter μ_0 defined above is bounded above by the incoherence parameter defined in Section 3.2.

Condition number We let

$$\kappa \equiv \max\{\sigma_{\min}(U)^{-1}, \sigma_{\min}(V)^{-1}\}\,. \tag{5.6}$$

Notice that the condition number of the low rank component UV^T is lower bounded by κ. Hence the bounds obtained for Alternating Least Squares are stronger than those obtained in Chapter 4 for the same dependence on κ.

Deterministic perturbation We assume W_{ij} be independent random variables with expectation $\overline{W}_{ij}$. This deterministic perturbation will be characterized by two parameters:

$$\begin{aligned} \theta_2 &\equiv \|\overline{W}\|_{2,2} = \sigma_{\max}(\overline{W})\,, \\ \theta_\infty &\equiv \max\{\|\overline{W}\|_{2,\infty}, \|\overline{W}^T\|_{2,\infty}\} \end{aligned}$$

where $\|A\|_{2,\infty} \equiv \max_{i\in[m]} \sqrt{\sum_{j\in[n]} A_{ij}^2}$.

Random perturbation The random components $(W_{ij} - \overline{W}_{ij})$ will be independent sub-Gaussian random variables with common parameter ω. Explicitly

$$\mathbb{P}\left(|W_{ij} - \overline{W}_{ij}| \geq t\right) \leq 2\exp\left(-\frac{t^2}{2\omega^2}\right)\,.$$

We make the following two assumptions on the noise structure. For a sufficiently large constant C:

$$\theta_\infty^2 \leq \frac{\mu r}{n}\,\theta_2^2\,, \tag{5.7}$$

$$\theta_2 + \left(\frac{n\omega}{\sqrt{\epsilon}}\right) \leq \frac{1}{8C\kappa^3}\,. \tag{5.8}$$

The first assumption states that the deterministic perturbation is not concentrated on a small subset of the rows or columns. The second condition is a bound on the maximum perturbation level.

Finally, with the above definitions, we will make a special choice of the parameter

γ entering Eq. (5.5), and we will let

$$d(X,U) = d_{(n/\mu r)^{1/2}}(X,U) \equiv \min_{\alpha\in\mathbb{R}^{r\times r}} \left\{ \|U - X\alpha\|_{2,2} + \sqrt{\frac{n}{\mu r}}\, \|U - X\alpha^T\|_{2,\infty} \right\}.$$

5.2.2 Statement

We now state our analytical results concerning the performance of Alternating Least Squares. We first bound the deviation between the estimated factors X^t and Y^t with the original factors U and V using the metric defined earlier in the section, namely $d_\gamma(\cdot)$. This will imply a bound on the deviation $\|M - \widehat{M}^t\|_F$.

Theorem 5.2.1. *There exist universal constants C_1, C_2 such that the following happens under the assumptions of the previous section. Setting $t_{\max} = C_1 \log n$, let (X^t, Y^t) be the factors estimated by* Alternating Least Squares *after $t \le t_{\max}$ iterations. If*

$$|E| \ge C_2 n \kappa^8 \mu r (\log n)^2 , \tag{5.9}$$

then, with probability larger than $1 - 1/n^4$, we have

$$d(X^t, U) \le \frac{1}{2^{2t}} + C_2\kappa^2 \left(\theta_2 + \frac{n\omega}{\sqrt{\epsilon}}\right), \qquad d(Y^t, Y) \le \frac{1}{2^{2t}} + C_2\kappa^2 \left(\theta_2 + \frac{n\omega}{\sqrt{\epsilon}}\right). \tag{5.10}$$

Further, letting $M \equiv UV^T$ denote the unknown rank r matrix and $\widehat{M}^t = X^t(Y^t)^T$ be its estimate after t iterations, we have the following bound on the relative root mean square error, holding with probability larger than $1 - 1/n^4$,

$$\|M - \widehat{M}^t\|_F \le \frac{6\sqrt{r}}{2^{2t}} + C_2\sqrt{r}\,\kappa^2 \left(\theta_2 + \frac{n\omega}{\sqrt{\epsilon}}\right). \tag{5.11}$$

Theorem 5.2.1 is the analogue of Theorem 4.2.2 for OptSpace. However, a few key differences are worth noting. First, Theorem 4.2.2 dealt with the more widely used notion of reconstruction, namely exact reconstruction. On the other hand, Theorem 5.2.1 deals with reconstruction up to an error smaller than n^{-C} for any arbitrary C. The two notions of reconstruction are of course different. However,

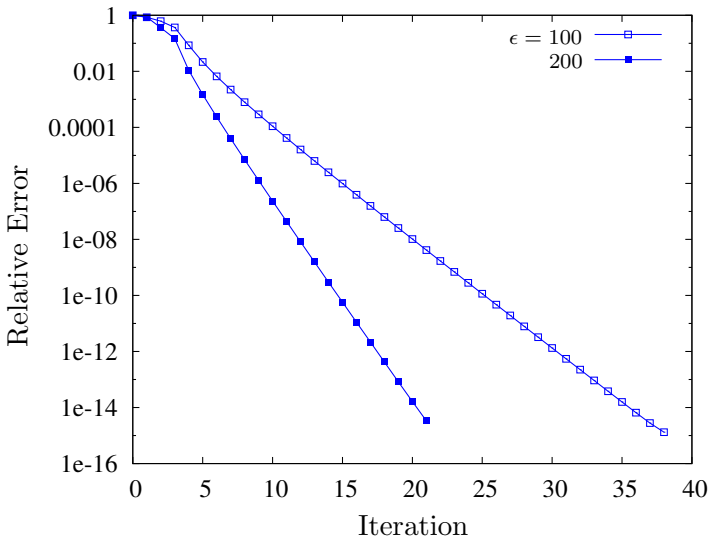

Figure 5.2: The relative error achieved by Alternating Least Squares as a function of the number of iterations. Simulations with $m = n = 1000$, rank $r = 10$ and $\epsilon = |E|/n = 100$ and 200. The plots suggest that the convergence of Alternating Least Squares is faster than predicted by Theorem 5.2.1 and should in fact depend on $|E|$.

for most practical applications, they are essentially the same. Indeed, it is common practice to accept reconstruction up to a small tolerance in the relative error (say $\|M - \widehat{M}\|_F / \|M\|_F \leq 10^{-4}$) while reporting algorithm performances [73, 59].

Second, the guarantees proved for Alternating Least Squares are stronger when r or μ are not bounded. Indeed Theorem 4.2.2 requires a dependence of $O(\mu^2 r^2)$ whereas Theorem 5.2.1 requires only $O(\mu r)$ samples.

Finally Theorem 5.2.1 proves the rate of convergence for Alternating Least Squares. Indeed, the error decreases exponentially with the number of iterations. This behavior is shown empirically in Figure 5.2 for matrices with $m = n = 1000$, rank $r = 10$ and $\epsilon = |E|/n = 100$ and 200. We plot the root mean squared error $\|M - \widehat{M}^t\|_F / n$ as a function of the number of iterations t. Empirically, it is clear that the rate of convergence depends on $|E|$. Indeed, the rate of convergence is directly proportional to ϵ.

The proof of Theorem 5.2.1 makes use of two key lemmas. The first one, Lemma

5.4.7, establishes that one step of the ALTERNATE LEAST SQUARES algorithm halves the distance $d(X^t, U)$ or (symmetrically) $d(Y^t, V)$. The second one, Lemma 5.4.12, establishes that the singular value decomposition in the first phase of ALTERNATE LEAST SQUARES produces a reasonably good estimate as measured through the distance $d(X^0, U)$. We refer to Section 5.4 for the proofs of these lemmas. We will end this Section by proving Theorem 5.2.1 using these lemmas.

Proof. (Theorem 5.2.1) Without loss of generality, we can assume $\|U\|_2, \|V\|_2 \leq 1$.

We first claim that, for each $t \in \{0, 1, \ldots, t_{\max}\}$,

$$d(X^t, U) \leq \frac{1}{2\kappa}, \qquad d(Y^t, V) \leq \frac{1}{2\kappa}\,. \tag{5.12}$$

Indeed, this holds for $t = 0$ by Lemma 5.4.12 (notice that $|E^0| = |E|/t_{\max} \geq C_2\kappa^8 \mu r \log n$ satisfies the hypotheses of that lemma). By Lemma 5.4.7, applied inductively, we further have

$$\begin{aligned} d(X^{t+1}, U) &\leq \frac{1}{4}\, d(X^t, U) + C\,\kappa^2 \left\{ \theta_2 + \sqrt{\frac{n}{\mu r}}\;\theta_\infty + \frac{n\omega}{\sqrt{\epsilon}} \right\} \\ &\overset{(a)}{\leq} \frac{1}{4}\, d(X^t, U) + 2C\,\kappa^2 \left\{ \theta_2 + \frac{n\omega}{\sqrt{\epsilon}} \right\} \\ &\overset{(b)}{\leq} \frac{1}{4}\, d(X^t, U) + \frac{1}{4\kappa}\,. \end{aligned} \tag{5.13}$$

Here (a) follows from assumption (5.7), and (b) from assumption (5.8). The claim (5.12) follows by induction. In particular, the conditions of Lemma 5.4.7 hold for all $t \leq t_{\max}$. By summing Eq. (5.13) we get

$$d(X^t, U) \leq \frac{1}{4^t}\, d(X^0, U) + 3C\,\kappa^2 \left\{ \theta_2 + \frac{n\omega}{\sqrt{\epsilon}} \right\},$$

which proves Eq. (5.10) since by Eq. (5.12), we have $d(X^0, U) \leq 1$. The bound on $d(Y^t, V)$ follows by the same argument.

Finally, by applying Eq. (5.43) in Lemma 5.4.7, we get

$$\|UV^T - X^t(Y^t)^T\|_2 \leq d(X^t, U) + 2\, d(Y^t, V)\,.$$

This yields immediately Eq. (5.11) since $\|UV^T - X^t(Y^t)^T\|_F \leq \sqrt{2r}\,\|UV^T - X^t(Y^t)^T\|_2$, and $\|UV^T\|_F \geq \|UV^T\|_2 \geq 1$. □

5.3 Message Passing

In this section, we introduce a message-passing version of Alternating Least Squares. In order to define the algorithm, let $G = (L, R, E)$ be a bipartite graph with vertices $L = [m]$ corresponding to the rows of N, and $R = [n]$ corresponding to the column of N. Two nodes are connected by an edge $(i, j) \in E$ if and only if the corresponding entry $N_{i,j}$ has been revealed.

Unlike Alternate Least Squares, the message passing algorithm update variables that are associated with the edges of G. More precisely, at iteration t, we keep track of two variables $\theta^{(t)}_{i\to j} \in \mathbb{R}^r$ and $\widehat{\theta}^{(t)}_{j\to i} \in \mathbb{R}^r$ for each $(i, j) \in E$. These will be referred to as 'messages'. Informally, message $\theta^{(t)}_{i\to j}$ corresponds to the estimate of x_i sent from node l_i to node r_j at time t. We will also modify the cost function (5.1), by adding a regularization term

$$\begin{aligned}\mathcal{L}_E(X, Y) &\equiv \left\|\mathcal{P}_E(XY^T - N)\right\|_F^2 + \lambda\|X\|_F^2 + \lambda\|Y\|_F^2 \\ &= \mathcal{R}_E(X, Y) + \lambda\|X\|_F^2 + \lambda\|Y\|_F^2\,.\end{aligned}$$

The message passing version of equations (5.2), (5.3) are

$$\theta^{(t)}_{i\to j} = \left(\lambda + \sum_{k\in\partial i\setminus j} \widehat{\theta}^{(t-1)}_{k\to i} \otimes \widehat{\theta}^{(t-1)}_{k\to i}\right)^{-1} \left(\sum_{k\in\partial i\setminus j} N_{ik}\widehat{\theta}^{(t-1)}_{k\to i}\right) \tag{5.14}$$

$$\widehat{\theta}^{(t)}_{j\to i} = \left(\lambda + \sum_{k\in\partial j\setminus i} \theta^{(t)}_{k\to j} \otimes \theta^{(t)}_{k\to j}\right)^{-1} \left(\sum_{k\in\partial j\setminus i} N_{kj}\theta^{(t)}_{k\to j}\right)\,. \tag{5.15}$$

Notice that, with respect to Eq. (5.2), (5.3), one term is excluded from the sum. At

MESSAGE PASSING FOR MATRIX FACTORIZATION

Input : set E, matrix $\mathcal{P}_E(N)$, number of iterations $t_{\max}$, rank r
Output : estimation $\widehat{M}$

1: Initialize messages $\theta^{(0)}_{i\to j} \in \mathbb{R}^r$ and $\widehat{\theta}^{(0)}_{j\to i} \in \mathbb{R}^r$
2: **for** $t = 1, 2, \ldots, t_{\max}$ **do** :
Update messages using equations (5.14), (5.15)
end for
3: Update the factor estimates using equations (5.16), (5.17)
Output $X^t(Y^t)^T$

Figure 5.3: A summary of the MESSAGE PASSING algorithm

iteration t, the factors $x_i^{(t)}$ and $y_j^{(t)}$ are estimated in terms of messages as follows

$$x_i^{(t)} = \left(\lambda + \sum_{k\in\partial i} \widehat{\theta}^{(t-1)}_{k\to i}\widehat{\theta}^{(t-1)T}_{k\to i}\right)^{-1} \left(\sum_{k\in\partial i} N_{ik}\widehat{\theta}^{(t-1)}_{k\to i}\right), \tag{5.16}$$

$$y_j^{(t)} = \left(\lambda + \sum_{k\in\partial j} \theta^{(t)}_{k\to j}\theta^{(t)T}_{k\to j}\right)^{-1} \left(\sum_{k\in\partial j} N_{kj}\theta^{(t)}_{k\to j}\right). \tag{5.17}$$

We describe the MESSAGE PASSING algorithm in Figure 5.3. At first sight, the time complexity of the message passing algorithm appears to be higher than that of alternate least squares. We show in Section 5.3.1 below that it can be implemented with a time complexity of $O(|E|r^2)$ per iteration, the same as that of ALTERNATE LEAST SQUARES. On the other hand, the memory requirements scale as $O(|E|r)$, to be compared with $O((m\vee n)r+|E|)$ for ALTERNATE LEAST SQUARES. For moderate values of r (typically 10 to 30) used in collaborative filtering applications, this is a manageable overhead.

In general, there is no reason to believe that MESSAGE PASSING will have better convergence properties than ALTERNATE LEAST SQUARES. On the other hand, it is not hard to prove that, if the graph G is a tree, it will converge to a global minimum of $\mathcal{R}_E(X, Y)$, cf. Eq. (5.14). Further, if it converges, the fixed point is a stationary point of $\mathcal{R}_E(X, Y)$ (the proof is completely analogous to the one of [79]). Asymptotic

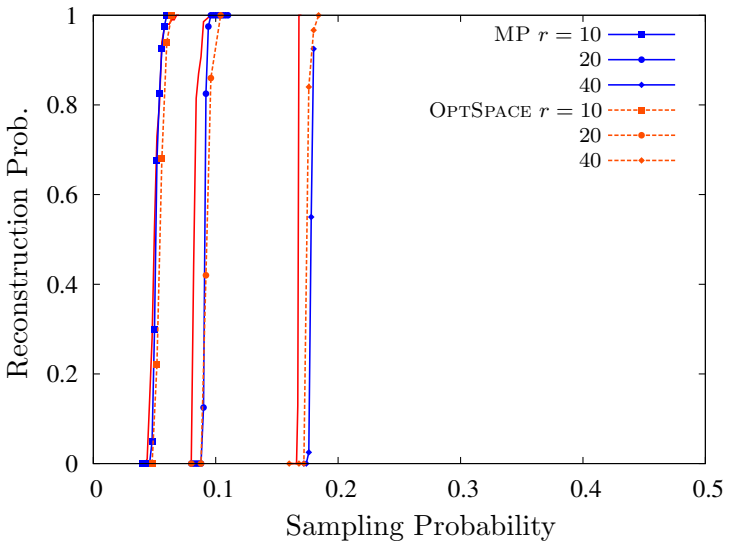

Figure 5.4: Empirical reconstruction probability for Message Passing as a function of the sampling probability $|E|/n^2$. Simulations with $m = n = 500$ and $r = 10, 20, 40$. The fundamental limit derived in [98] and the performance of OptSpace are shown for comparison.

analysis for closely related algorithms is carried out in [86, 54].

In Figure 5.4, we study the performance of Message Passing. As in Figure 4.5, we plot the empirical reconstruction probability, i.e that fraction of instances when the matrix M was reconstructed up to a tolerance $\|M - \widehat{M}\|_F / \|M\|_F \leq 10^{-4}$ as a function of the sampling probability $|E|/mn$. Again, we take $m = n = 500$ and perform the experiments for ranks $r = 10, 20$ and 40. For comparison, we have included the upper bound on reconstruction probability from [98] and the performance of OptSpace. Evidently, Message Passing outperforms OptSpace and is essentially indistinguishable from the fundamental limit.

Analogous to Figure 4.6, we study the performance of Message Passing under noisy data in Figure 5.5. As before, we set $m = n = 600$, rank $r = 2$ and noise variance $\sigma^2 = 1$. We plot the average root mean squared error $\|M - \widehat{M}\|_F / n$ as a function of the sampling probability $p = |E|/n^2$. For comparison, we have also plotted the *Oracle Bound* proved in [20] and the performance of OptSpace. We see that the performance of Message Passing is essentially indistinguishable from the

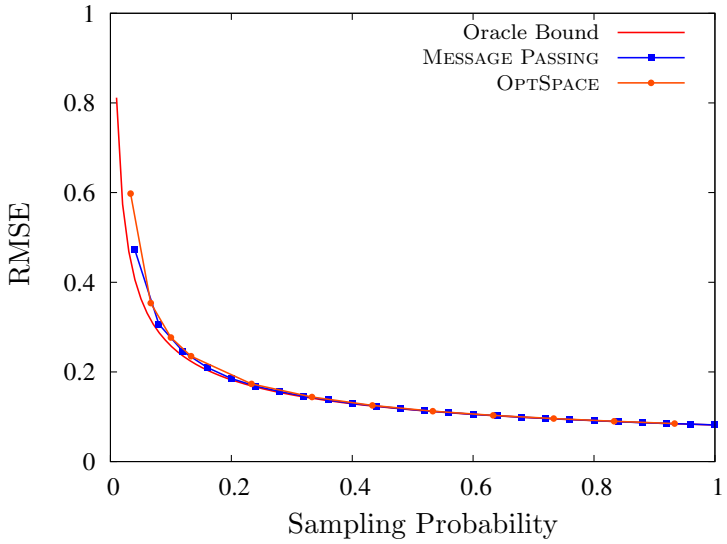

Figure 5.5: Average RMSE $\|M - \widehat{M}\|_F/n$ as a function of the sampling probability $|E|/n^2$. Simulations with $m = n = 600$, $r = 2$ and $\sigma^2 = 1$. The plain red curve is the *Oracle Bound* from [20]. The performance of OptSpace is shown for comparison.

lower bound after a certain threshold on p.

There are several desirable properties to the Message Passing algorithm. In the following, we describe two of these.

1. The Message Passing algorithm is evidently distributable. Each node updates the messages on edges adjacent on it. These updates depend only on the incoming message and are independent of the rest of the nodes. This leads to a highly parallelizable algorithm. Each node can be implemented as a separate computing unit (for eg. as a Pregel node [74]). The communication among nodes is limited by the fact that each node only needs to send messages to its neighbors in the graph G. This property is one of the chief reasons for the large-scale applicability of the algorithm.

2. A line of recent work is concerned with issues of privacy in large scale networks. In [53], the authors propose an algorithm for prediction user ratings in a distributed fashion but with the added constraint of *privacy*. Here, privacy refers to the fact that data is exchanged only between interested parties. It

is clear that both the MESSAGE PASSING algorithm and the ALTERNATING LEAST SQUARES algorithm fall into this category. Using the example from the paper, if the nodes correspond to content publishers and users, then both of our algorithms exchange data only between publishers and the corresponding subscribers.

5.3.1 Implementation Details

In this section, we describe an implementation of the MESSAGE PASSING algorithm described above and show that the complexity of each iteration is $O(|E|r^2)$. By the symmetry in the update equations (5.14), (5.15), and of (5.16), (5.17), we will only consider the equations for updating the messages $\theta_{i\to j}^{(t+1)}$, namely Eq. (5.14).

The naïve implementation of the equations has a complexity of $O(r^2 \sum_{i=1}^n |\partial i|^2)$. Indeed, consider any node i. Computing a single message takes $O(|\partial i|r^2)$ operations. Further, there are $\partial i|$ messages to be computed at each node i. Thus, a single iteration has a complexity of $O(r^2 \sum_{i=1}^n |\partial i|^2)$. However, observe that the messages differ from one another only slightly. Indeed, they differ from the one another in exactly one term in the "denominator" (inverse) part and one term in the "numerator". We will exploit this fact to reduce the per iteration computational complexity.

