2017

A Study on User Demographic Inference Via Ratings in Recommender Systems

Changbin Li
Louisiana State University and Agricultural and Mechanical College, lichangbin@outlook.com

A STUDY ON USER DEMOGRAPHIC INFERENCE VIA RATINGS IN RECOMMENDER SYSTEMS

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Masters in Science in Computer Science

in

The Division of Computer Science and Engineering, School of EECS

by
Changbin Li
B.E., Beijing University of Chemical Technology, 2011
M.E., Beijing Jiaotong University, 2014
August 2017

To my parents.

ACKNOWLEDGEMENTS

I spent almost two years' pleasant graduate studying in computer science at Louisiana State University. I thank everyone who helped me overcome the difficulties in my studying and daily life in the US.

First, thank Dr. Mingxuan Sun, my advisor, and the Division of Computer Science and Engineering for giving me a teaching assistantship during my graduate research. Dr.Sun brought me into the world of computer science, and the experience of teaching assistant helped me communicate with the undergraduate efficiently and friendly.

I thank every instructor for imparting major knowledge. Special thanks to Dr. Lu Peng for teaching us so wonderful computer architecture knowledge and giving me much encouragement and confidence.

I am lucky to meet and know so many friends, Shungeng Zhang, Zhiming Zhang, Elizabeth Poche, Doug Lafield, Sa Xiao, Shengjie Guo, and lab members, Qing Chen, Fei Li, Guangyue Xu, Jin Shang, Zhongzhu Peng, etc. The communication with them let me learn other new interesting things not limited in the research and enlarge my outlook.

I thank my committee, Dr. Jian Zhang, and Dr. Konstantin Busch, for their comments on this thesis, and I thank Dr. Mingxuan Sun again for directing this research.

Thank my classmates, Qingkai Lu, Ming Wang, Bin Zhang, etc. for encouraging each other to continue our research and explore the future world and life together.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Everyday, millions of people interact with online services that adopt recommender systems, such as personalized movie, news and product recommendation services. Research has shown that the demographic attributes of users such as age and gender can further improve the performance of recommender systems and can be very useful for many other applications such as marketing and social studies. However, users do not always provide those details in their online profiles due to privacy concern. On the other hand, user interactions such as ratings in recommender systems may provide an alternative way to infer demographic information. Most existing approaches can infer user demographics based on sufficient interaction history but could fail for users with few ratings. In this thesis, we study the association between users demographic information and their ratings, and explore the tradeoff between user privacy and the utility of personalization. In particular, we present a novel multi-task preference elicitation method, with which a recommender system asks a new user to rate selected items adaptively and infers the demographics rapidly via a few interactions. Experimental results on real-world datasets demonstrate the performance of the proposed method in terms of the accuracy of both demographics inference and rating prediction.

CHAPTER 1

INTRODUCTION

1.1 Recommender System

Everyday, millions of people interact with online services that adopt recommender systems, such as movie, news and product recommendation services. A user browses news at an airport, watches a movie in a café, and shops online almost anywhere. Usually, there are huge selections of items such as Amazon products, the consumers are inundated with information overload and they have to spend lots of time exploring different items, which is very inefficient. Recommender systems with machine learning algorithms emerge to provide each individual user with some items they might be interested in (such as movies, books, etc.) and assist users to make better decisions [28, 29].

Personalized recommendation algorithms are the focus of recommender systems, which has attracted more and more attention from both academia and industry. For example, Amazon and Youtube are always developing more and more advanced and accurate algorithms to improve the accuracy of their recommender systems and provide customers personalized products and videos. In addition, the famous Netflix competition was held between October 2006 and July 2009, which awarded 1 million US dollars to the best team for improving the prediction accuracy of the Netflix's own recommender system by 10%.

User demographics such as their age, gender and ethnicity information are important user profiles, which can be used to improve personalized recommendations and enable other richer services such as targeted advertisement and marketing. However, an increasing number of online users do not provide demographics due to privacy concern [3]. Inferring private attributes of users and improving personalized recommendations simultaneously is fundamentally important yet challenging.

1.1.1 Rating Prediction

Rating prediction problem is the most central problem in personalized recommender systems, which predicts the unknown ratings for a pair of a user and an item given previous user-item interaction history.

Table 1.1: An Example of Rating Matrix

	item A	item B	item C	item D	item E	item F
Jim	5	?	3	?	?	1
Kate	?	1	?	4	?	?
Bob	2	4	?	?	?	5
David	?	2	?	?	3	?

Suppose we have M users and N items, and a rating matrix R. R_{ij} denotes the element in the rating matrix, and it represents the rating of item j given by user i. R_{ij} represents the degree of preference of user i to item j. It could be expressed by either binary values $+1, -1$ indicating whether the user viewed, like, clicked the item or not, or directly rating numbers (e.g., $R_{ij} \in \mathbb{N} : 1 \leq r \leq 5$). As shown in Table 1.1, in addition, $R_{ij} =?$ denotes the unknown rating. Other numbers in the table represent known ratings. Higher score means higher preference. Jim, for instance, likes item A very much, but does not like item F. Since most users only rate a small set of items, the rating matrix is very sparse. The goal of the rating prediction in recommender systems is to predict the unobserved ratings in the rating matrix.

Currently, the rating prediction algorithms could be classified into three main categories: content-based filtering, collaborative filtering, and hybrid method [1, 19]. Content-based filtering provides a user with items similar to those the user liked in the past [1]. This kind of method needs to analyze the attributes of items in order to recommend similar items. In a movie recommendation application, for example, content-based filtering methods would recommend the movies whose genres, actors or directors are similar to the ones a user has rated highly in the past. According to the analysis of these movies, the system would find similar movies and recommend them to the user. Unlike the content-based filtering, collaborative filtering based approaches do not need content information and they recommend to a user items favored by the like-minded. However, collaborative filtering approaches suffer from the cold-start problem where few ratings are available. Hybrid methods try to overcome the limitations of the former two methods and combines them together.

1.1.2 Demographic Inference

Users' private attributes can be used to improve the accuracy of prediction in recommender system, which is a common trait shared by most of the recommender systems. In other words, the more detailed information of users the system has, the more accurate recommendation it would offer. Therefore, service providers would collect almost all information related to users, such as age, gender, browsing and clicking history, to ensure more accurate and personalized recommendations. For example, Amazon would utilize users' registration profiles and product-reviewing histories to recommend related items to users.

A recommender system may explicitly solicit user demographics through user registration. However, online users are not always willing to provide such information due to privacy concern [3]. On the other hand, user interactions such as ratings in recommender systems may provide an alternative way to infer demographic information. For example, a Netflix user who likes romance comedy and child-friendly movies may indicate that she is a mom. Existing attempts include the famous de-anonymization of Netflix Prize dataset [23] that link private Netflix rating data with public databases such as IMDB to partially infer some user identities. Other attempts [36] suggest that it is possible to infer user gender with as high as 80% accuracy given sufficient user ratings in recommender systems.

1.2 Challenges

Effectively inferring private attributes for users with few interactions is fundamentally important yet challenging. A large portion of users and items in relatively mature recommender systems are "cold". Typically, the population of users follows a power-law distribution with respect to the rating frequency and thus most users inhabit the long tail. For example, Netflix movie dataset contains 100 million movie ratings from $480,189$ users over $17,770$ movies and most users typically rate only a small number of movies. Figure 1.1 shows the histograms of the number of ratings per user for three datasets. Since most existing inference approaches largely depend on sufficient interaction history, they could fail for users with few interactions, which is well-known as the cold-start problem [10, 26, 31].

