Received November 5, 2016, accepted November 17, 2016, date of publication November 29, 2016, date of current version January 4, 2017.

Digital Object Identifier 10.1109/ACCESS.2016.2633282

Joint Prediction of Rating and Popularity for Cold-Start Item by Sentinel User Selection

ZHONGCHEN MIAO[1], JUNCHI YAN[2,3], KAI CHEN[1], XIAOKANG YANG[1], (Senior Member, IEEE), HONGYUAN ZHA[4], AND WENJUN ZHANG[1], (Fellow, IEEE)

[1]Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China
[2]School of Computer Science and Software Engineering, East China Normal University, Shanghai, China
[3]IBM Research-China, Shanghai, China
[4]School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA

Corresponding author: K. Chen (kchen@sjtu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1001003, in part by the National Natural Science Foundation of China under Grant 61602176, Grant 61672231, Grant 61521062, and Grant 61527804, in part by the China Postdoctoral Science Foundation Funded Project under Grant 2016M590337, in part by the Shanghai Science and Technology Committees of Scientific Research Project under Grant 14XD1402100 and Grant 15JC1401700, and in part by the 111 Program under Grant B07022.

ABSTRACT New item or topic profiling and recommendation are useful yet challenging, especially in face of a "cold-start" situation with sparse user-item ratings for the new arrivals. In this paper, a method of acquiring review opinions of the "sentinel" users on the cold-start items is proposed to elicit those items' latent profiles, and thus both user-specific ratings and future popularity of the items can be predicted simultaneously. Specifically, such a joint prediction task is formulated as a two-stage optimization problem, and a sentinel user selection algorithm is devised to facilitate effective latent profiles extraction for both item ratings and popularity predictions. Experiments with microblogging and movie data sets corroborate that the proposed method is capable of mitigating the cold-start problem and it outperforms several competitive peer methods.

INDEX TERMS Recommendation, cold-start, decision tree, matrix factorization, popularity prediction.

I. INTRODUCTION

Everyday, large numbers of new-arrival items (e.g., goods and movies) and emerging topics (e.g., news and memes) are making their debut on the e-commerce websites and online social networks. To catch the customers' attention and retain their interests in using these online services, recommendation systems have been widely adopted to promote the items or topics to users. Ideally, being a user-centered system, the items or topics to be recommended should be personalized for each individual user according to one's preferences, indicated by the shopping and browsing history of the user. Meanwhile, the recommendation engine ought to provide a shortened recommendation list to each user in order to avoid the user being overwhelmed by too many choices.

In the context of a recommendation system for new-arrival items, the challenge lies in two aspects. First, little content information of the new items are known by the system, and they often have sparse user adoptions or ratings. With too many missing values in user-item matrix or content attributes, existing collaborative filtering algorithms or content-based methods can hardly infer the new item profiles and user preferences, which hurts recommendation accuracy. The phenomenon is commonly known as the 'cold-start' problem [1], [2], where the 'cold-start' in this article specifically refers to the new items. Furthermore, consider the fact that the quantity of new-arrival items can be intimidating in practical online systems, it is likely that a user has to be faced with a long list of 'appealing' items whose predicted rating scores are similar to each other.

Therefore, in order to offer a short and distinctive recommendation list of new items for each user in the system, the factors besides user preferences should also be taken into consideration when determining what new items should be placed into the list. Specifically in this article, the potential popularity—the total number of predicted sales (or adoptions, etc.)—of a new item (or topic, etc.) in addition with its user-specific rating scores are considered, in the hope that the user may have better chance to adopt the more popular item within his/her tastes, which may bear higher potential social identification. Akin to the above mentioned item 'cold-start' problem in inferring profiles and ratings for new items, the task of new item popularity prediction also faces the similar challenge, as inadequate user adoptions or content information can hardly provide enough item features for popularity prediction. Please note that the potential popularity are counted not necessarily within one website, rather they can be computed over a much broader scope based on the applications.

In this article, we are aimed to develop a recommendation system such that the recommended items, especially the new ones, should not only fit with users' preferences, but also have high future social popularity, so as to improve the overall performance of the engine. An intuitive solution to the above challenges for new items is acquiring as many extra user-item ratings as possible to mitigate the 'cold-start' situation of the new items. In that case, website owners have to actively invite many customers (perhaps some rewards should also be offered) to try the new items, and then interview these users for their opinions on those items to enrich the rating dataset. Such a trial-and-interview process for new items can be both time and money consuming, which is further pronounced in the existence of redundancy for the data of user-item ratings and user preferences.

One way to alleviate this difficulty is to choose (a relatively small number of) representative users with a principled and well-designed method for their ratings collection under the limited rating acquisition budget. Rather than exhaustively or randomly inviting users, we present a method for selecting a small-sized set of representative users, whose review opinions on new items are exploited to elicit the item profiles. The user selection process for a single cold-start item runs in a sequential manner. In general, the pipeline runs as follows: i) at first, one user will be invited to try the new item and give his/her review opinion; ii) then to enhance user preference diversity, a second user with a different preference is dynamically selected and invited for rating on the same item; iii) in a similar manner, a series of users are sequentially selected according to the previously selected users' preferences. In this article, the above group of invited users for one item are called as the 'sentinel' users for that item. As the sentinel users for one new item are dynamically and optimally selected with (hopefully) minimal redundancy, the total size of sentinel users must be smaller than the whole size of users in the system. Thus, those reviewed items can still be regarded as new ones for most of the other customers in the system.

Essentially, the review opinions of the selected sentinel users are eliciting the latent profiles of the new items, which could discriminate one item from others. With the help of the sentinel users, a joint prediction approach to simultaneously predict the reviewed items' ratings by all system users and their potential popularity is proposed, comprising the following two main stages. The first stage involves the proposed algorithm of selecting optimal sentinel users as cold-start item reviewers, so as to factorize latent user and item profiles from user-item matrix. During the matrix factorization, new items can be parameterized with discriminative profiles according to the corresponding sentinel users' latent profiles and their review opinions on those items. In the second stage, the unknown user-item relations are predicted with the factorized profiles, and the item profiles are treated as feature vectors for training popularity regression model. Because latent profile factorization and joint predictions are highly intertwined to each other, iterative optimization of these two stages are performed. In particular, we solve the two-stage optimization and joint prediction task for cold-start items based on our extension to the functional matrix factorization method [3]. So the 'functional' in this article means that the matrix is factorized in a way of restricting item profiles to be the mapping of a series of sentinel users' review opinions on those items, while the mapped item profiles can also be used as feature vectors for item popularity regression.