Before we describe the implementation, let us introduce some notation. For any $i \in [n]$ and $k \in \partial i$, we let

$$\begin{aligned} A_i &\equiv \left(\lambda + \sum_{k\in\partial i} \widehat{\theta}_{k\to i}^{(t)} \widehat{\theta}_{k\to i}^{(t)T}\right), \\ b_{k\to i} &\equiv A_i^{-1} \widehat{\theta}_{k\to i}^{(t)}. \end{aligned}$$

Since the description is similar for each index i, we drop this index in the following discussion for notational clarity. We further drop the time index t, since the update equations are identical at each step. We write $\widehat{\theta}_k \equiv \widehat{\theta}_{k\to i}^{(t)}$, $\theta_k \equiv \theta_{i\to k}^{(t+1)}$ and $x \equiv x_i^{(t+1)}$.

Using this notation, the update equations are

$$\begin{aligned} x &= \sum_{k\in\partial i} N_{ik} b_k \,, \\ \theta_j &= \left(A - \widehat{\theta}_j \widehat{\theta}_j^T\right)^{-1} \left(\sum_{k\in\partial i} N_{ik}\widehat{\theta}_k - N_{ij}\widehat{\theta}_j\right) \\ &= \left(A^{-1} - \frac{A^{-1}\widehat{\theta}_j \widehat{\theta}_j^T A^{-1}}{1 - \widehat{\theta}_j^T A^{-1}\widehat{\theta}_j}\right) \left(\sum_{k\in\partial i} N_{ik}\widehat{\theta}_k - N_{ij}\widehat{\theta}_j\right) \\ &= x - N_{ij} b_j + \frac{\widehat{\theta}_j^T (x - N_{ij} b_j)}{1 - (\widehat{\theta}_j^T b_j)} b_j \,. \end{aligned}$$

Note that in this implementation A can be computed in time $O(r^2|\partial i|)$ and each b_k can be computed in time $O(r^2)$. Further x takes time $O(r^2|\partial i|)$ and each θ_j can be computed in time $O(r)$. Hence the time complexity of the algorithm is $O(r^2 \sum_i |\partial i|) = O(r^2|E|)$.

5.4 Proofs

In this section we prove the technical lemmas that are necessary for our main result, Theorem 5.2.1. In particular in Section 5.4.1 we analyze a single step of the Alternate Least Squares and prove that –under uniform sampling of the matrix entries– a single step reduces the distance $d(X, U)$ or $d(Y, Y)$ by a constant factor. This is formalized in Lemma 5.4.7. Finally, Section 5.4.2 shows that singular value decomposition provides a good initialization for the iteration, as measured through $d(X^0, U)$ and $d(Y^0, U)$, cf. Lemma 5.4.12.

In order to simplify the notation, and following a common practice in this area [24, 82], we write the proof for the case $m = n$. The general proof follows exactly the same argument.

5.4.1 One step analysis

In this section we analyze a single step of the algorithm. Namely, we assume that an estimate Y of the factor V is given and that a uniformly random subset $E^t \subseteq [n] \times [n]$ of the entries is observed, with $|E^t| = n\epsilon$. With an abuse of notation we will drop all the iteration indices and write $E^t = E$. An estimate X of U is constructed according to the algorithm iteration, namely

$$X = \operatorname{argmin} \left\{ \|\mathcal{P}_E(N - \widetilde{X}Y^T)\|_F^2 \ : \ \widetilde{X} \in \mathbb{R}^{n\times r} \right\}. \tag{5.18}$$

The least squares problem is separable, as mentioned above, yielding

$$x_i = \Big(\sum_{j\in\partial i} y_j \otimes y_j \Big)^{-1} \Big(\sum_{j\in\partial i} N_{ij} y_j \Big). \tag{5.19}$$

We begin with a simple linear algebra lemma.

Lemma 5.4.1. *If X is given by Eq. (5.18) and $N = UV^T + W$, then there exists $\alpha \in \mathbb{R}^{r\times r}$ such that*

$$u_i - \alpha x_i = (A+B)^{-1}(\widetilde{A}_i A_i^{-1} B_i - \widetilde{B}_i)u_i - (A+B)^{-1} A A_i^{-1} \sum_{j\in\partial i} W_{ij} y_j \,, \tag{5.20}$$

where we defined the following matrices in $\mathbb{R}^{r\times r}$ (expectations below are taken with respect to the random choice of ∂i):

$$A_i \equiv \sum_{j\in\partial i} y_j \otimes y_j \,, \qquad B_i \equiv \sum_{j\in\partial i} y_j \otimes (v_j - y_j) \,, \tag{5.21}$$

$$A \equiv \mathbb{E}\{A_i\} = \frac{\epsilon}{n} Y^T Y \,, \qquad B \equiv \mathbb{E}\{B_i\} = \frac{\epsilon}{n} Y^T (V - Y) \,, \tag{5.22}$$

$$\widetilde{A}_i \equiv A_i - A \,, \qquad \widetilde{B}_i \equiv B_i - B \,. \tag{5.23}$$

Proof. From equation (5.19), we have that

$$\begin{aligned} x_i &= A_i^{-1}\left(\sum_{j\in\partial i} y_j v_j^T u_i + \sum_{j\in\partial i} W_{ij} y_j\right) \\ &= A_i^{-1}\left(A_i + B_i\right) u_i + A_i^{-1}\sum_{j\in\partial i} W_{ij} y_j, \end{aligned}$$

which suggests the choice $\alpha = (A+B)^{-1}A$. Using this particular α, we get

$$\alpha x_i = (A+B)^{-1} A A_i^{-1}\left(A_i + B_i\right) u_i + (A+B)^{-1} A A_i^{-1}\sum_{j\in\partial i} W_{ij} y_j\,. \quad (5.24)$$

Consider the first term in the sum above. We have

$$\begin{aligned} (A+B)^{-1} A A_i^{-1}\left(A_i + B_i\right) &= (A+B)^{-1}\left(A_i - \widetilde{A}_i\right) A_i^{-1}\left(A_i + B_i\right) \\ &= (A+B)^{-1} A_i A_i^{-1}\left(A + B + \widetilde{A}_i + \widetilde{B}_i\right) \\ &\quad - (A+B)^{-1}\widetilde{A}_i A_i^{-1}\left(A_i + B_i\right) \\ &= I + (A+B)^{-1}\left(\widetilde{A}_i + \widetilde{B}_i\right) - (A+B)^{-1}\widetilde{A}_i A_i^{-1}\left(A_i + B_i\right) \\ &= I + (A+B)^{-1}\widetilde{B}_i - (A+B)^{-1}\widetilde{A}_i A_i^{-1} B_i\,. \end{aligned}$$

Using this identity in equation (5.24), we obtain

$$\alpha x_i = u_i + (A+B)^{-1}\left(\widetilde{B}_i - \widetilde{A}_i A_i^{-1} B_i\right) u_i + (A+B)^{-1} A A_i^{-1}\sum_{j\in\partial i} W_{ij} y_j\,,$$

which proves our claim. □

We now simplify the terms in equation (5.20) and bound $\|u_i - \alpha x_i\|_2$ in the following lemma.

Lemma 5.4.2. *Let X be given by equation (5.18) with $N = UV^T + W$. Then there exists α such that for $\|V - Y\|_2 \le \sigma_{\min}(V)/2$, and $A, A_i, \widetilde{A}_i$, $B, B_i, \widetilde{B}_i$ defined as per*

Eqs. (5.21) to (5.23), we have

$$\begin{aligned}\|u_i - \alpha x_i\|_2 &\leq 2\frac{n}{\epsilon}\sigma_{\min}(V)^{-2}\left(\frac{\|\widetilde{A}_i\|_2\|B_i\|_2}{\sigma_{\min}(A_i)} + \|\widetilde{B}_i\|_2\right)\|u_i\|_2 \\ &\quad +2\frac{n}{\epsilon}\sigma_{\min}(V)^{-2}\|AA_i^{-1}\|_2\left\|\sum_{j\in\partial i} W_{ij}y_j\right\|_2 .\end{aligned}$$

Proof. Using equation (5.20), we have that

$$\|u_i - \alpha x_i\|_2 \leq \frac{1}{\sigma_{\min}(A+B)}\left[\left(\frac{\|\widetilde{A}_i\|_2\|B_i\|_2}{\sigma_{\min}(A_i)} + \|\widetilde{B}_i\|_2\right)\|u_i\|_2 + \|AA_i^{-1}\|_2\left\|\sum_{j\in\partial i} W_{ij}y_j\right\|_2\right] ,$$

where we have repeatedly used the inequalities $\|XY\|_2 \leq \|X\|_2\|Y\|_2$, $\|X+Y\|_2 \leq \|X\|_2 + \|Y\|_2$ and the identity $\|X^{-1}\|_2 = \sigma_{\min}(X)^{-1}$ which are true for any matrices X and Y.

Since $A + B = (\epsilon/n)Y^TV$, we know that $\sigma_{\min}(A+B) \geq (\epsilon/n)\sigma_{\min}(Y)\sigma_{\min}(V)$. Since we assume that $\|V - Y\|_2 \leq \sigma_{\min}(V)/2$, we know that $\sigma_{\min}(Y) \geq \sigma_{\min}(V) - \|V-Y\|_2 \geq \sigma_{\min}(V)/2$ which yields $\sigma_{\min}(A+B) \geq (\epsilon/n)(\sigma_{\min}(V)^2/2)$. Using these in the equation above, we obtain the desired result. □

We now bound the quantities $\|\widetilde{A}_i\|_2, \|\widetilde{B}_i\|_2$ and $\|\sum_{j\in\partial i} W_{ij}y_j\|_2$ in the following lemma.

Lemma 5.4.3. *Let $\widetilde{A}_i$, $\widetilde{B}_i$ defined as per Eqs. (5.21) to (5.23). Further assume that W_{ij} satisfies the assumptions described in Section 5.2.1. Define*

$$\delta_1 \equiv \|Y - V\|_2 , \qquad \delta_2 \equiv \sqrt{\frac{n}{\mu r}}\|Y - V\|_{2,\infty} = \sqrt{\frac{n}{\mu r}}\max_{j\in[n]}\|v_j - y_j\|_2 .$$

Assume that $\|V-Y\|_2 \leq \sigma_{\min}(V)/2$ and that U and V satisfy the incoherence assumption with parameter μ as per Section 5.2.1. Further assume that $\|y_j\|_2^2 \leq C_1\sqrt{\mu r/n}$ and $\epsilon \geq C_2\mu r\log n$ for a large enough constant C_2. Then there exists a constant C,

depending only on C_1, C_2, such that

$$\mathbb{P}\left(\|\widetilde{A}_i\|_2 \leq \frac{\sqrt{C\epsilon\mu r \log n}}{n} \text{ for all } i \in [n]\right) \geq 1 - \frac{1}{n^{10}}, \tag{5.25}$$

$$\mathbb{P}\left(\|\widetilde{B}_i\|_2 \leq \frac{\sqrt{C\epsilon\mu r \log n}}{n}(\delta_1 + \delta_2) \text{ for all } i \in [n]\right) \geq 1 - \frac{1}{n^{10}}, \tag{5.26}$$

$$\mathbb{P}\left(\left\|\sum_{j\in\partial i} W_{ij} y_j\right\|_2 \leq C\omega\sqrt{\frac{\epsilon r \log n}{n}} + C\frac{\epsilon\theta}{n} \text{ for all } i \in [n]\right) \geq 1 - \frac{1}{n^{10}}. \tag{5.27}$$

Proof. The proof of this lemma relies on the Bernstein inequality for random matrices [5, 107]. We begin by writing

$$\widetilde{A}_i = \sum_{j\in\partial i} \Big\{ y_j \otimes y_j - \mathbb{E}[y_j \otimes y_j] \Big\}$$

where the expectation is taken with respect to ∂i. For notational convenience, let $\psi_j = y_j \otimes y_j - \mathbb{E}[y_j \otimes y_j]$. Clearly $\mathbb{E}[\psi_j] = 0$. Note that

$$\begin{aligned} \|\psi_j\|_2 &\leq \|y_j\|^2 + \frac{1}{n}\|Y\|_2^2 \\ &\leq \frac{\mu r}{n} + \frac{2}{n} \\ &\leq C'\frac{\mu r}{n}. \end{aligned}$$

and

$$\begin{aligned} \|\mathbb{E}[\psi_j^2]\|_2 &= \|\mathbb{E}[\|y_j\|_2^2 y_j \otimes y_j] - \mathbb{E}[y_j \otimes y_j]^2\|_2 \\ &\leq \frac{\mu r}{n}\frac{\|Y\|_2^2}{n} + \frac{\|Y\|_2^4}{n^2} \\ &\leq C'\frac{\mu r}{n^2}. \end{aligned}$$

for some suitable constant C'. In the last inequality, we have used the fact that

$\|Y\|_2 \leq \|V\|_2 + \|V - Y\|_2 \leq 2$. Using Theorem 6.1 of [107], we get that

$$\mathbb{P}\left(\|\widetilde{A}_i\|_2 \geq t\right) \leq r \exp\left(-\frac{t^2}{C'\epsilon\mu r/n^2 + C'\mu r t/n}\right)$$

We get the required result by using $t = \frac{\sqrt{C\epsilon\mu r \log n}}{n}$ for some suitable constant C and using union bound over the different $\widetilde{A}_i$. This completes the proof of equation (5.25).

The proof of equation (5.26) is similar to the proof above. Let

$$D_i = \begin{pmatrix} 0 & \widetilde{B}_i \\ \widetilde{B}_i^T & 0 \end{pmatrix}$$

Clearly, $\|\widetilde{B}_i\|_2 = \|D_i\|_2$. Define

$$\xi_j = \begin{pmatrix} 0 & y_j \otimes (v_j - y_j) - \mathbb{E}[y_j \otimes (v_j - y_j)] \\ (v_j - y_j) \otimes y_j - \mathbb{E}[(v_j - y_j) \otimes y_j] & 0 \end{pmatrix}$$

Hence $D_i = \sum_{j \in \partial i} \xi_j$. Again $\mathbb{E}[\xi_j] = 0$. Further,

$$\begin{aligned} \|\xi_j\|_2 &\leq \|y_j\|_2 \|v_j - y_j\|_2 + \|Y^T(V - Y)\|_2/n \\ &\leq \frac{C'\mu r}{n}\left(\delta_1 + \delta_2\right) , \end{aligned}$$

where we have used the fact that $\|y_j\|_2 \leq C\sqrt{\mu r/n}$ and $\|Y\|_2 \leq 2$. Further,

$$\begin{aligned} \|\mathbb{E}[\xi_j^2]\|_2 &\leq \|\mathbb{E}[\|y_j\|_2^2 (v_j - y_j) \otimes (v_j - y_j)]\|_2 + \|\mathbb{E}[\|v_j - y_j\|_2^2 y_j \otimes y_j]\|_2 \\ &\quad + \|\mathbb{E}[y_j \otimes (v_j - y_j)]\|_2^2 + \|\mathbb{E}[(v_j - y_j) \otimes y_j]\|_2^2 \\ &\leq \|y_j\|_2^2 \|\mathbb{E}[(v_j - y_j) \otimes (v_j - y_j)]\|_2 + \|v_j - y_j\|_2^2 \|\mathbb{E}[y_j \otimes y_j]\|_2 \\ &\quad + \|\mathbb{E}[y_j \otimes (v_j - y_j)]\|_2^2 + \|\mathbb{E}[(v_j - y_j) \otimes y_j]\|_2^2 \end{aligned}$$

We know that,

$$\begin{aligned}
\mathbb{E}[(v_j - y_j) \otimes (v_j - y_j)] &= \frac{1}{n}(V-Y)^T(V-Y)\,, \\
\mathbb{E}[y_j \otimes y_j] &= \frac{1}{n}Y^TY\,, \\
\mathbb{E}[(v_j - y_j) \otimes y_j] &= \frac{1}{n}(V-Y)^TY\,,
\end{aligned}$$

which yields

$$\begin{aligned}
\|\mathbb{E}[\xi_j^2]\|_2 &\leq \frac{1}{n}\|y_j\|_2^2\|V-Y\|_2^2 + \frac{1}{n}\|v_j - y_j\|_2^2\|Y\|_2^2 + \frac{1}{n^2}\|Y\|_2\|V-Y\|_2 \\
&\leq \frac{\mu r\delta_1^2}{n^2} + \frac{\mu r\delta_2^2}{n^2} + \frac{C'\delta_1\delta_2}{n^2} \\
&\leq \frac{C'\mu r}{n^2}\left(\delta_1+\delta_2\right)^2\,.
\end{aligned}$$

Using Theorem 6.1 of [107], and using $t = \frac{\sqrt{C\epsilon\mu r\log n}}{n}\left(\delta_1+\delta_2\right)$, we get the desired result after union bound.

We now turn to bounding $\sum_{j\in\partial i} W_{ij}y_j$. For notational convenience, define $\widetilde{W}_{ij} = W_{ij} - \overline{W}_{ij}$. $\widetilde{W}_{ij}$ are zero mean sub-Gaussian random variables with parameter ω. We start by writing

$$\sum_{j\in\partial i} W_{ij}y_j = \sum_{j\in\partial i}\overline{W}_{ij}y_j + \sum_{j\in\partial i}\widetilde{W}_{ij}y_j.$$

To bound the first part, note that

$$\begin{aligned}
\Big\|\sum_{j\in\partial i}\overline{W}_{ij}y_j\Big\|_2^2 &\leq \|\sum_{j\in\partial i} y_jy_j^T\|_2\sum_{j\in\partial i}\overline{W}_{ij}^2 \\
&= \|A_i\|_2\sum_{j\in\partial i}\overline{W}_{ij}^2 \\
&\leq \|A_i\|_2\left(\frac{\epsilon}{n}\sum_j\overline{W}_{ij}^2 + \sum_{j\in\partial i}\left\{\overline{W}_{ij}^2 - \mathbb{E}_{\partial i}[\overline{W}_{ij}^2]\right\}\right)\,.
\end{aligned}$$

From equation (5.25) we know that $\|A_i\|_2 \leq \|A\|_2 + \|\widetilde{A}_i\|_2 \leq C\epsilon/n$ for $\epsilon \geq C\mu r \log n$. Using the same techniques as simple Bernstein we can prove that

$$\sum_{j\in\partial i}\left\{\overline{W}_{ij}^2 - \mathbb{E}_{\partial i}[\overline{W}_{ij}^2]\right\} \;\leq\; \frac{\sqrt{C\epsilon\mu r\log n}}{n}\theta^2$$

which proves that $\|\sum_{j\in\partial i}\overline{W}_{ij}y_j\|_2 \leq C(\epsilon/n)\theta$ with probability larger than $1-1/n^{10}$ for $\epsilon \geq C\mu r\log n$.

We finally show that $\|\sum_{j\in\partial i}\widetilde{W}_{ij}y_j\|_2 \leq C\omega\sqrt{\epsilon r\log n/n}$ which completes the proof of equation (5.27). Let

$$\zeta_j = \begin{pmatrix} 0 & \widetilde{W}_{ij}y_j \\ \widetilde{W}_{ij}y_j^T & 0 \end{pmatrix}$$

and let $G_i = \sum_{j\in\partial i}\zeta_j$. Clearly $\|\sum_{j\in\partial i}\widetilde{W}_{ij}y_j\|_2 = \|G_i\|_2$. We will use Theorem 6.2 of [107]. We have $\mathbb{E}[\zeta_j] = 0$. Further,

$$\begin{aligned}\mathbb{E}[\zeta_j^p] &\preceq Cp!\,(\omega\|y_j\|_2)^{p-2}\,\mathbb{E}[\omega^2\zeta_j^2] \\ &\preceq Cp!\left(\omega\sqrt{\frac{\mu r}{n}}\right)^{p-2}\mathbb{E}[\omega^2\zeta_j^2]\end{aligned}$$

We also know that

$$\|\mathbb{E}[\omega^2\zeta_j^2]\|_2 \leq C\omega^2 r/n.$$

Hence, using the results of [107], we obtain

$$\mathbb{P}\left(\|\sum_{j\in\partial i}\widetilde{W}_{ij}y_j\|_2 \geq t\right) \leq n\exp\left(-\frac{t^2}{C\epsilon\omega^2 r/n + C\omega\sqrt{\mu r/n}t}\right).$$

Setting $t = C\omega\sqrt{\frac{\epsilon r\log n}{n}}$ and using union bound, we obtain the desired thesis. □

We can now prove the desired bound on $\|u_i - \alpha x_i\|_2$ in Lemma 5.4.4.

