NEW MATRIX COMPLETION MODELS FOR SOCIAL INFORMATION RETRIEVAL APPLICATION

by

JIN HUANG

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
December 2013

To my parents who set the role model and keep supporting me. You are always there when I need help.

ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr. Heng Huang for constantly motivating and encouraging me, and also for his invaluable advice during the course of my doctoral studies. I wish to thank my academic advisors Dr. Chris Ding, Dr. Junzhou Huang and Dr. Yu Lei, for their interest in my research and for taking time to serve in my dissertation committee.

I would also like to extend my appreciation to Dr Feiping Nie. He has proposed many great ideas for my publications and collaborated with me on several ongoing ideas.

I am grateful to all the professors who taught me during the years I spent in school, in different provinces of China and different states of America. Their dedication to research, enthusiasm to teaching motivate my graduate study.

Finally, I would like to express my deep gratitude to my parents who have encouraged and inspired me and sponsored my undergraduate and graduate studies. I am extremely fortunate to be so blessed. I also thank several of my friends who have helped me throughout my career.

NOVEMBER 21, 2013

ABSTRACT

NEW MATRIX COMPLETION MODELS FOR SOCIAL INFORMATION RETRIEVAL APPLICATION

JIN HUANG, Ph.D.

The University of Texas at Arlington, 2013

Supervising Professor: HENG HUANG

Many popular social web sites have emerged during the past decades and completely changed many users' everyday lives. Recently, social information retrieval models, where conventional information retrieval meets the social context of search and recommendation, have become the central topic in machine learning, data mining, information retrieval and many other areas.

A particular area of social information retrieval is the recommendation. Such recommendation ranges from classic movie rating recommendation in user-item matrices, trust and reputation modelling between members in any social network. If we model such recommendation in the form of matrices, then such recommendation can be formulated as recovering missing values in the matrices. This is a classic research topic and there are numerous literature papers regarding this.

In this dissertation, we propose a few different models in terms of social recommendation. Specifically, we develop different models to predict the trust between users in the discrete domain, trust and rating prediction via aggregating heterogeneous social networks, predicting the future events of users. We will introduce these models in different chapters, provide the mathematical derivation for the objective function optimization and demonstrate the effectiveness of these methods with other benchmark methods in each category. These methods provide new perspectives for discovering un-tagged relationships and predicting future events for social networks.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv

ABSTRACT v

LIST OF ILLUSTRATIONS x

LIST OF TABLES xiii

Chapter Page

1. Introduction 1

1.1 Chapter 1 Introduction 1

1.2 Main Scope 3

1.3 Organization Of Following Chapters 4

2. Robust Binary Rank-K Matrix Learning For Trust
And Link Prediction In Social Network 5

2.1 Chapter 2 Introduction 5

2.2 Related Work 8

2.3 Learn Binary Low-Rank Matrix 9

2.3.1 Problem Formulation 10

2.3.2 Conventional Recommendation Methods 10

2.3.3 Robust Binary Rank-k Matrix Learning 12

2.3.4 Optimization Algorithm 12

2.4 Experiments 16

2.4.1 Experiment On Synthetic Data 18

2.4.2 Experiment On Real Data Sets 20

2.5 Chapter 2 Conclusion 23

3. Social Trust Prediction Using Heterogeneous Networks 26

3.1 Chapter 3 Introduction 26

3.2 Related Work 28

3.3 Joint Manifold Factorization 29

3.3.1 Notations 30

3.3.2 Objective Function Formulation 31

3.3.3 General Formulation 32

3.4 Optimization And Algorithm 33

3.4.1 Initialization 33

3.4.2 Computation Of $\mathbf{U}$ 34

3.4.3 Computation Of $\mathbf{V}_1$ 35

3.4.4 COMPUTATION OF $\mathbf{V}_2$ 36

3.4.5 Computation Of c 36

3.4.6 Algorithm Complexity Analysis 37

3.4.7 Optimization Algorithm 39

3.5 Experiments 41

3.5.1 Evaluations On Synthetic Data 42

3.5.2 Evaluations On Real Data 44

3.5.3 Application To Homogeneous Data Set 47

3.5.4 Social Network Regularization Effect Investigation 49

3.6 Chapter 3 Conclusion 50

4. Future Events Recommendation Via Collaborative Ranking 52

4.1 Chapter 4 Introduction 52

4.2 Event Recommendation 56

4.3 Related Work 57

4.4 Collaborative Ranking Framework 59

4.4.1 Events Correlation Capture 61
4.4.2 Collaborative Ranking Objective Function 62
4.5 Optimization Algorithm 63
4.6 Convergence Proof 66
4.7 Experiments 67
4.7.1 Data Set Descriptions 68
4.7.2 Events Relevance And Evaluation Metric 69
4.7.3 Experiments Setup And Parameters 72
4.7.4 Main Results 73
4.7.5 Other Related Discussions 76
4.8 Chapter 4 Conclusion 78
5. Conclusions And Future Work 79
5.1 Key Contributions 79
5.2 Future Work 80

Appendix

A. Convergence Proof Of Theorems In Chapter 3 81
B. Convergence Proof Of Lemma In Chapter 4 85
REFERENCES 87
BIOGRAPHICAL STATEMENT 97

LIST OF ILLUSTRATIONS

Figure Page

2.1 A demonstration for scenarios that an individual user makes trust decisions in various forms, such as accepting others' adding requests from Facebook, believing in news from other twitter users, reading emails sent by others. Green arrow represents the sources this user should trust and red arrow represents the sources this user should distrust. 7

2.2 A demonstration for scenarios where users' relationships change due to unexpected events. We use 1s to represent trusts and 0s to represent distrusts while question mark for unobserved ones. Those changed elements are highlighted in red and the closest scenarios indicate the reasons for such changes. 13

2.3 Matrix completion on synthetic data set with various matrix sizes, noise levels and ranks. (a) RMSE by various methods for matrix completion with p=0.1, k=2 and around 10% known entries are corrupted. (b) RMSE with increased levels of noise, SVD result was omitted due to its poor performance. (c) RMSE with different approximation ranks. 20

2.4 Prediction Errors for 3 Trust Graphs. (a)-(c) are the prediction error plots using different threshold values for different data sets. (d) is the relative ratio plot using RDMC results as the baseline, the threshold for each one is the default one. 24

3.1 A demonstration for our motivation and learning process. The shared group structure matrix is jointly determined by the rating graph and trust graph. The rating matrix contains 2 groups of users' reviews about movies, where a smile face represents satisfactory and an angry face represents unsatisfactory. The trust matrix contains users' trust evaluation towards other users, where 1 represents trust and 0 represent distrust. The question mark represents missing value in both graphs. The 1s in cluster information matrix indicate users are in the corresponding group while 0s represent users are not in that group. 30

3.2 Investigation of Parameters in Our Method 45

4.1 A demonstration of general recommendation system. Rows index users and columns index movies. The digits represent individual users' rating towards corresponding movies. The question marks represent missing values to be predicted. Note that missing values are generally assumed to be distributed randomly. 53

4.2 A demonstration of our future events recommendation system. Users and events are indexed by rows and columns, respectively. Events are arranged according to the order of occurrence and grouped into k sessions. Users give their preference ratings regarding the past events. Columns in the k-th session are future events and their ratings are unknown. 57

4.3 A demonstration for individual user choose not to attend a particular event due to various reasons. It could be due to his busy schedule, budget constraint or unexpected reason. This is to illustrate the point individual user might not follow his usual behavior pattern. Therefore, simply relying on historical data to make inference is not reliable for practical application. . . . 61

4.4 An example of email seminar announcement. We use LDA model and bag of words model to extract the content feature. 71

4.5 A comparison of CR, LowRank and RankSVM with increasing training data. MAP is averaged across all users. It can be observed CR method yield consistently better performance. 75

4.6 α and γ curve plot when the other is fixed. MAP is averaged across all users. These two curves demonstrate how past events fit and global latent factor structure affect MAP. 76

4.7 Histogram plot for individual user MAP. We also include noise as benchmark for comparison. 77

LIST OF TABLES

Table Page

2.1 Parameters setting for all methods. Some Matlab implementations call subroutines in non-Matlab code or some numerical packages such as PROPACK [18] to efficiently deal with the sparsity of the matrices involved. p is the sampling ratio. 18

2.2 Investigation into ρ and number of iterations 19

2.3 Description of Data Set 21

2.4 MAE and RMSE for Trust Graphs 25

2.5 RBRK rank value vs MAE and RMSE 25

3.1 Prediction Result for MovieLens 43

3.2 Recall-Precision Curves Evaluations 47

3.3 Rating Graph Evaluation Results 48

3.4 Netflix Evaluation 49

3.5 MovieLens Evaluation 50

3.6 Manifold Term Investigation I 50

3.7 Manifold Term Investigation II 51

4.1 Data Statistics for CSAIL 70

4.2 Classification Methods Accuracy Comparison 75

CHAPTER 1

Introduction

1.1 Chapter 1 Introduction

Web 2.0 sites (such as Facebook and Twitter) have undergone a rapid development during the past a few years. The advancement in communication, especially in mobile technology, enabling people contact each other in innovative ways. Various web and social sites have been coming up, presenting new forms of interaction, communication and collaboration. There are a large number of volunteers working collaboratively on encyclopedia articles, making wikipedia one of the top visited sites and a good source of information. Individual users can tweet latest news and upload live videos, spreading news and messages much faster than conventional media such as TV and printed newspaper. Huge number of interest and profession groups are formed on virtual web sites, users can organize events and expand their networks. Therefore, these web sites have a profound influence on the whole society and fundamentally change many individual users' daily lives.

However, there are multiple factors which make social web sites far from an ideal collaborative platform for information sharing. There are a large number of hackers, spam emails, virus writers, identity thieves flooding on these web sites. Due to the anonymity nature of internet and social community users, preventing such crimes is difficult if not impossible. As a result, many users are taking precautious measures when engaging social network activity. Users are hesitant to accept a stranger's add friend request from Facebook, reluctant to trust tweet messages from un-verified twitter accounts. With such large amount of information about users' interaction activities, public profiles, and private content, the question of whom and what to trust has become an important challenge to the users. Many on-line social networks allow users to explicitly express evaluations on other users, or the content they created. For example, Facebook users may choose to accept or decline adding friend requests from strangers, based on their confidence in trustworthiness. Yahoo allows community members to rate any comment from another user as spam or regular message.

Clearly, social network analysis could potentially address many issues mentioned above and greatly improve users' experience on these web sites. Social network analysis is not new and indeed has been the research topic for many different disciplines. Social scientists adopted the concept of "social networks" since early 20th century to connote complex sets of relationships between members of social systems at all scales, from interpersonal to international. There are quite a few literature works on this topic [1, 2, 3]. To analyze those new social web sites, however, there are quite a few challenges:

1. Limited information. Social network study often needs to deal with data sparsity. For social networks in web 2.0 category, the sparsity becomes a more serious issue. Users are reluctant to share their profiles and activity records due to lack of diligence and privacy concern. For example, Facebook has more than 800 million users, however, individual users generally have at most a thousand friends in their friends lists. In other words, even if we fetch all users' friends lists information, if we try to grasp the pairwise relationship between all users, the available information is less than 0.01%, which is generally very difficult for most conventional machine learning or data mining methods to work.
2. Scalability. Web 2.0 companies usually start from small and gradually have a large user database when they become popular. Still take Facebook as an example, its 800 million nodes size makes it a challenge to do any reasonable social network analysis.
3. Heterogeneity. There is a possibility that entries in such social network are of different types. Take Youtube as an example, users,tags and videos are merged into

the same network. Social network analysis with respect to heterogeneous entities demand new theories and new methods.

4. Dynamicity. Active users in these web sites keep changing due to the registration of new users and loss of existing users. Therefore, the corresponding method needs to be adaptive to such change.

In this dissertation, we focus on proposing different models to address the first challenge. We tackle the information sparsity issue in two ways. The first is trying to take the best use of the available social network data and explore the hidden structure information. The other is to gather external correlated source information and help us to learn the structure.

1.2 Main Scope

There are numerous ways to represent the social network, such as graph or adjacency lists. These two methods both have their advantages and disadvantages. Graph representation is more straightforward and many graph theories can be used to explore the graph structure, but it could cause a big waste of memory and hard disk. On the other hand, adjacency lists are more frequently used to store the social network in its compact form, but analyzing them will definitely be more difficult.

In this dissertation, we adopt the graph representation. Clearly, it is expected that any popular social web site can not be fully loaded into memory on most individual personal computers. Therefore, such representation is generally applicable to medium size social web sites only. However, with the popular distributed computing framework such as MapReduce[4] and Hadoop[1], it is possible if given multiple personal computers. The data sets used in this dissertation are medium due to our limited computation resources, they are

[1] open source version of mapreduce

used more to demonstrate the potential of our proposed methods to data sparsity issue than to address the scale issue.

Also, in this dissertation, we assume social network users are the targets of our research. The 3 main models are the trust prediction between users, where the trust is the directed relation between two individual users; aggregating social trust graph and recommendation system into one learning framework; predicting users' future actions based on past events.

1.3 Organization Of Following Chapters

In the next 3 chapters, we will present the 3 models we mentioned above, with 1 chapter for each model. Due to the variation in the background setting, within each chapter, we first give necessary background information for each model, and the necessarily mathematical notations. Next, we propose our objective function and the corresponding optimization algorithm. We then compare it with a few classic classical methods to demonstrate the effectiveness of our model.

In the last chapter, we will conclude our work and propose a few possible topics for future work in the social network analysis area.

CHAPTER 2

Robust Binary Rank-K Matrix Learning For Trust And Link Prediction In Social Network

2.1 Chapter 2 Introduction

With the rapid growing number of registered users in various social web sites, privacy has become a more serious concern for both users and web sites. Among different kinds of online activities, adding (accepting) other users as friends is a primary one, since many user interactions are built upon this. With the increasing risk of exposing private profile to malicious users, the question of whom to trust has become an important challenge to individual users. Many online social communities allow users to tag (or implicit imply) other users to facilitate the trustworthiness evaluation and expand their networks. It is clear that we can represent these online users (nodes) and their trust relationships (links) using a graph, then whether a user should trust another user is equivalent to finding if there is a link between these two users.

Due to lack of diligence from users' side and the privacy concern, most users only have chance to explicitly tag a small number of users. For instance, individual users rarely have more than one thousand friends considering its 800 million total users in Facebook. Therefore, most trust graphs contain large number of missing values and conventional graph mining methods [5, 6] that rely on local features will have difficulty getting satisfactory prediction results. If we index users as row and columns, then the trust prediction problem can be formulated into a recommendation system between users. Recommendation systems are capable of handling matrices with large number of missing values, a good example of such application is Netflix prize. Recommendation systems generally produce outputs in one of the two ways-through content or collaborative filtering. Low-rank approximation methods are a popular category of collaborative filtering methods, they assume users' interests are determined by a few latent factors. In fact, the relationship between users are often also determined by a few factors, such as social circle, background and interest etc. It has been discovered in [7], people who are in the same social circle often share similar behavior and tastes. In [8], Crandall et al. give the following two main reasons. One is that people generally adopt behavior exhibited by those they interact with. Such process is called social influence. The other more distinct reason is people incline to form relationships with others who are already similar to them. In this paper, we will therefore find a low-rank matrix to predict users' tagging patterns towards other users based on the global structure.

However, there are two subtle issues with the above approach, due to the ignorance of the trust graph structure. Most social network users explicitly tag other users to indicate whether their trust confidence, in binary form. Those conventional methods fail to retain trust graph's discrete structure. Second issue, which is more subtle, appears when attempting to convert the predictions into discrete values using heuristic threshold values. Such conversion is clearly inefficient, due to the extra cost. Meanwhile, the choice of threshold value for each individual user can be very difficult due to the severe sparsity and skewed distribution of trust and distrust votes. Therefore, it is desirable to explicitly restrict the output to the binary discrete domain.

In this chapter, based on our previous work [9, 10], we propose a low-rank matrix completion method that retains the matrix binary structure. The main contributions of our method are essentially threefold:

- We establish the connection between trust prediction problem and matrix completion with low-rank and discrete constraints. Such formulation has theoretical justification from other disciplines and are built based on historical success on similar applica-

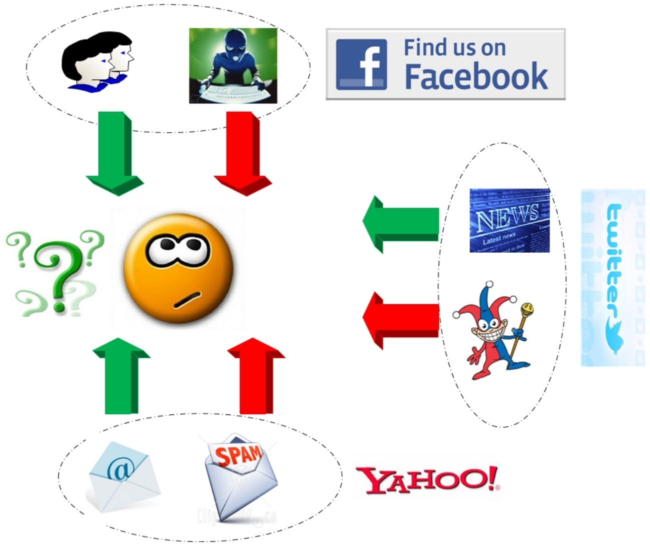


Figure 2.1. A demonstration for scenarios that an individual user makes trust decisions in various forms, such as accepting others' adding requests from Facebook, believing in news from other twitter users, reading emails sent by others. Green arrow represents the sources this user should trust and red arrow represents the sources this user should distrust..

tions. The objective function has a clear motivation and is easy to interpret, it also takes a collaborative filtering point view from both the global structure and the local users' pattern. The binary rank-k collaborative filtering to predict the relationship between users, is first such attempt to the best of our knowledge.

- Based on integer programming formulation, we propose a framework to break down the difficult problem into several manageable pieces. Such optimization simultaneously preserves its both low rank and discrete property, and can be solved effectively by the state-of-the-art optimization techniques. The asymptotic convergence of our framework is straightforward and can be derived from literature materials.
- We have conducted empirical studies using both synthetic and real data in Section **??**. Using synthetic data, we look into the influences of discrete constraint and rank on the prediction accuracies, which also show the robustness of our method. On real

data sets, we conduct empirical experiments and compare our proposed method with a set of classic methods. Results on the real data set also confirm the consistency of our method.

The rest of the chapter is organized as follows: In Section 2.2, we briefly review existing work for collaborative filtering and binary code learning. In Section 2.4, we first formulate the problem of learning binary codes for collaborative filtering as a discrete optimization problem and introduce our objective function. Then, the learning algorithm introduces auxiliary variables and reformulate the objective function, therefore the original difficult one can be decomposed into several sub-problems that can be solved efficiently. Empirical experiments on 3 public-domain data sets are conducted in Section 2.4. We conclude our work and present several future research directions in Section 2.5.

2.2 Related Work

Existing recommendation system methods can be roughly divided into 2 categories: collaborative filtering [11, 12] and content filtering [13, 14]. Collaborative filtering can be further classified as memory based and model based, survey papers of these papers include [15, 16] etc.

Recently, matrix factorization has become a popular direction for collaborative filtering [17, 18, 19, 20]. These methods explore the associations between users and items, so that the latent profiles in lower dimension spaces can capture their characteristics. Most of these methods work in the continuous domain and try to find certain representations between users and items. However, as stated in proceeding section, social network users need explicit messages regarding whether others are trustable, i.e, the corresponding entries of the matrix should be binary. In the literature, binary matrix factorization methods are relatively few except recent papers [21, 22].

Zhou and Zha [21] focus on designing efficient binary recommendation filtering algorithm, which is independent of number of items. They use Hamming distance to preserve the preference of users over items. By minimizing the divergence between training and predicted rating, the learning problem is formulated as a discrete optimization problem. Then they relax the objective function and solve it via existing methods.

Shen et al. [22] consider a rank one binary matrix approximation approach, so as to identify the dominant patterns of the data and preserve its discrete property. Trying to minimize the mismatches between given binary data and approximation matrix is formulated as a 0-1 integer linear problem. In this paper, they propose a relaxation algorithm with regularization term, which is applicable to medium size problem. What is more, they show the relaxation can be formulated into a maximum flow problem and solved efficiently.

These two papers study data matrix pattern on the discrete domain, but fail to consider the low-rank property of data matrix. SVD was a popular method for collaborative filtering, but the prediction result was significantly improved when matrix completion methods took the low-rank assumption into account. Therefore, to better predict the missing values in trust graph, we simultaneously take both binary and low-rank constraint into account and learn its topology structure from a global perspective.

2.3 Learn Binary Low-Rank Matrix

In this section, we describe the proposed method for learning binary low-rank matrix in the category of collaborative filtering. We first describe how trust graph prediction problem can be formulated into a general recommendation system problem. Then we give a introduction to plausible conventional low rank methods, point out their drawbacks and propose our objective function. Next, the learning method based on solving the constrained problem is derived in detail. Finally, we discuss the optimization cost and sketch the convergence proof.

2.3.1 Problem Formulation

The goal of trust prediction is to predict the unobserved relationship between online users according to their past tags. Formally, we assume M_{ij} represents the tag between user i and user j and Ω is the collection of available tags for the whole graph. In particular, we assume M_{ij} is a binary value with either 0 or 1, where 0 represents user i does not trust user j and 1 represents user i does. Note that such setting reflects the general scenarios for nowadays social web sites, where user i can choose to accept user j's connection request. As mentioned in the introduction, trust graphs contain large number of missing values due to the two reasons mentioned in the introduction section of this chapter, therefore, these graphs are challenges for conventional graph mining algorithms. As these graphs share high similarity with user-item matrix such as Netflix prize, recommendation system based methods are appropriate for these cases.

2.3.2 Conventional Recommendation Methods

In this part, we focus on introducing two low-rank approximation methods in the collaborative filtering category. The first one is SVD, which produces an approximation matrix with specified rank.

$$\begin{aligned} &\min_X \|X_\Omega - M_\Omega\|_2^2 \\ &s.t.\ rank(X) \le k. \end{aligned} \tag{2.1}$$

The solution to the above objective function is well known and therefore we skip it here. A significant drawback of SVD approach is its vulnerability to the initial noise due to ℓ_2 norm.

Candès et al [23] proposed to seek a low-rank matrix X such that

$$\begin{aligned} &\min \|X\|_*, \\ &s.t. \quad X_\Omega = M_\Omega \end{aligned} \tag{2.2}$$

where trace norm $\|X\|_*$ is the sum of singular values of matrix X. Other researchers have relaxed the constraints [24] to make the above problem easier to solve

$$\min_X \|X_\Omega - M_\Omega\|_F^2 + \lambda \|X\|_*,$$

where λ is the regularity parameter and

$$\|X_\Omega - M_\Omega\|_F^2 = \sum\nolimits_{(i,j)\in\Omega} (X_{ij} - M_{ij})^2. \tag{2.3}$$

However, there are two potential issues with trace norm minimization approaches. First, the incoherence conditions of the data matrix is often too restrictive, there is no prediction accuracy guarantee when the assumption is not satisfied. The theoretical results in [23] assume that the observed entries are sampled uniformly at random. Unfortunately, many real-world data sets exhibit power-law distributed samples instead [25]. Furthermore, Shi and Yu [26] pointed out that the yielded solution via trace norm minimization is often not low-rank or unique for practical applications. Second, the sparse entries are prone to the influence of outlying or corrupted observations.

There is another issue when applying the above two methods to trust graph prediction. For online social network users, they often desire to get explicit messages whom they should trust. In other words, the desired outputs should still be in binary format. Clearly, both SVD and trace norm minimization produce the outputs in the continuous domain, as a result a separate post-processing is necessary for the conversion. But it brings another two shortcomings: First, it is a ad-hoc procedure, especially difficult for sparse matrices like trust graphs. What is more, such yielded solution is no longer the convex solution to Eq. (2.2), which is against the motivation of trace norm.

