A Personalized Recommender System with Correlation Estimation

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Fan Yang

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Xiaotong Shen

May, 2018

Acknowledgements

First and foremost, I would like to take this opportunity to express my greatest appreciations to my thesis advisor Prof. Xiaotong Shen. Over the years, he has been generously sharing his time and knowledge, and offering me invaluable advice in both my academic research and personal development. I'm deeply grateful and honored to be influenced by his insights and professionalism. It is his constant support, help, and patience for various aspects of my life that made this work possible.

I would also like to thank the rest of my committee members, Prof. Charles Geyer, Prof. Adam Rothman and Prof. Wei Pan. I'm grateful to have them as my committee members and I deeply appreciate their guidance and words of encouragement along the way.

I'm also grateful to my friends at the University of Minnesota for making the experience here enjoyable. Thanks go to Yunzhang Zhu, Yiping Yuan, Qi Yan, Zhihua Su, Feng Yi, Ben Sherwood, Yiwen Sun, Yuwen Gu, Yanjia Yu, Bo Peng, Dootika Vats, Subhabrata Majumdar, Xuetong Sun and many other friends in the Stat department, as well as my friends in other departments.

My special thanks go to my mother and my sister, who loved me and supported me all the time.

Finally, thank Jun for being by my side. I'm grateful to have you joining my life and thank you for all your love and support.

Dedication

To the memory of my father Cuizhen Yang and
To my mother Xinrong Li.

Abstract

Recommender systems aim to predict users' ratings on items and suggest certain items to users that they are most likely to be interested in. Recent years there has been a lot of interest in developing recommender systems, especially personalized recommender systems to efficiently provide personalized services and increase conversion rates in commerce. Personalized recommender systems identify every individual's preferences through analyzing users' behavior, and sometimes also analyzing user and item feature information.

Existing recommender system methods typically ignore the correlations between ratings given by a user. However, based on our observation the correlations can be strong. We propose a new personalized recommender system method that takes into account the correlation structure of ratings by a user. General precision matrices are estimated for the ratings of each user and clustered among users by supervised clustering. Moreover, in the proposed model we utilize user and item feature information, such as the demographic information of users and genres of movies. Individual preferences are estimated and grouped over users and items to find similar individuals that are close in nature. Computationally, we designed an algorithm applying the difference of convex method and the alternating direction method of multipliers to deal with the nonconvexity of the loss function and the fusion type penalty respectively. Theoretical rate of convergence is investigated for our new method. We also show theoretically that incorporating the correlation structure gives higher asymptotic efficiency of the estimators compared to ignoring it. Both simulation studies and Movielens data indicate that our method outperforms existing competitive recommender system methods.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Recommender Systems are used to predict users' response to options/items. With the development of the internet, recommender systems are becoming more and more important. They are applied very widely to e-commerce, including recommending restaurants, hotels, news, mobile phone games, movies and so on. For example, Netflix recommends movies based on history ratings and movie rental information; Expedia recommends hotels to book based on history information. Online retailers like Amazon.com and ebay.com which sell a vast variety of goods and services also take advantage of recommender systems to sell their products. Amazon and eBay recommend items by suggesting new lists like "More Items to Consider" and "Customers Who Bought This Item Also Bought" etc.

Generally, the problem of recommender systems is for a given user, to recommend some items this user is likely to be interested in. There are many specific formats of recommender systems designed for data collected from different recommendation scenarios.

Some recommendation applications inquire the user's conditions or criteria before they give recommendations thus they require the user to interact with the system in order to provide a recommendation. For example, on yelp.com users can specify the city they want to search, the price they would like to pay (divided to 4 price levels), the neighborhoods, distance, features (breakfast, brunch etc.), style of meals (American, Chinese etc.), and then get recommendations that meet their needs. This kind of recommender system which depends on knowledge about the user's needs and also

about the products is known as knowledge-based recommender systems [8].

Other recommender systems do not ask the user to input their needs and requirements for the next recommendation. They only use what is already in the system such as history ratings of users on other items. The systems may also collect user demographic information such as age and gender, and item feature information. These recommender systems typically aim at predicting the rating of an item a user has not purchased or seen before, using the information available. After the predictions are generated, items with the highest predicted ratings are recommended to users. There are also recommender systems that target at predicting ranking of unrated items, such as the rankboost algorithm proposed in [21]. They care about the relative orders of products and recommend items with the lowest ranking. We focus on these types of recommender systems which don't ask user needs for next item because they require less involvement of users and is applicable to most practical recommendation problems. This is also the most widely known and most commonly used formulation of recommender systems.

Recommender systems that predict ratings are the most commonly used format of all recommender systems, and we are mainly talking about this kind of recommenders in this introduction. The framework can be stated as follows. Suppose we have n users and m items. Let r_{ij} be the rating of user i on item j. Then all the ratings can be written in a matrix $\boldsymbol{R} = (r_{ij})$, with some question marks to represent unknown ratings.

$$\boldsymbol{R} = \begin{pmatrix} r_{11} & r_{12} & ? & \dots & r_{1m} \\ ? & r_{22} & r_{23} & \dots & ? \\ r_{31} & ? & ? & \dots & r_{3m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ ? & ? & r_{n3} & \dots & ? \end{pmatrix}, \tag{1.1}$$

where each row is the ratings of one user, and each column is the ratings on one item. In recommender systems, only part of the matrix $\boldsymbol{R}$ is observed. The entry r_{ij} is observed if user i has rated item j, and not observed otherwise. Usually there are a huge number of items, and users only rated a few of them. So a very large proportion of $\boldsymbol{R}$ is missing. We want to predict the missing ratings accurately.

Methods for building recommender systems have been discussed a lot in computer

science literature since collaborative filtering recommender systems appeared in the mid-1990s [1]. Afterwards a lot of developments were made in both industry and academia. Currently, there are two prevalent classes of approaches, i.e. collaborative filtering and content-based recommender systems. Collaborative filtering makes use of the information from similar users to predict the future action. Popular methods include matrix factorization approaches such as SVD decomposition in [22, 30] and many variants, matrix completion approaches such as [34].

Content-based recommender systems (e.g. [12, 7]) compare the content of an item with a user's profile and are mostly used in recommending textual materials. Techniques such as TF-IDF [45] in information retrieval are utilized for item feature extraction. One advantage of content-based methods is that ratings on new items can be predicted which solves the "cold start" problem partially. There are also many hybrid recommender systems developed combining collaborative filtering and content-based methods, for example [44, 3, 64]. Context-aware recommender systems which take into account the context under which a user rates an item have also been introduced, such as in [28] and [5]. Hybrid and context-aware methods have become the trend in recommender systems.

The existing methods typically assume the ratings of a given user on different items are independent, and ignore the missing mechanism of $\boldsymbol{R}$ which is usually not missing completely at random. The method in [6] takes one step further: they proposed a group-specific singular value decomposition method, by clustering users or items according to a certain missing mechanism they observed. Hence their method captures the individuals' latent characteristics that are not used in other approaches, and provides more accurate prediction than the previously mentioned methods. These will be explained with more details in Chapter 2.

We propose a correlation-incorporated method, which takes one step further than the typical methods, along a different direction from [6]. We notice that a user's rating on different items could be highly correlated: in the MovieLen data, this is extraordinarily apparent for different episodes of a movie series (Star Wars, for example); and this is also obvious for different movies adopted from similar true stories or related literary works. In nowadays cyber-context, there is some phenomenon called Intellectual Property (IP). Many movies can root from the same IP, and if this IP is particularly

preferred or disliked by a certain user, it is expected that such a user will rate these different movies in a highly correlated way. This IP phenomenon is one evidence for us to consider the correlation of ratings over different items, for further discussion see section 3.4. Note that these correlations cannot be captured by the explicit feature or latent characteristics. Inspired by this observation, we take the precision matrix into consideration in our method.

Another motivation for considering the precision matrix is that we have a grouping of users, according to their correlations. Our method of estimating the precision matrix generalizes the method of [63], which automatically gives the grouping by fused type penalty. Furthermore, we combine this grouping by taking into account user preference on item features and item "preference" on user features. Also the grouping is automatically given through our algorithm. Note that estimation the of precision matrix together with the preference vectors requires a large amount of computation, and we can reduce the effort by the above-mentioned grouping. Moreover, the incorporation of the correlation structure is proved to deliver smaller asymptotic variance and prediction error theoretically, thus increases the accuracy of the method.

The structure of the rest of this thesis is as follows: Chapter 2 is literature review about state of the art recommender systems; Chapter 3 discusses about the statistical model we propose, algorithm and theoretical results; Chapter 4 shows our numerical results in simulations and Movielens dataset; Chapter 5 gives a conclusion and discussion of our method.

Chapter 2

Four Kinds of Recommender Systems

There are some very challenging problems to solve in recommender systems. In most real applications, the number of users and items are both huge. For example, Amazon.com currently has about 300 million active customers and sells over 400 million products. With such a large dataset, fast computation inevitably becomes an issue. Scalability of the algorithm is necessary in order for it to be used in recommender systems to solve practical problems. Another challenge is that, although there are a huge number of users and items, the number of items rated by a user takes up a very small proportion of all items, usually below 1%. This makes it hard to predict unknown ratings precisely. It can be easily imagined this is the case for Internet companies like Amazon.com or Netflix as they have so many items. In research, the movie recommendation problem is studied quite often because of the availability of datasets in the public domain. This practical problem also has the issue of extremely low percentage of observed ratings. The popular Movielens dataset is provided by GroupLens, a research lab which studies recommender systems and some other related areas at the University of Minnesota. It contains three movie rating data of 100k, 1M and 10M, and the average proportions of rated movies among all movies in these three datasets are 6.3%, 4.2% and 1.3% respectively.

According to what information is used for predicting ratings, recommender systems

can be classified into several kinds (here we again ignore knowledge-based recommender systems which don't predict ratings). They are listed below and the basic idea is stated here.

- Collaborative Filtering recommender systems

  This kind of recommender systems utilizes user history rating information. Similar users are found by similar ratings, and their ratings are aggregated for prediction. This allows preference of other users to be borrowed when predicting a user's rating on an item not consumed by him/her before.

- Content-Based recommender systems

  This kind of recommender systems utilizes item contents or features. Similar items are found by similar contents, and ratings on them are aggregated for prediction.

- Hybrid recommender systems

  This kind of recommender systems seek ways to combine collaborative filtering and content-based recommender systems. Thus they utilize both user history ratings and item content information.

- Context-Aware recommender systems

  As suggested by its name, this kind of recommender systems is aware of the context where the recommendation is made, for example what time the user consumes the item, or whether there is a companion. So context is an extra dimension considered besides user history ratings and item contents by context-aware recommender systems.

The following sections in this chapter are going to explain each of them in detail.

2.1 Collaborative Filtering

Collaborative filtering recommender systems are the earliest developed recommender systems. They appeared in the mid-1990s. The earliest works are [43, 24, 49]. Collaborative filtering utilizes the partially filled rating matrix R in (1.1). To predict for a user, rather than only using ratings of this user, collaborative filtering believes there is some

latent connection between all users and items and thus pools available information from all users on all items together in some way. Traditional collaborative filtering methods directly find similar users based on past ratings on common items. Recent collaborative filtering methods don't define an explicit similarity measure but implicitly infers user relations. This way it's more flexible and doesn't only depend on one metric between users. Below they are discussed in more detail. An extensive review can be found in [10].

2.1.1 Traditional Collaborative Filtering

Traditional collaborative filtering recommender systems are based on the idea that people who share the same preferences in the past should also have the same preferences in the future. Given history ratings, similar users can be found as the ones who gave the closest ratings on the commonly rated items. Then the ratings of a user can be predicted based on the ratings of his/her similar users using a weighted or unweighted average. To be specific, following the notations in the Introduction chapter, suppose user i rated m_i items. Denote the indices of items user i rated by $I_i \triangleq \{i_1, i_2, \cdots, i_{m_i}\} \subseteq \{1, 2, \cdots, m\}$. For every other users, we calculate the similarity with user i based on their ratings on items they both rated. Let $I_{ij} = \{k | k \in I_i \; and \; k \in I_j\}$. The two most commonly used similarity measures are the cosine and correlation measure.

$$\begin{aligned} \text{sim}_1(i,j) &= \frac{\sum\limits_{k \in I_{ij}} r_{ik} r_{jk}}{\sqrt{\sum\limits_{k \in I_{ij}} r_{ik}^2} \sqrt{\sum\limits_{k \in I_{ij}} r_{jk}^2}} \quad \text{(cosine)} \\ \text{sim}_2(i,j) &= \frac{\sum\limits_{k \in I_{ij}} (r_{ik} - \overline{r}_i)(r_{jk} - \overline{r}_j)}{\sqrt{\sum\limits_{k \in I_{ij}} (r_{ik} - \overline{r}_i)^2} \sqrt{\sum\limits_{k \in I_{ij}} (r_{jk} - \overline{r}_j)^2}} \quad \text{(correlation)} \end{aligned} \tag{2.1}$$

In above, $\overline{r}_i = \frac{\sum\limits_{k \in I_{ij}} r_{ik}}{|I_{ij}|}$ and $\overline{r}_j = \frac{\sum\limits_{k \in I_{ij}} r_{jk}}{|I_{ij}|}$.

Besides cosine and correlation similarities, many other measures can be used. For example as described in [49], inverse of distance measures such as mean squared difference. There are also variants of the cosine and correlation similarities such as the

constrained Pearson correlation in [49] which takes into account the positivity and negativity of ratings. Its idea is as follows. Many of the rating system adopt possible ratings as consecutive integers. For instance, if the ratings are in $1, 2, \cdots, s$, then $(s+1)/2$ is the middle rating. Ratings greater than or equal to $(s+1)/2$ are positive, and smaller than $(s+1)/2$ are negative. If only ratings that are both positive or negative are allowed to increase the similarity, then a similarity measure can be

$$\mathrm{sim}_3(i,j) = \frac{\sum\limits_{k\in I_{ij}} \left(r_{ik} - \frac{s+1}{2}\right)\left(r_{jk} - \frac{s+1}{2}\right)}{\sqrt{\sum\limits_{k\in I_{ij}} \left(r_{ik} - \frac{s+1}{2}\right)^2}\sqrt{\sum\limits_{k\in I_{ij}} \left(r_{jk} - \frac{s+1}{2}\right)^2}}. \tag{2.2}$$

Given similarities between user i with all the other users, the K nearest neighbors can be used to predict ratings on items not rated by user i yet, where K is a pre-specified integer. when $K = n-1$, it's all users. Of course, when predicting user i's rating on item l, only users who rated this item will be used. After the users are fixed, either a weighted or unweighted rating of these users can be calculated as the predicted rating for user i. If weights are used, they are usually based on the similarities. To present it, let U be the set of users used for predicting rating on item l of user i which is r_{il}. Some examples of averaging ratings are

$$\begin{aligned} &1. \quad \hat{r}_{il} = \frac{\sum\limits_{j\in U} r_{jl}}{|U|} \quad \text{(Unweighted average)} \\ &2. \quad \hat{r}_{il} = \frac{\sum\limits_{j\in U} \mathrm{sim}(i,j)\cdot r_{jl}}{\sum\limits_{j\in U} \mathrm{sim}(i,j)} \quad \text{(Weighted average)} \end{aligned} \tag{2.3}$$

There are also many other ways to average ratings. For example, to account for the fact that different users may have different mean ratings, the ratings can be aggregated

as

$$
\begin{aligned}
&3. \quad \hat{r}_{il} = \overline{r}_i + \frac{\sum\limits_{j \in U} (r_{jl} - \overline{r}_j)}{|U|} \quad \text{(Unweighted average)} \\
&4. \quad \hat{r}_{il} = \overline{r}_i + \frac{\sum\limits_{j \in U} \text{sim}(i,j) \cdot (r_{jl} - \overline{r}_j)}{\sum\limits_{j \in U} \text{sim}(i,j)} \quad \text{(Weighted average)}
\end{aligned}
\tag{2.4}
$$

The above methods to do collaborative filtering are the "traditional" methods and the algorithms are quite intuition-based. They are easy to implement but if users don't share many rated items, then similarities are not accurate and thus affect the accuracy of prediction.

2.1.2 Recent Collaborative Filtering

Many advanced methods have been developed for collaborative filtering. [35] described a user-based and an item-based naive Bayes classifier. The user-based classifier treats the rating of one user as the response, and ratings of other users as features. It assumes given the rating of one user, ratings of all other users are independent. A posterior probability of this user's rating given all other user ratings can be calculated and used for prediction. The item-based classifier is a similar idea.

Another popular approach is via matrix decomposition/completion. The Singular Value Decomposition (SVD) is applied in recommender systems to reduce the dimension of user and item feature space as well as approximating the history ratings [46, 30, 32]. Specifically, the rating matrix $\boldsymbol{R}_{n \times m}$ is decomposed into the product of two low-rank matrices $\boldsymbol{A}_{n \times k}$ and $\boldsymbol{B}_{m \times k}$ while minimizing $\|\boldsymbol{R} - \boldsymbol{A}\boldsymbol{B}^T\|_2^2$. The column dimension k for $\boldsymbol{A}$ and $\boldsymbol{B}$ satisfy $k \ll \min\{m, n\}$.

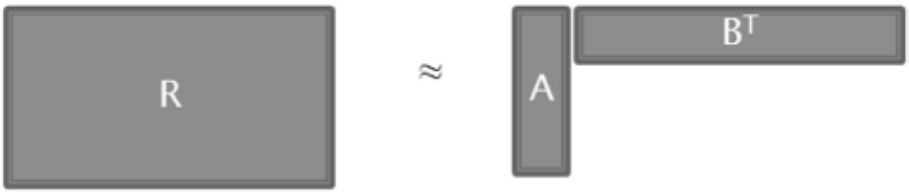

Figure 2.1: SVD in recommender systems[1]

$\boldsymbol{A}$ and $\boldsymbol{B}$ can be understood as the latent user and item factors that influences final ratings.

If $\boldsymbol{R}$ is a complete matrix, then the solution of $\boldsymbol{A}$ and $\boldsymbol{B}$ will be the SVD of $\boldsymbol{R}$ by taking the first k singular values. But since $\boldsymbol{R}$ is not complete, the minimization is done on the observed entries:

$$(\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}) = \underset{\boldsymbol{A},\boldsymbol{B}}{\operatorname{argmin}} \sum_{(i,j)\in O} (r_{ij} - \boldsymbol{a}_i^T \boldsymbol{b}_j)^2, \tag{2.5}$$

where O is the set of observed ratings, and $\boldsymbol{a}_i$ and $\boldsymbol{b}_j$ are the ith and jth row of $\boldsymbol{A}$ and $\boldsymbol{B}$ respectively. A regularized form of (2.5) is

$$(\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}) = \underset{\boldsymbol{A},\boldsymbol{B}}{\operatorname{argmin}} \sum_{(i,j)\in O} (r_{ij} - \boldsymbol{a}_i^T \boldsymbol{b}_j)^2 + \lambda(\sum_{i=1}^{n} \|\boldsymbol{a}_i\|_2^2 + \sum_{j=1}^{m} \|\boldsymbol{b}_j\|_2^2), \tag{2.6}$$

where $\lambda > 0$ is a regularization constant. The above shrinks $\boldsymbol{A}$ and $\boldsymbol{B}$ towards 0 to avoid overfitting. Alternating least squares algorithm can be used to solve (2.6), in which $\boldsymbol{A}$ and $\boldsymbol{B}$ are minimized alternately.

The method in [34] proposed to solve the following matrix completion problem for recommender systems:

$$\hat{\boldsymbol{Z}} = \underset{\boldsymbol{Z}}{\operatorname{argmin}} \sum_{(i,j)\in O} (r_{ij} - z_{ij})^2 + \lambda \|\boldsymbol{Z}\|_*. \tag{2.7}$$

Here $\boldsymbol{Z}$ is a $n \times m$ matrix. $\|\boldsymbol{Z}\|_*$ is the nuclear norm (also known as trace norm) which

[1]Stanford CS 294-34 slides

is defined as

$$\|\boldsymbol{Z}\|_* = \sum_{i=1}^{n} \sigma_i, \tag{2.8}$$

where the σ_i's are the singular values of $\boldsymbol{Z}$. The nuclear norm is used as a convex relaxation of rank. [34] also gave an efficient algorithm based on SVD to solve the above optimization.

Other people have proposed regularizing different matrix norms for the matrix completion. For example, the local max norm used in [20].

Rendle 2010 [41] proposed a new model class of factorization machine which can be applied to predict ratings. FMs model all the main terms and interactions of the input features. Parameters for the interaction terms are estimated through factorization. This allows coefficients to share components, and works well in sparse data setting such as recommender systems. The model equation is

$$\hat{y}(\boldsymbol{x}) = w_0 + \sum_i w_i x_i + \sum_{i,j} < \boldsymbol{v}_i, \boldsymbol{v}_j > x_i x_j, \tag{2.9}$$

where $w_0, \boldsymbol{w}, \boldsymbol{V}$ are parameters to estimate, and $\boldsymbol{x}$ is the input covariates for the corresponding response variable y. For example when $y = r_{ij}$, the rating of user i on item j, then $\boldsymbol{x}$ can be a vector of indicator variables with the first part as $n-1$ indicators to index the user, and the second part as $m-1$ indicators to index the item. If available, the user and item feature vectors can also be incorporated in $\boldsymbol{x}$, which makes it a hybrid recommender system.

Collaborative filtering recommender systems commonly have the new user and new item problem. For a new user entering the system, since no history rating is available, the system cannot make predictions for this user. Also for a new item which nobody has rated before, the system cannot predict ratings on it.

2.2 Content-Based Recommender Systems

Content-based recommender systems utilize the item features. The basic idea is users will like items similar to what they liked in the past. It doesn't combine information

across users. Current content-based recommender systems are mostly used in applications recommending items that contain texts such as web pages, documents [1], where the feature of the item is abundant enough to reflect the user preference.

Given the features of two items, say $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ respectively for item i and j, the similarity of these two items can be calculated using measures such as the cosine and correlation similarities given in (2.1). For example, the cosine similarity is

$$\text{sim}(\text{item}_i, \text{item}_j) = \frac{\boldsymbol{p}_i \cdot \boldsymbol{p}_j}{\|\boldsymbol{p}_i\|_2 \cdot \|\boldsymbol{p}_j\|_2}. \tag{2.10}$$

Then the predicted rating on an item can be decided by using the ratings on its nearest neighbors, such as using the unweighted or weighted average as in (2.3), (2.4).

In content-based recommender systems, since user preference is represented by the items rated in the past, how to extract informative features is an important issue. Recent advancements in information retrieval provides effective ways to extract features from textual contents. Typically texts are represented by its keywords. The simplest way is to count the times a word appears in the text. A more advanced and well-known approach to measure the importance of a word in a text is the TF-IDF (term frequency/inverse document frequency) measure [7, 45]. For keyword t, suppose it appears in document d for $f_{t,d}$ times, then its TF (term frequency) can be defined in several formats including but not limited to

$$\begin{aligned} \text{TF}(t,d) &= f_{t,d}, \\ \text{TF}(t,d) &= \frac{f_{t,d}}{\max\limits_{s} f_{s,d}}, \\ \text{TF}(t,d) &= 1 + \log(f_{t,d}). \end{aligned} \tag{2.11}$$

The IDF factor accounts for the fact that a keyword is not important if it appears in many documents. Suppose keyword t appears in m_d documents among all m documents. To make it clear, denote the collection of all documents by D. Then the IDF (inverse document frequency) and some of its variants are

$$\begin{aligned}
\text{IDF}(t, D) &= \log \frac{m}{m_d}, \\
\text{IDF}(t, D) &= \log \left(1 + \frac{m}{m_d}\right), \\
\text{IDF}(t, D) &= \log \left(1 + \frac{\max\limits_d m_d}{m_d}\right).
\end{aligned} \tag{2.12}$$

And the TF-IDF weight is the product of TF and IDF:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \cdot \text{IDF}(t, D). \tag{2.13}$$

Let $w_{t,d}$ be the weight of keyword t in document d. After all keyword weights are derived, each document can be represented by the weights . Use $\boldsymbol{w}_d$ to represent the profile/feature vector of document d,

$$\boldsymbol{w}_d = (w_{1,d}, w_{2,d}, \cdots, w_{T,d})^T. \tag{2.14}$$

Some content-based recommender systems build a user interest profile for each user. The user profile is based on the keyword weight vectors $\boldsymbol{w}_d$'s of documents rated by the user. A typical way is to use a weighted average of $\boldsymbol{w}_d$'s weighted according to the ratings. Then prediction on new items is made by comparing the user profile and the keyword weight vector of the new item. Items with high keyword similarities to the user profile are predicted to have high ratings and items with low keyword similarities are predicted to have low ratings.

Besides using the nearest neighbor type of approach, other techniques are applied to content-based recommender systems. Naive Bayes can be applied assuming the features are independent of each other given the rating. Machine learning methods such as decision trees, random forest and neural network can also be applied.

As item feature is of essential importance to content-based recommender systems, one limitation of content-based recommender systems is that it may not perform well if the features cannot represent the item sufficiently. Even for texts where effective ways of extracting features are available from information retrieval field, there is still a concern whether keywords alone is sufficient to describe a text. That's because even if

two documents use almost the same words, the way the words are organized can still be different. This is related to the writing style or quality of the document. Currently there's no valid methods to capture this aspect of texts. For non-textual items such as videos and images, automatic feature extraction still remains a problem. For example in movie recommendation, features such as the director, time of release, genre and so on can be obtained, but the movie video itself cannot be parsed and thus gives no features at all. With such limited features, content-based recommender systems may not learn the preferences of users well to give accurate predictions.

Another drawback of content-based recommender systems is that content-based systems can only recommend items similar to the ones rated before. Items dissimilar have a low similarity score and the predicted rating will be low. So they are always ignored and never get recommended to the user. But in fact, users are possible to select different items which are not similar to the previous ones next time. Thus content-based recommender systems limit user interests to the old ones and give too low ratings to dissimilar items.

Content-based recommender systems also have the new user problem, as collaborative filtering systems. For a new user who hasn't rated any item yet, a content-based system cannot learn his/her preferences. So no prediction can be made for this user. But different from collaborative filtering, content-based systems do not have the new item problem. The system doesn't rely on ratings of other people to predict for one user. It only relies on the "content" of the item. So even though no one has rated this item, based on the similarity of this item with the rated ones, the rating on this item can still be predicted.

2.3 Hybrid Recommender Systems

In many applications, both the user history rating information and item feature information can be obtained. Thus only using history ratings or only using item features is not efficient. A hybrid recommender system combining collaborative filtering which uses history ratings and content-based recommender systems which use item features can perform better. Furthermore, combining these two types of recommender systems can avoid the problems specific to one of them.

There are many ways to combine the two types of recommender systems. Hybrid recommender systems can be categorized into three classes [1, 9].

1. Use both collaborative filtering and content-based recommender system to predict the ratings. The ratings from these two systems are combined in some way.

2. Augment the feature space of one system from features in the other system.

3. Build a unified model that use both user history ratings and item features.

Details and some examples of three types of hybrid recommender systems are given below.

2.3.1 Combining Results and Augmenting Feature Space

Directly combining results is the most straightforward way of combining collaborative filtering and content-based recommender systems. To fulfill this, first both methods are implemented. To combine them, the simplest way is to do a linear combination. In real applications, the weights of two systems are often adjusted as more prediction are made and performance are seen. For example one possibility is to use equal weights at the beginning. As more predictions are made, the weight of the system that gives ratings closer to the observed gets larger, and the other weight gets smaller. Another approach used is switching between the two systems using some criteria. For example, the DailyLearner system tries content-based recommender system first. If it doesn't have high confidence in the predicted rating, then collaborative filtering is implemented and used.

An example of augmenting feature space can be to modify the similarity score calculation for two users in the traditional collaborative filtering system. To incorporate item feature information, for a user the history ratings can be augmented by the user profile built in a content-based recommender system. To be specific, suppose the user history ratings are in vector $\boldsymbol{r}_i$, and the user profile is $\boldsymbol{w}_i$. Then the augmented representation of this user is (all vectors are column vectors)

$$\boldsymbol{rw}_i = (\boldsymbol{r}_i^T, \boldsymbol{w}_i^T)^T. \tag{2.15}$$

And the similarity score between users can be calculated based on $\boldsymbol{rw}_i$'s. For user i and j, $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ are ratings on items they both rated. In cases where the number of commonly rated items is small for a pair of users, only using ratings to compute similarity may not give an accurate measure. The augmentation of item features can relieve this issue by adding more elements to the base of the comparison.