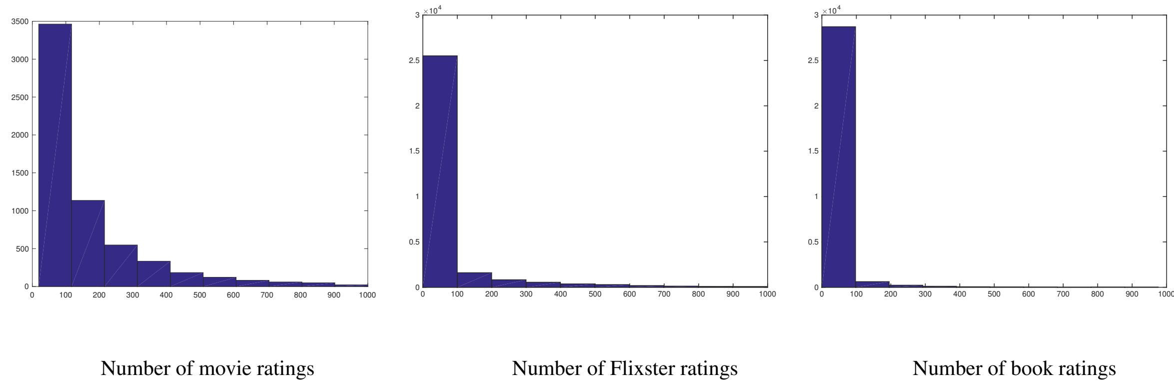

Figure 1.1: Histograms of the number of ratings per user for three datasets: MovieLens (left), Flixster (middle) and Book-Crossing (right).

A natural approach to circumvent cold-start scenario is to elicit new users' responses to a few selected questions and refine the estimation of user private attributes progressively. A lengthy exploration is intimidating, which may cause users to abandon the system at the very beginning. An adaptive process that queries users based on the previous responses is found to be more effective and variations of decision tree models [8] work well for this purpose. For example, a system queries new users "Do you like Sense and Sensibility?". The users are then directed to subtrees based on previous responses. The system gradually refines the estimation with higher confidence. Note that the primary goal of most web services is to attract and retain users, and thus the items selected to ask should be sufficient for both estimating user private attributes and improving recommendation accuracy at the same time.

1.3 Related Work

There has been a substantial body of work on demographic inference from various online human activities. Studies including [21] demonstrate that it is possible to infer private user attributes from online social networks given a small fraction of users who are willing to provide their private attributes such as location and interests. A variety of online activities are examined for demographic inference such as friendship on Facebook [16, 37], search queries [5], linguistic features of tweets [24, 25], and location check-ins [38].

Accurate demographic inference in recommender systems is challenging since most of the user information such as ratings is not as informative as activity information in Facebook, Twitter and LinkedIn. Until recently, studies such as [36, 4] make the first attempt and suggest that it is possible to infer user gender with as high as 80% accuracy given sufficient user ratings in recommender systems. Studies including [9, 36, 4] also explore different ways to perturb user ratings to prevent from private information leakage from the online security point of view.

The major difficulty in recommender systems is that the demographic prediction accuracy is affected by data sparsity since most users only provide a few ratings, which is known as the cold-start problem. Several studies [26, 27] focus on eliciting preference for new users to improve rating prediction accuracy through an interview process using a static set of questions. Approaches such as [27, 10, 39, 33] explore variations of decision tree models to adaptively select items to query. Active learning methods [6, 11, 13, 12, 4] select questions to query according to criteria such as minimizing the expected distance to the true user model. However, these methods usually involve computationally expensive optimization procedures, which are not efficient enough for online user interactions. Our work focuses on an effective and efficient cold-start recommendation method for both demographic inference and item recommendation.

Many studies on collaborative filtering recommender systems have focused on matrix factorization based methods [18, 14, 30, 17] including the Netflix competition winner [14] for rating prediction. Those methods seek to map users and items in a low-dimensional space to capture the intrinsic similarities. Other types of models [2, 31] utilize item features such as the genre, actor and director of a movie for cold-start recommendations. Our model is solely based on ratings and we adopt the latent factor based method for rating and demographic prediction.

1.4 Proposed Idea and Organization

We propose a novel preference elicitation method for new users, which learns the tasks of both demographic inference and rating prediction in a single framework. Specifically, a decision tree with each node corresponding to a query item is constructed and users are directed to one of the subtrees based on their previous responses. Latent user profiles are learned across the tasks of demographic inference and rating prediction simultaneously at each node, which enables knowledge transfer through the two related tasks and improves the prediction accuracy gradually for both tasks. An iterative optimization algorithm is proposed to alternate between decision tree learning and latent profile construction. In addition, the similarity between different items is better captured in a lower dimensional-space based on lower-rank matrix factorization. As a result, the items selected to query users are more effective in improving both recommendation and demographic inference accuracy. Experimental results on three benchmark datasets including the Flixster dataset, the MovieLens dataset and the Book-Crossing dataset demonstrate that the proposed method outperforms existing ones in cold-start recommendations.

The potential success of demographic inference for new users have positive impacts on not only recommender systems but also end users. In particular, if new users are aware of the type of privacy threats, they can learn to control the amount of information to release to better balance between preserving privacy and gaining personalized information. We also discuss the tradeoff between user privacy and the utility of personalization in the proposed method, where the former is captured by the prediction accuracy of demographics and the latter is captured by the recommendation accuracy.

The rest of the thesis is organized as the following:

In Chapter 2, we reviewed the classic collaborative filtering approaches including matrix factorization for rating prediction. We will also describe the evaluation criteria for recommendation accuracy.

In Chapter 3, we introduce our multi-task preference elicitation method, which can effectively infer new users' demographic information and improve the recommendation accuracy simultaneously.

In Chapter 4, we summarize and conclude this thesis.

CHAPTER 2

RATING PREDICTION

2.1 Review of Rating Prediction

As stated earlier, recommender system could help people alleviate information overload and reduce the decision-making time. According to a survey about the collaborative filtering (CF) methods [32], CF has been the most widely used recommender system recently. Many companies apply CF techniques into their recommender systems, such as Amazon, Netflix, Pandora, etc. Especially the born of matrix factorization model used to improve the prediction accuracy of Netflix's recommender system by 10% has attracted considerable attention no matter in industry or in academia.

However, the drawback of CF limited the improvement of prediction accuracy: data sparsity. Many users do not have much rating records in recommender systems. It is very difficult to predict personalized items for these users with few ratings. Under this circumstance, many researchers solicit the additional information to improve prediction accuracy. For example, Moshfeghi et al., add emotion and semantic based features to collaborative filtering in order to overcome the data sparsity [22]. Li et al., incorporate user's personal information including a users search query history, purchasing and browsing activities to improve one-class collaborative filtering [20]. Demographic information plays an important role in recommendation as well. For example, Wang et al. show that a demographic recommender system could classify different tourists based on their demographic information and make better recommendations according to different tourists classes [35]. In addition, [34] presents that the accuracy of rating prediction could be improved by the calculation of demographic correlations among different users and items.

In addition to users' information mentioned above, other user-contributed information, such as tags, multimedia content, and free-text reviews and comments, also has attracted much attention to improve the performance of recommender system [32].

2.2 Collaborative Filtering

As the most successful and widely used recommendation algorithm so far, CF only cares about the preference of items given by users, and does not use any information related to the content of items and users. As stated earlier, it recommends a item to a user based on the items previously rated by other users with similar taste and preference. Take a movie recommender system for example. If a system needs to recommend a movie to a customer, the first thing it should do is to find some other customers whose tastes are similar to the customer. Then the system would recommend the item rated highly by these customers with similar preference. Actually, CF is based on the assumption that a user who has shown his/her interests in the past and will continue these kind of interests in the future [32]. According to Breese et al. [7], CF techniques could be classified into two categories: memory-based and model-based approaches.

