Inferring Private Demographics of New Users in Recommender Systems

Mingxuan Sun
Louisiana State University
msun@csc.lsu.edu

Changbin Li
Louisiana State University
cli45@lsu.edu

Hongyuan Zha
Georgia Institute of Technology
zha@cc.gatech.edu

ABSTRACT

With the growing number of wireless and mobile devices ingrained into our daily lives, more and more people are interacting with online services that adopt recommender systems to suggest movies, news and points of interest. The private demographics of users such as age and gender in online recommender systems are very useful for many applications such as personalized ads, social study and marketing. However, users do not always provide details in their online profiles due to privacy concern. Most existing approaches can infer user private attributes based on sufficient interaction history but could fail for new users with few ratings. In this paper, we present a novel preference elicitation method, with which a recommender system asks cold-start users to rate selected items adaptively and infer the demographics rapidly via a few interactions. Specifically, 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 real-world datasets demonstrate the performance of the proposed method in terms of the accuracy of both demographics inference and rating prediction.

CCS CONCEPTS

• **Information systems** → **Personalization**; • **Security and privacy** → *Web application security*;

KEYWORDS

Demographic inference, user modeling, recommender systems

1 INTRODUCTION

A user browses news at an airport, watches a movie in a cafe, and looks for a restaurant or a tourist attraction in a city. Mobile devices coupled with recommender systems have emerged as key tools for personalized information access and have enabled significant business applications such as mobile tourism [2, 4, 27, 30]. User demographics such as their age, gender and ethnicity information can improve recommendations and enable other richer services such as targeted advertisement and marketing. A recommender system may explicitly solicit user demographics through user registration. However, online users do not always 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 deanonymization of Netflix Prize dataset [19] that link private Netflix rating data with public databases such as IMDB to partially infer some user identities. Other attempts [29] suggest that it is possible to infer user gender with as high as 80% accuracy given sufficient user ratings in recommender systems.

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". For example, Netflix movie dataset contains 100 million movie ratings from 480189 users over 17770 movies and most users typically rate only a small number of movies. Since most existing inference approaches largely depend on sufficient interaction history, they could fail for new users with few interactions [10, 23, 26].

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?". Based on the answers, the users are then directed to one of the subtrees each of which is associated with another question. The system gradually refines the estimation of user profiles 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.

In this paper, 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. 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

MSWiM '17, November 21–25, 2017, Miami, FL, USA

ACM ISBN 978-1-4503-5162-1/17/11…$15.00
https://doi.org/10.1145/3127540.3127566

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 Bookcrossing 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. The former is captured by the prediction accuracy of demographics and the latter is captured by the recommendation accuracy.

Our contributions are two folds: (1) we propose 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. (2) The proposed method can help new users to preserve their privacy by not giving answers to certain questions while enjoying some benefits from personalization.

2 RELATED WORK

There has been extensive studies on demographic inference from various online human activities. Studies including [18] 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 [15, 31], search queries [6], linguistic features of tweets [20, 22], and location check-ins [32].

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 [5, 29] 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 [5, 9, 21, 29] also explore different ways to perturb user generated contents such as ratings and locations 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. Several studies [23, 24] focus on preference elicitation to improve rating prediction accuracy through an interview process using a static set of questions. Approaches such as [10, 24, 28, 33] explore variations of decision tree models to adaptively select items to query. Active learning methods [5, 7, 11–13] select questions to query users adaptively. However, these methods usually involve computationally expensive optimization procedures, which are not feasible for online user interactions. Our work focuses on an effective and efficient cold-start recommendation method for both demographic inference and item recommendation.

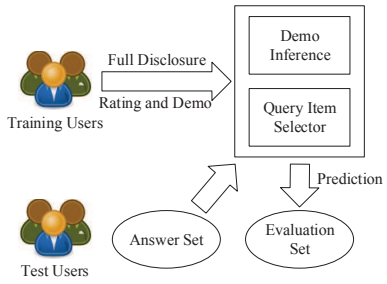

Figure 1: Evaluation framework for cold-start users.

Matrix factorization methods [14, 16, 17, 25] popularized by the Netflix competition winner [14] have been used to predict user ratings. Those methods seek to map users and items in a low-dimensional space to capture the intrinsic similarities. Content-based models such as [1, 26] utilize item features such as movie genre, directors and actors for cold-start recommendations. Our model is solely based on ratings and we adopt the latent factor based method for rating and demographic prediction.

