# LINK STRENGTH PREDICTION USING TRANSACTION-BASED MATRIX FACTORIZATION

by

Ali Bozorgkhan

BEng, University of Tehran, 2011

A Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science

in the
School of Computing Science
Faculty of Applied Sciences

© Ali Bozorgkhan 2013
SIMON FRASER UNIVERSITY
Summer 2013

All rights reserved.
However, in accordance with the *Copyright Act of Canada*, this work may be reproduced without authorization under the conditions for "Fair Dealing." Therefore, limited reproduction of this work for the purposes of private study, research, criticism, review and news reporting is likely to be in accordance with the law, particularly if cited appropriately.

## APPROVAL

**Name:** Ali Bozorgkhan

**Degree:** Master of Science

**Title of Thesis:** Link Strength Prediction Using Transaction-Based Matrix Factorization

**Examining Committee:** Dr. –
Chair

______________________________

Dr. Oliver Schulte, Associate Professor, Computing Science
Simon Fraser University
Senior Supervisor

______________________________

Dr. Martin Ester, Professor, Computing Science
Simon Fraser University
Supervisor

______________________________

Dr. –, Professor, Computing Science
Simon Fraser University
SFU Examiner

**Date Approved:** ______________________________

## Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection (currently available to the public at the “Institutional Repository” link of the SFU Library website (www.lib.sfu.ca) at http://summit/sfu.ca and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author’s written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

While licensing SFU to permit the above uses, the author retains copyright in the thesis, project or extended essays, including the right to change the work for subsequent purposes, including editing and publishing the work in whole or in part, and licensing other parties, as the author may desire.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library
Burnaby, British Columbia, Canada

revised Fall 2011

# Abstract


The revolution of social networks and methods of analyzing them have attracted interest in many research fields. Predicting whether a friendship holds in a social network between two individuals or not, link prediction, has been a heavily researched topic in the last decade . In this research I've investigated a related problem, link strength prediction: how to assign ratings or strengths to friendship links. A basic approach would ~~be~~ matrix factorization applied to only friendship ratings. However, the existence of extensive transactions among users may be used for better predictions. I propose a new type of multiple-matrix factorization model for incorporating a transaction matrix. I derive gradient descent update equations for learning latent factors that predict values in the target rating matrix. Multiple-matrix factorization can be seen as a data fusion technique, that combines evidence from different sources. In the social network application, the target matrix contains friendship ratings and the evidence matrices specify transaction intensities between users. To evaluate the model, I introduce data from Cloob, a popular Iranian social network as well as synthetic data.

*to my unborn niece*

*"How You Doing?"*

*Joey Tribbiani*

# Acknowledgments

Here go all the people you want to thank.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction and Overview

Online social networks (Facebook, Orkut, LinkedIn, Myspace) and methods of analyzing them have attracted extensive interest in many research fields. Much of the past work has been focused on social networks with binary relational ties (e.g., whether two people are friends or not), which is known as link prediction [16]. Because of the low-cost or effort for becoming friends in online social networks, the resulting networks have both strong and weak ties with little or no information to differentiate the two, so these binary indicators provide only a coarse indication of the nature of the relationship. Treating both strong and weak ties the same, when analyzing the characteristics of an individual, increases the noise level and leads to misleading results. A recent trend in online social networks, such as Google Plus and Facebook, has been to allow users to differentiate among their friendship links by forming lists or circles. [3] presents research on restricting Twitter lists to a subset of people followed by a user. I formulate the link strength problem to extend link prediction to estimate the strength of a tie given a pair of linked individuals where larger values indicate a stronger link [9, 4].

The topological structure of the network defines an explicit network that contains information about the strength of links. A basic approach would be to apply matrix factorization to this network to learn latent factors for each user. In this model, the predicted value of a link's strength between a pair of users is a function of dot product of their latent variables. Although this model performs reasonably well, the existence of a large volume of data recording transactions between users is a motivation for building more complex models that better predict strengths of friendships. The strength of a tie directly impacts the frequency of transactional events between users, as users tend to interact mostly with their strong ties.

These transactional intensities generate an implicitly weighted network among users that may be used in addition to the explicit network. Previous approaches have derived features from the implicit networks and then used the features to build a classification model for distinguishing weak from strong ties [9, 4]. Instead, multiple-matrix factorization performs data fusion, which is the process of combining data from multiple sources into one model for analysis.

Collective matrix factorization is a general approach for analyzing multiple matrices [26, 17, 23]. In this model, for each individual a single latent feature vector is introduced to explain all the links that the individual participates in. Collective matrix factorization is well-suited to generative models that represent the joint distribution over all relationships. In contrast, I use a new model where the goal is to predict the values of a single target matrix—strength of friendship—and implicit matrices, like transactional intensities, are used to weight the importance of connections between users.

Research on social recommendation systems has shown that friendship information can improve item recommendation [7, 26]. Investigations of social networks have shown that it is friends with strong ties, rather than weak ties, that exhibit similar preferences [4]. Therefore a promising application of link strength prediction is to combine it with social recommendation models (i.e., a user-item matrix) [25].

### 1.0.1 Evaluation

I use data from Cloob[1], a popular Iranian website, as well as synthetic data to validate my work. To my knowledge, the Cloob dataset is the only dataset that contains explicit friendship ratings, which serve as a ground truth for my evaluation. Experimental results show that my Transaction-Based Matrix Factorization model outperforms Collective Matrix Factorization, single table matrix factorization, as well as models that analyze the data matrices separately. An important challenge for link strength prediction is the presence of *zero-transaction friendships*, between user dyads that are friends but have no recorded activities. My experiments illustrate that even in the absence of direct transactions between two friends, their transactional behavior with their other friends is informative and leads Transaction-Based Matrix Factorization model to perform better than competitors. My experiments also examine two ways to improve the results. First by examining the impact

[1]http://www.cloob.com

of a number of model parameters, including the dimensionality of the latent factors and a trade-off parameter that controls the importance of the transaction intensities. Second, by using different methods to initialize the starting points of the algorithm. The starting points in the simplest way could be initialized randomly. Changing the ~~parameters~~ to more meaningful values using existing data could lead to more accurate results.

### 1.0.2 Contributions

- Since the user-user link strength rating is directed, I distinguish two types of latent feature vectors associated with each user: one that models the user's behavior as a rater, and a second that models the user's behavior as a ratee.
- I present the first evaluation of link strength prediction on a real-world data set with continuous ratings, as ground truth, as opposed to binary link strength labels.
- Different methods will be presented to reduce the error of link prediction.

### 1.0.3 Thesis Organization

**Chapter 2** briefly describes social networks with two examples of using them in researches and a graph representation of them. Cloob.com will be also introduced as the main dataset of this research. In addition, link strength prediction and previous models such as latent variable models and data fusion will be briefly described.

**Chapter 3** describes Matrix Factorization, Matrix Factorization with Baselines, Collective Matrix Factorization, Collective Matrix Factorization with Baselines and Transaction-Based Matrix Factorization in details. All these models use gradient descent for optimization which will also be explained.

**Chapter 4** contains more information about Cloob.com and the synthetic dataset. All the equation and parameters used in creating the synthetic dataset will be in this chapter. And last, cross validation will be explained as it is used in the evaluation part.

**Chapter 5** is the evaluation chapter which contains all the results from running different methods on my datasets which includes datafusion experiments, zero-transaction friendships and result improvements.

**Chapter 6** contains the conclusion of this research and the future works.

# Chapter 2

# Background

## 2.1 Social Networks

Social networks have an important role in our lives. When we want to find a job, we ask our friends and family to see if they know a company which hires new employees. We find people in online social networks like Facebook[1] and Twitter [2] and connect with them. These are just simple examples of usage of social networks in our social life.

## 2.2 Examples

Here are two examples of social network analysis.

### 2.2.1 Closely Connected Countries in Twitter

Twitter, with more than 500 million registered users [3] and billions of activities like tweets, retweets and followings, among them, has been a great network for researchers [10, 8] to analyze. Kulshrestha et al[14] have used Twitter's data to dissect it geographically. In section "Impact of geography & language", they have analyzed Twitter's dataset to check if users of a country A are likely to be connected with users of country B where A and B are neighbors or share a similar language. Figure 2.1 shows the 'closely connected' groups

[1] `facebook.com`

[2] `twitter.com`

[3] `http://www.mediabistro.com/alltwitter/500-million-registered-users_b18842`

of countries.