2.3.2 Building a Unified Model

Many recently developed hybrid recommender systems do not combine the result from the two methods or augment the features, like in section § 2.3.1. Instead, they utilize user history ratings and item features at the beginning step of the model building. Some examples of them are given below.

A unified probabilistic model was proposed in [47]. It builds a distribution over all user and movie pairs. The probability of user i purchase/rate item j is modeled using a latent class for users. Item content information is also incorporated to model the probability.

In [44] another probabilistic model was proposed. It employed the Restricted Boltzmann machines to combine collaborative and content information in a coherent manner. They only considered binary action on an item such as buying or not buying, watching a movie or not. Actions on all movies of user i, denoted as a_i is modeled to have joint probability

$$p(\boldsymbol{a}_i;\lambda) = \frac{1}{z(\lambda)}\exp\left(\sum_j \lambda_j a_{ij} + \sum_{j<k} \lambda_{jk} a_{ij} a_{ik}\right). \tag{2.16}$$

The unknown parameters are the λ_j's and λ_{jk}'s, corresponding to items and item pairs respectively. And z_λ is a normalization factor to make sure the probabilities sum to 1.

λ is modeled as

$$\begin{aligned} \lambda_j &= \boldsymbol{\mu}^T \boldsymbol{y}_j, \\ \lambda_{jk} &= \boldsymbol{y}_j^T H \boldsymbol{y}_k. \end{aligned} \tag{2.17}$$

Here $\boldsymbol{y}_j$ is the features for item j. $\boldsymbol{\mu}$ is an unknown parameter vector, and H is an unknown matrix assumed to be diagonal to reduce number of parameters to estimate. Actions of different users are assumed independent. Thus the log likelihood function

of all actions of all users is $\sum_i \log p(\boldsymbol{a}_i; \lambda)$. But as z_λ is difficult to write out explicitly, directly maximizing the log likelihood is hard. Instead, they used the psedo-likelihood

$$p(\boldsymbol{a}_i; \lambda) = \sum_j p(a_{ij} | \boldsymbol{a}_{i(-j)}; \lambda), \tag{2.18}$$

where $p(a_{ij}|\boldsymbol{a}_{i(-j)}; \lambda)$ is the conditional probability of action on item j given actions on all other items. Based on (2.16), this conditional probability has a logistic form and doesn't involve z_λ,

$$p(a_{ij}|\boldsymbol{a}_{i(-j)}; \lambda) = \frac{\exp(\lambda_j + \sum_{k \neq j} \lambda_{jk} a_k)}{1 + \exp(\lambda_j + \sum_{k \neq j} \lambda_{jk} a_k)}. \tag{2.19}$$

Then optimization is done on the psedo-likelihood to get estimates for λ.

Two content-boosted matrix factorization models were proposed in [36] based on the idea that if two items i and j have close features in the content-based system, then their latent factors in the matrix factorization $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$ (in (2.6)) should also be close. The first model is as follows: If the similarity of two item features $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ is greater than some threshold, then the objective function in (2.6) is augmented by a term encouraging the closeness of $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$. More formally,

$$(\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}) = \underset{\boldsymbol{A},\boldsymbol{B}}{\operatorname{argmin}} \sum_{(i,j) \in O} (r_{ij} - \boldsymbol{a}_i^T \boldsymbol{b}_j)^2 + \lambda (\sum_{i=1}^{n} \|\boldsymbol{a}_i\|_2^2 + \sum_{j=1}^{m} \|\boldsymbol{b}_j\|_2^2) - \lambda' \sum_{\boldsymbol{y}_i^T \boldsymbol{y}_j > c} \boldsymbol{b}_i^T \boldsymbol{b}_j. \tag{2.20}$$

Here $\lambda' > 0$ is another regularization constant.

The second model in [36] forces the similarity of the latent factors for two items to be in accordance with the similarity of their features. Suppose item features are of dimension T. Let $\boldsymbol{Y}_{m \times T} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m)^T$ be the item feature matrix. Instead of using $\boldsymbol{A} \cdot \boldsymbol{B}^T$ to approximate rating matrix $\boldsymbol{R}$, the item latent factor matrix $\boldsymbol{B}$ is factorized into $\boldsymbol{Y}_{m \times T} \cdot \boldsymbol{D}_{T \times k}$, the product of the item feature matrix and a coefficient matrix. The optimization problem becomes

$$(\hat{\boldsymbol{A}}, \hat{\boldsymbol{D}}) = \underset{\boldsymbol{A},\boldsymbol{D}}{\operatorname{argmin}} \sum_{(i,j) \in O} (r_{ij} - \boldsymbol{a}_i^T \boldsymbol{D}^T \boldsymbol{y}_j)^2 + \lambda \sum_{i=1}^{n} \|\boldsymbol{a}_i\|_2^2 + \lambda' \|\boldsymbol{D}\|^2. \tag{2.21}$$

A hierarchical Bayesian model was proposed in [4]. They use linear mixed effect model to predict ratings. Some demographic features of users such as age and gender are assumed available for the data which is true for some movie recommendation problems. The model is set up as follows:

$$\begin{aligned} r_{ij} &= \boldsymbol{z}_{ij}^T\boldsymbol{\mu} + \boldsymbol{x}_i^T\boldsymbol{\gamma}_j + \boldsymbol{y}_j^T\boldsymbol{\lambda}_i + e_{ij}, \\ \boldsymbol{\gamma}_j &\sim N(0, \boldsymbol{\Gamma}), \\ \boldsymbol{\lambda}_i &\sim N(0, \boldsymbol{\Lambda}), \\ e_{ij} &\sim N(0, \sigma^2), \end{aligned} \tag{2.22}$$

where $\boldsymbol{z}_{ij}$ is a vector containing features of user i and item j and their interactions, $\boldsymbol{x}_i$ is the feature vector of user i, $\boldsymbol{y}_j$ is the feature vector of item j. $\boldsymbol{\gamma}_j$ is the random effect of movie j, and $\boldsymbol{\lambda}_i$ is the random effect of user i. e_{ij} is a random noise. $\boldsymbol{\mu}, \boldsymbol{\Gamma}, \boldsymbol{\Lambda}, \sigma^2$ are unknown parameters and are estimated via Markov Chain Monte Carlo methods.

In fact, the model we propose has a similar setting of the mean part. But we do not use random effects, so the estimation doesn't need MCMC and is faster. And the random errors e_{ij} are assumed to be dependent on a user in our model. The details of our proposed models are given in the next chapter.

Also in a Bayesian framework, [3] proposed a regression-based latent factor model (RLFM). In their method, the latent factors are estimated through regressions on the explicit user and features. Assume a continuous rating, the model specifies that

$$\begin{aligned} r_{ij} &= \boldsymbol{z}_{ij}^T\boldsymbol{\mu} + \alpha_i + \beta_j + \boldsymbol{a}_i^T\boldsymbol{b}_j + e_{ij}, \\ \alpha_i &= \boldsymbol{g}_0'\boldsymbol{x}_i + \epsilon_i^\alpha, \qquad \epsilon_i^\alpha \sim N(0, c_\alpha), \\ \beta_j &= \boldsymbol{d}_0'\boldsymbol{y}_j + \epsilon_j^\beta, \qquad \epsilon_j^\beta \sim N(0, c_\beta), \\ \boldsymbol{a}_i &= \boldsymbol{G}\boldsymbol{x}_i + \boldsymbol{\epsilon}_i^u, \qquad \boldsymbol{\epsilon}_i^u \sim MVN(\boldsymbol{0}, C_u), \\ \boldsymbol{b}_j &= \boldsymbol{D}\boldsymbol{y}_j + \boldsymbol{\epsilon}_j^v, \qquad \boldsymbol{\epsilon}_j^v \sim MVN(\boldsymbol{0}, C_v). \end{aligned} \tag{2.23}$$

The parameters $\Theta = (\boldsymbol{\mu}, \boldsymbol{g}_0, \boldsymbol{d}_0, \boldsymbol{G}, \boldsymbol{D}, c_\alpha, c_\beta, C_u, C_v)$ are estimated through a scalable Monte Carlo EM algorithm.

In [64] Zhu et.al. proposed a likelihood based method seeking a sparsest latent feature factorization. They also incorporate explicit feature into preference prediction.

More detailedly, they assume r_{ij} has expectation θ_{ij}, and $\theta_{ij} = \boldsymbol{x}_i^T\boldsymbol{\alpha} + \boldsymbol{y}_j^T\boldsymbol{\beta} + \boldsymbol{a}_i^T\boldsymbol{b}_j$. In this method $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ are user and item explicit features; $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are regression coefficients; and $\boldsymbol{a}_i$ and $\boldsymbol{b}_j$ are user and item latent features. Then they minimize

$$\sum_{(i,j)\in\Omega} l(r_{ij}, \boldsymbol{x}_i^T\boldsymbol{\alpha} + \boldsymbol{y}_j^T\boldsymbol{\beta} + \boldsymbol{a}_i^T\boldsymbol{b}_j) + \lambda(\sum_i \sum_k J(|a_{ik}|) + \sum_j \sum_k J(|b_{jk}|)), \tag{2.24}$$

where l is the negative log likelihood, λ is a regularization constant, and J is a penalty function. The L_1 and L_0 (with TLP [52] as a computation proxy) penalties are applied. This method extends the SVD method by using likelihood to define the loss function, and also utilizes user and item feature information. A very recent work in [33] proposed a similar approach to [64]. In their method, they only considered using row covariates which corresponds to user features, but they impose that the latent features are orthogonal to the explicit features. They also allow the missing probability to be dependent on the observed features. The loss function is

$$f(\boldsymbol{\beta}, \boldsymbol{B}; \lambda_1, \lambda_2, \lambda_3, \alpha) = \frac{1}{nm}\|\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{B} - \boldsymbol{W}\circ\hat{\boldsymbol{\Theta}}^*\circ\boldsymbol{Y}\|_F^2 + \lambda_1\|\boldsymbol{\beta}\|_F^2 + \lambda_2(\alpha\|\boldsymbol{B}\|_* + (1-\alpha\|\boldsymbol{B}\|_F^2)), \tag{2.25}$$

where $\boldsymbol{X}$ is the user feature matrix, $\boldsymbol{\beta}$ is a coefficient matrix, , $\boldsymbol{B}_{n\times m}$ is a low rank matrix orthogonal to $\boldsymbol{X}$, $\circ$ is the Hadamard product,$\boldsymbol{W}$ is a binary missing indicator matrix, and $\hat{\boldsymbol{\Theta}}^*$ is the matrix of the inverse of the estimated missing probability.

In [6] a group-specific recommender system was proposed adding group-specific latent features for users and items. Since the ratings are usually not missing completely at random, they propose to use the missing pattern and features to group users and items. The model has r_{ij}'s expectation $\theta_{ij} = (\boldsymbol{a}_i + \boldsymbol{s}_{v_i})'(\boldsymbol{b}_j + \boldsymbol{t}_{u_j})$ where v_i is the group the ith user belongs to, u_j is the group the jth item belongs to, and $\boldsymbol{s}$ and $\boldsymbol{t}$ are the group specific effects. The loss function is defined as

$$\sum_{(i,j)\in\Omega} (r_{ij} - \theta_{ij})^2 + \lambda(\sum_i \|\boldsymbol{a}_i\|_2^2 + \sum_j \|\boldsymbol{b}_j\|_2^2 + \sum_v \|\boldsymbol{s}_v\|_2^2 + \sum_u \|\boldsymbol{t}_u\|_2^2). \tag{2.26}$$

With this natural grouping of users and items, their method shows a significant improvement over other existing and commonly used methods.

2.4 Context-Aware Recommender Systems

In real applications, sometimes the contextual variables may influence or even determine a user's choice of items. For example in movie recommendations, the people with whom users watch the movie may affect which movie they choose. If the movie is to be watched with kids, then Disney cartoon movies may be selected; if the movie is for fun with friends, other movies may have higher chance to be selected. And to recommend a restaurant, time in a day may be an important factor. If it's in the late morning, restaurants that serve brunch may get a high chance of being selected; if it's in the late afternoon, restaurants that serve dinner may be more likely to get selected. The specific meaning of context can vary for different recommendation problems.

Recently the importance of context for giving recommendations is realized. Recommender systems that also take context into consideration, besides the use of history ratings and item content information are developed to improve the accuracy of the prediction. A comprehensive overview of context-aware recommender systems was given in Adomavicius and Tuzhilin's book [2]. As mentioned in [2], for most existing recommender system which doesn't utilize context information, recommender systems attempt to fill in the rating matrix R in (1.1), and this represents the relation:

$$\text{User} \times \text{Item} \to \text{Rating}$$

But in context-aware recommender systems the space where the prediction is made is augmented, and the relation becomes:

$$\text{User} \times \text{Item} \times \text{Context} \to \text{Rating} \tag{2.27}$$

In context-aware recommender systems, the context information is represented by context variables. Typically they are categorized into some limited number of cases like user and item indices. *Time* is often used as a context variable in many applications. The number of context variables is not limited to 1. For example in movie recommendation, if variable *Companion* is used to describe the people with whom to watch a movie, then both *Time* and *Companion* can be used as the context variables. In a general recommender system, suppose there are C context variables "Context_1", "Context_2",$\cdots$,

"Context_C", then the space for prediction is of $C + 2$ dimensions, with the extra 2 dimensions for user and item. So a more detailed format of (2.27) is

$$\text{User} \times \text{Item} \times \text{Context}_1 \times \text{Context}_2 \times \cdots \times \text{Context}_\text{C} \to \text{Rating} \tag{2.28}$$

By [2] context-aware recommender systems are classified into three classes according to how the contextual information is used in the model building process, Contextual pre-filtering, Contextual post-filtering and Contextual modeling. They are explained in detail in the following part.

2.4.1 Contextual Pre-filtering and Post-filtering

The idea of contextual pre-filtering is group data into different context cases, and only use data in one specific context case to build a recommender system for this context. Following the notations for context variables, a context case is a combination of one possible value for each context variable, i.e, $(\text{Context}_1 = v_1, \text{Context}_2 = v_2, \cdots, \text{Context}_\text{C} = v_C)$ for some possible values $v_1, v_2, \cdots, v_C$ of the context variables.

A benefit of this approach is that for each context case, since the relation of the prediction goes back to $\text{User} \times \text{Item} \to \text{Rating}$, all previous techniques for non-contextual recommender systems can be used. But this approach also has a serious drawback. The number of context cases may be large as it's the combination of all context variables. There may be few observations under one specific case. In case the sizes of data for the context cases are small, the individual recommender systems built for context cases may not predict well due to lack of data. This method separates data and thus cannot incorporate information for other context cases.

Contextual post-filtering paradigm ignores the contextual information at first and builds a recommender system without considering it. After making the predictions, a list of the top N items with the highest predicted ratings is given. Then some adjustments of the predicted ratings are applied according to the context variables. Generally the adjustments are made by analyzing the preference of users under specific contexts. For example, if a user only goes to warm places like Florida in winter for vacation, then cold places for vacation can be filtered out on the recommendation list.

In [2], contextual post-filtering is divided into heuristic-based and model-based methods. Heuristic-based methods analyze the choices of items of a user under a context to find out common features of these items. Then based on these common features, either filtering or re-ranking of the recommendation list can be done. Filtering counts how many features a new item has that are common features of previously consumed items under this context. New items with this number of features smaller than some threshold value get filtered out. Reranking adjusts the ranking of the recommendations by taking the number of "good" features of an item into account. For example, use the product of the predicted rating from the non-contextual recommender systems and the number of "good" features as a new rating on the item. A new recommendation list of items is given according to the new rating. The higher the new rating, the higher the item appears on the list.

Model-based methods estimate the probability distribution of the item features under a context. Then the rating of an item is weighted by the probability of its features to generate a new rating. According to the new rating, the system can also choose to do filtering or re-ranking of the items.

Contextual post-filtering, like contextual pre-filtering, also has the benefit that all techniques for recommender systems without considering contexts can be applied. But how to effectively incorporate contextual information after predictions are made is still an open research area.

2.4.2 Contextual Modeling

Contextual modeling is the most active research area of contextual-aware recommender systems. Many researchers have been developing new methods for this area recently. Contextual modeling doesn't divide the modeling process into two stages like in contextual pre-filtering or contextual post-filtering. It builds contextual information into a unified model from the beginning of the modeling. Thus methods for traditional non-contextual recommender systems cannot be directly used in this setting.

One idea to do contextual modeling is to extend the heuristic-based traditional recommender systems that don't consider context. For example, the traditional similarity based collaborative filtering can be generalized as follows. To predict the rating of user i on item j under context t, denoted as r_{ijt}, we can use all the ratings $r_{i'jt'}$'s on item j

for a different user and context combination (i', t') where $i' \neq i$ or $t' \neq t$. The similarity of the combinations $(\text{user}_i, \text{context}_t)$ and $(\text{user}_{i'}, \text{context}_{t'})$ can be calculated based on ratings on other items except item j. This is an exact extension of the similarity based collaborative filtering. Many variants of it can be derived such as changing the similarity calculation to be based on features instead of ratings, that is to employ user features and context variables to compute similarity. Yet another way is to use both features and ratings.

Besides the heuristic-based methods from generalizing traditional recommender systems, there are some model-based methods proposed recently for contextual modeling. Some examples of them are given below.

Koren(2009) proposed *timeSVD++*, a model-based method considering time as the context variable in [29]. This work is an extension of the SVD matrix factorization method in collaborative filtering. It made some refinements of the basic SVD matrix factorization recommender system, and let the parameters vary with time. To be specific, first the proposed method extended the prediction model from

$$\hat{r}_{ui} = \boldsymbol{q}_i^T \boldsymbol{p}_u, \tag{2.29}$$

where $\hat{r}_{ui}$ is the predicted rating of user u's rating on item i, to

$$\hat{r}_{ui} = \mu + b_i + b_u + \boldsymbol{q}_i^T \left(\boldsymbol{p}_u + |R(u)|^{-1/2} \sum_{j \in R(u)} \boldsymbol{y}_j \right). \tag{2.30}$$

In above μ, b_i, b_u are additional parameters to capture the grand mean effect, user i effect and item j effect. $R(u)$ is the set of items rated by user u, and $\boldsymbol{y}_j$ is a item factor of the same dimension as $\boldsymbol{q}_i$'s and $\boldsymbol{p}_u$'s. The term $|R(u)|^{-1/2} \sum_{j \in R(u)} \boldsymbol{y}_j$ is added to reflect implicit information in the specific set of items rated. Furthermore, (2.30) is extended to allow parameters b_i, b_u and $\boldsymbol{p}_u$ to change over time and make the model dynamic as

$$\hat{r}_{ui} = \mu + b_i(t) + b_u(t) + \boldsymbol{q}_i^T \left(\boldsymbol{p}_u(t) + |R(u)|^{-1/2} \sum_{j \in R(u)} \boldsymbol{y}_j \right). \tag{2.31}$$

For item effect, it's assumed the change over time is slow. So they divided the time interval to smaller bins, and in each bin use a different item effect. More formally,

$$b_i(t) = b_i + b_{i,Bin(t)}. \tag{2.32}$$

But for effects relevant to a user, because the change of user preference is usually fast, the paper used more precise ways to describe it, such as splines of t. The model was experimented on Netflix dataset and did better than the nontemporal matrix factorization models.

Another contextual modeling method was proposed in [28]. They used N-dimensional tensor to include contextual information in recommender systems. Tensor is a generalization of matrix to more than two dimensions. The rating matrix in 2-D is extended to a rating tensor of N (N$>$ 2) dimensions with the extra (N-2) dimensions for context variables. The idea of building the recommender system is similar to the 2-D case. The HOSVD (High Order Singular Value Decomposition) was applied on the rating tensor with the optimization of parameters done using the observed ratings, and the sizes of the parameters regularized, as in the 2-D SVD decomposition collaborative filtering method. For the simplest case, if there is a single context variable, then the tensor is three dimensional. To show it, suppose the context variable takes l different values and following the notations before assume there are n users and m items. Use $\mathscr{R}$ to denote the rating tensor. Then $\mathscr{R}^{n\times m\times l}$ is decomposed into a core tensor $\mathscr{B}^{d_u\times d_m\times d_c}$ and matrices representing factors for users $\boldsymbol{U}_{n\times d_u}$, factors for items $\boldsymbol{M}_{m\times d_m}$, and factors for contexts $\boldsymbol{C}_{l\times d_c}$ as follows,

$$r_{ijk} = \mathscr{B} \times_1 \boldsymbol{U}_{i\cdot} \times_2 \boldsymbol{M}_{j\cdot} \times_3 \boldsymbol{C}_{k\cdot}, \tag{2.33}$$

where the $\times_1, \times_2, \times_3$ products are the mode-1, mode-2 and mode-3 products between a tensor and a matrix. For a general N-dimensional tensor $\mathscr{P}^{n_1\times n_2\times\cdots n_N}$ the mode-q product [13] with a matrix $\boldsymbol{A}_{d\times n_q}$ is another tensor of the same dimension

$$\mathscr{Q} = \mathscr{P} \times_q \boldsymbol{A} \in \mathbb{R}^{n_1\times n_2\times\cdots n_{q-1}\times d\times n_{q+1}\cdots n_N} \tag{2.34}$$

with

$$q_{i_1,i_2\cdots i_{q-1},j_q,i_{q+1},\cdots i_N} = \sum_{k=1}^{n_q} p_{i_1,i_2\cdots i_{q-1},k,i_q\cdots i_N} a_{j_q,k} \tag{2.35}$$

for $j_q = 1, 2, \cdots, d$.

With the decomposition in (2.33) they minimized the objective function

$$\sum_{i,j,k} l(r_{ijk}, y_{ijk}) + \lambda_U \|\boldsymbol{U}\|_2^2 + \lambda_M \|\boldsymbol{M}\|_2^2 + \lambda_C \|\boldsymbol{C}\|_2^2, \tag{2.36}$$

where y_{ijk} is the true rating and l is a loss function.

Rendle et.al. [42] used the factorization machine to realize a context-aware recommender system by increasing the feature space with context variables and adding pairwise interactions between user, item and context variables.

In [5] Xuan Bi et al. extended their method in the matrix scenario [6] to tensor to deal with context information. Identiability of the tensor method is proved which is not an issue in the matrix case. With the grouping defined by explicit features, for a new user, new item or new context, the group effect can be used for prediction. This is more accurate than using the grand mean. So this method helps solve the "cold start" problem. In our proposed method, we can also use explicit features to find closest users to a new user or closest items to a new item. Thus by K-nearest neighbor kind of approach, we can also predict for a new subject (user or item) and thus solve the "cold start" problem.

Chapter 3

Personalized Recommender System via Clustering

We propose a personalized recommender system model with correlation estimation. The dependencies of all the ratings made by a single user are taken into consideration and a separate precision matrix is estimated for each user. Similar user and items are identified via supervised clustering on individual preferences and user precision matrices. The idea of supervised clustering was previously discussed in [51, 38, 58] and we apply it in recommender systems for grouping users and items similar in nature. We propose to use a non-convex penalty for clustering. The ratings are built into a multivariate normal model incorporating both user and item feature information as predictors, where the random errors are allowed to have a general covariance matrix. We estimate the user "preference" and item "preference" for each user and each item. Users and items are clustered by adding regularization terms in the model objective function, thus different users and items are allowed to borrow information from each other.

In the grand map of the entire recommender system area, our method belongs to a unified hybrid recommender system as both user ratings and user and item features are taken into account in the model. Details of our model are explained in the following sections.

3.1 Model Specification

3.1.1 Models

Following the notations before, consider a situation in which we have a $n \times m$ rating matrix $\boldsymbol{R} = (r_{ij})_{n\times m}$. Each row and column of $\boldsymbol{R}$ correspond to one user and one item respectively, and some entries of $\boldsymbol{R}$ are missing. So r_{ij} is the rating of user i on movie j. Let z_{ij} be the binary indicator of missing, that is

$$z_{ij} = \begin{cases} 1 & \text{if } r_{ij} \text{ is observed} \\ 0 & \text{if } r_{ij} \text{ is missing} \end{cases}.$$

In this thesis we assume ignorable missing where the distribution of z_{ij}'s don't depend on the missing part of $\boldsymbol{R}$. For nonignorable missing, [31] can be referred for detailed discussion.

To account for correlations among item ratings associated with the same user, we assume that ratings from a user follow a multivariate normal distribution with some covariance matrix, and ratings from different users are independent.

To be specific, suppose user i rated m_i items with indices in set $I_i \triangleq \{i_1, i_2, \cdots, i_{m_i}\} \subseteq \{1, 2, \cdots, m\}$ as mentioned previouly in the Introduction, where $i_1 < i_2 < \cdots < i_{m_i}$. For observed ratings $\boldsymbol{r}_i = (r_{i,i_1}, r_{i,i_2}, \cdots, r_{i,i_{m_i}})^T$ from user i, we assume $\boldsymbol{r}_i \sim N(\boldsymbol{\mu}_i, \boldsymbol{\Omega}_i^{-1})$, where $\boldsymbol{\mu}_i = (\mu_{i,i_1}, \mu_{i,i_2}, \cdots, \mu_{i,i_{m_i}})^T$ is the mean of the observed ratings of user i, and $\boldsymbol{\Omega}_i$ is the precision matrix of observed ratings of user i to describe the correlations on ratings given by user i. Here the precision matrix is used instead of the covariance matrix to facilitate computation, because the log likelihood is convex in the precision matrix but not in the covariance matrix. More formally, our prediction model can be written as

$$r_{ij} = \mu_{ij} + \epsilon_{ij}, \quad \mu_{ij} = \boldsymbol{x}_i^T \boldsymbol{\alpha}_j + \boldsymbol{y}_j^T \boldsymbol{\beta}_i, \quad (\epsilon_{i,i_1}, \cdots, \epsilon_{i,i_{m_i}})^T \sim N(\boldsymbol{0}, \boldsymbol{\Omega}_i^{-1}) \tag{3.1}$$

for $i = 1, 2 \cdots, n$, $j = 1, 2, \cdots, m_i$, where $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ are user feature and item feature variables such as the demographic information of user i and the genre of movie j. To allow each user to have his/her own mean rating of items, the first element of an item feature vector $\boldsymbol{y}_j$ is always set to constant 1. Suppose user feature $\boldsymbol{x}_i's$ have dimension

K_1 and item feature $\boldsymbol{y}_j's$ have dimension K_2. Then $\boldsymbol{\alpha}_j$ is a K_1-dim vector representing "preference" of item j over user feature variables, and $\boldsymbol{\beta}_i$ is a K_2-dim vector representing "preference" of user i over item feature variables. So for the mean, items and users are treated equally. ϵ_{ij} is the random error. Let $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_m)$ be the $K_1 \times m$ item preference matrix, $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_n)^T$ be the $n \times K_2$ user preference matrix, and $\boldsymbol{\Omega} = (\boldsymbol{\Omega}_1, \cdots, \boldsymbol{\Omega}_n)$.

Without loss of generality, we assume the distribution of the missing indicator z_{ij}'s doesn't depend on the parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Omega}$ that are related to the distribution of $\{r_{ij}\}$. Then to do inference about $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Omega})$, we only need to look at the log likelihood of the observed part of $\boldsymbol{R}$. This can be written as

$$l(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Omega}) = \sum_{i=1}^{n} \left[\frac{1}{2} \text{logdet}(\boldsymbol{\Omega}_i) - \frac{(\boldsymbol{r}_i - \boldsymbol{\mu}_i)^T \boldsymbol{\Omega}_i (\boldsymbol{r}_i - \boldsymbol{\mu}_i)}{2} \right]. \tag{3.2}$$

To cluster the "preference" of different users and movies, we penalize the pairwise differences among $\boldsymbol{\alpha}_j$'s and $\boldsymbol{\beta}_i$'s. To group users, we also penalize the differences of the entries of different precision matrices corresponding to the same item or the same pair of items. A sparse structure on the precision matrix is also assumed to depict the conditional independence of ratings of one user on two items given the other ratings by the same user. For an item or an item pair, if at least one user rated it, we propagate to estimate the corresponding entry in the precision matrices for all users. Since at least one user rated each of the m items, we can estimate the diagonal elements for m items in the precision matrices for all users. To facilitate presentation, we put all estimated entries for each user i in a $m \times m$ precision matrix $\boldsymbol{\Omega}_{T_i}$. If an item pair (j, l) is rated by none of the n users, the entry $\omega_{T_i,jl}$ is fixed at 0 for all i. The submatrix of $\boldsymbol{\Omega}_{T_i}$ for items rated by user i is $\boldsymbol{\Omega}_i$.

