Algorithms for Matrix Completion

by

Yu Xin

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2012

Author .....
Department of Electrical Engineering and Computer Science
September 8, 2011

Certified by .....
Tommi S. Jaakkola
Professor
Thesis Supervisor

Accepted by .....
Professor Leslie A. Kolodziejski
Chair of the Committee on Graduate Students

Algorithms for Matrix Completion

by

Yu Xin

Submitted to the Department of Electrical Engineering and Computer Science
on September 8, 2011, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

We consider collaborative filtering methods for matrix completion. A typical approach is to find a low rank matrix that matches the observed ratings. However, the corresponding problem has local optima. In this thesis, we study two approaches to remedy this issue: reference vector method and trace norm regularization. The reference vector method explicitly constructs user and item features based on similarities to reference sets of users and items. Then the learning task reduces to a convex regression problem for which the global optimum can be obtained. Second, we develop and analyze a new algorithm for the trace-norm regularization approach. To facilitate smooth primal optimization, we introduce a soft variational trace-norm and analyze a class of alternating optimization algorithms. We introduce a scalable primal-dual block coordinate descent algorithm for large sparse matrix completion. The algorithm explicitly maintains a sparse dual and the corresponding low rank primal solution at the same time. Preliminary empirical results illustrate both the scalability and the accuracy of the algorithm.

Thesis Supervisor: Tommi S. Jaakkola
Title: Professor

Acknowledgments

I wish to express my deepest gratitude to my advisor Prof. Tommi Jaakkola. The thesis would not have been possible without his extraordinary help and endless patience. I'd also like to thank my family for supporting me spiritually throughout my life. Finally, I thank Yuezhi for her love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recommender systems (RS) suggest items to users. They are widely used by many content providers such as Amazon and Netflix. The idea is that each user's history reflects their personal interests, so it's possible to predict their future behavior based on that. User's history may include ratings, clicks, purchases and so on. A typical setup involves three entities: users, items and ratings. Rating is a value that represents how much user likes a particular item and may be given as number of stars or as binary indicator. User histories are recorded as triplets where each triplet contains a user ID, an item ID and a rating value. Only some of the possible triplets are observed. The task is to predict the missing ratings baed on user IDs and item IDs and can be seen as a matrix completion problem. The goal is to complete the sparse user-item matrix by filling the missing ratings.

Recommender Systems can be divided into content based methods and collaborative filtering (CF) methods. Content based methods assume that descriptions of items are given in addition to ratings. The descriptions are usually turned into feature vectors as in ordinal regression models [35]. Given item feature vectors and associated user ratings, the regression model can be learned for each user separately. One advantage of this approach is that prediction can be made for a new item without ratings. However, performance strongly depends on the quality of the feature vectors. In contrast, CF does not rely on item descriptions. It assumes instead that users with similar histories have similar preferences. Starting from the rating matrix, the

method aims to complete the matrix by adding constraints on the possible matrices. For example, a common constraint is low rank. Then the corresponding problem is to find a complete low rank matrix that matches the observed ratings. Different from the content based method, the predications for all users and items will be learned jointly. Our focus is on CF since it has been successfully used in many systems ([1], [4], [5]).

There are three main issues with CF methods scalability, sparsity and robustness. In RS, the number of users and items can easily be in the thousands, and ratings in millions. A CF method must be able to handle such large matrices. A careful design is necessary so the algorithms scale well to large problems. For instance, evaluating all the similarities between users has complexity is $O(n^2d)$ where n is the number of users and d is the number of common ratings. This is not possible to do for large matrices.

Only a small portion of all ratings are observed. For instance, in Netflix and Movielens dataset, less than 5% ratings are available. This raises a critical over-fitting problem, i.e. only a few parameters can be estimated reliably from few ratings. One way to solve this problem is by adding constraints or regularization. Examples include low-rank constraint and trace norm regularization. By choosing an appropriate regularization parameter, trace norm regularization will guarantee a low rank matrix solution (cf. section 4).

An interesting question is how many ratings are needed before accurate prediction can be made. "Cold start" problem refers to the case when only a few or even no ratings are provided for a user or an item. This is common since new users join frequently. To address this problem, extra data such as item description and user demographic are necessary. A challenge is to incorporate these data into CF methods.

The robustness is also an issue. Adversarial users can trick RS by providing fake ratings for their own interest. For instance, they can provide ratings as normal users while giving high or low ratings to a subset of items. Detecting such adversarial behavior is hard since it can hardly be distinguished from actual preferences. If the number of adversarial users is small compared to all users, performance may not be

degraded significantly. However, in some web-based systems, single user may register multiple accounts. When the number of adversarial accounts is comparable to normal accounts, predictions can be manipulated.

In this thesis, we focus on CF methods that address above three issues. One successful method is low-rank matrix factorization. As we discussed above, low-rank assumption is the key to efficient updates and to avoid over-fitting. The method solves a non-convex optimization problem by alternatingly optimizing user and item feature vectors. However, since objective function is not convex, the algorithm can easily be trapped at a local minimum. While a local minimum may be good enough for prediction, accuracy is not guaranteed.

In chapter 3, we propose a new CF method based on reference vectors. We start by selecting a set of reference users and items that represent distinctive types. Then we express users' feature vectors as similarities to reference users, and similarly for items. Given these feature vectors, predictions are made by solving a convex optimization problem. The alternating optimization algorithm can still be used in this method. But unlike low-rank matrix factorization, the reference method has a convex objective function, so the algorithm will converge to a global minimum.

One of our main contribution is an efficient algorithm for trace norm regularization. Trace norm regularization favors low rank solutions. It is a convex relaxation of the low rank constraint [6]. Previous approaches like proximal gradient algorithms require computing Singular Vector Decomposition (SVD) of rating matrix in each iteration. This is very expensive for large problems. In Chapter 4, we first study an alternating primal algorithm based on a variational form of trace norm. This algorithm is faster than accelerated proximal gradient algorithm, but still requires SVD in each iteration. Next we study the dual problem. We propose an efficient primal-dual algorithm that doesn't require SVD. Preliminary empirical results illustrate both the scalability and the accuracy of this algorithm.

We expand on the dual problem by adding constraints on the dual variables. For instance, previous research on robust CF considers rating matrix as a sum of low rank and sparse matrices (e.g. [18]). We find that the method is equivalent to adding

a simple constraint on each dual variable. Other possible extensions include group sparsity and item categories.

Chapter 2

Background

Given a partially observed matrix $R \in \mathbb{R}^{m \times n}$, the collaborative filtering problem is to predict the missing entries in R. Here each column $i \in \{1, 2, ...n\}$ represents an item, each row $u \in \{1, 2, ..., m\}$ represents a user, and entry $R_{u,i}$ represents the rating of user u of item i. In most CF problems, R is large and sparse, thus highlighting efficiency and over-fitting issues. For instance, Netflix training dataset contains $100,480,507$ ratings that $480,189$ users gave to $17,770$ movies. Only 1.2% ratings are observed. Movielens 10M dataset has 10M ratings for 10,000 movies by 72,000 users where only 1.4% ratings are observed.

2.1 Low Rank Matrix Factorization

One of the most widely used methods is low-rank matrix factorization. Let $P \in \mathcal{R}^{d \times m}$ and $Q \in \mathcal{R}^{d \times n}$ be feature matrices for users and items. Column vectors P_u and Q_i represent user u's preferences and item i's properties respectively. Given P and Q, the rating matrix is predicted as $\tilde{R} = P^T Q$. Intuitively, only a few factors affects user interests so the dimension d could be small. To avoid over-fitting, we need to reduce the number of free parameters. The number of free parameters is proportional to the dimension, so d should be small.

The feature matrices P and Q can be obtained by solving the following optimiza-

tion problem

$$\min_{P,Q} \sum_{(u,i)\in\Omega} \text{Loss}(R_{u,i}, P_u^T Q_i) + \lambda(\sum_u \|P_u\|_2^2 + \sum_i \|Q_i\|_2^2) \tag{2.1}$$

where $\text{Loss}(\cdot,\cdot)$ is a loss function over observed ratings and $\lambda(\sum_u \|P_u\|_2^2 + \sum_i \|Q_i\|_2^2)$ is regularization. The choice of loss function depends on the problem itself. For numerical ratings, L is usually the squared loss: $\text{Loss}(R_{u,i}, P_u^T Q_i) = (R_{u,i} - P_u^T Q_i)^2$ [4]. For binary classification problem, the loss function can be hinge loss $\text{Loss}(R_{u,i}, P_u^T Q_i) = \max(0, 1 - R_{u,i} P_u^T Q_i)$ [5]. Regularization is very important. Empirical results show that the method doesn't perform well without regularization as it will overfit quickly as d increases.

When one of P and Q is fixed, the objective function is convex. But it's not jointly convex w.r.t. P and Q. An algorithm for solving the optimization problem can easily be trapped in a local minimum. For some CF problems, local minima are good enough. But prediction accuracy may change when algorithms converge to different local minima.

To sovle the optimization problem, we can use alternating optimization algorithm that sequentially updates one of P or Q while fixing the other,

$$P_u = (\sum_{i:(u,i)\in\Omega} Q_i Q_i^T + \lambda I)^{-1} (\sum_{i:(u,i)\in\Omega} R_{u,i} Q_i), \quad u = 1, 2, ..., m \tag{2.2}$$

$$Q_i = (\sum_{u:(u,i)\in\Omega} P_u P_u^T + \lambda I)^{-1} (\sum_{i:(u,i)\in\Omega} R_{u,i} P_u), \quad i = 1, 2, ..., n \tag{2.3}$$

The above update rules make use of sparsity because the computations only involve observed ratings. The algorithm stops when P and Q change little in one iteration.

For extremely large data, stochastic gradient descent (SGD) algorithm is a better option. A single update of the corresponding columns of P and Q are made in response to each observed rating in the order of their appearance: given $(u, i, R_{u,i})$,

we update only

$$P_u = P_u - \alpha[(P_u^T Q_i - R_{u,i})Q_i + \lambda P_u) \quad (2.4)$$

$$Q_i = Q_i - \alpha[(P_u^T Q_i - R_{u,i})P_u + \lambda Q_i) \quad (2.5)$$

where α is step size. The value of α is critical in SGD. In practice, α is usually manually set by empirical analysis. The algorithm can be easily adapted to online learning scenario when new ratings are continuously observed.

2.2 Trace Norm Regularization

Trace norm is a 1-norm penalty on singular values of the matrix, i.e. $||W||_* = \sum_{k=1}^{n} \sigma_k$ where σ_k is W's k^{th} singular value. It naturally leads to low-rank solutions with sufficient regularization (see section 4). In CF, the optimization problem with trace norm regularization is

$$\min_W \sum_{(u,i)\in\Omega} \text{Loss}(R_{u,i}, W_{u,i}) + \lambda||W||_* = \min_W L(W) + \lambda||W||_* \quad (2.6)$$

where $\text{Loss}(\cdot,\cdot)$ is a loss function, $L(W)$ is the aggregate loss over observed ratings and W is prediction of rating matrix. Trace norm regularization is necessary for generalization to missing ratings. It is closely related to low-rank matrix factorization since

$$||W||_* = \min_{P^T Q = W} \frac{1}{2}(||P||_F^2 + ||Q||_F^2) \quad (2.7)$$

But the objective function is now convex.

2.2.1 Algorithms

One key difficulty with this method is that $||W||_*$ is not differentiable. A number of approaches have been proposed to deal with this problem. Proximal gradient

approach is one of them [3]. The approach first approximates loss function via Taylor expansion, then update W by solving the following optimization problem

$$W^{r+1} = \text{argmin}_W(L(W^r) + \nabla L(W^r)(W - W^r) + \frac{1}{2\tau}\|W - W^r\|_F^2 + \lambda\|W\|_*) \quad (2.8)$$

where τ controls the step size. There is a closed form solution to this problem which consists of two steps:

$$X^r = W^r - \tau\nabla_W L(W) \quad (2.9)$$

$$W^{r+1} = S_{\tau\lambda}(X^r) \quad (2.10)$$

where $S_{\tau\lambda}$ is a shrinkage operator. Suppose SVD of X^r is $X^r = U\sigma(X^r)V^T$, then

$$S_{\tau\lambda}(X^r) = UDV^T \quad (2.11)$$

where D is a diagonal matrix with entries $D_{i,i} = \max(0, \sigma_i(X) - \tau\lambda)$. The algorithm is guaranteed to converge when τ is small enough [3]. One drawback of this approach is that it requires SVD in each iteration. The complexity of SVD is $O(n^3)$ which is infeasible for large scale problems.

Trace norm can also be cast as a constrained optimization problem [6]. Actually, $\|W\|_* \leq t$ iff there exist A and B such that

$$\begin{bmatrix} A & X \\ X^T & B \end{bmatrix} \succeq 0 \ \textit{and} \ \text{tr}(A) + \text{tr}(B) \leq 2t \quad (2.12)$$

With this new formulation, the primal optimization problem can be rewritten as

$$\min_W \ L(W) + \frac{\lambda}{2}(\text{tr}(A) + \text{tr}(B)) \quad s.t. \begin{bmatrix} A & X \\ X^T & B \end{bmatrix} \succeq 0 \quad (2.13)$$

The optimization problem is a semi-definite program (SDP). Exactly solving large SDPs is still difficult, but we can use approximation algorithms [34].

2.2.2 Spectral Regularization

The idea of using trace norm to obtain low rank solution can be generalized to spectral regularization [10]. Suppose user profiles and item profiles are elements in Hilbert space $\mathcal{U}$ and $\mathcal{I}$. Consider a linear preference function $f(\cdot,\cdot)$:

$$f(u,i) = \langle \mathbf{u}, F\ \mathbf{i} \rangle \tag{2.14}$$

where F is a compact operator that maps from $\mathcal{U}$ to $\mathcal{I}$. The empirical loss of F is defined as

$$R_N(F) = \frac{1}{|\Omega|} \sum_{(u,i)\in\Omega} \text{Loss}(\langle \mathbf{u}, F\ \mathbf{i} \rangle, R_{u,i}) \tag{2.15}$$

where $\text{Loss}(\cdot,\cdot)$ is a loss function. Consider the case where $\mathbf{u}$ is simply an indicator vector for user u, and $\mathbf{i}$ is an indicator vector for item i. Then $\langle \mathbf{u}, F\ \mathbf{i} \rangle = F_{u,i}$. Spectral regularization can be described as a function that only depends on F's singular values

$$\Psi(F) = \sum_{i=1}^{d} s_i(\sigma_i(F)) \tag{2.16}$$

where s_i is a non-decreasing penalty function satisfying $s_i(0) = 0$ and $\sigma_i(F)$ is F's singular value. Let $\mathcal{U}_N$ and $\mathcal{I}_N$ be the linear span of $\mathbf{u}_k, k = 1, ..., N$ and $\mathbf{i}_k, k = 1, ..., N$. There exists a matrix $\alpha \in \mathbb{R}^{m\times n}$ such that,

$$F = \sum_{a}^{m} \sum_{b}^{n} \alpha_{a,b} \mathbf{p}_a \mathbf{q}_b \tag{2.17}$$

where $(\mathbf{p}_1, ...\mathbf{p_m})$ and $(\mathbf{q}_1, ..., \mathbf{q}_n)$ form orthogonal bases of $\mathcal{U}_N$ and $\mathcal{I}_N$. The singular values of F are the same as matrix α's singular values. With the empirical loss and spectral regularization, the estimation problem is

$$F_* \ = \ \text{argmin}_{F\in B(\mathcal{U},\mathcal{I})}\ R_N(F) + \lambda\Psi(F) \tag{2.18}$$

where $B(\mathcal{U}, \mathcal{I})$ is the set of compact operators from $\mathcal{U}$ to $\mathcal{I}$. An interesting class of spectral regularization is $s_i(\sigma_i(F)) = \sigma_i(F)^\beta$ where $\beta = 0, 1, 2$ specifies rank, trace norm, or Frobenius norm regularization, respectively.