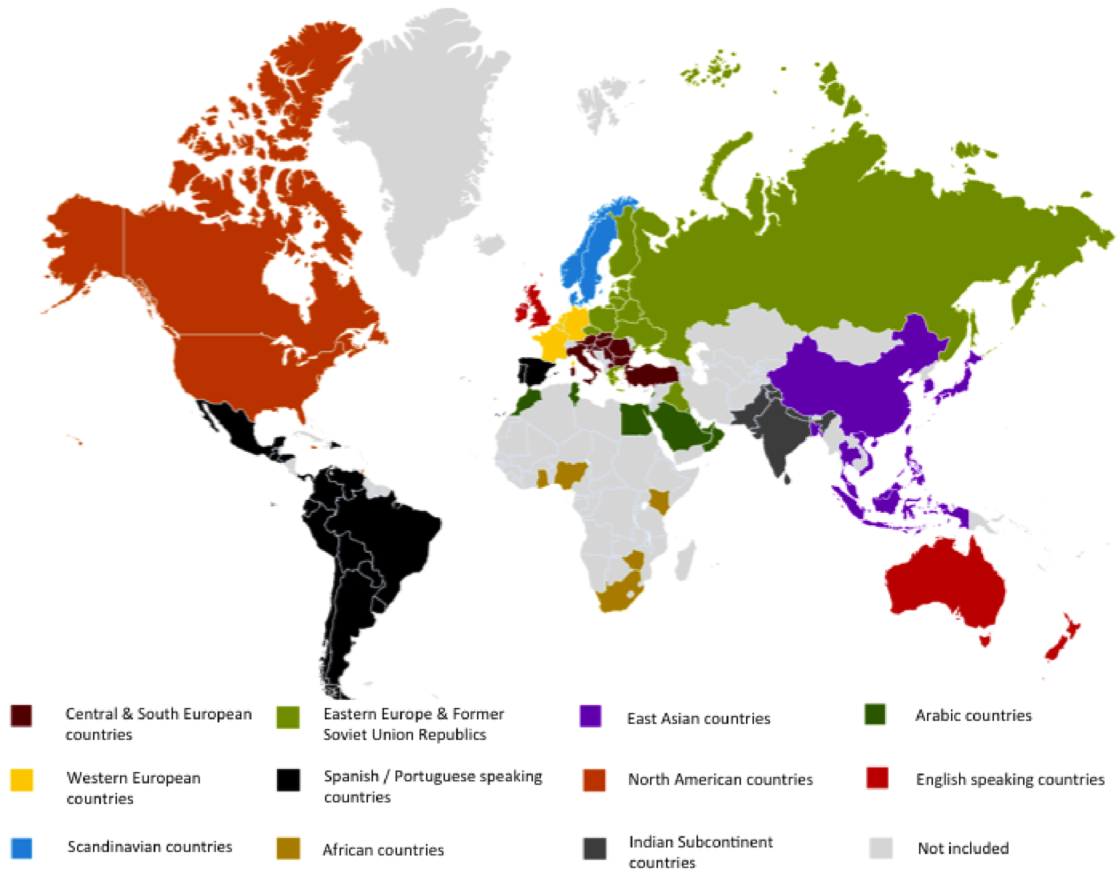


Figure 2.1: Groups of countries whose users are closely connected with one another.

As ~~its~~ shown in figure 2.1, all the 91 countries in the dataset are classified into 11 distinct groups. Members of each group have similar attributes like geographical coordinates, culture, linguistics, and politics. For example North American countries like Canada and The United States that speak English are in a group distinguished from the countries in South and Latin America that speak Spanish or Portuguese. This grouping shows the importance of the factors like boundaries and linguistics in international relations.

### 2.2.2 Political Geography

Conover et al[2] have analyzed activities among Twitter users during the 2010 midterm congressional elections. They extracted the political tweets by monitoring the hash tags used in the tweets. For example, hash tags like # us, # wc, and # lgbt indicate a left-leaning political tweet, # qsn and # politicalhumor indicate a right-leaning political tweet, and # israel and # rs indicate a tweet with both views.

During the election, some of the states had more left-leaning political tweets than expected and some had less. The expected number of tweets ~~coming~~ is the number of tweets ~~came~~ from that state over the total number of tweets coming from inside the United States.

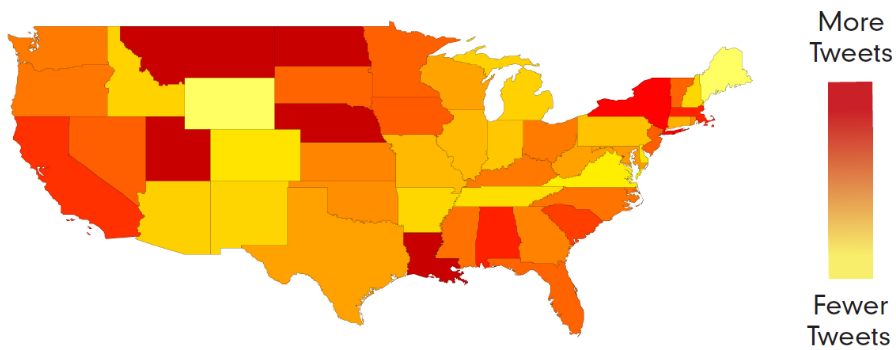


Figure 2.2: Ratio of the number of political tweets against the expected number during 2010 midterm congressional elections.

Figure 2.2 shows all of the states colored from yellow to dark red. Dark red means that more tweets came from the state than usual and yellow means less. The result indicates that traditional geographic distribution and geographic distribution of the left-leaning network community are strongly dependent.

## 2.3 Representation of Social Networks

A social network is usually represented by a graph whose nodes represent the objects in the network such as users and movies, and edges indicate a transaction or relation between a pair of nodes[15].

Depending on the network, the represented graph could be directed or undirected. For example, the graph representing activities among Facebook users is directed. User U having an activity with user V does not imply user V has also an activity with user U. Twitter's following graph would also be directed because a user could follow another user and not followed by them. On the other hand, some ~~representing~~ graphs could be undirected like facebook's friendships. An edge between user U and V in facebook indicates that U and V are friends.

Independent from the type of a graph, its edges could be binary or weighted:

1. Binary Edges: Edges either exist or not. Their existence only shows that there is a relation between the pair of nodes covering that edge.

2. Weighted Edges: Edges could have different values. In this case, an edge not only indicates that there is a link between two nodes but also shows the strength of their relationship. For example, in IMDB[4], users can rate movies from 1 to 10. The more a user likes a movie, the higher their rating goes.

Figure 2.3 shows simple examples of these four types of graphs.

[4] `imdb.com`

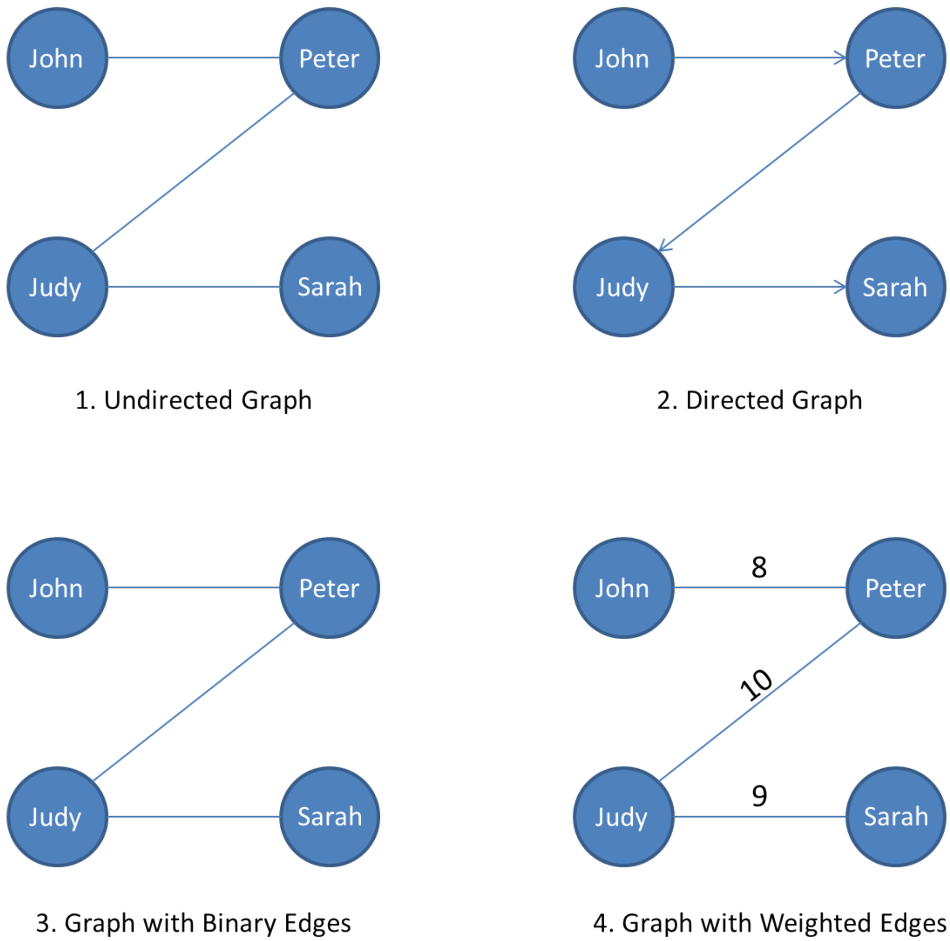


Figure 2.3: Simple examples of a User-User and a User-Item social network represented using different types of graphs. In (1) and (2), users John, Judy, Peter and Sarah have interactions with each other. In (3) and (4), users John and Judy have interactions with movies Flight and Crash. (1) An undirected graph showing that friendships between John and Peter, Peter and Judy, Judy and Sarah. (2) A directed graph showing an activity (e.g. writing on a friend's wall). John has wrote on Peter's wall, Peter has wrote on Judy's wall and Judy has wrote on Sarah's wall. (3) A binary graph showing the existence of a relation between a user and a movie. John and Judy have seen the movie Flight and Judy has also seen Crash. (4) A weighted graph showing the strength of link strength between users and items. John and Judy have watched flight and rated it 8 and 10 from 10 in IMDB which indicates that they have liked the movie. Judy has rated Crash 4 out of 10 which shows the lack of interest of her in that movie.

## 2.4 Cloob.com

Cloob[5] is the most popular online social network among Iranian Internet users in Iran. With more than 1.5 million registered users, 25 million friendships and 20 million activities among users, recorded between 2005 and 2010, it is now one of the top ten most-viewed websites in Iran according to Alexa[6].

Users in Cloob.com can find friends, send messages, share photos, chat with other users, join clubs and their communities and do large variety of different activities. Some features in Cloob are not free. Users need to purchase Cloob's money, which is called Croob, to spend on desired features like enabling "who visited my page" or "buying a club". Users also can have Croob transaction with other users which makes Cloob more and more like real life.

A special feature of Cloob's friendship service is that users can assign a value between 0 and 5 to indicate the strength of their friendship with their friends. As users can rate each other differently, there are two different records for each friendship. Among all the services that Cloob has, I have chosen this feature to work on during my graduate research work.

## 2.5 Link Strength Prediction

The act of predicting a future edge in a social network graph with its strength using current information is called link strength prediction. Figure 2.4 is an example of a simple social network. Each weight on the arrows indicates the strength of the friendship which is a number between 0 and 5. A larger rating means a closer friendship. As you can see, there is no link between John and Sarah, or between Peter and Judy. According to the friendship ratings, we could build the friendship table like Figure 2.5. In this example, Peter and John are close friends and John and Judy are good friends. With these two assumptions, we could predict that Peter and Judy could be 'good friends' but they have not found each other on this social network yet. We could predict the relationship strength between John and Sarah as well by looking at their friendships with other people.