2.2.1 Memory-Based Approaches

Memory-based methods can be classified into two categories: user-based and item-based approaches [32]. User-based CF predicts an item's rating given by collecting the ratings of that item given by some similar users. The similarity between different users could be identified by Cosine similarity and Pearson correlation. One row in a specific rating matrix could represent the characteristic of the corresponding user. The predicted rating of item j given by user i can be formulated as follows:

$$\hat{R_{ij}} = \frac{1}{C} \sum_{k \in Z_i} sims(i,k) R_{kj}, \tag{2.1}$$

where Z_i is the selected similar users to user i, $sims(i,k)$ is the similarity between user i and user k. C is a constant.

Likewise, item-based CF predicts an item's rating given by a user based on the collection of historical similar ratings given by this user. The item similarity could also be determined by Cosine similarity and Pearson correlation. One column in a specific rating matrix could represent the characteristic of the corresponding item. The predicted rating of item j given by user i can be formulated as follows:

$$\hat{R_{ij}} = \frac{1}{C} \sum_{k \in Z_j} sims(j,k) R_{ik}, \tag{2.2}$$

where Z_j is the selected similar items to item j, $sims(j,k)$ is the similarity between item j and item k. C is a constant.

It should be noted that the above two equations are just the simplest form of memory-based CF. More different adjustments could be applied into these two equations.

As for the similarity calculation mentioned above, Cosine similarity and Pearson correlation between two different users (items) could be formulated as follows:

Cosine similarity:

$$sim(i,j) = \frac{V_i \cdot V_j}{|V_i| \cdot |V_j|}, \tag{2.3}$$

where V_i and V_j are vectors representing two users (items) i and j, respectively.

Pearson correlation:

$$sim(i,j) = \frac{\sum_d (V_{i,d} - \bar{V_i})(V_{j,d} - \bar{V_j})}{\sqrt{\sum_d (V_{i,d} - \bar{V_i})^2}\sqrt{\sum_d (V_{j,d} - \bar{V_j})^2}}, \tag{2.4}$$

where V_i and V_j are vectors representing two users (items) i and j, respectively. $\bar{V_i}$ and $\bar{V_j}$ are the average of non-zero values in the vector V_i and V_j, respectively. $V_{i,d}$ ($V_{j,d}$) is the element in the corresponding vector $V_i(V_j)$.

2.2.2 Model-Based Approaches

In addition, model-based approaches are also applied in the research and even more popular than memory-based approaches. Unlike the memory-based approaches, model-based tries to use the aggregation of ratings in a rating matrix and learn a model to predict the unknown ratings. Generally, it splits the rating matrix into two parts. One part is used to train the model, and the other is used to test and evaluate the model. These models include Bayesian network model, latent semantic model, mixture probability model, etc. [32] Recently, matrix factorization model has arisen a lot of attention because of its excellent performance in the Netflix contest [15].

2.3 Matrix Factorization Approaches

2.3.1 Regularized SVD

The basic idea of matrix factorization is to factorize one matrix into an inner product of two matrices. Suppose we have a rating matrix $R \in \mathbb{R}^{M \times N}$. Using matrix factorization, we could have:

$$R \approx UV^T \tag{2.5}$$

where $U \in \mathbb{R}^{M \times K}$, represents the attributes of users, $V \in \mathbb{R}^{N \times K}$, represents the attributes of items, and $K \ll M, N$. After this, a sparse rating matrix could be converted into two dense lower dimensional latent matrices. If so, the rating of item j given by user i could be approximated by $U_i V_j^T$, where U_i is the associated vector of user i, V_j is the associated vector of item j.

Regularized singular value decomposition (SVD) is the basic form of matrix factorization. The easiest way to let two latent matrices represent a rating matrix is to minimize the difference between R and UV^T. However, the goal of recommender system is to predict the unseen ratings based on the observed scores. Therefore, we need to avoid overfitting using regularization. The modified objective function needed to be minimized would be:

$$E = \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{N} I_{ij} (R_{ij} - U_i V_j^T)^2 + \frac{\lambda_u}{2} ||U||_F^2 + \frac{\lambda_v}{2} ||V||_F^2, \tag{2.6}$$

where I_{ij} is an indicator variable which equals to 1 if $R_{ij} > 0$, and 0 otherwise. λ_u and λ_v are used to regularize the objective function. Gradient-descent would be used to minimize this objective function and learn latent features U and V.

2.3.2 Other Matrix Factorization Models

In addition to the basic matrix factorization mentioned above, there exist hundreds of other ways of learning U and V, such as Non-negative Matrix Factorization, Probabilistic Models, Maximum Margin Matrix Factorization, Non-linear Matrix Factorization, Fast Non-parametric Matrix Factorization, etc. [19]

2.4 Evaluation Criteria

2.4.1 Measurement of Prediction Problem

The goal of rating prediction is to predict the unknown ratings based on the observed scores. In order to evaluate the performance of the recommender system, we need to compare the predicted ratings and the corresponding true observed ratings used to test the model. There are three most widely-used metrics which include

- Mean Absolute Error (MAE)

$$MAE = \frac{1}{m} \sum_{(u,i)\in\Omega} \mid f(u,i) - M_{u,i} \mid, \tag{2.7}$$

  where $f(u,i)$ is the prediction of rating on item i by user u, and $M_{u,i}$ is the true rating value. m is the total number of predicted ratings.

- Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{1}{m} \sum_{(u,i)\in\Omega} (f(u,i) - M_{u,i})^2}, \tag{2.8}$$

  where $f(u,i)$ is the prediction of rating on item i by user u, and $M_{u,i}$ is the true rating value. m is the total number of predicted ratings.

- Normalized Mean Absolute Error (NMAE)

$$NMAE = \frac{MAE}{r_{max} - r_{min}}, \tag{2.9}$$

  where r_{max} and r_{min} represent the maximum and minimum of ratings in the dataset.

It should be noted that MAE and RMSE could also evaluate the accuracy of regression problems. In the experiments of this thesis, MAE and RMSE are considered as the measurement.

2.4.2 Measurement of Classification Problem

In the analysis of binary classification, 4 measurements: precision, recall, F_1, and accuracy, are the most widely-used metrics. Table 2.1 shows a confusion matrix and lists four different cases: TP, FN, TN, and FP. TP means true observation is positive, and is predicted to be positive. FN means true observation is positive, and is predicted to be negative. TN means true observation is negative, and is predicted to be negative as well. FP means true observation is negative, and is predicted to be positive. Among the four cases in Table 2.1, four measures can be generated.

Table 2.1: Confusion Matrix of Binary Classification

	prediction positive	prediction negative
true positive	True Positive (TP)	False Negative (FN)
true negative	False Positive (FP)	True Negative (TN)

Precision measures how many positive predictions are actual positive observations.

$$Prec = \frac{\text{positive predicted correctly}}{\text{all positive predictions}} = \frac{TP}{TP + FP} \tag{2.10}$$

Recall measures how many actual positive observations are predicted correctly.

$$Recall = \frac{\text{predicted to be positive}}{\text{all positive observations}} = \frac{TP}{TP + FN} \tag{2.11}$$

F_1 considers both the precision and recall to computer the score, which can be viewed as a weighted average of precision and recall like this,

$$F_1 = \frac{2 * Prec * Recall}{Prec + Recall} \tag{2.12}$$

Accuracy can also reflect how good a model is, which is the proportion of all predictions that are correct.

$$accuracy = \frac{\text{correct predictions}}{\text{all predictions}} = \frac{TP + TN}{TP + FN + FP + TN} \tag{2.13}$$

The higher all of these three measures are, the better the result of classification is.