Specifically, for item pair j and l $(j < l)$, suppose they are simultaneously rated by at least one user. Then we penalize the difference $|\omega_{T_i,jl} - \omega_{T_k,jl}|$ for all user pair i and k. The diagonal entry difference $|\omega_{T_i,jj} - \omega_{T_k,jj}|$ is also penalized for all item j and all user pair i and k. For the sparsity pursuit, we penalize $|\omega_{T_i,jl}|$ for $j \neq l$. Let $\boldsymbol{S}_i = (\boldsymbol{r_i} - \boldsymbol{\mu_i})(\boldsymbol{r_i} - \boldsymbol{\mu_i})^T$ and let J be a general penalty function, the penalized log likelihood is

$$
\begin{aligned}
l(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega}) = & \frac{1}{2}\sum_{i}\left[\text{logdet}(\boldsymbol{\Omega}_i) - \text{tr}(\boldsymbol{\Omega}_i\boldsymbol{S}_i)\right] - \frac{\lambda_1}{2}\sum_{i<k}\sum_{t} J(|\alpha_{it}-\alpha_{kt}|) \\
& - \frac{\lambda_1}{2}\sum_{i<k}\sum_{t} J(|\beta_{it}-\beta_{kt}|) - \lambda_1\sum_{i<k}\sum_{\substack{j\leqslant l \\ \exists h,\{j,l\}\subseteq I_h}} J(|\omega_{T_i,jl}-\omega_{T_k,jl}|) \\
& - \lambda_2\sum_{i}\sum_{j<l} J(|\omega_{T_i,jl}|)
\end{aligned}
\tag{3.3}
$$

where $\lambda_1, \lambda_2 > 0$ are regularization parameters. To maximize (3.3), equivalently we minimize

$$
\begin{aligned}
& - l(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega}) = \frac{1}{2}\sum_{i}\left[\text{tr}(\boldsymbol{\Omega}_i\boldsymbol{S}_i) - \text{logdet}(\boldsymbol{\Omega}_i)\right] + \frac{\lambda_1}{2}\sum_{i<k}\sum_{t} J(|\alpha_{it}-\alpha_{kt}|) \\
& + \frac{\lambda_1}{2}\sum_{i<k}\sum_{t} J(|\beta_{it}-\beta_{kt}|) + \lambda_2\sum_{i}\sum_{j<l} J(\omega_{T_i,jl}) + \lambda_1\sum_{i<k}\sum_{\substack{j\leqslant l \\ \exists h,\{j,l\}\subseteq I_h}} J(|\omega_{T_i,jl}-\omega_{T_k,jl}|)
\end{aligned}
\tag{3.4}
$$

with respect to $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\Omega}$.

For the penalty function J, we considered the L_1-norm and the L_0-norm. Since the L_0-norm is not continuous and hard to minimize, we use a computation surrogate for it, which is the truncated L_1 penalty [52] abbreviated as TLP. The objective function to minimize with the L_1 penalty is

$$
\begin{aligned}
& - l_1(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega}) = \frac{1}{2}\sum_{i}\left[\text{tr}(\boldsymbol{\Omega}_i\boldsymbol{S}_i) - \text{logdet}(\boldsymbol{\Omega}_i)\right] + \frac{\lambda_1}{2}\sum_{i<k}\|\boldsymbol{\alpha}_i-\boldsymbol{\alpha}_k\|_1 + \frac{\lambda_1}{2}\sum_{i<k}\|\boldsymbol{\beta}_i-\boldsymbol{\beta}_k\|_1 \\
& + \lambda_2\sum_{i}\sum_{j<l}|\omega_{T_i,jl}| + \lambda_1\sum_{i<k}\sum_{\substack{j\leqslant l \\ \exists h,\{j,l\}\subseteq I_h}} |\omega_{T_i,jl}-\omega_{T_k,jl}|.
\end{aligned}
\tag{3.5}
$$

(3.5) is convex in $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\Omega}$ separately as shown in Appendix A.1. But it's not convex in $(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega})$ together. In order to minimize it, we apply the difference of convex

algorithm. And inside each iteration of the difference of convex algorithm, the objective function is convex in $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Omega})$. So we minimize with respect to $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\Omega}$ alternately.

The loss function with the L_0 penalty is

$$
\begin{aligned}
-l_0(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Omega}) &= \frac{1}{2}\sum_i [\mathrm{tr}(\boldsymbol{\Omega}_i \boldsymbol{S}_i) - \mathrm{logdet}(\boldsymbol{\Omega}_i)] + \frac{\lambda_1}{2}\sum_{i<k} \|\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_k\|_0 + \frac{\lambda_1}{2}\sum_{i<k} \|\boldsymbol{\beta}_i - \boldsymbol{\beta}_k\|_0 \\
&+ \lambda_2 \sum_i \sum_{j<l} I(|\omega_{T_i,jl}| \neq 0) + \lambda_1 \sum_{i<k} \sum_{\substack{j \leqslant l \\ \exists h, \{j,l\} \subseteq I_h}} I(|\omega_{T_i,jl} - \omega_{T_k,jl}| \neq 0).
\end{aligned}
\tag{3.6}
$$

And the TLP function is defined as $J_\tau(x) = \min(|x|, \tau)$. Note that TLP is not convex and $J_\tau(x)/\tau$ approximates the L_0-penalty as $\tau > 0$ goes to 0_+. The objective function to minimize with the TLP penalty is

$$
\begin{aligned}
-l_{\mathrm{TLP}}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Omega}) &= \frac{1}{2}\sum_i [\mathrm{tr}(\boldsymbol{\Omega}_i \boldsymbol{S}_i) - \mathrm{logdet}(\boldsymbol{\Omega}_i)] + \frac{\lambda_1}{2}\sum_{i<k}\sum_t J_\tau(|\alpha_{it} - \alpha_{kt}|) \\
&+ \frac{\lambda_1}{2}\sum_{i<k}\sum_t J_\tau(|\beta_{it} - \beta_{kt}|) + \lambda_2 \sum_i \sum_{j<l} J_\tau(|\omega_{T_i,jl}|) + \lambda_1 \sum_{i<k} \sum_{\substack{j \leqslant l \\ \exists h, \{j,l\} \subseteq I_h}} J_\tau(|\omega_{T_i,jl} - \omega_{T_k,jl}|).
\end{aligned}
\tag{3.7}
$$

To minimize l_{TLP}, we also first apply the difference of convex algorithm, and inside each iteration update $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\Omega}$ alternately. We talk about the details of solving this problem in the next section.

3.1.2 A Special Case when $\Omega_i = \sigma^2 I$

If we ignore the correlations of ratings between different movies by the same user, we get a special case of (3.1).

$$
r_{ij} = \mu_{ij} + \epsilon_{ij}, \quad \mu_{ij} = \boldsymbol{x}_i^T \boldsymbol{\alpha}_j + \boldsymbol{y}_j^T \boldsymbol{\beta}_i, \quad \epsilon_{ij} \sim N(0, \sigma^2), \tag{3.8}
$$

where σ^2 is the error variance. Or equivalently, in this case $\boldsymbol{\Omega}_i = \sigma^2 I$ for $i = 1, \cdots n$. That is, ratings on all items from the same user are indepedent with the same variance.

In this case the estimate of σ^2 doesn't influence the estimates of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, thus can be omitted. To estimate $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, we minimize the objective function

$$\begin{aligned} -l(\boldsymbol{\alpha},\boldsymbol{\beta}) &= \frac{1}{2}\sum_i\sum_j \left(r_{ij}-\boldsymbol{x}_i^T\boldsymbol{\alpha}_j-\boldsymbol{y}_j^T\boldsymbol{\beta}_i\right)^2+\frac{\lambda_1}{2}\sum_{i<k}\sum_t J(|\alpha_{it}-\alpha_{kt}|) \\ &+\frac{\lambda_1}{2}\sum_{i<k}\sum_t J(|\beta_{it}-\beta_{kt}|). \end{aligned} \tag{3.9}$$

Since the covariance structure among ratings given by the same user is ignored, the model may fail to employ some useful information. In following sections, performance of this special case was compared to the general model.

3.2 Algorithm

To minimize (3.5) and (3.7) which are non-convex, we combine the difference of convex algorithm, alternating direction method of multipliers algorithm, blockwise coordinate descent algorithm, and accelerated alternating minimization algorithm[14] to solve convex relaxations of them. First the difference of convex algorithm is applied, then inside each iteration the mean and the precision matrices are updated alternately.

A DC decomposition of J_τ is $J_\tau(x)=|x|-\max(|x|-\tau,0)$. We use this decomposition for dealing with J_τ in (3.7).

3.2.1 Applying the difference of convex algorithm

For the L_1 method, we represent its loss function as the difference of (3.5) plus some quadratic terms for $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\Omega}$ and (3.5) itself. That is,

$$-l_1(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega}) = S_1^{l_1}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega})-S_2^{l_1}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega}), \tag{3.10}$$

where

$$
\begin{aligned}
S_1^{l_1}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega}) &= \frac{1}{2}\sum_i [\mathrm{tr}(\boldsymbol{\Omega}_i \boldsymbol{S}_i) - \mathrm{logdet}(\boldsymbol{\Omega}_i)] + \frac{\lambda_1}{2}\sum_{i<k}\|\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_k\|_1 + \frac{\lambda_1}{2}\sum_{i<k}\|\boldsymbol{\beta}_i - \boldsymbol{\beta}_k\|_1 \\
&\quad + \lambda_2 \sum_i \sum_{j<l} |\omega_{T_i,jl}| + \lambda_1 \sum_{i<k} \sum_{\substack{j\leqslant l \\ \exists h, \{j,l\}\subseteq I_h}} |\omega_{T_i,jl} - \omega_{T_k,jl}| \\
&\quad + c(\|\boldsymbol{\alpha}\|_F^2 + \|\boldsymbol{\beta}\|_F^2 + \sum_i \|\boldsymbol{\Omega}_i\|_F^2), \\
S_2^{l_1}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega}) &= c(\|\boldsymbol{\alpha}\|_F^2 + \|\boldsymbol{\beta}\|_F^2 + \sum_i \|\boldsymbol{\Omega}_i\|_F^2).
\end{aligned}
\tag{3.11}
$$

In above $c > 0$ is a constant. With a proper value of c, we can have $S_1^{l_1}$ to be a convex function of $(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega})$. Then at the (l)th iteration of the difference of convex algorithm, we replace $S_2^{l_1}$ with the linear approximation and minimize

$$
S_{l_1}^{(l)}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega}) = S_1^{l_1}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega}) - 2c(\langle \boldsymbol{\alpha}, \boldsymbol{\alpha}^{(l)}\rangle + \langle \boldsymbol{\beta}, \boldsymbol{\beta}^{(l)}\rangle + \sum_i \langle \boldsymbol{\Omega}_i, \boldsymbol{\Omega}_i^{(l)}\rangle). \tag{3.12}
$$

Here $\langle X, Y\rangle = \mathrm{tr}(XY^T)$ represents the Frobenius inner product of two matrices X and Y.

Similarly, for the TLP method, we represent the loss function as

$$
-l_{\mathrm{TLP}}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega}) = S_1^{\mathrm{TLP}}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega}) - S_2^{\mathrm{TLP}}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega}), \tag{3.13}
$$

with

$$
\begin{aligned}
&S_1^{\mathrm{TLP}}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega}) = S_1^{l_1}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega}),\\
&S_2^{\mathrm{TLP}}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega}) = \frac{\lambda_1}{2}\sum_{i<k}\sum_{c}\max(|\alpha_{ic}-\alpha_{kc}|-\tau,0) + \frac{\lambda_1}{2}\sum_{i<k}\sum_{c}\max(|\beta_{ic}-\beta_{kc}|-\tau,0)\\
&\qquad + \lambda_2\sum_{i}\sum_{j<l}\max(|\omega_{T_i,jl}|-\tau,0) + c(\|\boldsymbol{\alpha}\|_F^2 + \|\boldsymbol{\beta}\|_F^2 + \sum_i\|\boldsymbol{\Omega}_i\|_F^2)\\
&\qquad + \lambda_1\sum_{i<k}\sum_{\substack{j\leqslant l\\ \exists h,\{j,l\}\subseteq I_h}}\max(|\omega_{T_i,jl}-\omega_{T_k,jl}|-\tau,0).
\end{aligned}
\tag{3.14}
$$

Here c is the same constant as in (3.11). At the (l)th iteration of the difference of convex algorithm, we replace S_2^{TLP} with the linear approximation and minimize

$$
\begin{aligned}
S_{\mathrm{TLP}}^{(l)}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega}) = &\frac{1}{2}\sum_i[\mathrm{tr}(\boldsymbol{\Omega}_i\boldsymbol{S}_i) - \mathrm{logdet}(\boldsymbol{\Omega}_i)] + \frac{\lambda_1}{2}\sum_{(i,k,c)\in\mathscr{E}_{\boldsymbol{\alpha}}^{(l-1)}}|\alpha_{ic}-\alpha_{kc}|\\
&+\frac{\lambda_1}{2}\sum_{(i,k,c)\in\mathscr{E}_{\boldsymbol{\beta}}^{(l-1)}}|\beta_{ic}-\beta_{kc}| + \lambda_2\sum_{(i,j,l)\in\mathscr{E}_{\boldsymbol{\Omega}1}^{(l-1)}}|\omega_{T_i,jl}|\\
&+\lambda_1\sum_{(i,k,j,l)\in\mathscr{E}_{\boldsymbol{\Omega}2}^{(l-1)}}|\omega_{T_i,jl}-\omega_{T_k,jl}| + c(\|\boldsymbol{\alpha}\|_F^2+\|\boldsymbol{\beta}\|_F^2+\sum_i\|\boldsymbol{\Omega}_i\|_F^2)\\
&-2c(\langle\boldsymbol{\alpha},\boldsymbol{\alpha}^{(l)}\rangle + \langle\boldsymbol{\beta},\boldsymbol{\beta}^{(l)}\rangle + \sum_i\langle\boldsymbol{\Omega}_i,\boldsymbol{\Omega}_i^{(l)}\rangle).
\end{aligned}
\tag{3.15}
$$

where $\mathscr{E}_{\boldsymbol{\alpha}}^{(l-1)} = \{(i,k,c) : |\alpha_{ic}^{(l-1)} - \alpha_{kc}^{(l-1)}| \leqslant \tau\}$, $\mathscr{E}_{\boldsymbol{\beta}}^{(l-1)} = \{(i,k,c) : |\beta_{ic}^{(l-1)} - \beta_{kc}^{(l-1)}| \leqslant \tau\}$, $\mathscr{E}_{\boldsymbol{\Omega}1}^{(l-1)} = \{(i,j,l) : |\omega_{T_i,jl}^{(l-1)}| \leqslant \tau\}$, and $\mathscr{E}_{\boldsymbol{\Omega}2}^{(l-1)} = \{(i,k,j,l) : |\omega_{T_i,jl}^{(l-1)} - \omega_{T_k,jl}^{(l-1)}| \leqslant \tau\}$ are index sets determined by the values of the pameters in the previous difference of convex iteration.

Then objective functions inside the d.o.c. iterations (3.12) and (3.15) are convex in $(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega})$. Note that these two objective functions are very similar, and the only difference is we have the index sets for the TLP objective function. So next we only discuss about the method used to minimize (3.15) and skip the method for solving the L_1 problem. In order to deal with the non-differential and non-separable fused cost

in the TLP objective function, we apply Alternating Direction Method of Multipliers (ADMM).

To simplify notations below, below $i \sim_\beta k$ is used to represent $i \neq k$ and there is at least one c such that $|\beta_{ic}^{(l-1)} - \beta_{kc}^{(l-1)}| \leqslant \tau$; $i \sim_\alpha k$ is used to represent $i \neq k$ and there is at least one c such that $|\alpha_{ic}^{(l-1)} - \alpha_{kc}^{(l-1)}| \leqslant \tau$. To apply ADMM, we introduce constraints $\boldsymbol{\beta}_i - \boldsymbol{\beta}_k = \boldsymbol{\gamma}_{ik}$ for $i \sim_\beta k$ and $i < k$, $\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_k = \boldsymbol{\theta}_{ik}$ for $i \sim_\alpha k$ and $i < k$, $\boldsymbol{\Omega}_{T_i} = \boldsymbol{Z}_{T_i}$ for all i. That is, we solve the following equivalent problem:

$$
\begin{aligned}
\min & \frac{1}{2}\sum_i [\mathrm{tr}(\boldsymbol{\Omega}_i \boldsymbol{S}_i) - \mathrm{logdet}(\boldsymbol{\Omega}_i)] + \frac{\lambda_1}{2} \sum_{(i,k,c)\in\mathscr{E}_{\boldsymbol{\alpha}}^{(l-1)}} |\theta_{ikc}| + \frac{\lambda_1}{2} \sum_{(i,k,c)\in\mathscr{E}_{\boldsymbol{\beta}}^{(l-1)}} |\gamma_{ikc}| \\
& + \lambda_2 \sum_{(i,j,l)\in\mathscr{E}_{\boldsymbol{\Omega}1}^{(l-1)}} |z_{T_i,jl}| + \lambda_1 \sum_{(i,k,j,l)\in\mathscr{E}_{\boldsymbol{\Omega}2}^{(l-1)}} |z_{T_i,jl} - z_{T_k,jl}| \\
& + c(\|\boldsymbol{\alpha}\|_F^2 + \|\boldsymbol{\beta}\|_F^2 + \sum_i \|\boldsymbol{\Omega}_i\|_F^2) - 2c(\langle \boldsymbol{\alpha}, \boldsymbol{\alpha}^{(l)} \rangle + \langle \boldsymbol{\beta}, \boldsymbol{\beta}^{(l)} \rangle + \sum_i \langle \boldsymbol{\Omega}_i, \boldsymbol{\Omega}_i^{(l)} \rangle), \\
\text{with } & \boldsymbol{\beta}_i - \boldsymbol{\beta}_k = \boldsymbol{\gamma}_{ik} \text{ for } i \sim_\beta k \text{ and } i < k, \\
& \boldsymbol{\alpha}_i - \boldsymbol{\alpha}_k = \boldsymbol{\theta}_{ik} \text{ for } i \sim_\alpha k \text{ and } i < k, \\
& \boldsymbol{\Omega}_{T_i} = \boldsymbol{Z}_{T_i} \text{ for } i = 1, \cdots n.
\end{aligned}
\tag{3.16}
$$

With dual variables $\boldsymbol{u}_{ik}, \boldsymbol{\nu}_{ik}, \boldsymbol{U}_{T_i}$ and constant $\rho > 0$, the scaled augmented Lagrangian is

$$
\begin{aligned}
& \frac{1}{2}\sum_i [\mathrm{tr}(\boldsymbol{\Omega}_i \boldsymbol{S}_i) - \mathrm{logdet}(\boldsymbol{\Omega}_i)] + \frac{\lambda_1}{2} \sum_{(i,k,c)\in\mathscr{E}_{\boldsymbol{\alpha}}^{(l-1)}} |\theta_{ikc}| + \frac{\lambda_1}{2} \sum_{(i,k,c)\in\mathscr{E}_{\boldsymbol{\beta}}^{(l-1)}} |\gamma_{ikc}| \\
& + \lambda_2 \sum_{(i,j,l)\in\mathscr{E}_{\boldsymbol{\Omega}1}^{(l-1)}} |z_{T_i,jl}| + \lambda_1 \sum_{(i,k,j,l)\in\mathscr{E}_{\boldsymbol{\Omega}2}^{(l-1)}} |z_{T_i,jl} - z_{T_k,jl}| \\
& + c(\|\boldsymbol{\alpha}\|_F^2 + \|\boldsymbol{\beta}\|_F^2 + \sum_i \|\boldsymbol{\Omega}_i\|_F^2) - 2c(\langle \boldsymbol{\alpha}, \boldsymbol{\alpha}^{(l)} \rangle + \langle \boldsymbol{\beta}, \boldsymbol{\beta}^{(l)} \rangle + \sum_i \langle \boldsymbol{\Omega}_i, \boldsymbol{\Omega}_i^{(l)} \rangle) \\
& + \frac{\rho}{2} \sum_{i\sim_\beta k \& i<k} \|\boldsymbol{\beta}_i - \boldsymbol{\beta}_k - \boldsymbol{\gamma}_{ik} + \boldsymbol{u}_{ik}\|_2^2 + \frac{\rho}{2} \sum_{i\sim_\alpha k \& i<k} \|\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_k - \boldsymbol{\theta}_{ik} + \boldsymbol{v}_{ik}\|_2^2 \\
& + \frac{\rho}{2} \sum_i \|\boldsymbol{\Omega}_{T_i} - \boldsymbol{Z}_{T_i} + \boldsymbol{U}_{T_i}\|_F^2.
\end{aligned}
\tag{3.17}
$$

At each iteration of the ADMM algorithm, we minimize w.r.t. $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Omega}, \boldsymbol{\gamma}, \boldsymbol{\theta}, \boldsymbol{Z}$ and also update the dual variables $\boldsymbol{u}, \boldsymbol{\nu}, \boldsymbol{U}$. Repeat the iterations until convergence. Detailed algorithm for updating the mean parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\theta}$ and precision matrice parameters $\boldsymbol{Z}, \boldsymbol{U}$ are given in the following subsections.

At convergence of the ADMM algorithm we get $(\boldsymbol{\alpha}^{(l)}, \boldsymbol{\beta}^{(l)}, \boldsymbol{\Omega}^{(l)})$. The difference of convex algorithm is terminated when the decrease of the objective function (3.7) is smaller than some precision.

The outline of the algorithm to solve the TLP problem is summarized in Algorithm 1.

Algorithm 1 Algorithm outline to solve the TLP problem

1. Start from $(\boldsymbol{\alpha}^{(0)}, \boldsymbol{\beta}^{(0)}, \boldsymbol{\Omega}^{(0)})$.
2. To get $(\boldsymbol{\alpha}^{(l)}, \boldsymbol{\beta}^{(l)}, \boldsymbol{\Omega}^{(l)})$ from $(\boldsymbol{\alpha}^{(l-1)}, \boldsymbol{\beta}^{(l-1)}, \boldsymbol{\Omega}^{(l-1)})$,
   i. Start ADMM iterations to solve (3.15) with $(\boldsymbol{\alpha}^0, \boldsymbol{\beta}^0, \boldsymbol{\Omega}^0) = (\boldsymbol{\alpha}^{(l-1)}, \boldsymbol{\beta}^{(l-1)}, \boldsymbol{\Omega}^{(l-1)})$, $\boldsymbol{\gamma}^0 = \mathbf{0}, \boldsymbol{\theta}^0 = \mathbf{0}, \boldsymbol{u}^0 = \mathbf{0}, \boldsymbol{\nu}^0 = \mathbf{0}$, $\boldsymbol{Z}^0 = \boldsymbol{\Omega}^{(l-1)}$, $\boldsymbol{U}^0 = \mathbf{0}$.
   ii. Update from $(\boldsymbol{\alpha}^t, \boldsymbol{\beta}^t, \boldsymbol{\Omega}^t, \boldsymbol{\gamma}^t, \boldsymbol{\theta}^t, \boldsymbol{u}^t, \boldsymbol{\nu}^t, \boldsymbol{Z}^t, \boldsymbol{U}^t) \rightarrow (\boldsymbol{\alpha}^{t+1}, \boldsymbol{\beta}^{t+1}, \boldsymbol{\Omega}^{t+1}, \boldsymbol{\gamma}^{t+1}, \boldsymbol{\theta}^{t+1}, \boldsymbol{u}^{t+1}, \boldsymbol{\nu}^{t+1}, \boldsymbol{Z}^{t+1}, \boldsymbol{U}^{t+1})$ with the formulas in the subsections below.
   iii. Terminate the ADMM algorithm if the difference between variables in two iterations are below a precision.
3. Terminate if change in the objective function (3.7) is less than some precision ϵ. Otherwise repeat 2.

3.2.2 Mean updating

Suppose user i rated m_i items, let their indices be $I_i \triangleq \{i_1, i_2, \cdots, i_{m_i}\}$. Without loss of generality, assume that $i_1 < i_2 < \cdots < i_{m_i}$. Let $\boldsymbol{\alpha}_{I_i}$ denote the submatrix of $\boldsymbol{\alpha}$ consisting of columns indexed in I_i. Let $\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m)^T$ be the matrix of item features and $\boldsymbol{Y}_{I_i}$ be the submatrix of $\boldsymbol{Y}$ with rows indexd in I_i. To solve $\boldsymbol{\beta}$, note that

$$\boldsymbol{\mu}_i = \boldsymbol{\alpha}_{I_i}^T \boldsymbol{x}_i + \boldsymbol{Y}_{I_i} \boldsymbol{\beta}_i. \tag{3.18}$$

The part of the augmented Lagrangian (3.17) related to $\boldsymbol{\beta}$ is,

$$\frac{1}{2}\sum_i(\boldsymbol{r}_i-\boldsymbol{\mu}_i)^T\boldsymbol{\Omega}_i(\boldsymbol{r}_i-\boldsymbol{\mu}_i)+\frac{\lambda_1}{2}\sum_{(i,k,c)\in\mathscr{E}_{\boldsymbol{\beta}}^{(l-1)}}|\gamma_{ikc}|+\frac{\rho}{2}\sum_{i\sim_\beta k\&i<k}\|\boldsymbol{\beta}_i-\boldsymbol{\beta}_k-\boldsymbol{\gamma}_{ik}+\boldsymbol{u}_{ik}\|_2^2$$
$$+c\|\boldsymbol{\beta}\|_F^2-2c\langle\boldsymbol{\beta},\boldsymbol{\beta}^{(l)}\rangle. \tag{3.19}$$

Minimize the above w.r.t. variables $\boldsymbol{\beta},\boldsymbol{\gamma}$. This is a quadratic form of each $\boldsymbol{\beta}_i$. And $\sum_{i=1}^n(\boldsymbol{r}_i-\boldsymbol{\mu}_i)^T\boldsymbol{\Omega}_i(\boldsymbol{r}_i-\boldsymbol{\mu}_i)$ is separable in $\boldsymbol{\beta}_i's$. We update $\boldsymbol{\beta}$ by alternately updating $\boldsymbol{\beta}_i$. $\boldsymbol{\gamma}$ can be solved by soft-thresholding. Denote the soft-thresholding function by $ST(x,\alpha)=\text{sign}(x)(|x|-\alpha)_+$ for x real and $\alpha>0$. Let l_i be the number of $\boldsymbol{\beta}_k$'s satisfying $i\sim_\beta k$. Start from some $\boldsymbol{\beta}^0,\boldsymbol{\gamma}^0,\boldsymbol{u}^0$, update $\boldsymbol{\beta},\boldsymbol{\gamma}$ and $\boldsymbol{u}$ as follows. At step $t+1$,

- $\boldsymbol{\beta}_i^{t+1}=[\boldsymbol{Y}_{I_i}^T\boldsymbol{\Omega}_i\boldsymbol{Y}_{I_i}+\rho(l_i-1)I+2cI]^{-1}\big(\boldsymbol{Y}_{I_i}^T\boldsymbol{\Omega}_i(\boldsymbol{r}_i-\boldsymbol{\alpha}_{I_i}^T\boldsymbol{x}_i)+\rho\sum_{i\sim_\beta k\&i<k}\boldsymbol{\beta}_k^t+\rho\sum_{i\sim_\beta k\&i>k}\boldsymbol{\beta}_k^{t+1}+\rho\sum_{i<k\&i\sim_\beta k}(\boldsymbol{\gamma}_{ik}^t-\boldsymbol{u}_{ik}^t)-\rho\sum_{k<i\&i\sim_\beta k}(\boldsymbol{\gamma}_{ki}^t-\boldsymbol{u}_{ki}^t)+2c\boldsymbol{\beta}^{(l)}\big)$ for $i=1,2\cdots,n$.
- $\gamma_{ikc}^{t+1}=ST(\beta_{ic}^{t+1}-\beta_{ik}^{t+1}+u_{ikc}^t,\frac{\lambda_1}{2\rho})$ for $i<k$ and $(i,k,c)\in\mathscr{E}_{\boldsymbol{\beta}}^{(l-1)}$, $\gamma_{ikc}^{t+1}=\beta_{ic}^{t+1}-\beta_{ik}^{t+1}+u_{ikc}^t$ for $i<k$, $i\sim_\beta k$ and $(i,k,c)\notin\mathscr{E}_{\boldsymbol{\beta}}^{(l-1)}$.
- $\boldsymbol{u}_{ik}^{t+1}=\boldsymbol{u}_{ik}^t+\boldsymbol{\beta}_i^{t+1}-\boldsymbol{\beta}_k^{t+1}-\boldsymbol{\gamma}_{ik}^{t+1}$ for $i<k$ and $i\sim_\beta k$.