[5] `cloob.com`

[6] `alexa.com`

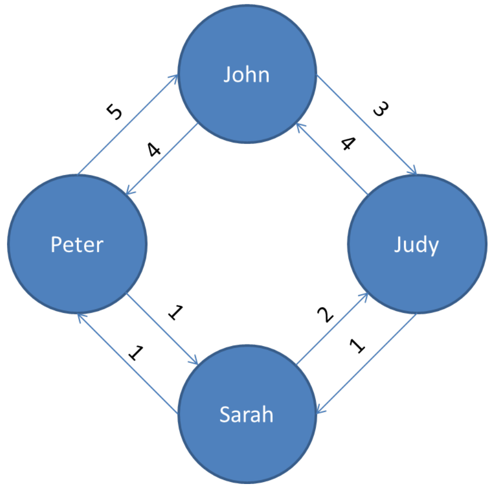


Figure 2.4: An example of a simple social network graph

| | Peter | John | Judy | Sarah |
|---|---|---|---|---|
| Peter | - | 5 | ? | 1 |
| John | 4 | - | 3 | ? |
| Judy | ? | 4 | - | 1 |
| Sarah | 1 | ? | 2 | - |

Figure 2.5: The friendship table of Figure 2.4

## 2.6 Previous Models

I have used two of the common learning methods in link prediction, Latent Variable Modeling and Data Fusion.

### 2.6.1 Latent Variable Modeling

Latent variable models attempt to explain complex relations between several variables by simple relations between the variables and an underlying unobservable, i.e. latent structure. Formally we have a collection $x = \{x_1, x_2, ..., x_p\}$ of manifest variables which can be observed, and a collection $y = \{y_1, y_2, ..., y_q\}$ of latent variables which are unobservable. Manifest variables are assumed to be conditionally independent given the latent variables. The following graph shows a simple latent variable model. Note that the size of collection $x$ should be much greater than the size of collection $y$ in order to have a useful model.

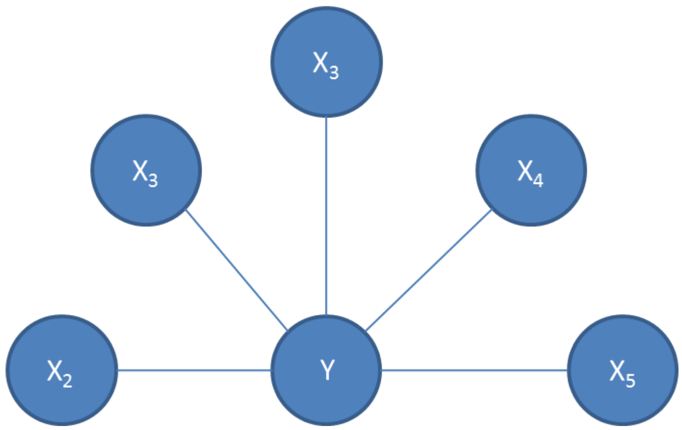


Figure 2.6: Simple example of a Latent Variable Model

### 2.6.2 Data Fusion

Using multiple sources simultaneously in order to get a better result than using sources individually is called data fusion [9]. An example of it could be an illustration given by the human system, which uses different sources to analyze its environment. The main 5 senses that could be used are sight, smell, hearing, taste and touch. In addition, the brain uses its memory, past experiences and knowledge other than the mentioned senses. By using all the information gathered, the brain 'fuses' all the information and creates a representation of the environment.[7]

Data fusion could be used in link prediction to improve the accuracy of the results. For example, in a social network, for predicting a friendship of user $U$ and $V$, we could only look at their mutual friends and decide whether they could be friends or not. But using their other information as well, like their city of birth, where they have worked or which university they have graduated from could improve the accuracy of the prediction.

[7]http://data-fusion.org

# Chapter 3

# Models

Suppose the social network consists of a set of $N$ users. The link strengths expressed by users for other users are given in a link strength matrix $R_{N\times N}$. This matrix is not symmetric, because users $u$ and $v$ can express different strengths for their friendship. Instead of using indices like $i, j$ for a generic user, I use the index $u$ for users as raters, and $v$ for users as ratees. The goal of link strength prediction is to infer the value of $r_{u,v}$ given that we know $u$ and $v$ are friends.

Each user has two sets of direct friends or neighbors: First, $N^{rater}$ for the friends that the user has rated, $N^{ratee}$ for the friends that have given the user a rating. I translate transactional intensities between two friends into real numbers. The transaction intensities are stored in the **transaction matrices** $T^i_{N\times N}$. The expression $T^i_{u,v} = x$ denotes that $x$ is the intensity of transaction type $i$ between $u$ and $v$ that were carried out by $u$. In general, transactions are directed (e.g., sending a message), so $T^i$ is asymmetric. In this research, I focus on one transaction matrix $T_{N\times N}$; however, my techniques extend to multiple matrices. I use the terms transactions and activities interchangeably.

## 3.1 Non-Negative Matrix Factorization and Matrix Factorization with Baseline

According to [24], the standard way to perform Non-Negative Matrix Factorization(NNMF) on matrix R is

$$R = U \times V \tag{3.1}$$

where U has $m$ rows and $k$ columns and V has $k$ rows and $n$ columns which are called latent factors and R has $m$ rows and $n$ columns. All three of the matrices should contain only non-negative values. Usually $k$ is a number smaller than both $n$ and $m$.

The main goal of using NNMF is to minimize $||R-U \times V||^2$ by equation 3.2 which forces $U \times V$ get closer and closer to $R$. This difference vanishes only when $R$ equals $U \times V$.

$$||R - U \times V||^2 = \sum_{ijk} (R_{ij} - U_{ik} \times V_{kj})^2 \tag{3.2}$$

One of the examples of NNMF being used is the ~~Netflix~~'s competition for improving the movie recommender system's accuracy[11, 13]. This algorithm has been used because it is easy to implement and practical. Other algorithms can be found to be more efficient but they are harder to implement.

The basic idea of my data fusion model is to learn informative latent factors for each individual. Since the patterns of how a user rates others and how others rate him are usually distinct, I associate two sets of latent factors to each user. As an example, a popular movie star may be linked to many fans that indicate a strong tie to him, but he would probably indicate a weak tie to fans that he does not personally know. Let $U_{N\times K}$ be the latent factors modeling how users rate and let $V_{N\times K}$ be the latent factors modeling how users are rated. In my example, the V factors of the movie star indicate that he tends to be rated high and U factors of the movie star show that he tends to rate low. I employ matrix factorization techniques to learn the latent characteristics of users. Matrix factorization maps users to a joint latent factor space of dimensionality $k$, such that the rating between two users would be the inner products of their latent factor as shown in equation 3.3.

$$r_{uv} = {U_u}^T V_v \tag{3.3}$$

The dimension for both ~~Vectors~~ $U_u$ and $V_v$ is $K \times 1$ which shows that $r_{uv}$ is a ~~single value parameter.~~

The baseline or bias parameters $b_u^{rater}$ and $b_v^{ratee}$ indicate the observed deviation of user $u$ as rater and user $v$ as ratee from average $\mu$ over all expressed strengths in the training dataset. Each user is assigned two baseline parameters that help significantly with predictive performance [12]. For example, suppose user John wants to rate his friendship with user Mary. Assume that the average of the ratings is $\mu$ which is 3.4 and Mary is a popular user whom users rate 0.8 above average. On the other hand, John is a critical user who rates 0.3 below average. We would predict John will rate his friendship with Mary 3.4 + 0.8 - 0.3 = 3.9.

The baseline predictors may be integrated with the NNMF model shown in equation 3.4.

$$r_{uv} = \mu + b_u^{rater} + b_v^{ratee} + {U_u}^T V_v \tag{3.4}$$

## 3.2 Collective Matrix Factorization and Collective Matrix Factorization with Baseline

In Collective Matrix Factorization(CMF) model, I introduce two new matrices, $A$ and $B$. $A_a$ is a transaction matrix of all users $a$ whom user $u$ has transactions with. On the other side, $B_b$ like $A_a$ is a transaction matrix of users $b$ who have transaction with user $v$. I factorize these two matrices and the $R$ matrix as:

$$R = U^T V, E = U^T A, F = B^T V \tag{3.5}$$

Figure 3.1 shows the CMF plate model. By defining $E$ and $F$, we could define the conditional distribution over the observed transactions and ratings as

$$p(R|U, V, \sigma_R^2) = \prod_{u=1}^{N} \prod_{v=1}^{N} \left[ \mathcal{N}\Big( r_{u,v} | g(U_u^T V_v), \sigma_R^2 \Big) \right]^{I_{u,v}^r} \tag{3.6}$$

$$p(E|U, A, \sigma_E^2) = \prod_{u=1}^{N} \prod_{a=1}^{N_u} \left[ \mathcal{N}\Big( e_{u,a} | g(U_u^T A_a), \sigma_e^2 \Big) \right]^{I_{u,a}^e} \tag{3.7}$$

$$p(F|B, V, \sigma_F^2) = \prod_{b=1}^{M_v} \prod_{v=1}^{N} \left[ \mathcal{N}\Big( f_{b,v} | g(B_b^T V_v), \sigma_f^2 \Big) \right]^{I_{b,v}^f} \tag{3.8}$$

~~which~~ $N_u$ is a set of users that user $u$ has transaction with and $M_v$ is a set of users who have transactions with user $v$. $I_{u,v}^r$, $I_{u,a}^e$ and $I_{b,v}^f$ are indicator functions that are equal to 1 if user $u$ is a friend of $v$, $u$ has a transaction with $a$ and $b$ has a transaction with $v$, respectively. They are equal to 0 otherwise.