CHAPTER 3

SIMULTANEOUS RATING PREDICTION AND DEMOGRAPHIC INFERENCE

3.1 Model

We describe the demographic inference model in the context of cold-start recommendation. We propose to construct an efficient rating elicitation process by exploring both demographics and ratings of warm-users in the training dataset. The key innovation is that we learn latent user profiles as a function of the responses to possible query items to best estimate both demographics and ratings. As described in Figure 3.1, the recommender system constructs a model to query users for better inferring private attributes based on training user data. At the visits of new users, the recommender system infers their demographic type and makes recommendations based on their answers to queries.

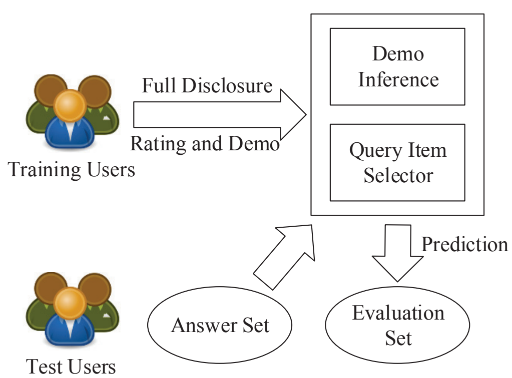

Figure 3.1: Evaluation framework for cold-start users.

In cold-start scenario, the system queries the user's rating on several selected items and constructs a rough user profile, which is then used to predict ratings for other items and at the same time to infer demographics. We propose to model the user profile as a function of user responses to the questions formed in the decision tree. Assume that there are n possible items to ask and each response takes a value in the set $\{1, -1, 0\}$ corresponding to *like*, *dislike* and *unknown*, respectively. Let x_i denote the answer of user i, which is an n-dimensional vector. Let T denote the function that maps the user response x_i to the user profile that is $u_i = T(x_i)$. We also assume that there exist latent item features and denote each item feature by v_j for item j.

For rating prediction, we assume that the rating r given user and item profiles follows a Gaussian distribution, that is:

$$p(r_{ij}|u_i, v_j, \sigma^2) = \mathcal{N}(v_j^\top u_i, \sigma^2). \tag{3.1}$$

Similarly, we can assume prior on user and item profiles. For example, $p(v_j|\sigma_v^2) = \mathcal{N}(v_j|0, \sigma_v^2)$.

For demographic prediction, we assume that the demographic label y such as age or gender follows some distribution

$$y_i \in p(y_i|\theta^\top u_i), \tag{3.2}$$

where θ is the regressor for continuous label prediction or the classifier for discrete label prediction.

Given observed ratings $O = \{(i, j) \mid r_{ij} \text{ is observed}\}$ and demographic information $S = \{i \mid y_i \text{ is observed}\}$, where $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, our goal is to learn the function T, item profile v_j for each item j, and the regressor θ to minimize the negative log posterior of the model, which is equivalent to the following objective:

$$\min_{T,V,\theta} \lambda_r \sum_{(i,j)\in O} \ell_r(r_{ij}, T(x_i)^\top v_j) + \lambda_s \sum_{i\in S} \ell_s(y_i, T(x_i)^\top \theta) + \lambda_v ||V||^2 + \lambda_\theta ||\theta||^2, \tag{3.3}$$

where $\ell_r(\cdot,\cdot)$ and $\ell_s(\cdot,\cdot)$ denote the loss functions for ratings and demographic information respectively, and λ_r and λ_s are weights. The last two parameters λ_v and λ_θ are regularization terms for $V = [v_1, v_2, \ldots, v_n]$, a matrix containing all item profiles v_j, and θ is the regressor. For simplicity, we name θ the regressor, which actually means the classifier in the case of discrete variable prediction. Specifically, the probability model of rating r_{ij} and demographic information y_i is encoded through the choice of loss functions. Similarly, the prior over parameters v and θ can also be translated into the regularization penalties.

We assume that the rating is continuous, i.e., $r_{ij} \in \mathbb{R}$, while the demographics can be continuous, i.e., $s_i \in \mathbb{R}$ such as the age, or binary, i.e., $s_i \in \{0, 1\}$ such as the gender. For a continuous variable, the loss function represents least mean square error, that is

$$\ell(y, \hat{y}) = (y - \hat{y})^2. \tag{3.4}$$

For binary variable, the choice of loss function can be the logistic regression error or least mean square error, that is

$$\ell(y, \hat{y}) = (1 - y\hat{y})^2. \tag{3.5}$$

3.1.1 Alternative Optimization

The parameters in the objective function defined in equation (3.3) can be learned through an alternative optimization. Specifically, we alternate between the following two steps:

1. Given item profile V and regressor θ, we try to fit a decision tree T such that:

$$\min_{T} \lambda_r \sum_{(i,j)\in O} \ell_r(r_{ij}, T(x_i)^\top v_j) + \lambda_s \sum_{i\in S} \ell_s(y_i, T(x_i)^\top \theta). \tag{3.6}$$

2. Given $T(x)$, we can compute v_j and θ such that

$$\min_{V} \sum_{(i,j)\in O} \ell_r(r_{ij}, T(x_i)^\top v_j) + \lambda_v ||V||^2, \tag{3.7}$$

$$\min_{\theta} \sum_{i\in S} \ell_s(y_i, T(x_i)^\top \theta) + \lambda_\theta ||\theta||^2. \tag{3.8}$$

The item profile v_j and the regressor θ can be initialized randomly. Another option for item profile initialization is through matrix factorization method such as [18] using training data. Given the decision tree T, a closed-form solution for the item profiles v_j ($j = 1, 2, \ldots, n$) exists:

$$v_j = \left(\sum_{(i,j)\in O} T(x_i)T(x_i)^\top + \lambda_v I \right)^{-1} \left(\sum_{(i,j)\in O} r_{ij} T(x_i) \right). \tag{3.9}$$

The regressor θ can be generally solved through gradient decent and updated as $\theta = \theta - \delta \triangle\theta$ where δ is the learning rate and $\triangle\theta$ is:

$$\triangle\theta = \sum_{i\in S} \ell'_s(y_i, \hat{y}_i) T(x_i) + \lambda_\theta \theta, \tag{3.10}$$

where $\hat{y}_i = T(x_i)^\top \theta$ consists of previous estimations.

The major challenge is that the number of possible items to query can be quite large, e.g., $n \sim 10^5$ in a movie recommender system. It is therefore computational prohibitive to search over all possible trees in order to get a global optimal solution to equation (3.6). We propose an efficient greedy algorithm to find an approximation.

3.1.2 Decision Tree Construction

Compared with classification and regression loss in traditional decision tree algorithms such as C4.5 and CART [8], our objective is to minimize the loss of both rating prediction and demographic inference as defined in equation (3.6). The decision tree is constructed in a top-down approach using training user data to minimize the loss recursively. A ternary decision tree to represent the mapping function T is suggested in previous work [10] to account for a large portion of users with no explicit responses.