Lemma 5.4.4. *Assume that X is given by equation (5.18) and let α be as defined in Lemma 5.4.1. Let U and V satisfy the incoherence assumption with parameter μ. Let W_{ij} satisfy the conditions described in Section 5.2.1. Further assume that $\|V - Y\|_2 \leq \sigma_{\min}(V)/2$. There exists a universal constant C such that if $\epsilon \geq C\sigma_{\min}(V)^{-8}\mu r \log n$, then with probability larger than $1 - 1/n^5$, for every i,*

$$\begin{aligned}\|u_i - \alpha x_i\|_2 &\leq \frac{1}{4}\|V - Y\|_2\sqrt{\frac{\mu_0 r}{n}} + \frac{1}{4}\max_j \|v_j - y_j\|_2 + C'\sigma_{\min}(V)^{-2}\theta \\ &\quad + C'\sigma_{\min}(V)^{-2}\left(\frac{n\omega}{\sqrt{\epsilon}}\right)\sqrt{\frac{r\log n}{n}}\end{aligned}$$

Proof. (Lemma 5.4.4) For notational convenience, we define $\delta_1 \equiv \|V - Y\|_2$ and $\delta_2 \equiv \max_j \|v_j - y_j\|_2\sqrt{\frac{n}{\mu r}}$. From Lemma 5.4.2 and using the incoherence assumption $\|u_i\|_2 \leq \sqrt{\mu r/n}$, we know that

$$\|u_i - \alpha x_i\|_2 \leq 2\frac{n}{\epsilon}\sigma_{\min}(V)^{-2}\left(\frac{\|\widetilde{A}_i\|_2\|B_i\|_2}{\sigma_{\min}(A_i)} + \|\widetilde{B}_i\|_2\right)\sqrt{\frac{\mu_0 r}{n}} \tag{5.28}$$

$$+2\frac{n}{\epsilon}\sigma_{\min}(V)^{-2}\|AA_i^{-1}\|_2\Big\|\sum_{j\in\partial i} W_{ij}y_j\Big\|_2 \tag{5.29}$$

$$\tag{5.30}$$

We know that

$$\begin{aligned}\sigma_{\min}(A) &\geq \frac{\epsilon}{n}\sigma_{\min}(Y)^2 \\ &\geq \frac{\epsilon}{n}\left(\sigma_{\min}(V) - \|V - Y\|_2\right)^2 \\ &\geq \frac{\epsilon}{n}\frac{\sigma_{\min}(V)^2}{4}\end{aligned} \tag{5.31}$$

and

$$\|A\|_2 \leq \frac{\epsilon}{n}\|Y\|_2^2 \leq \frac{\epsilon}{n}(\|V\|_2 + \|V - Y\|_2)^2 \leq 4\frac{\epsilon}{n} \tag{5.32}$$

and

$$\|B\|_2 \leq \frac{\epsilon}{n}\|Y\|_2\|V_Y\|_2 \leq \frac{\epsilon}{n}\delta_1 \tag{5.33}$$

Further, using Lemma 5.4.3, we have that for $\epsilon \geq \sigma_{\min}(V)^4\mu r \log n$

$$\sigma_{\min}(A_i) \geq \sigma_{\min}(A) - \|\widetilde{A}_i\|_2 \overset{(a)}{\geq} \frac{\epsilon}{n}\frac{\sigma_{\min}(V)^{-2}}{8}, \tag{5.34}$$

$$\|AA_i^{-1}\|_2 \leq \|I - AA_i^{-1}\widetilde{A}_i A^{-1}\|_2 \overset{(b)}{\leq} 1 + \frac{\|AA_i^{-1}\|_2}{2}. \tag{5.35}$$

Above, (a) follows from equation (5.31) followed by using equation (5.25) of Lemma 5.4.3 for $\epsilon \geq \sigma_{\min}(V)^8\mu r \log n$. For (b), note that $AA_i^{-1} = A(A + \widetilde{A}_i)^{-1} = I - AA_i^{-1}\widetilde{A}_i A^{-1}$. Further, we know that

$$\begin{aligned}
\|\widetilde{A}_i A^{-1}\|_2 &\leq \|\widetilde{A}_i\|_2\|A^{-1}\|_2 \\
&\leq \frac{\sqrt{C\epsilon\mu r \log n}}{n}\frac{n\sigma_{\min}(V)^{-2}}{\epsilon} \\
&\leq \frac{1}{2}
\end{aligned}$$

for $\epsilon \geq C\sigma_{\min}(V)^{-8}\mu r \log n$ where we have use equation (5.31) in the first inequality above. From (c) above, we can conclude that $\|AA_i^{-1}\|_2 \leq 2$. Further, we have

$$\begin{aligned}
\|B_i\|_2 &\leq \|B\|_2 + \|\widetilde{B}_i\|_2 \\
&\leq \frac{\epsilon}{n}\delta_1 + \frac{\sqrt{C\epsilon\mu r \log n}}{n}(\delta_1 + \delta_2) \\
&\leq 2\frac{\epsilon}{n}(\delta_1 + \delta_2)
\end{aligned} \tag{5.36}$$

for $\epsilon \geq C\mu r \log n$.

Substituting these inequalities into equation (5.29) above, we have

$$\begin{aligned}
\|u_i - \alpha x_i\|_2 &\overset{(c)}{\leq} 2\frac{n}{\epsilon}\sigma_{\min}(V)^{-2}\left(\sigma_{\min}(V)^{-2}\|\widetilde{A}_i\|_2\,(\delta_1+\delta_2) + \|\widetilde{B}_i\|_2\right)\sqrt{\frac{\mu_0 r}{n}} \\
&\quad +2\frac{n}{\epsilon}\sigma_{\min}(V)^{-2}\|AA_i^{-1}\|_2\Big\|\sum_{j\in\partial i} W_{ij}y_j\Big\|_2 \\
&\overset{(d)}{\leq} 2\frac{n}{\epsilon}\sigma_{\min}(V)^{-2}\left(\sigma_{\min}(V)^{-2}\frac{\sqrt{C\epsilon\mu r\log n}}{n}\,(2\delta_1+\delta_2) + \frac{\sqrt{C\epsilon\mu r\log n}}{n}\delta_2\right)\sqrt{\frac{\mu_0 r}{n}} \\
&\quad +2\frac{n}{\epsilon}\sigma_{\min}(V)^{-2}\|AA_i^{-1}\|_2\left(C\omega\sqrt{\frac{\epsilon r\log n}{n}} + C\frac{\epsilon}{n}\theta\right) \\
&\leq C'\sigma_{\min}(V)^{-4}\sqrt{\frac{\mu r\log n}{\epsilon}}\,(\delta_1+\delta_2)\sqrt{\frac{\mu_0 r}{n}} + C'\sigma_{\min}(V)^{-2}\theta \\
&\quad +C'\sigma_{\min}(V)^{-2}\left(\frac{n\omega}{\sqrt{\epsilon}}\right)\sqrt{\frac{r\log n}{n}} \\
&\overset{(e)}{\leq} (\delta_1/4+\delta_2/4)\sqrt{\frac{\mu_0 r}{n}} + C'\sigma_{\min}(V)^{-2}\theta + C'\sigma_{\min}(V)^{-2}\left(\frac{n\omega}{\sqrt{\epsilon}}\right)\sqrt{\frac{r\log n}{n}}
\end{aligned}$$

where (c) above follows from equations (5.34) and (5.36) and (d) follows from Lemma 5.4.3. Finally, we obtain (e) for $\epsilon \geq C\sigma_{\min}(V)^{-8}\mu r\log n$. □

In the following lemma, we bound $\|U - X\alpha^T\|_2$.

Lemma 5.4.5. *Assume that X is given by equation (5.18) and let α be as defined in Lemma 5.4.1. Define*

$$\begin{aligned}
Q_i &\equiv (A+B)^{-1}(\widetilde{A}_i A_i^{-1}B_i - \widetilde{B}_i) \\
R_i &\equiv (A+B)^{-1}AA_i^{-1} \\
z_i &\equiv \sum_{j\in\partial i} W_{ij}y_j
\end{aligned}$$

Then,

$$\|U - X\alpha^T\|_2 \leq \Big\|\sum_{i=1}^n Q_i u_i u_i^T Q_i^T\Big\|_2^{1/2} + \Big\|\sum_{i=1}^n R_i z_i z_i^T R_i^T\Big\|_2^{1/2}$$

Proof. (Lemma 5.4.5) We begin by writing $u_i - \alpha x_i$ using Lemma 5.4.1 as

$$\begin{aligned} u_i - \alpha x_i &= Q_i u_i + R_i \sum_{j \in \partial i} W_{ij} y_j \\ &= Q_i u_i + R_i z_i. \end{aligned}$$

Let $J \in \mathbb{R}^{n \times r}$ be the matrix whose i^{th} row $J_i^T = u_i^T Q_i^T$. Similarly, let $K \in \mathbb{R}^{n \times r}$ be matrix whose i^{th} row $K_i^T = z_i^T R_i^T$. Then,

Clearly,

$$\begin{aligned} \|U - X\alpha^T\|_2 &\leq \|J\|_2 + \|K\|_2 \\ &= \Big\| \sum_{i=1}^n J_i J_i^T \Big\|_2^{1/2} + \Big\| \sum_{i=1}^n K_i K_i^T \Big\|_2^{1/2} \\ &= \Big\| \sum_{i=1}^n Q_i u_i u_i^T Q_i^T \Big\|_2^{1/2} + \Big\| \sum_{i=1}^n R_i z_i z_i^T R_i^T \Big\|_2^{1/2} \end{aligned}$$

□

Next we state a technical lemma that bounds the different quantities in the result of Lemma 5.4.5.

Lemma 5.4.6. *Assume that X is given by equation (5.18) and let α be as defined in Lemma 5.4.1. Further assume that W_{ij} satisfy the assumptions described in Section 5.2.1. Let U, V satisfy the incoherence assumption with parameter μ and let $\|V - Y\|_2 \leq 1$. If $\epsilon \geq C\sigma_{\min}(V)^{-8}\mu r \log n$ for a large enough constant C, then,*

$$\begin{aligned} \mathbb{P}\left(\Big\| \sum_i^n Q_i u_i u_i^T Q_i^T \Big\|_2 \geq \frac{(\delta_1 + \delta_2)^2}{16} \right) &\leq \frac{1}{n^5}, \\ \mathbb{P}\left(\Big\| \sum_i^n R_i z_i z_i^T R_i^T \Big\|_2 \geq C' \sigma_{\min}(V)^{-4} \left(\|\overline{W}\|_2^2 + \frac{n^2\omega^2}{\epsilon} \right) \right) &\leq \frac{1}{n^5}, \end{aligned}$$

Proof. (Lemma 5.4.6) We begin by writing

$$\mathbb{I}_i = \mathbb{I}(\|Q_i\|_2 \leq (\delta_1 + \delta_2)/8)$$

Note that,

$$\sum_{i=1}^{n} Q_i u_i u_i^T Q_i^T = \sum_{i=1}^{n} \mathbb{I}_i Q_i u_i u_i^T Q_i^T + \sum_{i=1}^{n} (1-\mathbb{I}_i) Q_i u_i u_i^T Q_i^T \,.$$

For any $t > 0$, we have that

$$\begin{aligned}
\mathbb{P}\left(\Big\| \sum_{i=1}^{n} Q_i u_i u_i^T Q_i^T \Big\|_2 \geq t \right) &\leq \mathbb{P}\left(\Big\| \sum_{i=1}^{n} \mathbb{I}_i Q_i u_i u_i^T Q_i^T \Big\|_2 \geq \frac{t}{2} \right) \\
&\quad + \mathbb{P}\left(\Big\| \sum_{i=1}^{n} (1-\mathbb{I}_i) Q_i u_i u_i^T Q_i^T \Big\|_2 \geq \frac{t}{2} \right) \\
&\leq \mathbb{P}\left(\Big\| \sum_{i=1}^{n} \mathbb{I}_i Q_i u_i u_i^T Q_i^T \Big\|_2 \geq \frac{t}{2} \right) + \sum_{i=1}^{n} \mathbb{P}\left(\|Q_i\|_2 \geq \frac{\delta_1 + \delta_2}{8} \right) \\
&\leq \mathbb{P}\left(\Big\| \sum_{i=1}^{n} \mathbb{I}_i Q_i u_i u_i^T Q_i^T \Big\|_2 \geq \frac{t}{2} \right) + \sum_{i=1}^{n} \frac{1}{n^{10}}
\end{aligned}$$

where we have used Lemma 5.4.4 in the last inequality above. To bound the first term, we begin by writing

$$\begin{aligned}
\Big\| \sum_i \mathbb{I}_i Q_i u_i u_i^T Q_i^T \Big\|_2 &\leq \Big\| \sum_i \mathbb{I}_i Q_i u_i u_i^T Q_i^T - \mathbb{E}[\mathbb{I}_i Q_i u_i u_i^T Q_i^T] \Big\|_2 \qquad (5.37) \\
&\quad + \Big\| \sum_i \mathbb{E}[\mathbb{I}_i Q_i u_i u_i^T Q_i^T] \Big\|_2 \qquad (5.38)
\end{aligned}$$

where the expectation is wrt ∂i. To compute $\| \sum_i \mathbb{E}[\mathbb{I}_i Q_i u_i u_i^T Q_i^T] \|_2$, note that Q_i are all identically distributed and U is non random and hence

$$\begin{aligned}
\Big\| \sum_i \mathbb{E}[\mathbb{I}_i Q_i u_i u_i^T Q_i^T] \Big\|_2 &= \Big\| \mathbb{E}[\mathbb{I}_1 Q_1 \sum_i u_i u_i^T Q_1^T] \Big\|_2 \\
&= \|\mathbb{E}[\mathbb{I}_1 Q_1 U^T U Q_1^T]\|_2.
\end{aligned}$$

which yields,

$$\Big\| \sum_i \mathbb{E}[\mathbb{I}_i Q_i u_i u_i^T Q_i^T] \Big\|_2 \leq (\delta_1 + \delta_2)^2 / 64.$$

To bound the first part in equation (5.38) above, we use the Bernstein inequality. Let $\psi_i = \mathbb{I}_i Q_i u_i u_i^T Q_i^T - \mathbb{E}[\mathbb{I}_i Q_i u_i u_i^T Q_i^T]$. Clearly, $\mathbb{E}[\psi_i] = 0$. Further,

$$\begin{aligned} \|\psi_i\|_2 &\leq \|\mathbb{I}_i Q_i u_i u_i^T Q_i\|_2 + \|\mathbb{E}[\mathbb{I}_i Q_i u_i u_i^T Q_i^T]\|_2 \\ &\leq 2\frac{\mu r}{n}\left(\frac{\delta_1+\delta_2}{8}\right)^2 \end{aligned}$$

and

$$\begin{aligned} \Big\|\sum_i \mathbb{E}[\psi_i^2]\Big\|_2 &\leq \Big\|\sum_i \mathbb{E}[\mathbb{I}_i Q_i u_i u_i^T Q_i^T Q_i u_i u_i^T Q_i^T]\Big\|_2 + \Big\|\sum_i \mathbb{E}[\mathbb{I}_i Q_i u_i u_i^T Q_i^T]^2\Big\|_2 \\ &\leq C'\frac{\mu r}{n}\left(\frac{\delta_1+\delta_2}{8}\right)^2 \Big\|\sum_i \mathbb{E}[\mathbb{I}_i Q_i u_i u_i^T Q_i^T]\Big\|_2 \\ &\leq C'\frac{\mu r}{n}\left(\frac{\delta_1+\delta_2}{8}\right)^4 \end{aligned}$$

Hence, using Theorem 6.1 of [107], we have that

$$P\left(\Big\|\sum_i \mathbb{I}_i Q_i u_i u_i^T Q_i^T - \mathbb{E}[\mathbb{I}_i Q_i u_i u_i^T Q_i^T]\Big\|_2 \geq t\right) \leq r \quad \exp\left(-\frac{t^2}{C'\mu r/n\left(\frac{\delta_1+\delta_2}{8}\right)^4 + C'\mu r/n\left(\frac{\delta_1+\delta_2}{8}\right)^2 t}\right)$$

Setting $t = (\delta_1+\delta_2)^2/64$, we obtain the desired result for $\epsilon \geq C\mu r \log n$ with large enough constant C.

To prove equation (5.37), we begin by defining

$$S_i \equiv \sum_{j\in\partial i} y_j \otimes \overline{W}_j$$

where $\overline{W}_j \equiv$ is the j^{th} column of $\overline{W}$. Using the methods used in Lemma 5.4.3, we can show that

$$\mathbb{P}\left(\|S_i\|_2 \geq C'\frac{\epsilon}{n}\|\overline{W}\|_2 \text{ for some } i\right) \leq \frac{1}{n^{10}}\,. \tag{5.39}$$

Define,

$$\mathbb{I}_i = \mathbb{I}\left(\|S_i\|_2 \leq C'\frac{\epsilon}{n}\|\overline{W}\|_2\right) \cdot \mathbb{I}\left(\|R_i\|_2 \leq \frac{n}{\epsilon}\sigma_{\min}(V)^{-2}\right) \cdot \mathbb{I}\left(\|z_i\|_2 \leq C'\frac{\epsilon\theta}{n} + C'\omega\sqrt{\frac{\epsilon r \log n}{n}}\right)$$

From Lemma 5.4.3 and equation (5.39), we know that $\mathbb{I}_i = 1$ for all i with probability greater than $1 - 1/n^9$.

For any $t > 0$,

$$\begin{aligned}
\mathbb{P}\left(\Big\|\sum_i R_i z_i z_i^T R_i^T\Big\|_2 \geq t\right) &\leq \mathbb{P}\left(\Big\|\sum_i \mathbb{I}_i R_i z_i z_i^T R_i^T\Big\|_2 \geq \frac{t}{2}\right) \\
&\quad + \mathbb{P}\left(\Big\|(1-\mathbb{I}_i)\sum_i R_i z_i z_i^T R_i^T\Big\|_2 \geq \frac{t}{2}\right) \\
&\leq \mathbb{P}\left(\Big\|\sum_i \mathbb{I}_i R_i z_i z_i^T R_i^T\Big\|_2 \geq \frac{t}{2}\right) + \sum_{i=1}^n \mathbb{P}\left(\mathbb{I}_i = 0\right) \\
&\leq \mathbb{P}\left(\Big\|\sum_i \mathbb{I}_i R_i z_i z_i^T R_i^T\Big\|_2 \geq \frac{t}{2}\right) + \sum_{i=1}^n \frac{1}{n^9}\,.
\end{aligned}$$

In the following, we bound the first term in the above equation.

$$\begin{aligned}
\Big\|\sum_i \mathbb{I}_i R_i z_i z_i^T R_i^T\Big\|_2 &\leq \Big\|\sum_i \mathbb{I}_i R_i z_i z_i^T R_i^T - \mathbb{E}[\mathbb{I}_i R_i z_i z_i^T R_i^T]\Big\|_2 \\
&\quad + \Big\|\sum_i \mathbb{E}[\mathbb{I}_i R_i z_i z_i^T R_i^T]\Big\|_2 \qquad (5.40)
\end{aligned}$$

To compute $\| \sum_i \mathbb{E}[R_i z_i z_i^T R_i^T] \|_2$, note that

$$\begin{aligned}
\sum_i \mathbb{E}[\mathbb{I}_i R_i z_i z_i^T R_i^T] &= \sum_i \mathbb{E}[\mathbb{I}_i R_i \left(\sum_{j,j' \in \partial i} W_{ij} W_{ij'} y_j y_j'^T \right) R_i^T] \\
&= \sum_i \mathbb{E}[\mathbb{E}[\mathbb{I}_i R_i \left(\sum_{j,j' \in \partial i} W_{ij} W_{ij'} y_j y_j'^T \right) R_i^T | \partial i]] \\
&= \sum_i \mathbb{E}[\mathbb{I}_i R_i \left(S_i e_i e_i^T S_i^T + \omega^2 \sum_{j \in \partial i} y_j y_j'^T \right) R_i^T] \\
&= \sum_i \mathbb{E}[\mathbb{I}_1 R_1 \left(S_1 e_i e_i^T S_1^T + \omega^2 \sum_{j \in \partial 1} y_j y_j'^T \right) R_1^T] \\
&= \mathbb{E}[\mathbb{I}_1 R_1 S_1 S_1^T R_1^T] + n\omega^2 \mathbb{E}[\mathbb{I}_1 R_1 \left(\sum_{j \in \partial 1} y_j y_j'^T \right) R_1^T]
\end{aligned}$$

which implies $\| \sum_i \mathbb{E}[\mathbb{I}_i R_i z_i z_i^T R_i^T] \|_2 \leq C \sigma_{\min}(V)^{-4} \left(\|\overline{W}\|_2^2 + \frac{n^2 \omega^2}{\epsilon} \right)$. To bound the first term in equation (5.40), we use the Bernstein inequality. Let $\xi_i = \mathbb{I}_i R_i z_i z_i^T R_i^T - \mathbb{E}[\mathbb{I}_i R_i z_i z_i^T R_i^T]$. Clearly $\mathbb{E}[\xi_i] = 0$.

$$\begin{aligned}
\|\xi_i\|_2 &\leq C' \left(\frac{n \sigma_{\min}(V)^{-2}}{\epsilon} \right)^2 \|z_i\|_2^2 \\
&\leq C' \left(\frac{n \sigma_{\min}(V)^{-2}}{\epsilon} \right)^2 \left(\frac{\epsilon \theta}{n} + \omega \sqrt{\frac{\epsilon r \log n}{n}} \right)^2 \\
&\leq C' \sigma_{\min}(V)^{-4} \left(\frac{\mu r}{n} \|\overline{W}\|_2^2 + \frac{r \log n}{n} \frac{n^2 \omega^2}{\epsilon} \right)
\end{aligned}$$

where in the second inequality above, we have used the results of Lemma 5.4.3.