2.3.3 Robust Binary Rank-k Matrix Learning

Inspired by the discussion above, we decide our matrix completion should satisfy these properties: First, the output should be in binary discrete format for easy interpretability. Second, the approximation matrix should be in precise low-rank and captures the latent factors of the trust graph. Third, the matrix completion measure should be more robust than ℓ_2 norm used by SVD.

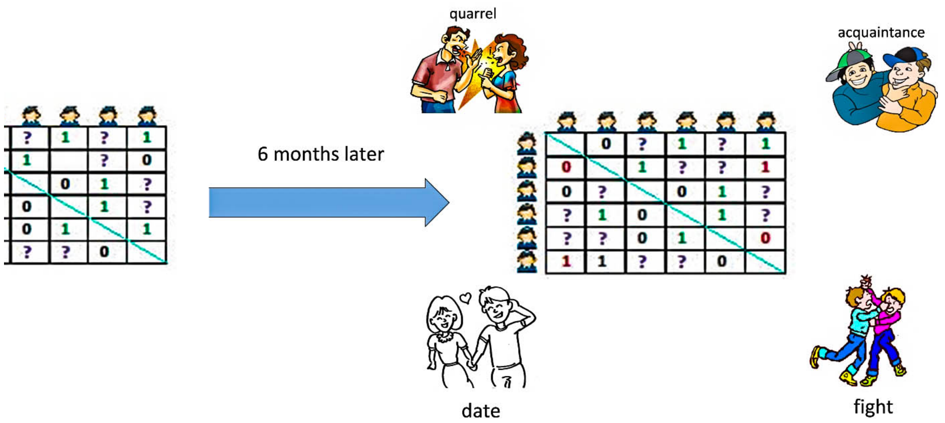
Therefore, we propose the following objective function:

$$\begin{aligned} &\min_X \|X_\Omega - M_\Omega\|_1 \\ &s.t.\ rank(X) \le k, X_{ij} \in \{0,1\} \end{aligned} \tag{2.4}$$

Here we explicitly specify the rank of output matrix and the discrete nature of matrix elements, use ℓ_1 norm as discrepancy measure to alleviate the outlier issue. A more subtle but important consideration for ℓ_1 norm is that trust graph is often dynamic, users' relationship could change due to unexpected events. See Figure 2.2 for a demonstration. We wish our prediction could be stable in spite of local entries change due to individual users. Based on these characteristics of our method, we call out method Robust Binary Rank-K (RBRK). In the next subsection, we will provide the optimization algorithm to Eq. (2.4).

2.3.4 Optimization Algorithm

In this part, we propose to incorporate the Augmented Lagrangian Method (ALM) [27] in our framework. The main idea is to eliminate equality constraints and instead add a penalty term to the cost function that assigns a very high cost to the infeasible points. ALM differs from other penalty-based approaches by simultaneously estimating the optimal solution and Lagrange multipliers in an iterative manner. The main advantages of ALM over other generic algorithms are the fast, accurate performance and independence of problem schemes [28].



?	1	?	1
1		?	0
	0	1	?
0		1	?
0	1		1
?	?	0	

	0	?	1	?	1
0		1	?	?	1
0	?		0	1	?
?	1	0		1	?
?	?	0	1		0
1	1	?	?	0	

Figure 2.2. A demonstration for scenarios where users' relationships change due to unexpected events. We use 1s to represent trusts and 0s to represent distrusts while question mark for unobserved ones. Those changed elements are highlighted in red and the closest scenarios indicate the reasons for such changes..

We first introduce an ancillary variable Y that will be used to approximate X and write Eq. (2.4) into the following one

$$\begin{aligned} &\min_{X,Y} \|X_\Omega - M_\Omega\|_1 \\ &s.t.\ \ X = Y. \end{aligned} \tag{2.5}$$

Then we write it in the following one that is suitable for ALM framework

$$\begin{aligned} &\min_{X,Y} \|X_\Omega - M_\Omega\|_1 + Tr\left(\Sigma^T(X - Y)\right) + \tfrac{\mu}{2}\|X - Y\|_F^2 \\ &s.t.\ \ rank(Y) \leq k, X_{ij} \in \{0, 1\} \end{aligned} \tag{2.6}$$

where Tr is the trace operation for matrix, Σ is the parameter to adjust the discrepancy between X and Y, and μ is the penalty control parameter. The objective function in Eq. (2.6) is not convex in both X and Y. Therefore, it is unrealistic to expect an algorithm to find the global minimum solution, and we apply the alternative optimization technique here. Note that there are also other steps in each iteration that accelerate the convergence, which is the important features of ALM framework.

Step 1: when X is fixed, optimizing with respect to Y is reduced to the following equation.

$$\begin{aligned} &\min_{Y} Tr\left(\Sigma^T(X-Y)\right) + \tfrac{\mu}{2}\left\|X-Y\right\|_F^2 \\ &s.t.\ rank(Y) \leq k. \end{aligned} \tag{2.7}$$

Such equation can be further reduced to the following equation.

$$\begin{aligned} &\min_{Y} \left\|Y - (X + \tfrac{1}{\mu}\sum)\right\|_F^2 \\ &s.t.\ rank(Y) \leq k \end{aligned} \tag{2.8}$$

Assuming the SVD decomposition of $X + \frac{1}{\mu}\Sigma$ is FSG^T, then the solution of Y is

$$Y = F_k S_k G_k^T, \tag{2.9}$$

where S_k contains top k largest values and F_k, G_k are the singular vector matrices corresponding to S_k.

Step 2: when Y is fixed, optimizing with respect to X is reduced to the following one

$$\begin{aligned} &\min_{X_{ij}} \left\|X_\Omega - M_\Omega\right\|_1 + \tfrac{\mu}{2}\left\|X - Y + \tfrac{1}{\mu}\Sigma\right\|_F^2. \\ &s.t.\ X_{ij} \in \{0,1\} \end{aligned} \tag{2.10}$$

Here we solve X based on whether $X_{ij} \in \Omega$ or not.

To solve X_Ω^c, the complement of X_Ω, it is easy to see Eq. (2.10) becomes

$$\begin{aligned} &\min_{X} \left\|X - Y + \tfrac{1}{\mu}\Sigma\right\|_F^2. \\ &s.t.\ X_{ij} \in \{0,1\},\ (i,j) \notin \Omega \end{aligned} \tag{2.11}$$

Note that $\|X\|_F = \sqrt{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} x_{ij}^2}$, therefore minimizing the matrix norm is equivalent to minimizing the square sum of matrix elements. Then Eq. (2.11) can be solved in an element-wise way and we get the following solution since each entry is chosen from the list of discrete values

$$X_{ij} = \mathop{\arg\min}_{c_k \in \{0,1\}} (c_k - Y_{ij} + \frac{1}{\mu}\Sigma_{ij})^2,\ \ (i,j) \notin \Omega. \tag{2.12}$$

Solving X_Ω comes down to the following optimization problem

$$\begin{array}{l} \min\limits_{X} \|X_\Omega - M_\Omega\|_1 + \frac{\mu}{2} \left\| X - Y + \frac{1}{\mu}\Sigma \right\|_F^2 . \\ s.t.\ \ X_{ij} \in \{0,1\}\,,\ \ (i,j) \in \Omega \end{array} \tag{2.13}$$

It can be solved in a similar manner as X_Ω^c.

$$\mathbf{X}_{ij} = \operatorname*{arg\,min}_{c_k \in \{0,1\}} |c_k - M_{ij}| + \frac{\mu}{2}(c_k - Y_{ij} + \frac{1}{\mu}\Sigma_{ij})^2,\ \ (i,j) \in \Omega \tag{2.14}$$

The complete solution for X is given by combining Eq. (2.12) and Eq. (2.14).

Step 3: re-calculate $\Sigma = \Sigma + \mu(X - Y)$ to update the discrepancy between X and Y.

Step 4: update $\mu = \rho\mu$ via a fixed coefficient $\rho > 1$, as number of iterations increase, μ grows exponentially.

The main idea of this optimization framework is to consider the low-rank and discrete constraints separately. In Eq. (2.5), we introduce Y to approximate X. This removes the low-rank constraint on X and the original difficult problem Eq. (2.4) can now be solved in an easy way. For each iteration, Y is the low-rank approximation to X in the continuous domain, such approximation is from the global perspective of the matrix structure. Meanwhile, X seeks the optimal discrete entry for each element in Y, based on the local information without using any threshold parameter. Such framework adopts the recent popular trace norm minimization technique, produces the discrete output and has the overall same computation complexity as SVT. The use of ℓ_1 norm also makes our objective function most robust to noise. This is especially important for application such as trust prediction, where graphs are extremely sparse and prone to outlier influence. The above analysis supports our claim regarding the merits of our method.

With the incorporation of ALM framework, it is easy to notice that $\mu \to \infty$ as the number of iterations increase, X and Y have to be equal in order to keep objective function in Eq. (2.6) finite. In other words, Y asymptotically converges to X, this sketches the intuitive asymptotic convergence of our algorithm. Please see more discussions on ALM algorithm convergence in [29, 30, 31, 32]. In practice, our method usually converges within 30 iterations on the data sets in the experiment section.

The complete steps of our algorithms are summarized in Algorithm (1). It is easy to observe the the computation cost for the algorithm is dominated by the SVD operation in step 1. This step is generally of order $O(m^2n)$, where m and n are row and column sizes of M. Here we use PROPACK package [33]. PROPACK uses the iterative Lanczos algorithm to compute the singular values and singular vectors directly, by using the Lanczos bidiagonalization algorithm with partial reorthogonalization. As a result, it is much faster than the conventional SVD methods. Note that as μ grows exponentially, usually it takes only a few iterations to converge. Therefore, our method is faster than the conventional SVD method for matrix completion. The convergence criteria is the relative change of the objective function value is less than 10^{-4}. The value of ρ has a significant impact on the convergence speed of our algorithm, larger ρ value would reduce the required steps for convergence but meanwhile compromise the accuracy of final objective function value.

2.4 Experiments

In this section, we empirically evaluate our RBRK method for the low rank matrix completion problem using 1 synthetic and 3 real-world data sets. In the first part regarding synthetic data, we investigate the influence of rank, discrete constraint etc on the output. We compare our method with a set of competitive methods on real data sets in the second part.

```
Algorithm 1: Robust Discrete Matrix Completion
    Input: available entries $\mathbf{M}_\Omega$, ALM parameters $\mu$,$\mathbf{\Sigma}$,$\rho$.
    Initialize $M$,$X$ and $Y$.
    repeat
        Update $Y$ with $Y = F_k S_k G_k^T$ where $F_k$,$S_k$ and $G_k$ are defined in Eq. (2.9).
        Update $X$ with formulas $X_\Omega$ and $X_\Omega^c$ respectively
            $X_{ij} = \arg\min_{c_k \in \{0,1\}} (c_k - Y_{ij} + \frac{1}{\mu}\Sigma_{ij})^2, \;\; (i,j) \notin \Omega$
            $\mathbf{X}_{ij} = \arg\min_{c_k \in \{0,1\}} |c_k - M_{ij}| + \frac{\mu}{2}(c_k - Y_{ij} + \frac{1}{\mu}\Sigma_{ij})^2, \;\; (i,j) \in \Omega$
        $\Sigma = \Sigma + \mu(X - Y)$.
        $\mu = \rho\mu$.
    until Convergence
    Output: $X$
```

The matrix completion evaluation metrics used in this paper are mean average error (MAE) and root mean square error (RMSE)

$$\begin{aligned} &\mathrm{MAE} = \mathrm{mean}\,|\mathbf{X}_{\mathrm{ij}} - \mathbf{M}_{\mathrm{ij}}| \\ &\mathrm{RMSE} = \frac{\sqrt{\mathrm{mean}(\mathbf{X}_{\mathrm{ij}} - \mathbf{M}_{\mathrm{ij}})^2}}{\mathrm{sd}(\mathbf{M}_{\mathrm{ij}})}, \;\; (\mathrm{i},\mathrm{j}) \notin \Omega \end{aligned} \tag{2.15}$$

where sd represents the standard deviation.

The competitive methods include SVD, singular value projection [1](SVP) [34], robust PCA [2](RPCA) [35], singular value thresholding [3](SVT) [36]. Since CFCodeReg [21] also works on binary matrix learning, we include its Orthogonal transformations version.

[1] www.cs.utexas.edu/ pjain/svp/

[2] perception.csl.uiuc.edu/matrix-rank/samplecode.html

[3] www-stat.stanford.edu/ candes/software.html

Table 2.1. Parameters setting for all methods. Some Matlab implementations call subroutines in non-Matlab code or some numerical packages such as PROPACK [18] to efficiently deal with the sparsity of the matrices involved. p is the sampling ratio.

Methods	Environment	Comments
RBRK	Matlab	$\rho = 1.08, \mu = 10^{-3}$,maxiter= 100
SVD	Matlab	tol= 10^{-3}
OPTSpace	Matlab	tol= 10^{-3}
SVP	Matlab+PROPACK	tol= 10^{-3}, vtol= 10^{-3},maxiter= 500,verbosity= 1, $\delta = \frac{1}{20p}$
SVT	Matlab+PROPACK	$\tau = \frac{5}{\sqrt{mn}}, \delta = \frac{1}{20p}$,maxiter= 500, tol= 10^{-4}
RPCA	Matlab+PROPACK	default
CFCodePair	Matlab	default

We set the parameters of all methods as listed in Table 2.1 for all subsequent experiments. We give the ground truth rank of the recovery matrix for all methods (except SVT) unless specified otherwise. Since the default step size for SVT and SVP would result in divergence for some our data sets, we set it a conservative value. For RBRK and SVD, we always initialize the missing entries with random values between 0 and 1, set Y the initialized M for RBRK. We report results below the average of 20 runs.

2.4.1 Experiment On Synthetic Data

We first evaluate our method against other methods for random low-rank matrices and uniform samples. We generate a random rank 2 matrix $M \in (0,1)^{n\times n}$ from 0 to 1 and generate random Bernoulli samples with probability 0.1. Our task is to predict the rest entries based on available samples.

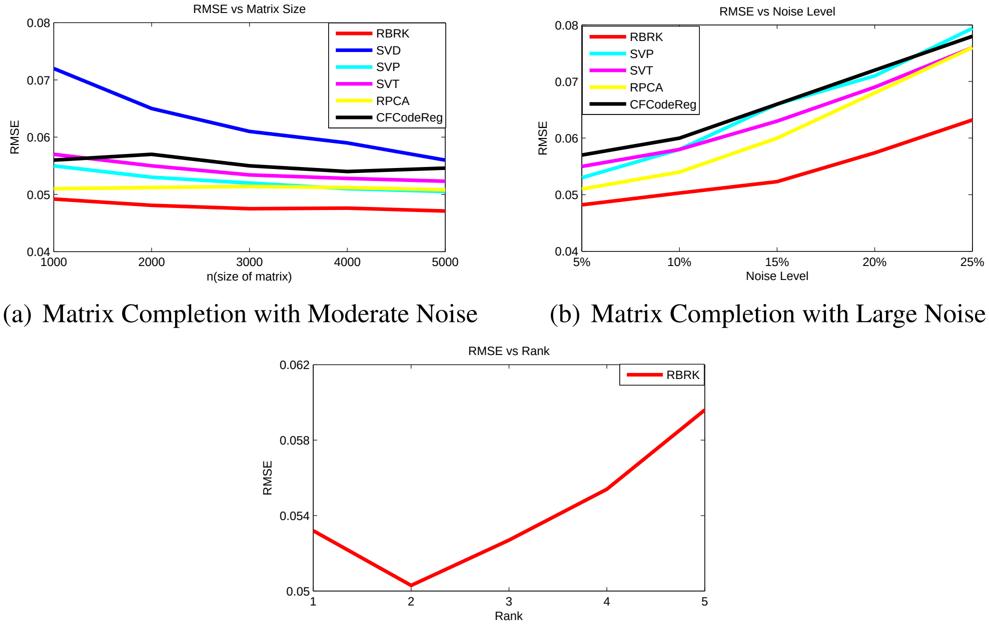
To make the prediction less trivial, we add approximately 5% Gaussian noise and conduct matrix completion experiment as n increases from 1000 to 5000. In Fig. (3(a))it can be observed that SVD is sensitive to moderate noise. Other methods are relatively robust to this level of noise and have very close performance.

Table 2.2. Investigation into ρ and number of iterations

ρ Value	Iterations	RMSE
1.01	123	4.82%
1.02	64	4.78%
1.04	33	4.76%
1.08	21	4.77%
1.2	12	5.36%

Next we fix the matrix size at 2000×2000 but vary the noise level from 5% to 25%. With the increased noise, some entries obviously become outliers with respect to others. In this experiment, we exclude SVD and SVT methods due to their performance in the previous one. In Fig. (3(b)), we can observe RBRK method shows significant better RMSE than all other methods since noise level 15%. With the increased level noise, the matrix has violated the structure assumptions many methods assumed. In contrast, since RBRK has no requirement on the matrix structure, it shows the most robust performance.

Now it comes to the system parameter testing. First we test the rank parameter. In previous experiments, we assume we know the ground truth rank, this is rarely the case for real applications. Hence it is crucial our method can maintain robust performance when the assumed rank is close to the ground truth. We plot the result in Figure 2.3. Second, as our objective function is not convex, we want to emphasize that the performance of RBRK is not very sensitive to the initializations and usually converge within the specified number of iterations given reasonable μ and ρ. Since μ is enlarged in an exponential way, its influence is relatively marginal compared to ρ. Therefore, we decide to list ρ, number of iterations and the corresponding RMSE in Table 2.4.1. We take $n = 2000$, noise level 5% and set $\mu = 10^{-3}$, the average number of iterations (rounded) and RMSE are out of such 10 runs. It can be observed that RMSE is quite stable for all these values, the number of iterations required for convergence is roughly proportional to ρ.



(a) Matrix Completion with Moderate Noise (b) Matrix Completion with Large Noise

(c) Matrix Completion with Moderate Noise

Figure 2.3. Matrix completion on synthetic data set with various matrix sizes, noise levels and ranks. (a) RMSE by various methods for matrix completion with p=0.1, k=2 and around 10% known entries are corrupted. (b) RMSE with increased levels of noise, SVD result was omitted due to its poor performance. (c) RMSE with different approximation ranks..

So far our evaluations are in the continuous domain, for the convenience of most competitive methods. However, for trust graph, entries are restrict to the binary values, then the recall and precision are more appropriate evaluation metrics here.

$$\textit{recall} = \frac{TP}{TP + FN}, \quad \textit{precision} = \frac{TP}{TP + FP}, \tag{2.16}$$

where TP, FN and FP are numbers of true positives, false negatives and false positives respectively. In next subsection, we will also use these two for evaluation.

2.4.2 Experiment On Real Data Sets

The 3 trust graph data sets we use are Epinions [37], Wikipedia [38] and Slashdot [39].

Table 2.3. Description of Data Set

Data set	Epinions	Wikipedia	Slashdot
# of Nodes	2000	2000	2000
# of Trust Links	149,146	81,232	74,561
% among All Links	3.73	2.03	1.86

Epinions was collected in a 5-week crawl from Epinions.com. It consists of two part-s, one is the trust ratings part. The Epinions data set consists of 49,290 users, 487,181 trust statement between users. Users express their web of trust, i.e, reviewers whose reviews and ratings they have consistently found to be valuable and offensive.

Wikipedia records the event that users hold elections to promote some users to administers, who are users with access to additional technical features that aid in maintenance. Here we consider a directed vote between two users as a trust link. Wikipedia contains about 7,000 users and 103,000 trust links.

Slashdot is a technology-related news website thats introduced the Slashdot Zoo feature which allows users to tag each other as friends or foes. The network contains friend/foe links between the users of Slashdot. It contains about 80,000 nodes and more than 900,000 edges.

Note that these (trust) links occupy a very portion of possible links. In other words, most distrust links can be implied implicitly. It can be observed that the distributions of links in these data sets are very skewed due to the domination of distrust links. To alleviate the data skewness for fair comparison and keep the computation manageable, we select top 2,000 highest degree users from each data set. Table 2.4.2 gives a summary description about the subsets used in our experiment. Note that the subsets still carry a skewed distribution in trust and distrust links.

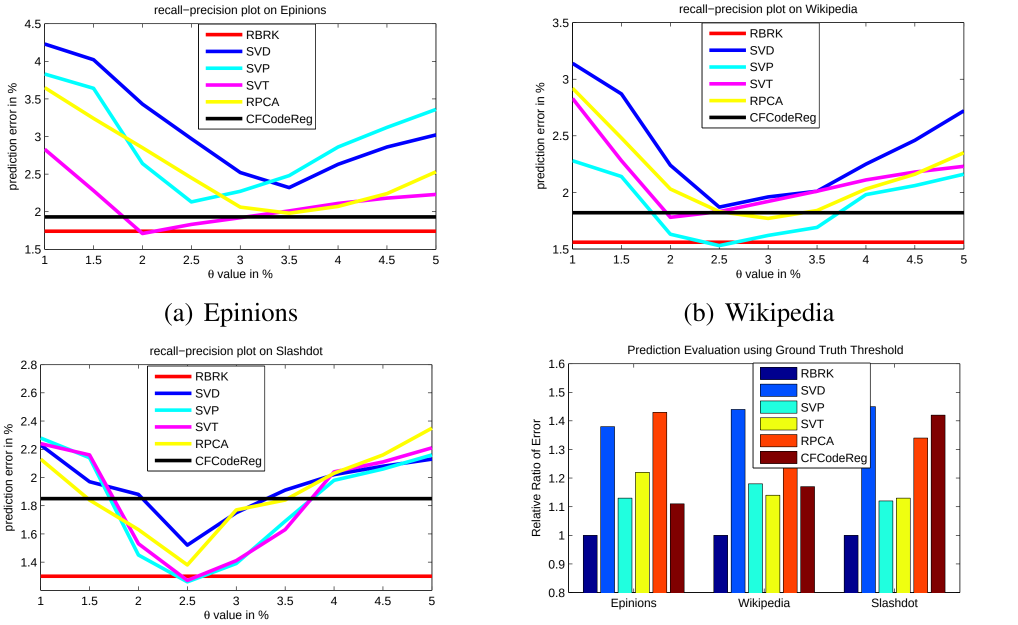
We initialize all methods with the same missing values of X, random values between 0 and 1 unless otherwise specified. For SVD, since the possible choices of rank are clearly dependent upon the particular data set structure, we specify the exact choices when we come to each one. To simulate the actual social network where most trust votes are unknown, we randomly hide 90% of the ground truth entries for prediction, then evaluate performances using different measure metrics and record the optimal values. The reported results in Table 2.4 are the average of 20 times such independent runs. It can be observed that RBRK yields a much lower MAE than all methods while not so much in RMSE. The inconsistence is easy to figure out, any number between 0 and 1 would be even smaller after it gets squared. As a result, due to RBRK discrete outcome requirement, RMSE will penalize it more for a wrong prediction. In previous synthetic data experiment part, we already demonstrated that the our method is not sensitive to the choice of rank. Here in table 5, we confirmed this conclusion again on the 3 real data sets.