Specifically, for each node in the tree, we learn the best set of questions by optimizing the objective defined in equation (3.6). The decision tree then splits the user set into three subsets L, D, and U according to the responses to those questions. The procedure is recursive until the tree reaches a certain depth. Starting from the root, given an item j to query, users are divided into three groups L, D, and U if the response value is $x_{ij} = 1, -1, 0$ corresponding to "like", "dislike" and "unknown". Generally, more than one item can be selected at each node to minimize user cognitive burden as suggested in [33]. In such cases, we assign each item with a weight and denote the n-dimensional weight vector by w, which defines a hyperplane to partition user responses into different groups. Users at the current node are split into group L if the answer $x_i^\top w$ is positive, group D if $x_i^\top w$ is negative, and group U only when a user answers none of the questions. To find the weight vector w that leads to the best split, we minimize the following function:

$$
\begin{aligned}
\min_{w} & \lambda_r \sum_{i \in L(w)} \sum_{(i,j) \in O} \ell_r(r_{ij}, u_L^\top v_j) + \lambda_s \sum_{i \in L(w) \cap S} \ell_s(y_i, u_L^\top \theta) \\
& + \lambda_r \sum_{i \in D(w)} \sum_{(i,j) \in O} \ell_r(r_{ij}, u_D^\top v_j) + \lambda_s \sum_{i \in D(w) \cap S} \ell_s(y_i, u_D^\top \theta) \\
& + \lambda_r \sum_{i \in U(w)} \sum_{(i,j) \in O} \ell_r(r_{ij}, u_U^\top v_j) + \lambda_s \sum_{i \in U(w) \cap S} \ell_s(y_i, u_U^\top \theta) \\
\text{s.t. } & \|w\|_0 \leq l,
\end{aligned}
\tag{3.11}
$$

where u_L, u_D and u_U are the optimal user profiles for the child nodes L, D and U. In addition, $\|w\|_0$ denotes the number of non-zeros in w and the constraint $\|w\|_0 \leq l$ determines that the weight vector w cannot have more than l non-zeros. For simplicity, we assume one item to ask at each node, that is $l = 1$. We set all except the jth entry in weight vector w to 0. The problem boils down to finding the best single item to split users so as to minimize prediction loss. However, our framework can be easily generalized to multi-item split by adopting existing techniques as described in [33].

The optimal profiles u_L, u_D and u_U are the ones to minimize prediction loss in each child. Specifically, the profile u_L in group L is solved by [33].

3.1.3 Computational Complexity

The full algorithm is summarized in Algorithms 1 and 2. For the tree construction, at each node, the complexity to compute latent profiles u_L, u_D and u_U for each possible split is $O(nk^2 + k^3)$ including inverting a square matrix of size k, where n is the number of items and k is the dimension of latent space. There are totally n possible splits since we consider one item to query at each node. Using a similar analysis in [39], the time complexity of preparing matrix coefficients for all possible splits is $\sum_{i=1}^{m} |O_i|^2$ at each tree level, where $|O_i|$ is the number of observed ratings of user i and m is the number of users. The complexity for building the whole tree is thus $O(d \sum_{i=1}^{m} |O_i|^2 + \beta n k^3 + \beta n^2 k^2)$, where d is the depth of the tree and β is the number of nodes in the tree. In practice, smaller parameter values for k and d are sufficient for good model performance. For example, the tree depth d is around 8 and k usually ranges from 10 to 20. The computational complexity for updating item profiles v_j for all $j = 1, 2, \ldots, n$ is $O(nk^3 + n|O^j|k^2)$ in equation (3.9), where $|O^j|$ is the number of users who rate item j. Similarly, the complexity for updating the regressor θ is $O(k^3 + mk^2)$ with choices of loss functions in equation (3.4) and (3.5). The alternative optimization usually converges in a few iterations.

3.2 Experiments

In the experiments, we would like to demonstrate that our multi-task model is effective in improving the prediction accuracy of both demographic inference and item recommendation with only a few sets of selected questions for cold-start users. We further discuss the tradeoff between user privacy and the utility of personalization, where the former is captured by the prediction accuracy of demographics and the latter is captured by the recommendation accuracy. We examine the estimation framework on three movie and book recommendation datasets: MovieLens, Flixster and Book-Crossing.

Algorithm 1 Alternative Optimization

Require: The training data $R = r_{ij}|(i, j) \in O, Y = y_i \in S$.

Ensure: Estimate decision tree T, item profile v_j ($j = 1, 2, \ldots, n$), and regressor θ.

Initialize v_j ($j = 1, 2, \ldots, n$) using [18].

Initialize θ randomly.

while not converge **do**

  Fit a decision tree T using Algorithm 2.

  Update v_j using Equation (3.9).

  Update θ using Equation (3.10).

end while

return T, v_j ($j = 1, 2, \ldots, n$) and θ.

Algorithm 2 Greedy Tree Model

function FitTree(UsersAtNode)

Compute u_L, u_D and u_U using [33].

Find the best split item or item set using [33].

Split users into three groups $L(w)$, $D(w)$ and $U(w)$.

if square error reduces after split **and** depth $<$ maxDepth **then**

  call FitTree($L(w)$), FitTree($D(w)$) and FitTree($U(w)$) to construct subtrees.

end if

return T with $T(x)$

end function

3.2.1 Experiment Setting

We evaluate the performance of our multi-task model in a cold-start setting. For each dataset, we randomly split the users into a training set and a test set, each containing 80% and 20% users, respectively. We assume that the users in the training set are warm-start users and their ratings and demographic information are visible to the system. We learn our model and construct the set of items as the probing questions based on training data. In contrast, we assume that the users in the test set are cold-start users. The ratings of each user in the test set are further split into two disjoint sets: answer and evaluation sets that contain 80% and 20% ratings. The answer set is used to simulate the responses of cold-start users in the rating elicitation process. The evaluation set is used to evaluate the rating prediction accuracy for withheld items after the elicitation process. Meanwhile, the demographic information of each user in the test set is used to evaluate the demographic prediction accuracy. The evaluation process is summarized in Figure 3.1. In the rating elicitation process, we select items to query user responses. For example, in the movie recommendation, we ask a user to rate each movie. For simplicity, we ask for user binary responses and the question is in the form "Do you like movie *50 first date*?" A user is expected to answer *like*, *dislike* or *unknown*. We follow the standard settings [27, 10, 39] and simulate test user responses as the following: the response is "like" if a user's rating is larger than 3 and "dislike" otherwise. The response is "unknown" if no rating is observed.

We seek to answer the following questions:

1. Does the proposed algorithm outperform baselines in terms of the demographic prediction accuracy with respect to the number of query items?
2. Does the multi-task model also enhance the recommendation accuracy?

3. How many items to query are sufficient for demographic inference? What is the tradeoff between user privacy and the utility of personalization?

3.2.2 Dataset and Evaluation Metrics

The MovieLens dataset[1] contains about $3,900$ movies, $6,040$ users and about 1 million ratings. In this dataset, about 4% of the user-movie interactions are observed and each user rates at least 20 movies. The ratings are integers ranging from 1 (dislike) to 5 (like). For the Flixster dataset, we select users with at least 20 ratings and movies with at least 60 ratings, which results in a subset of ratings for $5,795$ movies by $23,488$ users. The ratings are from 1 to 5. The Book-Crossing data [2] is the most sparse dataset, with about 0.2% rating density. We select users with at least 20 ratings and movies with at least 4 ratings and obtain a subset of ratings for $34,963$ movies by $5,411$ users. The ratings are from 1 to 10 and we normalize the ratings to 1 to 5 in the same scale as the other two datasets for comparison. In terms of demographic information, the MovieLens dataset has gender and discrete age labels. The Flixster dataset [3] has gender and continuous age labels. Both datasets have imbalanced gender distribution. There are about 71% males in MovieLens users and about 43% males in Flixster users. The Book-Crossing dataset has only continuous age labels. The mean age for Flixster and Book-Crossing are 24 and 35, respectively. In our experiments, we choose age regression tasks using Flixster and Book-Crossing data. We also choose Flixster and MovieLens for gender classification. For all three datasets, we compare the rating prediction accuracy. The details of each dataset are

[1]See `https://grouplens.org/datasets/movielens/`
[2]See `http://www2.informatik.uni-freiburg.de/~cziegler/BX/`
[3]See `http://www.cs.sfu.ca/~sja25/personal/datasets/`

shown in Table 3.1 and Table 3.2. Table 3.1 shows the minimum ratings of each user and each item, and ratings range in these three different datasets.