Note that in above the update for $\boldsymbol{\gamma}_{ik}$ can be done in parallel across paired indices (i,k), so as for $\boldsymbol{u}_{ik}$. For these parts and parts mentioned below that can be done parallelly, we used the openmp API for shared memory multiprocessing programming in C++ to do parallel computing.

$\boldsymbol{\alpha}$ can be updated likewise. At lth iteration of d.o.c., the part of the objective function related to $\boldsymbol{\alpha}$ is,

$$\frac{1}{2}\sum_i (\boldsymbol{r}_i - \boldsymbol{\mu}_i)^T \boldsymbol{\Omega}_i (\boldsymbol{r}_i - \boldsymbol{\mu}_i) + \frac{\lambda_1}{2} \sum_{(i,k,c)\in \mathscr{E}_{\boldsymbol{\alpha}}^{(l-1)}} |\theta|_{ikc} + \frac{\rho}{2} \sum_{i\sim_\alpha k \& i<k} \|\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_k - \boldsymbol{\theta}_{ik} + \boldsymbol{v}_{ik}\|_2^2$$
$$+ c\|\boldsymbol{\alpha}\|_F^2 - 2c\langle \boldsymbol{\alpha}, \boldsymbol{\alpha}^{(l)} \rangle. \tag{3.20}$$

Minimize it w.r.t. variables $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$. If a user i rated item j, and suppose it's the t_ith item rated by user i. Let $\boldsymbol{r}_{i(-j)}$ denote the ratings of user i excluding item j, and $\boldsymbol{\mu}_{i(-j)}$ accordingly. Let $\boldsymbol{\Omega}_{i,t_i\cdot(-t_i)}$ denote the t_ith row of $\boldsymbol{\Omega}_i$ without the t_ith element. The part of $\mathrm{tr}(\boldsymbol{\Omega}_i \boldsymbol{S}_i)$ that involves $\boldsymbol{\alpha}_j$ is

$$\omega_{i,t_i t_i}(r_{ij} - (\boldsymbol{x}_i^T \boldsymbol{\alpha}_j + \boldsymbol{y}_j^T \boldsymbol{\beta}_i))^2 + 2(r_{ij} - (\boldsymbol{x}_i^T \boldsymbol{\alpha}_j + \boldsymbol{y}_j^T \boldsymbol{\beta}_i))\boldsymbol{\Omega}_{i,t_i\cdot(-t_i)}(\boldsymbol{r}_{i(-j)} - \boldsymbol{\mu}_{i(-j)}). \tag{3.21}$$

The above is quadratic in $\boldsymbol{\alpha}_j$. $\boldsymbol{\theta}_{jk}$ can be solved by soft-thresholding. Let h_j be the number of $\boldsymbol{\alpha}_k$'s satisfying $j \sim_\alpha k$. Start from some $\boldsymbol{\alpha}^0, \boldsymbol{\theta}^0, \boldsymbol{v}^0$, update $\boldsymbol{\alpha}, \boldsymbol{\theta}$ and $\boldsymbol{v}$ as follows.

- $\boldsymbol{\alpha}_j^{t+1} = \Big[\sum_{j\in I_i} \omega_{i,t_i t_i} \boldsymbol{x}_i \boldsymbol{x}_i^T + \rho(h_j - 1)I + 2cI\Big]^{-1} \Big\{ \sum_{j\in I_i} [\boldsymbol{\Omega}_{i,t_i\cdot(-t_i)}(\boldsymbol{r}_{i(-j)} - \boldsymbol{\mu}_{i(-j)}) + \omega_{i,t_i t_i}(r_{ij} - \boldsymbol{y}_j^T \boldsymbol{\beta}_i)]\boldsymbol{x}_i + \rho \sum_{j\sim_\alpha k \& j<k} \boldsymbol{\alpha}_k^t + \rho \sum_{j\sim_\alpha k \& j>k} \boldsymbol{\alpha}_k^{t+1} + \rho \sum_{j\sim_\alpha k \& j<k} (\boldsymbol{\theta}_{jk}^t - \boldsymbol{v}_{jk}^t) - \rho \sum_{j\sim_\alpha k \& k<j} (\boldsymbol{\theta}_{kj}^t - \boldsymbol{v}_{kj}^t) + 2c\boldsymbol{\alpha}^{(l)} \Big\}$ for $j = 1, 2\cdots, m$.
- $\theta_{jkc}^{t+1} = ST(\alpha_{jc}^{t+1} - \alpha_{kc}^{t+1} + \nu_{jk}^t, \frac{\lambda_1}{2\rho})$ for $j < k$ and $(i,k,c) \in \mathscr{E}_{\boldsymbol{\alpha}}^{(l-1)}$; $\theta_{jkc}^{t+1} = \alpha_{jc}^{t+1} - \alpha_{kc}^{t+1}$ for $j < k$, $j \sim_\alpha k$ and $(i,k,c) \notin \mathscr{E}_{\boldsymbol{\alpha}}^{(l-1)}$.
- $\boldsymbol{v}_{jk}^{t+1} = \boldsymbol{v}_{jk}^t + \boldsymbol{\alpha}_j^{t+1} - \boldsymbol{\alpha}_k^{t+1} - \boldsymbol{\theta}_{jk}^{t+1}$ for $j < k$ and $j \sim_\alpha k$.

The update for $\boldsymbol{\theta}_{jk}$ and $\boldsymbol{v}_{jk}$ are done in parallel across index pairs (j,k).

3.2.3 Precision matrix updating

The part of the augmented Lagrangian (3.17) related to $\mathbf{\Omega}$ is

$$\begin{aligned} g_\rho(\mathbf{\Omega}, \boldsymbol{Z}, \boldsymbol{U}) = & \frac{1}{2}\sum_i [\mathrm{tr}(\mathbf{\Omega}_i \boldsymbol{S}_i) - \mathrm{logdet}(\mathbf{\Omega}_i)] + \frac{\rho}{2}\sum_i \|\mathbf{\Omega}_{T_i} - \boldsymbol{Z}_{T_i} + \boldsymbol{U}_{T_i}\|_F^2 \\ & + \lambda_2 \sum_{(i,j,l)\in \mathscr{E}_{\mathbf{\Omega}1}^{(l-1)}} |z_{T_i,jl}| + \lambda_1 \sum_{(i,k,j,l)\in \mathscr{E}_{\mathbf{\Omega}2}^{(l-1)}} |z_{T_i,jl} - z_{T_k,jl}| \\ & + c\sum_i \|\mathbf{\Omega}_i\|_F^2 - 2c\sum_i \langle \mathbf{\Omega}_i, \mathbf{\Omega}_i^{(l)} \rangle. \end{aligned} \tag{3.22}$$

At ADMM step $t+1$, first minimize $g_\rho(\mathbf{\Omega}, \boldsymbol{Z}^t, \boldsymbol{U}^t)$ with respect to $\mathbf{\Omega}$. Decompose the last term $\frac{\rho}{2}\sum_i \|\mathbf{\Omega}_{T_i} - \boldsymbol{Z}_{T_i} + \boldsymbol{U}_{T_i}\|_F^2$ to

$$\frac{\rho}{2}\sum_i \|\mathbf{\Omega}_i - \boldsymbol{Z}_i + \boldsymbol{U}_i\|_F^2 + \frac{\rho}{2}\sum_i \|\mathbf{\Omega}_{-i} - \boldsymbol{Z}_{-i} + \boldsymbol{U}_{-i}\|_F^2, \tag{3.23}$$

where $\mathbf{\Omega}_{-i}$ represent the part of $\mathbf{\Omega}_{T_i}$ after taking $\mathbf{\Omega}_i$ out, and $\boldsymbol{Z}_{-i}, \boldsymbol{U}_{-i}$ likewise. The $\|\cdot\|_F^2$ on this term is just sum of squared entries. So we can solve $\mathbf{\Omega}_i$ and $\mathbf{\Omega}_{-i}$ separately. The solution for $\mathbf{\Omega}_{-i}$ is $\boldsymbol{Z}_{-i} - \boldsymbol{U}_{-i}$. Take gradient with respect to $\mathbf{\Omega}_i$, we get

$$\frac{1}{2}\sum_i (\boldsymbol{S}_i - \mathbf{\Omega}_i^{-1}) + \rho(\mathbf{\Omega}_i - \boldsymbol{Z}_i^t + \boldsymbol{U}_i^t) = 0. \tag{3.24}$$

Let $\boldsymbol{T}_i^t \triangleq \frac{1}{2}\boldsymbol{S}_i - \rho(\boldsymbol{Z}_i^t - \boldsymbol{U}_i^t)$. Suppose $\boldsymbol{T}_i^t$ has eigen decomposition $\boldsymbol{V}\mathbf{\Lambda}\boldsymbol{V}^T$ with eigenvalues λ_{kk} for $k = 1 \cdots, m_i$, then the solution for $\mathbf{\Omega}_i^{t+1}$ is $\boldsymbol{V}\mathbf{\Lambda}'\boldsymbol{V}^T$ with $\lambda'_{kk} = \frac{-\lambda_{kk} + \sqrt{\lambda_{kk}^2 + 2\rho}}{2\rho}$ [16]. The positive-definiteness of $\mathbf{\Omega}_i^{t+1}$ is automatically satisfied. Notice this step can be done in parallel for all i.

To minimize $g_\rho(\mathbf{\Omega}^{t+1}, \boldsymbol{Z}, \boldsymbol{U}^t)$ w.r.t. $\boldsymbol{Z}$, it is to solve

$$\min_{\boldsymbol{Z}} \frac{1}{2}\sum_i \|\boldsymbol{Z}_{T_i} - \boldsymbol{A}_{T_i}\|_F^2 + \lambda_2 \sum_{(i,j,l)\in \mathscr{E}_{\mathbf{\Omega}1}^{(l-1)}} |z_{T_i,jl}| + \lambda_1 \sum_{(i,k,j,l)\in \mathscr{E}_{\mathbf{\Omega}2}^{(l-1)}} |z_{T_i,jl} - z_{T_k,jl}|, \tag{3.25}$$

where $\boldsymbol{A}_{T_i} = \mathbf{\Omega}_{T_i}^{t+1} + \boldsymbol{U}_{T_i}^t$. The above can be decomposed as the summation of terms for each item pair which are off-diagonal terms and terms for each item which are diagonal terms. Thus we can solve for each entry separately and in parallel. For pair $k < l$ which

at least one user rated, all relevant terms in (3.25) are

$$\sum_i (z_{T_i,kl} - a_{T_i,kl})^2 + \lambda_2 \sum_{i:(i,k,l)\in\mathscr{E}_{\boldsymbol{\Omega}1}^{(l-1)}} |z_{T_i,kl}| + \lambda_1 \sum_{(i,j):(i,j,k,l)\in\mathscr{E}_{\boldsymbol{\Omega}2}^{(l-1)}} |z_{T_i,kl} - z_{T_j,kl}|. \quad (3.26)$$

We use the alternating minimization algorithm to minimize

$$\sum_i (z_{T_i,kl} - a_{T_i,kl})^2 + \lambda_1 \sum_{(i,j):(i,j,k,l)\in\mathscr{E}_{\boldsymbol{\Omega}2}^{(l-1)}} |z_{T_i,kl} - z_{T_j,kl}|$$

as proposed in [14]. Then we do soft-thresholding to get the solution with the size penalty $\lambda_2 \sum_{i:(i,k,l)\in\mathscr{E}_{\boldsymbol{\Omega}1}^{(l-1)}} |z_{T_i,kl}|$. Diagonal entries can be solved in the same way, except without the size penalty.

Algorithm 2 ADMM algorithm for solving $\boldsymbol{\Omega}^{(l)}$

---

1. Start from some initial value $\boldsymbol{\Omega}^0, \boldsymbol{Z}^0, \boldsymbol{U}^0$.
2. $\boldsymbol{T}_i^t = \frac{1}{2}\boldsymbol{S}_i - \rho(\boldsymbol{Z}_i^t - \boldsymbol{U}_i^t)$. Suppose $\boldsymbol{T}_i^t$ has eigen decomposition $\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$, then update $\boldsymbol{\Omega}_i$ parallelly: $\boldsymbol{\Omega}_i^{t+1} = \boldsymbol{V}\boldsymbol{\Lambda}'\boldsymbol{V}^T$ with $\lambda'_{kk} = \dfrac{-\lambda_{kk} + \sqrt{\lambda_{kk}^2 + 2\rho}}{2\rho}$.
3. $\boldsymbol{Z}^{t+1} \leftarrow \operatorname{argmin}_{\boldsymbol{Z}}\{g_\rho(\boldsymbol{\Omega}^{t+1}, \boldsymbol{Z}, \boldsymbol{U}^t)\}$. That is, solve

$$\min_{\boldsymbol{Z}} \frac{1}{2}\sum_i \|\boldsymbol{Z}_{T_i} - \boldsymbol{A}_{T_i}\|_F^2 + \lambda_2 \sum_{(i,k,j,l)\in\mathscr{E}_{\boldsymbol{\Omega}}^{(l-1)}} |z_{T_i,jl} - z_{T_k,jl}|$$

   w.r.t. each entry of $\boldsymbol{Z}$ in parallel, where $\boldsymbol{A}_{T_i} = \boldsymbol{\Omega}_{T_i}^{t+1} + \boldsymbol{U}_{T_i}^t$.
4. Parallelly update $\boldsymbol{U}_{T_i}^{t+1} = \boldsymbol{U}_{T_i}^t + \boldsymbol{\Omega}_{T_i}^{t+1} - \boldsymbol{Z}_{T_i}^{t+1}$.
5. Terminate if change in $\boldsymbol{\Omega}$, $\boldsymbol{Z}$ and $\boldsymbol{U}$ are less than some precision ϵ. Otherwise repeat 2. 3. 4.

The dual variable $\boldsymbol{U}$ is updated as

$$\boldsymbol{U}_{T_i}^{t+1} = \boldsymbol{U}_{T_i}^t + \boldsymbol{\Omega}_{T_i}^{t+1} - \boldsymbol{Z}_{T_i}^{t+1}. \quad (3.27)$$

The algorithm to solve $\mathbf{\Omega}$ part is summarized in Algorithm 2.

The codes of the algorithm are all available at https://github.com/yang2732umn/RS_with_correlation.

3.2.4 Properties of the Algorithm

Proposition 3.2.1. *The estimate* $(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}, \hat{\mathbf{\Omega}})$ *from Algorithm 1 is a stationary point of the loss function* $-l_{TLP}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{\Omega})$ *in (3.7). For the* L_1 *method, the estimate from the* L_1*-version of Algorithm 1 is a stationary point of the loss function* $-l_1(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{\Omega})$ *in (3.5).*

Here the stationary point follows the same definition as in [55], i.e., all directional derivatives are non-negative.

For the computational complexity of the algorithm, the matrix inversion and matrix eigenvalue decomposition steps are $O((n+m)K^3 + \sum_{i=1}^{n} m_i^3)$. Suppose for item pair (j,k), it's rated by n_{jk} users. Then the computation complexity of solving (3.26) is $O(m^2 n_{jk}^2)$. Denote the number of difference of convex iterations as I_1, the number of ADMM iterations as I_2, and the number of the blockwise iterations as I_3, then the total computational complexity is $O(((n+m)K^3 + \sum_{i=1}^{n} m_i^3 + \sum_{j,k} n_{jk}^2) I_1 I_2 I_3)$.

For the storage cost of the algorithm, we look at the mean parameter part and precision matrix separately. The mean parameters need storage of user and item preference vectors, and is $(n+m)K$ numbers. For $\{\mathbf{\Omega}_i, i = 1, 2, \cdots, n\}$ the precision matrix part needs storage of $\sum_{i=1}^{n} m_i^2$ numbers where m_i is the number of items user i rated. For the part in $\mathbf{\Omega}_{-i}$ which represents the part of $\mathbf{\Omega}_{T_i}$ after taking $\mathbf{\Omega}_i$ out, suppose there are S pair of items rated by at least one user. Then the storage of $\{\mathbf{\Omega}_{-i}, i = 1, 2, \cdots, n\}$ only needs S numbers to record the corresponding entry of the precision matrix for the users who didn't rate this pair. This is due to the fact that, for a pair of items (j,k), for all users that didn't rate this pair, their $\omega_{T_i,jk}$'s are all equal. This can be easily seen from the definition of the penalized log likelihood function (3.3). Note $S < \frac{m^2}{2}$, so the storage cost is $O((n+m)K + \sum_{i=1}^{n} m_i^2 + \frac{m^2}{2})$.

3.3 Theoretical Results

In this section, the theoretical properties of our proposed method is provided in a general setting. For each user u, we allow $\boldsymbol{r}_u$ to follow a general distribution with mean related

parameters $\boldsymbol{\beta}_u$, $\boldsymbol{\alpha}$ and precision matrix related parameters $\boldsymbol{\Omega}_{T_u}$.

Let $\boldsymbol{\xi}$ represent a vectorized form of $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Omega}_T)$, and Ω be the indices of the user-item pairs corresponding to observed ratings. Denote the observed ratings by $R_\Omega = \{r_{uj} : (u, j) \in \Omega\}$. Let η_{uj} and $\boldsymbol{\eta}_u$ be defined as

$$\begin{aligned} \eta_{uj} &= \boldsymbol{x}_u^T \boldsymbol{\alpha}_j + \boldsymbol{y}_j^T \boldsymbol{\beta}_u, \\ \boldsymbol{\eta}_u &= \boldsymbol{\alpha}^T \boldsymbol{x}_u + \boldsymbol{Y} \boldsymbol{\beta}_u, \end{aligned} \tag{3.28}$$

where $\boldsymbol{Y} = (\boldsymbol{y}_1, \cdots, \boldsymbol{y}_m)^T$ is the item feature matrix.

Suppose the expected value of r_{uj} is a function of η_{uj}, which is

$$E(r_{uj}) = \mu(\eta_{uj}). \tag{3.29}$$

And suppose the covariance matrix of $\boldsymbol{r}_u$, the ratings on the m items by the uth user is a function of $\boldsymbol{\Omega}_{T_u}$, that is

$$Cov(\boldsymbol{r}_u) = \phi(\boldsymbol{\Omega}_{T_u}). \tag{3.30}$$

The distribution of the ratings by one user $\boldsymbol{r}_u$ can be either a multivariate continuous or categorical distribution. For example multivariate normal distribution has $\mu(\eta_{uj}) = \eta_{uj}$ and $\phi(\boldsymbol{\Omega}_{T_u}) = \boldsymbol{\Omega}_{T_u}^{-1}$. And if r_{uj} is bernoulli distribution, $\mu(\eta_{uj}) = \frac{1}{1+\exp(-\eta_{uj})}$.

Let $\boldsymbol{\theta}_A$ be a vectorization of $(\boldsymbol{\eta}, \boldsymbol{\Omega}_T)$, and $\boldsymbol{\theta}_{u,A}$ be the vectorization of $(\boldsymbol{\eta}_u, \boldsymbol{\Omega}_{T_u})$. Then the distribution of $\boldsymbol{r}_u$ depends on $\boldsymbol{\xi}$ only through $\boldsymbol{\theta}_{u,A}$. Denote the multivariate probability density function of $\boldsymbol{r}_u$ by $f_u = f(\boldsymbol{r}_u|\boldsymbol{\xi}) = f(\boldsymbol{r}_u|\boldsymbol{\theta}_{u,A})$. Then the density function of the observed part of $\boldsymbol{r}_u$ is also determined and denote it by $f_{u,o}$. The regularized negative log-likelihood function is defined as

$$L(\boldsymbol{\xi}|R_\Omega) = -\sum_u \log f_{u,o} + \lambda_{|\Omega|} D(\boldsymbol{\xi}), \tag{3.31}$$

where $\lambda_{|\Omega|}$ is the penalization coefficient, $|\Omega|$ is the size of the set Ω, which is the total number of observed ratings, and $D(\cdot)$ is a non-negative penalty function of $\boldsymbol{\xi}$ with several parts. It includes fused type penalty on the mean parameters and both fused type and size penalty on the precision matrices. For example, the L_1 version of $D(\cdot)$ is

$\frac{1}{2}\sum_{i<k}\|\boldsymbol{\alpha}_i-\boldsymbol{\alpha}_k\|_1+\frac{1}{2}\sum_{i<k}\|\boldsymbol{\beta}_i-\boldsymbol{\beta}_k\|_1+c_0\sum_i\sum_{j<l}|\omega_{T_i,jl}|+\sum_{i<k}\sum_{\substack{j\leqslant l\\ \exists h,\{j,l\}\subseteq I_h}}|\omega_{T_i,jl}-\omega_{T_k,jl}|$ where $c_0>0$ is a constant.

In practice, the ratings r_{uj}'s typically take non-negative finite values, so we can assume the size of the parameters and features are bounded by some constant. That is, $\|\boldsymbol{\xi}\|_\infty\leqslant\phi$, $\|\boldsymbol{X}\|_\infty\leqslant\phi$, $\|\boldsymbol{Y}\|_\infty\leqslant\phi$, where $\phi>0$ is a constant. The parameter vector space is defined as

$$\mathcal{S}(k)=\{\boldsymbol{\xi}:\|\boldsymbol{\xi}\|_\infty\leqslant\phi,\ D(\boldsymbol{\xi})\leqslant k^2\}. \tag{3.32}$$

Let $K=\max(K_1,K_2)$ be the larger number of the user and item feature vector dimensions. Suppose there are N and M clusters for user and item "preference" vectors. That is, in the n vectors of $\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\cdots,\boldsymbol{\beta}_n$, there are N unique vectors; and in the m vectors of $\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\cdots,\boldsymbol{\alpha}_m$, there are M unique vectors. Then the dimension of parameters in $(\boldsymbol{\alpha},\boldsymbol{\beta})$ is upper bounded by $(N+M)K$. For the paramters of $\{\boldsymbol{\Omega}_{T_u},u=1,\cdots n\}$, suppose there are $\tilde{N}$ distinct matrices, which are $\tilde{N}$ clusters. And the matrices are assumed to be sparse with at most $\tilde{K}<m$ non-zero entries in each row on average. Then the dimension of parameters in $\boldsymbol{\Omega}_T$ is upper bounded by $\tilde{N}m\tilde{K}$. Therefore the total dimension of $\boldsymbol{\xi}$ is $\dim(\boldsymbol{\xi})\leqslant(N+M)K+\tilde{N}m\tilde{K}$ and $\dim(\boldsymbol{\xi})$ goes to infinity as n,m goes to infinity. Since $\|\boldsymbol{\xi}\|_\infty\leqslant\phi$, we assume $k\sim O(\sqrt{(N(N-1)+M(M-1))K+\tilde{N}m\tilde{K}+\tilde{N}(\tilde{N}-1)m\tilde{K}})$. Similarly the parameter vector space for $\boldsymbol{\theta}_A$ is defined as

$$\mathcal{S}_{\Theta_A}(k)=\{\boldsymbol{\theta}_A:\|\boldsymbol{\xi}\|_\infty\leqslant L,\ D(\boldsymbol{\xi})\leqslant k^2\}. \tag{3.33}$$

Assumption 3.3.1. *There exists some constant $\bar{G}\geqslant 0$, and $\boldsymbol{\theta}_{u,A}$, $\tilde{\boldsymbol{\theta}}_{u,A}\in\mathcal{S}_{\Theta_A}(k)$,*

$$|f^{1/2}(\boldsymbol{r}_u|\boldsymbol{\theta}_{u,A})-f^{1/2}(\boldsymbol{r}_u|\tilde{\boldsymbol{\theta}}_{u,A})|\leqslant G(\boldsymbol{r}_u)\|\boldsymbol{\theta}_{u,A}-\tilde{\boldsymbol{\theta}}_{u,A}\|_2, \tag{3.34}$$

where $EG^2(\boldsymbol{r}_u)\leqslant\bar{G}^2$ for $u=1,\cdots,n$.

The Hellinger metric $h_{\Theta_A}(\cdot,\cdot)$ on $\mathcal{S}_{\Theta_A}(k)$ is defined as

$$h_{\Theta_A}(\boldsymbol{\theta}_{u,A},\tilde{\boldsymbol{\theta}}_{u,A})=\left[\int(f^{1/2}(\boldsymbol{r}_u|\boldsymbol{\theta}_{u,A})-f^{1/2}(\boldsymbol{r}_u|\tilde{\boldsymbol{\theta}}_{u,A}))^2d\nu(\boldsymbol{r}_u)\right]^{1/2}, \tag{3.35}$$

where $\nu(\cdot)$ is a probability measure. Based on Assumption 3.3.1, it's easy to see $h_{\Theta_A}(\boldsymbol{\theta}_{u,A}, \tilde{\boldsymbol{\theta}}_{u,A}) \leqslant \bar{G}\|\boldsymbol{\theta}_{u,A} - \tilde{\boldsymbol{\theta}}_{u,A}\|_2$. So it's bounded by $\|\boldsymbol{\theta}_{u,A} - \tilde{\boldsymbol{\theta}}_{u,A}\|_2$.

For $\boldsymbol{\xi}, \tilde{\boldsymbol{\xi}} \in \mathcal{S}(k)$, let

$$h_{\mathcal{S}(k)}(\boldsymbol{\xi}, \tilde{\boldsymbol{\xi}}) = \left[\frac{1}{n}\sum_{u=1}^{n} h^2_{\Theta_A}(\boldsymbol{\theta}_{u,A}, \tilde{\boldsymbol{\theta}}_{u,A})\right]^{1/2}. \tag{3.36}$$

It's easy to see that $h_{\mathcal{S}(k)}(\cdot,\cdot)$ is a still metric. For simplicity, in the following part of the dissertation, we use $h(\cdot,\cdot)$ to denote the Hellinger metric on $\mathcal{S}(k)$ and omit the subscript. In the lemma stated below, it is shown that $h(\boldsymbol{\xi}, \tilde{\boldsymbol{\xi}})$ is bounded by $\|\boldsymbol{\xi} - \tilde{\boldsymbol{\xi}}\|_2$.

Lemma 3.3.2. *Under Assumption 1, we can find a constant d_0, such that for $\boldsymbol{\xi}, \tilde{\boldsymbol{\xi}} \in \mathcal{S}(k)$,*

$$h(\boldsymbol{\xi}, \tilde{\boldsymbol{\xi}}) \leqslant d_0\|\boldsymbol{\xi} - \tilde{\boldsymbol{\xi}}\|_2\sqrt{\frac{\max(m,n)}{n}}. \tag{3.37}$$

Let $\hat{\boldsymbol{\xi}} = \arg\min_{\boldsymbol{\xi}\in\mathcal{S}(k)} L(\boldsymbol{\xi}|R_\Omega)$ be a penalized maximum likelihood estimator of $\boldsymbol{\xi}$, and let $\boldsymbol{\xi}_0$ be the true parameters. Theorem 3.3.3 states that $\hat{\boldsymbol{\xi}}$ converges to $\boldsymbol{\xi}$ exponentially in probability, with a convergence rate of $\epsilon_{|\Omega|}$.

Theorem 3.3.3. *Under Assumption 3.3.1 and suppose $\lambda_{|\Omega|} < \frac{1}{2k}\epsilon^2_{|\Omega|}$, there exists a constant $c > 0$, such that*

$$P(h(\boldsymbol{\xi}_0, \hat{\boldsymbol{\xi}}) \geqslant \epsilon_{|\Omega|}) \leqslant 7\exp(-c|\Omega|\epsilon^2_{|\Omega|}), \tag{3.38}$$

where

$$\epsilon_{|\Omega|} \sim \sqrt{\frac{(N+\tilde{N}m)\bar{K}}{|\Omega|}}\left(\log\frac{|\Omega|\sqrt{\max(n,m)}}{\sqrt{n(N+\tilde{N}m)\bar{K}}}\right)^{1/2}, \tag{3.39}$$

and $\bar{K} = \max(K, \tilde{K})$.