Now we want to inspect the prediction accuracy curves as the threshold value varies. As mentioned, trust graphs in this paper have a very skewed distribution. Epinions, the one with largest number of trust links, has less than 4% trust links, i.e, the whole graph has less than 4% 1s out of all entries. As a result, we tune the threshold from 0.01 to 0.05, with 0.5% as increment. For conventional methods that has continuous output, if the initial predicted value is less than the threshold θ, we predict it 0, otherwise 1. We plot the average results in Figure 2.4. Figure (a)-(c) show the prediction errors using various threshold values for each data set. It is easy to observe that, in most cases, RBRK outperforms most methods including those continuous methods, even after we tuned the threshold values in a reasonable manner. What is more, our result is very close to the optimal result from all methods. Note that the rank is the most performance affecting parameter as other ALM parameters mainly affects the convergence speed, however, from Table 2.5, RBRK maintains a robust performance in terms of various ranks. It can be concluded RBRK can achieve a satisfactory performance in most cases for real applications. In other words, our model describes the essence of the trust graph in the real world well and our optimization strategy provides a good solution. There is a possibility that the results from competitive methods could be better if we make more efforts to search the grid in a more exhaustive way. However, such search is often impractical due to the unknown ground truth, limited resources etc. To make this point more clear, we use the default threshold (the proportion of trust links among all entries) in all data sets and plot the relative ratio of prediction error of RBRK results. RBRK shows advantages more than 10% in all data sets in Figure 2.4(d). The threshold we choose is not optimal for all methods, but given situations when ideal threshold is unknown, RBRK clearly is a better choice. The experiments in this subsection clearly supports our motivation. They also demonstrate the superior performance of our method when discrete topology of the data set is desired to retain.

2.5 Chapter 2 Conclusion

Trust and distrust prediction is of critical importance in privacy protection for online users. Due to the extreme sparse tags among users, conventional graph mining algorithms are difficult to apply to such graphs. Therefore it might be appropriate to formulate the trust prediction problem into a collaborative filtering problem. However, an important difference here is that users prefer to get explicit message whether to trust others. In other words, the output is more appropriate to be discrete. Conventional matrix completion methods fail to retain the discrete nature of the trust graphs, and resort to threshold tuning to convert. Such heuristic approach relies on the prior information and fails to produce satisfying result most of the time.

In this chapter, we propose a Robust Binary Rank-K (RBRK) method that retains the discrete nature and meanwhile explores its latent factors. Our framework seeks a low-rank



(a) Epinions

(b) Wikipedia

(c) Slashdot

(d) Relative Ratio Error Plot using Ground Truth Threshold

Figure 2.4. Prediction Errors for 3 Trust Graphs. (a)-(c) are the prediction error plots using different threshold values for different data sets. (d) is the relative ratio plot using RDMC results as the baseline, the threshold for each one is the default one..

binary matrix and is robust to data noise and potential outliers. Different from conventional methods that tune the predictions using heuristic parameters, our method explicitly impose the discrete constraints on the prediction and avoid the post-process step. We solve the difficult integer programming problem via introducing an ancillary variable and decomposing the difficult problem into two manageable pieces. The empirical experiments on synthetic data set shows its performance are not prone to the data noise and system parameters change. The empirical experiments on 3 trust graph data sets demonstrate the effectiveness and robustness of our method, the prediction accuracy is close to the optimal result from competitive methods with threshold tuning.

Table 2.4. MAE and RMSE for Trust Graphs

Data set		MAE(%)	RMSE
Epinions	SVD	2.32 ± 0.12	0.95 ± 0.11
	SVP	2.13 ± 0.08	0.89 ± 0.14
	SVT	2.04 ± 0.09	0.86 ± 0.09
	RPCA	1.92 ± 0.05	$\mathbf{0.83 \pm 0.11}$
	RBRK	$\mathbf{1.74 \pm 0.32}$	0.84 ± 0.10
	CFCodeReg	1.97 ± 0.08	0.86 ± 0.11
Wikipedia	SVD	1.87 ± 0.09	0.86 ± 0.08
	SVP	1.77 ± 0.03	0.82 ± 0.18
	SVT	1.78 ± 0.11	$\mathbf{0.81 \pm 0.03}$
	RPCA	1.62 ± 0.08	0.83 ± 0.05
	RBRK	$\mathbf{1.56 \pm 0.09}$	0.82 ± 0.04
	CFCodeReg	1.71 ± 0.06	0.83 ± 0.05
Slashdot	SVD	1.52 ± 0.04	0.77 ± 0.05
	SVP	1.38 ± 0.02	0.73 ± 0.04
	SVT	1.21 ± 0.03	0.72 ± 0.03
	RPCA	1.17 ± 0.04	0.7 ± 0.03
	RBRK	$\mathbf{1.13 \pm 0.03}$	$\mathbf{0.68} \pm 0.02$
	CFCodeReg	1.19 ± 0.04	0.73 ± 0.04

Table 2.5. RBRK rank value vs MAE and RMSE

Data Set	rank=2	rank=3	rank=4	rank=5	rank=6
Epinion	1.83%	1.77%	1.74%	1.78%	1.81%
	0.89	0.86	0.84	0.86	0.87
Wikipedia	1.63%	1.56%	1.58%	1.60%	1.61%
	0.84	0.82	0.83	0.84	0.86
Slashdot	1.17%	1.13%	1.14%	1.16%	1.18%
	0.70	0.68	0.69	0.70	0.71

CHAPTER 3

Social Trust Prediction Using Heterogeneous Networks

3.1 Chapter 3 Introduction

This chapter is our another work to predict the pairwise relationship between social network users, i.e, conduct trust prediction between social network users. Here we still formulate such problem as recovering missing values in a matrix given only a small part of available observations. In chapter 2, we discussed how to utilize the available entries in the social graph itself to explore its structure, our proposed method seeks a low-rank matrix and meanwhile retains the discrete structure. As we mentioned in Chapter 1, the data sparsity is one of the most important characteristics and meanwhile the biggest challenge to work on this problem. To yield a satisfying result, there are two ways, one is to maximize the usage of available data such as what we did in Chapter 2; the other is to collect related and ancillary data, which we will propose one model to demonstrate this category of methods.

The gathering of online-user data is among the most exciting and controversial business issues in current century. It often brings up concerns about privacy, but it also presents extraordinary opportunities for personalized, one-to-one advertising. Many web sites record users' online activities including purchase history, click history, query log etc. Such information not only reveals individual user's profile to certain extent, but also enables us to find ”similar” users. Here ”similar” users clearly are subject to the measure we choose, but our point is to argue that such information helps us predict the relationship between two online users. It has been discovered in [7], people who are in the same social circle often share similar behaviors and tastes. In [8], Crandall et al. give the following two main reasons. One is that people generally adopt behaviors exhibited by those they interact with. Such process is called social influence. The other more distinct reason is people incline to form relationships with others who are already similar to them. Prior research works on inferring individual user's interests and attributes from his or her social neighbors [40, 41, 42, 43]. These papers show the possibility of improving the users' attributes prediction from the trust graph. However, a straightforward and interesting question can be raised here: is it possible to reverse the direction and explore the trust graph with the users behavior information instead? Or is it possible to achieve an even more aggressive goal, that is, if we construct the auxiliary information graph where a large amount of entries are also missing, can we utilize all the available information and improve the predictions for *both* graphs? Our model employs the idea of transfer learning, or more specific, multi-task learning. There is a survey paper about transfer learning [44].

In this chapter, we propose a Joint Social Networks Mining (JSNM) model to predict the trust and distrust in social network by aggregating heterogeneous social networks from both target trust domain and auxiliary information domain. In this chapter, when we say two graphs are heterogenous, it implies they are from different domains and have no apparent structural similarity and their entries generally have different scales. Without loss of generality, we assume there exists a collection of rating information from the identical social network users in the trust graph. Because the rating information can also be formulated into a graph, our approach is to alleviate the sparsity problem in trust graph by taking advantage of the supplementary knowledge about user behavior and discovering the implicit group-level similarity, which are jointly determined by the user-user trust graph matrix and user-item auxiliary graph matrix. This helps us find the optimal like-minded user groups across both domains. Moreover, we construct the individual affinity graphs to explore the individual geometric structures of the feature manifold to improve the prediction of the missing elements. In addition to the improvement in trust prediction accuracy, our model also helps predict the missing values in the auxiliary matrix. Meanwhile, our method can also be extended to the homogeneous data sets as a powerful collaborative filtering tool. The solution yielded by our algorithm is unique due to the orthonormal constraints and can be easily interpreted. Experimental evaluations have been carried out by using one synthetic data set and two real-world data sets. All empirical results demonstrate that our proposed JSNM method outperforms the classic methods using single social network graph.

The remainder of this paper is organized as follows. In Section 3.2, we first do a brief literature review about the trust or link prediction in social network. In Section 3.3, we describe the notations used in this paper and formulate the new objective function. We will derive our optimization method, provide the algorithm in Section 3.4 and prove the convergence of our new algorithm. We empirically validate the effectiveness of our method for trust prediction in Section 3.5 and conclude the paper in Section 3.6.

3.2 Related Work

Trust prediction can be viewed as a special case of the more general link prediction problem. There have been quite a few methods in link prediction from various perspectives, relational data modeling [45], structural proximity measures [46], and more advanced stochastic relational model [47, 48, 49]. As to the collaborative filtering methods, there are also a few of classic ones, such as memory-based methods [50] to find k-nearest neighbors based on defined similarity measure, model-based methods [51] to learn the preference models for similar users, matrix factorization methods [52, 53, 54] to find a low-rank approximation for the user-item matrix. It is tempted to apply the above collaborative filtering methods to solve the trust prediction problem, however, the trust graph has two structure properties different from the user-item matrix. Trust graph generally has transitivity and symmetric properties between a few nodes. Transitivity enables the trust propagation among users. Symmetry comes from the mutual trust between users in social network. Such additional properties distinguish the trust graph from the typical user-item graphs where collaborative filtering methods are applicable.

Our work is more related to the multi-relational learning, where several relations are modeled jointly and their structures are captured simultaneously. Most methods express the given relations as a few related matrices where a row or column represents an entity. Several methods have been developed to share parameters or structure information by jointly factorizing related matrices so that knowledge can be transferred across different tasks. In [55, 56], an entity is represented by the same latent feature in different matrices. A few Bayesian models were also proposed, such as nonparametric latent variable models [57, 58]. In particular, transfer learning has been applied to collaborative filtering [59], where Pan et al. proposed to take advantage of an auxiliary user-item rating matrix to help the prediction of the target user-item rating matrix. While this idea is intuitive and straightforward, such method is too idealistic to assume the existence of such a related and dense auxiliary rating matrix. Our recent work [60] was the first one utilizing the transfer learning between trust graph and rating graph to simultaneously predict human social behaviors. This chapter is mainly based on [61] and is an extension of our previous work in [60].

3.3 Joint Manifold Factorization

In this section, we will introduce our new JSNM objective function to aggregate the heterogeneous social networks. Prior to this, we will first reveal the implicit connection between the target user-user trust graph and auxiliary user-item rating graph [2].

As mentioned, trusted users in a social network often display similar behavior and tastes. Meanwhile, social network users become friends due to the similar background and interest. Therefore, the trust graph and rating graph should contain some structure similari-

[2] we will use abbreviation trust graph and rating graph for the following context

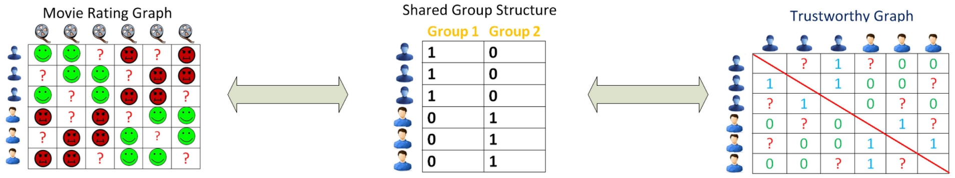


Figure 3.1. A demonstration for our motivation and learning process. The shared group structure matrix is jointly determined by the rating graph and trust graph. The rating matrix contains 2 groups of users' reviews about movies, where a smile face represents satisfactory and an angry face represents unsatisfactory. The trust matrix contains users' trust evaluation towards other users, where 1 represents trust and 0 represent distrust. The question mark represents missing value in both graphs. The 1s in cluster information matrix indicate users are in the corresponding group while 0s represent users are not in that group..

ty in spite of the apparent difference, if the coincidence of the similar ratings contributes to such trust. As a result, the trust prediction accuracy can be improved with the aid of rating graph information and vice versa. In summary, we transfer the knowledge from different domains to circumvent the sparsity constraint and help predict the entries in both matrices. Figure 3.1 is a demonstration of our motivation.

In our proposed solution, we plan to share the implicit group structure between two graphs, which is jointly determined by the trust graph and rating graph. This answers two most important questions for transfer learning: what to transfer and how to transfer [59].

3.3.1 Notations

We use boldface uppercase letters, such as $\mathbf{X}$ to denote matrices, $\mathbf{X}_{i.}$, $\mathbf{X}_{.j}$, X_{ij} to denote the ith row, jth column and the entry located at (i, j) of $\mathbf{X}$, respectively. In our setting, for simplicity, we only discuss two matrices $\mathbf{G}_1$ and $\mathbf{G}_2$ case, then extend the objective function to multiple matrices case. We further assume $\mathbf{G}_1 \in \mathbb{R}^{n\times m_1}$, $\mathbf{G}_2 \in \mathbb{R}^{n\times m_2}$ are the trust graph and rating graph respectively, where n is the number of identical users in both domains, m_1 is the number of users who receive trust votes, m_2 is the number of different items. $\Omega_1 \subset \mathbf{G}_1$ and $\Omega_2 \subset \mathbf{G}_2$ are entries known in corresponding graphs.

3.3.2 Objective Function Formulation

Inspired by the above assumption, we target at the joint matrix factorization to find out the shared group structure between two graphs.

$$\min_{\mathbf{U},\mathbf{V}_1,\mathbf{V}_2,c} \left\|\mathbf{G}_1 - \mathbf{U}\mathbf{V}_1^T\right\|_F^2 + \left\|c\mathbf{G}_2 - \mathbf{U}\mathbf{V}_2^T\right\|_F^2 \tag{3.1}$$

Here $\mathbf{U} \in \mathbb{R}^{n\times l}$, $\mathbf{V}_1 \in \mathbb{R}^{m_1\times l}$, $\mathbf{V}_2 \in \mathbb{R}^{m_2\times l}$ where l is the number of group parameter to be determined. $c > 0$ is a scalar adjusting the scale inconsistency between graphs since the two graphs are from different domains. Here $\mathbf{U}$ is jointly determined by the trust graph and rating graph structures, therefore it provides the shared group structure for both graphs. Since rows represent users in both graphs, we could group users based on $\mathbf{U}$ and then conduct the trust and rating prediction with $\mathbf{V}_1$,$\mathbf{V}_2$, respectively. It can be observed that $\mathbf{U}$ carries the knowledge of both trust graph and rating graph, such framework becomes especially useful since both graphs usually have data sparsity issues for real data sets.

While the above model takes into account of the common row group structure in terms of both matrices, it fails to take into account the social network constrain. To overcome this drawback, we include the Laplacian regularity term [62, 63]. To be specific,

$$\begin{aligned}&\min_{\mathbf{U},\mathbf{V}_1,\mathbf{V}_2,c} \left\|\mathbf{G}_1 - \mathbf{U}\mathbf{V}_1^T\right\|_F^2 + \left\|c\mathbf{G}_2 - \mathbf{U}\mathbf{V}_2^T\right\|_F^2 \\ &+\lambda Tr(\mathbf{V}_1^T\mathbf{L}_1\mathbf{V}_1) + \lambda Tr(\mathbf{V}_2^T\mathbf{L}_2\mathbf{V}_2) \\ &s.t.\ \ \mathbf{V}_1\mathbf{V}_1^T = \mathbf{I}, \mathbf{V}_2\mathbf{V}_2^T = \mathbf{I},\ \ \mathbf{U} \geq 0, \mathbf{V}_1 \geq 0, \mathbf{V}_2 \geq 0\end{aligned} \tag{3.2}$$

Here $\lambda > 0$ is a scalar parameter to be tuned, $\mathbf{L}_1$ and $\mathbf{L}_2$ are the Laplacian graphs based on the columns of $\mathbf{G}_1$ and $\mathbf{G}_2$ respectively, Tr is the trace operation which yields the sum of diagonal elements of the matrix.

In our objective function, to incorporate the social network information, we add two graph Laplacian regularization terms. As the trust graph, $\mathbf{G}_1$ graph explicitly shows the users trust/friendship relations in social networks. As the rating graph, $\mathbf{G}_2$ graph indicates the users taste/hobby similarity, from which we can learn users implicit relations. Thus, the graph Laplacian $\mathbf{L}_1$ and $\mathbf{L}_2$ represent the explicit and implicit social relations of users. When we predict the users trust relations, these existing social relations between users should be preserved. Thus, we add two graph Laplacian regularization terms as in Eq. (3.2).

The details of $\mathbf{L}_1$ and $\mathbf{L}_2$ constructions are given in the next section. We impose the orthogonal constraints on $\mathbf{V}_1$ and $\mathbf{V}_2$ to ensure the uniqueness of the solution. Suppose $\mathbf{U}^*$, $\mathbf{V}_1^*$ and $\mathbf{V}_2^*$ are the solutions to Eq. (3.2), then for any given non-zero constant $c_1 > 1$, $c_1\mathbf{U}^*$ and $\mathbf{V}_1^*/c_1$ would give same value in the first term and lower value for the third term, this is true no matter $\mathbf{U}^*$ and $\mathbf{V}_1^*$ are local or global optimum solutions, the same reasoning applies to $\mathbf{V}_2$, in other words, the optimal solution to Eq. (3.2) does not exist without the constraints. With the orthogonal and non-negative constraints for $\mathbf{V}_1$ and $\mathbf{V}_2$, our solution is the unique local optimum solution for the non-convex objective function 3.2.

3.3.3 General Formulation

There are a few possible generalizations to Eq. (3.2) we want to point out.

First, it can be easily extended to multiple matrices case. The objective function would then be

$$\begin{aligned}&\min_{\mathbf{U},\mathbf{V}_1\ldots\mathbf{V}_n}\sum_{i=1}^{n}\left\|c_i\mathbf{G}_i-\mathbf{U}\mathbf{V}_i^T\right\|_F^2+\lambda\sum_{i=1}^{n}Tr(\mathbf{V}_i^T\mathbf{L}_i\mathbf{V}_i)\\&s.t.\ \ \mathbf{V}_i^T\mathbf{V}_i=\mathbf{I},\mathbf{V}_i\geq 0,\ i=1,\ldots,n\end{aligned}\tag{3.3}$$

The U here would then contain the common information among multiple matrices.

Second, although our motivation is to capture the shared pattern among users, it could be used as a powerful collaborative filtering tool. For example, our framework can also be applied to item-user case, where the reviews are from users in different domains.

3.4 Optimization And Algorithm

In the following, we will derive solution to Eq. (3.2). As we see, minimizing Eq. (3.2) is with respect to $\mathbf{U}$, $\mathbf{V}_1$, $\mathbf{V}_2$ and c, and we can not give a closed-form solution. We will present an alternating scheme to optimize the objective, this procedure repeats until convergence.

3.4.1 Initialization

As mentioned in the introduction, the social graphs generally have large number of missing values, therefore the initialization is almost necessary in trust prediction to replace those missing values for methods that requires similarity calculation or structure exploration. In this paper, for any missing entry $\mathbf{G}_{ij}$, we use mean of the available entries in the corresponding row and column to impute this. For a user-item rating matrix, such initialization combines the available information for both the individual user rating habit and other users' ratings on a particular item. For a user-user trust matrix, such initialization consider both user i and user j 's social circle influence.

After the initial imputation, we construct the Laplacian Graphs of both social networks. As mentioned, the main purpose of the Laplacian terms is to incorporate the data geometric information, because it is found that many real world data distribute on low-dimensional manifold embedded in the high-dimensional ambient space [64]. The Laplacian graph is to discretely approximate the manifold, whose vertices correspond to the data samples, while the edge weight represents the affinity between the data points. One common assumption about the affinity between data points is the cluster assumption [65], which claims if two data samples are close to each other in the input space, then they are also close to each other in the embedding space. This assumption has been widely used in spectral clustering [66, 67, 68]. To be specific, in this paper, we define the edge weight matrix $\mathbf{W}$ as follows:

$$\mathbf{W}_{ij} = \begin{cases} 1 : \mathbf{G}_{i.} \in N_k(\mathbf{G}_{.j}) \; or \; \mathbf{G}_{j.} \in N_k(\mathbf{G}_{.i}) \\ 0 : otherwise \end{cases}$$

where $N_k(\mathbf{G}_{i.})$ denotes the set of k nearest neighbors of $\mathbf{G}_{i.}$. We calculate the Euclidean distances between users for each graph, then construct the corresponding $\mathbf{W}$s based on the top k similar users for each user. It is easy to see $\mathbf{W}$s are symmetric. Let graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is a diagonal matrix whose entries are column sums of $\mathbf{W}$, $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$. Corresponding to trust graph $\mathbf{G}_1$ and rating graph $\mathbf{G}_2$, we construct $\mathbf{L}_1$ and $\mathbf{L}_2$.

After that, we construct $\mathbf{V}_1$ and $\mathbf{V}_2$ based on k-means on columns for $\mathbf{G}_1$ and $\mathbf{G}_2$ respectively. For *i*-th row of $\mathbf{V}_1$, if this row belongs to *j*-th cluster, then $\mathbf{V}_1(i, j) = 1$, all other elements in *i*-th row are 0. $\mathbf{V}_2$ is initialized in the same manner.

Now we come to the optimization of our objective function. When we optimize the objective function Eq. (3.2), we iteratively solve $\mathbf{U}$, $\mathbf{V}_1$, $\mathbf{V}_2$ and c in an alternating manner. In other words, we will optimize the objective with respect to one variable while fixing the other variables. Such process repeats until convergence.