Further,

$$\begin{aligned}\left\|\sum_i \mathbb{E}[\xi_i^2]\right\|_2 &\leq \left\|\sum_i \|\xi_i\|_2 \mathbb{E}[\mathbb{I}_i R_i \bar{z}_i \bar{z}_i^T R_i^T]\right\|_2 \\ &\leq C'\sigma_{\min}(V)^{-4}\frac{\mu r}{n}\|\overline{W}\|_2^2\left\|\sum_i \mathbb{E}[\mathbb{I}_i R_i \bar{z}_i \bar{z}_i^T R_i^T]\right\|_2 \\ &\leq C'\sigma_{\min}(V)^{-8}\left(\frac{\mu r}{n}\|\overline{W}\|_2^2 + \frac{r\log n}{n}\frac{n^2\omega^2}{\epsilon}\right)\left(\|\overline{W}\|_2^2 + \frac{n^2\omega^2}{\epsilon}\right).\end{aligned}$$

Hence, using Theorem 6.1 of [107], we have that

$$\mathbb{P}\left(\left\|\sum_i \mathbb{I}_i R_i \bar{z}_i \bar{z}_i^T R_i^T - \mathbb{E}[\mathbb{I}_i R_i \bar{z}_i \bar{z}_i^T R_i^T]\right\|_2 \geq t\right) \leq r\exp\left(-\frac{t^2}{\|\sum_i \mathbb{E}[\xi_i^2]\|_2 + \max_i \|\xi_i\|_2 t}\right)$$

Setting $t = C'\sigma_{\min}(V)^{-4}\left(\|\overline{W}\|_2^2 + \frac{n^2\omega^2}{\epsilon}\right)$, we obtain the desired result.

□

Substituting the result of Lemma 5.4.6 into Lemma 5.4.5, we can now prove the following where we also recall the results of Lemma 5.4.4 for convenience.

Lemma 5.4.7. *There exist universal constants C_1, C_2 such that the following happens.*

Assume that the conditions in Section 5.2.1 hold and let Y be such that $\|V - Y\|_{2,2} \leq \sigma_{\min}(V)/2$ and $\|V - Y\|_{2,\infty} \leq C_1\sqrt{\mu r/n}$. Let $E \subseteq [m] \times [n]$ be random with $\mathbb{P}\{(i,j) \in E\} = \epsilon/\sqrt{mn}$ independently across entries, and independent from N, V, Y. Define X as follows

$$X = \operatorname{argmin}\left\{\|\mathcal{P}_E(N - \widetilde{X}Y^T)\|_F^2 \ : \ \widetilde{X} \in \mathbb{R}^{n\times r}\right\}. \tag{5.41}$$

If $\epsilon \geq C_2\sigma_{\min}(V)^{-8}\mu r \log n$, then with probability larger than $1 - 1/n^5$ we have

$$d(X,U) \leq \frac{1}{2}\,d(Y,V) + \frac{C_2}{\sigma_{\min}(V)^2}\left\{\theta_2 + \sqrt{\frac{n}{\mu r}}\,\theta_\infty + \frac{n\omega}{\sqrt{\epsilon}}\right\}. \tag{5.42}$$

Further, under the same assumptions,

$$\|UV^T - XY^T\|_2 \leq d(X,U) + 2\,d(Y,V)\,. \tag{5.43}$$

Proof. Let α be defined as in Lemma 5.4.1. The first two parts of the lemma have been proved above and are recalled here for convenience. To prove the last part, we write,

$$\begin{aligned}
\|UV^T - XY^T\|_2 &\leq \|(U - X\alpha^T)V^T\|_2 + \|X\alpha^T V^T - XY^T\|_2 \\
&\leq \|(U - X\alpha^T)V^T\|_2 + \|X(Y^TY)(V^TY)^{-1}V^T - XY^T\|_2 \\
&\leq \|(U - X\alpha^T)V^T\|_2 + \|Y - V(Y^TV)^{-1}(Y^TY)\|_2 \\
&\leq \|(U - X\alpha^T)V^T\|_2 \\
&\quad + \|Y - V(Y^TV)^{-1}(Y^TV) - V(Y^TV)^{-1}Y^T(Y - V)\|_2 \\
&\leq \|(U - X\alpha^T)V^T\|_2 + \|(I - V(Y^TV)^{-1}Y^T)(Y - V)\|_2
\end{aligned}$$

To bound the second term above, consider,

$$\begin{aligned}
I - V(Y^TV)^{-1}Y^T &= I - V\left\{(V^TV)^{-1} - (V^TV)^{-1}(Y - V)^TV(Y^TV)^{-1}\right\}Y^T \\
&= \left\{I - V(V^TV)^{-1}V\right\} - V(V^TV)^{-1}(Y - V)\left\{I - V(Y^TV)^{-1}Y^T\right\}
\end{aligned}$$

Clearly $\|I - V(V^TV)^{-1}V\|_2 \leq 1$ and $\|V(V^TV)^{-1}(Y - V)\|_2 \leq 1/2$ which yields

$$\|I - V(Y^TV)^{-1}Y^T\|_2 \leq 2$$

which proves the desired result. □

5.4.2 Initialization step

In this section, we analyze the initialization step of the Alternate Minimization algorithm. Recall that the algorithm is initialized to the (rescaled) singular vectors of

the given matrix $\mathcal{P}_E(N)$. We begin by proving a that the matrix $\mathcal{P}_E(N)$ is close to M in operator norm. Throughout this section, we set $m = n$ for simplicity of notation. The general case is proved analogously.

Lemma 5.4.8. *Let $N = UV^T + W$ and let the entries of N be revealed according to be model described in Section 5.2. Let U and V satisfy the incoherence condition with parameter μ. Then for $\epsilon \geq C\mu r \log n$ with a probability at least $1 - 1/n^5$, we have*

$$\left\| M - \frac{n}{\epsilon}\mathcal{P}_E(N) \right\|_2 \leq C\sqrt{\frac{\mu r \log n}{\epsilon}} + \frac{n}{\epsilon}\|W^E\|_2$$

Proof. (Lemma 5.4.8) We begin by writing

$$\left\| M - \frac{n}{\epsilon}\mathcal{P}_E(N) \right\|_2 \leq \frac{n}{\epsilon}\left\| \frac{\epsilon}{n}M - \mathcal{P}_E(M) \right\|_2 + \frac{n}{\epsilon}\|W^E\|_2$$

To compute the first part, note that

$$\frac{\epsilon}{n}M - \mathcal{P}_E(M) = \sum_{ij} M_{ij}\left(Z_{ij} - \frac{\epsilon}{n}\right) e_i e_j^T$$

where Z_{ij} are independent Bernoulli(ϵ/n) random variables and e_i is a unit vector with 1 at index i and 0 everywhere else. Let

$$\psi_{ij} = \begin{pmatrix} 0 & M_{ij}(Z_{ij} - \frac{\epsilon}{n})e_i e_j^T \\ M_{ij}(Z_{ij} - \frac{\epsilon}{n})e_j e_i^T & 0 \end{pmatrix}$$

Clearly $\mathbb{E}[\psi_{ij}] = 0$.

$$\|\psi_{ij}\|_2^2 \leq C'\frac{\mu r}{n}$$

and

$$\begin{aligned}\left\|\sum_{ij}\mathbb{E}[\psi_{ij}^2]\right\|_2 &\leq C'\frac{\epsilon}{n}\left(1-\frac{\epsilon}{n}\right)\left\|\sum_{ij}M_{ij}^2 e_i e_i^T\right\|_2 + C'\frac{\epsilon}{n}\left(1-\frac{\epsilon}{n}\right)\left\|\sum_{ij}M_{ij}^2 e_j e_j^T\right\|_2 \\ &\leq C'\frac{\epsilon}{n}\left(\max_i\sum_j M_{ij}^2 + \max_j\sum_i M_{ij}^2\right) \\ &\leq C'\frac{\epsilon\mu r}{n^2}.\end{aligned}$$

Hence, using Bernstein inequality, we get

$$\mathbb{P}\left(\left\|\frac{\epsilon}{n}M - M^E\right\|_2 \geq C'\frac{\sqrt{\epsilon\mu r\log n}}{n}\right) \leq \frac{1}{n^{10}}$$

for $\epsilon \geq C\mu r\log n$ which proves the desired result. □

We now compute $\|W^E\|_2$ for the specific model of W described in Section 5.2.1.

Lemma 5.4.9. *Let W_{ij} be independent sub-Gaussian random variables with mean $\overline{W}_{ij}$ and parameter ω. Define $\theta \equiv \max\{\max_i\sqrt{\sum_j \overline{W}_{ij}^2}, \max_j\sqrt{\sum_i \overline{W}_{ij}^2}\}$ and assume that $\overline{W}_{ij}^2 \leq \mu r\theta^2/n$ for all (i,j) and $\theta^2 \leq \mu r\|\overline{W}\|_2^2/n$. Then, for $\epsilon \geq C\mu r\log n$*

$$\mathbb{P}\left(\|W^E\|_2 \geq C'\frac{\epsilon}{n}\|\overline{W}\|_2 + C'\frac{n\omega}{\sqrt{\epsilon}}\right) \leq 1/n^{10}$$

Proof. (Lemma 5.4.9) We begin by writing

$$\begin{aligned}\|W^E\|_2 &= \|\mathcal{P}_E(\overline{W}) + \mathcal{P}_E(W-\overline{W})\|_2 \\ &\leq \|\mathcal{P}_E(\overline{W})\|_2 + \|\mathcal{P}_E(W-\overline{W})\|_2\end{aligned}$$

From Theorem 1.3 of [60], we know that with probability larger than $1-1/n^{10}$,

$$\|\mathcal{P}_E(W-\overline{W})\|_2 \leq C'\omega\sqrt{\epsilon}. \tag{5.44}$$

For the first part, we know that

$$\begin{aligned} \|\mathcal{P}_E(\overline{W})\|_2 &= \Big\| \sum_{ij} \overline{W}_{ij} Z_{ij} e_i e_j^T \Big\|_2 \\ &\leq \Big\| \sum_{ij} \overline{W}_{ij} \left(Z_{ij} - \frac{\epsilon}{n} \right) e_i e_j^T \Big\|_2 + \frac{\epsilon}{n} \|\overline{W}\|_2 \end{aligned}$$

where Z_{ij} are independent Bernoulli(ϵ/n) random variables. Let

$$\xi_{ij} = \begin{pmatrix} 0 & \overline{W}_{ij} Z_{ij} e_i e_j^T \\ \overline{W}_{ij} Z_{ij} e_i e_j^T & 0 \end{pmatrix}$$

Clearly $\mathbb{E}[\xi_{ij}] = 0$. Further,

$$\begin{aligned} \|\xi_{ij}\|_2^2 &\leq \overline{W}_{ij}^2 \\ &\leq \left(\frac{\mu r}{n} \right)^2 \|\overline{W}\|_2^2 \end{aligned}$$

and

$$\begin{aligned} \Big\| \sum_{ij} \mathbb{E}[\xi_{ij}^2] \Big\|_2 &\leq C' \frac{\epsilon}{n} \left(1 - \frac{\epsilon}{n} \right) \left(\Big\| \sum_{ij} \overline{W}_{ij}^2 e_i e_i^T \Big\|_2 + \Big\| \sum_{ij} \overline{W}_{ij}^2 e_j e_j^T \Big\|_2 \right) \\ &\leq C' \frac{\epsilon}{n} \theta^2 \\ &\leq C' \frac{\epsilon \mu r}{n^2} \|\overline{W}\|_2^2 \end{aligned}$$

and hence, using Bernstein inequality, we have that

$$\mathbb{P}\left(\|\mathcal{P}_E(\overline{W})\|_2 \geq C' \frac{\epsilon}{n} \|\overline{W}\|_2 \right) \leq \frac{1}{n^{10}}$$

for $\epsilon \geq C \mu r \log n$ which, combined with equation (5.44) yields the desired result. □

We now consider the initialization of the Alternate Minimization algorithm. Let $X \in \mathbb{R}^{n \times r}$, $S \in \mathbb{R}^{r \times r}$, $Y \in \mathbb{R}^{r \times r}$ be the first r singular value decomposition of $n/\epsilon \mathcal{P}_E(N)$. We can now prove the following lemma which bounds the distance between U and X in operator norm.

Lemma 5.4.10. *Consider the model described in Section 3.1. Let $N = UV^T + W$. Let X, S, Y be the first r singular value decomposition of $\frac{n}{\epsilon}\mathcal{P}_E(N)$. Let W_{ij} satisfy the conditions described in Section 5.2.1. Let U and V satisfy the incoherence assumption with parameter μ. Then, for $\epsilon \geq C\mu r \log n$ with probability greater than $1 - 1/n^{10}$, there exists an $\alpha \in \mathbb{R}^{r\times r}$ such that*

$$\|V - Y\alpha\|_2 \leq C'\sigma_{\min}(U)^{-1}\sqrt{\frac{\mu r \log n}{\epsilon}} + C'\sigma_{\min}(U)^{-1}\|\overline{W}\|_2 + C'\sigma_{\min}(U)^{-1}\left(\frac{n\omega}{\sqrt{\epsilon}}\right)$$

Proof. (Lemma 5.4.10) We know that,

$$\begin{aligned}
\|UV^T - XSY^T\|_2 &\leq \|UV^T - \frac{n}{\epsilon}\mathcal{P}_E(N)\|_2 + \|XSY^T - \frac{n}{\epsilon}\mathcal{P}_E(N)\|_2 \\
&\leq \|UV^T - \frac{n}{\epsilon}\mathcal{P}_E(N)\|_2 + \frac{n}{\epsilon}\sigma_{r+1}(\mathcal{P}_E(N)) \\
&\overset{(a)}{\leq} \|UV^T - \frac{n}{\epsilon}\mathcal{P}_E(N)\|_2 + \left(\|UV^T - \frac{n}{\epsilon}\mathcal{P}_E(N)\|_2 + \sigma_{r+1}(UV^T)\right) \\
&\leq 2\|UV^T - \frac{n}{\epsilon}\mathcal{P}_E(N)\|_2
\end{aligned}$$

In the second inequality, we have used the fact that (X, S, Y) is the first r singular value decomposition of $n/\epsilon\mathcal{P}_E(N)$. In (a) above, we have used the inequality : $\sigma_k(A+B) \leq \sigma_k(A) + \|B\|_2$ where $\sigma_k(A)$ is the k^{th} singular value of A and A, B are arbitrary matrices. Further, using Lemmas 5.4.8 and 5.4.9, we have that

$$\|UV^T - XSY^T\|_2 \leq C'\sqrt{\frac{\mu r \log n}{\epsilon}} + C'\frac{\epsilon}{n}\|\overline{W}\|_2 + C'\frac{n\omega}{\sqrt{\epsilon}} \tag{5.45}$$

Finally,

$$\begin{aligned}
\|V - YSX^TU(U^TU)^{-1}\|_2 &= \|(VU^T - YSX^T)U(U^TU)^{-1}\|_2 \\
&\leq \|(VU^T - YSX^T)\|_2\|U(U^TU)^{-1}\|_2 \\
&\leq C'\sigma_{\min}(U)^{-1}\sqrt{\frac{\mu r \log n}{\epsilon}} + C'\sigma_{\min}(U)^{-1}\|\overline{W}\|_2 \\
&\quad + C'\sigma_{\min}(U)^{-1}\left(\frac{n\omega}{\sqrt{\epsilon}}\right)
\end{aligned}$$

where we have used equation (5.45) and the fact that $\|U(U^TU)^{-1}\|_2 \leq \sigma_{\min}(U)^{-1}$. Finally, choosing $\alpha = SX^TU(U^TU)^{-1}$ yields the desired result. $\square$

We now prove the following lemma about $\max_j \|v_j - (YSX^TU(U^U)^{-1})_j\|_2$, where we use the notation X_i to denote the (transpose of) the i^{th} row of X.

Lemma 5.4.11. *Consider the model described in Section 3.1. Let X, S, Y be as defined above. Let W_{ij} be independent sub-Gaussian random variables with mean $\overline{W}_{ij}$ and parameter ω. Define $\theta \equiv \max\{\max_i \sqrt{\sum_j \overline{W}_{ij}^2}, \max_j \sqrt{\sum_i \overline{W}_{ij}^2}\}$ and assume that $\overline{W}_{ij}^2 \leq \mu r\theta^2/n$ for all (i,j) and $\theta^2 \leq \mu r\|\overline{W}\|_2^2/n$. Let α be defined in the statement of Lemma 5.4.10. Then, for $\epsilon \geq C'\kappa^8\mu r \log n$, we have that with high probability,*

$$\max_j \|v_j - \alpha^T y_j\|_2 \leq \sqrt{\frac{\mu r}{n}}\kappa\left(\sqrt{\frac{C\mu r \log n}{\epsilon}} + C\left(\frac{n\omega}{\sqrt{\epsilon}}\right) + C\|\overline{W}\|_2\right)$$

Proof. (Lemma 5.4.11) Note that,

$$\begin{aligned}
\|v_j - \alpha^T y_j\|_2 &= \|e_j^T \left(V - YSX^TU(U^TU)^{-1}\right)\|_2 \\
&= \|e_j^T \left(VU^T - YSX^T\right) U(U^TU)^{-1}\|_2 \\
&= \|e_j^T \left(VU^T - \frac{n}{\epsilon}\mathcal{P}_E(N)XX^T\right) U(U^TU)^{-1}\|_2 \\
&\leq \left\|v_j^T \left(U^T - \frac{n}{\epsilon}\sum_{i\in\partial j} u_i x_i^T X^T\right) U(U^TU)^{-1}\right\|_2 \\
&\quad + \frac{n}{\epsilon}\left\|\sum_{i\in\partial j} W_{ij}x_i\right\|_2 \|XU(U^TU)^{-1}\|_2 \\
&\leq \|v_j^TU^T \left(I - XX^T\right) U(U^TU)^{-1}\|_2 \\
&\quad + \left\|v_j^T \left(\frac{n}{\epsilon}\sum_{i\in\partial j} u_i x_i^T - \mathbb{E}[u_i x_i^T]\right)(X^TU)(U^TU)^{-1}\right\|_2 \\
&\quad + \frac{n}{\epsilon}\left\|\sum_{i\in\partial j} W_{ij}x_i\right\|_2 \|XU(U^TU)^{-1}\|_2 \\
&\leq \sqrt{\frac{\mu r}{n}}\left(\|U^T \left(I - XX^T\right) U(U^TU)\|_2 + \left\|\frac{n}{\epsilon}\sum_{i\in\partial j} u_i x_i^T - \mathbb{E}[u_i x_i^T]\right\|_2\right) \\
&\quad + \frac{n}{\epsilon}\left\|\sum_{i\in\partial j} W_{ij}x_i\right\|_2 \|XU(U^TU)^{-1}\|_2 \qquad (5.46)
\end{aligned}$$

In the following, we bound each of the objects in the above equation. First,

$$U^T \left(I - XX^T\right) U(U^TU)^{-1} = U^TX_\perp X_\perp^T U(U^TU)^{-1}$$

Further, since $X^TX_\perp = 0$, we have that

$$\begin{aligned}
\|U^TX_\perp\|_2 &= \|\left(U^T - (V^TV)^{-1}V^TYSX^T\right) X_\perp\|_2 \\
&= \|(V^TV)^{-1}V^T \left(VU^T - YSX^T\right) X_\perp\|_2 \\
&\leq \sigma_{\min}(V)^{-1}\|UV^T - XSY^T\|_2
\end{aligned}$$

which yields,

$$\|U^T\left(I - XX^T\right)U(U^TU)^{-1}\|_2 \leq \sigma_{\min}(U)^{-2}\sigma_{\min}(V)^{-2}\|UV^T - XSY^T\|_2^2$$

For the second and third term in equation (5.46), we use results similar to those in Lemma 5.4.3 and we get,

$$\begin{aligned}
\left\|\frac{n}{\epsilon}\sum_{i\in\partial j} u_i x_i^T - \mathbb{E}[u_i x_i^T]\right\|_2 &\leq \frac{\|X\|_\infty}{\sqrt{\mu r/n}}\sqrt{\frac{C\mu r\log n}{\epsilon}} \\
\frac{n}{\epsilon}\left\|\sum_{i\in\partial j} W_{ij}x_i\right\|_2\|XU(U^TU)^{-1}\|_2 &\leq \frac{\|X\|_\infty}{\sqrt{\mu r/n}}\left(C\left(\frac{n\omega}{\sqrt{\epsilon}}\right)\sqrt{\frac{r\log n}{n}} + C\|\overline{W}\|_2\sqrt{\frac{\mu r}{n}}\right)
\end{aligned}$$

where we have used the notation $\|X\|_\infty \equiv \max_j \|x_j\|_2$. Further,

$$\|X\|_\infty \leq \|XSY^TV(V^TV)^{-1}\|_\infty\|V^TV(Y^TV)^{-1}S^{-1}\|_2$$

Also,

$$\begin{aligned}
\sigma_{\min}(S(Y^TV)(V^TV)^{-1}) &\geq \sigma_{\min}(XS(Y^TV)(V^TV)^{-1}) \\
&= \sigma_{\min}(U - (UV^T - XSY^T)V(V^TV)^{-1}) \\
&\geq \sigma_{\min}(U) - \|UV^T - XSY^T\|_2\sigma_{\min}(V) \\
&\geq \sigma_{\min}(U)/2
\end{aligned}$$

using $\epsilon \geq C'\kappa^8\mu r\log n$ and the assumption of equation (5.8) Denote $\beta \equiv SY^TV(V^TV)^{-1}$. Then,

$$\|X\|_\infty \leq 2\sigma_{\min}(U)^{-1}\left(\sqrt{\frac{\mu r}{n}} + \|U - X\beta\|_\infty\right)$$