2.3 Bayesian Approach

In a probabilistic version of matrix factorization, user and item feature vectors are drawn from some distribution, and ratings are assumed to be conditionally independent given feature vectors. The conditional distributions are given by [20]:

$$P(R|U, V, \alpha) = \prod_{(u,i)\Omega} \mathcal{N}(R_{u,i}|P_u^T Q_i, \alpha^{-1} I) \tag{2.19}$$

$$P(P|\alpha_U) = \prod_i \mathcal{N}(P_u|0, \alpha_u^{-1} I) \tag{2.20}$$

$$P(Q|\alpha_I) = \prod_j \mathcal{N}(Q_i|0, \alpha_i^{-1} I) \tag{2.21}$$

Inferring posterior distribution with above assumption is analytically intractable due to its complexity. Instead, predictions are usually made by finding the MAP estimate. One drawback is that the values α, α_u and α_i are essential for the approach to generalize well, but they can only be easily tuned manually, for instance by searching. However, searching for the optimal values is computationally expensive for large problems. Another way to deal with this issue is to employ a Bayesian approach where we introduce priors for hyper-parameters and integrate over them to automatically control the generalization capacity [20]. Though exact inference is still intractable in this case, approximate inference can be performed by a MCMC algorithm. Empirical results show that this approach performs better than probabilistic matrix factorization.

Chapter 3

Reference method

One drawback of basic low rank matrix factorization is that the objective function is not convex, so learning algorithms can be trapped in a local minimum. One way to deal with this issue is by explicitly specifying one of the feature matrices for users and items, and learning the other. In this case, the objective function is convex and we can obtain the global minimum. The method is unable to explore the whole low rank space however since one feature matrix is fixed. But it may still outperform the local minimum from matrix factorization if the features are carefully designed.

In this chapter, we propose a new method for collaborative filtering. The method first selects a set of reference users and items, then specifies user and item features based on similarities to reference users and items. We consider two prediction models. One fixes the user feature matrix and learns the unknown item feature matrix. The other operates in the reverse order. The two models are then combined linearly. The unknown user and item feature matrices can be obtained by solving a convex optimization problem with an alternating optimization algorithm.

3.1 Selection criteria

We start from user features. Let $\mathcal{K}$ denote the set of reference users. Its size d is chosen as $d \ll \min(m, n)$. A user is described as $\bar{R}_{u,\cdot}$ where $\bar{R}$ are ratings after taking out user and item means, and $\bar{R}_{u,i} = 0$ if the rating is missing. The distance between

two users is the Euclidean distance between their ratings $d(u,v) = \|\bar{R}_{u,\cdot} - \bar{R}_{v,\cdot}\|^2$. We use a greedy packing algorithm to select the reference set. The first reference user is selected as $u_0 = \text{argmax}_u \|\bar{R}_{u,\cdot}\|$. Then the algorithm iteratively select one reference user by solving the following optimization problem:

$$u = \text{argmax}_u \text{min}_{v \in \mathcal{K}} d(u,v) \tag{3.1}$$

The algorithm selects a user that is furthest from current reference users. Actually, the reference users are vertices of the convex hull that contains all users. On the other hand, since missing ratings are considered as 0, the algorithm is inclined to choose users with more ratings, such that the ratings of all reference users are able to cover most of items. Another reasonable selection criteria is to select the cluster centers in user vector space as reference users.

3.2 Similarity metric

Since each reference user has distinctive preferences, the user's similarity to reference user reveals his/her personal interests. We express the user features as similarities to the reference set which gives us feature matrix A

$$A(u,v) = S(\bar{R}_{u,\cdot}, \bar{R}_{\mathcal{K}(v),\cdot}) \tag{3.2}$$

where S is a similarity function and $\mathcal{K}(v)$ is the v^{th} reference user. The similarity function is not necessarily symmetric and might not correspond to any kernel function. In a neighborhood method, similarities between two users are used directly as weights in prediction. To make prediction for all users at the same time, the neighborhood method needs to compute the similarities between any pair of users which is very expensive. Our method is different since the similarities are only used to parameterize users, and only the similarities to reference users are needed.

A few similarity metrics have been used in collaborative filtering literature. Ex-

amples include Cosine similarity and Pearson similarity :

$$S_c(R_{u,\cdot}, R_{v,\cdot}) = \frac{R_{u,\cdot} \cdot R_{v,\cdot}}{\|R_{u,\cdot}\| \cdot \|R_{v,\cdot}\|} \tag{3.3}$$

$$S_p(R_{u,\cdot}, R_{v,\cdot}) = \frac{\sum_{j \in N}(R_{u,j} - \bar{R}_u)(R_{v,j} - \bar{R}_v)}{\sqrt{\sum_{j \in N}(R_{u,j} - \bar{R}_u)^2}\sqrt{\sum_{j \in N}(R_{v,j} - \bar{R}_v)^2}} \tag{3.4}$$

where $\bar{R}_u$ is the average rating of user. These similarity metrics are based on Euclidean distance between users' rating vectors.

In RS, relative order is more important than the rating. We consider a weighted ranking function [7] as similarity metric. Suppose the scale of rating is 5, then the similarity between $R_{u,\cdot}$ and $R_{v,\cdot}$ is defined as:

$$S_r(R_{u,\cdot}, R_{v,\cdot}) = \frac{\sum_{(a,b) \in T_{u,v}} \delta(R_{v,a} > R_{v,b})\delta(R_{u,a} < R_{u,b})}{\sum_{(a,b) \in T_{u,v}} \delta(R_{u,a} \neq R_{u,b})} \tag{3.5}$$

where $T_{u,v} = \{a, b | (u,a), (u,b), (v,a), (v,b) \in \Omega\}$ are shared ratings of two users and $\delta(x) = 1$ if $x > 0$ and 0 otherwise. The similarity metric compares all pairs of observed ratings and counts the number of different rating orders. It is more robust than Pearson and Cosine similarity, but at the same time, it is more expensive to evaluate. Let $\bar{l}$ be the average number of ratings two users share. The time to compute weighted ranking similarity is $O(\bar{l}^2)$,while Cosine and Pearson similarity can be evaluated in $O(\bar{l})$. In CF, $\bar{l}$ may be small because of sparsity, so the difference in computation may be insignificant.

The above similarity metrics only depend on the observed ratings shared by both users. Generally, the more ratings two users share, the more confident is their similarity value. In the previous section, we mentioned the selection of reference set tends to select users with more ratings. This is one of its motivations.

3.3 Regression model

Following the same procedure, we can generate a reference set and feature matrix B for items. Given A and B, we consider a regression formulation

$$\min_{P,Q} \sum_{(u,i)\in\Omega} (R_{u,i} - P_u^T B_i - A_u^T Q_i)^2 + \lambda_1 \|P\|^2 + \lambda_2 \|Q\|^2 \tag{3.6}$$

where $P \in \mathbb{R}^{d\times m}$ and $Q \in \mathbb{R}^{d\times m}$ are parameters for users and items. $P_u^T B_i$ and $A_u^T Q_i$ can be seen as two linear regression models where P and Q are learnt jointly. Different from low rank matrix factorization method, the objective is convex with respect to P and Q, so we are able to find the global minimum. The problem can be solved by alternating optimization algorithm with closed form update,

$$P_u = (\sum_{i:(u,i)\in\Omega} B_i B_i^T + \lambda_1 I)^{-1} (\sum_{i:(u,i)\in\Omega} (R_{u,i} - A_u^T Q_i) B_i) \tag{3.7}$$

$$Q_i = (\sum_{u:(u,i)\in\Omega} A_u A_u^T + \lambda_2 I)^{-1} (\sum_{u:(u,i)\in\Omega} (R_{u,i} - P_u^T B_i) A_u) \tag{3.8}$$

Selecting the size of reference set is tricky since it involves a trade-off between expressiveness and avoiding over-fitting. Alternatively, we can start from a large value of d, and use trace norm regularization to encourage a low rank solution,

$$\min_{P,Q} \sum_{(u,i)\in\Omega} (R_{u,i} - P_u^T B_i - A_u^T Q_i)^2 + \lambda_1 \|P\|_* + \lambda_2 \|Q\|_* \tag{3.9}$$

The above formulation does not consider the interaction between A and B. We can add an additional term to take this into account,

$$\min_{P,Q,W} \sum_{(u,i)\in\Omega} (R_{u,i} - P_u^T B_i - A_u^T Q_i - A_u^T W B_i)^2 + \lambda_1 \|P\|^2 + \lambda_2 \|Q\|^2 + \lambda_3 \|W\|^2 \tag{3.10}$$

where $W \in \mathbb{R}^{d\times d}$ characterize how user features and item features are related in prediction.

3.4 Greedy selection of reference set

The performance of reference method highly depends on the feature matrices A and B. One issue about previous procedure is that A and B are pre-fixed before learning process, but ideally the construction of A and B should take learning results as feedback to improve performance.

We start with formulation 3.6. Define residual ratings as $R^r_{ui} = R_{ui} - P_u^T B_i - A_u^T Q_i$ for $(u, i) \in \Omega$. The residual error at iteration r over a validation set S is $e^r = (\frac{1}{|S|}(\sum_{(u,i)\in S}(R^r_{u,i})^2)^{\frac{1}{2}}$. The following algorithm iteratively selects a user and an item with the largest norm of residual ratings into the reference set:

0. Select $u_0 = \text{argmax}_u \sum_{(u,i)\in\Omega} R^2_{u,i}$ and $i_0 = \text{argmax}_i \sum_{(u,i)\in\Omega} R^2_{u,i}$ into reference set Re^u and Re^i. Starting from $r = 1$ and $e^0 = \inf$, repeat $1 - 4$ until $e^{r-1} - e^r \leq \delta$.

1. Update A and B by adding a column of similarities to new reference user and item.

2. Solve optimization problem 3.6 to obtain residual R^r,

3. Add $v = \text{argmax}_u \sum_{(u,i)\in\Omega}(R^r_{u,i})^2$ and $j = \text{argmax}_i \sum_{(u,i)\in\Omega}(R^r_{u,i})^2$ into reference set Re^u and Re^i respectively,

4. Evaluate e^r and let $r = r + 1$.

In step 3, reference user and item are selected based on residual ratings. Users and items with large norm of residual ratings mean that the current prediction model does not work well for these users and items. By adding them into reference set, the method will put more weight on them, and it will likely improve overall performance. One thing to notice is that the selection criteria is not always good since it might select outlier users and items with many ratings.

To avoid over-fitting, step 4 uses a validation set to estimate current performance. The algorithm terminates when adding a new reference user and item gives small improvement.

Step 2 is the most computationally intensive part. For optimization problem 3.6, there isn't a closed form solution. But since we only add one column to A and B, the new optimum may still be close to the old ones. Then the alternating

optimization algorithm used for solving the optimization problem may converge after a few iterations.

In the above algorithm, A and B expand one column every iteration. All columns of P and Q are updated in step 2. An alternative approach is to fix P and Q learned in previous iteration and only update the new columns, thinking $P_i^T B_i$ and $A_i^T Q_i$ as 1-dimensional linear models. The approach is similar to boosting. It also has a computational advantage since the optimization problem 3.6 reduces to simple linear regression.

3.5 Contextual information

Some Recommendation Systems may collect contextual information besides ratings. This information can be user demographic data, item category, expert opinion or possibly a short description of each item. Incorporating contextual information can help improve accuracy of predictions. They can also be used to solve cold start problem when only a few ratings are provided for a particular user or item.

The contextual information can be represented as additional feature vectors for users and items. Let C_u denote the user features. We can first estimate the contextual parameters without any CF signal, fix the parameters and estimate CF.

$$\min_{D,T} \sum_{(u,i)\in\Omega} (R_{u,i} - C_u^T D_i)^2 + \lambda_3 \|D\|^2 + \lambda_2 \|T\|^2 \tag{3.11}$$

$$\min_{P,Q,\mu} \sum_{(u,i)\in\Omega} (R_{u,i} - P_u^T B_i - A_u^T Q_i - \mu C_u^T D_i)^2 + \lambda_1 \|P\|^2 + \lambda_2 \|Q\|^2 \tag{3.12}$$

where μ is a constant depending on the contextual information's reliability. For a new user with no previous ratings, $A_u = P_u = 0$. But if the user feature C_u is provided, we can still predict the user's rating as $C_u^T D_i$. Similarly, we can also incorporate contextual information of items.

The contextual information might not be available for all users and items. In this case, we impute 0 for those missing features. One interesting problem is to infer contextual information when they are missing. For instance, a missing C_u can be

inferred as

$$\hat{C_u} = \text{argmin}_{C_u} \sum_{(u,i)\in\Omega} (R_{u,i} - P_u^T B_i - A_u^T Q_i - \mu C_u^T D_i - \mu S_i^T T_u)^2 \tag{3.13}$$

3.6 Online update

In RS, data are collected persistently. New data may change users's interest or item's popularity. To reflect these changes, the prediction model must be updated. Exactly solving the optimization problem is inefficient, but an approximation can be quickly obtained.

We will focus on the formulation 3.6. Suppose the objective function has reached the optimum with current ratings. When a new rating $R_{u,i}$ comes, the derivative of objective function is:

$$\frac{\partial f}{\partial P_u} = 2(P_u^T B_i + A_u^T Q_i - R_{u,i})B_i \tag{3.14}$$

$$\frac{\partial f}{\partial Q_i} = 2(P_u^T B_i + A_u^T Q_i - R_{u,i})A_u \tag{3.15}$$

A simple update rule modifies P_u and Q_i with step size μ,

$$P_u = P_u - \mu \frac{\partial f}{\partial P_u} \tag{3.16}$$

$$Q_i = Q_i - \mu \frac{\partial f}{\partial Q_i} \tag{3.17}$$

The above update is based on a linear approximation of the objective function. We can use the second order method to make more accurate approximation. Since our objective function is relatively simple, we can compute the Hessian matrix of P_u and Q_i,

$$C = \begin{bmatrix} \sum_{j,(u,j)\in\Omega} B_j B_j^T + \lambda I & B_i A_u^T \\ A_u B_i^T & \sum_{v,(v,i)\in\Omega} A_v A_v^T + \lambda I \end{bmatrix} \tag{3.18}$$

A stronger Newton-Raphson update rule is:

$$\begin{bmatrix} P_u \\ B_i \end{bmatrix} = \begin{bmatrix} P_u \\ B_i \end{bmatrix} - C^{-1} \begin{bmatrix} \frac{\partial f}{\partial P_u} \\ \frac{\partial f}{\partial Q_i} \end{bmatrix} \tag{3.19}$$

One advantage of this update is that it doesn't need a step size μ, which is usually estimated empirically.

3.7 Active learning

Ratings in recommender systems are expensive to collect. We'd like to reduce the number of necessary ratings for making predictions. A technique that can achieve this goal is active learning. Instead of assuming all the ratings are given in the beginning, active learning interactively collects new ratings by making queries to users.