Remark 3.3.4. Theorem 3.3.3 is quite general in terms of the rates of n and m. As n, m goes to infinity, N and $\tilde{N}$ can also go to infinity, and $\frac{|\Omega|}{(N+\tilde{N}m)\bar{K}}$ should also go to infinity.

Remark 3.3.5. Our method assumes the ratings given by a single user are not independent, which is different from other recommender system models. When defining Hellinger distance, we use

$$h_{\mathcal{S}(k)}(\boldsymbol{\xi}, \tilde{\boldsymbol{\xi}}) = \left[\frac{1}{n} \sum_{u=1}^{n} \int (f^{1/2}(\boldsymbol{r}_u|\boldsymbol{\theta}_{u,A}) - f^{1/2}(\boldsymbol{r}_u|\tilde{\boldsymbol{\theta}}_{u,A}))^2 d\nu(\boldsymbol{r}_u) \right]^{1/2},$$

where $f(\boldsymbol{r}_u|\boldsymbol{\theta}_{u,A})$ is the density function of the random vector $\boldsymbol{r}_u$.

Other methods have the Hellinger distance as

$$\tilde{h}_{\mathcal{S}(k)}(\boldsymbol{\xi}, \tilde{\boldsymbol{\xi}}) = \left[\frac{1}{nm} \sum_{u=1}^{n} \sum_{i=1}^{m} \int (f^{1/2}(r_{ui}|\boldsymbol{\theta}_{ui}) - f^{1/2}(r_{ui}|\tilde{\boldsymbol{\theta}}_{ui}))^2 d\nu(r_{ui}) \right]^{1/2},$$

where $f(r_{ui}|\boldsymbol{\theta}_{ui})$ is the density function of the univariate random variable r_{ui}.

In order to compare the convergence rates of the Hellinger distances of our method and other methods, we need to convert the Hellinger distances to the same scale. To achieve this, we can define another Hellinger distance for other methods assuming independence of ratings by a single user. That is, let

$$\tilde{h}_{\mathcal{S}(k)}(\boldsymbol{\xi}, \tilde{\boldsymbol{\xi}}) = \left[\frac{1}{n} \sum_{u=1}^{n} \int (\prod_{i=1}^{m} f^{1/2}(r_{ui}|\boldsymbol{\theta}_{ui}) - \prod_{i=1}^{m} f^{1/2}(r_{ui}|\tilde{\boldsymbol{\theta}}_{ui}))^2 d\nu(\boldsymbol{r}_u) \right]^{1/2},$$

Or equivalently, our Hellinger distance should be divided by $\sqrt{m}$. That gives us a convergence rate of $\sqrt{\frac{(N+\tilde{N}m)\bar{K}}{|\Omega|}} \Big(\log \frac{|\Omega|\sqrt{\max(n,m)}}{\sqrt{nm(N+\tilde{N}m)\bar{K}}} \Big)^{1/2}$.

The following theorem indicates that if $\tilde{\boldsymbol{\xi}}$ is outside the $\epsilon_{|\Omega|}$-neighborhood of $\boldsymbol{\xi}_0$ in Hellinger distance, then the probability that the regularized negative log likehood of $\tilde{\boldsymbol{\xi}}$ be close to that of $\boldsymbol{\xi}_0$ is exponentially small.

Theorem 3.3.6. *Under Assumption 3.3.1 and $\lambda_{|\Omega|} < \frac{1}{2k}\epsilon_{|\Omega|}^2$, there exist $c_1 > 0, c_2 > 0$, such that for $\epsilon_{|\Omega|} > 0$ and $h(\boldsymbol{\xi}_0, \tilde{\boldsymbol{\xi}}) \geqslant \epsilon_{|\Omega|}$, the following holds:*

$$P^*\big(\tfrac{1}{|\Omega|}(L(\boldsymbol{\xi}_0|R_\Omega) - L(\tilde{\boldsymbol{\xi}}|R_\Omega)) \geqslant -c_1\epsilon_{|\Omega|}^2\big) \leqslant 7\exp(-c_2|\Omega|\epsilon_{|\Omega|}^2), \tag{3.40}$$

where P^* is the outer measure (see Pollard (1984)).

Now we assume the distribution of $\boldsymbol{r}_u$ is in the exponential family. That is, the density f_u is a member of the exponential family in its canonical form. Denote the canonical parameters by $\boldsymbol{\theta}_u$, we can write f_u as

$$f(\boldsymbol{r}_u|\boldsymbol{\theta}_u) = h(\boldsymbol{r}_u)\exp(\boldsymbol{\theta}_u^T T(\boldsymbol{r}_u) - A(\boldsymbol{\theta}_u)). \tag{3.41}$$

With the exponential family distribution assumption, we have the following corollary holds. It still holds when f is in the over-dispersed exponential family. Below we use $\boldsymbol{\theta}_{u0}$ to represent the true value of $\boldsymbol{\theta}_u$.

Corollary 3.3.7. *Under Assumption 3.3.1 and $\lambda_{|\Omega|} < \frac{1}{2k}\epsilon_{|\Omega|}^2$, there exist $c_1 > 0$, $c_2 > 0$, such that for $\epsilon_{|\Omega|} > 0$, there exists $\delta_{|\Omega|} > 0$, and $\min_{1\leqslant u\leqslant n} \|\tilde{\boldsymbol{\theta}}_u - \boldsymbol{\theta}_{u0}\|_1 > \delta_{|\Omega|}$ implies*

$$P^*\big(\tfrac{1}{|\Omega|}(L(\boldsymbol{\xi}_0|R_\Omega) - L(\tilde{\boldsymbol{\xi}}|R_\Omega)) \geqslant -c_1\epsilon_{|\Omega|}^2\big) \leqslant 7\exp(-c_2|\Omega|\epsilon_{|\Omega|}^2), \tag{3.42}$$

where P^* is the outer measure (see Pollard (1984)).

We also have the following result with a minor change in the condition of the l_1-norm between $\tilde{\boldsymbol{\theta}}_u$'s and $\boldsymbol{\theta}_{u0}$'s.

Corollary 3.3.8. *Under Assumption 3.3.1 and $\lambda_{|\Omega|} < \frac{1}{2k}\epsilon_{|\Omega|}^2$, there exist $c_i > 0$, $i = 1, 2$, and a constant $\phi \in (0, 1]$ such that for $\frac{1}{\sqrt{\phi}}\epsilon_{|\Omega|} > 0$, there exists $\delta_{|\Omega|} > 0$. Assume there are at least ϕn values of u satisfying $\|\tilde{\boldsymbol{\theta}}_u - \boldsymbol{\theta}_{u0}\|_1 > \delta_{|\Omega|}$, then*

$$P^*\big(\tfrac{1}{|\Omega|}(L(\boldsymbol{\xi}_0|R_\Omega) - L(\tilde{\boldsymbol{\xi}}|R_\Omega)) \geqslant -c_1\epsilon_{|\Omega|}^2\big) \leqslant 7\exp(-c_2|\Omega|\epsilon_{|\Omega|}^2). \tag{3.43}$$

Proofs of the results are given in Appendix A.2.

3.4 Advantage of Using Precision Matrix

3.4.1 Correlation Validation on Data

We first use some real data to show that it's reasonable and necessary to assume the ratings given by users are correlated. The Movielens 100k dataset is investigated here

for an illustration. The movielens data are collected by GroupLens Research Lab at the University of Minnesota. They are available at https://grouplens.org/datasets/movielens/. The 100k dataset contains 100,000 ratings (1-5) from 943 users on 1682 movies. We use ratings on a pair of movies from all users that rated them both to assess the correlation between the two movies. By calculating the correlations between ratings of movies, we find some movie pairs present high correlations.

For example, the correlation between Star Wars (1977) and The Empire Strikes Back (1980), the sequel of the former, is 0.75 based on 345 user ratings; and the correlation between The Empire Strikes Back and its sequel Return of the Jedi (1983) is also quite high value of 0.72 based on 317 users who rated both. Not surprisingly, Star Wars (1977) and Return of the Jedi (1983) also have a high positive correlation of 0.67 based on 480 users. Other examples include the God Father I (1972) and II (1974) which has a correlation of 0.68, and the Die Hard (1990) and Die Hard with a Vengeance (1995) which has a correlation of 0.75 etc. Two graphs showing the correlations in this data are shown in Appendix B. This reflects the fact that people's ratings on a movie series tend to be highly positively correlated, though there may be years of time between watching two movies. Also, we notice that the Movielens 100k data only have movies before or in 1998. As time goes on, the same IP fuels more and more movies, which certainly strengthens the correlations between different items. Hence with more recent or future movie data, this phenomenon of correlation will be more evident.

3.4.2 Outperformance of the Correlated Linear Model Using Prediction Error as a Criterion

With the correlation assumption validated on data, if a statistical model assumes independence of ratings, it cannot depict the true underlying distribution. Typically if the true distribution is dependent, using a model assuming independence leads to larger variances for the coefficient estimates as well as the predictions. For instance, the ordinary least squares estimator is the best linear unbiased estimator for ordinary linear regression, but loses efficiency (in terms of larger variances) if the random errors are not independent and identically distributed. An analogy applies to our case here. While estimating the correlation structure via the precision matrix in our proposed method, we can isolate the mean effect from the covariance effect and reduce variability in the

parameter estimates. This way the mean parameters are estimated more accurately and thus the predicted ratings are more precise.

Through estimating the precision matrix, we can also have a grouping of users, according to the correlations on items. Our method of estimating the precision matrix generalizes the method of [63] by allowing incompleteness in the precision matrices. We don't require estimating the whole $m \times m$ precision matrix, but only the part where we have information available, either from the corresponding user or other users. Also, the grouping is automatically given through our algorithm. Note that estimation of the precision matrix requires a large amount of computation, and we can reduce the effort by the above-mentioned grouping. Moreover, the grouping of correlation makes it possible for our method to obtain more accuracy than other methods.

Next we show theoretically employing the covariance structure gives the benefit of a smaller asymptotic variance of the estimators and smaller prediction errors.

First we reformulate our model (3.1) as a linear regression with correlation structure. Let $\boldsymbol{r}$ be the vectorized response, $\boldsymbol{\gamma}$ be the vectorized form of $(\boldsymbol{\alpha}, \boldsymbol{\beta})$, $\boldsymbol{W}$ be the design matrix, and $\boldsymbol{\Omega}$ be the block-diagonal matrix of $\boldsymbol{\Omega}_i$'s as follows:

$$\boldsymbol{r} = \begin{pmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \\ \vdots \\ \boldsymbol{r}_n \end{pmatrix}, \ \boldsymbol{\gamma} = \begin{pmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_n \\ \boldsymbol{\alpha}_1 \\ \vdots \\ \boldsymbol{\alpha}_m \end{pmatrix}, \ \boldsymbol{\Omega} = \begin{pmatrix} \boldsymbol{\Omega}_1 & & \\ & \ddots & \\ & & \boldsymbol{\Omega}_n \end{pmatrix}. \tag{3.44}$$

So $\boldsymbol{r}$ has dimension $|\Omega| = \sum_{i=1}^{n} m_i$ which is the total number of observed ratings, $\boldsymbol{\gamma}$ has dimension $mK_1 + nK_2$. The design matrix $\boldsymbol{W}$ has dimension $|\Omega| \times (mK_1 + nK_2)$ and each row of $\boldsymbol{W}$ contains the item and user features related to a specific rating in $\boldsymbol{r}$ at the corresponding location. For example, if user 1 rated item 1, then the first row $\boldsymbol{w}_1$ of $\boldsymbol{W}$ is $(\boldsymbol{y}_1^T, \boldsymbol{0}, \cdots, \boldsymbol{0}, \boldsymbol{x}_1^T, \boldsymbol{0}, \cdots, \boldsymbol{0})$. So the model can be equivalently written as

$$\boldsymbol{r} = \boldsymbol{W}\boldsymbol{\gamma} + \boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \boldsymbol{\Omega}^{-1}). \tag{3.45}$$

Note this is exactly the same model as in our previous model specification (3.1). And for this linear regression model, the sample size is $|\Omega|$ and the number of regression coefficients is $mK_1 + nK_2$. Since our model assumes there's grouping among user preferences and among item preferences, we only penalize the differences between different $\boldsymbol{\beta}_i$'s and different $\boldsymbol{\alpha}_j$'s. It's easy to check all the theories and corollaries below apply to this special case.

We prove that for a linear regression model with dependent covariance of the random errors and grouping structure in the parameters, if the covariance matrix is known and the penalty is TLP on pairwise differences, then adopting the dependent likelihood in the penalized log-likelihood gives uniformly smaller asymptotic variances of the parameters than adopting the independent likelihood. Furthermore, in terms of prediction, using the dependent likelihood gives smaller prediction error compared to the independent likelihood. In the following discussion, instead of using notation of the special case in (3.45), we adopt the notation of a general linear regression setting.

Assume in linear regression, there is a correlation structure in the random errors, i.e.

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta}_n + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N(0, \boldsymbol{\Omega}^{-1}), \tag{3.46}$$

where $\boldsymbol{Y}_{n\times 1}$ is the response, $\boldsymbol{X}_{n\times p_n}$ is the design matrix, $\boldsymbol{\Omega}_{n\times n}$ is a general positive definite matrix and not in the form of $\sigma^2 I$. For $\boldsymbol{X}$, we treat it as given and fixed here. For $\boldsymbol{\beta}_n$, the subscript is to emphasize that its dimension p_n is allowed to go to infinity as n goes to infinity and it's assumed to have a grouping structure. Specifically, let $\boldsymbol{\beta}_{n0}$ be the true value of $\boldsymbol{\beta}_n$. Suppose in $\boldsymbol{\beta}_{n0}$ there are s_n groups, i.e. s_n unique values. Without loss of generality, let the first s_n values of $\boldsymbol{\beta}_{n0}$ be distinct, and each of the rest $p_n - s_n$ elements share the same value as one of $\beta_{n10}, \cdots, \beta_{ns_n0}$. Denote

$$\boldsymbol{\beta}_{n0} = (\boldsymbol{\beta}_{n10}^T, \boldsymbol{\beta}_{n20}^T)^T, \tag{3.47}$$

where $\boldsymbol{\beta}_{n10} = (\beta_{n10}, \cdots, \beta_{ns_n0})^T$ and $\boldsymbol{\beta}_{n20} = (\beta_{n(s_n+1)0}, \cdots, \beta_{np_n0})^T$. Let

$$t_n(\cdot) : \{s_n + 1, \cdots, p_n\} \to \{1, \cdots, s_n\} \tag{3.48}$$

be the function that has $\beta_{nj0} = \beta_{nt_n(j)0}$ for $j = s_n + 1, \cdots, p_n$. Let $\boldsymbol{Z}_{n\times s_n}$ be the

condensed design matrix, i.e.

$$\boldsymbol{z}_i = \boldsymbol{x}_i + \sum_{t_n(j)=i} \boldsymbol{x}_j, \tag{3.49}$$

where $\boldsymbol{z}_i$ and $\boldsymbol{x}_j$ represents the ith column of $\boldsymbol{Z}$ and the jth column of $\boldsymbol{X}$ respectively.

The log-likelihood function and the penalized log-likelihood function about $\boldsymbol{\beta}_n$ are

$$\begin{aligned} S_n(\boldsymbol{\beta}_n) &= -\frac{1}{2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}_n)^T \boldsymbol{\Omega}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}_n), \\ L_n(\boldsymbol{\beta_n}) &= S_n(\boldsymbol{\beta}_n) - n \sum_{1 \leqslant j < k \leqslant p_n} p_{\lambda_n,\tau_n}(|\beta_{nj} - \beta_{nk}|), \end{aligned} \tag{3.50}$$

where p is the TLP function with $p_{\lambda,\tau}(x) = \lambda \min(\frac{|x|}{\tau}, 1)$[1].

But if the model is misspecified as independent, we have the incorrect log-likelihood and the penalized log-likelihood

$$\begin{aligned} \tilde{S}_n(\boldsymbol{\beta}_n) &= -\frac{1}{2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}_n)^T (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}_n), \\ \tilde{L}_n(\boldsymbol{\beta_n}) &= \tilde{S}_n(\boldsymbol{\beta}_n) - n \sum_{1 \leqslant j < k \leqslant p_n} p_{\lambda_n,\tau_n}(|\beta_{nj} - \beta_{nk}|). \end{aligned} \tag{3.51}$$

In the following with some assumptions on the design matrices, we give two theorems about the properties of the penalized log-likelihood estimators. And in two corollaries it's shown that the dependence estimator uniformly outperforms the independence estimator in terms of the asymptotic variance and prediction error.

We have some regularity conditions on the design matrices and precision matrices to guarantee the asymptotic properties of the estimators: there exist constants h_1 and h_2 such that for all n, the following holds

(i) $0 < h_1 < \lambda_{\min}(\frac{1}{n}\boldsymbol{X}^T\boldsymbol{X}) \leqslant \lambda_{\max}(\frac{1}{n}\boldsymbol{X}^T\boldsymbol{X}) < h_2 < \infty$.

(ii) $0 < h_1 < \lambda_{\min}(\frac{1}{n}\boldsymbol{X}^T\boldsymbol{\Omega}^{-1}\boldsymbol{X}) \leqslant \lambda_{\max}(\frac{1}{n}\boldsymbol{X}^T\boldsymbol{\Omega}^{-1}\boldsymbol{X}) < h_2 < \infty$.

(iii) $0 < h_1 < \lambda_{\min}(\frac{1}{n}\boldsymbol{X}^T\boldsymbol{\Omega}\boldsymbol{X}) \leqslant \lambda_{\max}(\frac{1}{n}\boldsymbol{X}^T\boldsymbol{\Omega}\boldsymbol{X}) < h_2 < \infty$.

where $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ are the smallest and largest eigenvalues of a matrix.

[1]Here for convenience, a notation different from Section 3.1 and 3.2 is used for TLP. Note $p_{\lambda,\tau}(x) = \lambda J_\tau(x)/\tau$.

These regularity conditions assume the design matrix $\boldsymbol{X}$ is reasonably good. This type of condition was considered for example in [39, 15, 19, 66, 17]. As pointed out by [39], condition (i) holds when its row vectors $\{\boldsymbol{x}_i\}$ behave like a random sample from an appropriate multivariate distribution. Note condition (i) is exactly the same as the condition (A) in [15], condition (F) in [19], condition (A1) of [66] and condition (C) of [17]. Let the sequence of multivariate distributions of the rows of $\boldsymbol{X}$ have mean $\{\boldsymbol{\mu}_n\}$ and covariance matrix $\{\boldsymbol{\Sigma}_n\}$. Then the requirement is that $\{|\mu_{nj}|\}$ has uniform upper bound for all n, j and the sequence of $\{\boldsymbol{\Sigma}_n\}$ has bounded minimum and maximum eigenvalue.

Given that condition (i) above holds, one sufficient condition that guarantees conditions (ii) and (iii) is: there exist constants d_1 and d_2 such that

$$0 < d_1 < \lambda_{\min}(\boldsymbol{\Omega}) \leqslant \lambda_{\max}(\boldsymbol{\Omega}) < d_2 < \infty. \tag{3.52}$$

This is based on the observation that $\lambda_{\min}(\frac{1}{n}\boldsymbol{X}^T\boldsymbol{\Omega}\boldsymbol{X}) \geqslant \lambda_{\min}(\frac{1}{n}\boldsymbol{X}^T\boldsymbol{X})\lambda_{\min}(\boldsymbol{\Omega})$ and $\lambda_{\max}(\frac{1}{n}\boldsymbol{X}^T\boldsymbol{\Omega}\boldsymbol{X}) \leqslant \lambda_{\max}(\frac{1}{n}\boldsymbol{X}^T\boldsymbol{X})\lambda_{\max}(\boldsymbol{\Omega})$. So it requires the sequence of the covariance matrices of the response variables also have uniformly bounded minimum and maximum eigenvalue, which is also typically satisfied.

Next we state the theories about the penalized log-likelihood estimators.

Theorem 3.4.1. *Assume $\boldsymbol{\Omega}$ is known for all n, and the likelihood function is correctly specified as in (3.50). If in addition to the regularity conditions on the design matrices and precision matrices, the following conditions are satisfied,*

(i') $\liminf_{n\to\infty} \frac{1}{\tau_n}\min(|\beta_{nj0} - \beta_{nk0}| : \beta_{nj0} \neq \beta_{nk0}) > 1$

(ii') $\frac{p_n}{n} \to 0$

(iii') $\frac{\tau_n}{\sqrt{p_n/n}} \to \infty$

(iv') $\sqrt{\frac{n}{p_n}}\frac{\lambda_n}{\tau_n} \to \infty$

then as $n \to \infty$ the statements below hold:

(1) There exists a $\sqrt{n/p_n}$-consistent local maximizer $\hat{\boldsymbol{\beta}}_n$ of the penalized log-likelihood L_n, i.e. $\|\hat{\boldsymbol{\beta}}_n - \boldsymbol{\beta}_{n0}\|_2 = O_p(\sqrt{p_n/n})$.

(2) Grouping consistency: With probability tending to 1, $\hat{\beta}_{nj} = \hat{\beta}_{nt_n(j)}$ for $j > s_n$.

(3) With probability tending to 1: $\hat{\boldsymbol{\beta}}_{n1}$ is unbiased and for any dim-s_n vector $\boldsymbol{z}$,

$$\mathrm{var}(\boldsymbol{z}^T(\hat{\boldsymbol{\beta}}_{n1} - \boldsymbol{\beta}_{n10})) = \boldsymbol{z}^T \left(\boldsymbol{Z}^T \boldsymbol{\Omega} \boldsymbol{Z}\right)^{-1} \boldsymbol{z}.$$

Theorem 3.4.2. *Assume $\boldsymbol{\Omega}$ is known for all n, if the likelihood function is misspecified as independent as in (3.51), with the same conditions as in theorem 3.4.1, the following holds:*

(1) There exists a $\sqrt{n/p_n}$-consistent local maximizer $\tilde{\boldsymbol{\beta}}_n$ of the penalized log-likelihood $\tilde{L}_n$, i.e. $\|\tilde{\boldsymbol{\beta}}_n - \boldsymbol{\beta}_{n0}\|_2 = O_p(\sqrt{p_n/n})$.

(2) Grouping consistency: With probability tending to 1, $\tilde{\beta}_{nj} = \tilde{\beta}_{nt_n(j)}$ for $j > s_n$.

(3) With probability tending to 1: $\tilde{\boldsymbol{\beta}}_{n1}$ is unbiased and for any dim-s_n vector $\boldsymbol{z}$,

$$\mathrm{var}(\boldsymbol{z}^T(\tilde{\boldsymbol{\beta}}_{n1} - \boldsymbol{\beta}_{n10})) = \boldsymbol{z}^T(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}(\boldsymbol{Z}^T\boldsymbol{\Omega}^{-1}\boldsymbol{Z})(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\boldsymbol{z}.$$

Corollary 3.4.3. *For any dim-p_n vector $\boldsymbol{x}$, and the estimators $\hat{\boldsymbol{\beta}}_n$ and $\tilde{\boldsymbol{\beta}}_n$ in theorem 3.4.1 and theorem 3.4.2 respectively, the asymptotic variance of $\boldsymbol{x}^T\hat{\boldsymbol{\beta}}_n$ is smaller than or equal to that of $\boldsymbol{x}^T\tilde{\boldsymbol{\beta}}_n$. That implies if asymptotic variance is used as the criterion, $\hat{\boldsymbol{\beta}}_n$ uniformly outperforms the independence estimator $\tilde{\boldsymbol{\beta}}_n$.*

The corollary indicates that at any $\boldsymbol{x}$ for prediction, the dependent estimator achieves smaller asymptotic variance than the independence estimator. Consequently, for our setting of (3.45), at any vector $\boldsymbol{w}$ for prediction, asymptotically the variance of $\boldsymbol{w}^T\hat{\boldsymbol{\gamma}}$ with $\hat{\boldsymbol{\gamma}}$ being the dependence estimator using the correct dependent likelihood is smaller than or equal to that of $\boldsymbol{w}^T\tilde{\boldsymbol{\gamma}}$ with $\tilde{\boldsymbol{\gamma}}$ being the independence estimator using the misspecified independent likelihood.

Corollary 3.4.4. *For any dim-p_n vector $\boldsymbol{x}$, and the estimators $\hat{\boldsymbol{\beta}}_n$ and $\tilde{\boldsymbol{\beta}}_n$ in theorem 3.4.1 and theorem 3.4.2 respectively, with probability tending to 1, the prediction error of using estimator $\hat{\boldsymbol{\beta}}_n$ is smaller than or equal to that using $\tilde{\boldsymbol{\beta}}_n$.*

This corollary shows taking the correlation structure into account which is realized in our method delivers smaller prediction error and enhances the prediction accuracy. So theoretically we've shown the dependence estimator which utilizes the correlation structure outperforms the independence estimator. Next, we will use numerical results to verify this point.

Remark 3.4.5. For our model of (3.45), in addition to $\{\boldsymbol{\Omega}\}$ eigenvalues has a uniform lower and upper bound, assuming a similar condition to (i) that $\{\frac{1}{|\Omega|}\boldsymbol{W}^T\boldsymbol{W}\}$ uniformly have minimum and maximum eigenvalues bounds at the same rate, in particular $O(n) = O(m)$ which is usually the case in real data, then all theories and corollaries can be derived with similar arguments.

Chapter 4

Numerical Results

4.1 Simulation Studies

For simulation, 100 users and 30 items are used. To select the tuning parameters, we create the train, tune and test datasets and choose the tuning parameter that gives the best performance on the tuning data. The feature of each movie is a size 18 vector with values 0 and 1 with every element generated independently from the Bernoulli distribution with probability 0.5 (a constant feature of 1 is added as another item feature to fit individual user intercept effect); and the feature of each user is a size 24 vector with values 0 and 1, also generated from Bernoulli(1,0.5). It is set up in this way to be similar to the Movielens user and item features, which are also coded as 0's and 1's. In all cases we used 12 clusters for user "preferences" $\boldsymbol{\beta}$ and 10 clusters for item "preferences" $\boldsymbol{\alpha}$. All users in the same cluster have the same $\boldsymbol{\beta}$, and all items in the same cluster have the same $\boldsymbol{\alpha}$. We assign users and items to clusters, making the number of users or items in each cluster as close to each other as possible. For example, there are 100 users, so 8 clusters have 8 users and 4 clusters have 9 users. The 12 distinct $\boldsymbol{\beta}$ vectors and 10 distinct $\boldsymbol{\alpha}$ vectors are randomly generated from the multivariate normal distribution with some mean vector and identity covariance matrix. In our simulation, we used 8 for the mean of all elements in $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Use $\boldsymbol{Y}$ as the matrix of item features as above, and $\boldsymbol{\alpha}$ the item "preference" matrix also as defined before, the mean of user i's rating is $\boldsymbol{\mu}_i = \boldsymbol{\alpha}^T \boldsymbol{x}_i + \boldsymbol{Y}\boldsymbol{\beta}_i$. The missing probability for each user is set to be 80%. So on average, each user rated 6 items out of 30.

The random noise $\boldsymbol{\epsilon}_i$ is from a multivariate normal distribution with mean $\mathbf{0}$ and precision matrix $\boldsymbol{\Omega_i}$. In the simulation, we chose precision matrices which can lead to large correlations between parts of the ratings of the same user. After constructing two 30 by 30 such precision matrices denoted as $\boldsymbol{\Omega}_{s_1}$ and $\boldsymbol{\Omega}_{s_2}$, $\boldsymbol{\Omega}_{s_1}$ is used for the first 80 users and $\boldsymbol{\Omega}_{s_2}$ is used for the last 20 users.We take the inverse of the precision matrix for the train data to get its covariance matrix. Then the covariance matrix on train dataset is expanded to a larger covariance for the union of the train, tune and test dataset. The train, tune and test data random errors are generated together according to the expanded covariance matrix.

For each user, 60% of the observed ratings are randomly selected for the train set, and 20% for the tune set, 20% for the test set. To be able to predict the rating for each item of each user, we made sure that in the train set each user at least had one observed rating, and each item was rated at least by one user. To ensure every element in the tune precision matrix is estimated, we made another restriction that if a pair of items was observed for at least one user in the union of the train, tune and test dataset, then there was also at least one user who observed this pair in the train set. We used the log-likelihood as our criterion to select tuning parameter. Submatrix of the estimated 30 by 30 precision matrix of each user corresponding to the items in tune set is used as the estimated precision matrix for the tune set. 100 simulations are performed.