This yields

$$\begin{aligned}\|V - Y\alpha\|_\infty &\leq \sqrt{\tfrac{\mu r}{n}}\left(\sigma_{\min}(U)^{-2}\sigma_{\min}(V)^{-2}\|UV^T - XSY^T\|_2^2\right) \\ &\quad +2\sigma_{\min}(U)^{-1}\left(\sqrt{\tfrac{\mu r}{n}} + \|U - X\beta\|_\infty\right)\left(\sqrt{\tfrac{C\mu r \log n}{\epsilon}} + C\left(\tfrac{n\omega}{\sqrt{\epsilon}}\right) + C\|\overline{W}\|_2\right)\end{aligned}$$

We also know that for $\epsilon \geq C\kappa^8 \mu r \log n$ and using equation (5.8)

$$\sigma_{\min}(V)^{-2}\sigma_{\min}(U)^{-1}\|UV^T - XSY^T\|_2 \leq 1/4$$

which yields

$$\begin{aligned}\|V - Y\alpha\|_\infty &\leq \sqrt{\frac{\mu r}{n}}\kappa\left(\sqrt{\frac{C\mu r \log n}{\epsilon}} + C\left(\frac{n\omega}{\sqrt{\epsilon}}\right) + C\|\overline{W}\|_2\right) \\ &\quad +\|U - X\beta\|_\infty\kappa\left(\sqrt{\frac{C\mu r \log n}{\epsilon}} + C\left(\frac{n\omega}{\sqrt{\epsilon}}\right) + C\|\overline{W}\|_2\right) \quad (5.47)\end{aligned}$$

By symmetry of the problem, we have that

$$\begin{aligned}\|U - X\beta\|_\infty &\leq \sqrt{\frac{\mu r}{n}}\kappa\left(\sqrt{\frac{C\mu r \log n}{\epsilon}} + C\left(\frac{n\omega}{\sqrt{\epsilon}}\right) + C\|\overline{W}\|_2\right) \\ &\quad +\|V - Y\alpha\|_\infty\kappa\left(\sqrt{\frac{C\mu r \log n}{\epsilon}} + C\left(\frac{n\omega}{\sqrt{\epsilon}}\right) + C\|\overline{W}\|_2\right) \quad (5.48)\end{aligned}$$

Substituting equation (5.48) into equation (5.47), we get

$$\begin{aligned}\|V - Y\alpha\|_\infty &\leq \sqrt{\frac{\mu r}{n}}\kappa\left(\sqrt{\frac{C\mu r \log n}{\epsilon}} + C\left(\frac{n\omega}{\sqrt{\epsilon}}\right) + C\|\overline{W}\|_2\right) \\ &\quad +\sqrt{\frac{\mu r}{n}}\kappa^2\left(\sqrt{\frac{C\mu r \log n}{\epsilon}} + C\left(\frac{n\omega}{\sqrt{\epsilon}}\right) + C\|\overline{W}\|_2\right)^2 \\ &\quad +\|V - Y\alpha\|_\infty\kappa^2\left(\sqrt{\frac{C\mu r \log n}{\epsilon}} + C\left(\frac{n\omega}{\sqrt{\epsilon}}\right) + C\|\overline{W}\|_2\right)^2\end{aligned}$$

Using equation (5.8) we get that for $\epsilon \geq C\kappa^8 \mu r \log n$,

$$\|v_j - \alpha^T y_j\|_2 \leq \sqrt{\frac{\mu r}{n}}\kappa \left(\sqrt{\frac{C\mu r \log n}{\epsilon}} + C\left(\frac{n\omega}{\sqrt{\epsilon}}\right) + C\|\overline{W}\|_2 \right)$$

□

Finally, we collect the results of Lemmas 5.4.10 and 5.4.11 into the following lemma. Since we consider this result of independent interest, we state it in the general case of m, n distinct.

Lemma 5.4.12. *There exist a universal constants C such that the following happens. Assume that the conditions in Section 5.2.1 hold, and let*

$$\mathcal{P}_E(N) = \sum_{i=1}^{\min\{m,n\}} \sigma_i X_i Y_i^T$$

be the singular value decomposition of $\mathcal{P}_E(N)$ with $X_i \in \mathbb{R}^m$, $Y_i \in \mathbb{R}^n$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(m,n)} \geq 0$. Let $X = [X_1, |\cdots|X_r] \in \mathbb{R}^{m\times r}$ and $Y = [Y_1, |\cdots|Y_r] \in \mathbb{R}^{n\times r}$.

If $\epsilon \geq C_1\kappa^8 \mu r \log n$, we have that with probability greater than $1 - 1/n^5$, there exists an α such that

$$d(X, U) \leq \frac{\sigma_{\min}(U)}{2}, \qquad d(Y, V) \leq \frac{\sigma_{\min}(V)}{2}.$$

Proof. From Lemma 5.4.10, we have that

$$\|V - Y\alpha\|_2 \leq C'\sigma_{\min}(U)^{-1}\sqrt{\frac{\mu r \log n}{\epsilon}} + C'\sigma_{\min}(U)^{-1}\|\overline{W}\|_2 + C'\sigma_{\min}(U)^{-1}\left(\frac{n\omega}{\sqrt{\epsilon}}\right)$$

Using $\epsilon \geq C\kappa^8 \mu r \log n$ and equation (5.8), we have that

$$\|V - Y\alpha\|_2 \leq \sigma_{\min}(V)/2$$

Further, using Lemma 5.4.11, we have that

$$\begin{aligned}\max_j \|v_j - \alpha^T y_j\|_2 &\leq \sqrt{\frac{\mu r}{n}}\kappa\left(\sqrt{\frac{C\mu r \log n}{\epsilon}} + C\left(\frac{n\omega}{\sqrt{\epsilon}}\right) + C\|\overline{W}\|_2\right) \\ &\leq \sigma_{\min}(V)/2\sqrt{\frac{\mu r}{n}}\end{aligned}$$

□

Chapter 6

Comparisons

The previous chapters introduced efficient algorithms for the matrix completion problem. We described OptSpace [59] in Chapter 4 and Alternating Least Squares and Message Passing [57] in Chapter 5. We also presented analytical results concerning the performance of OptSpace and Alternating Least Squares. In this chapter, we will put these algorithms and results in perspective.

We begin by assuming the noiseless scenario ($W = 0$) in Section 6.1. In this context, we will look at fundamental limits for matrix completion in Section 6.1.1. These limits on performance arise from simple considerations like graph connectivity [58], the coupon collector problem [43] and structural rigidity [98].

Recently, there has been a parallel line of work on matrix completion that is based on Nuclear Norm Minimization [21, 24, 49]. Section 6.1.2 introduces this approach and discusses some of the guarantees derived for nuclear norm minimization. Finally, we compare the guarantees derived for OptSpace and Alternating Least Squares with these results.

A related but slightly different problem, sampling based low rank approximation [4, 46, 35, 37], is the subject of Section 6.1.3. Here, the entire matrix is assumed to be "known" or available and the "best" low rank approximation to the matrix is desired. However, computational considerations prohibit the use of computationally complex algorithms. This naturally leads to low rank approximations based on sampling the matrix. However, this is different from matrix completion since the sampling function

can be controlled. We present some representative results in this scenario and compare those to the performance of our algorithms.

Section 6.2 deals with the noisy scenario. We begin by presenting lower bounds on the reconstruction error derived by [20]. An important special case is when the noise is i.i.d Gaussian. In this scenario, [82] use a minimax argument to derive a lower bound. We compare these lower bounds and the upper bound for Nuclear Norm Minimization [20] to the guarantees derived for OptSpace and Alternating Least Squares.

The analytical results presented for OptSpace and Alternating Least Squares depict a very satisfactory situation. However, we would like to investigate the effectiveness of these algorithms in practice. We do this through a series of numerical experiments in Section 6.3.

We begin Section 6.3 by presenting a brief survey of related algorithmic approaches to matrix completion in Section 6.3.1. Many of these [19, 73, 65] are efficient approaches to solving the nuclear norm minimization problem[1]. Others are based on well studied collaborative filtering approaches [100, 90, 103]. We then present the results of our simulations and compare with some of these algorithms in both the noiseless and the noisy scenarios in Section 6.3.2. In Section 6.3.3, we apply our algorithms to standard collaborative filtering datasets like the Netflix dataset [3], the Movielens dataset [2] and the Jester Jokes dataset [1] and compare the results obtained to those reported in the literature. This chapter is based on joint work with Montanari and Oh [58, 59, 60, 61, 57].

6.1 Noiseless Scenario

In this section, we will consider the case when the matrix entries are revealed exactly. Recall that in this case, Theorem 4.2.2 states that OptSpace can reconstruct the matrix exactly from a uniformly random sample of size $O(n\mu r \max\{\log n, \mu r\})$.

[1]Standard SDP based approaches to solving the problem are computationally expensive - $O(n^3)$ - and hence unsuited for large datasets.

The corresponding result was proved for Alternating Least Squares in Theorem 5.2.1 which can achieve an error smaller than n^{-C} for any arbitrary C with $O(n\mu r(\log n)^2)$ samples. This is analogous to exact reconstruction in most scenarios.

6.1.1 Fundamental Limits

Let us begin the investigation on fundamental limits of matrix completion by considering the simple but instructive case of rank $r = 1$. In this case, graph-theoretical interpretations provide the limits we seek. Assume that we know 3 entries of the matrix M that belong to the same 2×2 minor. Explicitly, for two row indices $i, j \in [m]$ and two column indices $a, b \in [n]$, the entries M_{ia}, M_{ja} and M_{ib} are known. Unless $M_{ia} = 0$, the fourth entry of the same minor is then uniquely determined $M_{jb} = M_{ja}M_{ib}/M_{ia}$. The case $M_{ia} = 0$ can be treated separately but, for the sake of simplicity we shall assume that $M_{ia} \neq 0$.

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 rule $M_{jb} = M_{ja}M_{ib}/M_{ia}$. As anticipated above, this algorithm has a nice graph-theoretic interpretation. Consider the bipartite graph $G = (R, C, E)$ with vertices R corresponding to the rows and vertices C corresponding to the columns of M and edges E for the observed entries. Essentially, R is isomorphic to $[m]$ and C to $[n]$. If a 2×2 minor has a unique unknown entry, it means that the corresponding vertices $j \in R$, $b \in C$ 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.

Let us use a parametrization of $\epsilon = \sqrt{\alpha}(\log m + w)$. The number of isolated nodes

in G is a Poisson random variable with mean e^{-w}. Hence

$$\mathbb{P}(\exists \text{ an isolated node in } G) \to 1 - e^{-e^{-w}} \tag{6.1}$$

Thus, if $w \to -\infty$, then with high probability, there exists isolated nodes in G and matrix reconstruction is impossible by any algorithm. Further, it is known [18] that for $G(n,p)$ graphs, the probability that a random graph is connected is about the same as the probability that there are no isolated nodes. Together with 6.1 the analysis could be generalized to bipartite graphs. Thus, if $w \to \infty$, then the graph is connected, and the above recursive algorithm reconstructs M. Observe that the order of $|E|$ predicted by the above analysis coincides (within $\log n$ factors) with the guarantees achieved for OptSpace and Alternating Least Squares for incoherent matrices.

The above analysis is sharpened by the following consideration from [58]. In the large n-limit only the components with $O(n)$ vertices matter (as they have $O(n^2)$edges). It is a fundamental result in random graph theory [39] that there is no such component for $\epsilon \leq 1/\sqrt{\alpha}$. For $\epsilon \geq 1/\sqrt{\alpha}$ there is one such component involving approximately $n\xi$ vertices in R and $m\zeta$ vertices in C, where (ξ, ζ) is the largest solution of

$$\xi = 1 - \exp^{-\epsilon\sqrt{\alpha}\zeta} \qquad \zeta = 1 - \exp\left(-\frac{\epsilon}{\sqrt{\alpha}}\xi\right) \tag{6.2}$$

This analysis implies the following result as proved in [58].

Proposition 6.1.1. *Let M be a random rank 1 matrix, and denote by $\xi(\epsilon)$, $\zeta(\epsilon)$ the largest solution of (6.2). Then there exists an algorithm with $O(n^2)$ complexity achieving with high probability, a room mean squared error*

$$\text{RMSE} \leq \sqrt{1 - \xi(\epsilon)\zeta(\epsilon)}M_{\max} + O(\sqrt{\log n/n})$$

where $M_{\max}$ is the largest entry of the matrix M.

This result implies that arbitrarily small root mean squared error can be achieved with a large enough constant ϵ. A very simple extension of this algorithm to general

rank-r matrices is introduced and analyzed in [77] under the assumption that all the $r \times r$ 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 is via the coupon collector's problem. Assume that there are n coupons with labels in $[n]$ that we wish to collect. But the only way of obtaining a coupon is by picking one uniformly at random from the set of all coupons $[n]$. Then, it is well known that we need to pick $O(n \log n)$ coupons to be sure of having at least one of each type. Replacing coupons with rows of the matrix M, it follows that we need to pick $O(n \log n)$ samples to be sure of sampling at least one entry from each row. Clearly, if a particular row has no samples, then no algorithm can reconstruct M exactly. This essentially leads to the lower bound of $O(n \log n)$ samples for exact matrix reconstruction.

For general rank r matrices, a stronger necessary condition was proved assuming incoherence by Candès and Tao in [24]. They showed that if

$$|E| \leq (3/4)nr\mu \log n$$

then there exist multiple μ-incoherent matrices of rank r whose entries, with probability larger than $1/2$, coincide on the revealed set E. This implies that assuming only the rank r and incoherence parameter μ, no algorithm can guarantee success with less than $(3/4)nr\mu \log n$ random samples.

The above analyses provide probabilistic limits on completion for the uniform sampling model. It is possible to obtain bounds using a complete different approach - based on rigidity theory. In [98], Singer and Cucuringu introduced an algorithm that checks if there exists multiple rank r matrices that are identical given the sampled set E. The basic idea is the following. Factorize M as $M = XY^T$ and think of the rows of X (x_i) and Y (y_j) as points in $\mathbb{R}^r$. Then the revealed matrix entries can be thought of as defining the distance between two points and the edge set E as the graph imposed on these points. Now, if this "structure" is not rigid, it means that there are multiple solutions to the positions of the points (x_i and y_j) that satisfy the given constraints.

This implies that no algorithm can reconstruct M exactly. This bound is presented for comparison with our numerical results for exact matrix reconstruction in Section 6.3.2.

6.1.2 Nuclear Norm Minimization

In the previous section, we compared the guarantees of OptSpace and Alternating Least Squares with some fundamental limits of matrix completion. In this section, we compare these with the guarantees achieved by a parallel approach to matrix completion, namely nuclear norm minimization in the noiseless scenario. In this context, a natural optimization problem for matrix completion is the following :

$$\begin{aligned} &\text{minimize} && \text{rank}(X) \\ &\text{subject to} && \mathcal{P}_E(X) = \mathcal{P}_E(M) \end{aligned} \tag{6.3}$$

where the optimization is over matrices $X \in \mathbb{R}^{m\times n}$ and $\mathcal{P}_E(.) : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ was defined in Section 3.1 as

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

That is, consider all matrices that agree with the revealed set and return the one with the smallest rank. When there a unique rank r matrix that agrees with the revealed set, this problem recovers M exactly. However, this is an NP-hard problem and has a complexity that is doubly exponential in the problem dimension [27].

To overcome this difficulty, it is common to use the nuclear norm minimization heuristic, introduced by Fazel [40, 41] :

$$\begin{aligned} &\text{minimize} && \|X\|_* \\ &\text{subject to} && \mathcal{P}_E(X) = \mathcal{P}_E(M) \end{aligned} \tag{6.4}$$

where $\|X\|_*$ denotes the nuclear norm of X, i.e the sum of the singular values of X. It is a convex function of X and is commonly used as a proxy for rank(X). It has

been shown that this problem can be formulated as an SDP [41] and can be solved by off-the-shelf solvers with a computational complexity of $O(n^4)$. Note that this is a significant improvement over solving (6.3). However, an $O(n^4)$ complexity is still prohibitive for large datasets ($n \geq 1000$). In Section 6.3.1, we present a few low complexity algorithms for solving (6.4).

Nuclear norm minimization is closely related to *c*ompressed sensing [32, 22]. Here, we wish to find the sparsest vector satisfying a set of affine constraints :

$$\begin{aligned} \text{minimize} \quad & \|x\|_0 \\ \text{subject to} \quad & Ax = b \end{aligned} \tag{6.5}$$

where $\|x\|_0$ denotes the number of non-zero entries of x. This problem is again NP-hard. A common heuristic is to replace it with the convex problem:

$$\begin{aligned} \text{minimize} \quad & \|x\|_1 \\ \text{subject to} \quad & Ax = b \end{aligned} \tag{6.6}$$

where $\|.\|_1$ denotes the ℓ_1 norm of a vector, i.e the sum of the absolute values of its entries. It was shown [32, 22] that the convex optimization problem (6.6) coincides with the solution of (6.5) under appropriate conditions on the measurement matrix A.

The similarities between the matrix completion problems (6.3), (6.4) and the compressed sensing problems (6.5, 6.6) are striking. In (6.3), the rank counts the number of non-zero singular values of a matrix, which is analogous to the ℓ_0 function. In turn, the nuclear norm measures the sum of the singular values, analogous to the ℓ_1 function. These analogies lead to interesting connections between the two problems and a number of matrix completion algorithms are derived from their compressed sensing counterparts. Some of these algorithms are described in Section 6.3.1. Further connections between compressed sensing and nuclear norm minimization are explored in [88].

The relaxed optimization problem (6.4) has a significant reduction in computational complexity compared to (6.3). But when is it a good estimator of M? This question was addressed by Candès and Recht in [21]. A significant challenge in analyzing nuclear norm minimization, as compared to compressed sensing, is that the restricted isometry property – which proved to be quite instrumental in analyzing compressed sensing – is not satisfied in the matrix completion problem. 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 with parameter μ and uniform sampling, they proved that there exists a numerical constant C such that if $|E| \geq Cn^{6/5}rlogn$, then solving (6.4) recovers M correctly with high probability.

The guarantees were further tightened using ideas from [49, 87, 24]. A major breakthrough was the introduction of powerful matrix concentration inequalities by Gross [50] in the context of quantum state tomography. The most recent analysis shows that (6.4) recovers the matrix M exactly if

$$|E| \geq Cn\mu r(\log n)^2 \tag{6.7}$$

This bound coincides with the guarantees obtained for Alternating Least Squares in Theorem 5.2.1 when the condition number κ is bounded. In addition, Theorem 5.2.1 also provides the rate of convergence which is of independent interest.

The results obtained for OptSpace in Theorem 4.2.2 guarantee exact matrix recovery if

$$|E| \geq Cn\mu r\kappa^2 \max\{\log n, \mu r\kappa^4\}$$

This improves over (6.7) when $\mu r = O((\log n)^2)$ and the matrix M has bounded condition number. Further, when the rank and the incoherence parameter are bounded, the guarantee achieved by OptSpace, namely exact reconstruction for

$$|E| \geq Cn \log n$$

coincides with the lower bounds derived in Section 6.1.1 up to constants and is therefore order-optimal. Note that in many practical applications such as positioning [97] and structure-from-motion [26], the rank is known to the small. Indeed in these applications, the rank is comparable to the ambient dimension of 3. However, the bounds achieved for OptSpace are sub-optimal in the case of

(i) Large rank : The number of samples required for reconstruction should scale linearly in r rather than quadratically as suggested by Theorem 4.2.2. However, as should be clear from the numerical experiments of Section 6.3.2, this appears to be a drawback of our analysis rather than the algorithm.

(ii) Large condition number : This appears to be a limitation of the singular value decomposition step. However, [61] introduces a simple modification of OptSpace and shows empirically that it overcomes this problem.

Finally, note that the second drawback is shared by the Alternating Least Squares algorithm which also uses the singular value decomposition for initialization.