The formulation 3.6 can be seen as a probabilistic model,

$$P(R_{u,i}|P_u, Q_i) = \mathcal{N}(0, \sigma_0^2) \tag{3.20}$$

$$P(P_u) = \mathcal{N}(0, \sigma_u^2) \tag{3.21}$$

$$P(Q_i) = \mathcal{N}(0, \sigma_i^2) \tag{3.22}$$

where σ_0, σ_u and σ_i are variances that satisfy $\sigma_0^2/\sigma_u^2 = \lambda_1$ and $\sigma_0^2/\sigma_u^2 = \lambda_2$. One common strategy is to query unknown rating $R_{u,j}$ such that after incorporating $R_{u,j}$ the variance of $P(P_u|R, Q)$ is reduced most,

$$j = \text{argmin}_{k,(u,k)\notin\Omega} \det[(\sum_{i,(u,i)\in\Omega} B_i B_i^T + B_k B_k^T + \lambda_1 I)^{-1}] \tag{3.23}$$

$$= \text{argmax}_{k,(u,k)\notin\Omega} B_k^T (\sum_{i,(u,i)\in\Omega} B_i B_i^T + \lambda_1 I)^{-1} B_k \tag{3.24}$$

However, it does not take into account that P and Q are correlated. At the optimum,

$$Q_i^* = (\sum_{(u,i)\in\Omega} A_u A_u^T + \lambda I)^{-1} (\sum_{(u,i)\in\Omega} A_u^T (R_{u,i} - P_u B_i)) \tag{3.25}$$

A better strategy is to consider conditional distribution $P(P_u|Q_i^*(P_u), R)$ instead. $P(P_u|Q_i^*(P_u), R)$ follows Gaussian distribution, and the variance of P_u before new query is given by

$$\Sigma_u = (\sum_{i,(u,i)\in\Omega} [(1 - A_u^T(\sum_{v,(v,i)\in\Omega} A_v A_v^T + \lambda_2 I)^{-1} A_u) B_i B_i^T] + \lambda_1 I)^{-1} \tag{3.26}$$

After querying unknown rating $R_{u,k}$, the variance become

$$\Sigma_u^k = (\sum_{i,(u,i)\in\Omega} [(1 - A_u^T(\sum_{v,(v,i)\in\Omega} A_v A_v^T + \lambda_2 I)^{-1} A_u) B_i B_i^T] \tag{3.27}$$

$$+ (1 - A_u^T(\sum_{v,(v,k)\in\Omega} A_v A_v^T + \lambda_2 I)^{-1} A_u) B_k B_k^T + \lambda_1 I)^{-1} \tag{3.28}$$

The corresponding active learning strategy is to query item j that reduces the variance most which is equivalent to

$$j = \text{argmax}_{k,(u,k)\notin\Omega}[(1 - A_u^T(\sum_{v,(v,i)\in\Omega} A_v A_v^T + \lambda_2 I)^{-1} A_u) B_k \Sigma_u B_k^T]$$

The term $(1 - A_u^T(\sum_{v,(v,i)\in\Omega} A_v A_v^T + \lambda_2 I)^{-1} A_u)$ serves as weight for item k.

Chapter 4

Efficient algorithm for optimization with trace norm regularization

In this chapter, we consider algorithms for sparse matrix completion problem [26]. The goal is to predict the missing entries of a $n \times m$ ($n \geq m$) real valued target matrix R based on a small subset of observed entries. We analyze an alternating primal algorithm based on a variational form of trace norm. The algorithm has a closed form update and is guaranteed to converge at a linear rate. One shortcoming of the alternating primal algorithm and many proximal gradient methods (e.g. [29]) is that they are infeasible for large scale problems since they require SVD in each iteration. We then introduce a primal-dual algorithm that explicitly exploits sparse rating matrices and has an efficient updates.

4.1 Trace norm regularization for matrix completion

The predicted matrix W can be constrained to have low rank via trace norm regularization (e.g., [27, 6])

$$F(W) = \sum_{(u,i)\in\Omega} \text{Loss}(R_{u,i}, W_{u,i}) + \lambda \|W\|_* = L(W) + \lambda \|W\|_* \tag{4.1}$$

where $\text{Loss}(R_{u,i}, W_{u,i})$ is a convex loss function of $W_{u,i}$ such as the squared loss. $L(W)$ denotes the aggregate loss function across the observed entries. The trace-norm $\|W\|_*$ is a 1-norm penalty on the singular values of the matrix, i.e., $\|W\|_* = \sum_{j=1}^{m} \sigma_j(W)$ where $\sigma_j(W) \geq 0$ is the j^{th} singular value of W. The trace norm is the dual norm of spectral norm, i.e. $\|W\|_* = \sup_Y(\text{tr}(W^T Y) | \|Y\| \leq 1)$ where $\|Y\| = \max_j(\sigma_j(Y))$. The convexity of trace norm follows from the point-wise max of linear function. For large enough λ, some of the singular values are set exactly to zero, so the predicted matrix W has a low rank.

One key optimization challenge is that though $\|W\|_*$ is convex, it's not differentiable. We seek to remedy this by casting $\|W\|_*$ as a minimization problem over weighted Frobenius norms (see also [27, 23, 31])

$$\|W\|_* = \inf_{A>0} \left\{ \text{tr}(A^{-1} W^T W) + \frac{1}{4}\text{tr}(A) \right\} \tag{4.2}$$

where the A is a symmetric positive definite $m \times m$ matrix. To see this, consider the singular value decomposition $W = U\text{diag}(\sigma(W))V^T$, where $\sigma(W)$ is the vector of singular values. By differentiating the variational form with respect to A, and setting it to zero, we get

$$A = 2(W^T W)^{\frac{1}{2}} = 2V\text{diag}(\sigma(W))^{\frac{1}{2}} V^T \tag{4.3}$$

Plugging this A back into the objective function yields $\|W\|_* = \sum_i \sigma_i(W)$ as desired. In [31], the variational form was used to obtain a closed form solution for multi-task learning with trace norm regularization. However, the solution only exists when the predicted matrix W has full rank..

Note that by placing additional constraints on the choice of A, we can obtain different regularizers. For example, by constraining A to be diagonal in addition to positive definite, we obtain a group Lasso penalty over the rows of W: $\sum_u \|W_u\|_2 = \sum_u \min_{A_u}(\|W_u\|_2^2/A_u + A_u/4)$ where W_u denotes the u^{th} row. As another example, consider index sets $I_1, \ldots, I_k$ that form a k-partition of the rows $\{1, \ldots, n\}$. Forcing A to be block diagonal relative to this partition would yield a structured trace-norm

over the blocks: $\sum_{l=1}^{k} \|W_{I_l}\|_*$ where W_{I_l} is the sub-matrix of rows in partition I_l. Other constraints are possible as well.

We use the variational form to facilitate optimization by solving the infimum over A together with W. In other words, we would optimize

$$J_0(W, A) = L(W) + \lambda \left\{ \mathrm{tr}(A^{-1}W^TW) + \frac{1}{4}\mathrm{tr}(A) \right\} \tag{4.4}$$

with respect to A and W. The mapping $(A, W) \to WA^{-1}W^T$ is operator convex ([25], Proposition 8.5.25), so $J_0(W, A)$ is jointly convex in W and A. However, the infimum is not attained (A^{-1} diverges) unless W^TW has full rank. As a result, by setting A in response to W in the course of optimization, we may prematurely exclude certain dimensions of W^TW. To remedy this and to ensure that the regularizer remains strictly convex in W and A, we introduce the *soft trace norm*

$$\|W\|_s = \min_{A>0} \left\{ \mathrm{tr}(A^{-1}W^TW) + \epsilon^2\mathrm{tr}(A^{-1}) + \frac{1}{4}\mathrm{tr}(A) \right\} = \sum_{i=1}^{m} \sqrt{\sigma_i^2(W) + \epsilon^2} \tag{4.5}$$

The minimum over A is always attained at $A = 2(\epsilon^2 I + W^TW)^{\frac{1}{2}}$ and remains positive definite regardless of W. The extended objective $J_\epsilon(W, A)$ is thus more appropriately defined in terms of the variational soft trace-norm:

$$J_\epsilon(W, A) = L(W) + \lambda \left\{ \mathrm{tr}(A^{-1}W^TW) + \epsilon^2\mathrm{tr}(A^{-1}) + \frac{1}{4}\mathrm{tr}(A) \right\} \tag{4.6}$$

Comparing to $J_0(W, A)$, $J_\epsilon(W, A)$ has an additional term $\lambda\epsilon^2\mathrm{tr}(A^{-1})$. It can be bounded as $0 \leq \lambda\epsilon^2\mathrm{tr}(A^{-1}) \leq \lambda n\epsilon$ when $A = 2(\epsilon^2 I + W^TW)^{\frac{1}{2}}$ for some W. $J_\epsilon(W, A)$ is also more amenable to alternating optimization as discussed below.

4.2 Alternating primal algorithms

4.2.1 General convergence proof for alternating optimization algorithm

Let $F(W)$ be the objective to be minimized with respect to matrix W and let $J(W, A)$ denote an auxiliary function such that $F(W) = \min_{A \in \mathcal{A}} J(W, A)$. We consider primal optimization algorithms that can be written in the following alternating form:

$$A^r \in \arg\min_{A \in \mathcal{A}} J(W^r, A), \quad W^{r+1} \in \arg\min_{W \in \mathcal{W}} J(W, A^r) \tag{4.7}$$

Both $\mathcal{A}$ and $\mathcal{W}$ are assumed to be compact sets and the auxiliary function is assumed to be bounded from below. The respective minima can be attained but need not be unique in general.

We begin by characterizing such algorithms' convergence rate under specific assumptions about $F(W)$ and the auxiliary function $J(W, A)$. Notationally, $d_W F(W)$ indicates the sub-differential of $F(W)$ at W, and $\partial_W F(W) \in d_W F(W)$ is a (sub-)gradient.

Theorem 4.2.1. *Let $\mathcal{W}$ and $\mathcal{A}$ be compact sets and define $F(W) = \min_{A \in \mathcal{A}} J(W, A)$. We assume that $F(W)$ is bounded from below within $\mathcal{W}$. Let $W^* \in arg\min_{W \in \mathcal{W}} F(W)$. In addition,*

1) $\partial_W J(W, A) \in d_W F(W)$ for any $A \in arg\min_{A \in \mathcal{A}} J(W, A)$

2) $\partial_W J(W, A)$ is Lipschitz continuous with constant L for a fixed A.

Then, if $F(W)$ is strongly convex with constant k, the alternating algorithm attains $F(W^r) - F(W^*) \leq \delta$ in $O(\log(\frac{1}{\delta}))$ iterations. If $F(W)$ is only convex, $F(W^r) - F(W^*) \leq \delta$ in $O(\frac{1}{\delta})$ iterations. When $F(W)$ is not convex, the alternating optimization algorithm still converges in $O(\frac{1}{\delta^2})$ iterations to the solution set $\{W : \|\partial_W F(W))\|_F \leq \delta\}$.

For completeness, we prove the statements in Appendix A.1. The assumption that the respective sets $\mathcal{A}$ and $\mathcal{W}$ are compact is not strong for monotone algorithms. For

example, typically the set $\mathcal{W} = \{W : F(W) \leq F(W_0)\}$ is compact for some initial W_0 . Moreover, the monotone updates would necessarily keep W within such a set. Note that the assumption 2) pertains to the (sub-)gradient of $J(W, A)$, not the function itself.

We consider the function $J_\epsilon(W, A)$ defined earlier for the soft trace norm. Let $F_\epsilon(W) = \min_A J_\epsilon(W, A) = J_\epsilon(W, A^*) = L(W) + \|W\|_s$ where $A^* = 2(W^TW + \epsilon^2 I)^{\frac{1}{2}}$. Since $L(W)$ is convex and $\|W\|_s$ is strongly convex (lemma A.2), $F_\epsilon(W)$ is strongly convex and $\lim_{\|W\|\to\infty} F_\epsilon(W) = \infty$. W is therefore bounded in a compact set $\mathcal{W} = \{W : F_\epsilon(W) \leq F(W_0)\}$ for some initial W_0.

Lemma 4.2.2. *Soft trace norm $\|W\|_s$ is strongly convex in $\{W : \|W\|_F \leq C\}$ such that $\langle \partial\|W_1\|_s - \partial\|W_2\|_s, W_1 - W_2\rangle \geq \frac{\epsilon^2}{(C^2+\epsilon^2)^{3/2}}\|W_1 - W_2\|_F^2$. (for proof, see Appendix A.2)*

The first assumption in theorem 4.2.1 is satisfied since $\partial_W J_\epsilon(W, A^*) = d_W L(W) + 2\lambda W(A^*)^{-1} = d_W L(W) + \lambda W(W^TW + \epsilon^2 I)^{-\frac{1}{2}} = d_W L(W) + \lambda d_W \|W\|_s \in d_W F_\epsilon(W)$. The partial derivative contains one term from loss function and one linear term from soft trace norm regularization. Assuming the aggregate loss function $L(W)$ has a Lipschitz continuous derivative, and since A is bounded, $\partial_W J_\epsilon(W, A^*)$ is Lipschitz continuos. Therefore the two assumptions in Theorem 4.2.1 are satisfied by $J_\epsilon(W, A)$.

Note that the convergence guarantees in the theorem are relative to the function $F_\epsilon(W) = \min_A J_\epsilon(W, A)$, not the original trace-norm regularization problem. However, it can be shown that $F_\epsilon(W)$ is $C\epsilon$ close to the actual regularization objective $F(W)$. Since $F_\epsilon(W)$ is strongly convex, the algorithm converges in $O(\log(\frac{1}{\delta}))$ steps.

The problem of $J_0(W, A)$ without the relaxation of soft trace norm is that $A^* = \text{argmin}_A J_0(W, A)$ is not necessarily achieved in the positive definite matrix set $\mathcal{A} = \{A|A > 0\}$.

4.2.2 Closed form update

The alternating optimization algorithm for 4.6 has a closed form update for the squared loss. Let $\mathcal{I}_u = \{i : (u, i) \in \Omega\}$ be the index set of observed elements for a row

(user) u, and Φ_u be the index vector such that $\Phi_{ui} = 1$ iff $i \in \mathcal{I}_u$. By differentiating $J_\epsilon(W, A)$ with respect to W, and setting it to zero, we get

$$W_u = R_u(\lambda A^{-1} + diag(\Phi_u))^{-1} \tag{4.8}$$

Reordering rows and columns in A, the above equation can be simplified as

$$W_u = [R_{u,\mathcal{I}_u}, 0] \begin{bmatrix} \lambda I + A_{\mathcal{I}_u\mathcal{I}_u} & 0 \\ A_{\mathcal{I}_u\mathcal{I}_u^c} & \lambda I \end{bmatrix}^{-1} A = R_{u,\mathcal{I}_u}(\lambda I + A_{\mathcal{I}_u\mathcal{I}_u})^{-1} A_{\mathcal{I}_u,\cdot} \tag{4.9}$$

where $\mathcal{I}_u^c$ is the complement set of $\mathcal{I}_u$. Due to sparsity, the size of $\mathcal{I}_u$ is much smaller than the number of items, so the above equation can be computed efficiently.

4.2.3 Proximal gradient method

Another example of an alternating algorithm for minimizing Eq.(4.1) is the proximal gradient method such as [29, 22]

$$W^{r+1} = \arg\min_W \left\{ \|W\|_* + L(W^r) + \langle \partial L(W^r), W - W^r \rangle + \frac{1}{2t}\|W - W^r\|_F^2 \right\} \tag{4.10}$$

For this, we assume that the aggregate loss function $L(W)$ is twice continuously differentiable in addition to having a Lipschitz continuous derivative with constant L. We use the following auxiliary function (note the order of the arguments):

$$J(W, A) = \|A\|_* + L(W) + \langle \partial L(W), A - W \rangle + \frac{1}{2t}\|A - W\|_F^2 \tag{4.11}$$

The proximal step relative to this function is $A^r = \arg\min_A J(W^r, A)$, whereas $W^{r+1} = \arg\min_W J(W, A^r)$ merely sets $W^{r+1} = A^r$ provided that $t < 2/L$. Note that $F(W^r) = \min_A J(W, A)$ is not the value of the regularization problem at W^r but rather the predicted (upper bound) value after the proximal gradient update. Nevertheless, at the optimum $F(W^*)$ has the same value as the regularization objective. At A^r, $\partial_W J(W, A^r) = (I/t - \partial^2 L(W))(W - A)$ is Lipschitz continuos since we

assume $L(W)$ is twice continuously differentiable. The objective function $F(W)$ can be written as

$$\begin{aligned} F(W) &= \min_A(|A\|_* + \frac{1}{2t}\|A - W + t\partial L(W)\|_F^2) + L(W) - \frac{t}{2}(\partial L(W))^2 \\ &= e_t h(W - t\partial L(W)) + L(W) - \frac{t}{2}(\partial L(W))^2 \end{aligned} \tag{4.12}$$

where $e_t h(x) = \min_y(h(y) + (1/2t)\|y - x\|^2)$ is the Moreau envelope of function $h(x) = \|x\|_*$. It is convex and continuously differential with $\nabla e_t h(x) = (1/t)(x - y^*)$ where $y^* = \text{argmin}_y(h(y) + (1/2t)\|y - x\|^2)$ ([36], Theorem 2.26). The derivative of $F(W)$ is

$$\begin{aligned} dF(W) &= \frac{1}{t}(I - t\partial^2 L(W))(\nabla e_t h(W - t\partial L(W))) + \partial L(W) - t\partial^2 L(W)\partial L(W) \\ &= (I/t - \partial^2 L(W))(W - t\partial L(W) - A^r) + \partial L(W) - t\partial^2 L(W)\partial L(W) \\ &= \partial_W J(W, A^r) \end{aligned} \tag{4.13}$$

The two assumptions of Theorem 4.2.1 are therefore satisfied by proximal gradient method.