3.4.2 Computation Of U

Optimizing Eq. (3.2) with respect to $\mathbf{U}$ is equivalent to optimizing

$$\begin{aligned} & J_1 = \left\|\mathbf{G}_1 - \mathbf{U}\mathbf{V}_1^T\right\|_F^2 + \left\|c\mathbf{G}_2 - \mathbf{U}\mathbf{V}_2^T\right\|_F^2 \\ & s.t. \;\; \mathbf{V}_1^T\mathbf{V}_1 = \mathbf{I}, \;\; \mathbf{V}_2^T\mathbf{V}_2 = \mathbf{I}, \mathbf{V}_1 \geq 0, \mathbf{V}_2 \geq 0 \end{aligned} \tag{3.4}$$

Setting $\frac{\partial J_1}{\partial \mathbf{U}} = 0$ leads to the following updating formula

$$\mathbf{U} = \frac{\mathbf{G}_1\mathbf{V}_1 + c\mathbf{G}_2\mathbf{V}_2}{2} \tag{3.5}$$

3.4.3 Computation Of $\mathbf{V}_1$

Optimizing Eq. (3.2) with respect to $\mathbf{V}_1$ is equivalent to optimizing

$$\begin{aligned} & J_2 = \left\|\mathbf{G}_1 - \mathbf{U}\mathbf{V}_1^T\right\|_F^2 + \lambda Tr(\mathbf{V}_1^T\mathbf{L}_1\mathbf{V}_1) \\ & s.t. \;\; \mathbf{V}_1^T\mathbf{V}_1 = \mathbf{I}, \mathbf{V}_1 \geq 0 \end{aligned} \tag{3.6}$$

For the constraint $\mathbf{V}_1^T\mathbf{V}_1 = \mathbf{I}$, we can not get a closed-form solution of $\mathbf{V}_1$. Therefore we will present an iterative multiplicative updating algorithm. We introduce the Lagrangian multiplier $\boldsymbol{\alpha} \in \mathbb{R}^{l\times l}$, the corresponding Lagrangian function is

$$L(\mathbf{V}_1) = \left\|\mathbf{G}_1 - \mathbf{U}\mathbf{V}_1^T\right\|_F^2 + \lambda Tr(\mathbf{V}_1^T\mathbf{L}_1\mathbf{V}_1) - Tr(\boldsymbol{\alpha}(\mathbf{V}_1^T\mathbf{V}_1 - \mathbf{I})) \tag{3.7}$$

Setting $\frac{\partial L(\mathbf{V}_1)}{\partial \mathbf{V}_1} = 0$ and use the orthogonal constrain $\mathbf{V}_1^T\mathbf{V}_1 = \mathbf{I}$, we obtain

$$\begin{aligned} & -\mathbf{G}_1^T\mathbf{U} + \lambda\mathbf{L}_1\mathbf{V}_1 - \mathbf{V}_1\boldsymbol{\alpha} = 0 \\ & \Rightarrow \boldsymbol{\alpha} = -\mathbf{V}_1^T\mathbf{G}_1^T\mathbf{U} + \lambda\mathbf{V}_1^T\mathbf{L}_1\mathbf{V}_1 \end{aligned} \tag{3.8}$$

Using the Karush-Kuhn-Tucker condition [69] $\boldsymbol{\alpha} \cdot \mathbf{V}_1 = 0$, where $\cdot$ is the element-wise product operator and thereafter, we get

$$(-\mathbf{V}_1^T\mathbf{G}_1^T\mathbf{U} + \lambda\mathbf{V}_1^T\mathbf{L}_1\mathbf{V}_1) \cdot \mathbf{V}_1 = 0 \tag{3.9}$$

Introduce $\mathbf{L}_1 = \mathbf{L}_1^+ - \mathbf{L}_1^-$, $\mathbf{V}_1 = \mathbf{V}_1^+ - \mathbf{V}_1^-$ and $\mathbf{U} = \mathbf{U}^+ - \mathbf{U}^-$ where $U_{ij}^+ = (|U_{ij}| + U_{ij})/2$ and $U_{ij}^- = (|U_{ij}| - U_{ij})/2$ [70] and $\mathbf{L}_1$,$\mathbf{V}_1$ defined in a similar fashion, we obtain

$$(\mathbf{G}_1^T\mathbf{U}^- + \lambda\mathbf{L}_1^+\mathbf{V}_1 + \mathbf{V}_1\boldsymbol{\alpha}^- - \mathbf{G}_1^T\mathbf{U}^+ - \lambda\mathbf{L}_1^-\mathbf{V}_1 - \mathbf{V}_1\boldsymbol{\alpha}^+) \cdot \mathbf{V}_1 = 0 \tag{3.10}$$

Eq. (3.10) leads to the following updating formula

$$(\mathbf{V}_1)_{ij} \leftarrow (\mathbf{V}_1)_{ij}\sqrt{\frac{\left[\mathbf{G}_1^T\mathbf{U}^+ + \lambda\mathbf{L}_1^-\mathbf{V}_1 + \mathbf{V}_1\boldsymbol{\alpha}^+\right]_{ij}}{\left[\mathbf{G}_1^T\mathbf{U}^- + \lambda\mathbf{L}_1^+\mathbf{V}_1 + \mathbf{V}_2\boldsymbol{\alpha}^-\right]_{ij}}} \tag{3.11}$$

3.4.4 COMPUTATION OF $\mathbf{V}_2$

Optimizing Eq. (3.2) with respect to $\mathbf{V}_2$ is equivalent to optimizing

$$\begin{aligned} &J_3 = \left\| c\mathbf{G}_2 - \mathbf{U}\mathbf{V}_2^T \right\|_F^2 + \lambda Tr(\mathbf{V}_2^T \mathbf{L}_2 \mathbf{V}_2) \\ &s.t. \quad \mathbf{V}_2^T \mathbf{V}_2 = \mathbf{I}, \mathbf{V}_2 \geq 0 \end{aligned} \tag{3.12}$$

The optimization with the above equation is almost identical to the previous subsection,

$$L(\mathbf{V}_2) = \left\| c\mathbf{G}_2 - \mathbf{U}\mathbf{V}_2^T \right\|_F^2 + \lambda Tr(\mathbf{V}_2^T \mathbf{L}_2 \mathbf{V}_2) - Tr(\boldsymbol{\beta}(\mathbf{V}_2^T \mathbf{V}_2 - \mathbf{I})) \tag{3.13}$$

Setting $\frac{\partial L(\mathbf{V}_2)}{\partial \mathbf{V}_2} = 0$ and use the orthogonal constrain $\mathbf{V}_2^T \mathbf{V}_2 = \mathbf{I}$, we obtain

$$\begin{aligned} &-c\mathbf{G}_2^T \mathbf{U} + \lambda \mathbf{L}_2 \mathbf{V}_2 - \mathbf{V}_2 \boldsymbol{\beta} = 0 \\ &\Rightarrow \boldsymbol{\beta} = -c\mathbf{V}_2^T \mathbf{G}_2^T \mathbf{U} + \lambda \mathbf{V}_2^T \mathbf{L}_2 \mathbf{V}_2 \end{aligned} \tag{3.14}$$

Using the Karush-Kuhn-Tucker condition [69] $\boldsymbol{\beta} \cdot \mathbf{V}_2 = 0$, we get

$$(-c\mathbf{V}_2^T \mathbf{G}_2^T \mathbf{U} + \lambda \mathbf{V}_2^T \mathbf{L}_2 \mathbf{V}_2) \cdot \mathbf{V}_2 = 0 \tag{3.15}$$

Introduce $\mathbf{L}_2 = \mathbf{L}_2^+ - \mathbf{L}_2^-$, $\mathbf{V}_2 = \mathbf{V}_2^+ - \mathbf{V}_2^-$ and $\mathbf{U} = \mathbf{U}^+ - \mathbf{U}^-$ where $U_{ij}^+ = (|U_{ij}| + U_{ij})/2$ and $U_{ij}^- = (|U_{ij}| - U_{ij})/2$ [70] and $\mathbf{L}_2$,$\mathbf{V}_2$ defined in a similar fashion, we obtain

$$(c\mathbf{G}_2^T \mathbf{U}^- + \lambda \mathbf{L}_2^+ \mathbf{V}_2 + \mathbf{V}_2 \boldsymbol{\beta}^- - c\mathbf{G}_2^T \mathbf{U}^+ - \lambda \mathbf{L}_2^- \mathbf{V}_2 - \mathbf{V}_2 \boldsymbol{\beta}^+) \cdot \mathbf{V}_2 = 0 \tag{3.16}$$

Eq. (3.16) leads to the following updating formula

$$(\mathbf{V}_2)_{ij} \leftarrow (\mathbf{V}_2)_{ij} \sqrt{\frac{\left[c\mathbf{G}_2^T \mathbf{U}^+ + \lambda \mathbf{L}_2^- \mathbf{V}_2 + \mathbf{V}_2 \boldsymbol{\beta}^+\right]_{ij}}{\left[c\mathbf{G}_2^T \mathbf{U}^- + \lambda \mathbf{L}_2^+ \mathbf{V}_2 + \mathbf{V}_2 \boldsymbol{\beta}^-\right]_{ij}}} \tag{3.17}$$

3.4.5 Computation Of c

Optimizing Eq. (3.2) with respect to c is equivalent to optimizing

$$J_4 = \left\| c\mathbf{G}_2 - \mathbf{U}\mathbf{V}_2^T \right\|_F^2 \tag{3.18}$$

The above task is equivalent to

$$\min_c Tr(c\mathbf{G}_2 - \mathbf{U}\mathbf{V}_2^T)(c\mathbf{G}_2 - \mathbf{U}\mathbf{V}_2^T)^T$$

This can be written as

$$\min_c Ac^2 - 2Bc + D$$

where $A = Tr(\mathbf{G}_2\mathbf{G}_2^T)$, $B = Tr(\mathbf{U}\mathbf{V}_2^T\mathbf{G}_2^T)$, $D = Tr(\mathbf{U}\mathbf{V}_2^T\mathbf{V}_2\mathbf{U}^T)$ It is a quadratic equation in c, the solution is then

$$c = \frac{Tr(\mathbf{U}\mathbf{V}_2^T\mathbf{G}_2^T)}{Tr(\mathbf{G}_2\mathbf{G}_2^T)} \tag{3.19}$$

In summary, we present the iterative multiplicative updating algorithm of optimizing Eq. (3.2) in Algorithm 2. Because the targeted problem is a non-convex one, there is no guarantee that Algorithm 2 will converge to the global optimum. However, the orthogonal constraints in objective function ensure the yielded solution is unique.

The convergence criteria here is the relative change of the object function value at the consecutive steps is less than 10^{-4}. The above loop always exits within 20 iterations for the subsequent experiments.

3.4.6 Algorithm Complexity Analysis

In this part, we want to analyze the time complexity of our algorithm. We would analyze the cost for each phase separately. Let us assume $n \geq max(m_1, m_2)$ to keep the notations simple.

For the missing values initialization, each missing entry needs to calculate its row and column average, of order $O(n + m_1)$ and $O(n + m_2)$ respectively. Therefore, the initialization cost would be $O(n^2m_1)$ and $O(n^2m_2)$ respectively and the total cost would be $O(n^2(m_1 + m_2))$.

Now it comes to the $\mathbf{V}_1$ and $\mathbf{V}_2$ initialization. k-means of $\mathbf{G}_1$ takes $O(knm_1)$ and k-means of $\mathbf{G}_2$ takes $O(knm_2)$, therefore the total cost would be $O(kn(m_1 + m_2))$.

Algorithm 2: Joint Manifold Factorization Algorithm

```
Input: $\mathbf{G}_1$,$\mathbf{G}_2$, maximum number of iterations T
Output: Converged $\mathbf{U}$, $\mathbf{V}_1$ and $\mathbf{V}_2$
Initialize missing entries in $\mathbf{G}_1$ and $\mathbf{G}_2$ using the row-column average.
Initialize $V_1$ and $V_2$ using k-means clustering, initialize c according to scale discrepancy between graphs.
Construct Laplacian graphs $\mathbf{L}_1$ and $\mathbf{L}_2$.
while not converged and iteration t less than T do
    Compute $U = \frac{\mathbf{G}_1\mathbf{V}_1+c\mathbf{G}_2\mathbf{V}_2}{2}$
    Compute $(\mathbf{V}_1)_{ij} \leftarrow (\mathbf{V}_1)_{ij}\sqrt{\frac{\left[\mathbf{G}_1^T\mathbf{U}^+ + \lambda\mathbf{L}_1^-\mathbf{V}_1+\mathbf{V}_1\boldsymbol{\alpha}^+\right]_{ij}}{\left[\mathbf{G}_1^T\mathbf{U}^- + \lambda\mathbf{L}_1^+\mathbf{V}_1+\mathbf{V}_2\boldsymbol{\alpha}^-\right]_{ij}}}$
    Compute $(\mathbf{V}_2)_{ij} \leftarrow (\mathbf{V}_2)_{ij}\sqrt{\frac{\left[c\mathbf{G}_2^T\mathbf{U}^+ + \lambda\mathbf{L}_2^-\mathbf{V}_2+\mathbf{V}_2\boldsymbol{\beta}^+\right]_{ij}}{\left[c\mathbf{G}_2^T\mathbf{U}^- + \lambda\mathbf{L}_2^+\mathbf{V}_2+\mathbf{V}_2\boldsymbol{\beta}^-\right]_{ij}}}$
    Compute $c = \frac{Tr(\mathbf{U}\mathbf{V}_2^T\mathbf{G}_2^T)}{Tr(\mathbf{G}_2\mathbf{G}_2^T)}$
end
```

The last step of the initialization is to construct the Laplacian graphs. It takes $O(knm_1)$ and $O(knm_2)$ to construct the k-nearest neighbor graphs for $\mathbf{G}_1$ and $\mathbf{G}_2$ respectively. The total cost would then be $O(kn(m_1 + m_2))$.

Now it comes to the computation of $\mathbf{U}$, $\mathbf{V}_1$ and $\mathbf{V}_2$. We focus on the discussion of the t-th iteration.

From the $\mathbf{U}$ updating formula Eq. (3.5), it takes at most $O(m_1^3 + m_2^3)$, however, since $\mathbf{V}_1$ was initialized to have only one nonzero element in each row and in general sparse during the updating process, indeed it could be reduced to $O(m_1^2 + m_2^2)$ [71].

For the update of $\mathbf{V}_1$, since $\mathbf{V}_1$ is sparse, it takes $O(k^2m_1^2)$ to calculate $\boldsymbol{\alpha}$, as E-q. (3.11) is an element-wise operation, it takes $O(nkm_1)$ to update $\mathbf{V}_1$.

For the update of $\mathbf{V}_2$, again since $\mathbf{V}_2$ is sparse, it takes $O(k^2m_2^2)$ to calculate $\boldsymbol{\beta}$, as Eq. (3.17) is an element-wise operation, it takes $O(nkm_2)$ to update $\mathbf{V}_2$.

For the update of c, it takes $O(k^2m_2^2)$ to calculate A, $O(m_2^3)$ to calculate B, the total cost would then be $O(m_2^3)$.

Therefore the total cost for one iteration is $O(n^2(m_1 + m_2))$. As specified, our algorithm usually converges in a few iterations independent of matrix size, the total multiplicative update process takes $O(n^2(m_1 + m_2))$. The total complexity of our algorithm is then $O(n^2(m_1 + m_2))$.

3.4.7 Optimization Algorithm

In this subsection, we will prove the convergence of Algorithm 2. We use classic auxiliary function approach used in [72].

Definition 1 (Auxiliary Function) *[72] $Z(h, h')$ is an auxiliary function for $F(h)$ if the conditions*

$$Z(h, h') \geq F(h), Z(h, h) = F(h)$$

are satisfied.

Lemma 1 *[72] If Z is an auxiliary function for F, then F is non-increasing under the update*

$$h^{(t+1)} = \arg\min_{h} Z(h, h^{(t)})$$

Proof 1 $F(h^{(t+1)}) \leq Z(h^{(t+1)}, h^{(t)}) \leq Z(h^{(t)}, h^{(t)}) = F(h^{(t)})$

Lemma 2 *[70] For any nonnegative matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{k\times k}$, $\mathbf{S} \in \mathbb{R}^{n\times k}$, $\mathbf{S}' \in \mathbb{R}^{n\times k}$, and $\mathbf{A}$, $\mathbf{B}$ are symmetric, then the following inequality holds*

$$\sum_{i=1}^{n}\sum_{p=1}^{k} \frac{(\mathbf{A}\mathbf{S}'\mathbf{B})_{ip}\mathbf{S}_{ip}^2}{\mathbf{S}_{ip}^{'}} \geq Tr(\mathbf{S}^T\mathbf{A}\mathbf{S}\mathbf{B})$$

Theorem 1 *Let*

$$J(\mathbf{V}_1) = Tr(\lambda\mathbf{V}_1^T\mathbf{L}_1\mathbf{V}_1 - 2\mathbf{G}_1^T\mathbf{U}\mathbf{V}_1^T + \boldsymbol{\alpha}\mathbf{V}_1^T\mathbf{V}_1) \tag{3.20}$$

Then the following function

$$\begin{aligned}
&Z(\mathbf{V}_1, \mathbf{V}'_1) \\
&= \lambda \sum_{ij} \frac{(\mathbf{L}_1^+ \mathbf{V}'_1)_{ij} \mathbf{V}_{1,ij}^2}{\mathbf{V}'_{1,ij}} - \lambda \sum_{ijk} (\mathbf{L}_1^-)_{jk} \mathbf{V}'_{1,ji} \mathbf{V}'_{1,ki} (1 + \log \frac{\mathbf{V}_{1,ji} \mathbf{V}_{1,ki}}{\mathbf{V}'_{1,ji} \mathbf{V}'_{1,ki}}) \\
&-2 \sum_{ij} \mathbf{G}_1^T \mathbf{U}^+ \mathbf{V}'_{1,ij} (1 + \log \frac{\mathbf{V}_{1,ij}}{\mathbf{V}'_{1,ij}}) + 2 \sum_{ij} \mathbf{G}_1^T \mathbf{U}^- \frac{\mathbf{V}_{1,ij}^2 + \mathbf{V'}_{1,ij}^2}{2\mathbf{V}'_{1,ij}} \\
&+ \sum_{ij} \boldsymbol{\alpha}^+ \mathbf{V}_{1,ij}^2 - \sum_{ijk} \boldsymbol{\alpha}^- \mathbf{V}'_{1,ij} \mathbf{V}'_{1,ik} (1 + \log \frac{\mathbf{V}_{1,ij} \mathbf{V}_{1,ik}}{\mathbf{V}'_{1,ij} \mathbf{V}'_{1,ik}})
\end{aligned}$$

is an auxiliary function for $J(\mathbf{V}_1)$. Furthermore, it is a convex function in $\mathbf{V}_1$ and its global minimum is

$$(\mathbf{V}_1)_{ij} \leftarrow (\mathbf{V}_1)_{ij} \sqrt{\frac{\left[\mathbf{G}_1^T \mathbf{U}^+ + \lambda \mathbf{L}_1^- \mathbf{V}_1 + \mathbf{V}_1 \boldsymbol{\alpha}^+\right]_{ij}}{\left[\mathbf{G}_1^T \mathbf{U}^- + \lambda \mathbf{L}_1^+ \mathbf{V}_1 + \mathbf{V}_2 \boldsymbol{\alpha}^-\right]_{ij}}} \tag{3.21}$$

Proof 2 *See Appendix A.* □

Theorem 2 *Updating $\mathbf{V}_1$ using Eq. (3.11) will monotonically decrease the value of the objective in Eq. (3.2), hence it converges.*

Proof 3 *By Lemma 1 and Theorem 1, we can get that $J(\mathbf{V}_1^0) = Z(\mathbf{V}_1^0, \mathbf{V}_1^0) \geq Z(\mathbf{V}_1^1, \mathbf{V}_1^0) \geq J(\mathbf{V}_1^1) \geq \ldots$so $J(\mathbf{V}_1)$ is monotonically decreasing. As $J(\mathbf{V}_1)$ is nonnegative, i.e, bounded below, the theorem is self-evident.*

Theorem 3 *Let*

$$J(\mathbf{V}_2) = Tr(\lambda \mathbf{V}_2^T \mathbf{L}_2 \mathbf{V}_2 - 2c \mathbf{G}_2^T \mathbf{U} \mathbf{V}_2^T + \boldsymbol{\beta} \mathbf{V}_2^T \mathbf{V}_2) \tag{3.22}$$

Then the following function

$$\begin{aligned}
&Z(\mathbf{V}_2, \mathbf{V}'_2) \\
&= \lambda \sum_{ij} \frac{(\mathbf{L}_2^+ \mathbf{V}'_2)_{ij} \mathbf{V}_{2,ij}^2}{\mathbf{V}'_{2,ij}} - \lambda \sum_{ijk} (\mathbf{L}_2^-)_{jk} \mathbf{V}'_{2,ji} \mathbf{V}'_{2,ki} (1 + \log \frac{\mathbf{V}_{2,ji} \mathbf{V}_{2,ki}}{\mathbf{V}'_{2,ji} \mathbf{V}'_{2,ki}}) \\
&-2 \sum_{ij} c \mathbf{G}_2^T \mathbf{U}^+ \mathbf{V}'_{2,ij} (1 + \log \frac{\mathbf{V}_{2,ij}}{\mathbf{V}'_{2,ij}}) + 2 \sum_{ij} c \mathbf{G}_2^T \mathbf{U}^- \frac{\mathbf{V}_{2,ij}^2 + \mathbf{V'}_{2,ij}^2}{2\mathbf{V}'_{2,ij}} \\
&+ \sum_{ij} \boldsymbol{\beta}^+ \mathbf{V}_{2,ij}^2 - \sum_{ijk} \boldsymbol{\beta}^- \mathbf{V}'_{2,ij} \mathbf{V}'_{2,ik} (1 + \log \frac{\mathbf{V}_{2,ij} \mathbf{V}_{2,ik}}{\mathbf{V}'_{2,ij} \mathbf{V}'_{2,ik}})
\end{aligned}$$

is an auxiliary function for $J(\mathbf{V}_2)$. Furthermore, it is a convex function in $\mathbf{V}_2$ and its global minimum is

$$(\mathbf{V}_2)_{ij} \leftarrow (\mathbf{V}_2)_{ij}\sqrt{\frac{\left[c\mathbf{G}_2^T\mathbf{U}^+ + \lambda\mathbf{L}_2^-\mathbf{V}_2 + \mathbf{V}_2\boldsymbol{\beta}^+\right]_{ij}}{\left[c\mathbf{G}_2^T\mathbf{U}^- + \lambda\mathbf{L}_2^+\mathbf{V}_2 + \mathbf{V}_2\boldsymbol{\beta}^-\right]_{ij}}} \tag{3.23}$$

Proof 4 *See Appendix A.* □

3.5 Experiments

In this paper, we will compare the prediction performance with other methods on both trust graph and rating graph. The competitive methods include average filling (AF), k-nearest neighbors (KNN) using Jaccard's coefficient which is based on nodes similarities, SimRank [73] which is based on path ensembles, SVD approximation [74] and matrix completion via trace norm (MC) [75] which are based on the global graph structure.

We are going to give a brief description about MC since this is a relatively new technique in missing value imputation. MC seeks a lower rank matrix as SVD does. The key difference between MC and SVD is that MC tries to minimize the nuclear norm of the matrix (sum of singular values of matrix), therefore, its *convex* objective function guarantees its global optimum solution. On the other hand, SVD is often stuck at local optimum. MC is generally more robust to outliers than SVD. In this paper, we stack the trust graph and rating graph using common users (movie titles) for matrix completion method in this section in the form of $\mathbf{M} = [\mathbf{G}_1, \mathbf{G}_2]$, to be specific, it attempts to find X such that

$$\begin{aligned} &\min_{\mathbf{X}} \|\mathbf{X}\|_* \\ &s.t.\ \mathbf{X}_\Omega = \mathbf{M}_\Omega \end{aligned}$$

where $\mathbf{M}_\Omega$ is the subset of the observed elements and $\|\mathbf{X}\|_*$ is the trace norm of X. Researchers also relax the constraints and optimize the following one:

$$\min_X \|\mathbf{X}_\Omega - \mathbf{M}_\Omega\|_F^2 + \varsigma\|X\|_*$$

where ς is the regularity coefficient. This would serve as the benchmark transfer learning method in comparison to JSNM. It might be expected that trust graph and rating graph also share their structures with this method, however, as we will demonstrate in the experiment part, such a naive idea does not work well.

For KNN, we search k in the list $\{1, 2, \ldots, 9\}$, to impute the missing value using the node with the highest Jaccard similarity score. For SimRank method, we set the parameters using the default value suggested by the author. For SVD, we choose the rank from the list $(\frac{R}{10}, \frac{2R}{10} ..., R)$, where $R = \min(n, m)$, the minimum of the number of rows and columns. For MC, ς is tuned from the list $\{10^{-2}, 10^{-1}, 1, 10\}$.

3.5.1 Evaluations On Synthetic Data

In this part, we first do the experiments on a synthetic data set, it consists of the MovieLens100K rating graph [76] and the synthetic trust graph we would construct.

MovieLens100K consists of 100,000 ratings (from 1 to 5) from 943 users on 1,682 movies, here each user rated at least 20 movies. Since this data set has around 94% missing values, we first fill in the missing values with the mean of the available information in that row. Then we construct the Laplacian graph $\mathbf{W}$ based on users with parameters setting as follows: Euclidean distance as metric measure, heat kernel with scale parameter 5 and number of neighborhoods $k = 100$. After that, we normalize each column into a ℓ_2 unit vector. At last, we construct the trust graph $\mathbf{T}$ based on the threshold θ which is set at 0.01, $\mathbf{T}(i, j) = 1$ if $\mathbf{W}(i, j) > \theta$ and 0 otherwise. Via the above setting, the two users get 1 mutually(trust each other) if their reviews on items are similar. We find by such procedure, the ratio of 1s in the trust graph is about 12%.