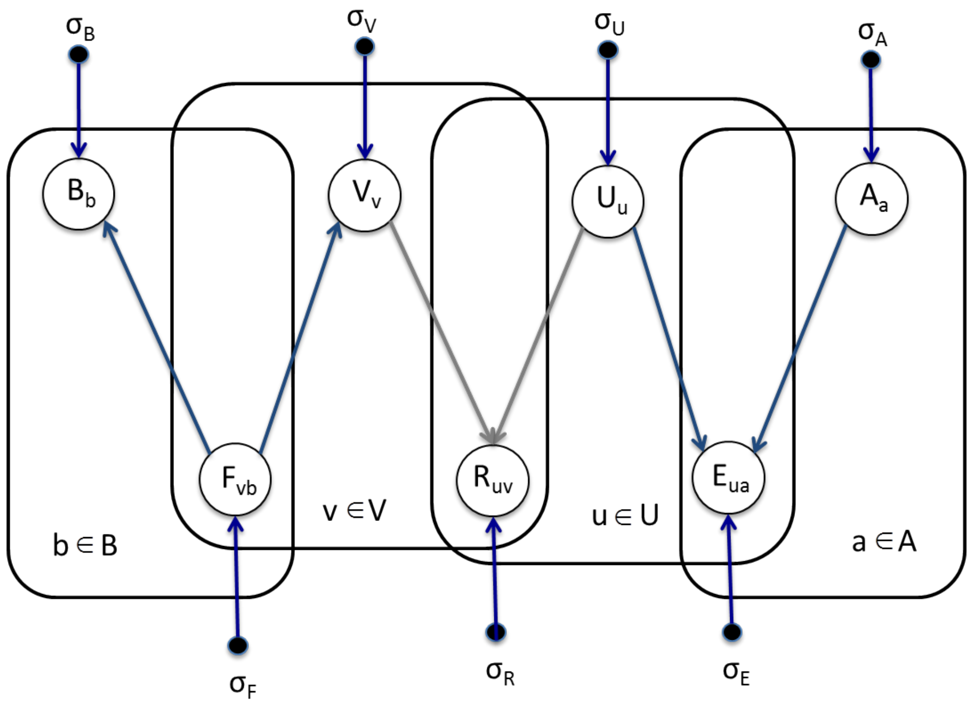


Figure 3.1: The CMF plate model.

As described in 3.1, we could use baselines for the CMF model as well(CMF+Base). By adding the baselines, the new distribution would be:

$$p(R|U,V,\sigma_R^2) = \prod_{u=1}^{N}\prod_{v=1}^{N}\left[\mathcal{N}\left(r_{u,v}|g(\mu + b_u^{rater} + b_v^{ratee} + U_u^T V_v), \sigma_R^2\right)\right]^{I_{u,v}^r} \tag{3.9}$$

## 3.3 Transaction-Based Matrix Factorization

After discussing the MF and MF+Base, now we want to add another parameter to predict the friendship which is the activities between users. In this section I go through the mathematical details of the proposed Transaction-Based Matrix Factorization(TMF) model. Basically I use a Gausian prediction model where the mean of the predictive distribution is a linear combination of the baseline predictors and the product of the latent factors.

I discuss how the baseline predictors $b_u^{rater}$ and $b_v^{ratee}$ and latent factors for $U$ and $V$ are learned given the strength of the friendship and transactional intensities between users.

The corresponding graphical model is presented in Figure 3.2 which can be computed using the following Gaussian,

$$p(R|U,V,\sigma_R^2) = \prod_{u=1}^{N}\prod_{v=1}^{N}\left[\mathcal{N}\left(r_{u,v}|g(\mu + b_u^{rater} + b_v^{ratee} + U_u^T V_v), \sigma_R^2\right)\right]^{I_{u,v}^r} \tag{3.10}$$

Here $\mathcal{N}(x|\mu,\sigma^2)$ is the normal distribution with mean $\mu$ and variance $\sigma^2$, and $I_{u,v}^r$ is the indicator function that is equal to 1 if $u$ has rated $v$ and equals to 0 otherwise. $g(x)$ is the logistic function that has the form $\frac{1}{1+e^{-x}}$. The logistic function is used in the seminal paper on probabilistic matrix factorization by Salakhutdinov and Mnih [22] and also by Jamali and Ester in the SMF model [7].

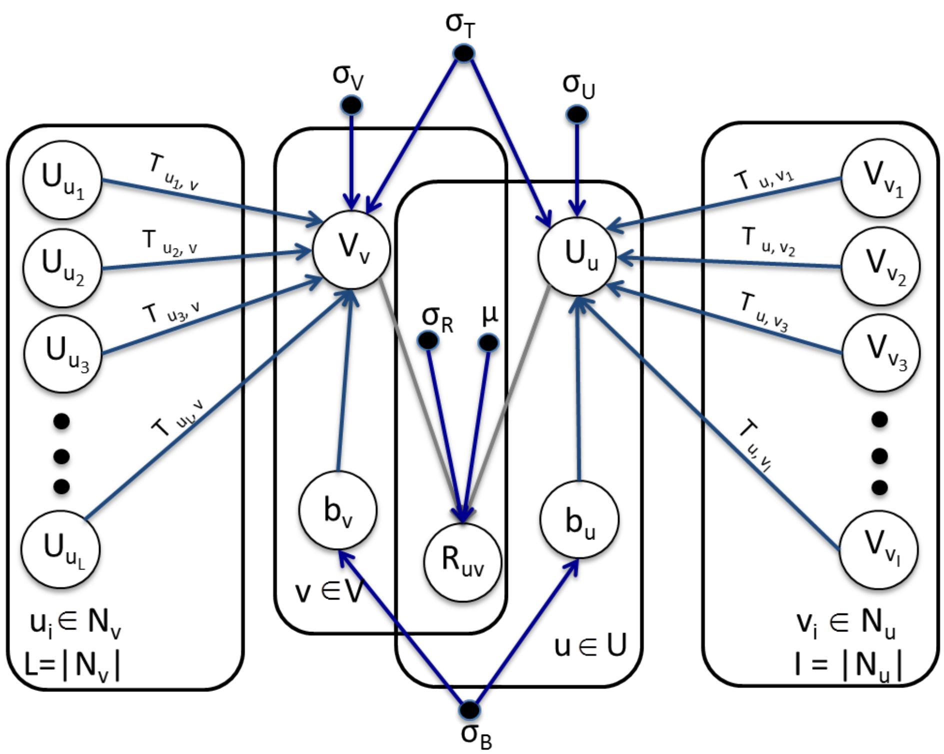


Figure 3.2: The Transaction-MF or (TMF) plate model. Observed quantities are the ratings $R_{uv}$ and the transaction intensities $T_{u,v}$.

**Learning** We first learn the baseline estimates without considering the latent factors, following [11]. The intuition is that the baseline terms capture the general rating trend of a rater against ratee, independently of the transaction intensities. In contrast, the latent factors $U$ and $V$ depend on both the ratings and the transaction intensities. In order to estimate $b_u^{rater}$ and $b_v^{ratee}$ for each user, we therefore maximize the log-likelihood of the ratings without considering the latent factors. Adding a L2-regularizer and trade-off parameter $\lambda_b$ then leads to minimizing the following objective:

$$\frac{1}{2}\sum_{u=1}^{N}\sum_{v=1}^{N} I_{u,v}^{r}\left(r_{u,v} - g(\mu + b_u^{rater} + b_v^{ratee})\right)^2 + \frac{\lambda_b}{2}\sum_{u=1}^{N} b_u^{rater2} + \frac{\lambda_b}{2}\sum_{v=1}^{N} b_v^{ratee2} \tag{3.11}$$

I use gradient descent to learn the baseline predictors. The following formulas give the update rules.

$$\begin{aligned} b_u^{rater} := b_u^{rater} - \gamma(&\sum_{v \in N_u} g'(\mu + b_v^{ratee} + b_u^{rater}) \\ &\times \left(g(\mu + b_v^{ratee} + b_u^{rater}) - r_{uv}\right) + \lambda_b b_u^{rater}) \end{aligned} \tag{3.12}$$

$$\begin{aligned} b_v^{ratee} := b_v^{ratee} - \gamma(&\sum_{u \in N_v} g'(\mu + b_v^{ratee} + b_u^{rater}) \\ &\times \left(g(\mu + b_v^{ratee} + b_u^{rater}) - r_{uv}\right) + \lambda_b b_v^{ratee}) \end{aligned} \tag{3.13}$$

Similar to [7], I make some simplifying assumptions to approximate the optimal latent factors. (1) The rater factors $U$ and ratee factors $V$ are mutually dependent. We approximate their joint distribution as the product of the conditional distributions. (2) The conditional distribution of one latent factor conditional on the model parameters and the other latent factors are split into two terms: (i) the shrinkage prior and (ii) the conditional distribution given the latent factors of friends. (3) The latent factor that describes how a user rates his friends depends linearly on the latent factors that describe how his friends are rated, with the transaction intensities between him and his friends as weights.