4.3 A primal-dual algorithm

The primal methods discussed earlier are infeasible for large problems as they update the full rating matrix in each iteration. We consider here dual algorithms that introduce dual variables only for each observed entry, thus leading to a sparse estimation problem. The algorithm works for any strongly convex loss function such as square loss and sigmoid loss. For illustration, the dual maximization problem (derived below) for square loss function is given by

$$tr(Q^T R) - tr(Q^T Q)/2 \quad \text{subject to} \quad Q^T Q \leq \lambda^2 I \tag{4.14}$$

where $Q_{u,i} = 0$ for all unobserved entries $(u, i) \notin \Omega$. Solving the dual problem has three challenges. The first one is the separation problem involving the spectral constraint $Q^T Q \leq \lambda^2 I$. We iteratively generate constraints since only a few of them are relevant. The second challenge is effectively solving the dual under a few spectral constraints for which we derive a new block coordinate descent approach (cf. [33]). The third challenge concerns the problem of reconstructing the primal rating matrix from the dual solution. The algorithm explicitly maintains a sparse dual and corresponding low rank primal solution throughout the optimization.

Derivation of the dual. We begin with a modified primal minimization problem

$$J(W, B) = \sum_{(u,i)\in\Omega} \text{Loss}(R_{u,i} - W_{u,i}) + \frac{1}{2} tr(B^{-1} W^T W) + \frac{\lambda^2}{2} tr(B) \tag{4.15}$$

where $B > 0$ is an $m \times m$ symmetric matrix, defined as $B = A/(2\lambda)$ in terms of the earlier notation. We assume that the loss function is a *strictly convex* function between the observed and predicted entries to ensure that the (dual) solution is unique.

To this end, the dual problem involves Legendre conjugate transformations of the loss functions:

$$\text{Loss}(z) = \max_q \{qz - \text{Loss}^*(q)\} \tag{4.16}$$

where the conjugate $\text{Loss}^*(q)$ is also strictly convex. For example, for the squared loss, $\text{Loss}^*(q) = q^2/2$. The Lagrangian involving both primal and dual variables is given by

$$\mathcal{L}(Q, W, B) = \sum_{(u,i)\in\Omega} \left[Q_{u,i}(R_{u,i} - W_{u,i}) - \text{Loss}^*(Q_{u,i}) \right] + \frac{1}{2} tr(B^{-1} W^T W) + \frac{\lambda^2}{2} tr(B) \tag{4.17}$$

where Q is a sparse matrix of dual variables as we can set $Q_{u,i} = 0$ when $(u, i) \notin D$.

To solve for $W_{v,j}$, we set $d/W_{v,j}\ \mathcal{L}(Q,W,B) = 0$, and get

$$-Q_{v,j} + [WB^{-1}]_{v,j} = 0, \quad (v,j) \in D \tag{4.18}$$

$$[WB^{-1}]_{v,j} = 0, \quad (v,j) \notin D \tag{4.19}$$

or, equivalently, $W = QB$. This is how the primal solution is reconstructed from the dual. Inserting this definition back into the Lagrangian, we obtain

$$\mathcal{L}(Q,B) = \sum_{(u,i)\in\Omega} \Big[Q_{u,i}(R_{u,i} - [QB]_{u,i}) - \text{Loss}^*(Q_{u,i}) \Big] + \frac{1}{2} tr(Q^T QB) + \frac{\lambda^2}{2} tr(\ldots) \tag{4.20}$$

$$= tr(Q^T R) - \sum_{(u,i)\in\Omega} \text{Loss}^*(Q_{u,i}) + \frac{1}{2} tr\big((\lambda^2 I - Q^T Q)B\big) \tag{4.21}$$

The dual objective to be maximized is then

$$\mathcal{L}(Q) = tr(Q^T R) - \sum_{(u,i)\in\Omega} \text{Loss}^*(Q_{u,i}) + \frac{1}{2} \inf_{B\geq 0} tr\big((\lambda^2 I - Q^T Q)B\big) \tag{4.22}$$

Note that it is no longer necessary to require $B > 0$ ($B \geq 0$ suffices). The infimum equals $-\infty$ unless $Q^T Q \leq \lambda^2 I$ implying that the dual can be written in the constrained form

$$\text{maximize } tr(Q^T R) - \sum_{(u,i)\in\Omega} \text{Loss}^*(Q_{u,i}) \text{ subject to } Q^T Q \leq \lambda^2 I \tag{4.23}$$

The matrix B appears as Lagrangian multipliers for the $Q^T Q \leq \lambda^2 I$ constraint. We will solve the dual by adding one constraint at a time in a cutting plane fashion, and also reconstruct B in the process. Only a few constraints may be necessary to approximately enforce $Q^T Q \leq \lambda^2 I$ and to reconstruct B.

The separation problem. The constraint $Q^T Q \leq \lambda^2 I$ is equivalent to $\|Qb\|^2 \leq \lambda^2$ for all b such that $\|b\| = 1$. We will iteratively add constraints represented by b. The separation problem is then: find b such that $\|Qb\|^2 > \lambda^2$ for current Q. It can be easily solved by finding the eigenvector of $Q^T Q$ with the largest eigenvalue. For

example, the power method

$$b = \text{randn}(m, 1). \text{ Iterate } b \leftarrow Q^T Q b, \quad b \leftarrow b/\|b\| \tag{4.24}$$

is particularly effective with sparse matrices. If $\|Qb\|^2 > \lambda^2$ for the resulting b of the power method, then we add a single constraint $\|Qb\|^2 \leq \lambda^2$ into the dual. Note that b does not have to be the eigenvector with largest eigenvalue, any b provides a valid albeit not necessarily the tightest constraint.

Assuming the largest eigenvalue of $Q^T Q b_0 = \psi_0 b_0$ satisfying $\psi_0 > \lambda^2 + \eta$. The initial vector $b = \alpha_0 b_0 + \alpha_1 b_1$ where b_1 is a normalized vector such that $\|Q^T Q b_1\| < \lambda^2$. After k iterations of the power method, $b^k = (Q^T Q)^k b = \alpha_0 \psi_0^k b_0 + (Q^T Q)^k b_1$. We have $\|b^k\| \leq \alpha_0 \psi_0^k + \alpha_1 \lambda^k$ and $\|b^{k+1}\| \geq \alpha_0 \psi_0^{k+1} - \alpha_1 \lambda^{2(k+1)}$. The two inequalities yield that $\|Q^T Q(b^k/\|b^k\|)\| > \lambda^2$ when $k > \log(2\alpha_1 \lambda^2/\alpha_0 \eta)/\log(1+\eta/\lambda^2) = O(\frac{1}{\eta}\log(\frac{1}{\eta}))$. In sum, after $O(\frac{1}{\eta}\log(\frac{1}{\eta}))$ steps, the power method is able to find a violated constraint.

Primal-dual block coordinate descent. The second problem is to solve the dual subject to $\|Qb^l\|^2 \leq \lambda^2$, $l = 1, \ldots, k$, in place of the full set of constraints $Q^T Q \leq \lambda^2 I$. This partially constrained dual problem can be written as

$$tr(Q^T R) - \sum_{(u,i)\in\Omega} \text{Loss}^*(Q_{u,i}) - \sum_{l=1}^{k} h(\|Qb^l\|^2 - \lambda^2) \tag{4.25}$$

where $h(z) = \infty$ if $z > 0$ and $h(z) = 0$ otherwise. Since $\alpha_1 z_1 + \alpha_2 z_2 > 0$ with $\alpha_i > 0$ and $\alpha_1 + \alpha_2 = 1$ stands only if $z_1 > 0$ or $z_2 > 0$, $h(\alpha_1 z_1 + \alpha_2 z_2) \leq \alpha_1 h(z_1) + \alpha_2 h(z_2)$, so $h(z)$ is convex. $h(z)$ is also nondecreasing from its definition. Then the convexity of $h(\|Qb\|^2 - \lambda^2)$ follows from

$$\begin{aligned} h(\|(\alpha_1 Q_1 + \alpha_2 Q_2)b\|^2 - \lambda^2) &\leq h(\alpha_1 \|Q_1 b\|^2 + \alpha_2 \|Q_2 b\|^2 - \lambda^2) && (4.26) \\ &\leq \alpha_1 h(\|Q_1 b\|^2 - \lambda) + \alpha_2 h(\|Q_2 b\|^2 - \lambda) && (4.27) \end{aligned}$$

where $\alpha_1 + \alpha_2 = 1$. The first inequality is from the convexity of $\|Qb\|^2$ and the monotonicity of $h(z)$. The second inequality is from the convexity of $h(z)$. We can

obtain its conjugate dual as

$$h(\|Qb\|^2 - \lambda^2) = \sup_{\xi \geq 0}\{\xi(\|Qb\|^2 - \lambda^2)/2\} = \sup_{\xi \geq 0, v}\{v^T Qb - \|v\|^2/(2\xi) - \xi\lambda^2/2\} \quad (4.28)$$

where the latter form is jointly concave in (ξ, v) where b is assumed fixed. This step lies at the core of our algorithm. By relaxing the supremum over (ξ, v), we obtain a *linear*, not quadratic, function of Q. The new Lagrangian is given by

$$\begin{aligned}
\mathcal{L}(Q, V, \xi) &= tr(Q^T R) - \sum_{(u,i)\in\Omega} \text{Loss}^*(Q_{u,i}) - \sum_{l=1}^{k} \left[(v^l)^T Q b^l - \frac{\|v^l\|^2}{2\xi^l} - \frac{\xi^l \lambda^2}{2} \right] \\
&= tr(Q^T (R - \sum_l v^l (b^l)^T)) - \sum_{(u,i)\in\Omega} \text{Loss}^*(Q_{u,i}) \\
&+ \sum_{l=1}^{k} \left[\frac{\|v^l\|^2}{2\xi^l} + \frac{\xi^l \lambda^2}{2} \right] \qquad (4.29)
\end{aligned}$$

which can be maximized with respect to Q for fixed (ξ^l, v^l), $l = 1, \ldots, k$. Indeed, our primal-dual algorithm seeks to iteratively minimize $\mathcal{L}(V, \xi) = \max_Q \mathcal{L}(Q, V, \xi)$ while explicitly maintaining $Q = Q(V, \xi)$. Note also that by maximizing over Q, we reconstitute the loss terms. The predicted rank k matrix W is obtained explicitly $W = \sum_l v^l (b^l)^T$. By allowing k constraints in the dual, we search over rank k predictions.

The iterative algorithm proceeds by selecting one l, fixing (ξ^j, v^j), $j \neq l$, and optimizing $\mathcal{L}(V, \xi)$ with respect to (ξ^l, v^l). The initial values of V^l and ξ^l are set to 0. The dual objective $\mathcal{L}(V, \xi)$ is monotonically decreasing with respect to V and ξ

$$\mathcal{L}(v_i^1, \xi_i^1, ..., v_{i+1}^l, \xi_{i+1}^l, ..., v_i^k, \xi_i^k) \leq \mathcal{L}(v_i^1, \xi_i^1, ..., v_i^l, \xi_i^l, ..., v_i^k, \xi_i^k) \quad (4.30)$$

where v_i^l and ξ_i^l are values for iteration i. Because of the monotonicity, the algorithm doesn't require a learning rate.

Let $\tilde{W} = \sum_{j \neq l} v^j (b^j)^T$, where only the observed entries need to be evaluated. By minimizing over v^l we get $v^l = \xi^l Q b^l$. The remaining minimization problem over

$\xi^l \geq 0$ is

$$\max_{Q} \left\{ tr(Q^T(R - \tilde{W})) - \sum_{(u,i)\in\Omega} \text{Loss}^*(Q_{u,i}) - \xi^l(\|Qb^l\|^2 - \lambda^2)/2 \right\} \tag{4.31}$$

where we have dropped all the terms that remain constant during the iterative step. Let $Q(\xi^l)$ denote the maximizing Q for a fixed ξ^l. For the squared loss, $Q(\xi^l)$ is obtained in closed form:

$$Q_{u,\mathcal{I}_u}(\xi^l) = (R_{u,\mathcal{I}_u} - \tilde{W}_{u,\mathcal{I}_u}) \left(1 - \frac{\xi^l}{1 + \xi^l \|b^l_{\mathcal{I}_u}\|^2} b^l_{\mathcal{I}_u} (b^l_{\mathcal{I}_u})^T \right), \quad u = 1, \ldots, n \tag{4.32}$$

where $\mathcal{I}_u = \{i : (u,i) \in \Omega\}$ is the index set of observed elements for a row (user) u. In general, an iterative solution is required. The optimal value $\xi^l \geq 0$ is subsequently set as follows. If $\|Q(0)b^l\|^2 \leq \lambda^2$, then $\xi^l = 0$. Otherwise, since $\|Q(\xi^l)b^l\|^2$ is monotonically decreasing as a function of ξ^l, we find (e.g., via bracketing) $\xi^l > 0$ such that $\|Q(\xi^l)b^l\|^2 = \lambda^2$.

Algorithm 1: Primal-Dual Block Coordinate Descent

For $i = 0, 1, 2, ..., k$, update b^i and W according to the following iterations:

Step 1 Compute $Q_{u,i} = R_{u,i} - W_{u,i}$ if $(u,i) \in \Omega$ and 0 otherwise. Find b^i such that $\|Qb^i\|^2 > \lambda^2$ via power method.

Step 2 Update V, ξ, W by function ***Update***(V, ξ, W, i) (4.3)

Function: ***Update***(V, ξ, W, i)

For $l = i, 0, 1, 2, ...i-1, i$,

Step 1 $W = W - \xi^l v^l b^l$.

Step 2 Compute $Q(\xi)$ from equation 4.32. If $\|Q(0)b^l\|^2 \leq \lambda^2$, then $\xi^l = 0$. Otherwise find ξ^* such that $\|Q(\xi^*)b^l\|^2 = \lambda^2$ via bracketing and set $\xi^l = \xi^*$.

Step 3 $W = W + \xi^l v^l b^l$.

4.3.1 Variations of trace norm regularization

Now we analyze some variations of trace norm regularization.

Weighted trace norm. It has been shown that the performance of collaborative filtering with trace norm regularization can be hurt if ratings are non-uniformly sampled ([17]). Weighted trace norm is then proposed to penalize rows and columns with different weights: $\|\text{diag}(\sqrt{n})W\text{diag}(\sqrt{m})\|_* = \|NWM\|_*$ where $n(u)$ and $m(i)$ are marginal probability of observing row u and column i. In practical implementation, $n(u)$ and $m(i)$ are replaced by empirical estimates $\hat{n}(u)$ and $\hat{m}(i)$. Let $\tilde{Y} = NRM$ and $\tilde{W} = NWM$, the Lagrangian involving Q, B, and $\tilde{W}$ is then

$$\mathcal{L}(Q, \tilde{W}, B) = \sum_{(u,i)\in\Omega} \left[\frac{Q_{u,i}}{N_{u,u}M_{i,i}}(\tilde{R}_{u,i} - \tilde{W}_{u,i}) - \text{Loss}^*(Q_{u,i}) \right] + \frac{1}{2}tr(B^{-1}\tilde{W}^T\tilde{W}) + \frac{\lambda^2}{2}tr(B) \quad (4.33)$$

By setting $d/\tilde{W}_{i,j}\mathcal{L}(Q, \tilde{W}, B) = 0$, we get $\tilde{W} = N^{-1}QM^{-1}B$. Inserting this equation into Lagrangian, we obtain the dual objective

$$\mathcal{L}(Q) = tr(Q^T\tilde{R}) - \sum_{(u,i)\in\Omega} \text{Loss}^*(Q_{u,i}) + \frac{1}{2}\inf_{B\geq 0} tr\left((\lambda^2 I - M^{-1}Q^TN^{-2}QM^{-1})B\right) \quad (4.34)$$

The dual can be written in constrained form

$$\text{maximize } tr(Q^T\tilde{R}) - \sum_{(u,i)\in\Omega} \text{Loss}^*(Q_{u,i}) \text{ subject to } M^{-1}Q^TN^{-2}QM^{-1} \leq \lambda^2 I \quad (4.35)$$

The optimization problem can still be solved by primal dual algorithm. Given current Q, find b^l satisfying $\|N^{-1}QM^{-1}b^l\| > \lambda$. Then $Q(\xi^l)$ has a closed form:

$$Q_{u,\mathcal{I}_u}(\xi^l) = (\tilde{R}_{u,\mathcal{I}_u} - \tilde{W}_{u,\mathcal{I}_u})\left(1 - \frac{\xi^l(M^{-1}b^l)_{\mathcal{I}_u}((M^{-1}b^l)_{\mathcal{I}_u})^T}{n(u) + \xi^l\|(M^{-1}b^l)_{\mathcal{I}_u}\|^2}\right), \quad u = 1, \ldots \quad (4.36)$$

ξ^l is selected such that $\|N^{-1}QM^{-1}b^l\| = \lambda$. After obtaining b^l and $v^l = \xi Qb^l$, the primal rating matrix is reconstructed as $W = N^{-1}(\sum_l v^l(b^l)^T)M^{-1}$.