Table 3.1: Dataset Description 1

Dataset	Min # of ratings(Users)	Min # of ratings(Items)	Ratings Range
MovieLens	20	1	$1-5$
Flixster	20	60	$1-5$
Book-Crossing	20	4	$1-10$

The rating prediction performance is evaluated with the root mean square error (RMSE). For age regression, we use standard mean absolute error (MAE), and rooted mean squared error (RMSE). The MAE measures the average of the absolute errors in test sets and the individual differences of each test data are weighted equally in the average. The RMSE measures the rooted squared error between truth and predicted values and then averaged over the samples. This means that the RMSE is most powerful to measure particularly undesirable large errors. For gender classification, we use precision, recall and fscore to measure the performance for imbalanced binary classes.

3.2.3 Prediction Accuracy of User Demographics

In this section, we evaluate the performance of our multi-task model in terms of demographic prediction accuracy in cold-start settings to answer the first question in Sec-

Table 3.2: Dataset Description 2

Dataset	Users	Items	Ratings	Density	Gender (M/F)	Average Age
MovieLens	$6,040$	$3,952$	$1,000,209$	4.19%	$71\%/29\%$	NA
Flixster	$23,488$	$5,795$	$5,625,681$	4.13%	$43\%/57\%$	24
Book-Crossing	$5,411$	$34,963$	$384,888$	0.20%	NA	35

tion 3.2.1. We compare our model "TreeMulti" with 4 baseline methods named as "Mean", "Variance", "Weight". and "TreeSingle". The baseline "Mean" selects the top l items to query based on the mean ratings in the training dataset. The items with higher mean values indicate the "goodness". On the other hand, the baseline "Variance" picks the top ones with highest rating variance across users [27]. The third one "Weight" [36] first trains a regressor toward age using ratings in the training dataset and picks the item whose corresponding regression coefficient has highest absolute values. The last one "TreeSingle" [8] is single-task decision tree model to predict demographics from ratings.

Figure 3.2 compares the prediction accuracy of our model with several standard baselines for age regression on datasets Flixster and Bookcrossing. The first row of Figure 3.2 compares the performance on dataset Flixster. For all methods, the age prediction error measured in MAE and RMSE decreases as more questions have been asked. Our model "TreeMulti" performs better than "TreeSingle" since the latent user profiles are learned through related tasks. Both tree models have big advantages over others especially within the first several questions. Specifically, our model achieves almost the same prediction accuracy within 5 questions as compared to 20 for others. Within 5 questions, we can predict age accuracy with MAE of 5 years, which is great considering a large range of users and sparse user responses. The model "Weight" also performs better than others in this sense and the model "Variance" performs only a little better than "Mean", which is reasonable since items with high means cannot help differentiating user types.

In the second row in Figure 3.2 where we compare performance on dataset Book-Crossing, we see similar trends. Overall, the dataset Book-Crossing is more challenging for prediction than Flixster since the training set is extremely sparse with a rating density of around 0.2%. Our model "TreeMulti" still performs better than others with no major differences among the others.

Figure 3.3 compares the prediction accuracy of our model with several standard baselines for gender classification on datasets Flixster and MovieLens. The first row of Figure 3.3 compares the performance on dataset Flixster. For all methods, the gender prediction accuracy measured in precision, recall and fscore increase as more questions have been asked in general. The only exception is that method "Mean" decreases with more questions in precision metric. Our model "TreeMulti" has a big advantage over others. Specifically, the prediction accuracy (fscore) of our model increases fastly from 3 to 5 questions and changes smoothly from 5 to 7 questions. The model "Weight" also performs better than others, followed by the model "Variance". Model "Mean" is the worst. The second row in Figure 3.3 compares gender classification performance on dataset MovieLens. Overall, the performance on MovieLens is better than Flixster. Our model "TreeMulti" performs better than others.

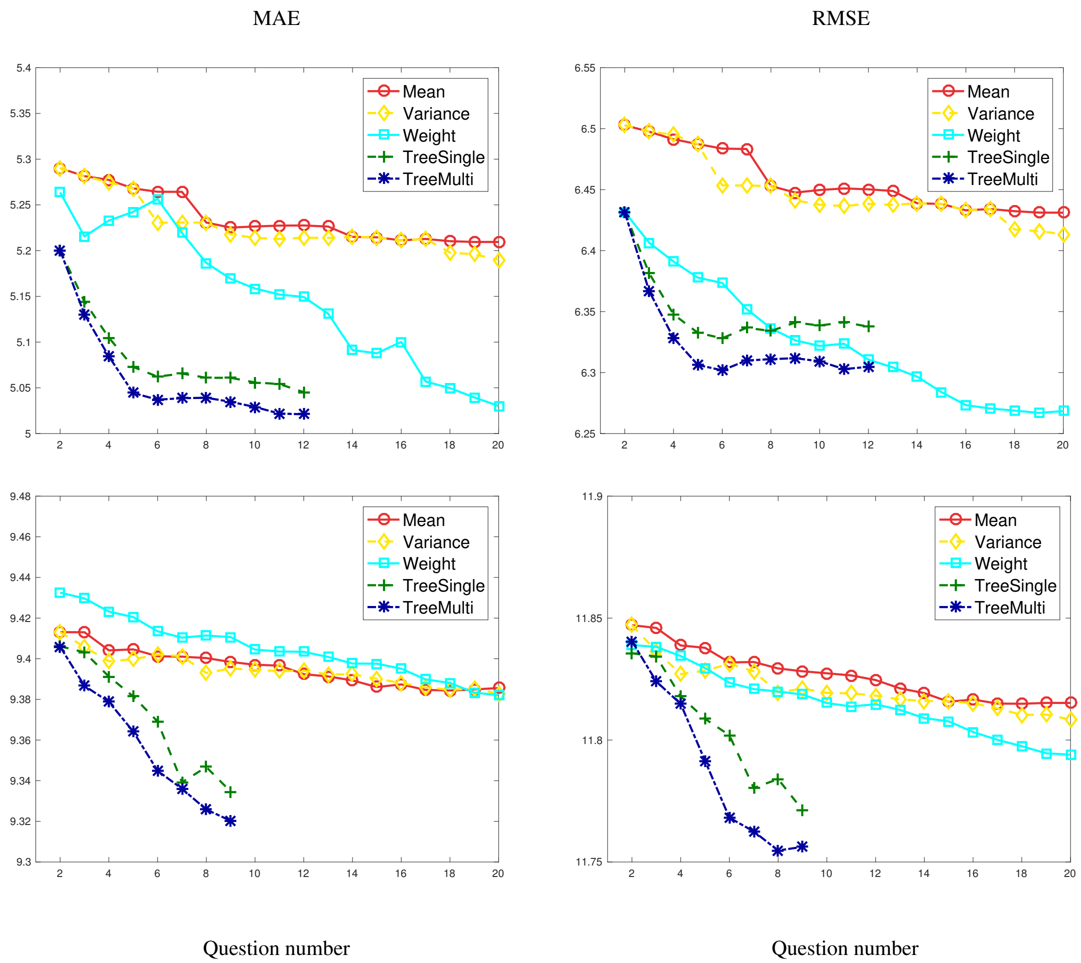

Figure 3.2: The age prediction metrics MAE and RMSE with respect to number of questions on two datasets Flixster (top row) and Book-Crossing (bottom row). For all methods, the prediction error decreases as the number of questions increases. It shows that our methods "TreeMulti" performs better than baselines for both datasets.