The core of the proposed joint prediction approach lies in the sentinel user selection step, which can be viewed as the process of building and optimizing a parameterized multi-way decision tree model. In an optimized decision tree model, each tree branch node is associated with a selected sentinel user, and each leaf node represents a cluster of items with similar profiles. As an example, in Fig.1 we illustrate an optimized four-layer, three-way decision tree model for sentinel user selection. Each branch node in Layer 1 through 3 has three sub branches, and each node holds a sentinel user. The leaf nodes in Layer 4 denote the clusters of items. Therefore, for assigning one new item to a proper leaf node (i.e. item cluster), the item has to travel from the tree root and go through one branch node in each layer. Along its path from tree root to a leaf node, the direction after each branch node towards the next layer is decided according to the review opinions on that item, given by the sentinel user whom is associated with the branch node. For the example model shown in Fig.1, a sentinel user's opinion such as likes, uncertain or dislikes on one item will lead the item to the corresponding left, middle or the right sub branch in the next layer, and finally the item will be assigned into a specific leaf node. To build such a decision tree model which allows all new items to be assigned into the correct leaf nodes and joint prediction can be achieved, the associated sentinel users of each branch node should be optimally selected, and their latent profiles should be well parameterized. The details of the decision tree building procedure will be elaborated later.

In our experiments, we show the performance of the proposed approach on two different categories of datasets. The first category involves the MovieLens[1] and Netflix[2] movie rating dataset, where movie ratings are viewed as user-item rating matrix, and real box-office sales of the movies are treated as item popularity. The second category involves a microblogging[3] dataset collected by ourselves, where micro-

[1] See GroupLens Website. http://grouplens.org/datasets/movielens/

[2] See Netflix Prize Website. http://netflixprize.com

[3] The microblogging data is gathered from Weibo. Weibo (NASDAQ: WB) is a famous Chinese microblogging (weibo) website. Akin to a hybrid of Twitter and Facebook, it is one of the most popular social network service in China. http://weibo.com

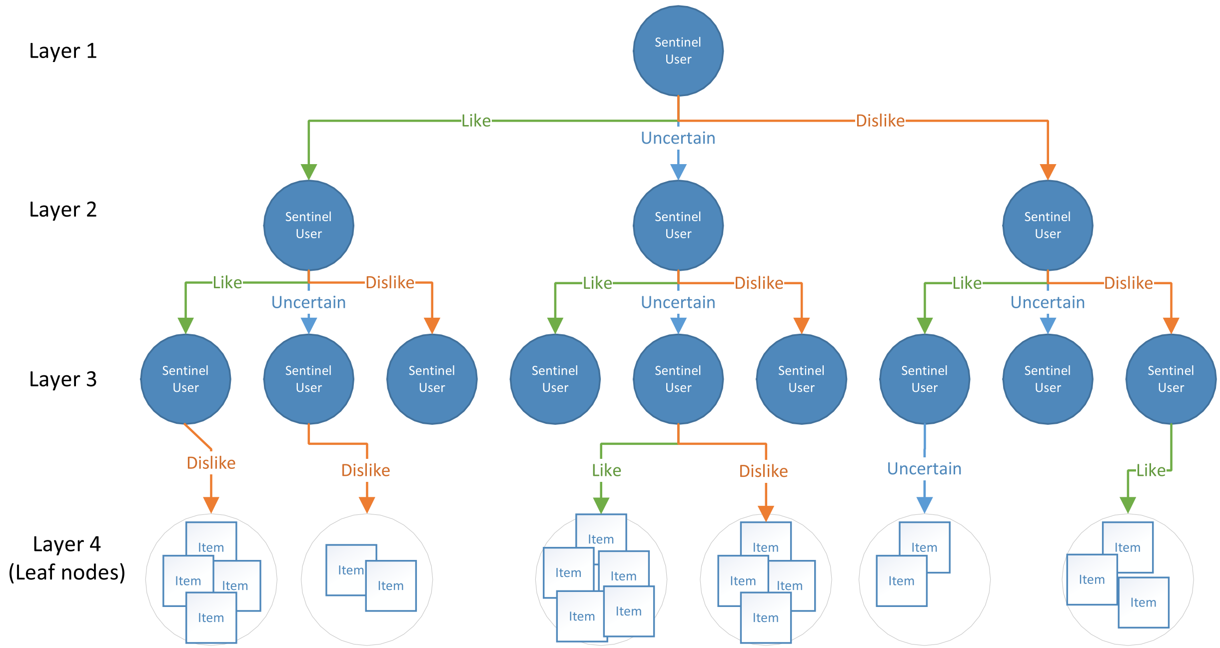


FIGURE 1. A four-layer, three-way decision tree model for selecting sentinel users. Each branch node in the top three layers represents a selected sentinel user, and each leaf node in the bottom layer denotes a cluster of items that have similar profiles (only a small number of leaf nodes is shown in this figure). As sentinel users' review opinions for various items may differ, items may go through different paths and may be assigned into different clusters (leaf nodes).

posts' authors are treated as users and the topics of microposts are regarded as items. Hence the microblog users' participation states in topics are viewed as user-item relation matrix, and participated users count of topics are used as item popularity. By evaluating the prediction accuracy of the trained models on both user-item rating/relation and item future popularity, we show that the proposed two-stage optimization approach involving the sentinel user selection component is effective in joint prediction for cold-start items.

The structure of this article is organized as follows: In Section II, existing works and challenges in collaborative filtering, recommendation and popularity prediction are briefly introduced, especially for cold-start items. In Section III, the goal of joint predictions on cold-start item's popularity and user-item ratings are first introduced, and then the procedure of decision tree model building for sentinel user selection, latent profile extraction and the optimization for joint prediction are described in details. After that, Section IV presents a description of the used dataset. Experimental results are given in Section V with discussions on the results. Section VI concludes this paper.

II. RELATED WORKS

Collaborative filtering (CF) has become a popular method for solving recommendation and matrix completion problems, whereby the existing works can be approximately divided into two streams [4]–[6]: memory-based methods and model-based methods. Memory-based methods often use the entire dataset to calculate similarity in both item and user domain [7], in which Pearson correlation or cosine similarity [8] are commonly used. In general, the calculated similarity can be viewed as a manner of discriminating items and users from others. Model-based methods often use content information together with algorithms such as clustering, Bayesian networks and probabilistic latent variable models [9] to generate models. For example, the work [10] shows that the regressions of latent profile with richer content information is able to achieve promising prediction accuracy, and the works [11]–[13] develop predictive feature-based regression models to tackle cold-start problems in personalized recommendations. Recently, matrix factorization and its improved algorithms have become a popular trend [3], [14]–[16] for CF, as the operations can fill out the unobserved elements of user-item matrix in a way similar to latent factor analysis.