Robust collaborative filtering. When some ratings or rows are corrupted, potentially by an adversary, collaborative filtering might fail to give right predictions.

One idea is to explain rating matrix as a sum of two matrices: $Y = W + S$ where W is a low rank matrix and S is a sparse matrix (e.g. [18]). W represents underlying user's preference, and S contains all the outliers that might harm performance. The estimation problem is

$$F(W) = \sum_{(u,i)\in\Omega} \text{Loss}(R_{u,i}, W_{u,i} + S_{u,i}) + \lambda\|W\|_* = L(W+S) + \lambda\|W\|_* + \mu\|S\|_1 \quad (4.37)$$

Notice that L_1 norm can be cast as a minimization problem: $\|s\|_1 = \min_{t>0}\frac{1}{2}(s^2/t+t)$. We get a modified primal optimization problem

$$\begin{aligned} J(W,B,E) &= \sum_{(u,i)\in\Omega} \text{Loss}(R_{u,i} - W_{u,i} - S_{u,i}) + \frac{1}{2}tr(B^{-1}W^TW) + \frac{\lambda^2}{2}tr(B) \quad (4.38) \\ &+ \sum_{(u,i)\in\Omega} \frac{1}{2}(\frac{S_{u,i}}{E_{u,i}} + \mu^2 E_{u,i}) \quad (4.39) \end{aligned}$$

where $E_{i,j} = \mu T_{i,j}$. Following previous procedure, we introduce dual variables $Q_{u,i}$. Then the Lagrangian is given by

$$\begin{aligned} \mathcal{L}(Q,W,S,B,E) &= \sum_{(u,i)\in\Omega} \Big[Q_{u,i}(R_{u,i} - W_{u,i} - S_{u,i}) - \text{Loss}^*(Q_{u,i})\Big] \quad (4.40) \\ &+ \frac{1}{2}tr(B^{-1}W^TW) + \frac{\lambda^2}{2}tr(B) + \sum_{(u,i)\in\Omega} \frac{1}{2}(\frac{S_{u,i}^2}{E_{u,i}} + \mu^2 E_{u,i}) \quad (4.41) \end{aligned}$$

By setting $d/W_{i,j}\mathcal{L}(Q,W,S,B,E) = 0$ and $d/S_{i,j}\mathcal{L}(Q,W,S,B,E) = 0$, we get $W = QB$ and $S_{u,i} = E_{u,i}Q_{u,i}$. Plugging these results back into the Lagrangian, we obtain

$$\begin{aligned} \mathcal{L}(Q) &= tr(Q^TR) - \sum_{(u,i)\in\Omega} \text{Loss}^*(Q_{u,i}) + \frac{1}{2}\inf_{B\geq 0} tr\big((\lambda^2 I - Q^TQ)B\big) \quad (4.42) \\ &+ \frac{1}{2}\sum_{(u,i)\in\Omega} (\inf_{E_{u,i}\geq 0}((\mu^2 - Q_{u,i}^2)E_{u,i}) \quad (4.43) \end{aligned}$$

The dual can be written in a constrained form:

$$\text{maximize } tr(Q^T\tilde{R}) - \sum_{(u,i)\in\Omega} \text{Loss}^*(Q_{u,i}) \text{ subject to } Q^TQ \leq \lambda^2 I \ , \ Q_{u,i}^2 \leq \mu^2 \quad (4.44)$$

We apply the primal dual algorithm to solve this problem. The objective can be written as

$$tr(Q^T\tilde{R}) - \sum_{(u,i)\in\Omega} \text{Loss}^*(Q_{u,i}) - \sum_{l=1}^{k} h(\|Qb^l\|^2 - \lambda^2) - \sum_{(u,i)\in\Omega} h(Q_{u,i}^2 - \mu^2) \quad (4.45)$$

The conjugate dual of $h(Q_{u,i}^2 - \mu^2)$ is

$$h(Q_{u,i}^2 - \mu^2) = \sup_{\zeta\geq 0}\{\zeta(Q_{u,i}^2 - \mu^2)/2\} = \sup_{\zeta\geq 0, z}\{Q_{u,i}z - z^2/(2\zeta) - \zeta\mu^2/2\} \quad (4.46)$$

The new Lagrangian is given by

$$\begin{aligned}
\mathcal{L}(Q,V,\xi,Z,\zeta) &= tr(Q^TR) - \sum_{(u,i)\in\Omega} \text{Loss}^*(Q_{u,i}) \\
&- \sum_{l=1}^{k} bigg[(v^l)^TQb^l - \frac{\|v^l\|^2}{2\xi^l} - \frac{\xi^l\lambda^2}{2}\Big] - \sum_{(u,i)\in\Omega}\Big[z_{u,i}Q_{u,i} - \frac{z_{u,i}^2}{2\zeta_{u,i}} - \frac{\zeta_{u,i}\mu^2}{2}\Big] \\
&= tr(Q^T(R - \sum_l v^l(b^l)^T) - Z) - \sum_{(u,i)\in\Omega} \text{Loss}^*(Q_{u,i}) \\
&+ \sum_{l=1}^{k}\Big[\frac{\|v^l\|^2}{2\xi^l} + \frac{\xi^l\lambda^2}{2}\Big] + \sum_{(u,i)\in\Omega}\Big[\frac{z_{u,i}^2}{2\zeta_{u,i}} + \frac{\zeta_{u,i}\mu^2}{2}\Big] \qquad (4.47)
\end{aligned}$$

The dual block coordinate algorithm solves the optimization problem 4.48 for v^l and ξ^l where $\bar{W} = \sum_{j\neq l}\xi^l(b^l)^T + Z\cdot\zeta$ and $(Z\cdot\zeta)_{u.i} = Z_{u.i}\zeta_{u.i}$. It then fixes v^l and ξ^l, and optimize $\mathcal{L}(Q,V,\xi,Z,\zeta)$ with respect to Z and ζ. At the minimum of Z, $z_{u,i} = Q_{u,i}\zeta_{u,i}$. The remaining optimization problem over $\zeta \geq 0$ is

$$\max_{Q_{u,i}}\Big\{Q_{u,i}(R_{u,i} - \tilde{W}_{u,i}) - \text{Loss}^*(Q_{u,i}) - \zeta_{u,i}(Q_{u.i}^2 - \mu^2)/2\Big\} \quad (4.48)$$

For square loss, if $(R_{u,i} - \bar{W}_{u,i})^2 \leq \mu^2$, $\zeta_{ui} = 0$. Otherwise, $\zeta_{u.i} = |(R_{u,i} - \bar{W}_{u,i})|/\mu - 1$ and $Q_{u.i} = \mu(R_{u,i} - \bar{W}_{u,i})/|(R_{u,i} - \bar{W}_{u,i})|$.

If the corrupted ratings appear row-wise in the sense that the whole row of ratings are provided by an adversary, we use a different (group lasso) penalty on S. The estimation problem is now:

$$F(W) = \sum_{(u,i)\in\Omega} \text{Loss}(R_{u,i}, W_{u,i} + S_{u,i}) + \lambda\|W\|_* \tag{4.49}$$

$$= L(W + S) + \lambda\|W\|_* + \nu \sum_u \sqrt{\sum_i S_{u,i}^2} \tag{4.50}$$

As the sparsity constraint, group sparsity introduces an additional constraint on dual variables $\sum_i Q_{u,i}^2 \leq \nu^2, u = 1, 2..., n$.

Structured trace norm. The above example motivates a way to introduce structured trace norm regularization. Information such as item category are common in collaborative filtering. They provide a ground truth relationship between users and items.

Suppose item categories $c(i)$ are available, then items within each category are more related to each other than to items from different categories. Current trace norm regularization only considers global relationship between all items while the local relationship within each category is ignored. To incorporate the local relationship, we assume rating matrix is a sum of a few matrices $R = W^g + \sum_l W^l$, where $W^g \in \mathbb{R}^{m\times n}$ is global matrix and $W^l \in \mathbb{R}^{m\times n}$ is local matrix such that $R_{u,i}^l$ is nonzero only if $c(i) = l$. The category information can be introduced by trace norm regularization on both global and local matrices,

$$\min_{W^g,W^l} \sum_{(u,i)\in\Omega} \text{Loss}(R_{u,i}, W_{u,i} + \sum_l W_{u,i}^l) + \lambda_1\|W^g\|_* + \lambda_2 \sum_l \|W^l\|_* \tag{4.51}$$

$$= L(W + S) + \lambda_1\|W^g\|_* + \lambda_2 \sum_l \|W^l\|_* \tag{4.52}$$

Using the same method above, we can derive its dual problem,

$$\text{maximize } tr(Q^T\tilde{R}) - \sum_{(u,i)\in\Omega} \text{Loss}^*(Q_{u,i}) \tag{4.53}$$

$$\text{subject to } Q^TQ \leq \lambda_1^2 I \ , \ Q_l^TQ_l \leq \lambda_2^2 I \ , \ \forall l. \tag{4.54}$$

where Q_l is the projection of Q on items in category l. We can use block coordinate descent algorithm to solve the optimization problem by introducing one constraints on Q^TQ and $Q_l^TQ_l$ iteratively in a cutting plane fashion.

Regularization constants λ_1 and λ_2 control the relative importance of global and local relationship. λ_2 should be less than λ_1 since $Q_l^TQ_l \leq \lambda_1^2 I$ is satisfied as long as $Q^TQ \leq \lambda_1^2$. The two constraints may define a convex feasible set, but it's possible that feasible set is empty.

Trace norm is a 1-norm on singular value. It grows linearly with the size of matrix. But if we use square loss as loss function, its value grows quadratic to the size of rating matrix. A better method is to constrain each category with different regularization constant, $\lambda(l) = \lambda_2 \times n(l)$ where $n(l)$ is the number of items in category l. In above analysis, we use item category as example for structured trace norm. The idea can be generalized to other structures including multi-task learning.

4.4 Generalization to $\|W\|_p$

Schatten p-norms are generalization of trace norm and Frobenius norm. It is defined as $\|W\|_p = (\sum_i^n \sigma_i^p)^{1/p}$. With similar ideas, we can define a variational form of Scahtten p-norm.

Lemma 4.4.1. $\|W\|_p^p = min_{V>0}\ \alpha * p * tr(V^\beta WW') + (4\alpha^p)^{\frac{1}{p-2}} * (2-p) * tr(V^{\frac{-p\beta}{2-p}})$ *where* $\alpha, \beta \in \mathbb{R}$ *are free parameters,* $\alpha > 0$, $\beta \neq 0$ *and* $p \neq 2$.

Proof. Take the derivative of the equation's right side and let it be 0, we have:

$$\alpha p\beta V^{\beta-1}WW' - (4\alpha^p)^{\frac{1}{p-2}} p\beta V^{\frac{-p\beta}{2-p}-1} = 0$$

$$V = (2\alpha)^{-\frac{1}{\beta}}(WW')^{\frac{p-2}{2\beta}}$$

Plug V's expression into original function, we get

$$\alpha * p * \mathrm{tr}(V^{\beta}WW') + (4\alpha^{p})^{\frac{1}{p-2}} * (2-p) * \mathrm{tr}(V^{\frac{-p\beta}{2-p}}) = (\frac{p}{2} + \frac{2-p}{2})\mathrm{tr}((WW')^{\frac{p}{2}}) = \|W\|_p^p$$

Further, if we take partial derivative of variational form to W, we get $2\alpha p V^{\beta}W = (WW')^{\frac{p-2}{2}}W$ which is exactly the sub gradient of $\|W\|_p^p$. α and β are free parameters, for simplicity we can choose $\alpha = \beta = 1$, then in this case:

$$\|W\|_p^p = \min_{V>0} p * \mathrm{tr}(VWW') + 4^{\frac{1}{p-2}} * (2-p) * \mathrm{tr}(V^{\frac{-p}{2-p}})$$

To derive $\mathrm{tr}(V^{\frac{-p}{2-p}})$, we can first compute V's SVD as $V = USU'$ since V is symmetric, then $\mathrm{tr}(V^{\frac{-p}{2-p}}) = \sum_i S_{ii}^{\frac{-p}{2-p}}$.

The above analysis does not apply when $p = 2$, but it is trivial since we can directly represent Frobenius norm as $\mathrm{tr}(WW')$. □

The basic idea of variational norm is to construct a quadratic approximation to the matrix norm. When fixing V, the term $\mathrm{tr}(VWW')$ is exactly a quadratic function of W satisfying: (1) the minimum is achieved at $W = 0$, (2) its partial derivative to W is the sub-gradient of $\|W\|_p^p$.

Similarly, we can define soft p-nom as:

$$\|W\|_s^p = \min_{V>0} p * \mathrm{tr}(V(WW' + \epsilon^2 I)) + 4^{\frac{1}{p-2}} * (2-p) * \mathrm{tr}(V^{\frac{-p}{2-p}}).$$

Introducing generalized Schatten p-norm other than trace norm may leads to better performance in matrix completion problem.

Chapter 5

Experiments

5.1 Experiments with reference method

We compare the reference method to a low rank matrix factorization method on Movielens 1M Dataset. The dataset contain 3900 movies by 6040 users. 50 reference users and 50 reference items are selected by the packing algorithm described before. The rank function that compares each pair of ratings is used as our similarity metric. We consider the formulation 3.6. Table 1 compare the RMSE performance of reference method and low rank matrix factorization for different values of the parameter λ. The rank of the matrix factorization is 50. The result shows that the reference method is more stable than basic matrix factorization.

The selection process and learning process are separated in the above method. They can be combined through the greedy algorithm that selects a reference user and a reference item in each iteration. Figure 5-1 illustrates the performance as a function of the number of reference vectors. The performance continuously improves when a new reference vector is added to the reference set. When the number of

λ	2	5	10	20	40
Reference Method	0.907	0.8916	0.8873	0.888	0.8922
Low rank Matrix Factorization	1.0185	0.9326	0.8896	0.8732	0.8868

Table 5.1: RMSE comparison of reference method and low rank matrix factorization

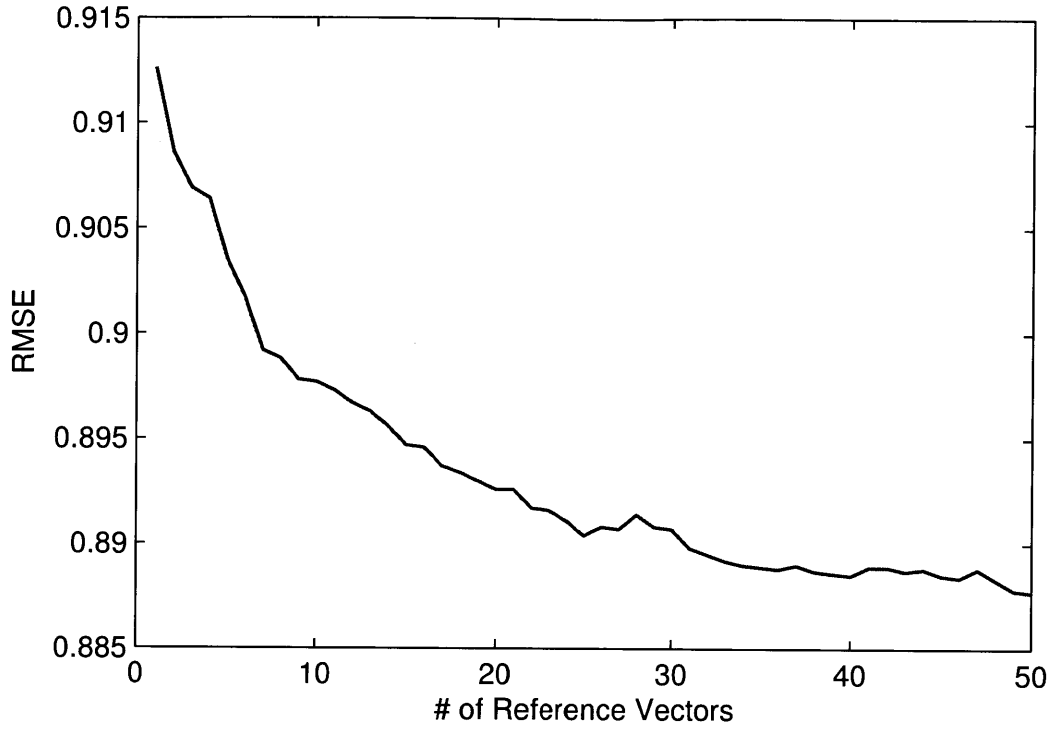


Figure 5-1: Test performance of RMSE as a function of the number of reference vectors.

reference vectors approaches to 50, the change of performance becomes insignificant. It indicates that a small number of reference vectors suffice for good accuracy. In this experiment, λ is set to be 10.