Due to the lack of ground truth for unobservable rating entries, we have to hide existing rating entries to simulate missing ones, here we randomly leave half of them available (about 3%) and mask half of them for test. Since the trust graph is constructed from

Table 3.1. Prediction Result for MovieLens

Prediction Measure	Methods	Result
MAE	AF	0.802 ± 0.005
	KNN	0.812 ± 0.006
	SimRank	0.814 ± 0.006
	SVD	0.962 ± 0.007
	MC	0.826 ± 0.005
	JSNM	$\mathbf{0.745} \pm 0.004$
RMSE	AF	0.996 ± 0.004
	KNN	1.019 ± 0.004
	SimRank	1.024 ± 0.004
	SVD	1.183 ± 0.008
	MC	1.032 ± 0.004
	JSNM	$\mathbf{0.931} \pm 0.003$

the Laplacian graph of the rating graph, so our evaluation would be limited to the rating graph in this subsection. Note that the trust graph is constructed from very limited rating entries, nevertheless, we show that with the auxiliary trust information, the accuracy of rating graph imputation is better than classical imputation methods which explores the rating graph alone.

We adopt two evaluation metrics: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE),

$$\begin{aligned} \mathrm{MAE} &= \sum_{\mathbf{R}_{ij} \in T_E} \left| \mathbf{R}_{ij} - \hat{\mathbf{R}}_{ij} \right| / \left| T_E \right| \\ \mathrm{RMSE} &= \sqrt{\sum_{\mathbf{R}_{ij} \in T_E} \left(\mathbf{R}_{ij} - \hat{\mathbf{R}}_{ij} \right)^2 / \left| T_E \right|} \end{aligned} \tag{3.24}$$

where $\mathbf{R}_{ij}$ and $\hat{\mathbf{R}}_{ij}$ are the true and predicted ratings respectively, $|T_E|$ is the number of test ratings. In all experiments, we run 10 random trials when generating the missing and observed ratings, use AF methods to initialize missing values, do the imputation with all the methods in the 2-fold cross validation process. The averaged results are reported in Table 3.1.

It can be observed that our method consistently outperforms other methods in terms of MAE and RMSE, it successfully incorporates the auxiliary information in the trust graph. We now want to investigate into the influence of parameters for our method. First, we set $l = 3$ and maximum iteration $T = 20$ and vary the value of λ, the MAE and RMSE results are shown in Figure 3.2(a).

Next, with $\lambda = 10^{-3}$ and $T = 20$, we plot the MAE and RMSE curves when the number of clusters l vary in Figure 3.2(b).

The MAE and RMSE results for each iteration have been displayed in Figure 3.2(c) with $l = 3$ and $\lambda = 10^{-3}$.

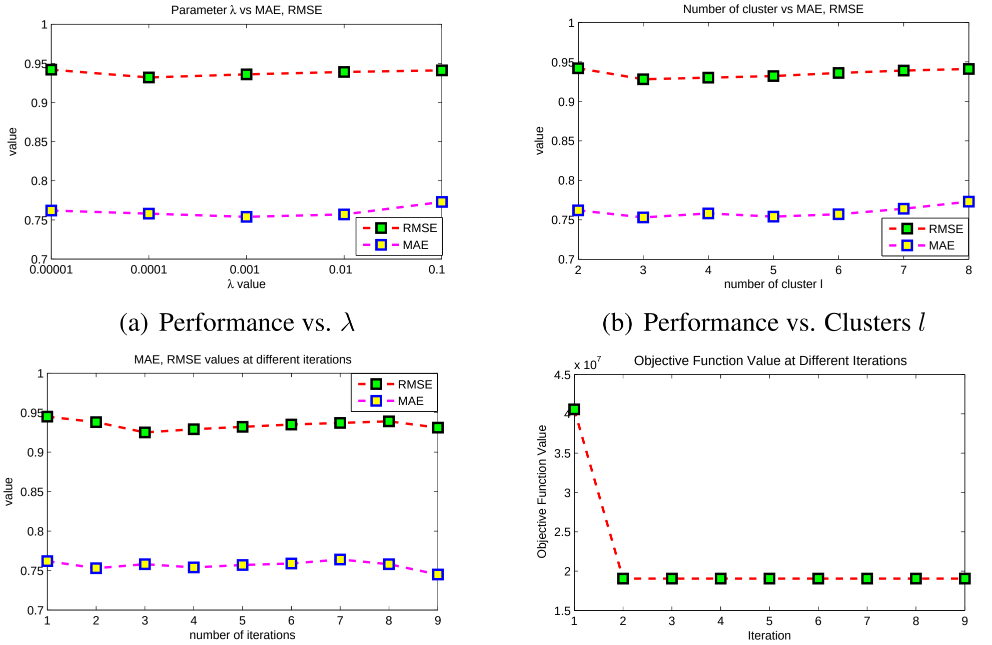
It can be observed that our method is generally robust to the choice the parameter λ, the number of clusters and maximum iterations. For the subsequent experiments, unless otherwise specified, we set $\lambda = 10^{-3}$, $l = 3$ and $T = 20$.

We provided the theoretical proof about the monotone decrease of our objective function in preceding section. To give a concrete example, we also include the objective function value plot using the above default setting. From Figure. 2(d), we can observe the our objective function is very stable as the iteration increases.

This synthetic data set demonstrates that with synthetic auxiliary trust graph, our method has better performance than other classical methods. Our next real data set shows that, such transfer learning process is mutual beneficial, it improves the prediction for both the trust graph and the rating graph.

3.5.2 Evaluations On Real Data

In this part, we will compare our method with other methods on trust prediction using Epinions data set. This data set was collected by Paolo Massa [77] in a 5-week crawl from Epinions.com. It consists of two parts: one is the ratings part, the other is the trust part. The Epinions data set consists of 49,290 users, 139,738 items, 664,824 reviews from



(a) Performance vs. λ

(b) Performance vs. Clusters l

(c) Performance vs. Iterations

(d) Objective Function Value vs. Iterations

Figure 3.2. Investigation of Parameters in Our Method.

users to items, 487,181 trust statement between users. Users express their web of trust, i.e, reviewers whose reviews and ratings they have consistently found to be valuable and offensive [77]. Therefore it is reasonable to assume most individual users tend to cast trust votes towards other users if the users have similar rating patterns towards those items. As a result, the rating matrix and trust matrix could have similar row structure given common users.

Inspired by the above observation, we design the experiments as follows: we select top 2,000 users with the highest degrees (cast and receive most votes), then we select items with more than 68 ratings from the above selected users. The resulting trust graph $\mathbf{G}_1$ of size $2,000 \times 2,000$ has 149,146 trust votes (represented by 1), which consists of 3.73 % of all possible votes, those distrust or unknown votes are represented by 0. The rating graph $\mathbf{G}_2$ of size $2,000 \times 96$ has 10,225 ratings (from 1 to 5), which consists of 5.33 % of all possible ratings, those missing ratings are represented by 0. Among those available ratings, the number of ratings 1,2 and 3 are roughly equal, 4 is twice as many as 1 and 5 is about 4 times as many as 1, such skew distribution might be due to users' reluctance to give low ratings for unsatisfactory items.

Evaluation Metric Since the binary trust votes has a very skewed distribution, precision and recall are more suitable than receiver operating characteristic (ROC) [78]. The precision, recall of the evaluation metric are defined as follows,

$$\begin{aligned} recall = \frac{TP}{TP+FN}, \quad precision &= \frac{TP}{TP+FP} \\ F1 &= \frac{2 \times recall \times precision}{recall + precision} \end{aligned} \tag{3.25}$$

where TP, FN and FP are numbers of true positives, false negatives and false positives, respectively. Since the predicted values for trust graph are generally not 0/1 integers for most methods, here we must decide the transformation criteria. The simplest one is probably the threshold method, if the predicted value is less than the threshold θ, we decide it is 0, otherwise it is 1.

We still hide half of the available entries and conduct the prediction via 2-fold cross validation as previous subsection. To evaluate the prediction result in a comprehensive manner, we calculate the recall and precision values for both trust and distrust, as θ value varies from 0 to 1 with step 0.01. We can then compute the corresponding AUC values for all methods for both trust and distrust predictions, together with F1 score values. From Table 3.2, JSNM has better performance than other methods except F1 score for trust links, where AF shows some slight advantage. Note that it is impractical and time consuming to tune threshold for real application, therefore our method still shows better performance in trust link prediction than AF method considering the significant AUC advantage. We can conclude our method has the best performance in trust prediction in all the methods we listed in terms of trust links and distrust links. Table 3.3 lists all methods' optimal value in terms of MAE and RMSE for the rating graph, again JSNM has the best MAE and RMSE

Table 3.2. Recall-Precision Curves Evaluations

Link	Methods	AUC	F1
Trust	AF	0.207	**0.223**
	KNN	0.183	0.218
	SimRank	0.185	0.218
	SVD	0.123	0.160
	MC	0.075	0.122
	JSNM	**0.215**	0.221
Distrust	AF	0.914	0.977
	KNN	0.916	0.977
	SimRank	0.916	0.977
	SVD	0.583	0.971
	MC	0.972	0.981
	JSNM	**0.992**	**0.991**

results. Based on Table 3.2 and 3.3, we can conclude that transfer learning does provide the bridge for the trust graph and rating graph to share the valuable information with each other. This helps alleviate the common data sparsity issue in social network data. On the other hand, as we have shown, naive transfer learning MC method does not work very well here, MC method fails to extract the common row structure with matrices stacked.

3.5.3 Application To Homogeneous Data Set

The previous two data sets both deal with the trust graph and the rating graph, which are heterogeneous in terms of domain and scale. There are also cases that homogeneous social graph inference desired. One example is to predict user preferences on books and movies. In this subsection, we want to demonstrate our framework also apply to such homogeneous type data, two movie rating data sets. Note that for homogeneous data sets, we would drop the scale adjusting parameter c in objective Eq. (3.2), which is a special case of our framework.

Table 3.3. Rating Graph Evaluation Results

Prediction Measure	Methods	Result
MAE	AF	0.864 ± 0.003
	KNN	0.839 ± 0.004
	SimRank	0.832 ± 0.005
	SVD	0.924 ± 0.006
	MC	0.828 ± 0.005
	JSNM	**0.772** ± 0.003
RMSE	AF	1.062 ± 0.006
	KNN	1.045 ± 0.003
	SimRank	1.034 ± 0.003
	SVD	1.263 ± 0.012
	MC	1.024 ± 0.004
	JSNM	**0.963** ± 0.004

In our experiment, two movie rating data sets used are Netflix training set and MovieLens [76]. The Netflix rating data contains more than 10^8 ratings with values from 1 to 5, which are given by around 500,000 users on around 20,000 movies. The MovieLens rating data contains more than 10^7 ratings with values from 1 to 5 and scale 0.5. We construct the data set used in this experiment as follows: first we extract common movies titles which has at least 100 ratings in both data sets, after that we select the first 100 users each. Via this way, we get both matrices with size $1,381 \times 100$. Next, we randomly split available ratings into 20 parts. Each time preserving 1 part and masking all others[3], we do the prediction and evaluate the performance of all methods. We calculate the average MAE and RMSE for these 20 experiments. Such process is repeated 10 times to calculate the mean and standard deviation.

From Tables 3.4 and 3.5, we can observe that our method still outperforms other methods in terms of MAE and RMSE. Meanwhile, the performance of all methods have

[3] for performance purpose, we do the row sampling based on movies and ensures each movie has a few available ratings

Table 3.4. Netflix Evaluation

Prediction Measure	Methods	Result
MAE	AF	0.942 ± 0.007
	KNN	0.928 ± 0.008
	SimRank	0.925 ± 0.008
	SVD	1.724 ± 0.014
	MC	0.983 ± 0.008
	JSNM	$\mathbf{0.903} \pm 0.005$
RMSE	AF	1.234 ± 0.005
	KNN	1.189 ± 0.005
	SimRank	1.164 ± 0.005
	SVD	1.924 ± 0.011
	MC	1.162 ± 0.006
	JSNM	$\mathbf{1.072} \pm 0.005$

decreased quite significantly compared with rating graph results in the previous subsection. One possible reason is that since these two matrices are now made up of common movies but different users, the ratings for any movie have more variability than the data in proceeding section. We would conduct more investigations in our future research for such type of data.

3.5.4 Social Network Regularization Effect Investigation

In this section, we look into the effect of manifold term in objective function Eq. (3.2). We compare the the performance of our framework (JSNM) with the objective function without the manifold terms. We call the new method Dual Graph Factorization (DGF). We still set $\lambda = 10^{-3}$ in JSNM and then repeat the same experiment procedure in the above subsections. We summarize the results in Table IV. In terms of trust prediction evaluation, JSNM improves DGF result 2-3% based on DGF result for trust link, 1% for distrust link. As to the rating evaluation, JSNM improves 3.7% and 4% on Epinion for both

Table 3.5. MovieLens Evaluation

Prediction Measure	Methods	Result
MAE	AF	0.802 ± 0.005
	KNN	0.812 ± 0.006
	SimRank	0.814 ± 0.006
	SVD	1.843 ± 0.007
	MC	1.045 ± 0.006
	JSNM	**0.764** ± 0.004
RMSE	AF	1.023 ± 0.004
	KNN	1.043 ± 0.004
	SimRank	1.037 ± 0.004
	SVD	2.046 ± 0.008
	MC	1.173 ± 0.005
	JSNM	**0.987** ± 0.003

Table 3.6. Manifold Term Investigation I

Link	Methods	AUC	F1
Trust	JSNM	**0.214** ± 0.003	**0.220** ± 0.002
	DGF	0.209 ± 0.003	0.213 ± 0.004
Distrust	JSNM	**0.992** ± 0.002	**0.992** ± 0.003
	DGF	0.982 ± 0.004	0.984 ± 0.003

MAE and RMSE, 1.2% and 1.5% on Netflix, 5% and 2% on MovieLens. From the table, we can conclude the social network regularity term plays a role in our framework.

3.6 Chapter 3 Conclusion

In this chapter, we developed the joint social network mining (JSNM) method to perform the trust prediction with the ancillary rating matrix. We transfer the common group structure knowledge between two related matrices and simultaneously explore the individual matrix geometric structure. With publicly available data sets, our method shows its advantage over classical trust prediction methods for both the trust matrix and rating

Table 3.7. Manifold Term Investigation II

Data	Methods	MAE	RMSE
Epinion	JSNM	**0.772** $\pm$ 0.004	**0.963** $\pm$ 0.005
	DGF	0.802 $\pm$ 0.011	1.004 $\pm$ 0.007
Netflix	JSNM	**0.904** $\pm$ 0.004	**1.074** $\pm$ 0.006
	DGF	0.918 $\pm$ 0.009	1.136 $\pm$ 0.011
MovieLens	JSNM	**0.766** $\pm$ 0.005	**0.987** $\pm$ 0.004
	DGF	0.783 $\pm$ 0.009	1.012 $\pm$ 0.012

matrix. Furthermore, our method can be also applied to homogeneous type data and yield similar improvement in the prediction. Matrix factorization has been also applied to our other work [79, 80, 81]

Although most web sites do not have (publish) both trust graph and rating graph data sets, we believe our method provides many web sites a new perspective to improve their service. Taking amazon.com and facebook.com for example, users may consent to information sharing between these two sites, as their friends lists and purchase histories generally cause no severe privacy leakage. Amazon may recommend users items their friends purchased so that boost their sale, on the other hand, facebook users could have the opportunity to link to other users and make new friends, who purchased similar topic of books, style of music and demonstrated same interest. In the future work, we will investigate the effectiveness of our framework applying to more general related dual graphs.

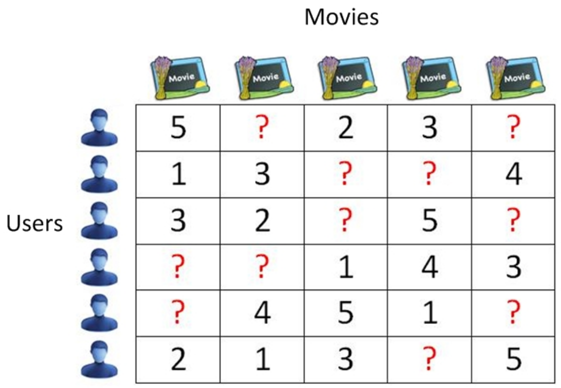
CHAPTER 4

Future Events Recommendation Via Collaborative Ranking

4.1 Chapter 4 Introduction

This chapter presents the model for future events recommendation, it is slightly different from conventional recommendation system. The general recommendation system setting consists of a collection of users and items, where user feedbacks are available for different subsets of items. Figure 4.1 is a user-movie mini recommendation system demonstration. Here users give ratings to movies they have watched. Generally speaking, each individual user only rate a small portion of movies in the whole collection. Therefore, such a user-movie matrix generally contains a large amount of missing values. The goal is to predict ratings for the remaining items users have yet to experience, based on limited rating history. One important note is that these missing values are generally assumed to be distributed randomly.

Recommendation systems can be classified into two groups based on their techniques: content-based system and collaborative filtering system. Content-based approaches examine features of items recommended and relate user preferences to those features. Collaborative filtering methods, in contrast, recommend items based on similarity measures between users and items and fill-in ratings for the remaining items. They often formulate this problem as a matrix completion problem. Matrix factorization is an important category of collaborative filtering algorithms, the rationale behind these methods in this category is that the preferences of a user are determined by a small number of unobserved factors. We will give a brief review over these methods in related work section. The content-based and collaborative filtering approaches are often complementary.



5	?	2	3	?
1	3	?	?	4
3	2	?	5	?
?	?	1	4	3
?	4	5	1	?
2	1	3	?	5

Figure 4.1. A demonstration of general recommendation system. Rows index users and columns index movies. The digits represent individual users' rating towards corresponding movies. The question marks represent missing values to be predicted. Note that missing values are generally assumed to be distributed randomly..

In this chapter, we consider a recommendation system in which the information items are events. Here, we predict users' preferences on future events and recommend users higher ranked ones that users are interested in. Traditionally, prediction and ranking are considered two orthogonal tasks and generally should not have much overlap. Most ranking problems deal with existing sample and sort according to pre-defined evaluation criteria. However, The current scenario is to rank events users not yet responded based on their historical preference. Therefore, it is necessary to conduct prediction prior to the ranking. This is a relatively new area in both information retrieval and recommendation system but with various applications. Such prediction can guide resource management and identify potential time conflict events. The following is a few concrete examples.

Example 1. Many deal websites (such as groupon and living social) are interested in learning more about targeted customers. Providing accurate personalized deal recommendation can both boost the sale revenue and reduce the advertising cost. These websites have records of consumers' detailed purchase history. Taking advantage of big data analysis technique and accurate events recommendation system, these vendors can provide their interested discount gift certificate and vacation packages they need in the future.

Example 2. Terrorist attacks preventing is now a global priority. Many actions have been taken to enhance security. One of the important measures is to understand evolving and emerging threats. With information-sharing partnerships, many countries make efforts to keep track of the list of terror suspects and their past activities. With the help of a smart future event prediction system, it is possible to predict terrorists' targets and next step plans based on their current locations and past behavior patterns. The corresponding strategies can be well planned and the loss can be reduced.

Example 3. One of the most important features for social web sites (e.g, Facebook and Twitter) has been the ability to create groups, allowing members to focus on following and contacting different sets of people. Group members can invite whole groups to some particular events. Those events, on the other hand, attract more members join the group and expand the social network. Predicting which events users are interested in and attracting new users to join have been the goals for those social web sites. Besides, future events recommendation has close connections with social network analysis, social behavior study and psychology research etc. The better understanding in this area helps the advancement of other disciplines.

The applications of future events recommendation are clearly beyond the above three examples and have significance in industry, government and academia. The main contributions of our paper include:

- We present a method that combines the content-based approach and collaborative filtering approach. Our method has the content-based element, since we recommend events to users based on user's past ratings and the implicit correlation between past events and future events. Moreover, our framework takes a global collaborative filtering perspective and investigates into the latent factors that determine users' preferences.
- We design a transductive framework. Our objective function successfully incorporates the future events information into our learning process. This helps our system yields accurate ranking results when training instances are scarce.
- We propose an objective function that has clear motivation and interpretation for each term. Beside this, we provide a concise optimization algorithm and show our solution is the global optimal solution to our objective function.
- We conduct empirical experiments to demonstrate the effectiveness of our method, comprehensive experiments and analysis include MAP comparison against classical methods, parameter tuning investigation for our method, MAP histogram for individual users.

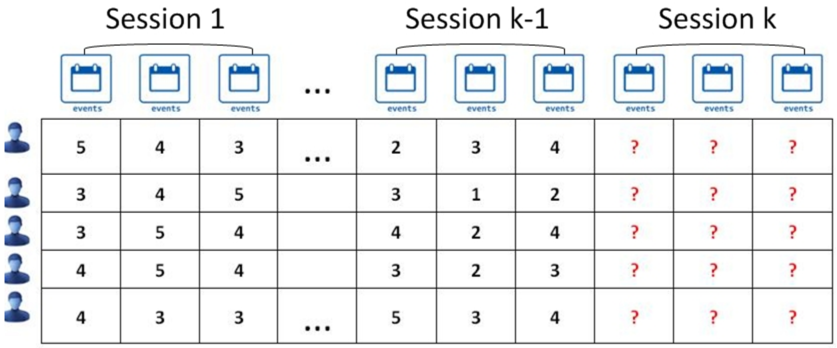
The paper proceeds as follows. Section 4.2 formulates the event recommendation as a ranking problem and provides all necessary notations. Section 4.3 gives a brief review about related work in recommendation system, especially in event recommendation. Section 4.4, Section 4.5 and Section 4.6 present our objective function, the algorithm and the mathematical proof, respectively. Empirical experiments are conducted in Section 4.7. We conclude our paper in Section 4.8.

4.2 Event Recommendation

In this section, we formulate the ranking problem and provide necessary annotation for the following context. Let us assume there are n individual users and each user is present in total m events in $k-1$ different sessions. In each session, user give preferences to events happened within the specified time frame. Figure 4.2 is a good demonstration of such system [1]. Now given a few future events, our task is to predict users' preference to those events and recommend events that are of high probabilities to be interested in.

For the convenience of discussion in the following context, we formulate this problem in matrix setting. We index users as rows and events as columns in matrix M. Without loss of generality, we sort events according to the order of session occurrence and therefore events in the same session are grouped together. The matrix entry value M_{ij} represents the coded preference of user i to event j. According to the above setting, those future events are expected at the end of the matrix and don't carry value in the corresponding columns. Figure 4.2 is a demonstration of our setting. The most significant difference between our events recommendation system and the conventional recommend system (as shown in Figure 4.1) is that, due to the nature of the problem, our matrix contains multiple columns without any feedback. In conventional recommendation [23, 36, 82], the missing values are often assumed to be distributed randomly. In other words, it would be rare for missing elements to be cluttered column-wise like Figure 4.2 does. This is the most significant difference between conventional recommendation system and our future events recommendation system. It is inappropriate to rearrange those future events columns due to the occurrence order, also it is futile for prediction purpose since these columns are still blank. Content-based methods alone generally are not applicable to our system since it provides no feedback reference at all for those future events. Collaborative filtering methods, on the

[1] In this paper, we focus on the scenario that there is no missing feedback for the past events for simplicity, because missing value handling will divert our discussion.



Session 1			…	Session k-1			Session k		
events	events	events		events	events	events	events	events	events
5	4	3	…	2	3	4	?	?	?
3	4	5		3	1	2	?	?	?
3	5	4		4	2	4	?	?	?
4	5	4		3	2	3	?	?	?
4	3	3	…	5	3	4	?	?	?

Figure 4.2. A demonstration of our future events recommendation system. Users and events are indexed by rows and columns, respectively. Events are arranged according to the order of occurrence and grouped into k sessions. Users give their preference ratings regarding the past events. Columns in the k-th session are future events and their ratings are unknown..

hand, focus more on global structure and therefore have the potential to handle such kind of sparsity. We will give a brief review over literature work in recommendation systems in the next section, especially collaborative filtering methods and event recommendation related work.

4.3 Related Work

In this part, we provide a brief review regarding the literature related to our work. First, we mention the previous work in event recommendation. Next, we give a quick overview about collaborative filtering methods(especially in low-rank approximation). Last, we introduce the recently popular collaborative filtering method-trace norm minimization, it is an important component of our framework and therefore we provide the necessary background.