One major challenge for CF is the so-called cold-start problem, i.e. the difficulty of extracting latent profile and computing similarity for new-arrival users or new-arrival items with few user-item ratings. Cross-domain methods [17] are proposed to cope with the cold-start user problem by gathering extra information of new users from other domains. As a comparison, to address the cold-start item problem, gathering more item information and building up user preference elicitation strategy has become one main research line [18]–[20]. The idea of setting up a user interview process to get additional user responses on cold-start items is introduced in [21] and [22], whereby the user selection strategy is static on measurements such as informativeness, popularity and coverage. However the responses of selected users are not considered [23] during user selection. In contrast, information gain through clustered neighbors (IGCN) algorithm [24], which uses pre-defined user clusters, can make adaptive selection according to users response. The idea of fitting a decision tree to get better prediction is proposed in [25], and functional matrix factorization [3] is developed to integrate decision tree building process with matrix factorization. Part of our work is inspired by this idea, and we extend the method and tailor it with our two-stage optimization for joint prediction.

There are also other decision tree based algorithms, such as the fitting algorithms for multiple question interviews [26], the so-called most popular sampling (MPS) method aimed to speed up the decision tree building procedure [27], and the supervised active learning approach [28].

In terms of item popularity analysis in recommendation systems, the research work [29] quantitatively examines the trade-off between item popularity and recommendation accuracy, especially for the long-tailed items with low popularity; the article [30] proposes a popularity-sensitive clustering method to improve the recommendation accuracy for the long-tailed items; paper [31] compares various recommendation systems' accuracy in together with recommended items' diversity metrics such as item popularity; the work [32] proposes cost-sensitive factorization machine to automatically leverage the trade-off between item popularity and recommendation quality; and the method presented in [33] takes into account the item popularity and its temporal dynamics in their recommendation algorithms. For event mining and item popularity prediction on the web, [34] predicts new item's future ratings count using bipartite clustering coefficient in rating networks, [35] and [36] use crowd-sourcing to detect emergency events and their states in social media, and [37] proposes a probit regression mode that maps input features into probabilities to predict the Bayesian click-through rate in search engines.

III. JOINT PREDICTION MODEL AND OPTIMIZATION ALGORITHM

In this section we first introduce the proposed approach for cold-start item joint prediction using sentinel users. The main novelty of this approach is that a small set of selected sentinel users' reviews are exploited by a two-stage optimization mechanism to cost-effectively overcome the cold-start problem. As a result, the prediction for the new-arrival item's user-specific rating and future popularity can be improved. In particular, we will explain the decision tree optimization algorithms that help identify the sentinel users, whose reviews are crucial to the joint prediction performance.

A. ITEM RATINGS PREDICTION AND POPULARITY PREDICTION

Given a user set $\mathcal{N}$ with N users and an item set $\mathcal{M}$ with M items in a system (e.g., an online shopping website), each user $i \in \mathcal{N}$ can give a rating score r_{ij} to an item $j \in \mathcal{M}$ that one may use or participate. Then a rating matrix $R_{N\times M}$ with N rows and M columns can be generated regarding every rating score r_{ij} as an element in R.[4] As hardly one item could be used and rated by all users in $\mathcal{N}$, the unknown or unobserved ratings for the items will be missing elements[5] in R. For each item j, the set of j's observed ratings that contains all non-missing elements in the jth column of matrix R is denoted as $\mathcal{O}_j=\{r_{ij}|r_{ij} \neq null, i \in \mathcal{N}\}$. Then the whole observed rating set for all items in the system can be denoted as $\mathcal{O} = \{\mathcal{O}_1 \cup \mathcal{O}_2 \cup \ldots \cup \mathcal{O}_M\}$.

The task of item rating prediction is to estimate the unobserved ratings (i.e. missing elements in R), so that the unused items can be recommended to the users if the corresponding predicted rating scores are high. Collaborative filtering (CF) algorithms are often used in this situation as there exists an important hypothesis that users with similar shopping records or rating patterns may have similar item preferences, and they are likely to buy or rate the unused items that meet their tastes. Matrix factorization is one important method in collaborative filtering, where user-item rating matrix $R = U^T V$ is factorized as user profile vectors $U_{K\times N}$ and item profile vectors $V_{K\times M}$. Every observed user rating $r_{ij} \in \mathcal{O}$ can then be treated as inner product of both K ($K \leq M$ and $K \leq N$) dimensional user i's profile vector u_i and item j's profile vector v_j. The loss function for estimating observed user ratings using matrix factorization can be calculated as follows:

$$\mathcal{L}_r = \sum_{r_{ij}\in\mathcal{O}, i\in\mathcal{N}, j\in\mathcal{M}} (r_{ij} - u_i^T v_j)^2 \tag{1}$$

To minimize the loss $\mathcal{L}_r$ in user-item ratings, optimal user profile u_i and item profile v_j should be calculated by solving the equation shown in Eq.2. Then the missing elements in R can be approximately estimated using the optimized profile vectors $\hat{r_{ij}} = u_i^T v_j$. The optimization solution for Eq.2 in this article is calculated with alternating minimization.

$$\underset{\{u_i|i\in\mathcal{N}\},\{v_j|j\in\mathcal{M}\}}{\text{argmin}} \sum_{r_{ij}\in\mathcal{O}} (r_{ij} - u_i^T v_j)^2 \tag{2}$$

By matrix factorization, each estimated item profile vector v_j can be regarded as the item feature vector. Accordingly, linear regression methods can be used to predict item future popularity b, where the loss function is given in Eq.3.

$$\mathcal{L}_b = \sum_{j\in\mathcal{M}} (b_j - \omega^T v_j)^2 \tag{3}$$

The linear regression coefficients ω in Eq.3 can then be trained by minimizing the loss $\mathcal{L}_b$ by Eq.4.

$$\underset{\omega}{\text{argmin}} \sum_{j\in\mathcal{M}} (b_j - \omega^T v_j)^2 \tag{4}$$

B. JOINT PREDICTION FOR COLD-START ITEM

For new item $j \in \mathcal{M}$ with many missing values in the rating matrix, common existing collaborative filtering methods and

[4] R can also be viewed as a user-item relation matrix, in which r_{ij} indicates the relation between user i and item j.