5.2 Experiments with primal algorithms

We compared the soft trace-norm alternating algorithm (STN) to an accelerated proximal gradient (APG) method [29] on MovieLens 10M Dataset. The dataset contain 69878 users and 10677 movies. Since both of the algorithms require SVD in each iteration, running them on the full dataset is very slow. Instead, we randomly sampled 5% users and 5% movies as our training set so as to preserve the proportions. The dataset is skewed in the sense that the number of users are much larger than the number of movies. Notice that the soft trace-norm algorithm computes SVD of WW^T

that only depends on the number of items so each iteration costs less than APG. We also consider using APG method (quadratic approximation to loss function) together with soft trace norm (STN+APG). The algorithm involves the following steps

$$A^r = \frac{1}{2}(\epsilon^2 I + (W^r)^T W^r)^{\frac{1}{2}} \tag{5.1}$$

$$Z^{r+1} = W^r + \frac{a^r - 1}{a^{r-1}} W^{r-1} \tag{5.2}$$

$$W^{r+1} = (Z^{r+1} + 2t(Y - Z^{r+1})(I + \lambda t A^{-1})^{-1} \tag{5.3}$$

where $a^r = \frac{1}{2}(1+\sqrt{1+4(a^{r-1})^2})$ are a sequence of numbers starting from $a^0 = a^1 = 0$ and t is the step size. In theory, $t = 1/L$ is the upper bound for convergence where L is Lipschitz continuity constant of loss function. For square loss, $L = 2$.

Figure 5-2a and 5-2b compare the convergence rate of the three algorithms for $\lambda = 0.1$ and $\lambda = 10$ to their respective objectives (APG's objective is the original trace norm regularization problem while STN and STN+APG optimize a soft version). We choose $\epsilon = (10^{-3}/\lambda)^{0.5}$ such that the smooth penalty $\lambda\epsilon^2$ remains constant. In both cases, the STN algorithm is actually faster than APG. The step size of APG t is fixed at 0.5 which is theoretically the upper bound. A slightly larger value of t performs slightly better but at $t \geq 0.7$ the algorithm no longer converges. Difference of the two algorithms in terms of convergence rate becomes smaller with increasing λ. When λ is very large, both converge in a few iterations, as expected. Besides the value of the objective function, we are also interested in the resulting differences in the predicted rating matrices W. Let W^* refer to the rating matrix that APG converges to. We measure $D(W) = \frac{\|W-W^*\|_F}{\sqrt{mn}}$. We can assess, e.g., how quickly STN converges to a nearly the same matrix. Figure 5-3a and 5-3b provides $D(W)$ as a function of computation time. The asymptotic differences as measured by $D(W)$ are small.

5.3 Experiments with primal-dual algorithm

Unlike with the primal algorithms, we are now able to easily run the method on the full MovieLens 10M Dataset containing 69878 users and 10677 movies. Before adding

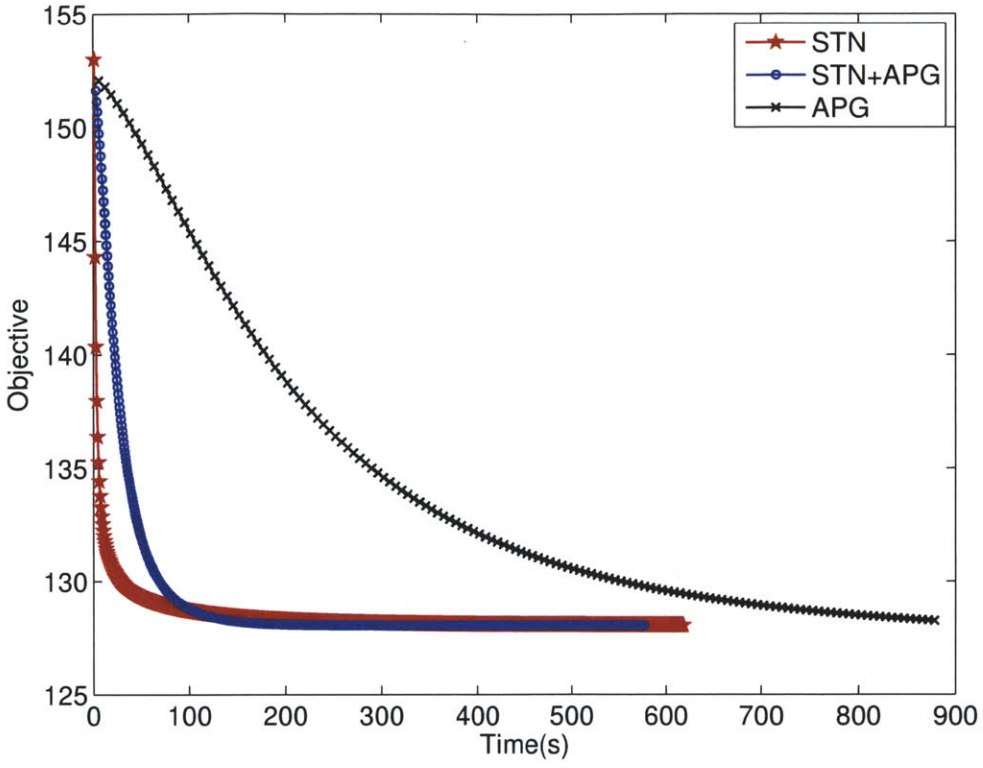
a new constraint, we iterate over current constraints once to update ξ^l. The algorithm (4.3) stops when all k selected constraints are satisfied. The test performance is quite competitive, leading to test root mean squared error (RMSE) of 0.855 with $\lambda = 50$ and $k = 50$ constraints. All the 50 constraints are tight (necessary) though with decreasing effect on the solution (cf. figure 5-4b). To illustrate scaling, figure 5-4a shows the test RMSE as a function of training time (seconds). The algorithm finishes in 15min on a Macbook Intel laptop (2.66GHz). Figure 5-4b illustrates the performance as a function of dimensions (constraints) and shows how indeed only a few constraints are necessary for good performance. Both accuracy and efficiency are quite competitive with other methods (e.g. [34]).

We are also interested in the method's sensitivity to the regularization parameter λ. Figure 5-5a illustrates the test RMSE as a function of λ. An optimal value λ^* exists for test RMSE. The performance declined dramatically when $\lambda < \lambda^*$. Next we look at how maximum eigenvalue of Q^TQ changes in dual block coordinate descent algorithm. Figure 5-5b illustrates the maximum eigenvalue of Q^TQ as a function of iteration. The blue line is the maximum eigenvalue when $\lambda = 50$. The red line is baseline $\lambda^2 = 2500$. The decline rate of maximum eigenvalue becomes smaller with iterations, but the gap between the red line and the baseline still exists after 50 iterations. It indicates that when we approximately solve the dual problem by a low rank solution, the spectral constraint $Q^TQ \leq \lambda^2 I$ may not be exactly satisfied. Figure 5-6 illustrates the dual objective $\mathcal{L}(V, \xi)$ as a function of iterations. The dual objective is monotonically decreasing as expected.

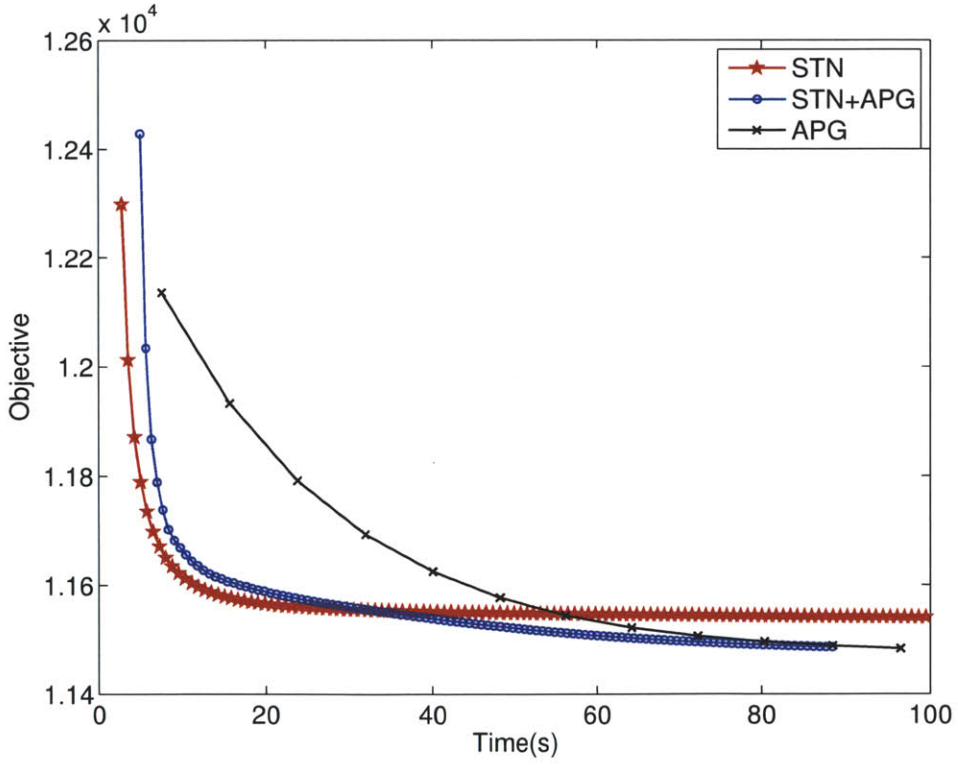
The primal-dual algorithm sequentially updates all the ξ^l and v^l once in one iteration (see 4.3). It may not be the fastest or the most robust method. We consider three different variants. $Alg1$ only updates ξ^i and v^i corresponding to the new constraint in iteration i. $Alg2$ is the algorithm we described earlier that updates all the ξ^l and v^l once. $Alg3$ keeps on updating all the ξ^l and v^l until all the constraints are satisfied. To speed up the algorithm, in $Alg3$ the constraint $\|Qb^l\|^2 \leq \lambda^2$ is relaxed to $\frac{1}{mn}|\xi^l_{new} - \xi^l_{old}|\|v^l\| < 0.001$ which characterizes the impact of updating ξ^l on predictions. An iteration is finished when all the the relaxed constraint are satisfied.

Comparing the three algorithms to each other, $Alg1$ is the fastest and $Alg3$ is the most robust. Figure 5-6a illustrates the dual objective as a function of iterations. λ is set to 50. The dual objective is monotonically decreasing for all three algorithms as expected. $Alg2$ and $Alg3$ are very close while $Alg1$ is slightly worse. Figure 5-6b illustrates the test RMSE as a function of iterations. The test RMSE for the three algorithms are very close. Since there is no significant difference between the three algorithms, $Alg1$ is preferred because of its speed.

In previous experiments, we fix the number of constraints at the beginning of the algorithm. It's interesting to see how many constraints are actually needed. Figure 5-7 illustrates the maximum iteration of $Alg1$ as a function of λ. The stopping criteria is $\frac{1}{mn}|\xi^l|\|v^l\| < 0.001$ which constrain the impact of adding the new constraint on the predicted matrix. The maximum iteration decreases significantly with increasing λ. Since $Alg1$ only updates ξ^l and v^l once when they are first selected, the maximum number of constraints therefore may be over estimated.

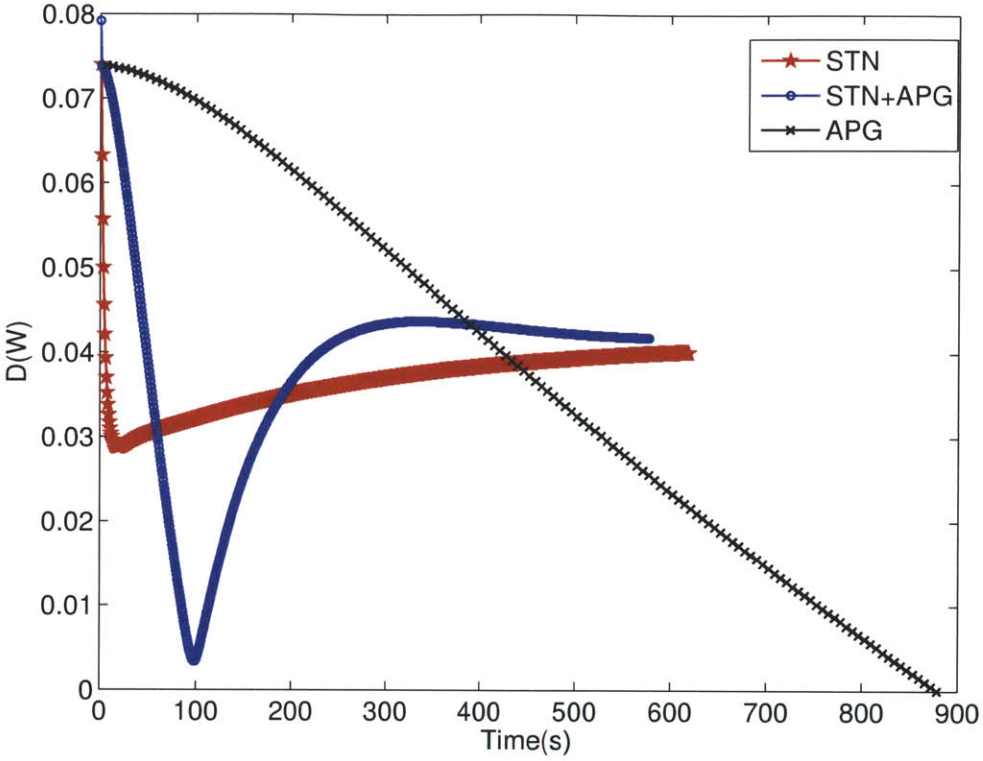


a)

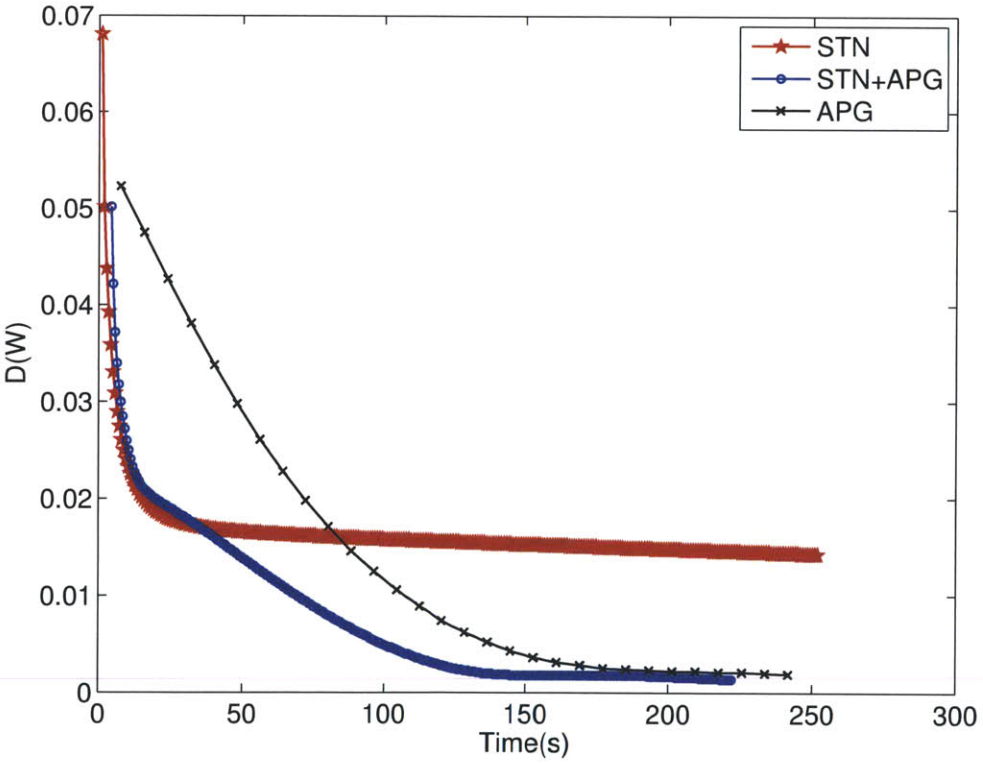


b)

Figure 5-2: Convergence rates of the objective function with a) $\lambda = 0.1$ and b) $\lambda = 10$.