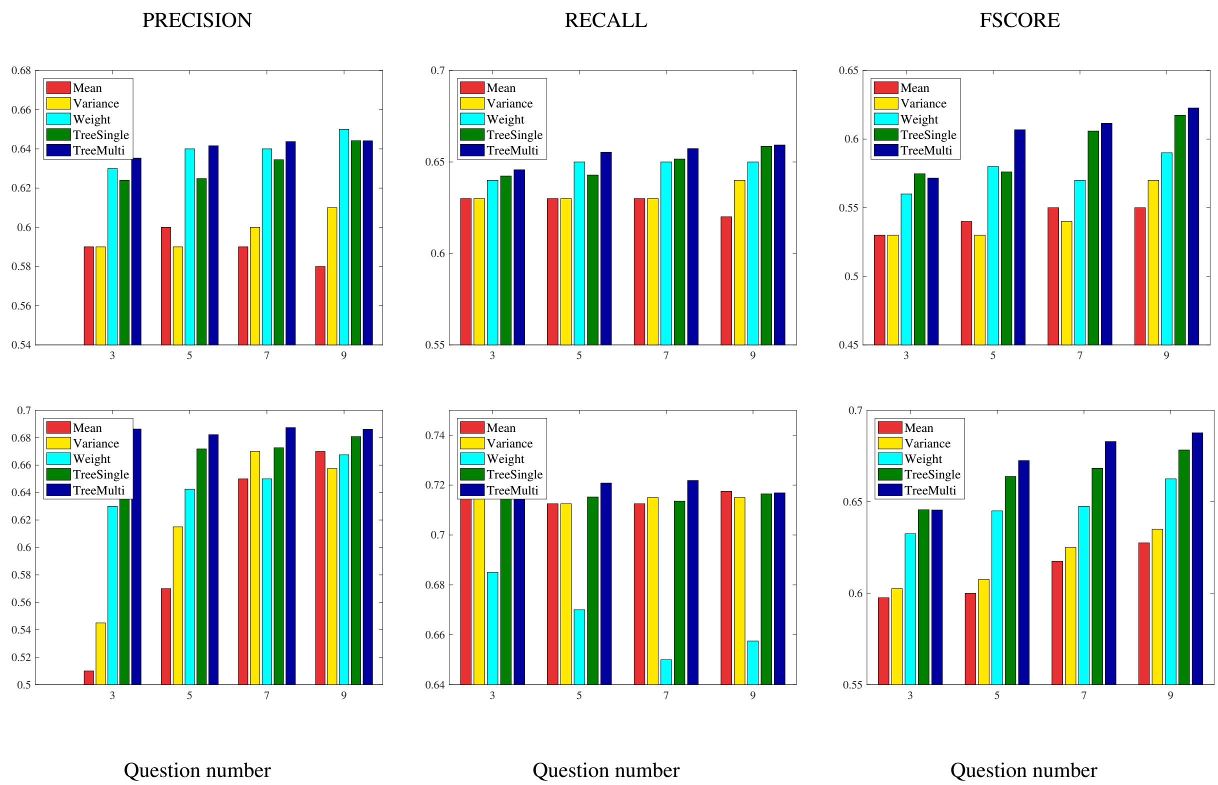

Figure 3.3: The gender prediction metrics precision, recall and fscore with respect to number of questions on datasets Flixster (top row) and MovieLens (bottom row). For all methods, the prediction accuracy increases as the number of questions increases. It shows that our method "TreeMulti" performs better than baselines for both datasets.

3.2.4 Recommendation Accuracy

We now evaluate our multi-task model in terms of rating prediction to answer the second question in Section 3.2.1. We compare our model with two state-of-the-art baselines. One is the bootstrapping tree model [10], denoted as "TreeMean", which predicts user-item ratings using the mean ratings at each node. Since the model does not learn latent user and item profiles, it needs to estimate the ratings for all the n items in the system at each leaf node. The other is the strongest decision tree with matrix factorization, denoted as "fMF" [39]. The model estimates user/item profiles as latent factors and learn the profiles through matrix factorization. Our proposed algorithm differs from others in that it integrates both rating and user demographics through shared user profile learning, and thus enhances prediction accuracy.

We use the following parameter settings: for all three types of trees, we set the same maximum depth and regularization parameter $\lambda = 0.01$ for user and item profiles. We apply 5-fold cross validation to determine other parameters such as latent dimensions. The results on MovieLens, Flixster and Book-Crossing datasets are reported in Table 3.3. First of all, for all three models, the performance improves (RMSE error decreases) as the number of query items increases. The three algorithms generally are capable of refining user preference via adaptive rating elicitation for tackling cold-start problems. Comparing the performance of all models, we can see that our "TreeMulti" model consistently outperforms others in all the three datasets. The improvements over model "fMF" are significant according to t-test with significance level $p = 0.05$.

Table 3.3: Rating prediction error (RMSE) for cold-start users with respect to the number of query items on Datasets MovieLens, Flixster and Book-Crossing.

Data \Method		$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$
Movie	TreeMulti	0.9247	0.9236	0.9226	0.9209	0.9212
	fMF	0.9310	0.9302	0.9282	0.9264	0.9241
	TreeMean	0.9447	0.9364	0.9320	0.9305	0.9302
Flixster	TreeMulti	0.8954	0.8946	0.8940	0.8939	0.8934
	fMF	0.9067	0.9050	0.9049	0.9048	0.9048
	TreeMean	0.9091	0.9089	0.9087	0.9085	0.9084
Book	TreeMulti	1.3462	1.3414	1.3383	1.3364	1.3356
	fMF	1.4094	1.4097	1.4040	1.4007	1.3938
	TreeMean	1.4865	1.4658	1.4605	1.4594	1.4590

In Tables 3.4 and 3.5, we present the query questions in user sessions using MovieLens dataset as well as the top-5 recommendations for them after the sessions. We can see that the recommended movies are quite related to the movies that the users liked based on their genres. In addition, the users who like romance or family movies more than drama or action movies are likely to be female. Those results illustrate that the elicitation process is reasonable.

3.2.5 Tradeoff between Privacy and Personalization

Experiments on three datasets MovieLens, Flixster and Book-Crossing show that our proposed method is sufficient to predict new user demographics using labelled training data from users who share private information. In particular, a recommender system can infer a new user's gender with 69% accuracy using as few as 10 selected queries. The result is promising given the fact that the reported gender accuracy for users with full rating history is 80% [36]. Similarly, the prediction error of a new user's age is smaller than 5 years in best-case scenario with no more than 12 queries using Flixster dataset. In general, the prediction accuracy depends on the rating density of training data. In comparison with MovieLens and Flixster, the prediction error of Book-Crossing user demographics is lower since the rating density of the training data is only round 0.2%.

In general, the more a user interacts with a recommender system, the more privacy threats the user is exposed to. However, the user will also gain more from personalized service. The experiment results from our proposed method show that favorable tradeoff for new users can be established. For example, as illustrated in Figure 3.3 and Table 3.3,

Table 3.4: Examples of rating querying using MovieLens. The predicted gender for the case is Female.

No. Query Items	Query	Response
1	Terminator 2	Unknown
2	Sense and Sensibility	Like
3	Groundhog Day	Like

Rank	Movie Title
1	Casablanca
2	The Wrong Trousers
3	Life Is Beautiful
4	Much Ado About Nothing
5	Shakespeare in Love

Table 3.5: Examples of rating querying using MovieLens. The predicted gender for the case is Male.

No. Query Items	Query	Response
1	Terminator 2	Like
2	Dangerous Liaisons	Unknown
3	Independence Day	Like
4	Peter Pan	Dislike

Rank	Movie Title
1	The Matrix
2	Star Wars: Episode IV
3	Raiders of the Lost Ark
4	The Shawshank Redemption
5	Die Hard

a MovieLens user may choose to answer the first 3 questions and withhold the answers to the rest questions. As a result, the gender prediction accuracy will decrease from 69% to 64% with 7% reduction. Meanwhile, the recommendation error will change from 0.9209 to 0.9236 with only 0.3% increment. The experiments confirm that it is possible for new users to preserve their privacy by not giving answers to certain questions while still benefiting from personalization.