[5] Note the missing elements are not equal to zero.

content-based methods have difficulty to find the accurate item profile for j and the correlations between item j and other items, so does the item feature vector for popularity prediction. Ideally, the so-called 'cold-start' problem can be solved with enlarged $\mathcal{O}_j$ by acquiring extra user ratings for item j. Nevertheless, as discussed in Section I, inefficient item rating acquisition methods can be very expensive in both money and time cost.

To reduce the data acquisition cost for solving cold-start problems and to increase the prediction accuracy, we propose a novel mechanism to dynamically select reviewer users and extract the latent profiles of cold-start items from the selected users' review opinions. Concretely, for each cold-start item j, we will adaptively find an optimal set of users who have not rated j, and then invite them to try and give review opinions (such as like, dislike or uncertain) on j. With the optimally selected users, the key features of item j can be captured according to the users' latent profiles and their reviews on j. Then j's latent item profile that can discriminate one item from the others is extracted, which is crucial in item rating and popularity prediction. This group of representative reviewers are called as the selected 'sentinel' users for j: $\mathbf{p}_j = \{p_j^{(1)}, p_j^{(2)}, ..., p_j^{(P)}\}$ with the max number of P users for each cold-start item. Then the sentinel users' ratings/reviews for item j can be denoted as $A_j = \{a_j^{(1)}, a_j^{(2)}, ..., a_j^{(P)}\}$.

As every user among the $\mathcal{N}$ customers can be potentially selected as a sentinel user, it must meet the following requirements to get an effective and efficient interview process:

- The number of selected sentinel users to review for each item should be quite small and no larger than P (P is commonly less than 7 in practice), as more time and money will be needed when more users are involved in the whole review process.
- To fully understand the features of an item, the selected sentinel users should not be overlapping heavily in their preference patterns since P should be kept small. Therefore, the selected users' profiles should be discriminatively extracted, with reduced redundancy in user patterns.
- Extracted item profiles should also be discriminative, so they can be used as feature vectors for item rating and popularity prediction.

In order to satisfy the above rules, we define a mapping function T in Eq. 5 that maps the series of P sentinel user reviews on item j into j's latent item profile vector v_j, ensuring the association of item profile and its sentinel user reviews.

$$v_j \triangleq T(A_j) = T(\{a_j^{(1)}, a_j^{(2)}, ..., a_j^{(P)}\}) \tag{5}$$

Having this mapping function T and sentinel user reviews A_j, item j's profile and system user preferences on that item could be extracted by running matrix factorization on the newly extended observed ratings set $\mathcal{O}_j \leftarrow \mathcal{O}_j \cup A_j$ that combines the original observed ratings and the selected sentinel user ratings.

To perform joint prediction on both item rating and future popularity, the user and item profiles should be parameterized in a way that the rating prediction loss $\mathcal{L}_r$ (by Eq.1) and the popularity prediction loss $\mathcal{L}_b$ (by Eq.3) can be minimized simultaneously. Therefore, we further use Eq.5 and devise the weighted loss by Eq.6 where λ is the weighting parameter. The effect of λ will be discussed later in Section V.

$$\min(\mathcal{L}_r + \lambda \cdot \mathcal{L}_b) = \min\Bigg(\sum_{\substack{r_{ij}\in\mathcal{O}\\ j\in\mathcal{M}}} \big(r_{ij} - u_i^T T(A_j)\big)^2 + \lambda \cdot \sum_{j\in\mathcal{M}} \big(b_j - \omega^T T(A_j)\big)^2\Bigg) \tag{6}$$

To minimize the joint prediction loss and solve the equation formulated in Eq.6, we extend the original functional matrix factorization method [3] by making it suitable for the two-stage optimization tasks: latent profiles extraction and linear feature regression. To be specific, the first stage of the optimization is to select an optimal set of sentinel users according to the latent profiles $U = (u_1, u_2, ..., u_N)$ of the users in $\mathcal{N}$, and then put the selected sentinel users' reviews into rating matrix R to factorize discriminative latent item profiles $V = (v_1, v_2, ..., v_M)$. Then in the second stage, optimizations on user profile U and item popularity regression coefficient w are performed based on the factorized item profile V in the previous stage and the item popularity vector b, leading to the minimization of both predicting error on rating and popularity. The above two optimization stages are highly coupled, so they should run iteratively to update the values of U, V and ω in Eq.6 respectively, where detailed procedures are listed below. For description convenience, regularization coefficient is not listed in the following equations.

1) **Updating item profile vectors** V=$(v_1, v_2, ..., v_M) = \big(T(A_1), T(A_2), ..., T(A_M)\big)$:
Given optimized ω and user profile U that are calculated by previous iteration round (In the first round, these values can be pre-defined or randomly initialized), each item profile v_j ($j \in \mathcal{M}$) should be optimized with Eq.7 in order to get overall minimized joint prediction error shown in Eq.6. As each item j's profile v_j is associated with its selected sentinel users' reviews A_j using mapping function T (shown in Eq.5), optimization of v_j in this step is in fact highly relied on the effect of sentinel users selection algorithm and corresponding mapping function T, which will be explained in detail in Section III-C.

$$\underset{T(\cdot)=V}{\operatorname{argmin}}\Bigg(\sum_{\substack{r_{ij}\in\mathcal{O}\\ j\in\mathcal{M}}} \big(r_{ij} - u_i^T T(A_j)\big)^2 + \lambda \cdot \sum_{j\in\mathcal{M}} \big(b_j - \omega^T T(A_j)\big)^2\Bigg) \tag{7}$$

2) **Updating user profile vectors** U=$(u_1, u_2, ..., u_N)$:

Given optimized $V = T(\cdot)$ in previous step, the optimized user profile for every user $i \in \mathcal{N}$ can be computed using the following closed-form optimization equation Eq.8. In the equation, $T(A_j)^T$ means the transpose vector of $T(A_j)$.

$$u_i = \Big(\sum_{j\in\mathcal{M}} T(A_j)T(A_j)^T\Big)^{-1}\Big(\sum_{\substack{r_{ij}\in\mathcal{O}\\ j\in\mathcal{M}}} r_{ij}T(A_j)\Big) \quad (8)$$

3) **Updating popularity linear regression coefficient** ω: Given optimized $V = T(\cdot)$ in the first step, we can use least square regression to find closed-form optimization results of the linear regression coefficient ω for item future popularity prediction, where the equation is shown below in Eq.9.

$$\omega = \Big(\sum_{j\in\mathcal{M}} T(A_j)T(A_j)^T\Big)^{-1}\Big(\sum_{j\in\mathcal{M}} b_jT(A_j)\Big) \quad (9)$$

The above three optimization steps run iteratively in rounds, so user profile U and regression coefficient ω updated in step 2 and step 3 are back used in step 1 of the next round of optimization operations. The optimization iterations will end when prediction error converges or when a user-defined max iterations count is reached. The proposed two-stage optimization algorithms for cold-start item rating and popularity joint prediction is summarized in Alg. 1, in which the item profile V optimization algorithms and the sentinel user selection method are explained in Section III-C.