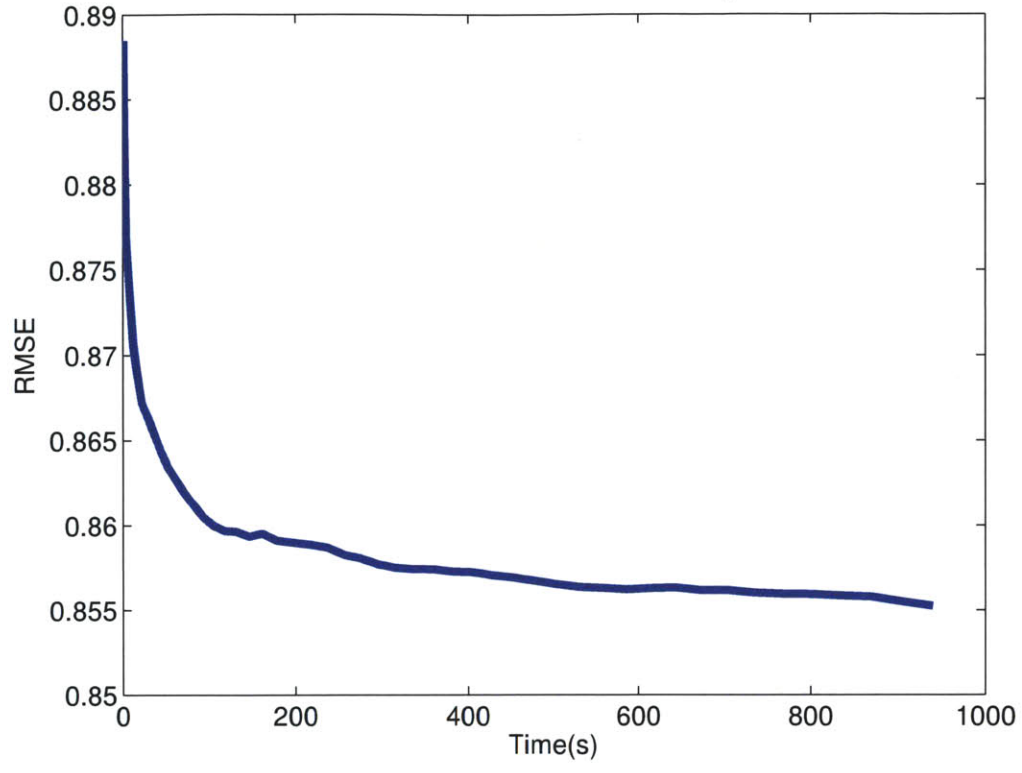


a)

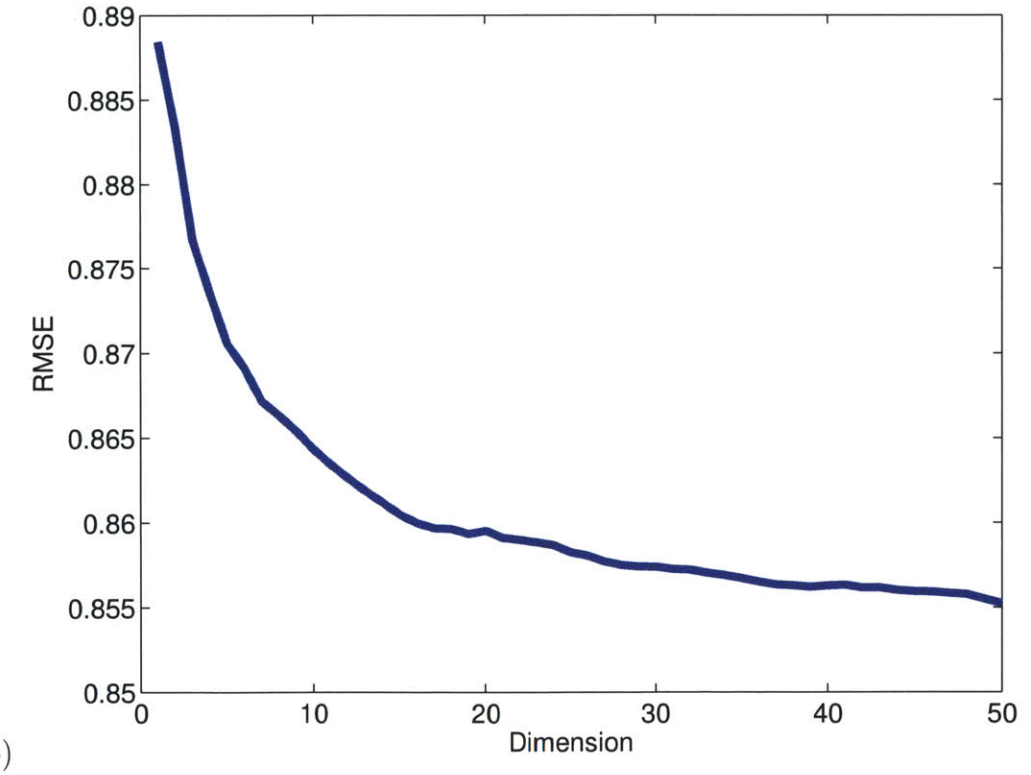


b)

Figure 5-3: Distance to the matrix solution obtained by AGP with a) $\lambda = 0.1$ and b) $\lambda = 10$.

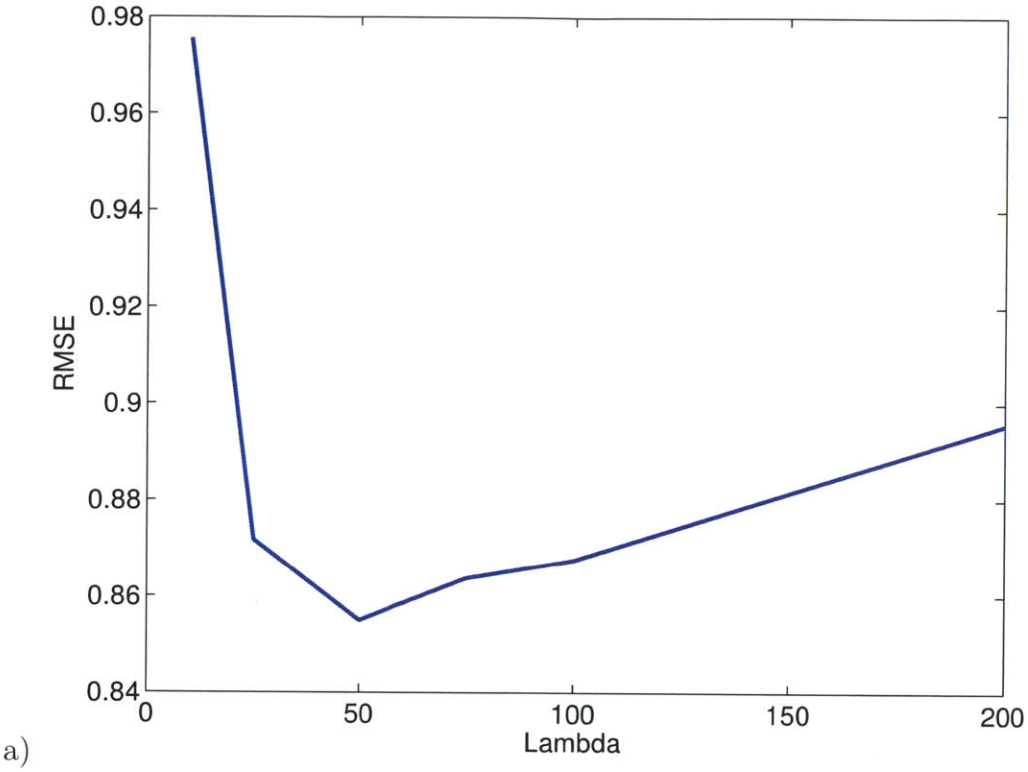


a)

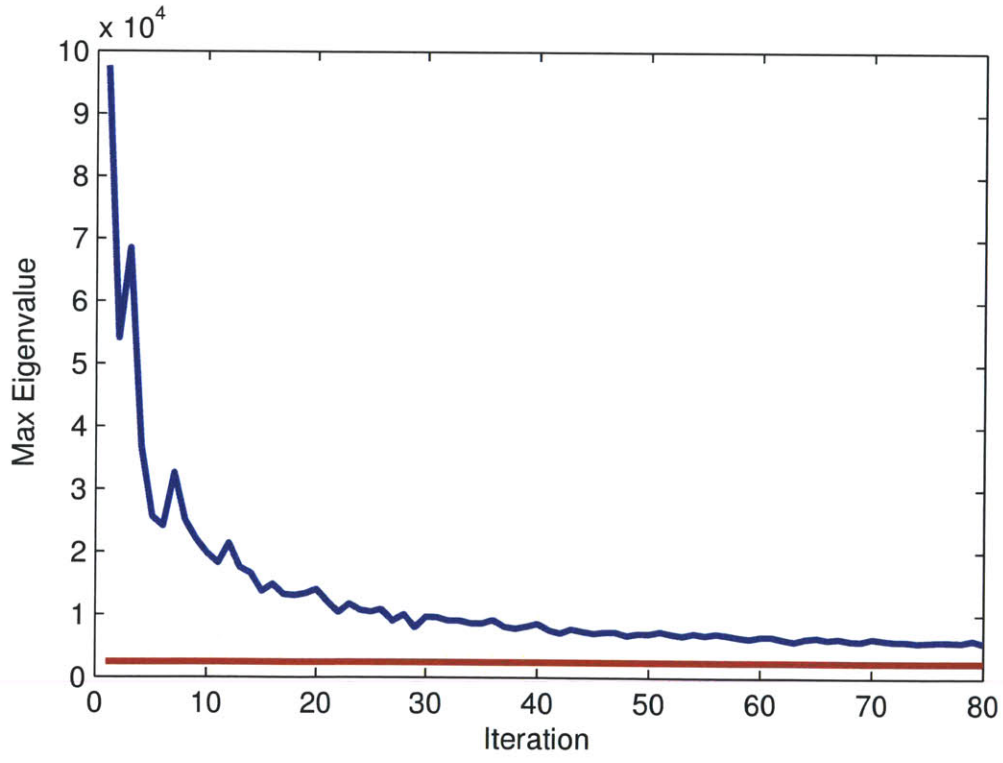


b)

Figure 5-4: a) test RMSE as a function of training time. b) test RMSE as a function of the number of dual constraints.

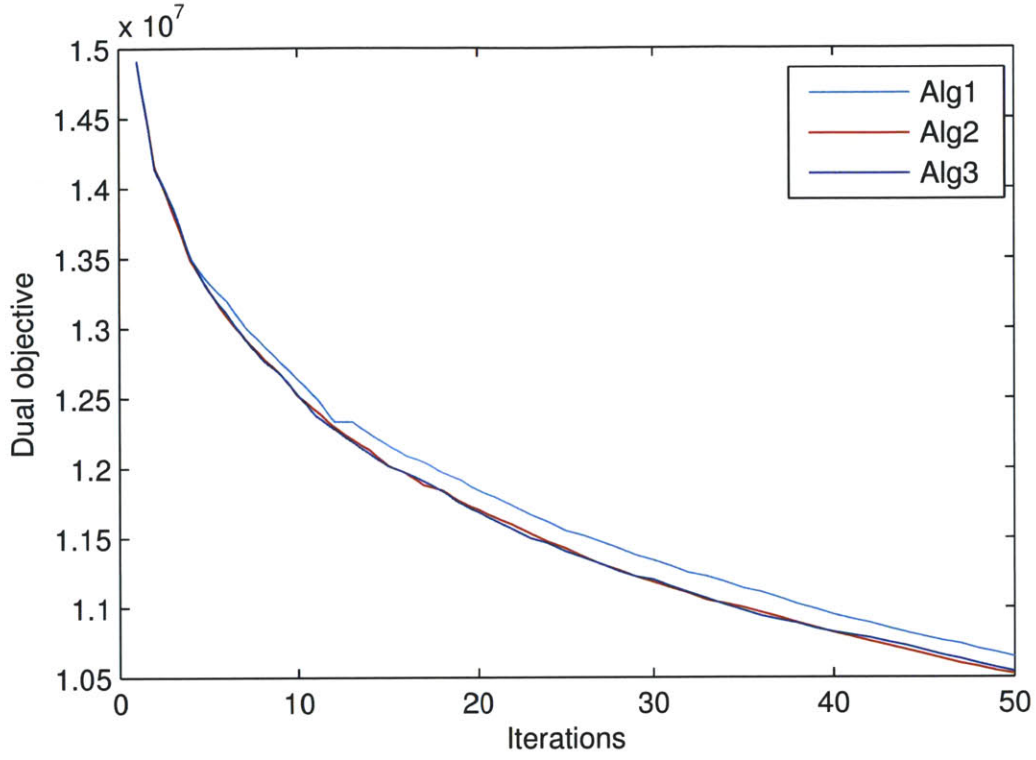


a)

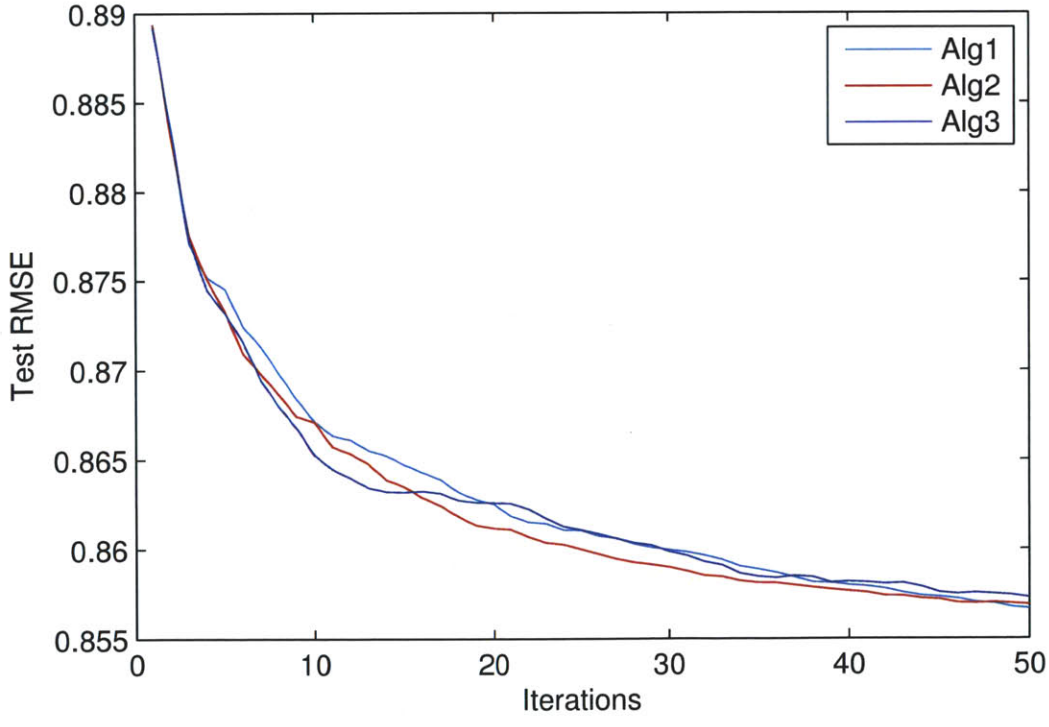


b)