We can apply Bayes' theorem and the assumptions stated to obtain a tractable approximation to the posterior probability of the latent variables $U$ and $V$, shown in the following formula.

$$
\begin{aligned}
p(U,V|R,T,\sigma_R^2,\sigma_T^2,\sigma_U^2,\sigma_V^2) \approx\ & p(R|U,V,\sigma_R^2)\times p(U|V,T,\sigma_U^2,\sigma_T^2)\times p(V|U,T,\sigma_V^2,\sigma_T^2)\\
=\ & \prod_{u=1}^{N}\prod_{v=1}^{N}\left[\mathcal{N}\Big(r_{u,v}|g(\mu+b_u^{rater}+b_v^{ratee}+U_u^TV_v),\sigma_R^2\Big)\right]^{I^r_{u,v}}\\
&\times\prod_{u=1}^{N}\mathcal{N}(U_u|\sum_{v\in N_u}T_{u,v}V_v,\sigma_T^2\mathbf{I})\times\prod_{u=1}^{N}\mathcal{N}(U_u|0,\sigma_u^2\mathbf{I})\\
&\times\prod_{v=1}^{N}\mathcal{N}(V_v|\sum_{u\in N_v}T_{u,v}U_u,\sigma_T^2\mathbf{I})\times\prod_{v=1}^{N}\mathcal{N}(V_v|0,\sigma_v^2\mathbf{I})
\end{aligned}
\tag{3.14}
$$

The log of the approximate posterior probability can be computed using the following formula. In the following formula $\lambda_U = \sigma_R^2/\sigma_U^2$, $\lambda_V = \sigma_R^2/\sigma_V^2$ and $\lambda_T = \sigma_R^2/\sigma_T^2$.

$$
\begin{aligned}
L(R,T,U,V) =\ & \frac{1}{2}\sum_{u=1}^{N}\sum_{v=1}^{N}I^r_{u,v}\Big(r_{u,v}-g(\mu+b_u^{rater}+b_v^{ratee}+U_u^TV_v)\Big)^2\\
&+\frac{\lambda_U}{2}\sum_{u=1}^{N}U_u^TU_u+\frac{\lambda_V}{2}\sum_{v=1}^{N}V_v^TV_v\\
&+\frac{\lambda_T}{2}\sum_{u=1}^{N}\Big((U_u-\sum_{v\in N_u}T_{u,v}V_v)^T(U_u-\sum_{v\in N_u}T_{u,v}V_v)\Big)\\
&+\frac{\lambda_T}{2}\sum_{v=1}^{N}\Big((V_v-\sum_{u\in N_v}T_{u,v}U_u)^T(V_v-\sum_{u\in N_v}T_{u,v}U_u)\Big)
\end{aligned}
\tag{3.15}
$$

This formula is the main objective function with the predictive baselines and transactional intensities included. Taking the derivative with respect to $U_u$ and $V_v$ yields gradient descent update formulas for the baseline predictors and latent factors.

$$
\begin{aligned}
\frac{\partial L}{\partial U_u} =\ & \sum_{v=1}^{N}I^r_{u,v}\Big[V_vg'(\mu+b_u^{rater}+b_v^{ratee}+U_u^TV_v)\\
&\times\Big(g(\mu+b_u^{rater}+b_v^{ratee}+U_u^TV_v)-r_{u,v}\Big)\Big]\\
&+\lambda_UU_u+\lambda_T(U_u-\sum_{v\in N_u}T_{u,v}V_v)+\lambda_T\Big(\sum_{v\in N_u}T_{u,v}(U_uT_{u,v}-V_v)\Big)
\end{aligned}
\tag{3.16}
$$

$$\begin{aligned}\frac{\partial L}{\partial V_v} = \sum_{u=1}^{N} I^r_{u,v} \Big[ & U_u g'(\mu + b_u^{rater} + b_v^{ratee} + U_u^T V_v) \\ & \times \left( g(\mu + b_u^{rater} + b_v^{ratee} + U_u^T V_v) - r_{u,v} \right) \Big] \\ & + \lambda_V V_v + \lambda_T (V_v - \sum_{u \in N_v} T_{u,v} V_v) + \lambda_T \left( \sum_{u \in N_v} T_{u,v} (V_v T_{u,v} - U_u) \right)\end{aligned} \tag{3.17}$$

Algorithm 1 shows the pseudocode of the main algorithm.

**Algorithm 1** Main Algorithm

**procedure** MAIN($friendShipFile, activityFile$)
    read friendship test and train files
    initialize u and v
    initialize baselines and calculate $\mu$
    read activities from file
    bestError := FLOAT MAX
    newError := 0
    **while** difference between bestError and newError > $\epsilon$ **do**
        update u and v
        calculate the new error
        **if** newError < bestError **then**
            bestError := newError

## 3.4 Optimization Using Gradient Descent

One of the most common ways for optimization is gradient descent. It is easy to implement and scales for large datasets. Generally, the idea is to find the minimum point of a function $f(x)$, by moving step by step in the negative direction of the gradient of $f(x)$. Algorithm 2 shows the steps of this method.

Figure 3.3 shows a simple example of an execution of gradient descent method by starting from a ‘first guess’ point and moving towards the minimum. Choosing the step size is a critical matter in gradient descent. Choosing small steps leads to slow execution, which could make it infeasible for large datasets. Choosing big steps may produce an inaccurate result. Choosing big steps at first and reducing the step size as getting closer to the answer is the best approach. Figure 3.4 shows two different examples of step choosing.[18]

**Algorithm 2** Gradient Descent

**procedure** GRADIENTDESCENT
    Guess an initialization point
    **while** The difference between current step and last step is bigger than epsilon **do**
        Find the descent direction
        Choose a step
        Update your current state

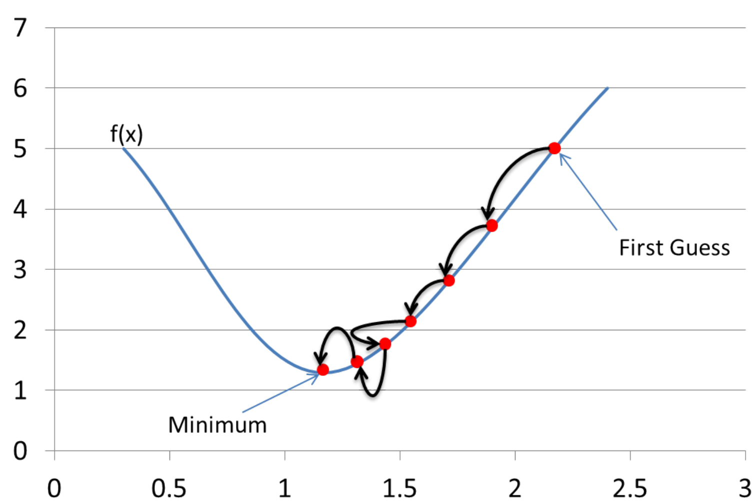


Figure 3.3: Simple example of gradient descent. Starting from the 'First Guess' point and moving towards the 'Minimum' point.

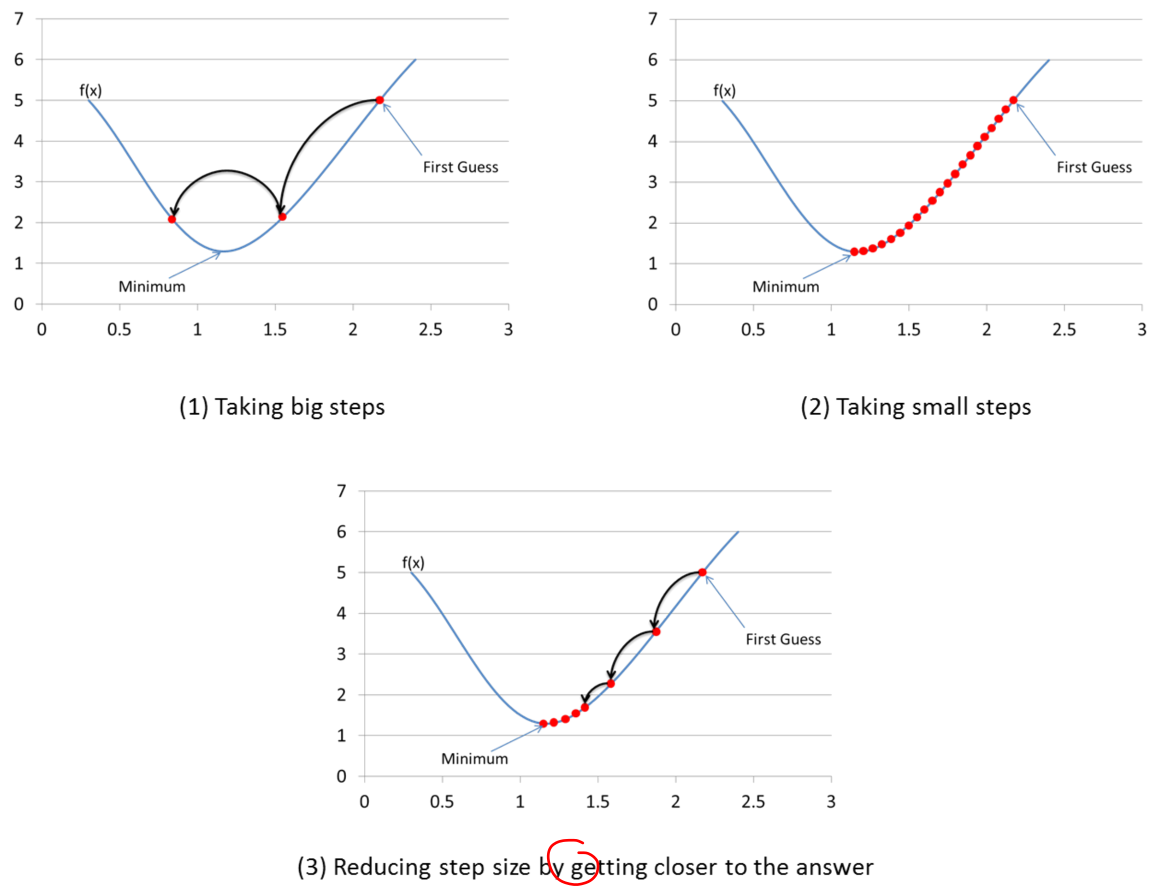


Figure 3.4: Choosing small steps decreases the efficiency (1) and choosing big steps decreases the accuracy (2). Reducing the step size when getting closer to the result is the most efficient approach(3).

# Chapter 4

# Datasets and Cross Validation

## 4.1 Datasets

A real world data from Cloob and a synthetic dataset are used to validate the model and compare it with different models.

### 4.1.1 Cloob.com

As explained before, a special feature of Cloob is that users can enter a value between 0 and 5 to indicate the strength of a friendship, which provides us with a ground truth for evaluating link strength models. Since the website does not require users to rate their friends, many of the friendships have the default rating. I omit the ties with default value from my experiments and concentrate on ties of other strength because firstly the strength of default friendships are unknown since they contain both strong and weak ties and secondly due to the large skew on the target class, including the default values in the learning procedure of the latent factors, makes the model perform poorly in predicting other values of strength. After removing all friendship ratings with the default value, the strengths are between 1 and 5.