Algorithm 1 Iteratively Optimization for Cold-Start Item Rating and Popularity Joint Prediction

Input: User set $\mathcal{N}$, Item set $\mathcal{M}$, Observed item ratings set $\mathcal{O}$, item popularity vector b
Output: Optimized users profile U, items profile V=$T(\cdot)$ and popularity regression coefficient ω
Initial u_i randomly/with pre-defined values, for $i \in \mathcal{N}$;
Initial ω randomly/with pre-defined values;
while ($\mathcal{L}_r + \lambda \cdot \mathcal{L}_b$) *shown in Eq.6 not minimized* **do**
  Update $V = T(\cdot)$ in Eq.7 using Alg. 2;
  Update U with Eq.8, for $i \in \mathcal{N}$;
  Update ω with Eq.9;
  if *user-defined iteration count reached* **then**
    break;
  end
end
return *Optimized* U,V,ω *for prediction*;

C. SENTINEL USER SELECTION ALGORITHM

The optimization of aforementioned item profile V in Eq.7 is vital in the proposed joint prediction approach, which is closely related to the selected sentinel users and their reviews on the items. For an intuitive explanation of the effect of sentinel users, we use a decision tree model shown in Fig.1 to illustrate the sentinel user review procedure. The model is a $(P+1)$-layer, three-way decision tree (in Fig.1, P=3), and every branch node from layer 1 (the root node) through layer P has three children nodes in the next layer. Suppose the decision tree model is already optimized, each branch node in the tree is associated with a selected sentinel user and is parameterized by that user's profile. Every leaf node at the $(P+1)$th layer represents a cluster of items that have similar latent item profiles.

When a new item arrives, it needs to be assigned into a proper item cluster among all clusters in the $(P+1)$th layer. To this end, the new item will go through the decision tree from layer 1 until layer $P+1$, and meet P branch nodes along its path. For every branch node that the item meets in each layer, that node's associated user will give reviews on that item, which can be mapped into three cases showing the 'Like', 'Uncertain' or 'Dislike' attitude of that user to the item. Then the direction towards the left, middle or the right sub branch for the next layer can be determined correspondingly. Finally the item will be assigned into one of the (at most 3^P) leaf nodes. Thus, the new item's profile can be regarded to be estimated according to the P sentinel users' profiles and their reviews.

When selecting different sentinel users to associate with the branch nodes in the decision tree, various item partitioning results in the leaf nodes will be produced, and the extracted item profiles could differ as they are calculated by the selected sentinel users' profiles and reviews. As a result, the rating and popularity prediction error will also be different. The task of our model training process is to construct an optimized decision tree such that each branch node can be associated and parameterized with one optimally selected sentinel user and its profile, so that the least prediction errors can be achieved.

As every branch node can be seen as the root node of a sub-tree, optimization of the decision tree model becomes a *recursive* process that initializes first from the tree root. During the optimization, each branch node is viewed to have the best sentinel user selection result if the user leads to the least predicting errors on both rating and popularity for all items in the sub-tree's leaf nodes. Given one user's profile u_i and item popularity regression coefficient ω (see the optimization step 1 in Section III-B), and given a branch node e with items set $\mathcal{M}_e$ in its sub-tree, the profile of the items in $\mathcal{M}_e$ (denoted as $v_{\mathcal{M}_e}$) can be estimated by user i's reviews $T(a^{(i)})$ on $\mathcal{M}_e$ using Eq.10.

$$v_{\mathcal{M}_e} = T(a^{(i)}) = \Big(\sum_{j\in\mathcal{M}_e}\big(\sum_{r_{ij}\in\mathcal{O}_j} u_iu_i^T + \lambda\omega\omega^T\big)\Big)^{-1} \cdot\Big(\sum_{j\in\mathcal{M}_e}\big(\sum_{r_{ij}\in\mathcal{O}_j} r_{ij}u_i + \lambda b_j\omega\big)\Big) \quad (10)$$

Then for each user $i \in \mathcal{N}$ that can be the candidate sentinel user for current branch node e, the joint rating and popularity prediction loss $c_i(e)$ for all the items $\mathcal{M}_e$ in node e's leaf nodes can be calculated by the following Eq.11.

$$c_i(e) = \sum_{\substack{r_{ij}\in\mathcal{O}\\ j\in\mathcal{M}_e}} \left(r_{ij} - u_i^T T(a_j^{(i)})\right)^2 + \lambda \cdot \sum_{j\in\mathcal{M}_e} \left(b_j - \omega^T T(a_j^{(i)})\right)^2 \quad (11)$$

Among all available candidate users, the user p who has the least prediction errors ($c_p = \min c_i,\ i \in \mathcal{N}$) is selected as the optimal sentinel user for current node e. After that, all of the items $\mathcal{M}_e$ in the sub-tree are partitioned into three disjoint subsets $L(p)$, $U(p)$ and $D(p)$ using Eq.12, according to the '**L**ike', '**U**ncertain' and '**D**islike' review opinions given by e's associated optimal sentinel user p.

$$\begin{aligned} L(p) &= \{j | r_{pj} = \text{like}, \ or \ r_{pj} > 3\} \\ U(p) &= \{j | r_{pj} = \text{uncertain}, \ or \ r_{pj} \notin \mathcal{O}\} \\ D(p) &= \{j | r_{pj} = \text{dislike}, \ or \ r_{pj} \leq 3\} \end{aligned} \quad (12)$$

After item partitioning based on selected sentinel user p's reviews with Eq.12, optimization of the profiles $v_{L(p)}$, $v_{U(p)}$, and $v_{D(p)}$ for the items in the three disjoint subset $L(p)$, $U(p)$, and $D(p)$ can then be divided into three independent optimization processes, shown as the three terms in Eq.13 respectively. Thus the three terms can be optimized separately.