The code for implementation of our method is written in C++, and the OpenMP API for parallel computing is applied whenever the computation can be done in parallel. Also, when possible we always used the warm start initial values for the parameters, either from a previous set of tuning parameters or from a previous method. This can give faster convergence of the algorithm.

We compared seven models: (a.) the linear regression model using rating as the response and user and item features as predictors, (b.) the SOFT-IMPUTE in [34] that penalizes the nuclear norm in matrix completion, (c.) the regularized singular value decomposition method (RSVD) discussed in [22] and [30], (d.) a regression-based latent factor model (RLFM) in [3], (e.) the special the case of our proposed model which doesn't consider the precision matrix with L_1 norm clustering, (f.) our proposed model with L_1 norm clustering and (g.) our proposed model with TLP penalized clustering.

For the SOFT-IMPUTE method, grid points from 0.01 to 20 are selected for the

regularization parameter λ. For the RSVD method, the regularization parameter takes grid values from 0.1 to 2. For the RLFM, we used 6 latent factors. The special case of the proposed method without precision matrix used grid values from 0.05 to 800. The proposed model with L_1 norm used three values for λ_1 as 10, 5, 1 and three values for λ_2 as 1, 0.1, 0.05. The proposed model with TLP norm used three values for λ_1 as 5, 0.5 ,0.2, and three values for λ_2 as 1, 0.1, 0.05, and two values for τ as 0.01, 0.005.

To compare the performance of the different methods, the root mean squared error (RMSE) and weighted root mean squared error (wRMSE) on the test set are calculated. The wRMSE used the true test set precision matrix to weight the errors. As the data is generated with heterogeneous error, the wRMSE is our main criterion. Results are summarized in Table 4.1.

Table 4.1: Simulation results for seven methods are reported: LM is the linear regression model using rating as the response and user and item features as predictors; SOFT is the SOFT-IMPUTE method; RSVD is regularized singular value decomposition; s-L1 is special L_1 clustering ignoring precision matrix; RLFM is the regression-based latent factor model; g-L1 is the general L_1 clustering (considering precision matrix); g-TLP is the general TLP clustering (considering precision matrix). Numbers in the parentheses are the standard errors.

	wRMSE	RMSE
LM	33.781(16.619)	5.983(1.963)
SOFT	80.605(36.839)	11.038(3.789)
RSVD	66.853(23.401)	10.785(1.816)
RLFM	43.362(30.011)	7.722(5.008)
s-L1	27.494(9.087)	10.636(1.374)
g-L1	25.001(6.959)	5.029(1.399)
g-TLP	24.752(6.508)	5.021(1.475)

The table indicates that our proposed models perform much better than the Linear Model, the Soft-Impute, RSVD and RLFM methods. Specifically, the proposed three methods improve over the Linear Model, Soft-Impute, RSVD, RLFM in wRMSE by about 23%, 67%, 61%, 40% respectively. And in terms of RMSE, the general L_1 and TLP clustering methods considering precision matrix show improvements over the Linear Model, the Soft-Impute, RSVD and RLFM by about 16%, 54%, 53%, 35% respectively.

Also, for both criteria, the general L_1 and TLP clustering methods considering precision matrix perform better than the special L_1 clustering method ignoring precision matrix. Note the linear model (LM) performs better compared to the Soft-Impute, RSVD and RLFM methods due to the fact that the simulation data are generated from a linear model with dependent errors. All the comparisons mentioned here are tested to be highly significant with very small p-values via two-sample t-tests.

4.2 Movielens Data

As mentioned in Chapter 3, the movielens data are collected by GroupLens Research Lab at the University of Minnesota. And they are available at https://grouplens.org/datasets/movielens/. We compared seven methods including linear regression, Soft-Impute, RSVD, RLFM, special L_1 method ignoring precision matrix, L_1 method considering the precision matrix, TLP method considering precision matrix on the Movielens 100k dataset. It contains 100,000 ratings (1-5) from 943 users on 1682 movies. User features used include age, gender and occupation. Movie features used include genre and release year. We deleted movies rated by no more than 5 users, and that left us with 1298 movies. The missing percentage of the data is 92%. We divided the data for each user to 60% training, 20% tuning and 20% testing, and made sure each item, each user, and each item pair that appeared in the whole dataset also appeared in the training data.

For the seven methods, except the LM and RLFM, tuning parameters are selected from grid points to minimize the negative log-likelihood on the tuning data. For RLFM, we used 10 latent factors in the model. We calculated the RMSE for the seven methods. Since the true covariance matrix on the test data is unavailable, the wRMSE is omitted. The results are summarized in Table 4.2. The general TLP clustering method considering precision matrix outperforms the Linear Model, the Soft-Impute method, the RSVD method, the RLFM method, the special L_1 clustering method ignoring precision matrix, the general L_1 clustering method considering precision matrix with the amount of improvement 12.37%, 5.31%, 1.18%, 0.57%, 0.30% and 0.13% respectively. So the proposed TLP penalized method considering precision matrix has the best performance and our three models deliver higher predictive accuracy compared to the other

four methods.

Table 4.2: Movielens 100k RMSE with seven methods: LM is the linear regression model using rating as the response and user and item features as predictors; SOFT is the Soft-Impute method; RSVD is regularized singular value decomposition; RLFM is the regression-based latent factor model; s-L1 is special L_1 clustering ignoring precision matrix; g-L1 is the general L_1 clustering (considering precision matrix); g-TLP is the general TLP clustering (considering precision matrix).

	LM	SOFT	RSVD	RLFM	s-L1	g-L1	g-TLP
RMSE	1.0632	0.9840	0.9428	0.9370	0.9345	0.9329	0.9317

Chapter 5

Conclusion and Discussion

We propose and explore a personalized recommender system via clustering users and items based on their individual "preferences". Besides modeling the mean structure using user and item features, we also model the covariance structure between ratings given by the same user through estimating the individual precision matrices. This is a major contribution of our work and no other recommender system in the literature which does covariance parameter estimation are noticed. Through this thesis, methods of undirected graphical models are introduced into personalized recommender systems. Theoretically, it's shown when covariance exists, taking the covariance into account uniformly gives a smaller asymptotic variance of the estimators and smaller prediction errors. Numerical results indicate estimating the covariance parameters improves the prediction accuracy significantly.

For extension of our method, we can add latent features in the model for users and items. We can also incorporate context information by adding extra terms for preference of users and items on different context features and preference of contexts on user and item features. For the correlation part, for example we can consider only the correlations of user ratings on items under one context. Computationally, it is also an option to use Maximum block improvement, in this way one can have the linear convergence rate of the algorithm. Finally, as time goes by, the correlations between movies become more significant, and it is natural to expect our method being more efficient in some new or future movie data sets.

References

[1] Gediminas Adomavicius and Alexander 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*, 17(6):734–749, 2005.

[2] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender Systems Handbook*, pages 217–253. Springer US, 2011.

[3] Deepak Agarwal and Bee-Chung Chen. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 19–28, New York, NY, USA, 2009. ACM.

[4] Asim Ansari, Skander Essegaier, and Rajeev Kohli. Internet recommendation systems. *Journal of Marketing Research*, 37(3):363–375, 2000.

[5] Xuan Bi, Annie Qu, and Xiaotong Shen. Multilayer tensor factorization with applications to recommender systems. *The Annals of Statistics*, To Appear, 2017.

[6] Xuan Bi, Annie Qu, Junhui Wang, and Xiaotong Shen. A group-specific recommender system. *Journal of the American Statistical Association*, 112(519):1344–1353, 2017.

[7] Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl. *Content-Based Recommendation Systems*, pages 325–341. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[8] Robin Burke. Knowledge-based recommender systems. In *Encyclopedia of Library and Information Systems*, page 2000. Marcel Dekker, 2000.

[9] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[10] Fidel Cacheda, Víctor Carneiro, Diego Fernández, and Vreixo Formoso. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Trans. Web*, 5(1):2:1–2:33, February 2011.

[11] Fidel Cacheda, Víctor Carneiro, Diego Fernández, and Vreixo Formoso. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Trans. Web*, 5(1):2:1–2:33, February 2011.

[12] Jorge Castro, Rosa M. Rodriguez, and Manuel J. Barranco. Weighting of features in content-based filtering with entropy and dependence measures. *International Journal of Computational Intelligence Systems*, 7(1):80–89, 2014.

[13] Jie Chen and Yousef Saad. On the tensor svd and the optimal low rank orthogonal approximation of tensors. *SIAM Journal on Matrix Analysis and Applications*, 30(4):1709–1734, 2009.

[14] Eric C. Chi and Kenneth Lange. Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics*, 23(1):111–128, 2014.

[15] Hyunkeun Cho and Annie Qu. Model selection for correlated data with diverging number of parameters. *Statistica Sinica*, 23(2):901–927, 2013.

[16] Patrick Danaher, Pei Wang, and Daniela M. Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B*, 76(2):373–397, 2014.

[17] Lee Dicker, Baosheng Huang, and Xihong Lin. Variable selection and estimation with the seamless-l ¡sub¿0¡/sub¿ penalty. *Statistica Sinica*, 23(2):929–962, 2013.

[18] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

[19] Jianqing Fan and Heng Peng. Nonconcave penalized likelihood with a diverging number of parameters. *The Annals of Statistics*, 32(3):928–961, 2004.

[20] Rina Foygel, Nathan Srebro, and Ruslan Salakhutdinov. Matrix reconstruction with the local max norm. In *NIPS*, pages 944–952, 2012.

[21] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, December 2003.

[22] Simon Funk. Netflix update: Try this at home. http://sifter.org/simon/journal/20061211.html, 2006.

[23] Asela Gunawardana and Christopher Meek. A unified approach to building hybrid recommender systems. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 117–124, New York, NY, USA, 2009. ACM.

[24] Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. Recommending and evaluating choices in a virtual community of use. *Proc. Conf. Human Factors in Computing Systems*, 1995.

[25] Holger Hoefling. A path algorithm for the fused lasso signal approximator. *Journal of Computational and Graphical Statistics*, 19(4):984–1006, 2010.

[26] Mingyi Hong and Zhi-Quan Luo. On the linear convergence of the alternating direction method of multipliers. *Math. Program.*, 162(1-2):165–199, March 2017.

[27] Cho-Jui Hsieh, Matyas A. Sustik, Inderjit S. Dhillon, and Pradeep Ravikumar. Sparse inverse covariance matrix estimation using quadratic approximation. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2330–2338. http://nips.cc/, 2011.

[28] Alexandros Karatzoglou, Xavier Amatriain, Nuria Oliver, and Linas Baltrunas. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *In Proceedings of the fourth ACM conference on Recommender systems*, 2010.

[29] Yehuda Koren. Collaborative filtering with temporal dynamics. In *In Proc. of KDD '09*, pages 447–456, 2009.

[30] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems, 2009.

[31] Roderick J. A. Little and Donald B. Rubin. *Statistical Analysis with Missing Data.* John Wiley and Sons, Inc., second edition edition, 2002.

[32] Lester Mackey. Stanford practical machine learning collaborative filtering slides.

[33] Xiaojun Mao, Song Xi Chen, and Raymond K. W. Wong. Matrix completion with covariate information. *Journal of the American Statistical Association*, To Appear, 2017.

[34] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11, 2010.

[35] Koji Miyahara and Michael J. Pazzani. Improvement of collaborative filtering with the simple bayesian classifier. *Information Processing Society of Japan*, 43(11), 2002.

[36] Jennifer Nguyen and Mu Zhu. Content-boosted matrix factorization techniques for recommender systems. *Statistical Analysis and Data Mining*, 6(4):286 – 301, 2013.

[37] Mina Ossiander. A central limit theorem under metric entropy with l_2 bracketing. *The Annals of Probability*, 15(3):897–919, 1987.

[38] Wei Pan, Xiaotong Shen, and Binghui Liu. Cluster analysis: Unsupervised learning via supervised learning with a non-convex penalty. *Journal of Machine Learning Research*, 14:1865–1889, 2013.

[39] Stephen Portnoy. Asymptotic behavior of m estimators of p regression parameters when p^2/n is large; ii. normal approximation. *Ann. Statist.*, 13(4):1403–1417, 12 1985.

[40] Garvesh Raskutti, Martin J. Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over l_q -balls. *IEEE Trans. Inf. Theor.*, 57(10):6976–6994, October 2011.

[41] Steffen Rendle. Factorization machines. pages 995–1000. IEEE Publishing, December 2010.

[42] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval*, SIGIR '11, pages 635–644. ACM, July 2011.

[43] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. pages 175–186. ACM Press, 1994.

[44] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 791–798, New York, NY, USA, 2007. ACM.

[45] Gerald Salton, editor. *Automatic Text Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1988.

[46] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John T. Riedl. Application of dimensionality reduction in recommender system – a case study. In *ACM WebKDD Workshop*, 2000.

[47] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings Of The 25Th Annual International Acm Sigir Conference On Research And Development In Information Retrieval*, pages 253–260, 2002.

[48] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings Of The 25Th Annual International Acm Sigir Conference On Research And Development In Information Retrieval*, pages 253–260, 2002.

[49] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating "word of mouth". pages 210–217. ACM Press, 1995.

[50] Xiaotong Shen. On the method of penalization. *Statistica Sinica*, 8:337–357, 1998.

[51] Xiaotong Shen and Hsin-Cheng Huang. Grouping pursuit through a regularization solution surface. *Journal of the American Statistical Association*, 105(490):727–739, 2010.

[52] Xiaotong Shen, Wei Pan, and Yunzhang Zhu. Likelihood-based selection and sharp parameter estimation. *Journal of American Statistical Association*, 107:223–232, 2012.

[53] Nathan Srebro and Tommi Jaakkola. Generalization error bounds for collaborative prediction with low-rank matrices. In *In Advances In Neural Information Processing Systems 17*, pages 5–27. MIT Press, 2005.

[54] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, Jun 2001.

[55] Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Math. Program.*, 117(1):387–423, July 2008.

[56] Naisyin Wang. Marginal nonparametric kernel regression accounting for within-subject correlation. *Biometrika*, 90(1):43–52, 2003.

[57] Wing Hung Wong and Xiaotong Shen. Probability inequalities for likelihood ratios and convergence rates of sieve mles. *Ann. Statist.*, 23(2):339–362, 04 1995.

[58] Chong Wu, Sunghoon Kwon, Xiaotong Shen, and Wei Pan. A new algorithm and theory for penalized regression-based clustering. *Journal of Machine Learning Research*, 17(188):1–25, 2016.

[59] Fan Yang and Xiaotong Shen. A personalized recommender system with correlation estimation. *In Preparation.*

[60] Gui-Bo Ye and Xiaohui Xie. Split bregman method for large scale fused lasso. *Computational Statistics and Data Analysis*, 55(4):1552–1569, 2011.

[61] Ian En-Hsu Yen, Nanyun Peng, Po-Wei Wang, and Shou-De Lin. On convergence rate of concave-convex procedure. *Proceedings of the NIPS 2012 Optimization Workshop*, 2012.

[62] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the netflix prize. In *Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management*, AAIM '08, pages 337–348, Berlin, Heidelberg, 2008. Springer-Verlag.

[63] Yunzhang Zhu, Xiaotong Shen, and Wei Pan. Structural pursuit over multiple undirected graphs. *Journal of the American Statistical Association*, 109(508):1683–1696, 2014.

[64] Yunzhang Zhu, Xiaotong Shen, and Changqing Ye. Personalized prediction and sparsity pursuit in latent factor models. *Journal of the American Statistical Association*, 111(513):241–252, 2016.

[65] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

[66] Hui Zou and Hao Helen Zhang. On the adaptive elastic-net with a diverging number of parameters. *Ann. Statist.*, 37(4):1733–1751, 08 2009.

Appendix A

Proofs

A.1 Technical details for section 3.2

Proof of Convexity of (3.5) in $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\Omega}$ separately:

(3.5) is the objective function with the L_1 penalty

$$-l_1(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\Omega}) = \frac{1}{2}\sum_i [\mathrm{tr}(\boldsymbol{\Omega}_i \boldsymbol{S}_i) - \mathrm{logdet}(\boldsymbol{\Omega}_i)] + \frac{\lambda_1}{2}\sum_{i<k} \|\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_k\|_1$$
$$+ \frac{\lambda_1}{2}\sum_{i<k} \|\boldsymbol{\beta}_i - \boldsymbol{\beta}_k\|_1 + \lambda_2 \sum_{k\leqslant l} \sum_{\{k,l\}\subseteq \bigcup_h I_h} |\omega_{T_i,kl} - \omega_{T_j,kl}|$$

(a) Convexity in $\boldsymbol{\alpha}$:

The part involving $\boldsymbol{\alpha}$ is

$$\frac{1}{2}\sum_i \mathrm{tr}(\boldsymbol{\Omega}_i \boldsymbol{S}_i) + \frac{\lambda_1}{2}\sum_{i<k} \|\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_k\|_1$$
$$= \frac{1}{2}\sum_i (\boldsymbol{r}_i - \boldsymbol{\alpha}_{I_i}^T \boldsymbol{x}_i - \boldsymbol{Y}_{I_i}\boldsymbol{\beta}_i)^T \boldsymbol{\Omega}_i (\boldsymbol{r}_i - \boldsymbol{\alpha}_{I_i}^T \boldsymbol{x}_i - \boldsymbol{Y}_{I_i}\boldsymbol{\beta}_i) + \frac{\lambda_1}{2}\sum_{i<k} \|\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_k\|_1$$

The first term is quadratic in $\boldsymbol{\alpha}$ and because $\boldsymbol{\Omega}_i$ is positive definite, thus this term is convex in $\boldsymbol{\alpha}$. The second term consists of L_1-norm of the differences of the $\boldsymbol{\alpha}$'s

which are basically absolute values, and is also convex in $\boldsymbol{\alpha}$. So their sum is also convex in $\boldsymbol{\alpha}$.

(b) Convexity in $\boldsymbol{\beta}$:

The part involving $\boldsymbol{\beta}$ is

$$\begin{aligned}&\frac{1}{2}\sum_i \mathrm{tr}(\boldsymbol{\Omega}_i \boldsymbol{S}_i) + \frac{\lambda_1}{2}\sum_{i<k}\|\boldsymbol{\beta}_i - \boldsymbol{\beta}_k\|_1 \\ =&\frac{1}{2}\sum_i (\boldsymbol{r}_i - \boldsymbol{\alpha}_{I_i}^T \boldsymbol{x}_i - \boldsymbol{Y}_{I_i}\boldsymbol{\beta}_i)^T \boldsymbol{\Omega}_i (\boldsymbol{r}_i - \boldsymbol{\alpha}_{I_i}^T \boldsymbol{x}_i - \boldsymbol{Y}_{I_i}\boldsymbol{\beta}_i) + \frac{\lambda_1}{2}\sum_{i<k}\|\boldsymbol{\beta}_i - \boldsymbol{\beta}_k\|_1\end{aligned}$$

The first term is quadratic in $\boldsymbol{\beta}$ and because $\boldsymbol{\Omega}_i$ is positive definite, thus this term is convex in $\boldsymbol{\beta}$. The second term consists of L_1-norm of the differences of the $\boldsymbol{\beta}$'s which are basically absolute values, and is also convex in $\boldsymbol{\beta}$. So their sum is also convex in $\boldsymbol{\beta}$.

(c) Convexity in $\boldsymbol{\Omega}$: The part involving $\boldsymbol{\Omega}$ is

$$\frac{1}{2}\sum_i [\mathrm{tr}(\boldsymbol{\Omega}_i \boldsymbol{S}_i) - \mathrm{logdet}(\boldsymbol{\Omega}_i)] + \lambda_2 \sum_{k\leqslant l}\sum_{\{k,l\}\subseteq \bigcup_h I_h} |\omega_{T_i,kl} - \omega_{T_j,kl}|$$

Each summand in the first term only involves $\boldsymbol{\Omega}_i$. It is the negative log normal likelihood and is convex in $\boldsymbol{\Omega}_i$. Thus their sum is a convex function of $\boldsymbol{\Omega}$. The second term consists of absolute values of the differences of $\boldsymbol{\Omega}$ entries, and is also convex in $\boldsymbol{\Omega}$. So their sum is also convex in $\boldsymbol{\Omega}$.

Proof of Proposition 3.2.1:

First we show that we can find a $c > 0$ to make $S_1^{l_1}$ in (3.11) convex. Let function $M(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Omega}) = \sum_{i=1}^n \left[\frac{1}{2}\mathrm{logdet}(\boldsymbol{\Omega}_i) - \frac{(\boldsymbol{r}_i - \boldsymbol{\mu}_i)^T \boldsymbol{\Omega}_i (\boldsymbol{r}_i - \boldsymbol{\mu}_i)}{2}\right]$ which is the log likelihood part of the penalized log likelihood function, where $\boldsymbol{\mu}_i = \boldsymbol{\alpha}_{I_i}^T \boldsymbol{x}_i + \boldsymbol{Y}_{I_i}\boldsymbol{\beta}_i$. Note that $M()$ is convex in $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Omega}$ separately. Then the Hessian matrix of $M()$ (think of $\boldsymbol{\Omega}$ as a long vector also) w.r.t. $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Omega}$ has three positive definite blocks on the diagonal. For the off-diagonal of the Hessian, it only comes from the term $(\boldsymbol{r}_i - \boldsymbol{\mu}_i)^T \boldsymbol{\Omega}_i (\boldsymbol{r}_i - \boldsymbol{\mu}_i)$ as $\mathrm{logdet}(\boldsymbol{\Omega}_i)$ only involves $\boldsymbol{\Omega}$. Since we assumed the parameter space satisfies $\|\boldsymbol{\alpha}\|_\infty \leqslant L$, $\|\boldsymbol{\beta}\|_\infty \leqslant L$

and $\|\boldsymbol{\Omega}\|_\infty \leqslant L$, it can be seen that the off-diagonal of the Hessian is also bounded. Thus we can find $c > 0$ and make $M(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Omega}) + c(\|\boldsymbol{\alpha}\|_F^2 + \|\boldsymbol{\beta}\|_F^2 + \sum_i \|\boldsymbol{\Omega}_i\|_F^2)$ have a Hessian with the diagonal blocks dominating, which makes the Hessian of $M(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Omega}) + c(\|\boldsymbol{\alpha}\|_F^2 + \|\boldsymbol{\beta}\|_F^2 + \sum_i \|\boldsymbol{\Omega}_i\|_F^2)$ positive definite and thus convex.

By Theorem 3.3 of [61], the difference of convex algorithm converges to a stationary point of the TLP problem (3.7). For the objective function inside the d.o.c. algorithm, the ADMM method applied converges to an optimal solution by Theorem 4.1 of [54]. Thus the overall algorithm converges to a stationary point of (3.7). The same argument works for the L_1 problem. This completes the proof.

A.2 Technical details for section 3.3

Proof of Lemma 3.3.2:

Since $\boldsymbol{\theta}_{u,A} = (\boldsymbol{\eta}_u, \mathrm{vec}(\boldsymbol{\Omega}_{T_u}))$, we have

$$\|\boldsymbol{\theta}_{u,A} - \tilde{\boldsymbol{\theta}}_{u,A}\|_2^2 = \|\boldsymbol{\eta}_u - \tilde{\boldsymbol{\eta}}_u\|_2^2 + \|\boldsymbol{\Omega}_{T_u} - \tilde{\boldsymbol{\Omega}}_{T_u}\|_F^2,$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix. For the first part, $\boldsymbol{\eta}_u = \boldsymbol{\alpha}^T \boldsymbol{x}_u + \boldsymbol{Y}\boldsymbol{\beta}_u$. Thus,

$$\begin{aligned}
\|\boldsymbol{\eta}_u - \tilde{\boldsymbol{\eta}}_u\|_2^2 &= \|(\boldsymbol{\alpha} - \tilde{\boldsymbol{\alpha}})^T \boldsymbol{x}_u + \boldsymbol{Y}(\boldsymbol{\beta}_u - \tilde{\boldsymbol{\beta}}_u)\|_2^2 \\
&\leqslant 2(\|(\boldsymbol{\alpha} - \tilde{\boldsymbol{\alpha}})^T \boldsymbol{x}_u\|_2^2 + \|\boldsymbol{Y}(\boldsymbol{\beta}_u - \tilde{\boldsymbol{\beta}}_u)\|_2^2) \\
&\leqslant 4\|\boldsymbol{x}_u\|_\infty^2 \|(\boldsymbol{\alpha} - \tilde{\boldsymbol{\alpha}})^T\|_F^2 + 4m\|Y\|_{\max}^2 \|\boldsymbol{\beta}_u - \tilde{\boldsymbol{\beta}}_u\|_2^2,
\end{aligned}$$

where $\|\cdot\|_{\max}$ is the max norm of a matrix with $\|A\|_{\max} = \max_{i,j}(|a_{ij}|)$. Let $X = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$ be the user feature matrix. The last inequality above is obtained from the fact that

$$\begin{aligned}
\|(\boldsymbol{\alpha}-\tilde{\boldsymbol{\alpha}})^T\boldsymbol{x}_u\|_2^2 &= \sum_{i=1}^{m}(\sum_{j=1}^{K_1}(\alpha_{ji}-\tilde{\alpha}_{ji})x_{ju})^2 \\
&\leqslant 2\sum_{i=1}^{m}\sum_{j=1}^{K_1}(\alpha_{ji}-\tilde{\alpha}_{ji})^2x_{ju}^2 \\
&\leqslant 2\|\boldsymbol{x}_u\|_\infty^2\|(\boldsymbol{\alpha}-\tilde{\boldsymbol{\alpha}})^T\|_F^2,
\end{aligned}$$

and

$$\begin{aligned}
\|\boldsymbol{Y}(\boldsymbol{\beta}_u-\tilde{\boldsymbol{\beta}}_u)\|_2^2 &= \sum_{i=1}^{m}(\sum_{j=1}^{K_2}y_{ij}(\beta_{uj}-\tilde{\beta}_{uj}))^2 \\
&\leqslant 2\sum_{i=1}^{m}\sum_{j=1}^{K_2}y_{ij}^2(\beta_{uj}-\tilde{\beta}_{uj})^2 \\
&\leqslant 2\max_j(\sum_{i=1}^{m}y_{ij}^2)\|\boldsymbol{\beta}_u-\tilde{\boldsymbol{\beta}}_u\|_2^2 \\
&\leqslant 2m\|Y\|_\infty^2\|\boldsymbol{\beta}_u-\tilde{\boldsymbol{\beta}}_u\|_2^2.
\end{aligned}$$

Then using Assumption 3.3.1, there is a constant $C_1 \geqslant 0$, such that

$$
\begin{aligned}
h(\boldsymbol{\xi},\tilde{\boldsymbol{\xi}}) &= [\frac{1}{n}\sum_{u=1}^{n} h^2_{\Theta_A}(\boldsymbol{\theta}_{u,A},\tilde{\boldsymbol{\theta}}_{u,A})]^{1/2} \\
&= [\frac{1}{n}\sum_{u=1}^{n}(\int (f^{1/2}(\boldsymbol{r}_u|\boldsymbol{\theta}_{u,A}) - f^{1/2}(\boldsymbol{r}_u|\tilde{\boldsymbol{\theta}}_{u,A}))^2 d\nu(\boldsymbol{r}_u))]^{1/2} \\
&\leqslant [\frac{1}{n}\sum_{u=1}^{n}\int G^2(\boldsymbol{r}_u)\|\boldsymbol{\theta}_{u,A} - \tilde{\boldsymbol{\theta}}_{u,A}\|_2^2 d\nu(\boldsymbol{r}_u)]^{1/2} \\
&\leqslant \bar{G}[\frac{1}{n}\sum_{u=1}^{n}\|\boldsymbol{\theta}_{u,A} - \tilde{\boldsymbol{\theta}}_{u,A}\|_2^2]^{1/2} \\
&\leqslant \bar{G}[\frac{1}{n}\sum_{u=1}^{n}(4\|\boldsymbol{x}_u\|_\infty^2\|(\boldsymbol{\alpha}-\tilde{\boldsymbol{\alpha}})^T\|_F^2 + 4m\|Y\|_\infty^2\|\boldsymbol{\beta}_u - \tilde{\boldsymbol{\beta}}_u\|_2^2 + \|\boldsymbol{\Omega}_{T_u} - \tilde{\boldsymbol{\Omega}}_{T_u}\|_F^2)]^{1/2} \\
&\leqslant \bar{G}[\frac{1}{n}(4n\|X\|_\infty^2\|(\boldsymbol{\alpha}-\tilde{\boldsymbol{\alpha}})^T\|_F^2 + 4m\|Y\|_\infty^2\|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\|_F^2 + \|\boldsymbol{\Omega}_T - \tilde{\boldsymbol{\Omega}}_T\|_F^2)]^{1/2} \\
&\leqslant \bar{G}C_1\sqrt{\frac{\max(n,m)}{n}}\|\boldsymbol{\xi} - \tilde{\boldsymbol{\xi}}\|_2.
\end{aligned}
$$

Let $d_0 = \bar{G}C_1$, the result then follows. This completes the proof.