Table 4.1 shows the number of friendships and activities after removing the default values. I evaluate the ratings given by users that have rated at least five of their friendships, and the ratings given to users that have been rated at least five times. The network contains 20 million transactional events between 2005 to 2010. Users can apply five types of activities to one another: comment, paint, send pictures, like, and write testimonial. To convert information about the five types of activities into a single entry in a transaction intensity matrix, I add up the five activity counts for every pair of friends. An interesting extension for future work would be to learn with five separate activity matrices.

| | |
|---|---|
| Number of friendships | 502,060 |
| Number of Users | 45,150 |
| Total Activity Count | 7,913,596 |

Table 4.1: Number of friendships and activities after removing the default values for the Cloob dataset.

### 4.1.2 Synthetic Data

The idea behind this research is that the implicit network created from the transactional intensities among users correlates with explicit network generated by the users ratings. I first generate a synthetic dataset in which there is a very strong correlation between strengths of ties and the intensity of transaction between users. In this dataset, the following two assumptions are exaggerated compared to what you expect in a real online social network. (1) All of the friendship ties generated in this dataset are directly dependent on the number of transactions. This assumption does not generally hold as two users $u$ and $v$ may have known each other for a long time, leading to a strong tie, but have only recently become friends in an online social network, so they don't have many recorded transactions. (2) The average number of activities between users is much higher than what it normally is; there are usually many ties in social networks with zero or very few activities. The exaggerations are introduced to create a dataset where the basic assumption of the TMF model is true, so my expectation is that if the model assumptions are true, my learning algorithm for TMF should outperform other methods. This dataset is created using the following method.

First I generate users and their friendships using four distributions. The first two distributions are used to generate the baseline values $b_u^{rater}$ and $b_v^{ratee}$ for each user. We assume that each user could have one of three different types of behavior when defining strength for a friendship; they could assign higher than, same as, or lower than average strength for their friendships. The same three behaviors could be assumed for how they have been rated. I assign a uniform distribution over the nine different types of people. The following formulas show how the $b_u^{rater}$ and $b_v^{ratee}$ are generated for users using the normal distribution. The mean value of the distributions depend on the user's type, which is -1, 0, or 1:

$$
\begin{aligned}
b_u^{rater} &\sim \mathcal{N}(\{-1, 0, +1\}, \sigma_b^2) \\
b_v^{ratee} &\sim \mathcal{N}(\{-1, 0, +1\}, \sigma_b^2)
\end{aligned}
\tag{4.1}
$$

I then introduce a third parameter $acc_u$ for users to generate their tendency to make friends. This parameter is the probability of acceptance of friendships for users. To generate friendships, I randomly sample two users $u$ and $v$ and produce a random value $x$ between 0 and 1. If $x$ is smaller than the $acc_u$, the link is kept, otherwise I repeat the step and sample two other users. I set the acceptance parameter to achieve a power law distribution on the number of friendships [21].

Finally, the rating between users is randomly generated according to the following distribution:

$$r_{uv} \sim \mathcal{N}(\mu + b_u^{rater} + b_v^{ratee} + a_{uv}, \sigma_R^2), \tag{4.2}$$

The $a_{uv}$ parameter produces the number of activities between users $u$ and $v$, distributed as $a_{uv} \sim max(\mathcal{N}(\mu_a, \sigma_a^2), 0)$.

Thus the number of activities between user pairs in social networks is modelled with a normal distribution. For simplicity, only one type of activity may be carried out between users. The minimum number of activities between users is set to 0 to avoid a negative number of transactions. In my experiment I arbitrarily set $\sigma_b^2 = 0.5$, $\sigma_R^2 = 0.1$, $\mu = 3$, and $\mu_a = 25$.

Table 4.2 shows the number of generated friendships and activities and table 4.3 summarizes the parameters used for the synthetic data set.

| | |
|---|---|
| Total friendship links | 50000 |
| Total users | 27000 |
| Total activities | 50000 |

Table 4.2: Number of friendships and activities for the synthetic dataset.

| Parameter | Interpretation | Generation Method |
|---|---|---|
| $\mu$ | Average frienship rating in population | Set to 3 |
| $\sigma_b^2$ | Variance for generating baseline predictors for each user | Set to 0.5 |
| $b_u^{rater}$ | observed deviation of user u as rater from $\mu$ | $\mathcal{N}(\{-1,0,+1\},\sigma_b^2)$ |
| $b_v^{ratee}$ | observed deviation of user v as rated from $\mu$ | $\mathcal{N}(\{-1,0,+1\},\sigma_b^2)$ |
| $\mu_a$ | Average number of activities between two users | Set to 25 |
| $\sigma_a^2$ | Variance for generating activities between a pair of users | Set to 15 |
| $a_{uv}$ | Number of activities between user u and v | $max(\mathcal{N}(\mu_a,\sigma_a^2),0)$ |
| $acc_u$ | Acceptance rate of friendships for users | Set to 0.9,0.7,0.5,0.3,0.1 |
| $x$ | To decide whether a friendship link is kept or not | Randomly generated between 0 and 1 |
| $\sigma_R^2$ | Variance for generating rating between two users | Set to 0.1 |
| $r_{uv}$ | Rating between user u and v | $\mathcal{N}(\mu + b_u^{rater} + b_v^{ratee} + a_{uv}, \sigma_R^2)$ |

Table 4.3: Parameters used for generating the synthetic data

## 4.2 Evaluation using K-Fold Cross Validation

One of the most used approaches to estimate the accuracy of a predicting model is K-fold cross validation. In K-fold cross validation, we use $K - 1/K$ fraction of the dataset as our training set. A training set is a set that we run our experiments over and over on and learn the system from. To check the accuracy of the learning method, we run the code on the validation set, the remaining data. By doing this approach for $K$ times, each time $1/K$ as the validation set and the rest as the training set, we could calculate an average error for our learning method.


Figure 4.1: An example of selecting parts of a dataset using k-fold cross validation. In this example k = 4. [1]

# Chapter 5

# Experiments

## 5.1 Experimental Setup

The evaluation metric used in my experiments is $RMSE = \sqrt{\frac{\sum_{(u,i)\in R_{test}}(r_{u,v}-\hat{r}_{u,v})^2}{|R_{test}|}}$, where $R_{test}$ is the set of all pairs (u, v) in the test data. The error is evaluated at parameter settings obtained after gradient descent has converged. For different models and methods, convergence is attained after different numbers of iterations, which I also report. Note that by running the gradient descent until convergence I focus on comparing the predictive power of the models, rather than the computational difficulty of optimizing parameters in each. As explained in Algorithm 1, $u$, $v$ and the baselines should be initialized at the beginning of the experiment. These variables are initialized randomly between 0 and 1 in all of the experiments unless said otherwise. I compare the following seven methods.

1. **LRT**: Linear Regression on Transactions (LRT) learns a feature for each of the five transactions to predict the strength of the friendships. Weka linear regression package [5] has been used for this experiment.

2. **MF**: Uses Matrix Factorization to learn the latent factors without using the baseline predictors or activities as described in section 3.1.

3. **MF+Base**: Uses Matrix Factorization to learn the latent factors using the baseline predictors, but not using activities, as described in section 3.1. [13]

4. **LRT+MF**: A weighted combination of the LRT and the MF model. The weights are learned using regression, again using Weka.

5. **CMF**: Collective Matrix Factorization uses gradient descent to learn hidden factors to explain both the strength of the links and the transactional intensities as described in section 3.2.

6. **CMF+Base**: The addition of the baseline predictors to Collective matrix factorization as described in section 3.2.

7. **TMF**: Transaction-based Matrix factorization is my model discussed in section 3.3.

Table 5.1 lists the parameters and their meaning used in this evaluation. Since the focus of this research is not on the prior probabilities, the values of $\lambda_b = 0.001$, $\lambda_u = 0.001$ and $\lambda_v = 0.001$ are fixed for all experiments.

| Parameter | Interpretation | Estimation Method |
|---|---|---|
| $\mu$ | Average friendship rating in population | Empirical Mean |
| $\lambda_b$ | L2-regularizer for baseline predictors | Fixed for all methods |
| $\lambda_u$ | L2-regularizer for raters | Fixed for all methods |
| $\lambda_v$ | L2-regularizer for ratees | Fixed for all methods |
| $\lambda_t$ | L2-regularizer for transaction intensities | Evaluated with different values |
| $k$ | dimensionality of the latent factors | Evaluated with different values |
| $b_u^{rater}$ | observed deviation of user u as rater from $\mu$ | Gradient descent |
| $b_v^{ratee}$ | observed deviation of user v as rated from $\mu$ | Gradient descent |
| $U_u$ | latent factors modeling the behavior of raters | Gradient descent |
| $V_v$ | latent factors modeling the behavior of ratees | Gradient descent |

Table 5.1: The interpretation and the methods of estimation for all the parameters used in this research.

## 5.2 Impact of Transaction Information

It is interesting to study the effect of the transactional intensities on the link strength. Parameter $\lambda_T$ in the TMF model controls the influence of the transactional intensities. Setting $\lambda_T = 0$ totally ignores the transactions to make the model similar to the MF+Base model. Large values of $\lambda_T$ indicate a strong influence of the transaction information used to derive predictive features, moving close to the LRT model. Figures 5.1 and 5.2 shows the transactional intensities' influence. Figure 5.2 illustrates that TMF does much better on the synthetic dataset, which was to be expected since the data satisfies the assumptions of the model. The MF methods do not perform well because they fail to make use of the transaction information. Figure 5.1 shows that methods that incorporate base-line predictors do well on the Cloob dataset. In recommendation problems like the Netflix challenge, the MF+Base approach is a state-of-the-art method, so its strong performance is to be expected. Still, data fusion with the TMF model leads to further improvement. In both datasets, we observe that for the data fusion methods (TMF and CMF), as we increase $\lambda_T$, the performance first improves up to a point and then starts to drop. Therefore, the transactional intensities are helpful with link strength; however, if too much weight is assigned to them, they would override the effects of the friendship ratings. For the parameters of MF and TMF I used the latent factor dimension $k = 5$.