CHAPTER 4

CONCLUSIONS

We proposed a novel and effective method to simultaneously infer private information and enhance user preference prediction for cold-start users, which is critical for recommender systems. Specially, latent user profiles are learned across the tasks of demographic inference and rating prediction simultaneously, which enables knowledge transfer through the two related tasks and improves the prediction accuracy for both tasks. The proposed method can also facilitate the understanding of the tradeoff between user privacy and the utility of personalization. Experimental results on three benchmark datasets including the Flixster dataset, the MovieLens dataset and the Book-Crossing dataset demonstrate the performance of the proposed method in terms of the accuracy of both demographics inference and rating prediction. The work lays a solid foundation for future work of privacy-preserving recommender systems with full user control.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE transactions on knowledge and data engineering*, vol. 17, no. 6, 2005, pp. 734–749.

[2] D. Agarwal and B. Chen, "Regression-based latent factor models," *Proc. of the ACM SIGKDD*, 2009, pp. 19–28.

[3] N. F. Awad and M. Krishnan, "The personalization privacy paradox: an empirical evaluation of information transparency and the willingness to be profiled online for personalization," *MIS quarterly*, 2006, pp. 13–28.

[4] S. Bhagat, U. Weinsberg, S. Ioannidis, and N. Taft, "Recommending with an agenda: Active learning of private attributes using matrix factorization," *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 2014, pp. 65–72.

[5] B. Bi, M. Shokouhi, M. Kosinski, and T. Graepel, "Inferring the demographics of search users: social data meets search queries," *Proc. of the International Conference on World Wide Web*. ACM, 2013, pp. 131–140.

[6] C. Boutilier, R. Zemel, and B. Marlin, "Active collaborative filtering," *Proc. of the 19th Conference on Uncertainty in Artificial Intelligence*, 2003, pp. 98–106.

[7] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1998, pp. 43–52.

[8] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*, Wadsworth and Brooks, Monterey, CA, 1984.

[9] J. Calandrino, A. Kilzer, A. Narayanan, E. Felten, and V. Shmatikov, ""You Might Also Like:" Privacy Risks of Collaborative Filtering," *2011 IEEE Symposium on Security and Privacy*. IEEE, 2011, pp. 231–246.

[10] N. Golbandi, Y. Koren, and R. Lempel, "Adaptive bootstrapping of recommender systems using decision trees," *Proc. of the 4th ACM International Conference on Web Search and Data Mining*, 2011, pp. 595–604.

[11] A. Harpale and Y. Yang, "Personalized active learning for collaborative filtering," *Proc. of the ACM SIGIR Conference*, 2008, pp. 91–98.

[12] L. He, N. Liu, and Q. Yang, "Active Dual Collaborative Filtering with both Item and Attribute Feedback," *Proc. of AAAI*, 2011.

[13] R. Jin and L. Si, "A Bayesian approach toward active learning for collaborative filtering," *Proc. of the 20th Conference on Uncertainty in Artificial Intelligence*, 2004, pp. 278–285.

[14] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Transactions on Knowledge Discovery from Data*, vol. 4, no. 1, 2010, pp. 1–24.

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

[16] M. Kosinski, D. Stillwell, and T. Graepel, "Private traits and attributes are predictable from digital records of human behavior," *Proceedings of the National Academy of Sciences*, vol. 110, no. 15, 2013, pp. 5802–5805.

[17] B. Lakshminarayanan, G. Bouchard, and C. Archambeau, "Robust Bayesian Matrix Factorisation," *Proc. of the International Conference on Artificial Intelligence and Statistics*, 2011.

[18] N. D. Lawrence and R. Urtasun, "Non-linear matrix factorization with gaussian processes," *Proc. of the ICML*, 2009.

[19] J. Lee, *Local approaches for collaborative filtering*, doctoral dissertation, Georgia Institute of Technology, 2015.

[20] Y. Li, J. Hu, C. Zhai, and Y. Chen, "Improving one-class collaborative filtering by incorporating rich user information," *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 2010, pp. 959–968.

[21] A. Mislove, B. Viswanath, K. Gummadi, and P. Druschel, "You are who you know: inferring user profiles in online social networks," *Proc. of the ACM International Conference on Web Search and Data Mining*. ACM, 2010, pp. 251–260.

[22] Y. Moshfeghi, B. Piwowarski, and J. M. Jose, "Handling data sparsity in collaborative filtering using emotion and semantic based features," *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011, pp. 625–634.

[23] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," *2008 IEEE Symposium on Security and Privacy*. IEEE, 2008, pp. 111–125.

[24] M. Pennacchiotti and A. Popescu, "Democrats, republicans and starbucks afficionados: user classification in twitter," *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2011, pp. 430–438.

[25] D. Rao, D. Yarowsky, A. Shreevats, and M. Gupta, "Classifying latent user attributes in twitter," *Proceedings of the 2nd international workshop on Search and mining user-generated contents*. ACM, 2010, pp. 37–44.

[26] A. Rashid, I. Albert, D. Cosley, S. Lam, S. McNee, J. Konstan, and J. Riedl, “Getting to know you: learning new user preferences in recommender systems,” *Proc. of the 7th International Conference on Intelligent User Interfaces*, 2002, pp. 127–134.

[27] A. Rashid, G. Karypis, and J. Riedl, “Learning preferences of new users in recommender systems: an information theoretic approach,” *SIGKDD Workshop on Web Mining and Web Usage Analysis*, 2008.

[28] P. Resnick and H. R. Varian, “Recommender systems,” *Communications of the ACM*, vol. 40, no. 3, 1997, pp. 56–58.

[29] F. Ricci, L. Rokach, and B. Shapira, *Introduction to recommender systems handbook*, Springer, 2011.

[30] R. Salakhutdinov and A. Mnih, “Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo,” *Proc. of the International Conference on Machine Learning*, 2008.

[31] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, “Methods and metrics for cold-start recommendations,” *Proc. of the ACM SIGIR Conference*, 2002, pp. 253–260.

[32] Y. Shi, M. Larson, and A. Hanjalic, “Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges,” *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, 2014, p. 3.

[33] M. Sun, F. Li, J. Lee, K. Zhou, G. Lebanon, and H. Zha, “Learning Multiple-question Decision Trees for Cold-start Recommendation,” *Proc. of Conference on Web Search and Data Mining*, 2013.

[34] M. Vozalis and K. G. Margaritis, “Collaborative filtering enhanced by demographic correlation,” *AIAI Symposium on Professional Practice in AI, of the 18th world Computer Congress*, 2004.

[35] Y. Wang, S. C.-F. Chan, and G. Ngai, “Applicability of demographic recommender system to tourist attractions: a case study on trip advisor,” *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 03*. IEEE Computer Society, 2012, pp. 97–101.

[36] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft, “Blurme: inferring and obfuscating user gender based on ratings,” *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 2012, pp. 195–202.

[37] E. Zheleva and L. Getoor, “To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles,” *Proc. of the International Conference on World Wide Web*. ACM, 2009, pp. 531–540.

[38] Y. Zhong, N. Yuan, W. Zhong, F. Zhang, and X. Xie, "You are where you go: Inferring demographic attributes from location check-ins," *Proc. of the ACM International Conference on Web Search and Data Mining*. ACM, 2015, pp. 295–304.

[39] K. Zhou, S. Yang, and H. Zha, "Functional matrix factorizations for cold-start recommendation," *Proc. of the ACM SIGIR Conference*, 2011, pp. 315–324.

VITA

Changbin Li was born on January, 1990, in Shandong, China. He holds a Bachelor in Communication Engineering from Beijing University of Chemical Technology, and a Master in Biomedical Engineering from Beijing Jiaotong University, China. Thereafter, he worked in Beijing as an Algorithm Engineer. In the spring of 2016, he was accepted to the graduate program in computer science at Louisiana State University. He worked as a teaching assistant to Dr. Mingxuan Sun while working toward the masters degree in computer science. His primary interests include machine learning and recommender system.