Proof of Theorem 3.3.3:

First let $g(\delta) = d_0\sqrt{\frac{\max(n,m)}{n}}\delta$, which is a strictly increasing continous function. Then it satisfies the condition of Lemma 2.1 of Ossiander (1987) [37]. This is because based on Lemma 3.3.2,

$$
\begin{aligned}
&\left[\frac{1}{n}\sum_{u=1}^{n} E(\sup_{\tilde{\boldsymbol{\xi}}\in B_\delta(\boldsymbol{\xi})}|f_u^{1/2}(\boldsymbol{r},\boldsymbol{\xi}) - f_u^{1/2}(\boldsymbol{r},\tilde{\boldsymbol{\xi}})|^2)\right]^{1/2} \\
= &\left[\frac{1}{n}\sum_{u=1}^{n}\int \sup_{\tilde{\boldsymbol{\xi}}\in B_\delta(\boldsymbol{\xi})}|f^{1/2}(\boldsymbol{r}_u,\boldsymbol{\xi}) - f^{1/2}(\boldsymbol{r}_u,\tilde{\boldsymbol{\xi}})|^2 d\nu(\boldsymbol{r}_u)\right]^{1/2} \\
\leqslant &\left[\bar{G}^2C_1^2\frac{\max(n,m)}{n}\sup_{\tilde{\boldsymbol{\xi}}\in B_\delta(\boldsymbol{\xi})}\|\boldsymbol{\xi}-\tilde{\boldsymbol{\xi}}\|_2^2\right]^{1/2} \\
\leqslant &\ g(\delta).
\end{aligned}
$$

Therefore we have for $u > 0$,

$$H^B(u, \mathcal{S}(k), \rho) \leqslant H(g^{-1}(u/2), \mathcal{S}(k), \rho),$$

where H^B is the metric entropy of $\mathcal{S}(k)$ with bracketing of $f^{1/2}$, H is the ordinary metric entropy of $\mathcal{S}(k)$, and ρ is the L_2-norm.

Next we provide an upper bound for $H(g^{-1}(u/2), \mathcal{S}(k), \rho)$. Since $\|\boldsymbol{\xi}\|_\infty \leqslant \phi$, $N \leqslant n$, $M \leqslant m$, for $x > 0$, we have

$$\begin{aligned}
& H(x, \mathcal{S}(k), \rho) \\
\leqslant \quad & \log\Big\{\max\Big[\Big(\frac{\phi\sqrt{((N+M)K + \tilde{N}m\tilde{K})}}{x}\Big)^{(N+M)K+\tilde{N}m\tilde{K}}, 1\Big]\Big\} \\
\leqslant \quad & \max\Big[((N+M)K + \tilde{N}m\tilde{K})\log\Big(\frac{\phi\sqrt{((N+M)K + \tilde{N}m\tilde{K})}}{x}\Big), 0\Big] \\
\leqslant \quad & \max\Big[((N+M+\tilde{N}m)\bar{K})\log\Big(\frac{\phi\sqrt{((N+M)K + \tilde{N}m\tilde{K})}}{x}\Big), 0\Big] \\
\leqslant \quad & \max\Big[((N+M+\tilde{N}m)\bar{K})\log\Big(\frac{\phi\sqrt{((N+M+\tilde{N}m)\bar{K})}}{x}\Big), 0\Big].
\end{aligned}$$

Since $g^{-1}(u/2) = \dfrac{\sqrt{n}}{2d_0\sqrt{\max(n,m)}}u$, we have

$$\begin{aligned}
0 \quad \leqslant \quad & H^B(u, \mathcal{S}(k), \rho) \\
\leqslant \quad & H(g^{-1}(u/2), \mathcal{S}(k), \rho) \\
\leqslant \quad & \max\Big[((N+M+\tilde{N}m)\bar{K})\log\Big(\frac{2d_0\phi\sqrt{\max(n,m)(N+M+\tilde{N}m)\bar{K}}}{\sqrt{n}u}\Big), 0\Big] \\
= \quad & \max\Big[((N+M+\tilde{N}m)\bar{K})\log\Big(\frac{C\sqrt{\max(n,m)(N+M+\tilde{N}m)\bar{K}}}{\sqrt{n}u}\Big), 0\Big],
\end{aligned}$$

where $C = 2d_0\phi$.

We now find the convergence rate $\epsilon_{|\Omega|}$, which is the smallest ϵ that satisfies the conditions of Theorem 1 of Shen (1998) [50]. That is

$$\sup_{k\geqslant k_0}\psi_1(\epsilon,k)\leqslant c_2|\Omega|^{1/2}$$

for a constant k_0, where $\psi_1(\epsilon,k)=\int_x^{x^{1/2}}\{H^B(u,\mathcal{F}(k))\}^{1/2}du/x$ with $x=c_1\epsilon^2+\lambda_{|\Omega|}(k-k_0)$, and $\mathcal{F}(k)=\{f(\boldsymbol{r},\boldsymbol{\xi})^{1/2}:\boldsymbol{\xi}\in\mathcal{S}(k)\}$.

Note that when $x\geqslant 1$, $\psi_1\leqslant 0\leqslant c_2|\Omega|^{1/2}$. So we only look at the case when $0<x<1$. Notice that for sufficiently large L, and with $x\leqslant u\leqslant x^{1/2}$, $\frac{C\sqrt{\max(n,m)(N+M+\tilde{N}m)\bar{K}}}{\sqrt{n}u}>1$. So

$$\begin{aligned}&\max\Big[((N+M+\tilde{N}m)\bar{K})\log\Big(\frac{C\sqrt{\max(n,m)(N+M+\tilde{N}m)\bar{K}}}{\sqrt{n}u}\Big),0\Big]\\ =\quad&((N+M+\tilde{N}m)\bar{K})\log\Big(\frac{C\sqrt{\max(n,m)(N+M+\tilde{N}m)\bar{K}}}{\sqrt{n}u}\Big).\end{aligned}$$

Thus

$$\begin{aligned}\psi_1(\epsilon,k)=&\int_x^{x^{1/2}}\{H^B(u,\mathcal{F}(k))\}^{1/2}du/x\\ \leqslant&\frac{((N+M+\tilde{N}m)\bar{K})^{1/2}}{x}\int_x^{x^{1/2}}\Big\{\log\Big(\frac{C\sqrt{\max(n,m)(N+M+\tilde{N}m)\bar{K}}}{\sqrt{n}}\Big)-\log u\Big\}^{1/2}du\\ \leqslant&((N+M+\tilde{N}m)\bar{K})^{1/2}(\frac{1}{\sqrt{x}}-1)\Big\{\log\Big(\frac{C\sqrt{\max(n,m)(N+M+\tilde{N}m)\bar{K}}}{\sqrt{n}}\Big)-\log x\Big\}^{1/2}.\end{aligned}$$

Notice that $k\sim O(\sqrt{(N(N-1)+M(M-1))K+\tilde{N}m\tilde{K}+\tilde{N}(\tilde{N}-1)m\tilde{K}})$ and $\lambda_{|\Omega|}<\frac{1}{2k}\epsilon^2_{|\Omega|}$, we have $\lambda_{|\Omega|}=o(\epsilon^2_{|\Omega|})$. Also note $M\leqslant m$. Therefore, we solve

$$\begin{aligned}\sup_{k\geqslant k_0}\psi_1(\epsilon,k)&=\psi_1(\epsilon,k_0)\\ &\sim((N+\tilde{N}m)\bar{K})^{1/2}\frac{1}{\epsilon_{|\Omega|}}\Big\{\log\Big(\frac{\sqrt{\max(n,m)(N+\tilde{N}m)\bar{K}}}{\epsilon^2_{|\Omega|}\sqrt{n}}\Big)\Big\}^{1/2}\\ &=c_2|\Omega|^{1/2}.\end{aligned}$$

Then the smallest rate $\epsilon_{|\Omega|}$ satisfies

$$\frac{1}{\epsilon_{|\Omega|}}\Big\{\log\Big(\frac{\sqrt{\max(n,m)(N+\tilde{N}m)\bar{K}}}{\epsilon_{|\Omega|}^2\sqrt{n}}\Big)\Big\}^{1/2} \sim \frac{|\Omega|^{1/2}}{((N+\tilde{N}m)\bar{K})^{1/2}}.$$

So we have

$$\epsilon_{|\Omega|} \sim \sqrt{\frac{(N+\tilde{N}m)\bar{K}}{|\Omega|}}\Big(\log\frac{|\Omega|\sqrt{\max(n,m)}}{\sqrt{n(N+\tilde{N}m)\bar{K}}}\Big)^{1/2}.$$

For $\epsilon_{|\Omega|}$ and $\lambda_{|\Omega|}$, the conditions of Corollary 1 of Shen (1998) [50] are now satisfied. The result then follows. This completes the proof. □

Explanation of Remark 3.3.5:

With the Hellinger metric in (3.36) divided by $\sqrt{m}$, result in Lemma 3.3.2 becomes

$$h(\boldsymbol{\xi},\tilde{\boldsymbol{\xi}}) \leqslant d_0\|\boldsymbol{\xi}-\tilde{\boldsymbol{\xi}}\|_2\sqrt{\frac{\max(m,n)}{nm}}.$$

Thus the $g(\delta)$ in the proof of Theorem 3.3.3 becomes

$$g(\delta) = d_0\sqrt{\frac{\max(n,m)}{nm}}\delta.$$

Then we have

$$H^B(u,\mathcal{S}(k),\rho) \leqslant \max\Big[((n+m+\tilde{N}m)\bar{K})\log\Big(\frac{C\sqrt{\max(n,m)(n+m+\tilde{N}m)\bar{K}}}{\sqrt{nm}u}\Big),0\Big].$$

Solve for

$$\sup_{k\geqslant k_0}\psi_1(\epsilon,k) \leqslant c_2|\Omega|^{1/2}.$$

With similar arguments, we can get

$$\tilde{\epsilon}_{|\Omega|} \sim \sqrt{\frac{(N+\tilde{N}m)\bar{K}}{|\Omega|}}\Big(\log\frac{|\Omega|\sqrt{\max(n,m)}}{\sqrt{nm(N+\tilde{N}m)\bar{K}}}\Big)^{1/2}.$$

Proof of Theorem 3.3.6:

This is a direct result implied from Theorem 1 of Shen (1998) [50].□

Proof of Corollary 3.3.7:

For simplicity, write $h_{\Theta_A}(\boldsymbol{\theta}_{u,A}, \tilde{\boldsymbol{\theta}}_{u,A})$ as $h_\Theta(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}})$, and $f(\boldsymbol{r}_u|\boldsymbol{\theta}_u)$ as $f(\boldsymbol{r}|\boldsymbol{\theta})$. We now lower-bound $h(\boldsymbol{\xi}, \tilde{\boldsymbol{\xi}})$ by a function of $\|\boldsymbol{\theta}_u - \tilde{\boldsymbol{\theta}}_u\|_2$.

$$\begin{aligned} h^2_\Theta(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) &= \int (f^{1/2}(\boldsymbol{r}|\boldsymbol{\theta}) - f^{1/2}(\boldsymbol{r}|\tilde{\boldsymbol{\theta}}))^2 d\nu(\boldsymbol{r}) \\ &= \Big[\int_{\{f(\boldsymbol{r}|\boldsymbol{\theta})>f(\boldsymbol{r}|\tilde{\boldsymbol{\theta}})\}} + \int_{\{f(\boldsymbol{r}|\tilde{\boldsymbol{\theta}})>f(\boldsymbol{r}|\boldsymbol{\theta})\}}\Big](f^{1/2}(\boldsymbol{r}|\boldsymbol{\theta}) - f^{1/2}(\boldsymbol{r}|\tilde{\boldsymbol{\theta}}))^2 d\nu(\boldsymbol{r}) \\ &:= I_1 + I_2, \end{aligned}$$

where

$$\begin{aligned} I_1 &= \int_{\{f(\boldsymbol{r}|\boldsymbol{\theta})>f(\boldsymbol{r}|\tilde{\boldsymbol{\theta}})\}} (f^{1/2}(\boldsymbol{r}|\boldsymbol{\theta}) - f^{1/2}(\boldsymbol{r}|\tilde{\boldsymbol{\theta}}))^2 d\nu(\boldsymbol{r}), \\ I_2 &= \int_{\{f(\boldsymbol{r}|\tilde{\boldsymbol{\theta}})>f(\boldsymbol{r}|\boldsymbol{\theta})\}} (f^{1/2}(\boldsymbol{r}|\boldsymbol{\theta}) - f^{1/2}(\boldsymbol{r}|\tilde{\boldsymbol{\theta}}))^2 d\nu(\boldsymbol{r}). \end{aligned}$$

For I_1, since $f(\boldsymbol{r}|\boldsymbol{\theta}) > f(\boldsymbol{r}|\tilde{\boldsymbol{\theta}})$, we have $Z := (\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta})^T T(\boldsymbol{r}) - (A(\tilde{\boldsymbol{\theta}}) - A(\boldsymbol{\theta})) \leqslant 0$. Since $\|\boldsymbol{\xi}\|_\infty \leqslant L$, we have $\|\boldsymbol{\theta}\|_\infty$ also bounded and$\boldsymbol{\theta}$ is in a closed set of R^m. Hence $A'(\boldsymbol{\theta}) = E_{\boldsymbol{\theta}}[T(\boldsymbol{r})]$ is also bounded in l_∞-norm. Let $L_A = \sup_{\boldsymbol{\theta}} \big(\|E_{\boldsymbol{\theta}} T(\boldsymbol{r})\|_\infty\big)$, then

$$|A(\tilde{\boldsymbol{\theta}}) - A(\boldsymbol{\theta})| \leqslant L_A \|\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_1.$$

Hence $-Z = |Z| \geqslant |(\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta})^T T(\boldsymbol{r})| - L_A \|\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_1$. If $\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}$ and $T(\boldsymbol{r})$ have the same sign elementwise, then

$$|(\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta})^T T(\boldsymbol{r})| \geqslant \|T(\boldsymbol{r})\|_{\min} \|\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_1,$$

where $\|\boldsymbol{x}\|_{\min} = \min_i |x_i|$. So $-Z = |Z| \geqslant (\|T(\boldsymbol{r})\|_{\min} - L_A)\|\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_1$. Thus

$$1 - \exp(-\frac{1}{2}|Z|) \geqslant \max\{1 - \exp[\frac{1}{2}(L_A - \|T(\boldsymbol{r})\|_{\min})\|\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_1], 0\}.$$

When vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ have the same signs elementwise, we write $\text{sign}(\boldsymbol{a}) = \text{sign}(\boldsymbol{b})$. Let

$S_1 = \{\boldsymbol{r} : f(\boldsymbol{r}|\boldsymbol{\theta}) > f(\boldsymbol{r}|\tilde{\boldsymbol{\theta}}) \text{ and } \text{sign}(T(\boldsymbol{r})) = \text{sign}(\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta})\}$, we have

$$\begin{aligned} I_1 &= \int_{\{f(\boldsymbol{r}|\boldsymbol{\theta})>f(\boldsymbol{r}|\tilde{\boldsymbol{\theta}})\}} f(\boldsymbol{r}|\boldsymbol{\theta})[1-\exp(-\frac{1}{2}|Z|)]^2 d\nu(\boldsymbol{r}) \\ &\geqslant \int_{S_1} f(\boldsymbol{r}|\boldsymbol{\theta})\{\max\{1-\exp[\frac{1}{2}(L_A - \|T(\boldsymbol{r})\|_{\min})\|\tilde{\boldsymbol{\theta}}-\boldsymbol{\theta}\|_1], 0\}\}^2 d\nu(\boldsymbol{r}). \end{aligned}$$

Similarly, let $S_2 = \{\boldsymbol{r} : f(\boldsymbol{r}|\tilde{\boldsymbol{\theta}}) > f(\boldsymbol{r}|\boldsymbol{\theta}) \text{ and } \text{sign}(T(\boldsymbol{r})) = \text{sign}(\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta})\}$, we have

$$\begin{aligned} I_2 &= \int_{\{f(\boldsymbol{r}|\tilde{\boldsymbol{\theta}})>f(\boldsymbol{r}|\boldsymbol{\theta})\}} f(\boldsymbol{r}|\tilde{\boldsymbol{\theta}})[1-\exp(-\frac{1}{2}Z)]^2 d\nu(\boldsymbol{r}) \\ &\geqslant \int_{S_2} f(\boldsymbol{r}|\tilde{\boldsymbol{\theta}})\{\max\{1-\exp[\frac{1}{2}(L_A - \|T(\boldsymbol{r})\|_{\min})\|\tilde{\boldsymbol{\theta}}-\boldsymbol{\theta}\|_1], 0\}\}^2 d\nu(\boldsymbol{r}) \\ &\geqslant \int_{S_2} f(\boldsymbol{r}|\boldsymbol{\theta})\{\max\{1-\exp[\frac{1}{2}(L_A - \|T(\boldsymbol{r})\|_{\min})\|\tilde{\boldsymbol{\theta}}-\boldsymbol{\theta}\|_1], 0\}\}^2 d\nu(\boldsymbol{r}). \end{aligned}$$

Notice $1-\exp[\frac{1}{2}(L_A - \|T(\boldsymbol{r})\|_{\min})\|\tilde{\boldsymbol{\theta}}-\boldsymbol{\theta}\|_1] \geqslant 0$ if and only if $\|T(\boldsymbol{r})\|_{\min} \geqslant L_A$. Let $S = \{\boldsymbol{r} : \|T(\boldsymbol{r})\|_{\min} \geqslant L_A \text{ and } \text{sign}(T(\boldsymbol{r})) = \text{sign}(\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta})\}$. Hence,

$$\begin{aligned} h^2_{\Theta}(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) &= I_1 + I_2 \\ &\geqslant \int_S f(\boldsymbol{r}|\boldsymbol{\theta})\{1-\exp[\frac{1}{2}(L_A - \|T(\boldsymbol{r})\|_{\min})\|\tilde{\boldsymbol{\theta}}-\boldsymbol{\theta}\|_1]\}^2 d\nu(\boldsymbol{r}), \end{aligned}$$

which is a non-decreasing function of $\|\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_1$.

Hence for each $\boldsymbol{\theta}_{u0}$, and given the $\epsilon_{|\Omega|}$ in Theorem 3.3.3, there exists a $\delta_{|\Omega|}(\boldsymbol{\theta}_{u0}) > 0$ such that $\|\tilde{\boldsymbol{\theta}}_u - \boldsymbol{\theta}_{u0}\|_1 > \delta_{|\Omega|}(\boldsymbol{\theta}_{u0})$ implies $h_{\Theta}(\tilde{\boldsymbol{\theta}}_u, \boldsymbol{\theta}_{u0}) \geqslant \epsilon_{|\Omega|}$. Take $\delta_{|\Omega|} = \max_u \delta_{|\Omega|}(\boldsymbol{\theta}_{u0})$, then $\|\tilde{\boldsymbol{\theta}}_u - \boldsymbol{\theta}_{u0}\|_1 > \delta_{|\Omega|}$ for each u implies $h_{\Theta}(\tilde{\boldsymbol{\theta}}_u, \boldsymbol{\theta}_{u0}) \geqslant \epsilon_{|\Omega|}$ for all u. By definition of $h(\boldsymbol{\xi}, \tilde{\boldsymbol{\xi}})$, we have $h^2(\boldsymbol{\xi}_0, \tilde{\boldsymbol{\xi}}) = \frac{1}{n}\sum_{u=1}^n h^2_{\Theta}(\tilde{\boldsymbol{\theta}}_u, \boldsymbol{\theta}_{u0}) \geqslant \epsilon^2_{|\Omega|}$ and so $h(\boldsymbol{\xi}_0, \tilde{\boldsymbol{\xi}}) \geqslant \epsilon_{|\Omega|}$. The result then follows from Theorem 3.3.6. □

Proof of Corollary 3.3.8:

Define $\Phi = \{u : \|\tilde{\boldsymbol{\theta}}_u - \boldsymbol{\theta}_{u0}\|_1 > \delta_{|\Omega|}\}$. Then the size of Φ satisfies $|\Phi| \geqslant \phi n$. From

the proof of Corollary 3.3.7 it can be seen, for $u \in \Phi$, $h_\Theta(\tilde{\boldsymbol{\theta}}_u, \boldsymbol{\theta}_{u0}) \geqslant \frac{1}{\sqrt{\phi}}\epsilon_{|\Omega|}$. Then

$$h(\boldsymbol{\xi}_0, \tilde{\boldsymbol{\xi}}) = \sqrt{\frac{1}{n}\sum_{u=1}^{n} h_\Theta^2(\tilde{\boldsymbol{\theta}}_u, \boldsymbol{\theta}_{u0})} \geqslant \sqrt{\frac{1}{n}\phi n \frac{\epsilon_{|\Omega|}^2}{\phi}} \geqslant \epsilon_{|\Omega|}.$$

This completes the proof. □

A.3 Technical details for section 3.4

Proof of Theorem 3.4.1:

Proof of Statement **(1):**

Let $c_n = \sqrt{p_n/n}$. For $\|\boldsymbol{u}\| = A$, since $p(0) = 0$,

$$\begin{aligned} &L_n(\boldsymbol{\beta}_{n0} + c_n\boldsymbol{u}) - L_n(\boldsymbol{\beta}_{n0}) \\ \leqslant &S_n(\boldsymbol{\beta}_{n0} + c_n\boldsymbol{u}) - S_n(\boldsymbol{\beta}_{n0}) - n \sum_{j<k, \beta_{nj0} \neq \beta_{nk0}} \{p_{\lambda_n,\tau_n}(|\beta_{nj0} - \beta_{nk0} + c_n(u_j - u_k)|) \\ &- p_{\lambda_n,\tau_n}(|\beta_{nj0} - \beta_{nk0}|)\}. \end{aligned} \tag{5.1}$$

For the third term, because of condition (iii'), $c_n(u_j - u_k) = o(\tau_n)$. Since $\liminf_{n\to\infty} \frac{1}{\tau_n}\min(|\beta_{nj0} - \beta_{nk0}| : \beta_{nj0} \neq \beta_{nk0}) > 1$, when n is large enough, $|\beta_{nj0} - \beta_{nk0}| > \tau_n$ and $|\beta_{nj0} - \beta_{nk0} + c_n(u_j - u_k)| > \tau_n$. Also note for TLP, if $|x| > \tau$, $p_{\lambda,\tau}(|x|) = \lambda$. Thus when n is large enough the third term in (5.1) is 0.

For the first two terms of (5.1), since S_n is quadratic in $\boldsymbol{\beta}_n$, we have

$$\begin{aligned} &S_n(\boldsymbol{\beta}_{n0} + c_n\boldsymbol{u}) - S_n(\boldsymbol{\beta}_{n0}) \\ =&c_n(\frac{\partial S_n(\boldsymbol{\beta}_{n0})}{\partial \boldsymbol{\beta_n}})^T\boldsymbol{u} + \frac{1}{2}\boldsymbol{u}^T\frac{\partial^2 S_n(\boldsymbol{\beta}_{n0})}{\partial\boldsymbol{\beta}_n\partial\boldsymbol{\beta}_n^T}\boldsymbol{u}c_n^2 \end{aligned} \tag{5.2}$$

The gradient and Hessian of S_n have the following forms:

$$\frac{\partial S_n(\boldsymbol{\beta}_{n0})}{\partial\boldsymbol{\beta}_n} = \boldsymbol{X}^T\boldsymbol{\Omega}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}_{n0}), \ \ \frac{\partial^2 S_n(\boldsymbol{\beta}_{n0})}{\partial\boldsymbol{\beta}_n\partial\boldsymbol{\beta}_n^T} = -\boldsymbol{X}^T\boldsymbol{\Omega}\boldsymbol{X}. \tag{5.3}$$

Since the eigenvalue of $\frac{\boldsymbol{X}^T\boldsymbol{\Omega}\boldsymbol{X}}{n}$ is assumed to have a uniform upper bound, we have $\frac{1}{\sqrt{n}}\frac{\partial S_n(\boldsymbol{\beta}_{n0})}{\partial \beta_{nj}} = O_p(1)$. Thus $\|\frac{\partial S_n(\boldsymbol{\beta}_{n0})}{\partial \boldsymbol{\beta}_n}\| = O_p(\sqrt{np_n})$.

So applying Cauchy-Schwarz inequality, we get

$$|c_n\boldsymbol{u}^T(\frac{\partial S_n(\boldsymbol{\beta}_{n0})}{\partial \boldsymbol{\beta}_n})| \leqslant c_n\|\boldsymbol{u}\|\|\frac{\partial S_n(\boldsymbol{\beta}_{n0})}{\partial \boldsymbol{\beta}_n}\| = O_p(p_n)\|\boldsymbol{u}\|. \tag{5.4}$$

$$\frac{1}{2}\boldsymbol{u}^T\frac{\partial^2 S(\boldsymbol{\beta}_{n0})}{\partial \boldsymbol{\beta}_n \partial \boldsymbol{\beta}_n^T}\boldsymbol{u}c_n^2 = -\frac{1}{2}p_n\boldsymbol{u}^T\frac{\boldsymbol{X}^T\boldsymbol{\Omega}\boldsymbol{X}}{n}\boldsymbol{u}. \tag{5.5}$$

Because $\lambda_{\min}(\frac{\boldsymbol{X}^T\boldsymbol{\Omega}\boldsymbol{X}}{n}) > h_1$, $\boldsymbol{u}^T\frac{\boldsymbol{X}^T\boldsymbol{\Omega}\boldsymbol{X}}{n}\boldsymbol{u} \geqslant h_1\|\boldsymbol{u}\|_2^2$. So when $\|\boldsymbol{u}\|$ is large enough, the second term of (5.2) which is negative dominates the first term. Thus, given any $\epsilon > 0$, there exist A and N large enough, such that when $n > N$,

$$P(\sup_{\|\boldsymbol{u}\|=A} L_n(\boldsymbol{\beta}_{n0} + c_n\boldsymbol{u}) < L_n(\boldsymbol{\beta}_{n0})) \geqslant 1 - \epsilon.$$

That means there is a local maximizer in the ball of $\{\boldsymbol{\beta}_{n0} + c_n\boldsymbol{u} : \|\boldsymbol{u}\| \leqslant A\}$ with probability at least $1 - \epsilon$. Hence there exists a local maximizer $\hat{\boldsymbol{\beta}}_n$ of $L_n(\boldsymbol{\beta}_n)$ and $\|\hat{\boldsymbol{\beta}}_n - \boldsymbol{\beta}_{n0}\| = O_p(\sqrt{p_n/n})$. □

To prove Statement **(2)**, first we prove the following claim.

Claim 5.0.1. *Suppose $\|\boldsymbol{\beta}_n - \boldsymbol{\beta}_{n0}\| = O_p(\sqrt{p_n/n})$ and λ_n, τ_n satisfy the same conditions of theorem 3.4.1, i.e.*

(i') $\liminf_{n\to\infty} \frac{1}{\tau_n}\min(|\beta_{nj0} - \beta_{nk0}| : \beta_{nj0} \neq \beta_{nk0}) > 1$

(ii') $\frac{p_n}{n} \to 0$

(iii') $\frac{\tau_n}{\sqrt{p_n/n}} \to \infty$

(iv') $\sqrt{\frac{n}{p_n}}\frac{\lambda_n}{\tau_n} \to \infty$

When $\frac{\partial L_n(\boldsymbol{\beta}_n)}{\partial \beta_{nj}}$ exists at β_{nj}, with probability tending to 1, the sign of $\frac{\partial L_n(\boldsymbol{\beta}_n)}{\partial \beta_{nj}}$ is determined by $-\sum_{k\neq j, \beta_{nk0}=\beta_{nj0}} \operatorname{sign}(\beta_{nj} - \beta_{nk})$.