$$\begin{aligned} \underset{\substack{v_{L(p)}\\ v_{U(p)}\\ v_{D(p)}}}{\operatorname{argmin}} \Bigg(&\sum_{j\in L(p)} \Big(\sum_{r_{ij}\in\mathcal{O}_j} (r_{ij} - u_i^T v_j)^2 + \lambda (b_j - \omega^T v_j)^2 \Big) \\ &+ \sum_{j\in U(p)} \Big(\sum_{r_{ij}\in\mathcal{O}_j} (r_{ij} - u_i^T v_j)^2 + \lambda (b_j - \omega^T v_j)^2 \Big) \\ &+ \sum_{j\in D(p)} \Big(\sum_{r_{ij}\in\mathcal{O}_j} (r_{ij} - u_i^T v_j)^2 + \lambda (b_j - \omega^T v_j)^2 \Big) \Bigg) \end{aligned} \quad (13)$$

The form of the three optimization terms shown in Eq.13 are very similar, so does their solutions. For simplicity, we use the symbol $X(p)$ to represent one of the three subsets $L(p)$, $U(p)$ or $D(p)$. Thus, the optimized item profile $v_{X(p)}$ for items in subset $X(p)$ can be shown in Eq.14. The real equations can be simply grabbed by replacing $X(p)$ with $L(p)$, $U(p)$ or $D(p)$ in Eq.14.

$$\begin{aligned} v_{X(p)} = &\Big(\sum_{j\in X(p)} \big(\sum_{r_{ij}\in\mathcal{O}_j} u_i u_i^T + \lambda \omega \omega^T \big) \Big)^{-1} \\ &\cdot \Big(\sum_{j\in X(p)} \big(\sum_{r_{ij}\in\mathcal{O}_j} r_{ij} u_i + \lambda b_j \omega \big) \Big) \end{aligned} \quad (14)$$

As the whole tree optimization is a recursively building process, similar operations described above and the calculations in Eq.11, Eq.12, Eq.13 and Eq.14 will be recursively applied to the branch nodes in the second, third, ..., Pth layer with gradually partitioned subsets L,U and D, whose size are getting smaller and smaller in each sub-tree. In the meanwhile, each branch node in the tree will also gradually select and associate a sentinel user who has the least joint predicting errors on the items in the corresponding sub-tree. The recursive tree model building process ends when a complete decision tree structure is formed till layer $P+1$, or when the number of items in a sub-tree is too small to split. After that, the overall sentinel users for each item can be eventually determined and can then be utilized for cold-start item rating and popularity prediction.

For a $P+1$ layer, three-way tree model, at most $(3^{P+1}-1)/2$ sentinel users will be selected in total, and each sentinel user selection involves N error calculations with Eq.11 on the whole user set $\mathcal{N}$. The computational complexity for Eq.11 can be denoted as $O(K \cdot |\mathcal{O}|)$=$O(KMN)$, which is determined by the latent profile dimension K and the size of observed ratings $|\mathcal{O}|$ in the user-item rating matrix. Then the overall complexity for training sentinel user selection model in one iteration will be $O(KMN) \cdot O(3^{P+1}) \cdot O(N)$=$O(3^P KMN^2)$.

Algorithm 2 Recursive Building of a Three-Way Decision Tree Model

Function BuildTree(CurrentNode e, ItemSet $\mathcal{M}_e$, CurrentTreeLayer d) {
for *each user* $i \in \mathcal{N}$ **do**
  Given u_i and ω in Alg. 1, compute estimated item profile $v_{\mathcal{M}_e}$ with Eq.10;
  Compute predicting error $c_i(e)$ for user i with Eq.11;
end
Select user p who has min $c_i(e)$ as sentinel user for e;
Split $\mathcal{M}_e$ into $L(p)$, $U(p)$ and $D(p)$ by p's reviews with Eq.12;
while *error reduces after split* ***AND*** $d < P$ **do**
  //To recursively construct e's sub branches
  Call BuildTree(e's left child, $L(p)$, d+1);
  Call BuildTree(e's middle child, $U(p)$, d+1);
  Call BuildTree(e's right child, $D(p)$, d+1);
end
Compute item profiles $v_{L(p)}$, $v_{U(p)}$ and $v_{D(p)}$ with Eq.13 and Eq.14;
return *Decision Tree Model}*

We summarize the sentinel user selection algorithm in Alg. 2, which is a recursive decision tree building function that will call itself recursively with updated parameters. When putting Alg. 1 and Alg. 2 together, the proposed two-stage optimization framework and the proposed joint prediction model for cold-start item rating and popularity can run normally.

It should be pointed out that depending on the mapping strategy of user reviews shown in Eq.12, decision tree models other than three-way can also be built and utilized. For example, two-way decision tree model can be used if there are only two possible review opinions for sentinel users. The commonly used 1–5 star rating scores can also be mapped into 5 sub branches for each branch node, so a five-way decision tree model can be built (without counting the 'unknown' branch), and items will be partitioned into 5 disjoint subsets in each split. However in practice, it is not suitable to build too big sub branches as it will dramatically increase the model and computational complexity.

IV. DATASET SETTINGS

Here we describe in details on two application cases and the datasets used in our experiment, which are used for evaluating the joint prediction performance with our sentinel user selection algorithm.

- The first application case is predicting customers' ratings on new-arrival movies and the future popularity of the movies. If the new movie's user-specific ratings and its future popularity are predicted in good accuracy, it will have important implication to the decision makers for how to promote their movies in a cost-effective manner, especially given limited marketing budget.
- The second application case is predicting online social network users' participation states in trending topics, and the topics' future popularity. As there are huge numbers of emerging topics in social networks everyday, it will bring about significant business values if the topic trends and its potential participation users can be predictively identified.

A. MOVIE DATASET

Two movie rating datasets MovieLens-10M and Netflix are used in our experiments to evaluate the joint prediction performance for cold-start movies' rating and popularity.

TABLE 1. Movie dataset statistics.

Dataset	Description	Movie No.	User No.	Rating No.
MovieLens-10M	Original	10,681	71,567	10,000,054
MovieLens-10M	Joint Box-office	5,980	69,878	7,840,570
Netflix	Original	17,770	473,416	100,480,507
Netflix	Joint Box-office	4,901	466,663	27,201,108

The movie rating scores in the two datasets range from 1 to 5 and from 1 to 6 respectively, and the ratings are mapped into three levels (similar to Eq.12), where rating score>3 and ≤ 3 means the user 'likes' and 'dislikes' the movie respectively, and the review opinion is mapped to 'uncertain' if the corresponding score is missing in the dataset. In terms of a movie j's future ground-truth popularity, it is measured by the corresponding U.S. domestic movie box-office data[6] bo_j. To get a properly ranged movie popularity vectors b, a non-linear transformation $b_j = log_{10}(bo_j) - 4$ for each movie j is applied on the box-office data bo that maps the movie box-office range $\$10^4$–$\$10^9$ into popularity range 0–5. We then exclude the movies in MovieLens-10M and Netflix datasets whose box-office data is unavailable or cannot be mapped into the popularity range 0–5. As a result, two new datasets with movie rating and popularity are generated for our evaluation. The statistics of the two datasets are shown in Table 1, and the distribution of each movie's box-office and its ratings count in Netflix dataset are shown in Fig.2.