3 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. Our goal is to learn latent user profiles to best estimate both demographics and ratings. As described in Figure 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.

3.1 Simultaneous Rating Prediction and Demographic Inference

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 be the response of user i, which is an n-dimensional vector. Let T denote the function mapping the user response x_i to the user profile that is $u_i = T(x_i)$. We also assume that there exists 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:

$$p(r_{ij}|u_i, v_j, \sigma^2) = \mathcal{N}(v_j^\top u_i, \sigma^2). \tag{1}$$

Similarly, we can assume priors 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

$$y_i \in p(y_i|\theta^\top u_i), \tag{2}$$

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:

$$\begin{aligned}\min_{T,V,\theta} \lambda_r \sum_{(i,j)\in O} \ell_r(r_{ij}, T(x_i)^\top v_j) + \lambda_s \sum_{i\in S} \ell_s(y_i, T(x_i)^\top \theta) \\ + \lambda_v ||V||^2 + \lambda_\theta ||\theta||^2,\end{aligned} \tag{3}$$

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

$$\ell(y, \hat{y}) = (y - \hat{y})^2. \tag{4}$$

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

$$\ell(y, \hat{y}) = (1 - y\hat{y})^2. \tag{5}$$

3.2 Alternative Optimization

The parameters in the objective function defined in equation (3) are learned through an alternative optimization following the two steps:

(1) Given item profile V and regressor θ, a decision tree T is learned such that:

$$\min_T \lambda_r \sum_{(i,j)\in O} \ell_r(r_{ij}, T(x_i)^\top v_j) + \lambda_s \sum_{i\in S} \ell_s(y_i, T(x_i)^\top \theta). \tag{6}$$

(2) Given $T(x)$, variables v_j and θ are learned such that

$$\min_V \sum_{(i,j)\in O} \ell_r(r_{ij}, T(x_i)^\top v_j) + \lambda_v ||V||^2, \tag{7}$$

$$\min_\theta \sum_{i\in S} \ell_s(y_i, T(x_i)^\top \theta) + \lambda_\theta ||\theta||^2. \tag{8}$$

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 [17] using training data. Given the decision tree T, a closed-form solution for the item profiles v_j ($j = 1, 2, \ldots, n$) exists:

$$v_j = \left(\sum_{(i,j)\in O} T(x_i) T(x_i)^\top + \lambda_v I \right)^{-1} \left(\sum_{(i,j)\in O} r_{ij} T(x_i) \right). \tag{9}$$

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:

$$\triangle\theta = \sum_{i\in S} \ell_s'(y_i, \hat{y_i}) T(x_i) + \lambda_\theta w, \tag{10}$$

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 is very 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 (6). We propose an efficient greedy algorithm to find an approximation.

3.3 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 (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 decision tree, the best set of questions are learned by optimizing the objective defined in equation (6). Users are then split into three subsets L, D, and U according to the responses to those questions. The procedure is recursive until the decision tree grows to a certain depth. Starting from the root, given an item j to query, users are divided to 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 [28]. 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, or group U when none of the questions are answered. To find the weight vector w that leads to the best split, we minimize the following function:

$$\min_{w} \lambda_r \sum_{i\in L(w)} \sum_{(i,j)\in O} \ell_r(r_{ij}, u_L^\top v_j) + \lambda_s \sum_{i\in L(w)\cap S} \ell_s(y_i, u_L^\top \theta) + \lambda_r \sum_{i\in D(w)} \sum_{(i,j)\in O} \ell_r(r_{ij}, u_D^\top v_j) + \lambda_s \sum_{i\in D(w)\cap S} \ell_s(y_i, u_D^\top \theta) + \lambda_r \sum_{i\in U(w)} \sum_{(i,j)\in O} \ell_r(r_{ij}, u_U^\top v_j) + \lambda_s \sum_{i\in U(w)\cap S} \ell_s(y_i, u_U^\top \theta)$$
$$\text{s.t. } \|w\|_0 \le l, \tag{11}$$

where u_L, u_D and u_U are the optimal user profiles at child nodes L, D and U. In addition, $\|w\|_0$ is the number of non-zeros and the constraint $\|w\|_0 \le l$ determines that the number should be no greater than l. 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 [28].