To prove the claim, note when differentiable,

$$\frac{\partial L_n(\boldsymbol{\beta}_n)}{\partial \beta_{nj}} = \frac{\partial S_n(\boldsymbol{\beta}_n)}{\partial \beta_{nj}} - n\sum_{k\neq j} p'_{\lambda_n,\tau_n}(|\beta_{nj}-\beta_{nk}|)\text{sign}(\beta_{nj}-\beta_{nk}). \tag{5.6}$$

For the first term,

$$\frac{\partial S_n(\boldsymbol{\beta}_n)}{\partial \beta_{nj}} = \frac{\partial S_n(\boldsymbol{\beta}_{n0})}{\partial \beta_{nj}} + \sum_{k=1}^{p_n} \frac{\partial^2 S_n(\boldsymbol{\beta}_{n0})}{\partial \beta_{nj}\partial \beta_{nk}}(\beta_{nk}-\beta_{nk0}). \tag{5.7}$$

As argued in the proof of Statement (1), $\frac{1}{\sqrt{n}}\frac{\partial S_n(\boldsymbol{\beta}_{n0})}{\partial \beta_{nj}} = O_p(1)$. Thus $\frac{\partial S_n(\boldsymbol{\beta}_{n0})}{\partial \beta_{nj}} = O_p(\sqrt{n}) = O_p(\sqrt{np_n})$.

For $\sum_{k=1}^{p_n} \frac{\partial^2 S_n(\boldsymbol{\beta}_{n0})}{\partial \beta_{nj}\partial \beta_{nk}}(\beta_{nk}-\beta_{nk0})$, with Cauchy-Schwarz inequality, it's smaller than or equal to

$$n\sqrt{[\sum_{k=1}^{p_n}(\frac{\boldsymbol{X}^T\boldsymbol{\Omega}\boldsymbol{X}}{n}_{(j,k)})^2]}\|\boldsymbol{\beta}_n-\boldsymbol{\beta}_{n0}\|. \tag{5.8}$$

Because of the eigenvalue assumption on $\frac{\boldsymbol{X}^T\boldsymbol{\Omega}\boldsymbol{X}}{n}$,

$$[\sum_{k=1}^{p_n}(\frac{\boldsymbol{X}^T\boldsymbol{\Omega}\boldsymbol{X}}{n}_{(j,k)})^2] = O(1), \tag{5.9}$$

thus $\sum_{k=1}^{p_n} \frac{\partial^2 S_n(\boldsymbol{\beta}_{n0})}{\partial \beta_{nj}\partial \beta_{nk}}(\beta_{nk}-\beta_{nk0})$ is also $O_p(\sqrt{np_n})$.

For the second term of (5.6),

$$\begin{aligned} & n\sum_{k\neq j} p'_{\lambda_n,\tau_n}(|\beta_{nj}-\beta_{nk}|)\text{sign}(\beta_{nj}-\beta_{nk}) \\ =& n\sum_{k\neq j,\beta_{nk0}\neq\beta_{nj0}} p'_{\lambda_n,\tau_n}(|\beta_{nj}-\beta_{nk}|)\text{sign}(\beta_{nj}-\beta_{nk}) \\ &+ n\sum_{k\neq j,\beta_{nk0}=\beta_{nj0}} p'_{\lambda_n,\tau_n}(|\beta_{nj}-\beta_{nk}|)\text{sign}(\beta_{nj}-\beta_{nk}). \end{aligned} \tag{5.10}$$

For the first term of (5.10), since $\liminf_{n\to\infty}\frac{1}{\tau_n}\min(|\beta_{nj0}-\beta_{nk0}| : \beta_{nj0}\neq\beta_{nk0}) > 1$, when n is large enough $|\beta_{nj0}-\beta_{nk0}| > \tau_n$. Since $\|\boldsymbol{\beta}_n-\boldsymbol{\beta}_{n0}\| = O_p(\sqrt{\frac{p_n}{n}})$, it's easy to see

$|\beta_{nj} - \beta_{nj0}| = O_p(\sqrt{\frac{p_n}{n}})$ for all j and

$$|\beta_{nj} - \beta_{nk}| - |\beta_{nj0} - \beta_{nk0}| = O_p(\sqrt{\frac{p_n}{n}}).$$

And because $\tau_n / \sqrt{\frac{p_n}{n}} \to \infty$,

$$|\beta_{nj} - \beta_{nk}| - |\beta_{nj0} - \beta_{nk0}| = o_p(\tau_n).$$

So when n is large enough, we also have $|\beta_{nj} - \beta_{nk}| > \tau_n$, thus $p'_{\lambda_n,\tau_n}(|\beta_{nj} - \beta_{nk}|) = 0$. For the second term of (5.10), since $\tau_n / \sqrt{\frac{p_n}{n}} \to \infty$, when n is large enough, $p'_{\lambda_n,\tau_n}(|\beta_{nj} - \beta_{nk}|) = \frac{\lambda_n}{\tau_n} > 0$. So with a large enough n, (5.10) simplifies to

$$n\frac{\lambda_n}{\tau_n} \sum_{k \neq j, \beta_{nk0} = \beta_{nj0}} \text{sign}(\beta_{nj} - \beta_{nk}).$$

So for (5.6), the first term is of order $O_p(\sqrt{np_n})$, and the second term is of order $n\frac{\lambda_n}{\tau_n}$. Since $\sqrt{\frac{n}{p_n}}\frac{\lambda_n}{\tau_n} \to \infty$, the sign of (5.6) is determined by the second part which reduces to

$$- \sum_{k \neq j, \beta_{nk0} = \beta_{nj0}} \text{sign}(\beta_{nj} - \beta_{nk}).$$

So the claim is proved. □

Proof of Statement **(2)**:

Next we show for the $\sqrt{n/p_n}$-consistent local maximizer $\hat{\boldsymbol{\beta}}_n$ of $L_n(\boldsymbol{\beta}_n)$, with probability tending to 1 it has the grouping consistency that

$$\hat{\beta}_{nj} = \hat{\beta}_{nt_n(j)} \text{ for all } j > s_n, \tag{5.11}$$

with t_n the true group mapping function defined in (3.48).

At a local maximizer $\hat{\boldsymbol{\beta}}_n$, for $\hat{\beta}_{nj}$ and a small enough ϵ, it must satisfy

$$\left.\frac{\partial L_n(\boldsymbol{\beta}_n)}{\partial \beta_{nj}}\right|_{\hat{\beta}_{nj}-\epsilon} \geqslant 0 \text{ and } \left.\frac{\partial L_n(\boldsymbol{\beta}_n)}{\partial \beta_{nj}}\right|_{\hat{\beta}_{nj}+\epsilon} \leqslant 0.$$

Now we show if (5.11) is violated, the above inequalities cannot hold.

Define groups $M_{nk} \triangleq \{k\} \cup \{j > s_n : t_n(j) = k\}$ for $k = 1, 2, \cdots, s_n$. Without loss of generality, look at the group M_{n1}. We show if $\hat{\beta}_{nj}$'s with $j \in M_{n1}$ are not all equal, there's a contradiction to $\hat{\boldsymbol{\beta}}_n$ being a local maximizer. Suppose there are H different values in $\{\hat{\beta}_{nj} : j \in M_{n1}\}$, and order them increasingly as $\hat{\beta}_{n(1)}, \hat{\beta}_{n(2)}, \cdots, \hat{\beta}_{n(H)}$. Let g_i be the size of the subset of M_{n1} which corresponds to $\hat{\beta}_{n(i)}$, i.e.

$$g_i = |\{j : j \in M_{n1} \text{ and } \hat{\beta}_{nj} = \hat{\beta}_{n(i)}\}|.$$

So $\sum_i g_i = |M_{n1}| \triangleq G$. First look at the subset that has the largest value $\hat{\beta}_{n(H)}$. Suppose for some j, $\hat{\beta}_{nj} = \hat{\beta}_{n(H)}$, we look at the partial derivative $\left.\frac{\partial L_n(\boldsymbol{\beta}_n)}{\partial \beta_{nj}}\right|_{\hat{\beta}_{nj}-\epsilon}$ with ϵ small enough. Based on the claim 5.0.1, we only need to look at the sum of signs. It can be seen for $\hat{\beta}_{nj} - \epsilon$, in $\sum_{k \neq j, \beta_{nk0} = \beta_{nj0}} \text{sign}(\beta_{nj} - \beta_{nk})$ there are $g_H - 1$ negatives and $G - g_H$ positives. To make the partial derivative at $\hat{\beta}_{nj} - \epsilon$ non-negative, $G - 2g_H + 1 \leqslant 0$ must hold. So $g_H \geqslant (G+1)/2$. Similarly look at the subset that has the smallest value $\hat{\beta}_{n(1)}$. Suppose for some k, $\hat{\beta}_{nk} = \hat{\beta}_{n(1)}$, then for $\left.\frac{\partial L_n(\boldsymbol{\beta}_n)}{\partial \beta_{nk}}\right|_{\hat{\beta}_{nk}+\epsilon}$, there are $g_1 - 1$ positives and $G - g_1$ negatives. To make the partial derivative non-positive, we have $2g_1 - G - 1 \geqslant 0$, so $g_1 \geqslant (G+1)/2$. Then $g_1 + g_H \geqslant G + 1 > G$ which cannot hold. The contradiction means $H = 1$, i.e. all $\hat{\beta}_{nj}$'s for $j \in M_{n1}$ are equal. The grouping consistency is proved. □

Proof of Statement **(3):**

Based on Statement **(2)**, with probability tending to 1, the $\sqrt{n/p_n}$-consistent local maximizer of the penalized likelihood takes the form of $(\hat{\boldsymbol{\beta}}_{n1}^T, \tilde{\hat{\boldsymbol{\beta}}}_{n2}^T)^T$, where $\hat{\boldsymbol{\beta}}_{n1}$ is the estimator for the s_n unique values $\boldsymbol{\beta}_{n10}$ and in $\tilde{\hat{\boldsymbol{\beta}}}_{n2}$, $\tilde{\hat{\beta}}_{nj} = \hat{\beta}_{nt_n(j)}$ for $j > s_n$. It's easy to see that $\hat{\boldsymbol{\beta}}_{n1}$ is a $\sqrt{n/p_n}$-consistent local maximizer of $L_n((\boldsymbol{\beta}_{n1}^T, \tilde{\boldsymbol{\beta}}_{n2}^T)^T)$, where $\tilde{\beta}_{nj} = \beta_{nt_n(j)}$ for $\tilde{\beta}_{nj}$ in $\tilde{\boldsymbol{\beta}}_{n2}$ which makes L_n a function of $\boldsymbol{\beta}_{n1}$ alone. Redefine this

function as $L_{n,1}(\boldsymbol{\beta}_{n1})$. So $\hat{\boldsymbol{\beta}}_{n1}$ satisfies $\dfrac{\partial L_{n,1}(\hat{\boldsymbol{\beta}}_{n1})}{\partial \beta_{nj}} = 0$ for $j = 1, 2, \cdots, s_n$. Accordingly we also have a log-likelihood function $S_{n,1}(\boldsymbol{\beta}_{n1})$. Detailedly,

$$\begin{aligned} &S_{n,1}(\boldsymbol{\beta}_{n1}) = (\boldsymbol{Y} - \boldsymbol{Z}\boldsymbol{\beta}_{n1})^T \boldsymbol{\Omega} (\boldsymbol{Y} - \boldsymbol{Z}\boldsymbol{\beta}_{n1}), \\ &L_{n,1}(\boldsymbol{\beta}_{n1}) = S_{n,1}(\boldsymbol{\beta}_{n1}) - n \sum_{1 \leqslant k < j \leqslant s_n} |M_{nj}||M_{nk}| p_{\lambda_n, \tau_n}(|\beta_{nj} - \beta_{nk}|). \end{aligned} \tag{5.12}$$

Thus

$$\begin{aligned} &\frac{\partial L_{n,1}(\hat{\boldsymbol{\beta}}_{n1})}{\partial \beta_{nj}} \\ =&\frac{\partial S_{n,1}(\hat{\boldsymbol{\beta}}_{n1})}{\partial \beta_{nj}} - n \sum_{k \neq j} |M_{nj}||M_{nk}| p'_{\lambda_n, \tau_n}(|\hat{\beta}_{nj} - \hat{\beta}_{nk}|) \text{sign}(\hat{\beta}_{nj} - \hat{\beta}_{nk}) \end{aligned} \tag{5.13}$$

Since $\liminf_{n \to \infty} \frac{1}{\tau_n} \min(|\beta_{nj0} - \beta_{nk0}| : \beta_{nj0} \neq \beta_{nk0}) > 1$ and $p'_{\lambda, \tau}(x) = 0$ for $x > \tau$, with a large enough n, we have the second term in (5.13) equals 0. For the first term we have

$$\frac{\partial S_{n,1}(\hat{\boldsymbol{\beta}}_{n1})}{\partial \boldsymbol{\beta}_{n1}} = \frac{\partial S_{n,1}(\boldsymbol{\beta}_{n10})}{\partial \boldsymbol{\beta}_{n1}} + \frac{\partial^2 S_{n,1}(\boldsymbol{\beta}_{n10})}{\partial \boldsymbol{\beta}_{n1} \partial \boldsymbol{\beta}_{n1}^T} (\hat{\boldsymbol{\beta}}_{n1} - \boldsymbol{\beta}_{n10}).$$

So

$$\boldsymbol{Z}^T \boldsymbol{\Omega} (\boldsymbol{Y} - \boldsymbol{Z}\boldsymbol{\beta}_{n10}) - \boldsymbol{Z}^T \boldsymbol{\Omega} \boldsymbol{Z} (\hat{\boldsymbol{\beta}}_{n1} - \boldsymbol{\beta}_{n10}) = 0.$$

Since $\boldsymbol{X}^T \boldsymbol{\Omega} \boldsymbol{X}$ are positive definite for all n, it's easy to see $\boldsymbol{Z}^T \boldsymbol{\Omega} \boldsymbol{Z}$ are also positive definite for all n. Multiplying the above equation by $(\boldsymbol{Z}^T \boldsymbol{\Omega} \boldsymbol{Z})^{-1}$ we get

$$(\boldsymbol{Z}^T \boldsymbol{\Omega} \boldsymbol{Z})^{-1} \boldsymbol{Z}^T \boldsymbol{\Omega} (\boldsymbol{Y} - \boldsymbol{Z}\boldsymbol{\beta}_{n10}) - (\hat{\boldsymbol{\beta}}_{n1} - \boldsymbol{\beta}_{n10}) = 0.$$

Take expectation we can see $E(\hat{\boldsymbol{\beta}}_{n1} - \boldsymbol{\beta}_{n10}) = \boldsymbol{0}$, so $\hat{\boldsymbol{\beta}}_{n1}$ is unbiased. For any given vector $\boldsymbol{z}$,

$$\boldsymbol{z}^T (\boldsymbol{Z}^T \boldsymbol{\Omega} \boldsymbol{Z})^{-1} \boldsymbol{Z}^T \boldsymbol{\Omega} (\boldsymbol{Y} - \boldsymbol{Z}\boldsymbol{\beta}_{n10}) = \boldsymbol{z}^T (\hat{\boldsymbol{\beta}}_{n1} - \boldsymbol{\beta}_{n10}).$$

So with probability tending to 1,

$$\text{var}(\boldsymbol{z}^T (\hat{\boldsymbol{\beta}}_{n1} - \boldsymbol{\beta}_{n10})) = \boldsymbol{z}^T \left(\boldsymbol{Z}^T \boldsymbol{\Omega} \boldsymbol{Z} \right)^{-1} \boldsymbol{z}.$$

Proof of Theorem 3.4.2:

The proof follows similar arguments as in the proof of theorem 3.4.1.

For the proof of Statement (1), the gradient and Hessian of $\tilde{S}_n$ are

$$\frac{\partial \tilde{S}_n(\boldsymbol{\beta}_{n0})}{\partial \boldsymbol{\beta}_n} = \boldsymbol{X}^T(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}_{n0}), \ \frac{\partial^2 \tilde{S}_n(\boldsymbol{\beta}_{n0})}{\partial \boldsymbol{\beta}_n \partial \boldsymbol{\beta}_n^T} = -\boldsymbol{X}^T\boldsymbol{X}. \tag{5.14}$$

Since $\lambda_{\max}(\frac{1}{n}\boldsymbol{X}^T\boldsymbol{\Omega}^{-1}\boldsymbol{X}) < h_2 < \infty$ and $0 < h_1 < \lambda_{\min}(\frac{1}{n}\boldsymbol{X}^T\boldsymbol{X})$, again using Cauchy-Schwarz inequality, we have in

$$\tilde{S}_n(\boldsymbol{\beta}_{n0} + c_n\boldsymbol{u}) - \tilde{S}_n(\boldsymbol{\beta}_{n0}) = c_n(\frac{\partial \tilde{S}_n(\boldsymbol{\beta}_{n0})}{\partial \boldsymbol{\beta_n}})^T\boldsymbol{u} + \frac{1}{2}\boldsymbol{u}^T\frac{\partial^2 \tilde{S}_n(\boldsymbol{\beta}_{n0})}{\partial \boldsymbol{\beta}_n \partial \boldsymbol{\beta}_n^T}\boldsymbol{u}c_n^2,$$

the first term is dominated by the second, which is negative.

So given any $\epsilon > 0$, there exists A and N large enough, such that when $n > N$,

$$P(\sup_{\{\|\boldsymbol{u}\|=A\}} L_n(\boldsymbol{\beta}_{n0} + c_n\boldsymbol{u}) < L_n(\boldsymbol{\beta}_{n0})) \geqslant 1 - \epsilon.$$

This implies there exists a local maximizer $\hat{\boldsymbol{\beta}}_n$ of $L_n(\boldsymbol{\beta}_n)$ and $\|\hat{\boldsymbol{\beta}}_n - \boldsymbol{\beta}_{n0}\| = O_p(\sqrt{p_n/n})$. Statement (1) is proved.

For the proof of Statement (2), it uses the eigenvalue assumption on $\frac{1}{n}\boldsymbol{X}^T\boldsymbol{\Omega}^{-1}\boldsymbol{X}$ and $\frac{1}{n}\boldsymbol{X}^T\boldsymbol{X}$, otherwise it's the same as the proof for theorem 3.4.1.

For the proof of Statement (3),

$$\frac{\partial \tilde{S}_{n,1}(\tilde{\boldsymbol{\beta}}_{n1})}{\partial \boldsymbol{\beta}_{n1}} = \frac{\partial \tilde{S}_{n,1}(\boldsymbol{\beta}_{n10})}{\partial \boldsymbol{\beta}_{n1}} + \frac{\partial^2 \tilde{S}_{n,1}(\boldsymbol{\beta}_{n10})}{\partial \boldsymbol{\beta}_{n1} \partial \boldsymbol{\beta}_{n1}^T}(\tilde{\boldsymbol{\beta}}_{n1} - \boldsymbol{\beta}_{n10}).$$

So

$$\boldsymbol{Z}^T(\boldsymbol{Y} - \boldsymbol{Z}\boldsymbol{\beta}_{n10}) - \boldsymbol{Z}^T\boldsymbol{Z}(\tilde{\boldsymbol{\beta}}_{n1} - \boldsymbol{\beta}_{n10}) = 0.$$

Since $\boldsymbol{X}^T\boldsymbol{X}$ are positive definite for all n, it's easy to see $\boldsymbol{Z}^T\boldsymbol{Z}$ are also positive definite for all n. Multiplying the above equation by $(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}$ we get

$$(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\boldsymbol{Z}^T(\boldsymbol{Y} - \boldsymbol{Z}\boldsymbol{\beta}_{n10}) - (\tilde{\boldsymbol{\beta}}_{n1} - \boldsymbol{\beta}_{n10}) = 0.$$

Taking expectation we can see $\mathrm{E}(\tilde{\boldsymbol{\beta}}_{n1} - \boldsymbol{\beta}_{n10}) = \mathbf{0}$.

And for any given vector $\boldsymbol{z}$,

$$\boldsymbol{z}^T(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\boldsymbol{Z}^T(\boldsymbol{Y}-\boldsymbol{Z}\boldsymbol{\beta}_{n10}) = \boldsymbol{z}^T(\tilde{\boldsymbol{\beta}}_{n1}-\boldsymbol{\beta}_{n10}).$$

So with probability tending to 1,

$$\text{var}(\boldsymbol{z}^T(\tilde{\boldsymbol{\beta}}_{n1}-\boldsymbol{\beta}_{n10})) = \boldsymbol{z}^T[(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\left(\boldsymbol{Z}^T\boldsymbol{\Omega}^{-1}\boldsymbol{Z}\right)(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}]\boldsymbol{z}.$$

Proof of Corollary 3.4.3:

By standard Gauss-Markov theorem in linear regression, for a linear model $\boldsymbol{Y} = \boldsymbol{Z}\boldsymbol{\beta}_{10}+\boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \boldsymbol{\Omega}^{-1})$, the best linear unbiased estimator is $\hat{\boldsymbol{\beta}}_B = (\boldsymbol{Z}^T\boldsymbol{\Omega}\boldsymbol{Z})^{-1}\boldsymbol{Z}^T\boldsymbol{\Omega}\boldsymbol{Y}$. Its covariance matrix is $(\boldsymbol{Z}^T\boldsymbol{\Omega}\boldsymbol{Z})^{-1}$. The estimator $\hat{\boldsymbol{\beta}}_U = (\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\boldsymbol{Z}^T\boldsymbol{Y}$ is also unbiased and has variance $(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}(\boldsymbol{Z}^T\boldsymbol{\Omega}^{-1}\boldsymbol{Z})(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}$. So for any vector $\boldsymbol{z}$, we have $\text{var}(\boldsymbol{z}^T\hat{\boldsymbol{\beta}}_B) \leqslant \text{var}(\boldsymbol{z}^T\hat{\boldsymbol{\beta}}_U)$, which is

$$\boldsymbol{z}^T(\boldsymbol{Z}^T\boldsymbol{\Omega}\boldsymbol{Z})^{-1}\boldsymbol{z} \leqslant \boldsymbol{z}^T(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\boldsymbol{Z}^T\boldsymbol{\Omega}^{-1}\boldsymbol{Z}(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\boldsymbol{z}.$$

This shows the asymptotic variance of $\boldsymbol{z}^T\hat{\boldsymbol{\beta}}_{n1}$ is always smaller than or equal to that of $\boldsymbol{z}^T\tilde{\boldsymbol{\beta}}_{n1}$. For any $\boldsymbol{x}$, with probability tending to 1, $\text{var}(\boldsymbol{x}^T\hat{\boldsymbol{\beta}}_n) = \text{var}(\boldsymbol{z}^T\hat{\boldsymbol{\beta}}_{n1})$ and $\text{var}(\boldsymbol{x}^T\tilde{\boldsymbol{\beta}}_n) = \text{var}(\boldsymbol{z}^T\tilde{\boldsymbol{\beta}}_{n1})$ where $\boldsymbol{z}$ is the condensed form of $\boldsymbol{x}$ summing up covariates corresponding to the coefficients in the same group. So with probability tending to 1, $\text{var}(\boldsymbol{x}^T\hat{\boldsymbol{\beta}}_n) \leqslant \text{var}(\boldsymbol{x}^T\tilde{\boldsymbol{\beta}}_n)$ □

Proof of Corollary 3.4.4:

Since it's proved with probability tending to 1, $\hat{\boldsymbol{\beta}}_n$ and $\tilde{\boldsymbol{\beta}}_n$ both have grouping consistency and $\hat{\boldsymbol{\beta}}_{n1}$ and $\tilde{\boldsymbol{\beta}}_{n1}$ are both unbiased, it can be seen $\hat{\boldsymbol{\beta}}_n$ and $\tilde{\boldsymbol{\beta}}_n$ are also unbiased. The prediction error at any $\boldsymbol{x}$ is

$$\begin{aligned}
&E(y-\hat{y})^2\\
=&E(\boldsymbol{x}^T\boldsymbol{\beta}_{n0}+\epsilon-\boldsymbol{x}^T\hat{\boldsymbol{\beta}}_n)^2\\
=&E(\epsilon^2)+(E(\boldsymbol{x}^T\hat{\boldsymbol{\beta}}_n)-\boldsymbol{x}^T\boldsymbol{\beta}_{n0})^2+E(\boldsymbol{x}^T\hat{\boldsymbol{\beta}}_n-\boldsymbol{x}^TE\hat{\boldsymbol{\beta}}_n)^2\\
=&\sigma_\epsilon^2+\text{bias}^2+\text{variance}.
\end{aligned}$$

For both $\hat{\boldsymbol{\beta}}_n$ and $\tilde{\boldsymbol{\beta}}_n$, with probability tending to 1, the bias terms are 0. Based on corollary 3.4.3, $\hat{\boldsymbol{\beta}}_n$ gives a smaller variance. So the conclusion is derived. $\square$

Appendix B

This appendix uses plots to illustrate the correlation phenomenon of movie ratings mentioned in Section 3.4 with the Movielens 100k data.

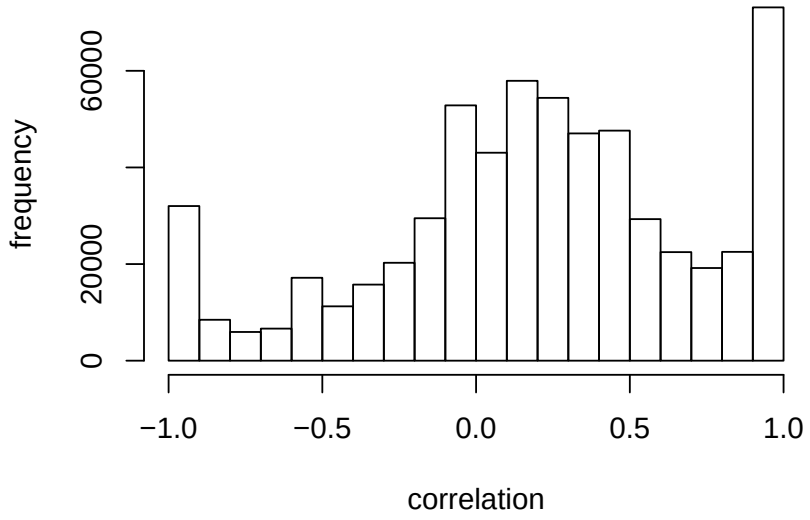

Figure 5.1: Correlation of two movie ratings

This figure shows the correlation of two movie ratings for the Movielens 100k data. We can see the center of the correlation is around 0.3, and there are many movie pairs that have very high positive correlation.

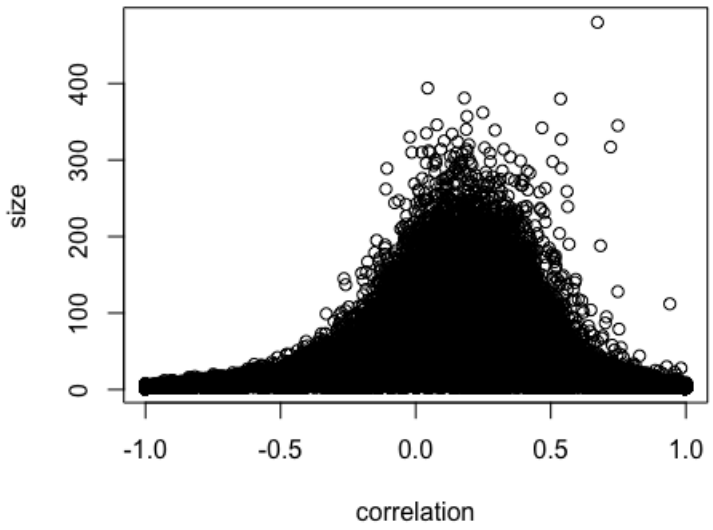

Figure 5.2: Sample size v.s. correlation

This figure is the scatterplot of correlations of movie ratings and the sample size for the Movielens 100k data. The sample size is the number of users that rated the two movies simultaneously. 27% of movie pairs have correlations above 0.5. Among these highly correlated pairs, there're some correlations calculated based on relatively small sample size, which means they can be either nonsignificant or represent the group effect for a small group. Furthermore, from the plot we can also see there are some movie pairs with very large positive correlation based on ratings from a large number of users. These correlations with large sample sizes are more representative. Among them there are the Star Wars series, the God Father series, the Die Hard series, etc.