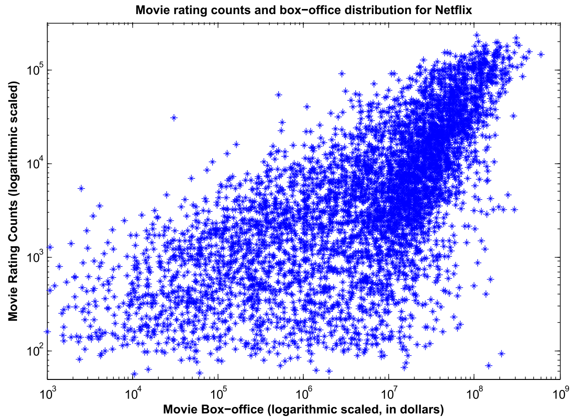


FIGURE 2. Distribution of box-office and rating count for Netflix dataset, where every blue star mark in the figure represents a movie. It can be seen that the movies with higher real box-office tend to have higher rating counts.

B. MICROBLOGGING DATASET

In online social network services such as microblogging services like Twitter and Weibo, the topic of a micropost (or a tweet) can be reflected by the entity or keywords in its content. The semantically related microposts and reposts (or retweets) relevant to the same entity or keywords within a given time window belong to the same topic. A microblog user i is viewed to be participated in a topic j if he or she posts or reposts a topic j-related microposts in his or her time-line. Therefore, a binary valued variable r_{ij} can be used to represent such user-topic relation state: When a user i is participated in a topic j, $r_{ij} = 1$. Otherwise, $r_{ij} = -1$. Then a user-topic relation matrix R can be generated for its elements being $r_{ij}, i \in \mathcal{N}, j \in \mathcal{M}$.

From another perspective, each element in matrix R shows each user's attitude/rating for a topic, indicating he/she cares or not about that topic. So it can also be treated as a user-item rating matrix, whose missing elements are expected to be predicted when a new item (topic) arrives. The total number of users involved in each topic can be viewed as the item popularity. As there are only two possible values (1 and -1) in this user-item matrix, a two-way decision tree model can be trained and used for the sentinel user selection algorithm, where a topic's assigning direction towards the left or the right sub-branch for each branch node is determined by the associated sentinel user's attitude on that topic.

In our experiments, we collect the trending topics' title, timestamp and abstracts from Weibo for a whole month, starting from September 10 to October 10, 2012. Then we recursively retrieve the microposts, all their cascaded reposts, and corresponding author names from Weibo, using the trending topics' titles and abstracts as search query keywords. Each collected micropost and all its reposts are matched to a trending topic based on Term Frequency-Inverted

[6]See Box Office Mojo website. http://boxofficemojo.com

Document Frequency (TF-IDF) and timestamp comparison results. After that, the total number of participants for each trending topic j in the gathered dataset can be calculated and used to measure its item popularity b_j. As most of the participants count range from 1000 to 5000, the item popularity b can be mapped into range 0–5 by multiplying a scale factor $\alpha = 0.001$ with the participants count of each topic. For those trending topics with microblog users count larger than 5000, their mapped popularity b is bounded by 5. The elements in user-topic matrix R can also be generated from the observed microposts and topics, where r_{ij}=1 if microblog user i is observed in participating topic j, otherwise the value of r_{ij}=-1. Additionally, in order to remove the direct connection between b and the number of elements having r=1 for each item in R, we exclude the users and their microposts who participates in no more than three topics when generating matrix R from the gathered dataset. In other words, in microblogging dataset experiments, the full-sized dataset is only used when calculating ground-truth item popularity b, and then the inactive users and their microposts are filtered out. After that, the smaller-sized filtered dataset is used for generating user-topic rating matrix R. The statistics of the gathered microblogging dataset is given in Table 2, and participants count comparison for each topic before and after filtering is shown in Fig.3.

TABLE 2. Microblogging dataset statistics.

Dataset	Topic No.	User No.	Micropost No.
Full	171	98,637	240,853
Filtered	171	29,013	136,926

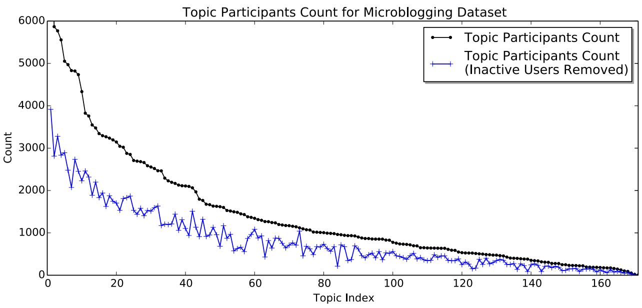


FIGURE 3. Participated microblog users count for each topic in microblogging dataset. The x-axis is the index of each topic from 1 to 171 (ranked by the participated users count). The points in the black curve show the total number of microblog users for each topic in the gathered dataset, which will be mapped into item popularity vector b after multiplying scale factor α. The points in the blue curve show the number of microblog users for each topic in the filtered dataset, i.e. the number of r=1 elements for each column in matrix R. The users who participate in no more than three topics and their microposts are removed in the filtered dataset.

C. DATASET SETTINGS FOR COLD-START ITEMS

In fact, the items in the movie dataset and microblogging dataset introduced above are not strictly 'cold-start' items, as there do exist some ratings for each item in the datasets.

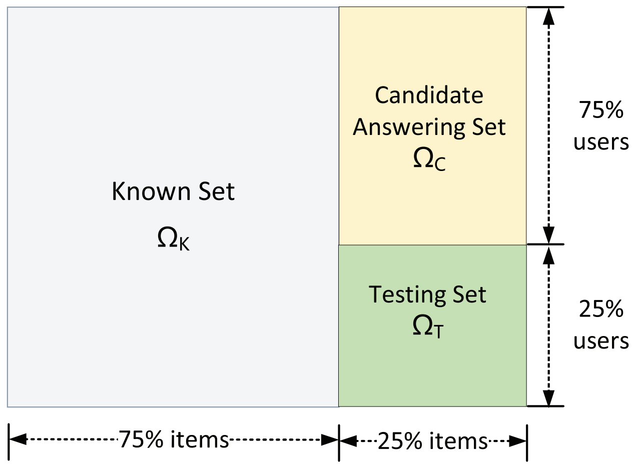


FIGURE 4. Split on original dataset for setting up training dataset Ω_K, cold-start testing dataset Ω_T, and candidate answering dataset Ω_C for picking up sentinel user reviews. Rows and columns are randomly shuffled before split.