6.1.3 Sampling Based Approximations

Low rank approximation of matrices play a important role in numerical analysis, data mining, control and a number of other applications. Singular Value Decomposition is at the heart of these low rank approximation methods for finding the optimal low rank matrix in the mean squared error sense. However, computing the SVD of a dense $n \times n$ matrix requires $O(n^3)$ operations which is prohibitive in large scale applications. Particular applications of interest include spectral clustering [83, 44] applied to image segmentation, manifold learning [95, 104], and low rank approximation of kernel matrices [110, 114].

Consider a typical example in image segmentation. To segment an image with 1000×1000 pixels requires computing the SVD of a dense $10^6 \times 10^6$ affinity matrix. Computing the SVD of this matrix exactly is infeasible. Indeed even storing the entire matrix requires a few terabytes of storage. Further, even computing the top r singular vectors and the associated singular values requires $O(n^2 r)$ operations with the power method, which is still prohibitive.

Sampling based methods provide a powerful alternative to conventional SVD algorithms. A key observation is that in many of these practical applications, it suffices to compute an approximation to the SVD. Generically, these methods sample the original matrix and compute low rank approximations based on the sampled set. This improves over the conventional method in two ways : i) It speeds up the computation and ii) It requires less storage. Observe that both OptSpace and Alternating Least Squares can be viewed in this context where the samples are constrained to be drawn from the uniformly random sampling model. In this section, we review some of the recent developments in these techniques and compare them, chiefly, with the guarantees achieved by the Projection step of OptSpace.

There are various ways of quantifying the loss due to approximation. For simplicity, let us consider square matrices of dimensions n and rank r. A commonly used metric is via the spectral norm $\|.\|_2 : \mathbb{R}^{n\times n} \to \mathbb{R}$. The goodness of an approximation S to the matrix M is measured by the spectral norm of the difference $\|M-S\|_2$. Indeed, if S_r is the best rank r approximation of S in the SVD sense, then $\|M-S_r\|_F \leq 2\sqrt{2r}\|M-S\|_2$ [4]. So the key challenge is to find a matrix S that has a small $\|M-S\|_2$ and for which, there is an efficient way to compute the low rank component S_r, usually a sparse S.

The simple uniform sampling model

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

can be shown to be quite effective. Indeed, the following Remark can be easily shown using matrix concentration inequalities [5, 107].

Remark 6.1.2. *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 $M_{\max} = \max_{i,j} M_{ij}$. An analogous result was proved in [21].

Achlioptas and McSherry proved [4] a tighter bound, eliminating the logarithmic factor but holding only for $p \geq (8 \log n)^4/n$. That is, they show that for $p \geq (8 \log n)^4/n$, with high probability,

$$\|M - S\|_2 \leq 4M_{\max}\sqrt{\frac{n}{p}}$$

The guarantees achieved by OptSpace as demonstrated in Theorem 4.2.1 improve over both these results by eliminating the logarithmic factor in the error bound and the conditions on p. Indeed, let $S_{\rm OS}$ be the sparse matrix obtained by random sampling followed by trimming. Then it was shown in [59] that, with a probability larger than $1 - 1/n^3$, there is a constant C such that

$$\|M - S_{\rm OS}\|_2 \leq CM_{\max}\sqrt{\frac{n}{p}}$$

for all p. As explained in [59] the trimming step is crucial in obtaining the tighter result.

Non uniform sampling techniques have also been explored in the literature. In [4], the authors introduce and analyze the following adaptive sampling scheme. Let pn^2 be the average number of sampled entries as before. Let

$$S_{ij} = \begin{cases} M_{ij}/p_{ij} & \text{with probability } p_{ij}\,, \\ 0 & \text{with probability } 1 - p_{ij} \end{cases}$$

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

$$\|M - S\|_2 \leq 4\sqrt{2}\frac{\|M\|_F}{n}\sqrt{\frac{n}{p}}$$

For comparison with the previous results, consider that $M_{\max} \geq \|M\|_F/n \geq M_{\max}/n$. Thus, intuitively when the matrix is well "spread out", in other words incoherent, then gain due to adaptive sampling is at most by a constant factor. However, when the matrix is not incoherent, the gain could be significant.

Following a different approach, [7] develops an adaptive sampling scheme with $p_{ij} = \min\{1, c\sqrt{n}|M_{ij}|\}$. The analysis follows a discretization technique [45, 42] similar to that used to prove Theorem 4.2.1. It was proved in [7], that with the above sampling scheme, there exists constants C and C' such that with probability larger than $1 - \exp(-C'n)$,

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

This gives a tighter bound compared to the results of [4] when $\sum_{ij} |M_{ij}| \leq n\sqrt{p}\|M\|_F$

As has been demonstrated above, the added freedom of choosing the sampling scheme provides for fruitful new directions. The question of finding the "best" sampling scheme is still open. However, the above techniques provide some of the state-of-the-art heuristics available.

An alternative approach to the techniques presented above is to sample entire rows or columns of M. Popular column sampling techniques include algorithms like ConstantTimeSVD and LinearTimeSVD [33, 46, 36], the Nyström based method [110] and the CUR decomposition method [35]. [37] contains a detailed comparison of these algorithms.

6.2 Noisy Scenario

Let us begin by recalling the guarantees obtained for OptSpace and Alternating Least Squares in the noisy scenario. For simplicity, we will assume that $m = n$ throughout this section. Let $\widehat{M}_{\text{OS}}$ be the output of OptSpace. Then, Theorem 4.2.2 bounds the error achieved by $\widehat{M}_{\text{OS}}$ as

$$\frac{1}{\sqrt{mn}}\|M - \widehat{M}_{\text{OS}}\|_F \leq C\frac{n\kappa^2\sqrt{r}}{|E|}\|W^E\|_2 \tag{6.8}$$

Analogously, let $\widehat{M}_{\text{ALS}}$ be the output of Alternating Least Squares. Then, Theorem 5.2.1 bounds the error as

$$\frac{1}{\sqrt{mn}}\|M - \widehat{M}_{\text{ALS}}\|_F \leq C\frac{\kappa^2\sqrt{r}}{n}\|\overline{W}\|_2 + C\kappa^2\sqrt{\frac{n}{|E|}}\omega \tag{6.9}$$

where $\overline{W}$ is the deterministic part of the noise W and the random part is assumed to be i.i.d sub-Gaussian with parameter ω. Note that the results for OptSpace hold for any noise matrix W.

6.2.1 Lower Bounds

Oracle estimators are a common way of obtaining lower bounds on estimation problems. Following this line of work, Candès and Plan introduced an oracle estimator for the matrix completion problem in [20]. The matrix completion problem with noise can be thought of as solving two coupled problems. On the one hand, we are attempting to reduce the dimension in which we look for the original matrix. On the other hand, we wish to obtain a good fit with the observed entries. Suppose there is an oracle informing us about a linear space in which the original matrix lives. In particular, suppose we know that $M \in T$ where

$$T = \{UY^T + XV^T | X \in \mathbb{R}^{n\times r}, Y \in \mathbb{R}^{n\times r}\}$$

This is analogous to knowing the support of the vector in compressed sensing [23]. With this information, we would know that M lives in a linear space of dimension $2nr - r^2$ and would solve the problem by the method of least squares.

$$\begin{aligned} \text{minimize} \quad & \|\mathcal{P}_E(X - N)\|_F \\ \text{subject to} \quad & X \in T \end{aligned} \tag{6.10}$$

That is, we would find the matrix T, which best fits the data in the least square sense. Let $\mathcal{A} = \mathcal{P}_E\mathcal{P}_T$, where $\mathcal{P}_T$ is the projection operator into T. Let $\mathcal{A}^*$ be the adjoint of $\mathcal{A}$ and under the hypotheses of Theorems 4.2.2 and 5.2.1, the operator

$\mathcal{A}^*\mathcal{A} = \mathcal{P}_T\mathcal{P}_E\mathcal{P}_T$ is invertible. Hence, the least squares solution to (6.10) is given by

$$\begin{aligned} \widehat{M}_{\text{Oracle}} &= (\mathcal{A}^*\mathcal{A})^{-1}\mathcal{A}^*(N) \\ &= M + (\mathcal{A}^*\mathcal{A})^{-1}\mathcal{A}^*(W) \end{aligned}$$

Hence,

$$\|\widehat{M}_{\text{Oracle}} - M\|_F = \|(\mathcal{A}^*\mathcal{A})^{-1}\mathcal{A}^*(W)\|_F$$

An analysis of the error $\|(\mathcal{A}^*\mathcal{A})^{-1}\mathcal{A}^*(W)\|_F$ is contained in [20] where it is shown that the error term concentrates around $n\|W^E\|_F/\sqrt{|E|}$. In the case of Gaussian noise with i.i.d $\mathcal{N}(0,\omega^2)$ entries, the oracle estimator has an error of

$$\frac{\|M - \widehat{M}_{\text{Oracle}}\|_F}{n} \approx \omega\sqrt{\frac{2nr - r^2}{|E|}} \tag{6.11}$$

For the Gaussian noise scenario, a lower bound on the estimation error was proved in [82] using a minimax argument. Consider a family of matrices

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

This set is slightly different from the original version but contains the set in [82]. Using an information theoretic argument, it was shown that there exists constants C, C' such that

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

where the infimum is taken over all $\widehat{M}$ that are measurable functions of the $|E|$ samples.

6.2.2 Nuclear norm minimization

The case of nuclear norm minimization with noisy observations is somewhat less understood than the noiseless scenario. Most of the earlier works consider a relaxation of the nuclear norm minimization formulation, in which the constraint in (6.4) is softened.

$$\begin{aligned} \text{minimize} \quad & \|X\|_* \\ \text{subject to} \quad & \|\mathcal{P}_E(X - N)\|_F \leq \delta \end{aligned} \tag{6.12}$$

where δ is the estimated noise power. Another implementation of this idea consists in minimizing the Lagrangian

$$\mathcal{L}_N(X) = \frac{1}{2} \|\mathcal{P}_E(X - N)\|_F^2 + \lambda \|X\|_* . \tag{6.13}$$

We will refer to either estimators as $\widehat{M}_{\text{NN}}$. Negahban and Wainwright [82] and Koltchinskii, Lounici, Tsybakov [62], considered a model in which M is not required to be incoherent, and only a bound on its largest entry is assumed[2]. Within the present notations, and for uniform sampling of the entries, these analyses yield

$$\frac{1}{\sqrt{mn}} \|M - \widehat{M}^{\text{NN}}\|_F \leq C \left(\frac{nr}{|E|}\right)^{1/2} \left\{M_{\max} + \omega\right\} \log n \tag{6.14}$$

which is dominated by the simple spectral bound of Theorem 4.2.1 when $\alpha = O(1)$. So the projection step of OptSpace and the first the initialization step of Alternating Least Squares already achieve the guarantees of [82].

The modified problem (6.12) was analyzed by Candès and Plan in [20] under the incoherent model. They show that when $|E| \geq Cn\mu r(\log n)^2$,

$$\frac{1}{\sqrt{mn}} \|M - \widehat{M}_{\text{NN}}\|_F \leq 7\sqrt{\frac{n}{|E|}} \|\mathcal{P}_E(W)\|_F + \frac{2}{\sqrt{mn}} \|\mathcal{P}_E(W)\|_F \tag{6.15}$$

[2]Ref. [82] phrases this hypothesis in terms of a 'spikiness' parameter but, for the present discussion, this is equivalent to an assumption on the size of the maximum entry.

In [20], the constant in front of the first term is slightly smaller than 7, but in any case larger than $4\sqrt{2}$.

Theorem 4.2.2 improves over this result in several respects: (1) We do not have the second term on the right-hand side of (6.15), that actually increases with the number of observed entries; (2) Our error decreases as $n/|E|$ rather than $(n/|E|)^{1/2}$; (3) The noise enters Theorem 4.2.2 through the operator norm $\|W^E\|_2$ instead of its Frobenius norm $\|W^E\|_F \geq \|W^E\|_2$. For E uniformly random, one expects $\|W^E\|_F$ to be roughly of order $\|W^E\|_2\sqrt{n}$. For instance, within the independent entries model with bounded variance ω, $\|W^E\|_F = \Theta(\sqrt{|E|})$ while $\|W^E\|_2$ is of order $\sqrt{|E|/n}$ (up to logarithmic terms).

For the sake of comparison, let us consider two specific cases. First, if the noise is purely random $\overline{W} = 0$ and i.i.d sub-Gaussian with parameter ω, then

$$\begin{aligned}
\frac{1}{\sqrt{mn}}\|M - \widehat{M}_{\text{ALS}}\|_F &\leq C\omega\sqrt{nr}|E| \\
\frac{1}{\sqrt{mn}}\|M - \widehat{M}_{\text{OS}}\|_F &\leq C'\omega\sqrt{nr}|E|
\end{aligned}$$

under the hypotheses of Theorems 4.2.2 and 5.2.1 for matrices with bounded condition numbers. In this context, the oracle lower bound (6.11) coincides with the guarantees presented above and hence both OptSpace and Alternating Least Squares are order-optimal.

To compare with (6.15), it is easy to check that $(1/2)|E|\omega^2 \leq \|\mathcal{P}_E(W)\|_F^2 \leq 2|E|\omega^2$ and hence,

$$\begin{aligned}
\frac{1}{\sqrt{mn}}\|M - \widehat{M}_{\text{NN}}\|_F &\leq 16\sqrt{n}\omega \\
\frac{1}{\sqrt{mn}}\|M - \widehat{M}_{\text{ALS}}\|_F &\leq C'\omega \\
\frac{1}{\sqrt{mn}}\|M - \widehat{M}_{\text{OS}}\|_F &\leq C'\omega
\end{aligned}$$

where we have used the hypotheses of Theorems 4.2.2 and 5.2.1 to simplify the bounds for OptSpace and Alternating Least Squares. Under this condition, the bounds dominate the one in [20] by a factor of $\sqrt{n}$.

Consider next the case in which W is deterministic i.e. $W = \overline{W}$. We let

$$r_{\text{eff}}(\overline{W}) = \frac{\|\overline{W}\|_F^2}{\|\overline{W}\|_2^2} \tag{6.16}$$

denote the effective rank of $\overline{W}$. Notice that $1 \le r_{\text{eff}}(\overline{W}) \le m^{1/2}$. If the entries of $\overline{W}$ have comparable size, then with high probability $\|\mathcal{P}_E(\overline{W})\|_F^2$ concentrates around its expectation $|E|\|\overline{W}\|_F^2/(mn)$. Similarly, $\|\mathcal{P}_E(\overline{W}^E)\|_2$ concentrates around $|E|/mn\|\overline{W}\|_2$. Using these in (6.15),

$$\|M - \widehat{M}_{\text{NN}}\|_F \le C\sqrt{r_{\text{eff}}(\overline{W})\, m}\, \|\overline{W}\|_2\,, \tag{6.17}$$

$$\|M - \widehat{M}_{\text{ALS}}\|_F \le C\sqrt{r}\, \kappa^2\, \|\overline{W}\|_2\,, \tag{6.18}$$

$$\|M - \widehat{M}_{\text{OS}}\|_F \le C\sqrt{r}\, \kappa^2\, \|\overline{W}\|_2\,. \tag{6.19}$$

The upper bounds for OptSpace and Alternating Least Squares dominate the one in [20] if $\kappa^2 \le (m/r)^{1/2}$ regardless r_{eff}, and are dominated if $\kappa^2 \ge m/r^{1/2}$ regardless of r_{eff}. In the intermediate regime $(m/r)^{1/2}\kappa^2 \le m/r^{1/2}$ neither one dominates unconditionally. It would be interesting, for $\kappa^2 \ge (m/r)^{1/2}$, to redefine the decomposition $M + W$ in order to obtain a better upper bound.

6.3 Numerical comparisons

In this section, we present some empirical observations regarding the performance of OptSpace, Alternating Least Squares (standard version) and Message Passing. We experiment with randomly generated (synthetic) datasets as well as real collaborative filtering datasets. We will often compare these algorithms to other recent algorithms like Singular Value Thresholding (SVT) [19], Fixed Point Continuation with Approximate SVD (FPCA) [73] and Atomic Decomposition for Minimum Rank Approximation (ADMiRA) [65].

We implemented our algorithms in C and are publicly available[3]. The comparisons were performed on a 3.4GHz Desktop computer with 4GB RAM. For efficient singular value decomposition of sparse matrices, we used a modification of SVDLIBC[4], which is based on SVDPACKC.

6.3.1 Related Algorithms

We begin by including a survey of related algorithms in the literature. A key challenge in using solving the nuclear norm minimization problem (6.4) using SDPs is that the computational complexity of generic SDP scales like $O(n^4)$. Compare this to the $O(|E|r)$ complexity per iteration of OptSpace or the $O(|E|r^2)$ complexity per iteration of Alternating Least Squares. To overcome this issue, a number of efficient algorithms have been proposed in the literature that solve variations of the nuclear norm minimization problem.

Cai, Candès and Shen [19] proposed an efficient first-order procedure called Singular Value Thresholding (SVT) to solve the nuclear norm minimization in (6.4) directly. The algorithm essentially proceeds by alternately shrinking the singular values of the estimate and moving towards the known entries in the estimate. However, this procedure does not generalize to the case of noisy entries. As mentioned above, in this setting, the hard constrained is relaxed to

$$\text{minimize} \quad \lambda\|X\|_* + \tfrac{1}{2}\|\mathcal{P}_E(X-N)\|_F^2 \tag{6.20}$$

A number of efficient algorithms for solving (6.20) have recently been developed. Fixed Point Continuation with Approximate SVD (FPCA) [73] is based on the Bregman iterative algorithm [113]. Accelerated Proximal Gradient (APG) [105] is based on the fast iterative-thresholding algorithm [11]. The Soft-Impute and Hard-Impute algorithms developed in [75] iteratively replace the missing entries with those obtained from the thresholded SVD.

[3]available at `http://www.stanford.edu/~raghuram/optspace`
[4]available at `http://tedlab.mit.edu/~dr/SVDLIBC`

Taking a slightly different approach, Lee and Bresler consider the following problem

$$\text{minimize} \quad \|\mathcal{P}_E(X - N)\|_F \tag{6.21}$$
$$\text{subject to} \quad \text{rank}(X) \leq r \tag{6.22}$$

when the rank r is known. Inspired by CoSaMP, they introduced the Atomic Decomposition for Minimum Rank Approximation (ADMiRA) algorithm in [65] to find an approximate solution to (6.21). Singular Value Projection(SVP) [76] is another approach to solving (6.21) that is inspired by Iterative Hard Thresholding (IHT) [16].

Another approach is to consider the following cost function over $X \in \mathbb{R}^{m\times r}$ and $Y \in \mathbb{R}^{n\times r}$.

$$\text{minimize} \quad \|\mathcal{P}_E(XY^T - N)\|_F$$

and r is the estimated rank of M. A number of algorithms have been proposed to minimize the above cost function [63, 12, 51, 111]. Subspace Evolution and Transfer (SET) [29] is a manifold optimization approach to this problem. Similar to OptSpace, the variables are constrained to be on the Grassmann manifold. [109] consider the following slightly modified optimization problem.

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

where the optimization is over $X \in \mathbb{R}^{m\times r}$, $Y \in \mathbb{R}^{n\times r}$ and $Z \in \mathbb{R}^{m\times n}$. They introduce the Low-rank Matrix Fitting (LMaFit) algorithm to minimize the above cost function. Note that the above optimization problem is not convex. However, Wen et. al. [109] provide empirical evidence to demonstrate the usefulness of LMaFit.

Approaches similar to Alternating Least Squares have been considered extensively in the collaborative filtering literature. Srebro and Jaakkola introduced the Weighted Low Rank Approximation problem in [100]. Here, gradient descent methods

are used to minimize

$$\sum_{ij} \theta_{ij} \left(XY^T - N)\right)_{ij}^2$$

over $X \in \mathbb{R}^{m\times r}$ and $Y \in \mathbb{R}^{n\times r}$ and θ_{ij} are fixed weights. Note that if θ_{ij} are all identical, the above problem reduces to the Singular Value Decomposition problem. One the other hand, if $\theta_{ij} = 1$ for all $(i, j) \in E$ and 0 otherwise, then the above reduces to the exact matrix reconstruction problem.

The use of objective functions of the form $\|\mathcal{P}_E(XY^T - N)\|_F$ was justified in [91] using a probabilistic model. Assume that each observed entry N_{ij} is the dot product $u_i^T v_j$ corrupted by Gaussian noise. Further assume a Gaussian prior on u_i and v_j. Then, the log posterior naturally leads to the following optimization problem.

$$\text{minimize} \quad \tfrac{1}{2}\|\mathcal{P}_E(XY^T - N)\|_F^2 + \lambda\|X\|_F^2 + \lambda\|Y\|_F^2 \,. \tag{6.23}$$

Gradient descent was used to minimize the above objective function in [91]. Real datasets are often highly non-uniform in terms of the number of revealed entries per row or column. This suggests that non-uniform regularization for the factors could yield better results. This approach was pursued in [93] where they suggest regularizing the factors proportional to the number of sampled entries from the particular row or column.

$$\text{minimize} \quad \tfrac{1}{2}\|\mathcal{P}_E(XY^T - N)\|_F^2 + \lambda \textstyle\sum_i p_i\|x_i\|_F^2 + \lambda \sum_j q_j\|y_j\|_F^2 \,.$$

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

It is possible to draw interesting relations between the non-convex cost function (6.23) and nuclear norm minimization (6.20) [88, 90, 101]. Indeed the basic connection is easy to see from the following identity.

$$\|A\|_* = \frac{1}{2} \min_{XY^T=A} \left(\|X\|_F^2 + \|Y\|_F^2\right) . \tag{6.24}$$

It is immediate from the above identity that the quadratic regularization terms of (6.23) induce the nuclear norm regularization of (6.20). The key difference is that (6.20) is a convex problem, and is hence solvable in principle, often with high complexity solvers. On the other hand, (6.23) is a non-convex problem but the cost function is differentiable with respect to the optimization variables. This opens the way to a number of gradient descent and similar heuristic, but efficient, approaches.