Our future event recommendation system clearly has close connection with general recommendation systems. More specifically, our approach belongs to personalized recommendation-recommend things based on the individual's past behavior. There are many classic works in terms of generic prediction [83, 14, 76]. However, the problem of collaborative ranking of future events recommendation has not received well attention. There are 2 recent papers according to our knowledge. The first one [84] proposes a fuzzy relational framework, which is a hybrid of content and collaborative filtering approach, to recommend future events if they are similar to past events that similar users have liked. The major drawback of this paper is the approach was not evaluated empirically. The second one [85] presents a collaborative method based on matrix factorization, the rationale is similar to the first one while the setting is more like our recommendation system.

Now we talk about collaborative filtering methods. In [86], Xu et al. further classified collaborative filtering methods into 3 categories: memory-based, model-based and hybrid recommenders. In this chapter, we focus on low-rank approximation related methods [19, 52, 54], which belong to model-based category. The rationale behind these factor model methods is that there are usually a small number of latent factors influencing the preferences, and a user's preference is determined by how each factor applies to that user. This assumption applies here since in practice user's interest in particular events is often influenced by relatively stable past behavior pattern.

Last, we present a brief introduction to trace norm minimization method, a recent popular framework for matrix completion. Candés et al. [23, 36, 82] use the trace norm of user-item matrix as a convex relaxation of its rank, to seek a low-rank matrix X to approximate the original matrix and therefore make recommendation. Specifically, to optimize the following equation

$$\min_{X} \|X_\Omega - M_\Omega\|_F^2 + \gamma \|X\|_* , \tag{4.1}$$

where Ω is the set of available entries and $\|X\|_*$ denotes the trace norm of X, which is the sum of singular values of X. $\gamma > 0$ is the regularity parameter balancing the observations fit and the rank of matrix X. Note that such yielded X is the global optimal solution and not subject to the influence of initialization, due to the convex property of the objective function. There is an important conclusion in [23]: for most $n \times n$ matrices, if the number of uniformly sampled entries is no less than $Cn^{1.2}r \log n$, where C is a positive constant and r (not too large) is the rank of matrix, then there is a high probability that such matrices can be perfectly recovered. Moreover, the conclusion can hold for any matrix if 1.2 is replaced by 1.25 [23].

However, there are two potential issues with this method. First, the incoherence conditions of the data matrix is often too restrictive, there is no prediction accuracy guarantee when the assumption is not satisfied. Second, the theoretical results in [23, 82] assume that the observed entries are sampled uniformly at random. Unfortunately, many real-world data sets exhibit power-law distributed samples instead [25]. In terms of our events recommendation system, as demonstrated in Fig.(4.2), all unknown values are condensed into a few columns. Due to the above two reasons, trace norm minimization method *alone* is not a good choice for events recommendation.

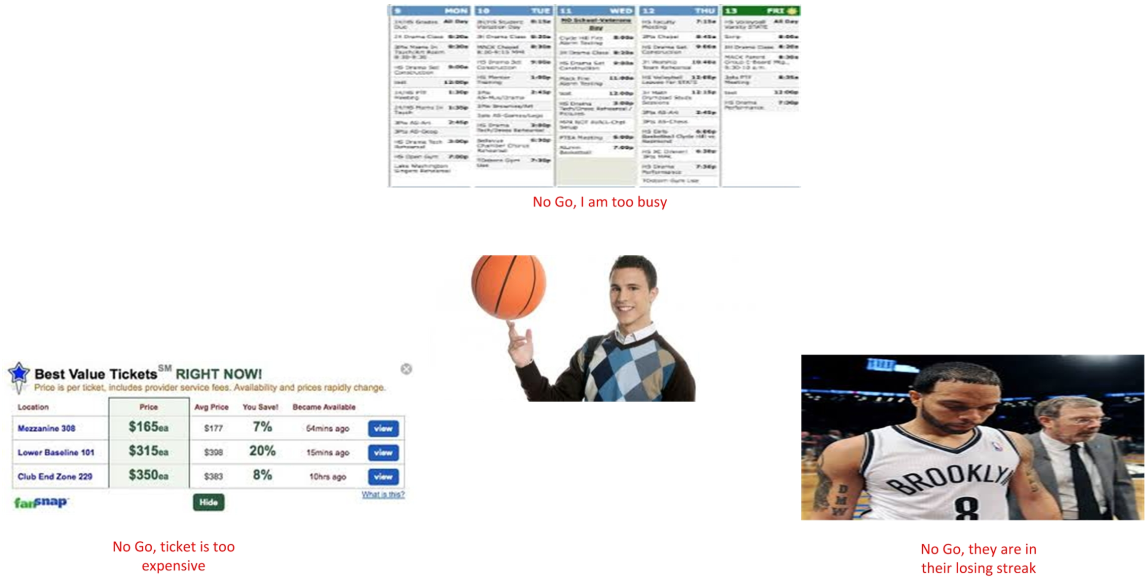
4.4 Collaborative Ranking Framework

In this section, we present our motivation first and then propose the corresponding objective function. As mentioned in the introduction part, we try to find a matrix that best approximates the user-event matrix and therefore recommend highly ranked events to individual users. Similar to other general recommendation systems, our framework also assumes the preference of users to these events are based on a few latent factors. Based on this, we seek a low-rank matrix X, where each x_i (individual row of X) consist of two parts. The first part is to approximate each individual user's past events, the second part is for the prediction purpose. Specifically, let us assume the past events matrix is $M \in \mathbb{R}^{n \times m}$, here n is the number of users, m is the total number of past events. In the following context, when we refer to $\mathbf{m}_i$, it is the events activity record vector for individual user i. We are generating vectors in the form of $\mathbf{x}_i = [\hat{\mathbf{x}}_i, \widetilde{\mathbf{x}}_i]$, here $\hat{\mathbf{x}}_i$ corresponds to $\mathbf{m}_i$ and $\widetilde{\mathbf{x}}_i$ predicts how users will respond to those future events. As introduced in the proceeding part, the trace norm is a convex approximation of matrix rank and therefore becomes an important component of our objective function. In addition to this, as we are predicting future events based on users' past activity history, our prediction matrix should approximate these available entries in the original matrix, so it is natural to come up with the following tentative objective function:

$$\sum_{i=1}^{n} \|\hat{\mathbf{x}}_i - \mathbf{m}_i\|_2^2 + \gamma \|X\|_* . \tag{4.2}$$

Here $X = [\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_n^T]^T$ and $\gamma > 0$ is the regularity parameter. Note that the trace norm regularity term $\|X\|_*$ includes users' responses to past events and the predicted responses to future events, instead of past events alone. After all, we expect individual user's behavior pattern should be consistent and his preference on future events can be inferred from past events.

While Eq. (4.2) takes advantage of the convex property of trace norm and the latent feature of the user-event matrix, it fails to *explicitly* incorporate the correlation of the events. Although the available entries(past events activity) could also provide the events correlation information to some extent, these entries are generally subject to constraints beyond users' preference. As demonstrated in Figure 4.3, a basketball fan may decide not to attend a live game due to schedule conflict, expensive admission ticket or even recent bad performance of his favorite team, which is contrary to his usual behavior pattern. Therefore, we decide to include the events correlation information in our objective function to enhance prediction power, which will be presented in details in next part.



Best Value Tickets^{SM} RIGHT NOW!
Price is per ticket, includes provider service fees. Availability and prices rapidly change.

Location	Price	Avg Price	You Save!	Became Available	
Mezzanine 308	$165ea	$177	7%	64mins ago	view
Lower Baseline 101	$315ea	$398	20%	15mins ago	view
Club End Zone 229	$350ea	$383	8%	10hrs ago	view

No Go, I am too busy
farsnap
Hide
What is this?
No Go, ticket is too expensive
BROOKLYN
8
No Go, they are in their losing streak

Figure 4.3. A demonstration for individual user choose not to attend a particular event due to various reasons. It could be due to his busy schedule, budget constraint or unexpected reason. This is to illustrate the point individual user might not follow his usual behavior pattern. Therefore, simply relying on historical data to make inference is not reliable for practical application..

4.4.1 Events Correlation Capture

This part is to provide a derivation of the events correlation term, which will be included in our final objective function. Let us assume we have defined an appropriate evaluation metric for events similarity (the specific measure is dependent upon the events domain knowledge and we will discuss in the experiment section), get the similarity matrix W and the similarity between event i and event j is w_{ij}. Recall that our prediction vector for individual k is $\mathbf{x}_k$, we use $\mathbf{x}_{k,i}$ to denote his preference for event i. Then our intuition is that for individual k, the difference between $\mathbf{x}_{k,i}$ and $\mathbf{x}_{k,j}$ should be small given the similarity between event i and event j is high. To model such relation, we propose to minimize the following term: $\min\limits_{\mathbf{x}_k} \frac{1}{2} \sum\limits_{i,j=1}^{m} w_{ij} \left(x_{k,i} - x_{k,j}\right)^2$, the reason of putting $\frac{1}{2}$ here will become clear shortly. Notice that

$$\begin{aligned}
&\tfrac{1}{2} \sum_{i,j=1}^{m} w_{ij} \left(x_{k,i} - x_{k,j}\right)^2 \\
&= \tfrac{1}{2} \left(\sum_{i=1}^{m} d_i x_{k,i}^2 - 2 \sum_{i,j=1}^{m} x_{k,i} x_{k,j} w_{ij} + \sum_{j=1}^{m} d_j x_{k,j}^2 \right) \\
&= \sum_{i=1}^{m} d_i x_{k,i}^2 - \sum_{i,j=1}^{m} x_{k,i} x_{k,j} w_{ij} \\
&= \sum_{k=1}^{m} \mathbf{x}_k (D - W) \mathbf{x}_k^T
\end{aligned} \tag{4.3}$$

where $d_i = \sum\limits_{j=1}^{n} w_{ij}$, is the diagonal elements of diagonal matrix D. Note that $L = D - W$ is called the Laplacian matrix in literature and plays an central role in spectral clustering [87, 88]. Since W is symmetric, it is also easy to see L is symmetric and positive definite.

Clearly we need to minimize the final term in Eq. (4.3), this is consistent with Eq. (4.2). With the derivation in this part, we are ready to present our final objective function.

4.4.2 Collaborative Ranking Objective Function

With the discussion in proceeding parts, we propose the objective function as follows:

$$\min_{\mathbf{x}_i} \sum_{i=1}^{n} \mathbf{x}_i L \mathbf{x}_i^T + \alpha \sum_{i=1}^{n} \left\| \hat{\mathbf{x}}_i - \mathbf{m}_i \right\|_2^2 + \gamma \left\| X \right\|_* . \tag{4.4}$$

The first term is the summarization of n events correlation terms, as derived in Eq. (4.3). The second part of the equation is identical to Eq. (4.2), except the positive parameter α, which is to ensure the our predictor vectors fit past events well. Here parameters $\alpha > 0$ and $\gamma > 0$ also ensure the solution $\mathbf{x}_i$ is non-trivial. Otherwise, Eq. (4.4) reduces to the summarization of n independent Eq. (4.3), it is clear all prediction vectors can be 0. The first term captures the events correlation from a local perspective, while the trace norm term seeks a global structure approximation to the user-event matrix.

Technical speaking, our framework falls within the category of transductive learning [89], as here we included the feature of future events (correlation with past events) in our learning process, this is the most significant difference between our work and prior work in event recommendation. Here is an example to illustrate the difference between transductive learning and inductive learning to facilitate the understanding. Assuming there are a few training instances and test instances, as well as labels for these training instances, we are to classify test instances with Support Vector Machine (SVM). Inductive learning uses training instances *exclusively* to learn the classification margin while all test instances also play a role in learning the margin for transductive learning. It has been shown that transductive learning generally yields better learning result than inductive learning [90, 91], because it incorporates the information of test instances.

4.5 Optimization Algorithm

So far we have presented the objective function and the interpretation of each term. This section, we will present the optimization algorithm to Eq. (4.4). First of all, due to the trace norm property and the fact the trace over a scalar is still itself, it can be converted into the following one

$$\min_{\mathbf{x}_i} \sum_{i=1}^{n} \mathbf{x}_i L \mathbf{x}_i^T + \alpha \sum_{i=1}^{n} \|\hat{\mathbf{x}}_i - \mathbf{m}_i\|_2^2 + \gamma \sum_{i=1}^{n} \mathbf{x}_i D \mathbf{x}_i^T, \tag{4.5}$$

where $D = \frac{1}{2}(X^T X)^{-\frac{1}{2}}$.

The new objective function is the summarization of n independent vectors and can be solved in a decoupled manner. Therefore, in the following context, we will minimize Eq. (4.6) with respect to individual $\mathbf{x}_i$. To make the optimization more clear, we combine L and D together and get the following equation:

$$\min_{\hat{X}, \widetilde{X}} Tr([\hat{X}, \widetilde{X}] \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix} [\hat{X}, \widetilde{X}]^T) + \alpha \left\|\hat{X} - M\right\|_F^2 \tag{4.6}$$

Here Tr is the trace operation, $\hat{X} = [\hat{\mathbf{x}}_1^T, \ldots, \hat{\mathbf{x}}_n^T]^T$, $\widetilde{X} = [\widetilde{\mathbf{x}}_{\mathbf{1}}^T, \ldots, \widetilde{\mathbf{x}}_n^T]^T$ $N = L + \gamma D = \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix}$, the sizes of blocks within N are in accordance with $\hat{X}$ and $\widetilde{X}$. Note that here $N_{12} = N_{21}^T$.

Eq. (4.6) can be written into the following equation:

$$\begin{aligned} &\min_{\hat{X},\widetilde{X}} Tr(\hat{X} N_{11} \hat{X}^T) + 2\, Tr(\hat{X} N_{12} \widetilde{X}^T) \\ &+ Tr(\widetilde{X} N_{22} \widetilde{X}^T) + \alpha Tr((\hat{X} - M)(\hat{X} - M)^T) \end{aligned} \tag{4.7}$$

To get the solution of $\hat{X}$ and $\widetilde{X}$ to the above objective function, we take derivative with respect to them and set to zero respectively.

$$\begin{cases} \hat{X} N_{11} + \widetilde{X} N_{21} + \alpha \hat{X} - \alpha M = 0 \\ \widetilde{X} N_{22} + \hat{X} N_{12} = 0 \end{cases} \tag{4.8}$$

We get the solution of $\hat{X}$ and $\widetilde{X}$ respectively.

$$\begin{cases} \hat{X} = \alpha((N_{11} - N_{12} N_{22}^{-1} N_{21} + \alpha I)^{-1})^T M^T \\ \widetilde{X} = -\hat{X} N_{12} N_{22}^{-1} \end{cases} \tag{4.9}$$

It is clear that our objective function involves multiple variables. In order to minimize Eq. (4.4), we implement an alternative optimization algorithm. In other words, in each iteration we optimize one variable while fixing other variables, such process repeats until convergence. One subtle point is that once we get updated $\hat{X}$ and $\widetilde{X}$, i.e, updated X, we need to update D accordingly. Our objective function decreases at each iteration and naturally lower bounded. What is more, it is a convex function and therefore we are guaranteed to yield a global optimum. The rigid theoretical convergence proof will be presented in next section.

Algorithm 3 is a summarization of our optimization process. The convergence criterion here is the relative change of the objective function falls below 10^{-4}. Empirical experiments show our algorithm converges within 30 iterations, for data sets used in this paper. We call our framework Collaborative Ranking (CR). Here the term ”collaborative” is twofold: first, it indicates users and events deliver information in a cooperate way to help the prediction. second, training instances and test instances work together to establish the model.

Algorithm 3: Collaborative Ranking Algorithm

```
Input: Events Laplacian matrix L, user events preference vector $\mathbf{m}_i$s,
       parameter $\alpha$ and $\gamma$.
Output: User events preference prediction matrix $\widetilde{X}$
Initialize $D = I$
while Not Converged do
    Step 1: $N = L + \gamma D$
    Step 2: $\hat{X} = \alpha((N_{11} - N_{12}N_{22}^{-1}N_{21} + \alpha I)^{-1})^T M^T$
    Step 3: $\widetilde{X} = -\hat{X}N_{12}N_{22}^{-1}$
    Step 4: $D = \frac{1}{2}(X^T X)^{-\frac{1}{2}}$
end
```

We summarize the highlights of our theoretical contributions here.

- Our algorithm has the pairwise events similarity term, it enables the content-based recommendation based on past events relevance with future ones. On the other hand, the trace norm captures the users’ pattern from a global view. Our hybrid system combines both prospectives in a novel way.
- Instead of generating explicit ranking rule with the training events, we incorporate the test events feature into the transductive learning framework and directly produce the ranking result without intermediate result.

- Our convex objective function has clear motivation and interpretation, yields global optimal solution for the alternative optimization algorithm.

4.6 Convergence Proof

In this section, we want to show that our algorithm guarantees the objective function Eq. (4.6) decreases at both steps. Since our objective function is a convex one, this implies our converged solution is a global optimal one and not subject to the initialization. In the remainder of this section, we will prove the above claim.

Lemma 3 *Assuming two matrices A and B, we have the following inequality:*

$$\frac{1}{2}Tr(AB^{-\frac{1}{2}}) - Tr(A^{\frac{1}{2}}) \geq \frac{1}{2}Tr(BB^{-\frac{1}{2}}) - Tr(B^{\frac{1}{2}}) \tag{4.10}$$

Proof 5 *Interested readers please refer to the proof of Lemma 3 in the appendix section.*

Theorem 4 *The objective function value in each iteration will decrease monotonically, according to steps in Algorithm 3.*

Proof 6 *Let us denote the solution of X at t-th iteration $X^{(t)}$ and the corresponding solution D by $D^{(t)}$.*

According to Step 1 and Step 2 in Algorithm 3, we have

$$X^{(t+1)} = \arg\min_{X} Tr(XNX^T) + \alpha \left\| \hat{X}^{(t)} - M \right\|_F^2 \tag{4.11}$$

This is equivalent to

$$\begin{aligned} & Tr(X^{(t+1)}N^{(t)}X^{(t+1)^T}) + \alpha \left\| \hat{X}^{(t+1)} - M \right\|_F^2 \\ & \leq Tr(X^{(t)}N^{(t)}X^{(t)^T}) + \alpha \left\| \hat{X}^{(t)} - M \right\|_F^2 \end{aligned} \tag{4.12}$$

Due to the fact $N^{(t)} = L + \gamma D^{(t)}$ and $D = \frac{1}{2}(X^{(t)^T}X^{(t)})^{-\frac{1}{2}}$, we substitute them into the above equation and get

$$\begin{aligned} & Tr(X^{(t+1)}LX^{(t+1)^T}) + \alpha \left\| \hat{X}^{(t+1)} - M \right\|_F^2 \\ & + \tfrac{\gamma}{2}Tr(X^{(t+1)}(X^{(t)^T}X^{(t)})^{-\frac{1}{2}}X^{(t+1)^T}) \le Tr(X^{(t)}LX^{(t)^T}) \\ & + \alpha \left\| \hat{X}^{(t)} - M \right\|_F^2 + \tfrac{\gamma}{2}Tr(X^{(t)}(X^{(t)^T}X^{(t)})^{-\frac{1}{2}}X^{(t)^T}) \end{aligned} \tag{4.13}$$

Now according to Lemma 3, we have

$$\begin{aligned} & \gamma Tr(X^{(t+1)^T}X^{(t+1)})^{\frac{1}{2}} - \tfrac{\gamma}{2}Tr(X^{(t+1)}(X^{(t)^T}X^{(t)})^{-\frac{1}{2}}X^{(t+1)^T}) \\ & \le \gamma Tr(X^{(t)^T}X^{(t)})^{\frac{1}{2}} - \tfrac{\gamma}{2}Tr(X^{(t)}(X^{(t)^T}X^{(t)})^{-\frac{1}{2}}X^{(t)^T}) \end{aligned} \tag{4.14}$$

Combining the above two inequalities, we get

$$\begin{aligned} & Tr(X^{(t+1)}LX^{(t+1)^T}) + \alpha \left\| \hat{X}^{(t+1)} - M \right\|_F^2 \\ & + \gamma Tr(X^{(t+1)^T}X^{(t+1)})^{\frac{1}{2}} \le Tr(X^{(t)}LX^{(t)^T}) \\ & + \alpha \left\| \hat{X}^{(t)} - M \right\|_F^2 + \gamma Tr(X^{(t)^T}X^{(t)})^{\frac{1}{2}} \end{aligned} \tag{4.15}$$

From the above inequality, we know the proposed theorem holds.

It is clear that our objective function value is lower bounded by 0, as a result, our algorithm will converge to a global optimum.

4.7 Experiments

As to this section, we have completed the theoretical parts of our work. In this section, we conduct empirical experiments on two public data sets and demonstrate the effectiveness of our method. This section consists of several parts. The first part is an introduction of the data sets. The second part talks about event features and measure metrics. The third part presents the experiments setup and main results. At last, we launch other related discussion.

4.7.1 Data Set Descriptions

The data set is about upcoming seminar preferences ratings by CS graduate students from MIT and CMU[2]. We call this data set CSAIL. Among consecutive 15 weeks experiments, participating computer science graduate students at MIT and CMU receive weekly digest emails about the seminars in next week. In each week, users receive a list of seminar titles, then users could further read the more detailed announcement after click the title he was interested in. An example of seminar announcement is displayed in Figure 4.4. Users are generally required to select at least one relevant talk to attend. If users really have no interest in any talk, they select the option of attending none (”forced option”). However, it is assumed that such ranking still carries information about relative ranking of the alternatives. After users made the selection, users can no longer modify their choices.

Table 4.1 shows relevant weekly statistics about seminar announcements presented and the associated user preference record. It can be observed the talk frequency varies widely throughout experiment period, ranging from 2 to 21 talks on a given week. It also includes the average number of talks judged relevant during a given week and the number of derived preference pairs. In total, 8.6% responses were marked ”forced”, which indicated no relevant talk was given in the week. There is an important observation, each student chose about 2 seminars, even when there were more seminars being offered. This might be due to students' schedule, otherwise it is very likely there should be a strong correlation between total number of students participated and the number of seminars offered that week. This is a good indication that preference ratings from users are subject to external factors besides latent factors, as we have mentioned in the introduction part.

The second data set is about the event recommendation competition at kaggle.com[3]. We name this data set KAGGLE. This website asks all contest participants to predict what

[2] http://mis.haifa.ac.il/ einatm?id=326

[3] http://www.kaggle.com/c/event-recommendation-engine-challenge/data

events its users will be interested in, based on events they have responded to in the past. For all events, users can choose either interested in or not interested in. This data set contains a lot of users' information, such as gender, demographic information and even their online friends' list. For the sake of simplifying experiments, we only preserve the columns directly related to events. There are 101 such columns, and they are processed as follows. First, kaggle determined the 100 most common word stems (obtained via Porter Stemming) occuring in the name or description of a large random subset of its events. The last 101 columns are count1, count2, $\cdots$, count100, count101, where countN is an integer representing the number of times the N-th most common word stem appears in the name or description of this event. count101 is a count of the rest of the words whose stem wasn't one of the 100 most common stems.

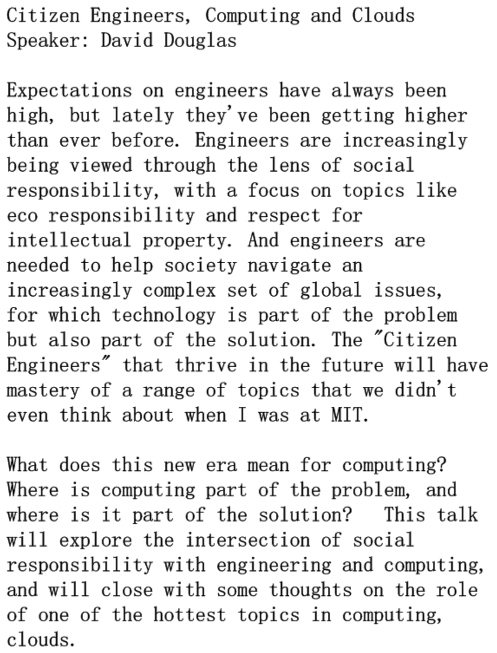
Based on the characteristic of both data sets, we use 1 to denote users will attend the corresponding seminar or is interested in the particular event, 0 for otherwise. Clearly, the user-event item matrices from both data sets are binary.