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:

$$u_L = \arg\min_{u} \lambda_r \sum_{i\in L(w)} \sum_{(i,j)\in O} \ell_r(r_{ij}, u_L^\top v_j) + \lambda_s \sum_{i\in L(w)\cap S} \ell_s(y_i, u_L^\top \theta) + \lambda_u ||u - u_p||^2, \tag{12}$$

where λ_u is a regularization parameter for user profile u at the current node so that the profile is regularized towards u_p to avoid overfitting. The user profile u can be generally solved through gradient decent and updated as $u = u - \delta\triangle u$ where δ is the learning rate and $\triangle u$ is:

$$\triangle u = \lambda_r \sum_{i\in L(w)} \sum_{(i,j)\in O} \ell_r'(r_{ij}, \hat{r_{ij}}) v_j + \lambda_s \sum_{i\in L(w)\cap S} \ell_s'(y_i, \hat{y_i})\theta + \lambda_u(u - u_p), \tag{13}$$

where $\hat{r_{ij}} = u^\top v_j$ and $\hat{y_i} = u^\top \theta$ are previous estimations.

Specifically, for predicting both ratings and demographics such as age, we adopt least mean square error (L_2) as the loss function. In such cases, the user profile has a closed-form solution:

$$u_L = \left(\lambda_r \sum_{i\in L(w)} \sum_{(i,j)\in O} v_j v_j^\top + \lambda_s \sum_{i\in L(w)\cap S} \theta\theta^\top + \lambda_u I\right)^{-1} \left(\lambda_r \sum_{i\in L(w)} \sum_{(i,j)\in O} r_{ij} v_j + \lambda_s \sum_{i\in L(w)\cap S} y_i\theta + \lambda_u u_p\right). \tag{14}$$

The profiles u_D and u_U for the other two children can be computed in a similar way. In summary, we iterate over possible items and select the best one according to (11) for single-item split at each node. While for multi-item split, we alternatively optimize (11) using techniques as suggested in [28] and (12) until convergence. After the current node is constructed, we recursively construct its child nodes in a similar way.

3.4 Computational Complexity

We summarize the algorithm 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 inversing a square matrix of size k. There are totally n possible splits since we consider one item to query at each node. Using a similar analysis in [33], 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 ratings of user i and m is the number of users. The complexity for building the whole tree is thus $O(d\sum_{i=1}^{m} |O_i|^2 + \beta nk^3 + \beta n^2k^2)$, where d is the depth of the tree and β is the total number of nodes in the decision tree. Usually, 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 equation (9) is $O(nk^3 + n|O^j|k^2)$ 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 (4) and (5). The alternative optimization usually converges in a few iterations.

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.
1: Initialize v_j ($j = 1, 2, \ldots, n$) using [17].
2: Initialize θ randomly.
3: **while** not converge **do**
4: Learn a decision tree T as in Algorithm 2.
5: Update v_j by Equation (9).
6: Update θ by Equation (10).
7: **end while**
8: **return** T, v_j ($j = 1, 2, \ldots, n$) and θ.

Algorithm 2 Greedy Tree Construction

1: **function** FitTree(AtNode)
2: Compute u_L, u_D and u_U using Equation (12).
3: Find the best split item or item set in Equation (11) using [28].
4: Split users into three groups $L(w)$, $D(w)$ and $U(w)$.
5: **if** square error reduces after split **and** depth < maxDepth **then**
6: call FitTree($L(w)$), FitTree($D(w)$) and FitTree($U(w)$) to construct subtrees.
7: **end if**
8: **return** T with $T(x)$
9: **end function**

4 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. The estimation framework is examined on three movie recommendation datasets: MovieLens, Flixster and Bookcrossing.

4.1 Experiment Setting

We evaluate the performance of our multi-task model in a cold-start setting. For each dataset, users are randomly split into a training set and a test set with 80%/20% ratio, respectively. The users in the training set are assumed to be warm-start users and their ratings and demographic information are visible to the system. Our model is learned and the set of items are constructed as the probing questions based on training data. In contrast, the users in the test set are assumed to be cold-start users. Their ratings are further split into two disjoint sets: answer and evaluation sets that contain 80% and 20% ratings. We use the answer set to simulate cold-start user responses in the rating elicitation process. We use the evaluation set to evaluate the rating prediction accuracy for withheld items. Meanwhile, the demographic information of each user in the test set can be used to evaluate the demographic prediction accuracy. The evaluation process is summarized in Figure 1. In the rating elicitation process, we select items to query user responses. For simplicity, we ask for user binary responses and the question is in the form "Do you like movie *50 first date*?" Following the classic settings [10, 24, 33], we 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?