In collaborative filtering applications, matrix completion was also studied from a graphical models perspective in [92], 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 RBMs have several advantages over spectral methods for collaborative filtering. More recently, [9] used a graphical model to characterize the probability distribution underlying the collaborative filtering dataset. A message passing algorithm, dubbed IMP, was introduced to infer the underlying distribution from the observed entries.

6.3.2 Synthetic Datasets

In this section, the data matrices are generated as $M = UV^T$ where $U, V \in \mathbb{R}^{n\times r}$, $m = n$, and the entries of U and V are independent $\mathsf{N}(0, 1/\sqrt{n})$ random variables. Each entry of the matrix M is revealed independently with a probability p. With a slight abuse of notation, we will use $|E|$ to denote the expected number of observed entries, i.e. $|E| = pmn$. (In reality $|E|$ concentrates around pmn.)

Exact matrix completion

We begin by experimenting with the algorithms in the noiseless scenario. We 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 6.1 shows the reconstruction rates of the different algorithms as a function of the average number of entries per row (or column) $|E|/n$ for $n = 1000$ and ranks

10 and 20. Figures 6.2 and 6.3 show the corresponding results for $n = 2000$ and $n = 4000$ respectively. The results of our algorithms are compared against SVT[5], FPCA[6] and ADMiRA. A threshold behavior characterizes all of the algorithms. As predicted from the theory, the threshold location in $|E|$ scales linearly both with n and with r. For reference, we have also plotted the bound proved in [98], below which no algorithm can correctly recover M. We see that the performance of Message Passing is very close and sometimes indistinguishable from the bound. For more extensive comparisons, we refer to [61]

Matrix completion from noisy entries

Real life situations are seldom as clean as was depicted in the previous section. Most datasets come with associated noise and we would like our algorithms to be robust against noise. To model this scenario, we carried out experiments with additive noise. We corrupt the entries with i.i.d Gaussian noise $W_{ij} \sim \mathsf{N}(0, \omega^2)$ and study the root mean squared error RMSE $\equiv \|M - \widehat{M}\|_F^2/mn$ achieved.

In the first set of experiments, we set the noise variance such that the noise ratio is $\|W\|_F/\|M\|_F = 2$ in expectation. We run the algorithms with varying sizes of the revealed set. In Figure 6.4, we plot the RMSE as a function of the sampling probability for matrices with dimensions $n, m = 1000$ and rank $r = 10$ (left panel) and $r = 20$ (right panel). For comparison, we also plot the oracle lower bound (6.11). It is clear from the plots that both OptSpace and Message Passing almost coincide with the lower bound for datasets of size $|E|$ larger than a small threshold. We also compare the performance of the algorithms with that of ADMiRA [65]. In Figures 6.5 and 6.6, we present the corresponding results for matrix dimensions $n = 2000$ and $n = 4000$ respectively. The qualitative performance of the algorithms are essentially unchanged even for large dimensional matrices.

In the previous set of experiments, we used the unregularized cost function for both OptSpace and Message Passing. In the following set of experiments, we use

[5]available at `http://svt.caltech.edu`

[6]available at `http://www.columbia.edu/~sm2756/FPCA.htm`

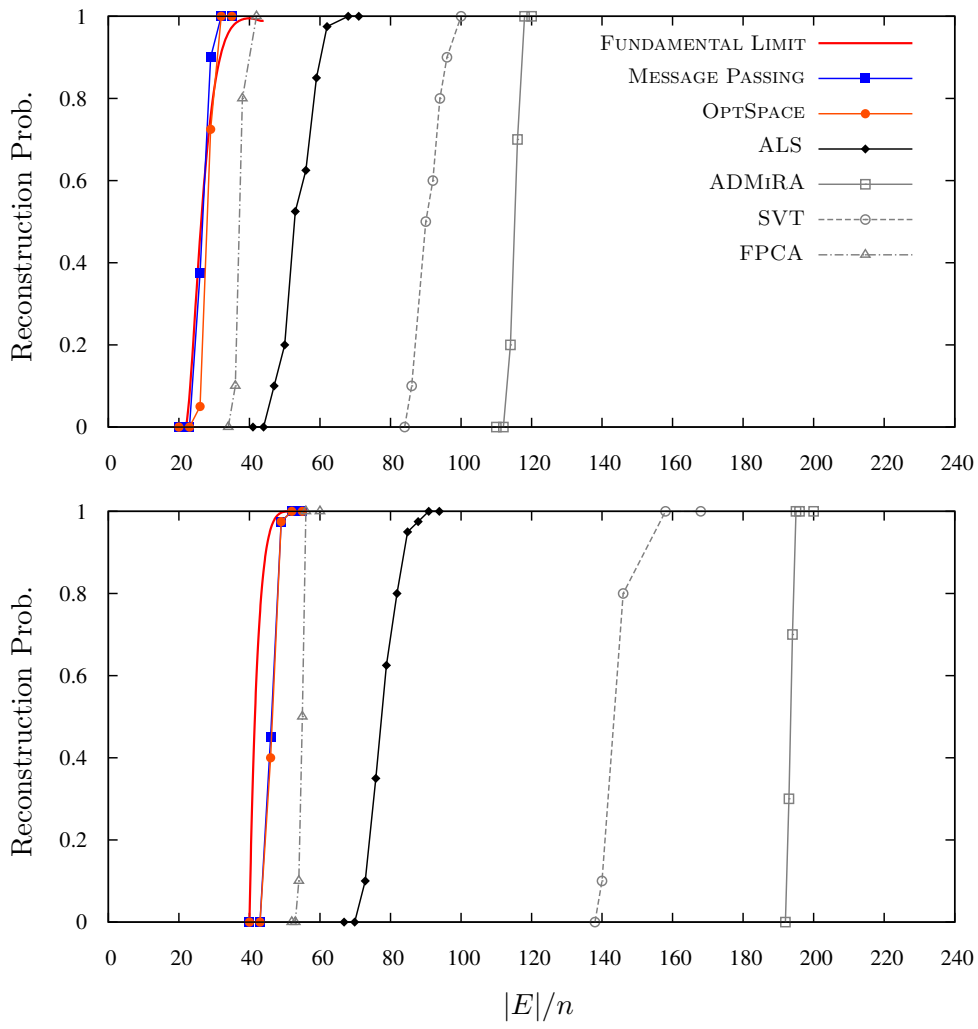

Figure 6.1: Empirical reconstruction probability as a function of $|E|/n$ for different algorithms for $n = 1000$ and $r = 10$ (top) and $r = 20$ (bottom).

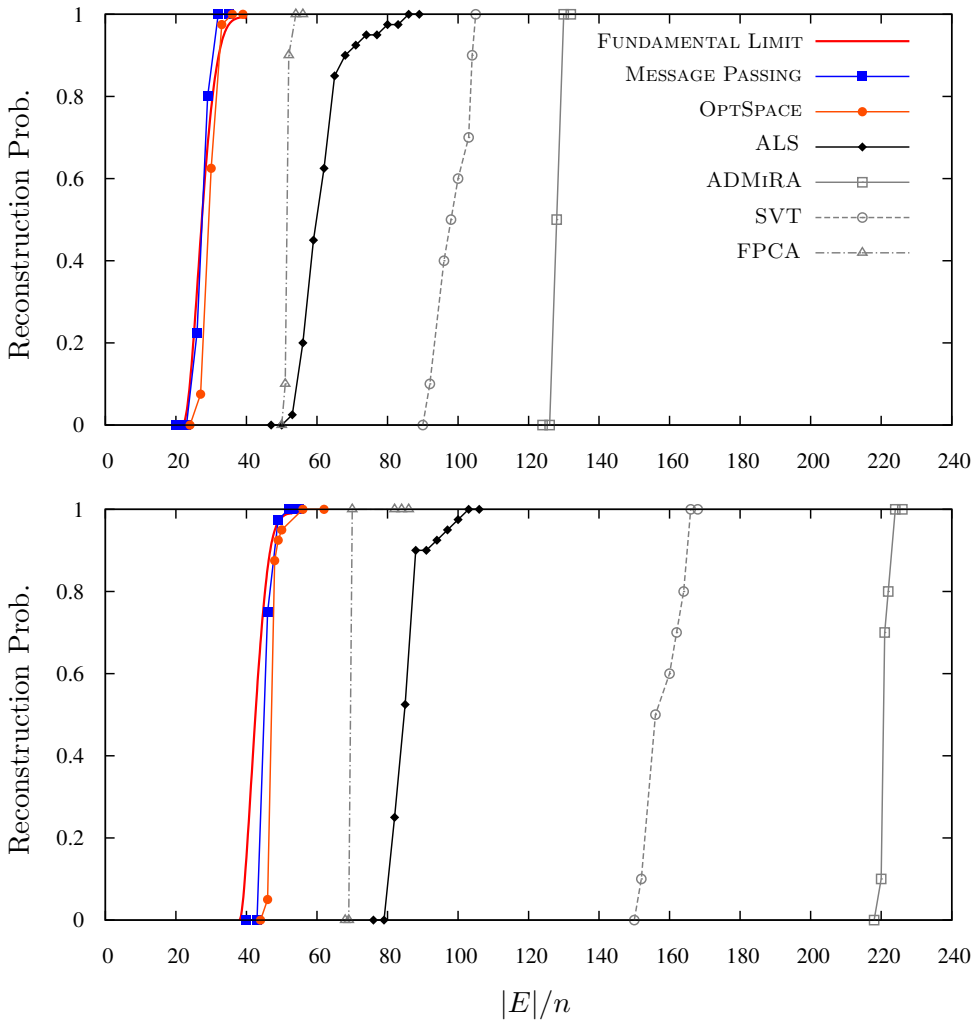

Figure 6.2: Empirical reconstruction probability as a function of $|E|/n$ for different algorithms for $n = 2000$ and $r = 10$ (top) and $r = 20$ (bottom).

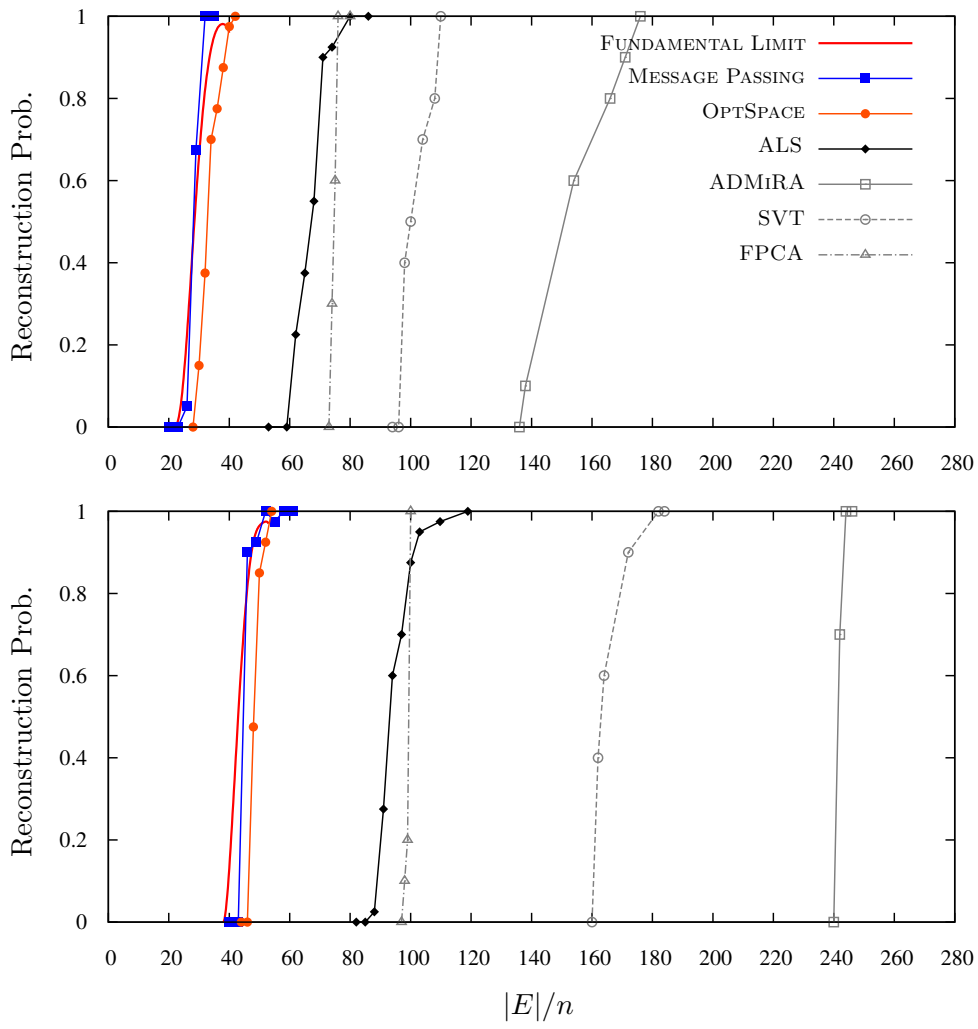

Figure 6.3: Empirical reconstruction probability as a function of $|E|/n$ for different algorithms for $n = 4000$ and $r = 10$ (top) and $r = 20$ (bottom).

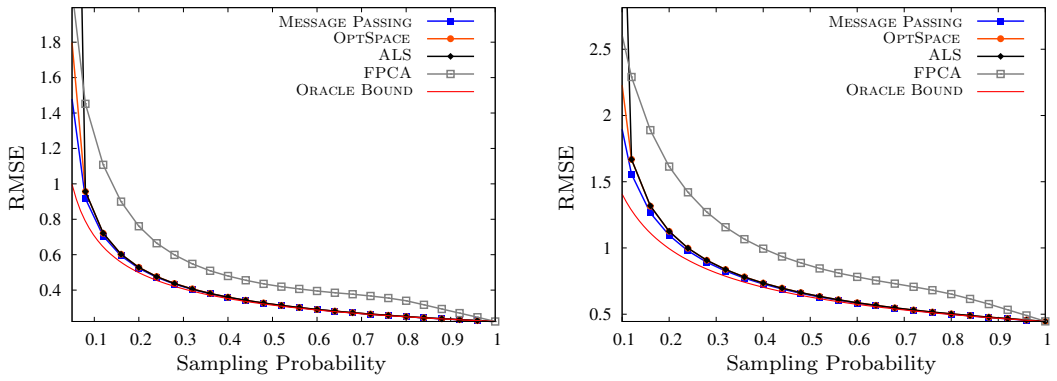

Figure 6.4: RMSE as a function of the sampling probability for $n = 1000$ and ranks $r = 10$ (left) and $r = 20$ (right)

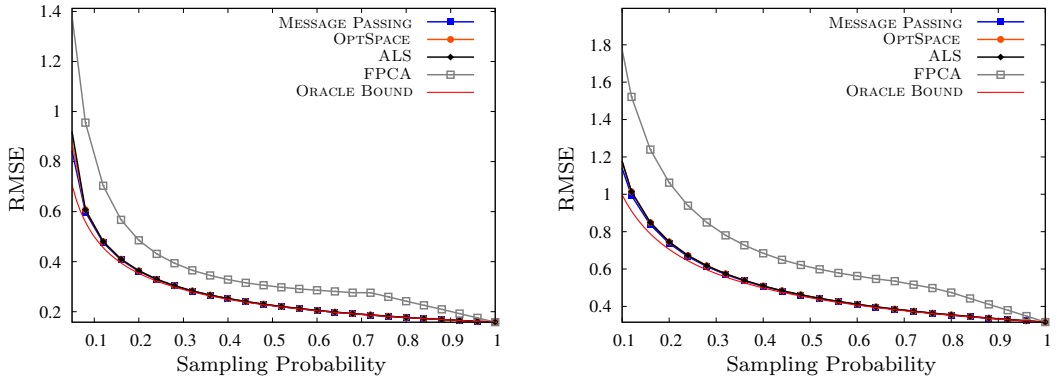

Figure 6.5: RMSE as a function of the sampling probability for $n = 2000$ and ranks $r = 10$ (left) and $r = 20$ (right)

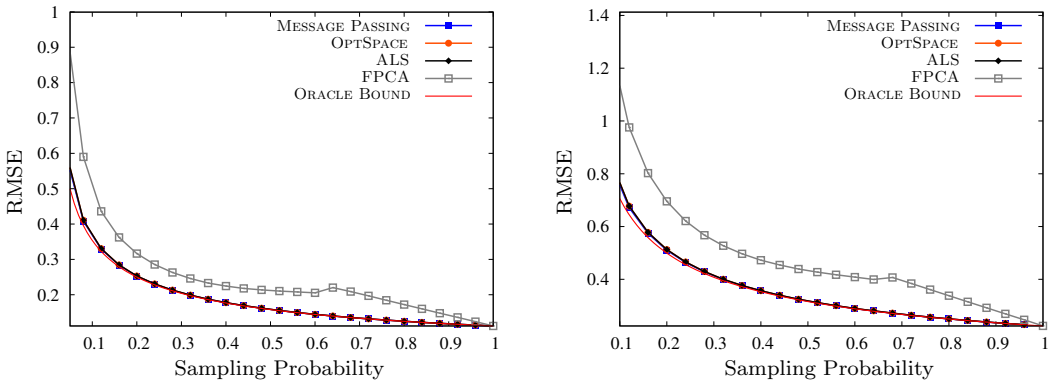

Figure 6.6: RMSE as a function of the sampling probability for $n = 4000$ and ranks $r = 10$ (left) and $r = 20$ (right)

the regularized cost function

$$\begin{aligned} \mathcal{L}_E(X, Y) &\equiv \left\| \mathcal{P}_E(XY^T - N) \right\|_F^2 + \lambda \|X\|_F^2 + \lambda \|Y\|_F^2 \\ &= \mathcal{R}_E(X, Y) + \lambda \|X\|_F^2 + \lambda \|Y\|_F^2 \end{aligned}$$

with the value of λ obtained by cross validation.

In Figure 6.7, we plot the RMSE as a function of the rank used by the algorithms, say $\widehat{r}$. The dimensions of the matrix $n, m = 100$, the rank used to generate the matrix was $r = 10$ and the noise ratio $\|W\|_F / \|M\|_F$ is 1.0. Figures 6.8 and 6.9 show the corresponding results for matrices with ranks 6 and 5 and noise ratios 1.0 and 0.1 respectively.

These examples have been taken from [75]. We compare our results with the results obtained by the SOFT-IMPUTE+ algorithm of [75] and SVT [19]. Note that since SVT does not use rank information, we results are constant across used rank.