Figure 5-5: a) test RMSE as a function of λ. b) Maximum eigenvalue as a function of iteration.



a)



b)

Figure 5-6: a) Dual objective as a function of iterations. b) Test RMSE as a function of iterations

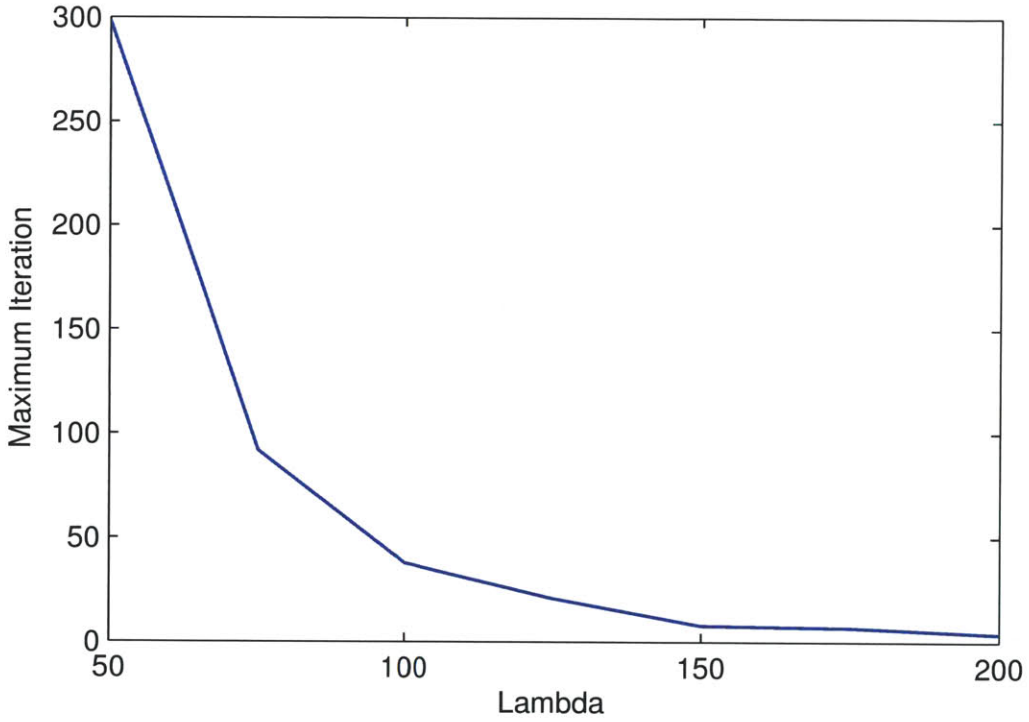


Figure 5-7: Maximum iteration as a function of Lambda

Chapter 6

Conclusion

The key contributions of this thesis are two-fold: a new collaborative flittering method based on a set of reference vectors, and a new primal-dual algorithm for sparse matrix completion with trace norm regularization.

The reference method selects a set of reference vectors that are used to map users and items into feature vectors. The resulting user and item features are then fixed, and we are left with solving convex regression problems. In some cases, the reference method was shown to outperform standard low rank matrix factorization. One key component of the reference method is the similarity metric that compares user/item ratings to the reference set. We used a weighted rank function as a similarity metric. The function compares each pair of ratings and thus robust but comparatively heavy. One interesting problem is to see how the choice of similarity metric affects the performance.

The new primal-dual algorithm for sparse matrix completion is the first method that explicitly constructs primal solution from its dual in trace norm regularization. To solve the constrained dual optimization problem, we proposed a block coordinate descent algorithm in cutting plan fashion that introduces a constraint in each iteration. It remains effective for large problems such as Movieslens 10M which contains 69878 users and 10677 movies. The advantage of this method is that it does not require SVD in each iteration. Our empirical results show that this algorithm is faster than accelerated proximal gradient algorithm.

We showed also that robust collaborative filtering that separates rating matrix into a low rank matrix and a sparse matrix, is equivalent to adding a simple constraint on the dual variables. This motivates to study other variants to the trace norm that incorporate information such as item category. The structured trace norm captures global relationship among all items and local relationship within each category at the same time. One of our future work is designing new constraints on dual variables that result in interesting features such as robustness.

Appendix A

Proof

A.1 Proof of theorem 4.2.1

Let $(W^0, A^0), (W^1, A^1), ...$ be a sequence generated by alternating optimization algorithm. The algorithm guarantees that $F(W)$ is monotonically decreasing:

$$F(W^r) = J(W^r, A^r) \geq J(W^{r+1}, A^r) \geq J(W^{r+1}, A^{r+1}) = F(W^{r+1})$$

From the Lipschitz property of $J(W, A)$, we obtain a quadratic upper bound:

$$J(W, A^r) \leq J(W^r, A^r) + \langle \partial_W J(W^r, A^r), W - W^r \rangle + \frac{L}{2}\|W - W^r\|_F^2 \tag{A.1}$$

Let $W = W^r - \frac{1}{L}\partial_W J(W^r, A^r)$, then the upper bound implies that

$$J(W^{r+1}, A^r) - J(W^r, A^r) \leq J(W, A^r) - J(W^r, A^r) \leq -\frac{1}{2L}\|\partial_W J(W^r, A^r)\|_F^2 \tag{A.2}$$

By condition 1), $\partial_W J(W^r, A^r) = \partial_W F(W)$ (for a specific sub-gradient of F). Each iteration then has sufficient decrease

$$F(W^{r+1}) - F(W^r) \leq -\frac{1}{2L}\|\partial_W F(W^r)\|_F^2 \tag{A.3}$$

For non-convex $F(W)$, before W^r enters the solution set, we have $\|\partial_W F(W^r)\| \geq \epsilon$. By summing over the inequality for all r, we get

$$r \leq \frac{2L}{\epsilon^2}(F(W^0) - F(W^r)) \leq \frac{2L}{\epsilon^2}(F(W^0) - F(W^*)) \tag{A.4}$$

Since W is in a compact set, there exists a constant C such that $\|W^r\| \leq C$. If $F(W)$ is also convex, we have

$$F(W^r) - F(W^*) \leq \langle \partial_W F(W^r), W^r - W^* \rangle \leq \|\partial_W F(W^r)\|_F \|W^r - W^*\|_F \tag{A.5}$$

Combining this with the sufficient decrease, we get

$$F(W^r) - F(W^{r+1}) \geq \frac{(F(W^r) - F(W^*))^2}{2L\|W^r - W^*\|_F^2} \geq \frac{1}{8LC^2}(F(W^r) - F(W^*))^2 \tag{A.6}$$

The above inequality implies

$$\frac{1}{F(W^{r+1}) - F(W^*)} - \frac{1}{F(W^r) - F(W^*)} \geq \frac{F(W^r) - F(W^*)}{8LC^2(F(W^{r+1}) - F(W^*))} \geq \frac{1}{8LC^2} \tag{A.7}$$

Summing over all r, we get

$$F(W^r) - F(W^*) \leq \frac{8LC^2(F(W^0) - F(W^*))}{r(F(W^0) - F(W^*)) + 8LC^2} = O(\frac{1}{r}) \tag{A.8}$$

If $F(W)$ is strongly convex with constant k, then

$$F(W^r) - F(W^{r+1}) \geq \frac{(F(W^r) - F(W^*))^2}{2L\|W^r - W^*\|_F^2} \geq \frac{k}{2L}(F(W^r) - F(W^*)) \tag{A.9}$$

The above inequality implies linear convergence rate such that

$$F(W^r) - F(W^*) \leq (1 - \frac{k}{2L})^r (F(W^0) - F(W^*)) \tag{A.10}$$

A.2 Proof of lemma 4.2.2

It has been shown in [30] that for any composite function $f(\sigma(\cdot)) = f \circ \sigma$, its sub gradient is:

$$\partial(f \circ \sigma)(W) = \{P(\text{Diag}\mu)Q | \mu \in \partial f(\sigma(W))\}$$

In the soft trace norm case $f(\sigma_1, ..., \sigma_n) = \sum_i^n \sqrt{\sigma_i^2 + \epsilon^2}$, so $\partial \|W_1\|_s = P\mu(W_1)Q$, where $\mu(W)$ is a diagonal matrix with $\mu_i(W) = \frac{\sigma_i}{\sqrt{\sigma_i^2+\epsilon^2}}$.

Let $W_2 = A\sigma(W_2)B$ and $\gamma = \frac{\epsilon^2}{(C^2+\epsilon^2)^3}$. $(\mu_i - \gamma\sigma_i)$ is larger than 0 and has the same monotonicity as σ_i. Together with Von Neumann's trace theorem (see theorem 2.1 in [30]) such that $\langle X, Y\rangle \leq \langle \sigma(X), \sigma(Y)\rangle$ for any X and Y,

$$\langle W_1, A(\mu(W_2) - \gamma\sigma(W_2))B\rangle \leq \sum_i \sigma_i(W_1)(\mu_i(W_2) - \gamma\sigma_i(W_2)) \tag{A.11}$$

With this inequality, we get

$$\begin{aligned} & \langle W - W_2, P\mu(W_1)Q - A\mu(W_2)B\rangle - \gamma\langle W_1 - W_2, W_1 - W_2\rangle \\ \geq \quad & \sum_i [(\sigma_i(W_1) - \sigma_i(W_2))(\mu_i(W_1) - \mu_i(W_2)) - \gamma(\sigma_i(W_1) - \sigma_i(W_2))^2] \\ \geq \quad & 0 \end{aligned}$$

Bibliography

[1] Y. Koren, *Collaborative Filtering with Temporal Dynamics*, KDD 2009, Paris, France, (2009)

[2] G. A. Watson, *Characterization of the subdifferential of some matrix norms*, Linear Algebra and its Applications, Volume 170, June 1992, Pages 33-45

[3] S. Ma, D. Goldfarb, L. Chen, *Fixed point and Bregman iterative methods for matrix rank minimization*, Technical Report, Department of IEOR, Columbia University, October, 2008.

[4] N. Srebro, *Learning with Matrix Factorizations*, PhD thesis, Massachusetts Institute of Technology, 2004.

[5] N. Srebro and T. Jaakkola. *Weighted low rank approximation.* In 20th International Conference on Machine Learning, 2003

[6] N. Srebro, J. Rennie and T. Jaakkola, *Maximum Margin Matrix Factorizations*, Advances in Neural Information Processing Systems (NIPS) 17, 2005

[7] R. Kumar, S. Vassilvitskii, *Generalized distance between rankings*, WWW '10 Proceedings of the 19th international conference on World wide web

[8] X. Amatriain, H. Kwak, D. Korea, N. Lathia, N. Oliver, J. M. Pujol, *The Wisdom of the Few A Collaborative Filtering Approach Based on Expert Opinions from the Web*, SIGIR '09

[9] P. Melville, R. J. Mooney and R. Nagarajan, *Content-Boosted Collaborative Filtering for Improved Recommendations*, AAAI-2002

[10] J. Abernethy, F. Bach, T. Evgeniou and J. Vert, *A New Approach to Collaborative Filtering: Operator Estimation with Spectral Regularization*, The Journal of Machine Learning Research, Volume 10, 2009

[11] Z. Liu and L. Vandenberghe *Interior-point method for nuclear norm approximation with application to system identification*, Technical report, UCLA Electrical Engineering Department, 2008.

[12] J.-F. Cai, E. J. Cand?es, and Z. Shen, *A singular value thresholding algorithm for matrix completion*, Technical Report 08-77, UCLA Computational and Applied Math, 2008.

[13] B. M. Marlin, R. S. Zemel, Sam Roweis, M. Slaney, *Collaborative Filtering and the Missing at Random Assumption*, UAI 2007

[14] A. S. Lewis, *The Convex Analysis of Unitarily Invariant Matrix Functions*, Journal of Convex Analysis Volume 2 (1995), No.1/2, 173183

[15] A. J. Smola, S.V. N. Vishwanathan, and Q. V. Le, *Bundle Methods for Machine Learning*, Advances in Neural Information Processing Systems 20

[16] A. Goldberg, X. Zhu, B. Recht, J. Sui, and R. Nowak. *Transduction with matrix completion: Three birds with one stone.* In Advances in Neural Information Processing Systems (NIPS) 24. 2010.

[17] R. Salakhutdinov and N. Srebro, *Collaborative Filtering in a Non-Uniform World: Learning with the Weighted Trace Norm.* Neural Information Processing Systems 24, 2011.

[18] D. Hsu, Sham M. Kakade, and T. Zhang, *Robust matrix decomposition with sparse corruptions*, IEEE Transactions on Information Theory, 2011 (to appear).

[19] T. K. Pong, P. Tseng, S. Ji, and J. Ye, *Trace Norm Regularization: Reformulations, Algorithms, and Multi-Task Learning*, SIAM J. Optim. 20, pp. 3465-3489.

[20] R. Salakhutdinov and A. Mnih, *Bayesian Probabilistic Matrix Factorization using MCMC.* In 25th International Conference on Machine Learning.

[21] B. Sarwar, G. Karypis, J. Konstan, J. Reidl*Item-based collaborative filtering recommendation algorithms.* Proceedings of the 10th international conference on World Wide Web.

[22] A. Agarwal, S. N. Negahban, and M. J. Wainwright, *Fast global convergence of gradient methods for high-dimensional statistical recovery*, NIPS, 2010.

[23] A. Argyriou, C. A. Micchelli, and M. Pontil, *On Spectral Learning*, Journal of Machine Learning Research 11(2010), 905-923.

[24] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM Journal on Imaging Sciences, 2009.

[25] D. S. Bernstein. *Matrix Mathematics: Theory, Facts, and Formulas with Application to Linear System Theory*, Princeton University Press, 2005.

[26] E. J. Candes and B. Recht, *Exact matrix completion via convex optimization*, Found. of Comput. Math., 9 717-772, 2008.

[27] M. Fazel, H. Hindi, and S. Boyd, *A rank minimization heuristic with application to minimum order system approximation*, Proceedings American Control Conference, volume 6, pages 4734-4739, 2001.

[28] A. Goldberg, X. Zhu, B. Recht, J. Sui, and R. Nowak, *Transduction with matrix completion: Three birds with one stone*, NIPS, 2010.

[29] S. Ji and J. Ye, *An accelerated gradient method for trace norm minimization*, ICML, 2009.

[30] A. S. Lewis, *The Convex Analysis of Unitarily Invariant Matrix Functions*, Journal of Convex Analysis Volume 2 (1995), No.1/2, 173183.

[31] T. K. Pong, P. Tseng, S. Ji, and J. Ye, *Trace Norm Regularization: Reformulations, Algorithms, and Multi-task Learning*, SIAM Journal on Optimization, 2009.

[32] K-C. Toh, S. Yun, *An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems*, preprint, Department of Mathematics, National University of Singapore, March 2009.

[33] P. Tseng, *Dual Coordinate Ascent Methods for Nonstrictly Convex Minimization*, Mathematical Programming, Vol. 59, pp. 231247, 1993

[34] M. Jaggi *A Simple Algorithm for Nuclear Norm Regularized Problems*, ICML, 2010.

[35] A. Shashua, A. Levin, *Ranking with Large Margin Principle: Two Approaches*, NIPS 2003.

[36] R. T. Rockafellar and R. J-B Wet, *Variational Analysis*, Grundlehren der Mathematischen Wissenschaften 317, Springer-Verlag 1997