4.2 Dataset and Evaluation Metrics

The MovieLens dataset 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 Bookcrossing data 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 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 Bookcrossing dataset has only continuous age labels. The mean age for Flixster and Bookcrossing are 24 and 35, respectively. In our experiments, we choose age regression tasks using Flixster and Bookcrossing 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 shown in Table 1.

Table 1: Dataset description.

Dataset	Users	Items	Ratings	Gender	Age
MovieLens	6,040	3,952	1,000,209	71%/29%	NA
Flixster	23,488	5,795	5,625,681	43%/57%	24
Book	5,411	34,963	384,888	NA	35

The rating prediction performance is evaluated with the root mean square error (RMSE). For age regression, we use standard mean absolute error (MAE), rooted mean squared error (RMSE) and mean absolute percentage error (MAPE). 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. The MAPE is more readable across different datasets. For gender classification, we use precision, recall and fscore to measure the performance for imbalanced binary classes.

4.3 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 Section 4.1. 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 [24]. The third one "Weight" [29] 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 2 compares the prediction accuracy of our model with baselines for age regression on two datasets. The first row of Figure 2 compares the performance on dataset Flixster. For all models, the age prediction error measured in MAE and RMSE decreases when the number of questions increases. 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

Table 2: Rating prediction error (RMSE) for cold-start users with respect to the number of query items on Datasets MovieLens, Flixster and Bookcrossing.

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

Table 3: 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

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 2, we compare performance on dataset Bookcrossing and we see similar trends. Overall, the dataset Bookcrossing 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 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 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 compares gender classification performance on dataset MovieLens. Overall, the performance on MovieLens is better than Flixster. Our model "TreeMulti" performs better than others.

4.4 Recommendation Accuracy

We now evaluate our multi-task model in terms of rating prediction to answer the second question in Section 4.1. 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. The other is the strongest decision tree with matrix factorization, denoted as "fMF" [33]. 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.

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 Bookcrossing datasets are reported in Table 2. First of all, for all three models, the performance improves as the number of questions increases. The three algorithms generally are capable of refining user preference via adaptive rating elicitation for tackling cold-start problems. For all models, we can see that our "TreeMulti" model consistently outperforms others in all the three datasets.

In Tables 3 and 4, 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.

4.5 Tradeoff between Privacy and Personalization

Experiments on three datasets MovieLens, Flixster and Bookcrossing show that our proposed method is sufficient to predict new user demographics using labeled 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% [29]. In addition, the prediction error of a new user's age in Flixster is smaller than 5 years in best-case scenario with no more than 12 queries. In general, the prediction accuracy depends on the rating density of training data. In comparison with MovieLens and Flixster, the prediction error of Bookcrossing 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 and Table 2, 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.