These figures demonstrate the effect of estimating incorrectly the rank r of M. Notice that, for $\widehat{r} = \min(m, n)$, the cost function $\mathcal{L}_E(X, Y)$, cf. Eq. (6.25) is equivalent to the convex objective function (6.13), cf. [101]. A few qualitative features of these figures are of interest. First, the reconstruction error is minimum when the rank is estimated correctly $\widehat{r} = r$. Second, the error degrades monotonically as $\widehat{r}$ gets

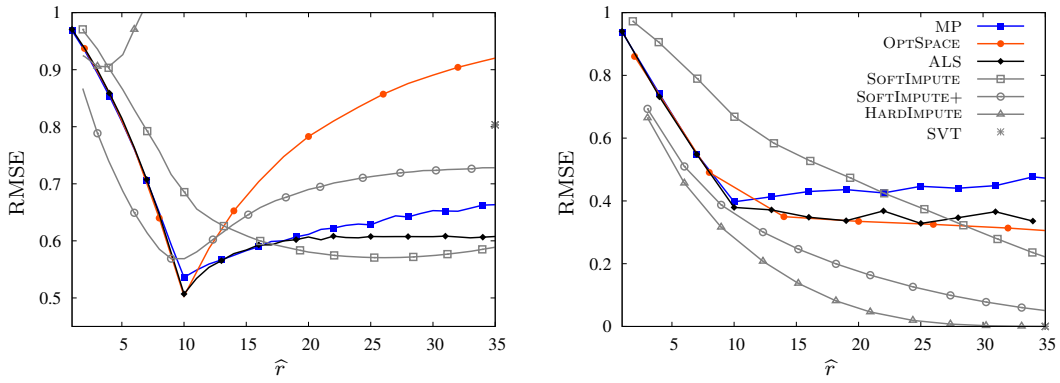

Figure 6.7: Test (left) and Training (right) RMSE as a function of reconstruction rank for the different algorithms for $n = 100$, $r = 10$, $|E| = 5000$

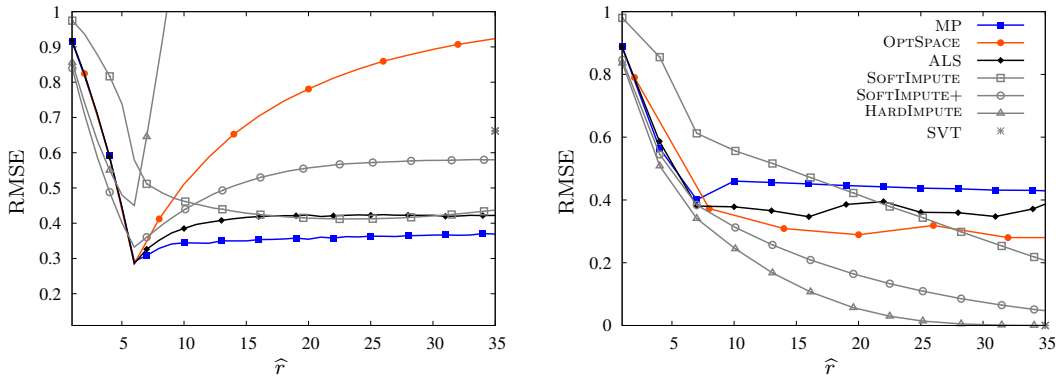

Figure 6.8: Test (left) and Training (right) RMSE as a function of reconstruction rank for the different algorithms for $n = 100$, $r = 6$, $|E| = 5000$

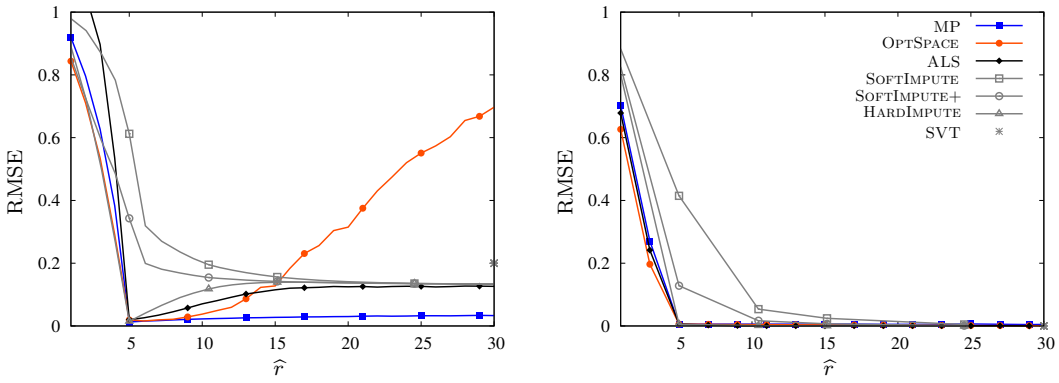

Figure 6.9: Test (left) and Training (right) RMSE as a function of the reconstruction rank for the different algorithms for $n = 100$, $r = 5$, $|E| = 2000$

larger. The ability to choose $\widehat{r}$ allows to trade between computational complexity (presumably the optimization problem becomes simpler for large $\widehat{r}$ and in particular is convex for $\widehat{r} = \min(m, n)$) and statistical accuracy. Third, the degradation in statistical accuracy is graceful: overestimating r only implies a modest increase in error. This is due to the regularization term in $\mathcal{L}_E(X, Y)$.

6.3.3 Real Datasets

The Netflix dataset

The Netflix dataset was released as part of the Netflix Challenge [3] competition announced in 2006. The training dataset consists of about 100 million ratings from about 500,000 users (anonymized) and about 17,000 movies. An additional 1 million (user, movie) pairs with the corresponding ratings were provided as a probe set (disjoint from the training data) to be used for offline testing of algorithms. The ratings were all integers from 1 to 5. The algorithms were evaluated based on the Root Mean Squared Error (RMSE) on a test set. The target for the competition was 0.8563, a 10% improved over Netflix's own algorithm which achieved an RMSE of 0.9514.

We preprocessed the data by subtracting from each rating, the mean rating for the corresponding movie. Further, we normalized the entries such that squared sum of

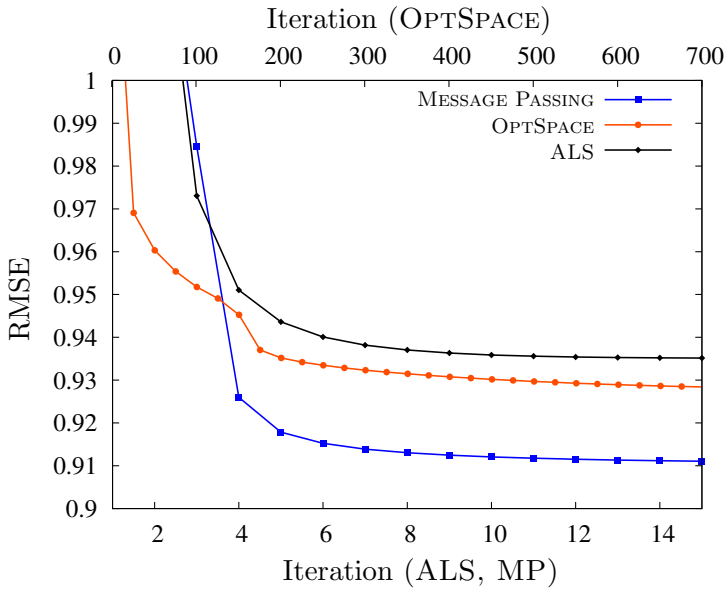

Figure 6.10: Netflix Dataset : RMSE on the probe set as a function of number of iterations. The lower x-axis corresponds to the number of iterations of Alternating Least Squares and Message Passing and the upper x-axis corresponds to the number of iterations of OptSpace. Final RMSE : 0.9105. $n = 480183$, $m = 17770$, $|E| = 99072112$, $r = 30$. The Message Passing algorithm converges within 15 iterations and achieves an RMSE of 0.9105.

ratings for any movie was equal to the number of ratings delivered for that movie. As is customary [14], we model the ratings thus normalized as the –partially observed– matrix N. The rating given by user i to movie j is therefore $N_{ij} = M_{ij} + W_{ij} = u_i^T v_j + W_{ij}$ where $u_i, v_j \in \mathbb{R}^r$ are latent feature vectors for the user and the movie. In Figure 6.10 we plot the RMSE achieved on the probe set as a function of the number of iterations for Alternate Least Squares, Naive Alternate Least Squares, and Message Passing. The message passing algorithm converges more quickly than the other algorithms, and essentially achieves its best accuracy after only 15 iterations.

The final RMSE achieved by this simple matrix factorization approach is: 0.9105. We think that the present techniques can be generalized to richer factorization models as well [64].

The Jester Jokes dataset

We applied the same approach as above for the Jester Jokes dataset. This is a collection of about 4.1 million ratings by about 73,000 users [1]. These ratings for jokes were collected between April 1999 and May 2003. It is popular to use the Mean Absolute Error (MAE) to measure the quality of predictions with this dataset. For a given test set T of (user, joke) pairs, let M_{ij} be the actual rating and $\widehat{M}_{ij}$ be the predicted rating. Then the MAE is defined as

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

and the Normalized Mean Absolute Error (NMAE) is defined as $\text{NMAE} = \text{MAE}/(M_{\max} - M_{\min})$. The ratings for this dataset are in $[-10, 10]$. The Eigentaste algorithm [48] achieves an NMAE of 0.187. There have been a number of improvements to the original algorithm (see [81]) and [66] contains a comparison of the performance of several different algorithms.

Query time – the time taken to obtain a single prediction from a trained model – is an important parameter for collaborative filtering algorithms since it corresponds to the delay experienced by the users. Some of the algorithms listed in [66] have a constant *query time* while others have a *query time* of $O(n)$ where n is the number of users. Message Passing, Alternating Least Squares and OptSpace all have $O(1)$ query time. We note that the performance of the message passing algorithm in terms of the NMAE (plotted in Figure 6.11 as a function of iteration) is significantly better than the best $O(1)$ *query time* algorithm in [66] and comparable to the best $O(n)$ *query time* algorithm.

Finally, we repeat the experiment by varying the number of users in the dataset. For each user, we remove 2 ratings to form the test set [61]. We also compare our performance with those of FPCA and ADMiRA. We report these finding in Table 6.1.

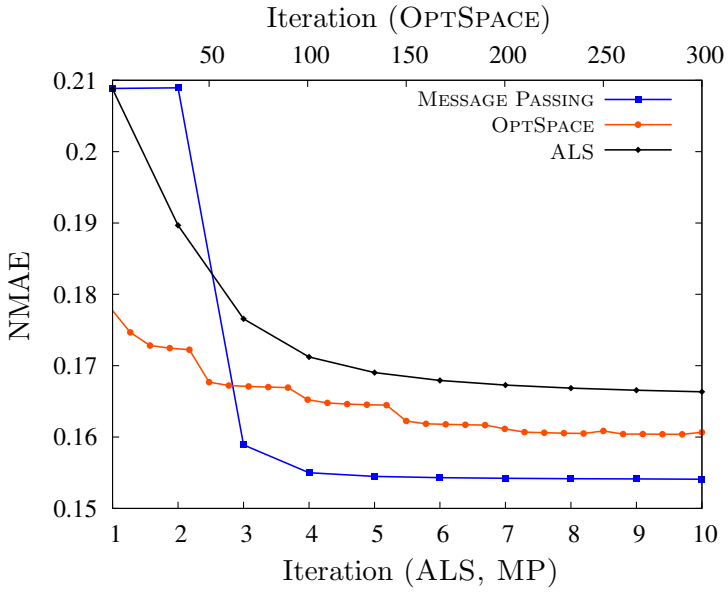

Figure 6.11: Jester Dataset : NMAE on the test set as a function of the number of iterations. The lower x-axis corresponds to the number of iterations of Alternating Least Squares and Message Passing and the upper x-axis corresponds to the number of iterations of OptSpace. Final NMAE : 0.153. $n = 73421$, $m = 100$, $|E| = 3.8 \times 10^6$, $r = 40$.

Dataset	m	n	$\lvert E \rvert$	r	Message Passing	OptSpace	ALS	FPCA	ADMiRA
Netflix	408183	17770	99072112	30	0.9105	0.9280	0.9348	–	–
Jester Jokes	100	100	7484	2	.1808	.1782	.1799	0.2039	0.1819
	1000	100	73626	9	.1571	.1609	.1621	.1611	.1619
	2000	100	146700	9	.1554	.1566	.1562	.1610	.1629
	4000	100	290473	9	.1568	.1580	.1582	.1629	.1632
Movielens	943	1682	80000	10	.1713	.1776	.1749	0.1902	.2428

Table 6.1: A comparison of the performance of the different algorithms on real datasets. Some of the data has been taken from [61]. We could not run some of the algorithms on the Netflix dataset. This is indicated by a "–". To conform with existing literature, we use the RMSE metric for the Netflix dataset and the NMAE metric for the Jester Jokes and the Movielens datasets. We see that Message Passing consistently outperforms all the other algorithms.

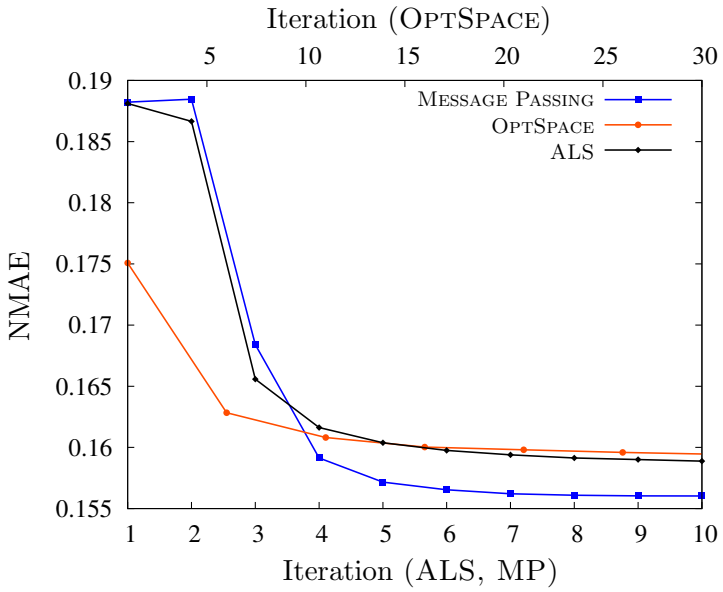

Figure 6.12: Movielens Dataset : NMAE on the test set as a function of the number of iterations. The lower x-axis corresponds to the number of iterations of Alternating Least Squares and Message Passing and the upper x-axis corresponds to the number of iterations of OptSpace. Final NMAE : 0.1556. $n = 6040$, $m = 3952$, $|E| = 0.9 \times 10^6$, $r = 25$.

The Movielens dataset

The Movielens dataset [2] is a collection of 1 million ratings for 6000 users by 4000 movies. The ratings are integers between 1 and 5. Again, we use the NMAE as the performance metric. See [25, 89] and references therein for details about the dataset and algorithms. In [89], the authors report a best NMAE of 0.1662. The best NMAE for this dataset reported in the literature is 0.1596 [25]. In Figure 6.12, we plot the NMAE achieved by message passing for this dataset. The final NMAE is 0.1556.

For purposes of comparison, we also run the experiment on a smaller dataset consisting of 943 users, 1682 movies and 100000 ratings. Here, we compare the performance of Alternating Least Squares and Message Passing with that obtained by FPCA and ADMiRA. Table 6.1 provides a summary of our findings. We find that Message Passing consistently out performs the algorithms on all the datasets that were involved in the experiments.

Bibliography

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

[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):9, 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] L. Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific J. Math.*, 16(1):1–3, 1966.

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

[8] Y. Azar, A. Fiat, A. Karlin, F. McSherry, and J. Saia. Spectral analysis of data. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 619–626, New York, NY, USA, 2001. ACM.

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

[10] Z. D. Bai, B. Q. Miao, and G. M. Pan. On asymptotics of eigenvectors of large sample covariance matrices. *Ann. of Probab.*, 35:1532–1572, 2007.

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

[12] 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.

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

[14] M. W. Berry, Z. Drmać, and E. R. Jessup. Matrices, vector spaces, and information retrieval. *SIAM Review*, 41(2):335–362, 1999.

[15] M. W. Berry and D. I. Martin. Parallel svd for scalable information retrieval. *Proc. of the Intl. Workshop on Parallel matrix algorithms and applications*, 2000.

[16] 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.

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

[18] B. Bollobás. *Random Graphs*. Cambridge University Press, January 2001.

[19] 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.

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

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

[22] 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.

[23] E. J. Candès and T. Tao. The Dantzig selector : statistical estimation when p is much larger than n. *Annals of Statistics*, 35, 2007.

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

[25] L. Candillier, F. Meyer, and M. Boullè. Comparing state-of-the-art collaborative filtering systems. *Lecture Notes in Computer Science*, 4571:548–562, 2007.

[26] 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.

[27] 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.

[28] M. Cucuringu, Y. Lipman, and A. Singer. Sensor network localization by eigenvector synchronization over the euclidean group. 2010. `https://web.math.princeton.edu/ amits/publications/sensors_final.pdf`.

[29] W. Dai and O. Milenkovic. Set: an algorithm for consistent matrix completion. arXiv:0909.2705, 2009.

[30] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[31] S. Deerwester, S. T. Dumias, 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.

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

[33] 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.

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

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] P. Erdös and A. Rènyi. On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, pages 17–61, 1960.

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

[41] M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the American Control Conference*, volume 6, pages 4734–4739, 2001.

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

[43] W. Feller. *An Introduction to Probability Theory and its Applications*, volume 1. John Wiley & Sons Inc, New York, 1968.

[44] 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.

[45] 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.

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

[47] A. Gelman and J. Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, 2006. Chapter 25.

[48] 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.

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

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

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

[52] T. Hastie, R. Tibshirani, and J. Frideman. *The Elements of Statistical Learning*. Springer, 2003.

[53] S. Isaacman, S. Ioannidis, A. Chaintreau, and M. Martonosi. Distributed rating prediction in user generated content streams. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, pages 69–76, New York, NY, USA, 2011. ACM.

[54] D. Karger, S. Oh, and D. Shah. Iterative learning for reliable crowdsourcing systems. 2011.

[55] R. H. Keshavan, V. Mirrokni, and M. Thakur. Large-scale message-passing-based matrix factorization applied to link prediction. *in preparation*, 2012.

[56] R. H. Keshavan and A. Montanari. Regularization for matrix completion. *Proc. of the IEEE Int. Symposium on Inform. Theory*, 2010.

[57] R. H. Keshavan and A. Montanari. A rigorous analysis of matrix completion via alternating least squares. 2012.

[58] 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.

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

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

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

[62] V. Koltchinskii, K. Lounici, and A. B. Tsybakov. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *Ann. Statist.*, 39(5):2302–2329, 2011.

[63] 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.

[64] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.

[65] K. Lee and Y. Bresler. Admira: Atomic decomposition for minimum rank approximation. arXiv:0905.0044, 2009.

[66] D. Lemire. Scale and translation invariant collaborative filtering systems. *Information Retrieval*, 8(1):129–150, 2005.

[67] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-dened clusters. Oct 2008. arXiv:0810.1355.

[68] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007.

[69] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7:76–80, 2003.

[70] R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. Wiley-Interscience, 2002.

[71] 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.

[72] A. Talwalkar M. Mohri. On the estimation of coherence. arXiv:1009.0861, 2010.

[73] S. Ma, D. Goldfarb, and L. Chen. Fixed point and Bregman iterative methods for matrix rank minimization. arXiv:0905.1643, 2009.

[74] G. Malewicz, M. H. Austern, A. J. C. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski. Pregel : A system for large-scale graph processing. *SIGMOD*, June 2010.

[75] 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.

[76] R. Meka, P. Jain, and I. S. Dhillon. Guaranteed rank minimization via singular value projection. arXiv:0909.5457, 2009.

[77] R. Meka, P. Jain, and I. S. Dhillon. Matrix completion from power-law distributed samples. In *Advances in Neural Information Processing Systems*, 2009.

[78] M. D. Mitzenmacher. *The power of two choices in randomized load balancing*. PhD thesis, University of California, Berkeley, 1996.

[79] C. C. Moallemi. A message-passing paradigm for optimization. 2007.

[80] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1):17–32, 1981.

[81] T. Nathanson, E. Bitton, and K. Goldberg. Eigentaste 5.0: Constant-time adaptability in a recommender system using item clustering. *RecSys*, pages 149–152, 2007.

[82] S. Negahban and M. J. Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. arXiv:1009.2118, 2010.

[83] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2001.

[84] S. Oh, A. Karbasi, and A. Montanari. Sensor network localization from local connectivity: Performance analysis for the mds-map algorithm. *Information Theory Workshop*, 2010.

[85] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. (1999-66), November 1999. Previous number = SIDL-WP-1999-0120.

[86] S. Rangan and A. Fletcher. Iterative estimation of constrained rank-one matrices in noise. arXiv:1202.2759, 2012.

[87] B. Recht. A simpler approach to matrix completion. arXiv:0910.0651, 2009.

[88] B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum rank solutions of matrix equations via nuclear norm minimization. arXiv:0706.4138, 2007.

[89] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *ICML*, 2005.

[90] 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.

[91] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

[92] 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.

[93] R. Salakhutdinov and N. Srebro. Collaborative Filtering in a Non-Uniform World: Learning with the Weighted Trace Norm. arXiv:1002.2780, February 2010.

[94] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Itembased collaborative filtering recommendation algorithms. *Proc. of the 10th International Conference onthe World Wide Web*, pages 285–295, 2001.

[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] J. W. Silverstein and Z. D. Bai. On the empirical distribution of eigenvalues of a class of large-dimentional random matrices. *J. Multivariate Anal.*, 54:175–192, 1995.

[97] A. Singer. A remark on global positioning from local distances. *Proceedings ofthe 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] N. Srebro. *Learning with Matrix Factorizations*. PhD thesis, MIT, 2004. http://ttic.uchicago.edu/∼nati/Publications/thesis.pdf.

[100] N. Srebro and T. Jaakkola. Weighted low-rank approximations. In *20th International Conference on Machine Learning*, pages 720–727. AAAI Press, 2003.

[101] 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.

[102] G. W. Stewart. On the early history of the singular value decomposition. 1992.

[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] 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.

[105] 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.

[106] 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.

[107] J. A. Tropp. User-friendly tail bounds for sums of random matrices. `arXiv:1004.4389`, 2010.

[108] 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*, 17(6):520–525, 2001.

[109] 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.

[110] 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.

[111] 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.

[112] K. Y. Yeung and W. L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, 2001.

[113] 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.

[114] 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.

[115] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu. Spatio-temporal compressive sensing and internet traffic matrices. *SIGCOMM Comput. Commun. Rev.*, 39:267–278, 2009.