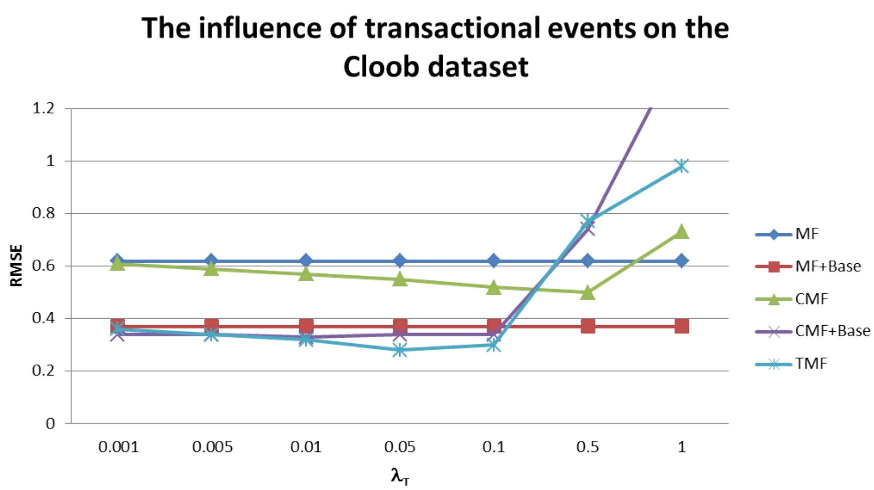


Figure 5.1: Transactional intensities influence on link strength in the Cloob dataset.

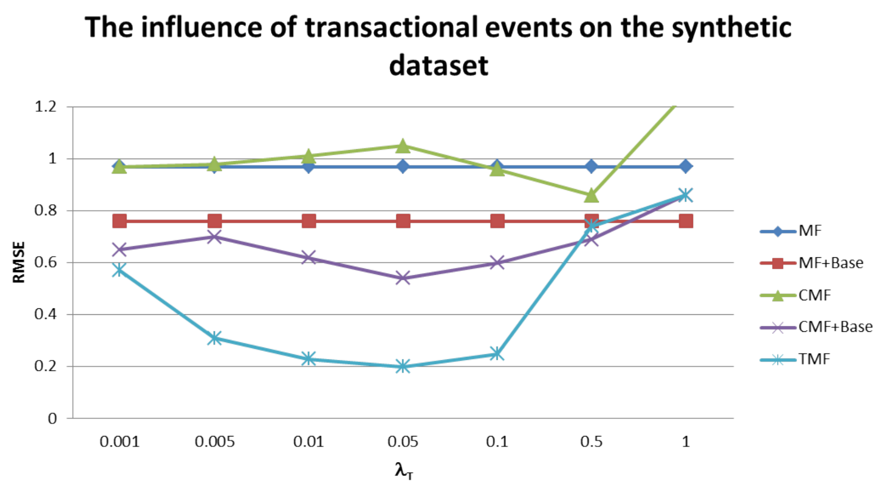


Figure 5.2: Transactional intensities influence on link strength in the synthetic dataset.

## 5.3 Data Fusion Experiments

Data fusion, as discussed in section 2.6.2, is the process of combining data from multiple sources for analysis instead of using each of the sources individually. Among my comparison methods, the CMF and TMF approaches can be viewed as performing data fusion, whereas the other approaches use only one information source (LRT, MF), or model the information sources independently (LRT+MF). I perform experiments to evaluate whether a data fusion approach leads to more accurate predictive performance or not. Table 5.2 illustrates the results for this experiment on the Cloob dataset.

For the parameters of MF and TMF I used the latent factor dimension $k = 5$ and the optimal value of $\lambda_t$, as established by cross-validation. LRT assigns very low weights to each of the activities and predicts values very close to the mean of the strengths of ties, which indicates that the activities are almost independent of the strength of ties. This occurs mainly because many pairs of friends lack any transactions—I refer to this situation as a *zero-transaction* friendship. The MF model performs better than LRT and is able to find factors that perform reasonably well in the dataset. LRT+MF outperforms both LRT and MF model by using the information from both matrices separately. The TMF approach uses data fusion and leads to the best performance compared to learning weighted combination of the two models.

| Method | RMSE |
|---|---|
| LRT | 1.17 ± 0.012 |
| MF | 0.61 ± 0.023 |
| MF + LRT | 0.47 ± 0.019 |
| CMF | 0.50 ± 0.015 |
| TMF | **0.28 ± 0.008** |

Table 5.2: Data fusion results on Cloob dataset.

**Zero-Transaction Friendships** To investigate further the importance of data fusion, I separately report results regarding zero-transaction friendships. Such friendships are numerous; the Cloob database records 329,940 of them, or 65% of all friendships. In this case a model based on explicit transaction information does not apply, whereas the effect of latent factor learning is to propagate information from different third-party transactions. For example, if Jack and James are friends with no transactions between them, but Jack has recorded transactions with 20 other friends, and James has recorded transactions with 30 other friends, this information will influence the latent factors for Jack and James, and hence the prediction of the link strength between them. Table 5.3 shows the cross-validation RMSE for zero-transaction friendship pairs.

| Method | Result |
|---|---|
| MF | 0.55 ± 0.032 |
| MF-Base | 0.31 ± 0.023 |
| CMF | 0.47 ± 0.012 |
| CMF-Base | 0.31 ± 0.041 |
| TMF | **0.24 ± 0.014** |

Table 5.3: Performance on zero-transaction user pairs

## 5.4 Result Improvements

Two different ways have been tested for improving the results. Changing the number of latent factors and initializing using shrinkage.

### 5.4.1 Number of Latent Factors

In this section, I examine the impact of the dimensionality of the latent factors $k$. In experiments on other subsections the $\lambda_T$ with the best performance is used. In experiments on Tables 5.4 and 5.5 report the average RMSE and the number of iterations for different dimensions of the latent factors.

| Model | k = 2 | | k = 5 | | k = 10 | |
|---|---|---|---|---|---|---|
| | RMSE | itr | RMSE | itr | RMSE | itr |
| MF | 0.67 ± 0.045 | 143 | 0.61 ± 0.023 | 88 | 0.71 ± 0.078 | 110 |
| MF+Base | 0.40 ± 0.014 | 246 | 0.35 ± 0.025 | 284 | 0.38 ± 0.030 | 331 |
| CMF | 0.90 ± 0.010 | 269 | 0.50 ± 0.015 | 422 | 2.52 ± 0.239 | 300 |
| CMF+Base | 0.38 ± 0.018 | 299 | 0.33 ± 0.010 | 308 | 0.36 ± 0.005 | 662 |
| TMF | **0.35 ± 0.015** | 403 | **0.28 ± 0.008** | 627 | **0.22 ± 0.032** | 1165 |

Table 5.4: Average RMSE and number of iterations for different dimensionality in the Cloob dataset.

| Method | k = 1 | | k = 5 | | k = 10 | |
|---|---|---|---|---|---|---|
| | RMSE | itr | RMSE | itr | RMSE | itr |
| MF | 0.80 ± 0.026 | 191 | 0.97 ± 0.283 | 344 | 3.37 ± 0.299 | 445 |
| MF+Base | 0.65 ± 0.011 | 258 | 0.76 ± 0.217 | 485 | 2.71 ± 0.310 | 678 |
| CMF | 0.70 ± 0.007 | 115 | 0.86 ± 0.174 | 434 | 2.52 ± 0.239 | 300 |
| CMF+Base | 0.64 ± 0.011 | 283 | 0.54 ± 0.091 | 389 | 2.58 ± 0.190 | 586 |
| TMF | **0.61 ± 0.012** | 295 | **0.20 ± 0.002** | 340 | **0.21 ± 0.003** | 413 |

Table 5.5: Average RMSE and number of iterations for different dimensionalities in the synthetic dataset.

The number of iterations seems quite reasonable for a gradient descent method, especially considering the high number of hidden factors to be assigned. On both datasets, the TMF model performs best on each dimension.

On the synthetic dataset, MF, MF+Base, and CMF have their best performance when $k = 1$. This is because the data size is small and bigger values of $k$ lead to ~~over fitting~~. CMF+Base and TMF have their best performance on $k = 5$. On the cloob dataset, since dataset is much larger, higher dimensionality helps all of the models. MF, MF+Base, CMF, and CMF+Base have their best performance when $k = 5$. The performance of TMF improves with dimensionality, which suggests that the model is well-suited to the dependencies in the data.

### 5.4.2 Initializing $u$ and $v$

Another way to improve the results is to change the initialization values of $u$ and $v$ from random to more meaningful values. In order to find a better starting point for vectors $u$ and $v$, we can use shrinkage [19] which is a well-understood technique in statistics. ~~for~~ each pair of users $U$ and $V$, there is a link strength and set of five different types of activities. The simplest model for regression is the linear combination of the activities. Equation 5.1 shows the formula for this model for latent variable $k = 5$.

$$r = w_0 + w_1ac_1 + w_2ac_2 + w_3ac_3 + w_4ac_4 + w_5ac_5 \tag{5.1}$$

Matrix $w$ could be calculated using the following algorithm [1].

**Algorithm 3** Calculating weight matrix w

```
1: procedure CALCULATEWEIGHTS(activityFile, friendshipFile)
2:     activities = load all activities from activityFile
3:     friendships = load all friendships from friendshipFile
4:     weights = MoorePenrosePseudoInverse(activities) * friendships
```

Using algorithm 3, global weights ($w^G$) and two local weights for each user A($w_A^{rater}$ and $w_A^{ratee}$) can be calculated. Global weights $w^G$ shows the relation between activities and ratings in the whole network. For example, in Cloob.com, sending a picture to a friend shows more closeness of their friendship than writing on their wall. In this case, the weight of *sending picture activity* is more than the weight of *writing on wall activity.* Local weights can be defined like global weights but they are only for a specific user instead of the whole network. These weights can be used as initialization values for matrices $u$ and $v$. The intuition is that if different activity weights are appropriate for predicting the rating behavior of two users, then these users are likely to be of a different type. Experiments are done for MF, MF+Base and TMF with the following two initialization methods for latent variable $k = 5$.