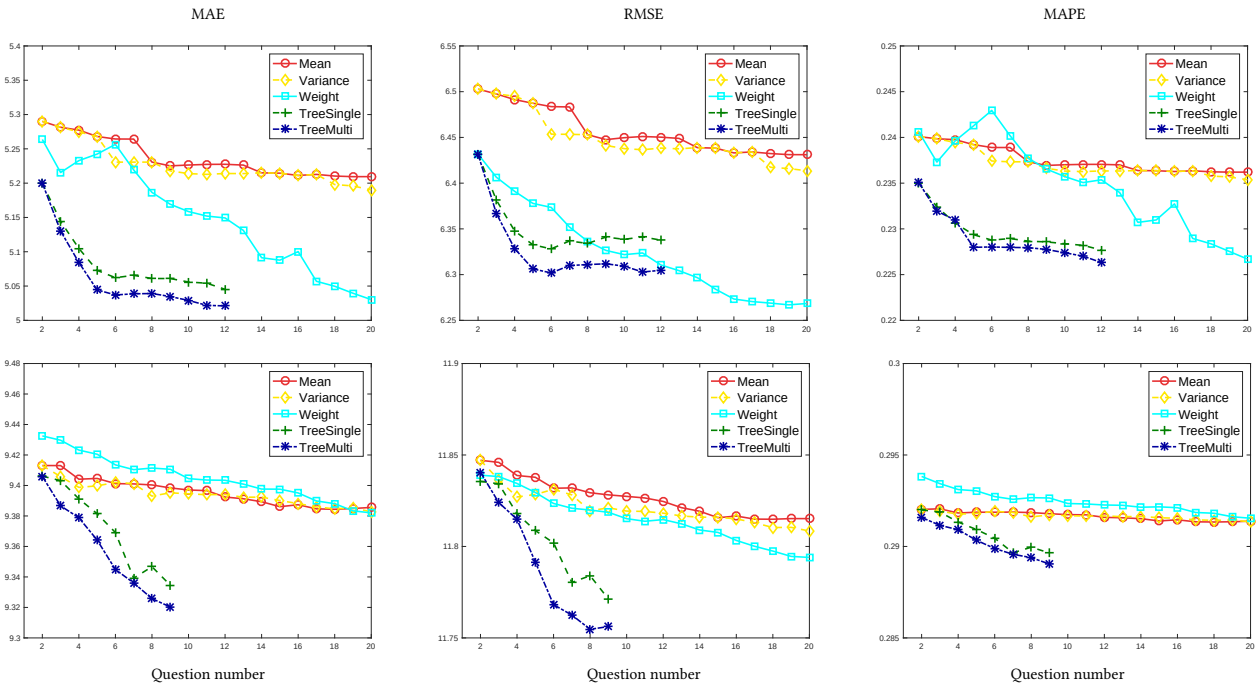

Figure 2: The age prediction metrics MAE and RMSE with respect to number of questions on two datasets Flixster (top row) and Bookcrossing (bottom row). For all models, the prediction error decreases as the number of questions increases. It shows that our methods "TreeMulti" performs better than baselines for both datasets.