4.7.2 Events Relevance And Evaluation Metric

Since the second data set contains merely top frequent words, event features for 2nd data set have to be based on those isolated words. Our subsequent discussions focus on the first data set. As mentioned in the proceeding parts, events relevance is an important part of our objective function. Clearly, we need to calculate the relevance between seminars based on the announcements. As observed from Figure 4.4, each email can be viewed as a document. There are many literature works discussing document relevance [92, 93]. In this paper, we consider 2 methods to extract event features. One method is called term-frequency inverse document frequency (TF-IDF) [94, 95] and the other is Latent Dirichlet Allocation (LDA) [96].

Table 4.1. Data Statistics for CSAIL

Data set	Week	Talks	Relevant	Pairs
MIT	1	8	2.0(0.8)	11.2(3.3)
	2	8	1.9(0.9)	10.9(3.5)
	3	7	1.4(0.7)	7.2(2.1)
	4	3	1.3(0.5)	2.0(0)
	5	20	2.7(1.7)	43.7(22.7)
	6	12	1.8(1.1)	17.0(7.7)
	7	5	1.1(0.3)	4.1(0.5)
	8	12	1.9(1.0)	18.2(7.3)
	9	21	2.3(2.0)	39.7(26.2)
	10	17	2.4(1.4)	33.1(14.3)
	11	7	1.9(1.0)	8.8(2.4)
	12	7	1.7(1.0)	8.0(2.4)
	13	5	1.2(0.6)	4.3(0.7)
	14	21	2.7(1.8)	45.6(23.9)
	15	5	1.2(0.5)	4.4(0.8)
CMU	1	11	2.5(1.3)	19.5(7.2)
	2	8	1.4(0.6)	8.6(2.8)
	3	7	1.6(0.7)	8.2(2.4)
	4	11	1.4(0.8)	12.9(5.1)
	5	11	1.6(0.8)	14.3(5.6)
	6	11	1.7(1.2)	14.5(6.6)
	7	11	1.8(1.2)	15.0(6.5)
	8	2	1.0(0.1)	1.0(0.1)
	9	14	2.3(1.6)	14.9(12.0)
	10	13	1.7(1.1)	17.7(7.9)
	11	11	1.9(1.2)	15.7(7.0)
	12	7	1.2(0.7)	6.4(1.2)
	13	17	2.0(1.5)	27.6(15.1)
	14	12	1.4(1.0)	14.1(6.0)
	15	17	2.6(2.5)	31.2(15.9)



Citizen Engineers, Computing and Clouds
Speaker: David Douglas

Expectations on engineers have always been high, but lately they've been getting higher than ever before. Engineers are increasingly being viewed through the lens of social responsibility, with a focus on topics like eco responsibility and respect for intellectual property. And engineers are needed to help society navigate an increasingly complex set of global issues, for which technology is part of the problem but also part of the solution. The ″Citizen Engineers″ that thrive in the future will have mastery of a range of topics that we didn't even think about when I was at MIT.

What does this new era mean for computing? Where is computing part of the problem, and where is it part of the solution?   This talk will explore the intersection of social responsibility with engineering and computing, and will close with some thoughts on the role of one of the hottest topics in computing, clouds.

Figure 4.4. An example of email seminar announcement. We use LDA model and bag of words model to extract the content feature..

TF-IDF is a numerical statistic which reflects how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining. The TF-IDF value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to alleviate the fact that some words are generally more common than others. Following the idea, we tune documents into fixed-length feature vectors such that each coordinate of feature vector correspond to a word occurrence with TF-IDF weighted word counts. The rationale here is that users' interest could be triggered by particular key words. As a result, the coordinates of feature vectors can carry valuable information. Since CSAIL consists of documents which are full of words, TF-IDF is applicable here. As to KAGGLE, since these feature columns are the frequencies of these top frequent words, we get the weights by simply normalizing each word frequency.

LDA is more sophisticated. LDA can identify topics from the announcement and therefore can extract potential better features. These CS graduate students may decide whether to attend those seminars based on how much overlap between their research topics and the focus of the talk. Such decision relies more on the topic instead of individual words. Therefore, LDA should be a more appropriate choice here. Learning an LDA model requires estimation of the associated parameters and we apply Gibbs sampling method [97] here. These parameters determine the generation of topic compositions and topic-dependent word distributions from the training documents.

In summary, we apply both TF-IDF and LDA on both data sets while TF-IDF only on the second data set. For features yielded by LDA, we calculate the events similarity using the ad-hoc technique proposed in [98]. On the other hand, we determine the events similarity using extended boolean method [99] for TF-IDF. Since events (documents) similarity is not the main focus of this paper, we are not discuss these two methods in detail, interested readers may refer to the above two papers.

Now we present the evaluation metric Mean Average Precision (MAP), which is widely accepted measure in information retrieval [100]. Define the precision at rank k, $prec(k)$, to be the precision of the correct entries up to rank k. The MAP is the average precision for each position that holds a correct entry:

$$MAP = \frac{1}{m} \sum_{k=1}^{m} prec(k) \tag{4.16}$$

A quick example will make it clear. Given a ranked list of 5 items, where the items at rank 2 and 4 are known to be correct. The MAP is then $\frac{1}{2}(\frac{1}{2} + \frac{1}{2}) = 0.5$.

4.7.3 Experiments Setup And Parameters

Overall, we divide each data set into train and test parts. For CSAIL, the train part includes first 10 weeks feedbacks and the test includes feedbacks from week 11 to week

15. We apply the models learned to generate a ranked list for every student and each week in the test set. For KAGGLE, we sample the first 3,000 users from original training data set and the first 2,000 as training, the rest 1,000 as test. Since we treat this ranking problem in recommend system setting, we recommend test events to each individual user according to descending order of the imputed value.

To evaluate our event evaluation system, we compare CR against RankSVM[4][101, 102] and LowRank [85]. In CR, we tune α and γ from the list of$\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$. For LowRank, we set the number of latent factors K in the list of $\{4, 6, 8, 10, 12\}$ and regularity coefficient C from 1 to 5. Regarding RankSVM, the trade-off coefficient between training error and margin is selected from$\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ and other parameters are set as default.

4.7.4 Main Results

With the specified train data, all methods mentioned above learn the models first and evaluate the recommendation performance with MAP. The MAP results on both data sets are summarized in Table 4.2. Here we report the optimal MAP result out of both TF-IDF and LDA in Table 4.2. The optimal MAP result for CR yields when $\alpha = 0.1$ and $\gamma = 0.01$. We would like to postpone the discussion of this table until the next experiment, when more results are revealed.

Our next experiment is to investigate the impact of train set size on performance. In a real recommendation system, users are dynamic and can join and drop out of the recommendation system at different points of time. In addition to this, users may only provide feedback to a subset of events due to lack of diligence or unknown reasons. For example, users could be bored with selecting a long list of potential interested events, he

[4] `http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html`

may just quickly go through the list and select a few items just by random. Our experiment is to simulate such case.

Out of 10 weeks' CSAIL data, n (between 1 and 10) weeks are randomly selected for training. For every value of n, we train recommendation models based on the sampled data in the described fashion. To reduce the bias, every experiment is repeated 10 times. The learned models are then evaluated using the test set (weeks 11-15) and we report the average performance.

The corresponding experiment is slightly different for KAGGLE. There is no explicit time order for those events. Therefore, we increase the number of users used in the training part, from 400 to 2,000 with increment of 400. Again, during the 10 times repeated experiments, we randomly select the specified number of users from the training set and test the models on the separate test set.

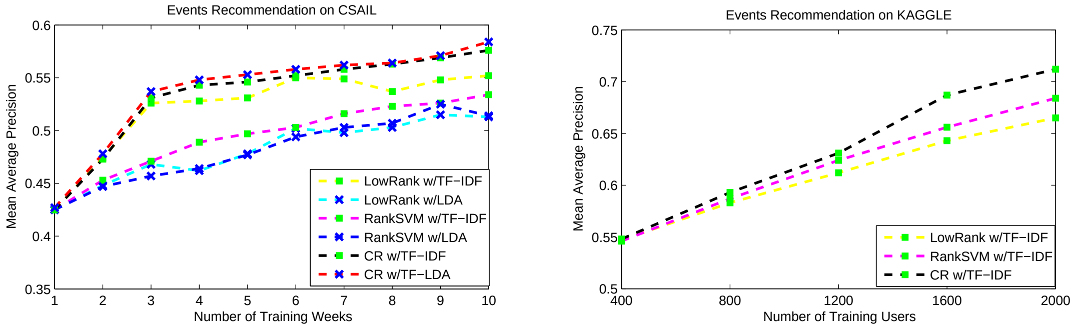
The experimental results are displayed in Figure 4.5. Figure 4.5 delivers more information than Table 4.2. There are a few observations and we would like to initiate the discussion here.

First of all, CR yields *consistently* better average MAP on both data sets, which demonstrates its effectiveness. As highlighted in the contribution summary, our hybrid system combines content-based approach and collaborative filtering approach in a natural way, as long as the parameters are well tuned. Such an idea originates from ensemble learning, a theory in statistics and machine learning. Ensemble learning theory says that better predictive performance could be obtained by ensemble methods, which use multiple models from the constituent models [103]. Content-based methods and collaborative filtering methods have both been proven to be effective for certain recommendation problems [104]. Therefore, there is no surprise CR model gets impressive performance.

Second, start from almost identical performance, all methods have a relatively stable performance after get sufficient number of past events feedbacks. In particular, CR yields

Table 4.2. Classification Methods Accuracy Comparison

Data Set	LowRank	RankSVM	CR
CSAIL	0.552	0.534	**0.584**
KAGGLE	0.665	0.684	**0.712**



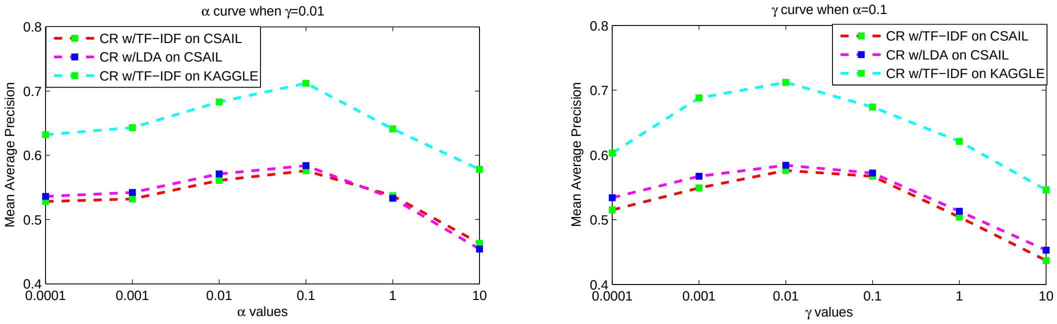
(a) CSAIL

(b) KAGGLE

Figure 4.5. A comparison of CR, LowRank and RankSVM with increasing training data. MAP is averaged across all users. It can be observed CR method yield consistently better performance..

impressive result once it gets 3 weeks training data, which most models never achieve even after all training samples are used. This is due to the transductive learning property of our framework, CR model extracts the useful information from the test samples. A close but not precise analogy is that CR uses in total 8 weeks data (3 training weeks with 5 test weeks) to train its model. What is more, although not an issue for current data set, user-item matrices generally contain vast number of missing values due to various reasons, which implies significantly fewer training feedbacks. CR model is especially useful for those cases.

Third, the differences between the choice of TF-IDF or LDA are moderate with respect to each recommendation model. Note that MAP of random ordering is 0.37 for C-SAIL and 0.42 for KAGGLE. Therefore, we can conclude that these two feature extraction methods should be effective. Is there any alternative feature extraction for better document categorization? We will look into this question in the future.



(a) α Curve when $\gamma = 0.01$

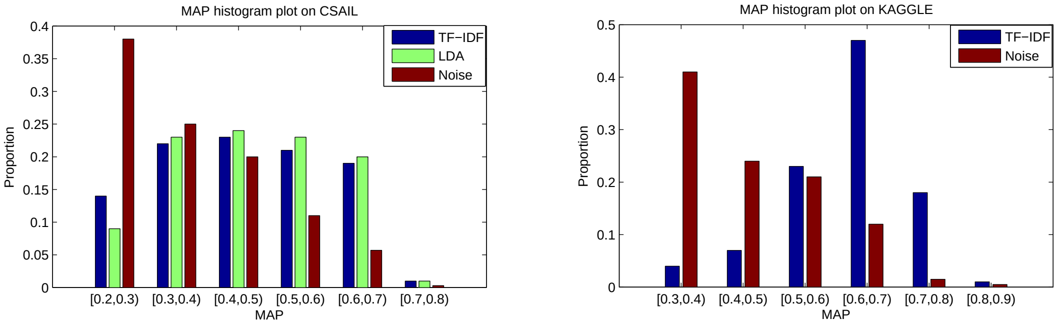
(b) γ Curve when $\alpha = 0.1$

Figure 4.6. α and γ curve plot when the other is fixed. MAP is averaged across all users. These two curves demonstrate how past events fit and global latent factor structure affect MAP..

4.7.5 Other Related Discussions

This part, we first look into influence of parameters in our framework. Recall that α controls the fit between prediction vector and past events vector, on the hand, γ ensures the low-rank of the user-event prediction matrix. The combination of α value and γ determine a trade off between the overall fit for past events and the emphasis of latent factors structure. Similar idea is also proposed in Eq. (4.1) and explained in Section **??**. As mentioned in the proceeding part, the optimal setting for CR framework is $\alpha = 0.1$, $\gamma = 0.01$. Therefore, we plot the curves of α and γ, when we fix one parameter and vary the other one. Figure 4.6 shows the curves when we conduct experiments using the full training data set. The MAP performance curves are stable as long as α and γ are in a reasonable range.

Now we want to check individual users' performance. The results so far all focus on all test users, it is also worth to look into variance among individual users' performance. We plot the histogram figures in Figure 4.7 for individual test users in both data sets. The results are in accordance with the previous conclusions. There are two subtle points here. First, as pointed out in Table 4.1 and data set description for KAGGLE, MAP can vary between users due to variance of weekly event set size or his active level. In general, it is expected accuracy will be higher for users who are interested in a large number of events,



(a) MAP histogram plot on CSAIL

(b) MAP histogram plot on KAGGLE

Figure 4.7. Histogram plot for individual user MAP. We also include noise as benchmark for comparison..

because large amounts of past events provide sufficient information to determine the latent areas/hobbies he is interested in. On the other hand, if users are interested in events with low correlations, then MAP results for those users are expected to be lower. In this case, due to schedule conflict or user limited time, data sparsity may become an issue for exploring the latent factors.

Last, we want to mention a few alternative models regarding these two data sets. Our task is to predict which events users are most likely to be interested in, our proposed method clearly belongs to ranking category. But it is possible to model it as classification or regression. We can classify events as interested or non-interested, or we can produce the probability users attending events with regression. We tried classic models such as random forest[5] [105], large scale SVM [106], logistic regression [107] etc. Although not listed in our experiment results, the results of these methods are not as good as ours. The main issues with these models is the data sparsity, which motivates the eventual choice of recommendation system as our baseline framework.

[5] http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

4.8 Chapter 4 Conclusion

In this chapter, we tackle the problem of recommending events for which no available feedback exists. We introduced a collaborative ranking system, whose performance surpasses existing algorithms. Since the direct feedbacks from future events are clearly not available, we have to utilize other related information from those test instances to train our learning function. As a result, we propose a transductive learning framework to incorporate the events Laplacian term. Our framework combines the content-based approach and collaborative filtering approach in a novel way. The shares some similarities with our work [108], where we incorporate the label for supervised learning. The convex objective function guarantees the global optimal property of our solution and is not subject to the influence of initialization of missing values. The empirical experiments on 2 real data set from various aspects demonstrate the effectiveness of our method.

There are two directions for our future research. First, we are interested in applying our learning framework to other related tasks. As mentioned in the introduction part, event recommendation can be applied to various scenarios. In particular, along with the evolution of social network web sites, accurate event recommendation can definitely help find users interested events and boost the online interactions between users. Second, in this paper, we assume there is no missing value in our training instances to simplify the model. For real event recommendation projects, collecting exhaustive data generally exceeds the budget and is often impossible for various reasons. Therefore, it is desirable to design a framework that tolerates such data sparsity. Our transductive learning can alleviate such issue, but a better training data learning model is still desirable.

CHAPTER 5

Conclusions And Future Work

Social network is a rich data source of large quantity and high variety. It is a field with many great and inspiring challenges for data mining. This dissertation proposes the concept of social information completion and presents many matrix completion frameworks to handle missing value prediction in large-scale social media networks. The key contributions of this work are summarized below, followed by future work.

5.1 Key Contributions

The following summarizes the main contributions used in our paper.

- We design a model that seeks a low-rank matrix and meanwhile maintains its discrete mode, this avoids the ad-hoc post-processing stage for conventional methods that work on continuous domain.
- We propose a transfer learning framework between social graph and rating matrix, this alleviates the data sparsity issue and significantly improve the prediction accuracies on both matrices.
- We create a new method that is applicable to future events prediction, such prediction differs from conventional methods that user-event matrix has continuous blocks of missing values.
- We suggest a novel targeted marketing model that maximizes the profit, such tree-based models work on segments instead of individuals.

5.2 Future Work

As mentioned in the introduction, the social web sites have generated enormous data of various forms and structures. While this dissertation proposes many different models in machine learning, they are only applicable to medium size social network. In order to extend these methods to real data sets, the parallel version utilizing Map-reduce [4] framework would be required and those algorithms need to be adjusted accordingly.

APPENDIX A

Convergence Proof Of Theorems In Chapter 3

In this appendix, we present the theoretical proof regarding the convergence of our algorithm in Chapter 3.

A.1 Proof to Theorem 1

Proof 7 *We rewrite Eq. (3.20) as*

$$\begin{aligned}J(\mathbf{V}_1) = Tr(\lambda\mathbf{V}_1^T\mathbf{L}_1^+\mathbf{V}_1 - \lambda\mathbf{V}_1^T\mathbf{L}_1^-\mathbf{V}_1 - 2\mathbf{G}_1^T\mathbf{U}^+\mathbf{V}_1\\ +2\mathbf{G}_1^T\mathbf{U}^-\mathbf{V}_1 + \boldsymbol{\alpha}^+\mathbf{V}_1^T\mathbf{V}_1 - \boldsymbol{\alpha}^-\mathbf{V}_1^T\mathbf{V}_1)\end{aligned}$$

According to Lemma 1, we have

$$\begin{aligned}Tr(\mathbf{V}_1^T\mathbf{L}_1^+\mathbf{V}_1) \leq \sum_{ij}\tfrac{(\mathbf{L}_1^+\mathbf{V}_1')_{ij}\mathbf{V}_{1,ij}^2}{\mathbf{V}_{1,ij}'}\\ \boldsymbol{\alpha}^+Tr(\mathbf{V}_1^T\mathbf{V}_1) \leq \boldsymbol{\alpha}^+\sum_{ij}\mathbf{V}_{1,ij}^2\end{aligned}$$

Meanwhile, by the inequality $x \leq \frac{(x^2+y^2)}{2y}, \forall y > 0$, *we have*

$$\begin{aligned}Tr(\mathbf{G}_1^T\mathbf{U}^-\mathbf{V}_1^T) = \sum_{i,j}(\mathbf{G}_1^T\mathbf{U}^-)_{ij}V_{1,ij}\\ \leq \sum_{i,j}(\mathbf{G}_1^T\mathbf{U}^-)_{ij}\tfrac{\mathbf{V}_{1,ij}^2+\mathbf{V}_{1,ij}'^2}{\mathbf{V}_{1,ij}'}\end{aligned}$$

On the other hand, to get the lower bound for the remaining terms, we employ the inequality $z \geq 1 + \log z, \forall z > 0$, *then*

$$\begin{aligned}Tr(\mathbf{G}_1^T\mathbf{U}^+\mathbf{V}_1^T) \geq \sum_{ij}(\mathbf{G}_1^T\mathbf{U}^+)\mathbf{V}_{1,ij}'(1+\log\tfrac{\mathbf{V}_{1,ij}}{\mathbf{V}_{1,ij}'})\\ Tr(\mathbf{V}_1^T\mathbf{L}_1^-\mathbf{V}_1) \geq \sum_{ijk}(\mathbf{L}_1^-)_{jk}\mathbf{V}_{ji}'\mathbf{V}_{ki}'(1+\log\tfrac{\mathbf{V}_{1,ji}\mathbf{V}_{1,ki}}{\mathbf{V}_{1,ji}'\mathbf{V}_{1,ki}'})\\ Tr(\mathbf{V}_1^T\mathbf{V}_1) \geq \sum_{ijk}\mathbf{V}_{1,ij}'\mathbf{V}_{1,ik}'(1+\log\tfrac{\mathbf{V}_{1,ij}\mathbf{V}_{1,ik}}{\mathbf{V}_{1,ij}'\mathbf{V}_{1,ik}'})\end{aligned}$$

By summing over all the bounds, we get $Z(\mathbf{V}_1, \mathbf{V}_1')$ *and it is easy to conclude that*

$$Z(\mathbf{V}_1, \mathbf{V}_1') \geq J(\mathbf{V}_1), \quad Z(\mathbf{V}_1, \mathbf{V}_1) = J(\mathbf{V}_1)$$

To find the minimum of $Z(\mathbf{V}_1, \mathbf{V}'_1)$, we take derivative with respect to $\mathbf{V}_{1,ij}$,

$$\begin{aligned}&\tfrac{\partial Z(\mathbf{V}_1,\mathbf{V}'_1)}{\partial \mathbf{V}_{1,ij}} = 2\lambda \tfrac{(\mathbf{L}_1^+\mathbf{V}'_1)_{ij}\mathbf{V}_{1,ij}}{\mathbf{V}'_{1,ij}} - 2\lambda(\mathbf{L}_1^-\mathbf{V}'_1)_{ij}\tfrac{\mathbf{V}'_{1,ij}}{\mathbf{V}_{1,ij}}\\ &-2(\mathbf{G}_1^T\mathbf{U}^+)_{ij}\tfrac{\mathbf{V}'_{1,ij}}{\mathbf{V}_{1,ij}} + 2(\mathbf{G}_1^T\mathbf{U}^-)_{ij}\tfrac{\mathbf{V}_{1,ij}}{\mathbf{V}'_{1,ij}}\\ &+2\boldsymbol{\alpha}^+\mathbf{V}_{1,ij} - 2\boldsymbol{\alpha}^-\tfrac{\mathbf{V}'_{1,ij}}{\mathbf{V}_{1,ij}}\end{aligned}$$

and the Hessian matrix of $Z(\mathbf{V}_1, \mathbf{V}'_1)$

$$\begin{aligned}&\tfrac{\partial^2 Z(\mathbf{V}_1,\mathbf{V}'_1)}{\partial \mathbf{V}_{1,ij}\partial \mathbf{V}_{1,kl}} = \delta_{ik}\delta_{jl}(2\lambda\tfrac{(\mathbf{L}_1^+\mathbf{V}'_1)_{ij}}{\mathbf{V}'_{1,ij}} + 2\lambda(\mathbf{L}_1^-\mathbf{V}'_1)_{ij}\tfrac{\mathbf{V}'_{1,ij}}{\mathbf{V}^2_{1,ij}}\\ &+2(\mathbf{G}_1^T\mathbf{U}^+)_{ij}\tfrac{\mathbf{V}'_{1,ij}}{\mathbf{V}^2_{1,ij}} + 2\tfrac{(\mathbf{G}_1^T U^-)_{ij}}{\mathbf{V}'_{1,ij}}\\ &+2\boldsymbol{\alpha}^+ + 2\boldsymbol{\alpha}^-\tfrac{\mathbf{V}'_{1,ij}}{\mathbf{V}^2_{1,ij}})\end{aligned}$$

is a diagonal matrix with positive elements due to $\mathbf{V}_1$ initialization and update rule; δ_{ik} is the delta function, $\delta_{ik} = 1$ if $i = k$ and 0 otherwise. Therefore $Z(\mathbf{V}_1, \mathbf{V}'_1)$ a convex function of $\mathbf{V}_1$, we can obtain the global minimum of $Z(\mathbf{V}_1, \mathbf{V}'_1)$ by setting $\frac{\partial Z(\mathbf{V}_1,\mathbf{V}'_1)}{\partial \mathbf{V}'_{1,ij}} = 0$ and solve for $\mathbf{V}_1$, we can get Eq. (3.21). □