### Initialization using Local Weights

Each user $A$ has two sets of weights, as a rater and a ratee. These two vectors $u_A$ and $v_A$ will be initialized using the following equation:

$$
\begin{aligned}
u_A[i] &= w_A^{rater}[i] \\
v_A[i] &= w_A^{ratee}[i]
\end{aligned}
\tag{5.2}
$$

where i indicates the index of the element in a vector. $b_A^{rater}$ and $b_A^{ratee}$ will be also calculated as an average of the initiated values of $u_A$ and $v_A$, respectively. Table 5.6 shows the best RMSE in experimenting with different $\lambda_t$ for different models.

| Model | k = 5 | |
|---|---|---|
| | RMSE | $\lambda_T$ |
| MF | $0.22 \pm 0.012$ | NA |
| MF+Base | $0.16 \pm 0.004$ | 0.1 |
| TMF | $0.15 \pm 0.005$ | 0.01 |

Table 5.6: Initializing $u$ and $v$ using local weights

Figure 5.3 shows the results of the experiments using local weights as the initial values for $u$ and $v$.

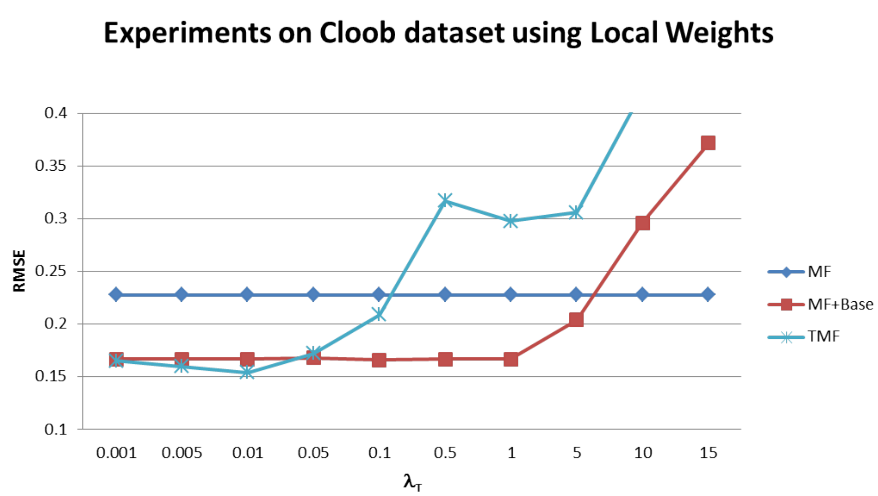


Figure 5.3: Results of initializing $u$ and $v$ using local weights

As the results show, TMF still is the most accurate approach for $k = 0.01$.

### Initialization using Shrinkage

Another way to initialize $u$ and $v$ is to use both local and global weights. This time we initialize $u$ and $v$ as follows:

$$\begin{aligned} u_A[i] &= w^G - w_A^{rater}[i] \\ v_A[i] &= w^G - w_A^{ratee}[i] \end{aligned} \tag{5.3}$$

Table 5.7 shows the best RMSE of different models with initializing using shrinkage.

| Model | k = 5 | |
|---|---|---|
| | RMSE | $\lambda_T$ |
| MF | $0.26 \pm 0.009$ | NA |
| MF+Base | $0.16 \pm 0.002$ | 0.5 |
| TMF | $0.15 \pm 0.004$ | 0.01 |

Table 5.7: Using shrinkage to initialize $u$ and $v$

Figure 5.4 shows the result of running the experiments by initializing $u$ and $v$ using the local and global weights.

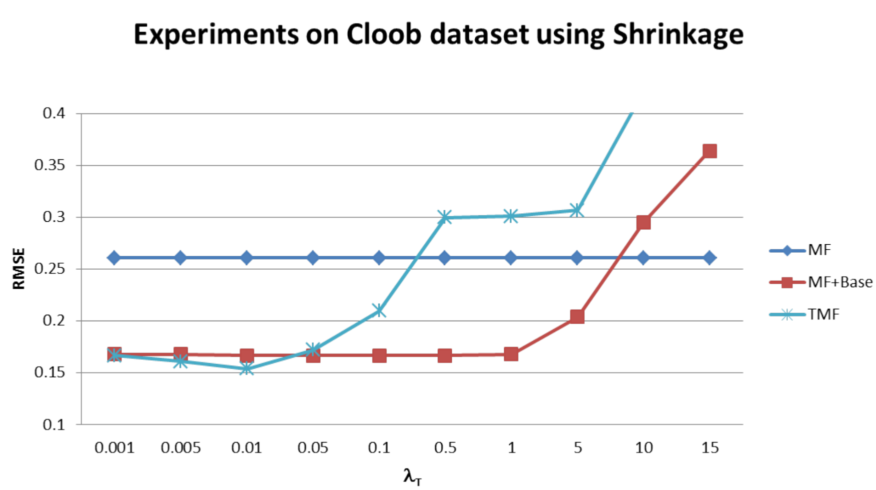


Figure 5.4: Results of initializing $u$ and $v$ using shrinkage

This experiment shows the same results as using only local weights for initializing $u$ and $v$.

## 5.5 Summary

Eight different models have been introduced and used in various experiments in this chapter. The Transaction-Based Matrix Factorization model has shown the best result in all the experiments for both Cloob and the synthetic dataset. Two different methods were introduced to improve the results. First, changing the value of the latent variable $k$. Effectiveness of changing $k$ always depends on the dataset. For example, as seen in table 5.4, increasing $k$ from 5 to 10 has ~~lead~~ to a less accurate result but for the synthetic dataset, result for $k = 10$ is more accurate than $k = 5$. In both case, as shown in table 5.4 and 5.5, TMF is the most accurate model even with changing the value of $k$. Second, changing the initialization method of $u$, $v$ and the baselines. Two different ways explained for initializing the main algorithm. As shown in Figure 5.3 and 5.4, still TMF has the best result among the other three methods.

# Chapter 6

# Conclusion and Future Works

## 6.1 Conclusion

I introduce a new matrix factorization model where the goal is to predict the values of a single explicit matrix, and implicit matrices are used to weight the importance of connections between users. I derived gradient descent equations for learning two sets of latent factors for each user, as a rater and as a ratee. In experiments on generated synthetic data and on real-world data, my transaction-MF model outperforms collective matrix factorization, single table matrix factorization, as well as regression models that analyze the data matrices separately.

## 6.2 Future Works

1. The experiments of this research *aggregated* the values of different transactional intensities to create a single implicit network. This method has information loss and may be inappropriate as the effect of more frequent type of activities can overcome the effect of more important, yet less frequent type of activities. I will investigate the use of multiple implicit networks with different weights in the future.

2. Online social networks, in addition to the links, often contain useful temporal information. The temporal behavior of users may play an important role in link strength prediction. Consider a retired user that has not had any activities or new friendships

over a long period, but used to be a very active user. Without considering the temporal side of his behavior, the false prediction is made that this user is very likely to obtain new strong friendships. The Cloob dataset contains timestamps, and supports investigation of temporal information for better link strength prediction.

3. Link prediction models often combine latent features with observed features [6, 20]. Observed features may include descriptive attributes of users (e.g., culture, behavior, personality), and features derived from the link topology (e.g.,number of friends in common). A promising direction is to combine latent and observed features for link strength prediction.

# Bibliography


[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., ~~Secaucus~~, NJ, USA, 2006.

[2] Michael Conover, Bruno Gonçalves, Alessandro Flammini, and Filippo Menczer. Partisan asymmetries in online political activity. *CoRR*, abs/1205.1010, 2012.

[3] Eric Gilbert. Predicting tie strength in a new medium. In *CSCW '12*, pages 1047–1056. ACM, 2012.

[4] Eric Gilbert and Karrie Karahalios. Predicting tie strength with social media. In *CHI*, pages 211–220, 2009.

[5] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[6] P. D. Hoff. Multiplicative latent factor models for description and prediction of social networks. *Computational and Mathematical Organization Theory*, 2007.

[7] Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*, pages 135–142, 2010.

[8] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, WebKDD/SNA-KDD '07, pages 56–65, New York, NY, USA, 2007. ACM.

[9] Indika Kahanda and Jennifer Neville. Using transactional information to predict link strength in online social networks. In *ICWSM*, 2009.

[10] F. Kooti, W.A. Mason, K.P. Gummadi, and M. Cha. Predicting emerging social conventions in online social networks. 2012.

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

[12] Yehuda Koren. Collaborative filtering with temporal dynamics. *Commun. ACM*, 53(4):89–97, 2010.

[13] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.

[14] J. Kulshrestha, F. Kooti, A. Nikravesh, and K.P. Gummadi. Geographic dissection of the twitter network. In *In Proceedings of the 6th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2012.

[15] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *Proceedings of the twelfth international conference on Information and knowledge management*, CIKM '03, pages 556–559, New York, NY, USA, 2003. ACM.

[16] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.*, 58:1019–1031, May 2007.

[17] Hao Ma, Irwin King, and Michael R. Lyu. Learning to recommend with social trust ensemble. In *SIGIR*, pages 203–210. ACM, 2009.

[18] Alexandre M. M.Bayen. Lecture on steepest descent.

[19] Andrew McCallum, Ronald Rosenfeld, Tom M. Mitchell, and Andrew Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 359–367, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[20] A. Menon and C. Elkan. Link prediction via matrix factorization. *Machine Learning and Knowledge Discovery in Databases*, pages 437–452, 2011.

[21] M. E. J. Newman. Power laws, Pareto distributions and Zipf's law, December 2005.

[22] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *ICML*, pages 880–887, 2008.

[23] Ajit P. Singh and Geoffrey J. Gordon. Relational learning via collective matrix factorization. KDD '08, pages 650–658. ACM, 2008.

[24] David Skillicorn. *Understanding Complex Datasets: Data Mining with Matrix Decompositions*. Chapman and Hall/CRC; 1 edition, 2007.

[25] R. Xiang, J. Neville, and M. Rogati. Modeling relationship strength in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 981–990. ACM, 2010.

[26] Shuang-Hong Yang, Bo Long, Alex Smola, Narayanan Sadagopan, Zhaohui Zheng, and Hongyuan Zha. Like like alike: joint friendship and interest propagation in social networks. WWW '11, pages 537–546.