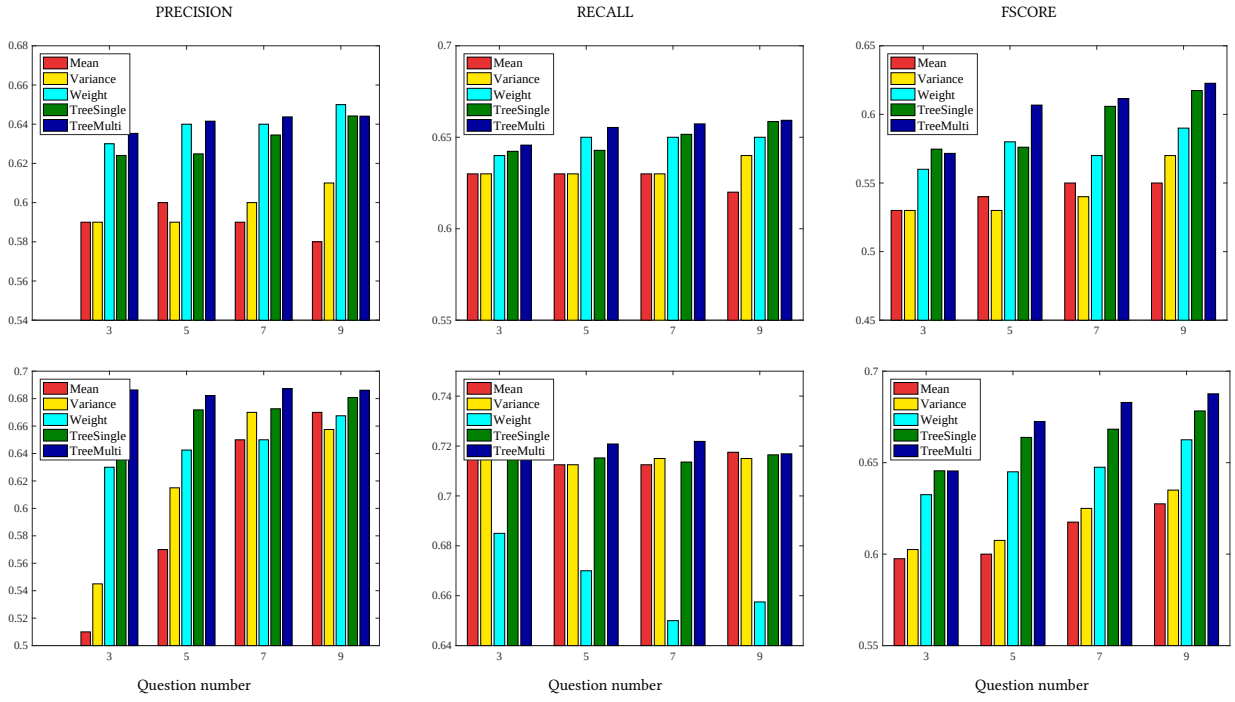

Figure 3: The gender prediction metrics precision, recall and fscore with respect to number of questions on datasets Flixster (top row) and MovieLens (bottom row). For all methods, the prediction accuracy increases as the number of questions increases. It shows that our method "TreeMulti" performs better than baselines for both datasets.

Table 4: 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

5 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. Experimental results on three benchmark datasets including the Flixster dataset, the MovieLens dataset and the Bookcrossing dataset demonstrate that the proposed method outperforms existing ones. The proposed method can help new users to preserve their privacy by not giving answers to certain questions while enjoying some benefits from personalization. We further discuss the tradeoff between user privacy and the utility of personalization, which lays a solid foundation for future work of privacy-preserving recommender systems with full user control.

6 ACKNOWLEDGEMENT

We would like to thank the reviewers for their constructive and insightful comments. This work was supported in part by the Louisiana Board of Regents under Grant LEQSF(2017-20)-RD-A-29.

REFERENCES

[1] D. Agarwal and B.C. Chen. 2009. Regression-based latent factor models. In *Proc. of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 19–28.

[2] O. Averjanova, F. Ricci, and Q. N. Nguyen. 2008. Map-based interaction with a conversational mobile recommender system. In *Proc. of the Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*. 212–218.

[3] N. F. Awad and M.S. Krishnan. 2006. The personalization privacy paradox: an empirical evaluation of information transparency and the willingness to be profiled online for personalization. *MIS quarterly* 30, 1 (2006), 13–28.

[4] L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci. 2012. Context relevance assessment and exploitation in mobile recommender systems. *Personal and Ubiquitous Computing* 16, 5 (2012), 507–526.

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

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

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

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

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

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

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

[12] L. He, N.N. Liu, and Q. Yang. 2011. Active dual collaborative filtering with both item and attribute feedback. In *Proc. of the 25th AAAI Conference on Artificial Intelligence*.

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

[14] Y. Koren. 2010. Factor in the neighbors: scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data* 4, 1 (2010), 1–24.

[15] M. Kosinski, D. Stillwell, and T. Graepel. 2013. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences* 110, 15 (2013), 5802–5805.

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

[17] N. D. Lawrence and R. Urtasun. 2009. Non-linear matrix factorization with gaussian processes. In *Proc. of the International Conference on Machine Learning*.

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

[19] A. Narayanan and V. Shmatikov. 2008. Robust de-anonymization of large sparse datasets. In *Proc. of the IEEE Symposium on Security and Privacy*. 111–125.

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

[21] K. P. Puttaswamy and B. Y. Zhao. 2010. Preserving privacy in location-based mobile social applications. In *Proc. of the 11th Workshop on Mobile Computing Systems & Applications*. 1–6.

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

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

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

[25] R. Salakhutdinov and A. Mnih. 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proc. of the International Conference on Machine Learning*.

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

[27] M. Strobbe, O. Van Laere, S. Dauwe, B. Dhoedt, F. De Turck, P. Demeester, C. van Nimwegen, and J. Vanattenhoven. 2010. Interest based selection of user generated content for rich communication services. *Journal of Network and Computer Applications* 33, 2 (2010), 84–97.

[28] M. Sun, F. Li, J. Lee, K. Zhou, G. Lebanon, and H. Zha. 2013. Learning multiple-question decision trees for cold-start recommendation. In *Proc. of Conference on Web Search and Data Mining*.

[29] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft. 2012. Blurme: inferring and obfuscating user gender based on ratings. In *Proc. of the 6th ACM conference on Recommender systems*. 195–202.

[30] B. Zenker and B. Ludwig. 2009. ROSE: assisting pedestrians to find preferred events and comfortable public transport connections. In *Proc. of the 6th International Conference on Mobile Technology, Application & Systems*. 16.

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

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

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