A.2 Proof to Theorem 3

Proof 8 *We rewrite Eq. (3.22) as*

$$\begin{aligned}&J(\mathbf{V}_2) = Tr(\lambda\mathbf{V}_2^T\mathbf{L}_2^+\mathbf{V}_2 - \lambda\mathbf{V}_2^T\mathbf{L}_2^-\mathbf{V}_2 - 2c\mathbf{G}_2^T\mathbf{U}^+\mathbf{V}_2\\ &+2c\mathbf{G}_2^T\mathbf{U}^-\mathbf{V}_2 + \boldsymbol{\beta}^+\mathbf{V}_2^T\mathbf{V}_2 - \boldsymbol{\beta}^-\mathbf{V}_2^T\mathbf{V}_2)\end{aligned}$$

According to Lemma 1, we have

$$\begin{aligned}&Tr(\mathbf{V}_2^T\mathbf{L}_2^+\mathbf{V}_2) \le \sum_{ij}\tfrac{(\mathbf{L}_2^+\mathbf{V}'_2)_{ij}\mathbf{V}^2_{2,ij}}{\mathbf{V}'_{2,ij}}\\ &\boldsymbol{\beta}^+Tr(\mathbf{V}_2^T\mathbf{V}_2) \le \boldsymbol{\beta}^+\sum_{ij}\mathbf{V}^2_{2,ij}\end{aligned}$$

Meanwhile, by the inequality $x \le \frac{(x^2+y^2)}{2y}, \forall y > 0$, we have

$$\begin{aligned}&Tr(\mathbf{G}_2^T\mathbf{U}^-\mathbf{V}_2^T) = \sum_{i,j}(\mathbf{G}_2^T\mathbf{U}^-)_{ij}V_{2,ij}\\ &\le \sum_{i,j}(\mathbf{G}_2^T\mathbf{U}^-)_{ij}\tfrac{\mathbf{V}^2_{2,ij}+\mathbf{V}'^2_{2,ij}}{\mathbf{V}'_{2,ij}}\end{aligned}$$

On the other hand, to get the lower bound for the remaining terms, we employ the inequality $z \geq 1 + \log z, \forall z > 0$, then

$$
\begin{aligned}
&Tr(\mathbf{G}_2^T \mathbf{U}^+ \mathbf{V}_2^T) \geq \sum_{ij} (\mathbf{G}_2^T \mathbf{U}^+) \mathbf{V}'_{2,ij} (1 + \log \tfrac{\mathbf{V}_{2,ij}}{\mathbf{V}'_{2,ij}}) \\
&Tr(\mathbf{V}_2^T \mathbf{L}_2^- \mathbf{V}_2) \geq \sum_{ijk} (\mathbf{L}_2^-)_{jk} \mathbf{V}'_{ji} \mathbf{V}'_{ki} (1 + \log \tfrac{\mathbf{V}_{2,ji}\mathbf{V}_{2,ki}}{\mathbf{V}'_{2,ji}\mathbf{V}'_{2,ki}}) \\
&Tr(\mathbf{V}_2^T \mathbf{V}_2) \geq \sum_{ijk} \mathbf{V}'_{2,ij} \mathbf{V}'_{2,ik} (1 + \log \tfrac{\mathbf{V}_{2,ij}\mathbf{V}_{2,ik}}{\mathbf{V}'_{2,ij}\mathbf{V}'_{2,ik}})
\end{aligned}
$$

By summing over all the bounds, we get $Z(\mathbf{V}_2, \mathbf{V}'_2)$ and it is easy to conclude that

$$
Z(\mathbf{V}_2, \mathbf{V}'_2) \geq J(\mathbf{V}_2), \quad Z(\mathbf{V}_2, \mathbf{V}_2) = J(\mathbf{V}_2)
$$

To find the minimum of $Z(\mathbf{V}_2, \mathbf{V}'_2)$, we take derivative with respect to $\mathbf{V}_{2,ij}$,

$$
\begin{aligned}
&\tfrac{\partial Z(\mathbf{V}_2, \mathbf{V}'_2)}{\partial \mathbf{V}_{2,ij}} = 2\lambda \tfrac{(\mathbf{L}_2^+ \mathbf{V}'_2)_{ij} \mathbf{V}_{2,ij}}{\mathbf{V}'_{2,ij}} - 2\lambda (\mathbf{L}_2^- \mathbf{V}'_2)_{ij} \tfrac{\mathbf{V}'_{2,ij}}{\mathbf{V}_{2,ij}} \\
&-2(c\mathbf{G}_2^T \mathbf{U}^+)_{ij} \tfrac{\mathbf{V}'_{2,ij}}{\mathbf{V}_{2,ij}} + 2(c\mathbf{G}_2^T \mathbf{U}^-)_{ij} \tfrac{\mathbf{V}_{2,ij}}{\mathbf{V}'_{2,ij}} \\
&+2\boldsymbol{\beta}^+ \mathbf{V}_{2,ij} - 2\boldsymbol{\beta}^- \tfrac{\mathbf{V}'_{2,ij}}{\mathbf{V}_{2,ij}}
\end{aligned}
$$

and the Hessian matrix of $Z(\mathbf{V}_2, \mathbf{V}'_2)$

$$
\begin{aligned}
&\tfrac{\partial^2 Z(\mathbf{V}_2, \mathbf{V}'_2)}{\partial \mathbf{V}_{2,ij} \partial \mathbf{V}_{2,kl}} = \delta_{ik}\delta_{jl}(2\lambda \tfrac{(\mathbf{L}_2^+ \mathbf{V}'_2)_{ij}}{\mathbf{V}'_{2,ij}} + 2\lambda (\mathbf{L}_2^- \mathbf{V}'_2)_{ij} \tfrac{\mathbf{V}'_{2,ij}}{\mathbf{V}^2_{2,ij}} \\
&+2(c\mathbf{G}_2^T \mathbf{U}^+)_{ij} \tfrac{\mathbf{V}'_{2,ij}}{\mathbf{V}^2_{2,ij}} + 2\tfrac{(c\mathbf{G}_2^T U^-)_{ij}}{\mathbf{V}'_{2,ij}} \\
&+2\boldsymbol{\beta}^+ + 2\boldsymbol{\beta}^- \tfrac{\mathbf{V}'_{2,ij}}{\mathbf{V}^2_{2,ij}})
\end{aligned}
$$

is a diagonal matrix with positive elements due to $\mathbf{V}_2$ initialization and update rule; δ_{ik} is the delta function, $\delta_{ik} = 1$ if $i = k$ and 0 otherwise. Therefore $Z(\mathbf{V}_2, \mathbf{V}'_2)$ a convex function of $\mathbf{V}_2$, we can obtain the global minimum of $Z(\mathbf{V}_2, \mathbf{V}'_2)$ by setting $\frac{\partial Z(\mathbf{V}_2, \mathbf{V}'_2)}{\partial \mathbf{V}'_{2,ij}} = 0$ and solve for $\mathbf{V}_2$, we can get Eq. (3.23). □

APPENDIX B

Convergence Proof Of Lemma In Chapter 4

In this appendix, we provide the proof to Lemma 3.

B.1 Proof to Lemma 3

Proof 9 *Obviously,* $(A^{\frac{1}{2}} - B^{\frac{1}{2}})^2$ *is positive semi-definite. It is then easy to see* $B^{-\frac{1}{4}}(A^{\frac{1}{2}} - B^{\frac{1}{2}})^2 B^{-\frac{1}{4}}$ *is positive semi-definite also. Therefore, we can establish the following inequality*

$$Tr(B^{-\frac{1}{4}}(A^{\frac{1}{2}} - B^{\frac{1}{2}})^2 B^{-\frac{1}{4}}) \geq 0 \tag{B.1}$$

Starting from Eq. (B.1), we get

$$\begin{aligned}
&\Rightarrow Tr((A^{\frac{1}{2}} - B^{\frac{1}{2}})^2 B^{-\frac{1}{2}}) \geq 0 \\
&\Rightarrow Tr((A - A^{\frac{1}{2}}B^{\frac{1}{2}} - B^{\frac{1}{2}}A^{\frac{1}{2}} + B)B^{-\frac{1}{2}}) \geq 0 \\
&\Rightarrow Tr((A + B - 2A^{\frac{1}{2}}B^{\frac{1}{2}})B^{-\frac{1}{2}}) \geq 0 \\
&\Rightarrow Tr(AB^{-\frac{1}{2}} + B^{\frac{1}{2}} - 2A^{\frac{1}{2}}) \geq 0 \\
&\Rightarrow \tfrac{1}{2}Tr(AB^{-\frac{1}{2}}) - Tr(A^{\frac{1}{2}}) \geq \tfrac{1}{2}Tr(BB^{-\frac{1}{2}}) - Tr(B^{\frac{1}{2}})
\end{aligned} \tag{B.2}$$

REFERENCES

[1] M. Granovetter, "The strength of weak ties," *American Journal of Sociology*, vol. 78, pp. 1360–1380, 1973.

[2] S. Wasserman and K. Faust, *Social Networks Analysis: Methods and Applications*. Cambridge University Press, 1994.

[3] C. Kadushin, *Understanding social networks: Theories, concepts, and findings*. Oxford University Press, 2012.

[4] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," in *Sixth Symposium on Operating System Design and Implementation*, San Francisco, CA, USA, 2004.

[5] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust," in *WWW*, 2004, pp. 403–412.

[6] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *WWW*, 2003, pp. 640–651.

[7] M. McPherson, L. Smith-Lovin, and J. Cook, "Birds of a feather: Homophily in social networks," *Annual Review of Sociology*, vol. 27, 2001.

[8] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri, "Feedback effects between similarity and social influence in online communities," in *SIGKDD*, 2008, pp. 160–168.

[9] J. Huang and et al., "Social trust prediction using rank-k matrix recovery," in *23rd International Joint Conference on Artificial Intelligence*, 2013.

[10] J. Huang, F. Nie, and H. Huang, "Robust discrete matrix completion," in *27th AAAI Conference on Artificial Intelligence*, 2013.

[11] P. Melville, R. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in *AAAI*, 2002, pp. 187–192.

[12] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004.

[13] M. Pazzani and D. Billsus, "Content-based recommendation systems," *The Adaptive Web*, vol. 4321, pp. 325–341, 2007.

[14] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *WWW*, 2001, pp. 285–295.

[15] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[16] X. Su and T. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, no. 4, 2009.

[17] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *WWW*, 2010, pp. 811–820.

[18] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[19] J. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *ICML*, 2005, pp. 713–719.

[20] R. R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *NIPS*, 2008, pp. 1257–1264.

[21] K. Zhou and H. Zha, "Learning binary codes for collaborative filtering," in *SIGKDD*, 2012, pp. 498–506.

[22] S. J. B. Shen and J. Ye, "Mining discrete patterns via binary matrix factorization," in *SIGKDD*, 2009, pp. 757–766.

[23] E. C. . B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[24] S. Ji and J. Ye, "An accelerated gradient method for trace norm minimization," in *ICML*, 2009, pp. 457–464.

[25] P. J. R. Meka and I. Dhillon, "Matrix completion from power-law distributed samples," in *NIPS*, 2009, pp. 1258–1266.

[26] X. Shi and P. Yu, "Limitations of matrix completion via trace norm minimization," *SIGKDD Explorations*, vol. 12, no. 2, pp. 16–20.

[27] D. Bertsekas, *Nonlinear Programming*. New York: Athena Scientific.

[28] S. Dewester, S. Dumains, T. Landauer, G. Furnas, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[29] D. Bertsekas, *Constrained optimization and lagrange multiplier methods*. Athena Scientific, 1996.

[30] M. Hestenes, "Multiplier and gradient methods," *Journal of Optimization Theory and Applications*, vol. 4, pp. 303–320, 1969.

[31] M. Powell, *A method for nonlinear constraints in minimization problems*. In R. Fletcher, editor, Optimization. Academic Press, London and New York, 1969.

[32] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv preprint arXiv:1009.5055*, 2010.

[33] R. Larsen, "Propackcsoftware for large and sparse svd calculations," Tech. Rep., 2005. [Online]. Available: http://sun.stanford.edu/rmunk/PROPACK

[34] P. Jain, R. Meka, and I. Dhillon, "Guaranteed rank minimization via singular value projection," in *NIPS*, 2010, pp. 937–945.

[35] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: exact recovery of corrupted low-rank matrices via convex optimization," in *NIPS*, 2009, pp. 2080–2088.

[36] F. Cai, E. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2008.

[37] P. Massa and P. Avesani, "Trust-aware bootstrapping of recommender systems," in *ECAI Workshop on Recommender Systems*, 2006.

[38] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *WWW*, 2010.

[39] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Mathematics*, vol. 6, no. 1, pp. 29–123, 2009.

[40] P. Bedi, H. Kaur, and S. Marwaha, "Trust based recommender system for semantic web," in *Proceedings of the 20th international joint conference on Artifical intelligence*. San Francisco, CA: Morgan Kaufmann Publishers Inc, 2007, pp. 2677–2682.

[41] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proceedings of the 2007 ACM conference on Recommender systems*. New York, NY: ACM, 2007, pp. 17–24.

[42] A. Mislove, B. Viswanath, P. Gummadi, and P. Druschel, "You are who you know: Inferring user profiles in online social networks," in *Proceedings of the third ACM international conference on Web search and data mining*. New York, NY: ACM, 2010, pp. 251–260.

[43] Z. Wen and C. Lin, "On the quality of inferring interests from social neighbors," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY: ACM, 2010, pp. 373–382.

[44] S. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transaction on Knowledge and Data Engineering*, vol. 22, pp. 1345–1359, 2010.

[45] L. Getoor and C. Diehl, “Link mining: a survey.” *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 2, pp. 3–12, 2005.

[46] D. Liben-Nowell and J. Kleinberg, “The link prediction problem for social network-s,” in *Proceedings of the twelfth international conference on Information and knowledge management*. New York, NY: ACM, 2003, pp. 556–559.

[47] K. Yu, W. Chu, S. Yu, V. Tresp, and X. Zhao, “Stochastic relational models for discriminative link prediction,” in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2006, pp. 333–340.

[48] K. Yu and W. Chu, “Gaussian process models for link analysis and transfer learning,” in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2007, pp. 1657–1664.

[49] K. Yu, J. Lafferty, S. Zhu, and Y. Gong, “Large-scale collaborative prediction using a nonparametric random effects model,” in *Proceedings of the 26th Annual International Conference on Machine Learning*. New York, NY: ACM, 2007, pp. 1185–1192.

[50] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proceedings of the 10th international conference on World Wide Web*. New York, NY: ACM, 2001, pp. 285–295.

[51] T. Hofmann and J. Puzicha, “Latent class models for collaborative filtering,” in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*. New York, NY: ACM, 1999, pp. 688–693.

[52] N. Srebro and T. Jaakkola, “Weighted low-rank approximations,” in *Proceedings of the 20th Annual International Conference on Machine Learning*. Palo Alto, California: AAAI Press, 2003, pp. 720–727.

[53] R. Salakhutdinov and A. Mnih, "Probabilitistic matrix factorization," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2007, pp. 1257–1264.

[54] ——, "Bayesian probabilistic matrix factorization using marknov chain monte carlo," in *Proceedings of the 25th International Conference on Machine learning*. New York, NY: ACM, 2008, pp. 880–887.

[55] A. Singh and G. Gordon, "Relational learning via collective matrix factorization," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and Data Mining*. New York, NY: ACM, 2008, pp. 650–658.

[56] S. Zhu, K. Yu, Y. Chi, and Y. Gong, "Combining content and link for classification using matrix factorization," in *Proceedings of the 30th annual international ACM SIGIR conference on research and development in Information Retrieval*. New York, NY: ACM, 2007, pp. 487–494.

[57] Z. Xu, K. Kersting, and V. Tresp, "Multi-relational learning with gaussian process," in *Proceedings of the 21st International Jont Conference on Artifical Intelligence*. San Francisco, CA: Morgan Kaufmann Publishers Inc, 2009, pp. 1309–1314.

[58] B. Cao, N. Liu, and Q. Yang, "Transfer learning for collective link prediction in multiple heterogenous domains," in *Proceedings of the 27th International Conference on Machine Learning*. New York, NY: ACM, 2010, pp. 159–166.

[59] W. Pan, W. Xiang, N. Liu, and Q. Yang, "Transfer learning in collaborative filtering for sparsity reduction," in *Proceedings of the 24th AAAI Conference on Artificial Intelligence*. Palo Alto, California: AAAI Press, 2010, pp. 230–235.

[60] J. Huang and et al., "Trust prediction via aggregating heterogeneous social networks," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. New York, NY: ACM, 2012, pp. 1774–1778.

[61] ——, “Social trust prediction using heterogeneous networks,” *ACM Transactions on Knowledge Discovery from Data*, 2013.

[62] X. He and P. Niyogi, “Locality preserving projections,” in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2003, pp. 153–160.

[63] D. Cai, X. He, X. Wu, and J. Han, “Non-negative factorization on manifold,” in *Proceedings of the 8th IEEE International Conference on Data Mining*. Los Alamitos, CA: IEEE, 2008, pp. 63–72.

[64] S. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[65] O. Chapelle, B. Scholkopf, and A. Zien, *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006.

[66] U. von Luxburg, “A tutorial on spectral clustering,” *Statistics and Computing*, vol. 17, no. 4, pp. 395 – 416, 2006.

[67] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[68] Y. W. A.Y. Ng, M.I. Jordan, “On spectrral clustering: Analysis and an algorithm,” in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2001, pp. 849–856.

[69] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge: Cambridge University Press, 2004.

[70] C. Ding, T. Li, and M. Jordan, “Convex and semi-nonnegative matrix factorization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, 2010.

[71] R. Yuster and U. Zwick, “Fast sparse matrix multiplication,” *ACM Transactions on Algorithms*, vol. 1, no. 1, pp. 2–13, 2005.

[72] D. Lee and H. Seung, “Algorithms for non-negative matrix factorization,” in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2000, pp. 556–562.

[73] G. Jeh and J. Widom, “A measure of structural-context similarity,” in *Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and Data Mining*. New York, NY: ACM, 2002, pp. 538–543.

[74] D. Billsus and M. J. Pazzani, “Learning collaborative information filters,” in *Proceedings of the 15th International Conference on Machine Learning*. San Francisco, CA: Morgan Kaufmann Publishers Inc, 1998, pp. 46–54.

[75] E. Candes and B. Recht, “Exact matrix completion via convex optimization,” *Communications of the ACM*, vol. 55, no. 6, pp. 111–119, 2012.

[76] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl, “An algorithmic framework for performing collaborative filtering,” in *Conference on Research and Development in Information Retrieval*. New York, NY: ACM, 1999, pp. 230–237.

[77] P. Massa and P. Avesani, “Trust metrics in recommender systems,” *Computing with Social Trust*, pp. 259–285, 2009.

[78] J. Davis and M. Goadrich, “The relationship between precision-recall and roc curves,” in *Proceedings of the 15th International Conference on Machine Learning*. New York, NY: ACM, 2006, pp. 233–240.

[79] H. Huang and et al., “A new sparse simplex model for brain anatomical and genetic network analysis,” in *MICCAI*, 2013.

[80] J. Huang, F. Nie, and H. Huang, “Spectral rotation vs k-means in spectral clustering,” in *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[81] J. Huang, F. Nie, H. Huang, and D. Chris, “Robust manifold non-negative matrix factorization,” 2013.

[82] E. J. Candes and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Transaction on Information Theory*, vol. 56, no. 5, pp. 2053–2080, 2010.

[83] J. Breese, D. Heckman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *UAI*, 1998, pp. 43–52.

[84] C. Cornelis, X. Guo, J. Lu, and G. Zhang, "A fuzzy relational approach to event recommendation," in *Indian International Conference on Artificial Intelligence*, pp. 2231–2242", YEAR=2005.

[85] J. L. S. T. T. J. E. Minkov, B. Charrow, "Collaborative future event recommendation," in *CIKM*, 2010, pp. 819–828.

[86] X. Su and T. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. January, no. 4, 2009.

[87] F. Chung, *Spectral Graph Theory*. American Mathematical Society, 1997.

[88] U. Luxberg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[89] V. N. Vapnik, *Statistical learning theory*. Wiley, 1998.

[90] T. Joachims, "Transductive inference for text classification using support vector machines," in *ICML*, 1999, pp. 200–209.

[91] ——, "Transductive learning via spectral graph partitioning," in *ICML*, 2003, pp. 290–297.

[92] F. Radlinski, P. Bennett, B. Carterette, and T. Joachims, "Redundancy, diversity and interdependent document relevance," in *SIGIR*, 2009, pp. 46–52.

[93] G. Dupret and C. Liao, "A model to estimate intrinsic document relevance from the clickthrough logs of a web search engine," in *WSDM*, 2010, pp. 181–190.

[94] R. Mooney and L. Roy, "Content-based book recommending using learning for text categorization," in *ACM conference on Digital libraries Pages*, 2000, pp. 195–204.

[95] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2002.

[96] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[97] T. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, pp. 5228–5235, 2004.

[98] X. Wei and W. Croft, "Lda-based document models for ad-hoc retrieval," in *SIGIR*, 2006, pp. 178–185.

[99] G. Salton, E. Fox, and H. Wu, "Extended boolean information retrieval," *Communications of the ACM*, vol. 26, no. 11, pp. 1022–1036, 1983.

[100] C. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[101] T. Joachims, "Optimizing search engines using clickthrough data," in *SIGKDD*, 2002, pp. 133–142.

[102] ——, "Training linear svms in linear time," in *SIGKDD*, 2006, pp. 217–226.

[103] Y. Liu and X. Yao, "Ensemble learning via negative correlation," *Neural Networks*, vol. 12, no. 10, pp. 1399–1404, 1999.

[104] M. Balabanovic and Y. Shoham, "Fab: content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[105] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[106] T. Joachims, "Making large-scale support vector machine learning practical," *Advances in kernel methods*.

[107] D. Hosmer and S. Lemeshow, *Applied Logistic Regression*. Wiley, 2005.

[108] J. Huang, F. Nie, H. Huang, and C. Ding, "Supervised and projected sparse coding for image classification," in *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

BIOGRAPHICAL STATEMENT

Jin Huang was born in Suzhou, China, in 1981. He received his B.S. degree in mathematics from Dalian University of Technology in 2004, his two M.S. degrees from University of Louisiana and Bowling Green State University, in mathematics and statistics, respectively. From 2009 to 2013, he was pursuing Ph.D degree in computer science under the supervision of Dr Heng Huang. He has published numerous 1st author papers in IJCAI, AAAI and CIKM, he also has collaborated paper in TKDE, MICCAI etc. He also has intern experience in different research institutes such as NIH, UC Berkeley Lawrence Lab.