Therefore, to evaluate prediction performance for cold-start items, the following operations are used to adapt the datasets suitable for the proposed approach and to make the testing item 'cold-start'.

At first, the rows (representing users) and columns (representing items) in the rating matrix should be *randomly* shuffled. Then the whole matrix are cut into the following three disjoint parts, shown in Fig.4. The left biggest part, containing 75% items and their ratings among all users, is denoted as the *Known Set* Ω_K. This set holds the historical user-item ratings known by the recommendation system that is used for model training. For the rest 25% items in the matrix, their ratings among all users are split into two disjoint parts with 75%-25% ratio on the user domain. As a result, the items in the smaller set (in the bottom right of Fig.4) apparently have much lower average rating counts than the items in training set Ω_K, so the items in this smaller set can be seen as cold-start items. Thus, the set is regarded as *Testing Set* Ω_T for joint prediction performance evaluation, and the testing cold-start items in Ω_T are denoted as $\mathcal{E}_T$.

The last set (in the top right of Fig.4) with the rest 75% user ratings for the testing items $\mathcal{E}_T$ is used in neither model training (sentinel user selection), nor in the final prediction performance calculation. However, as introduced in previous sections, the selected sentinel users' ratings on the testing cold-start items $\mathcal{E}_T$ should be gathered and exploited when we are making the joint predicting. Therefore, this set is reserved to simulate the possible review answers of the selected sentinel users on those testing cold-start items $\mathcal{E}_T$, so the set is called as *Candidate Answering Set* Ω_C. It should be pointed out that for each testing cold-start item in $\mathcal{E}_T$, at most P ratings will picked up from Ω_C and used as the P selected sentinel user reviews for it. As P is much smaller than the average ratings count per item, the testing item can still be viewed as cold-start ones. In real-world usage, Ω_C can be replaced by real user interviews with the selected sentinel users.

V. EXPERIMENTAL EVALUATION AND DISCUSSION

In this section, we will demonstrate the effect of the proposed method taking advantage of the acquired sentinel user ratings to overcome the cold-start item prediction problem. We will also show that the joint predictions for both item ratings and future popularity outperform the peer methods.

Given a known items set $\mathcal{E}_K$ and their corresponding ratings dataset Ω_K, a decision tree model will be trained and the sentinel users will be selected. Then for a testing cold-start item set $\mathcal{E}_T$ ($\mathcal{E}_K \cap \mathcal{E}_T = \varnothing$) and their ratings set Ω_T, we will exploit the trained model as well as the selected sentinel users' reviews on $\mathcal{E}_T$ that picked up from Ω_C to extract the testing item profiles, and then predict user-specific item ratings $\hat{r_{ij}}$ and popularity $\hat{b_j}$ for each testing item $j \in \mathcal{E}_T$. After that, RMSE of predicted item ratings $\mathrm{R}_{\mathrm{rat}}$ and RMSE of predicted item popularity $\mathrm{R}_{\mathrm{pop}}$ are calculated by Eq.15 to evaluate the two aspects of joint prediction performance.

$$\mathrm{R}_{\mathrm{rat}} = \sqrt{\sum_{r_{ij}\in\Omega_T} (\hat{r_{ij}} - r_{ij})^2/|\Omega_T|}$$
$$\mathrm{R}_{\mathrm{pop}} = \sqrt{\sum_{j\in\mathcal{E}_T} (\hat{b_j} - b_j)^2/|\mathcal{E}_T|} \tag{15}$$

A. JOINT PREDICTION PERFORMANCE

The tradeoff weight λ introduced in Eq.6 is an important parameter that affects the joint prediction accuracy. If the value of λ is small, the user who produces less $\mathrm{R}_{\mathrm{rat}}$ rather than less $\mathrm{R}_{\mathrm{pop}}$ is more likely to be selected as a sentinel user. Particularly when $\lambda = 0$, the algorithm in [3] can be viewed as a special case of our method, in which model training task will downgrade to only optimizing prediction accuracy for cold-start item ratings; thus popularity prediction error will increase undoubtedly. Therefore, we run experiments with various λ values and evaluate the results to show how it affects the balance of the two optimization targets in joint prediction.

Additionally, in movie dataset experiments, we also use two kinds of scenarios in exploiting the selected sentinel user's reviews. In Scenario 1, joint predictions and evaluations are made using the proposed algorithms and procedures explained above, in which the selected sentinel users' reviews for testing items $\mathcal{E}_T$ are picked up from Ω_C. As Ω_C is a relatively sparse matrix, some of the selected sentinel users' reviews on a testing item $j \in \mathcal{E}_T$ might not exist in Ω_C. Consequently, the review opinions for item j by these sentinel users will be considered as 'Uncertain' (shown in Eq.12), thus less discriminative item profile would be produced for j than when the review opinions are 'Like' or 'Dislike'. The similar phenomenon may also happen when we are taking real user interviews, as a selected sentinel user may refuse or may be time inconvenient to try the item and give his reviews on time. Therefore, in order to show the impact of the quality of the selected sentinel users' reviews on prediction performance, in Scenario 2 every testing item j has to be rated as 'Like' or 'Dislike' by at least one of its P sentinel users. If all of the P sentinel users' reviews for j are 'Uncertain', that item is considered to be not properly reviewed and the prediction result might be inaccurate, and thus in Scenario 2 it is excluded from $\mathcal{E}_T$ in the RMSE calculation with Eq.15. In terms of microblogging dataset, only Scenario 1 will be applied as the possible user reviews are either 1 or −1.

The MovieLens-10M dataset prediction performance are shown in Table 3. The protocol involves 5 optimization iterations, each item's maximum sentinel users count P=3, profile dimension K=10, regularization coefficient 0.001 and 1 for Eq.8 and Eq.9 respectively. Microblogging dataset experiment evaluation results are listed in Table 4, which has 3 optimization iterations and per item's maximum sentinel users count P=5. All the other parameters are the same with the movie dataset experiments.

TABLE 3. Sensitivity test for parameter λ: prediction RMSE of our method for cold-start item, on MovieLens-10M dataset.

λ	Cold-start Scenario 1		Cold-start Scenario 2	
	$\mathrm{R}_{\mathrm{rat}}$	$\mathrm{R}_{\mathrm{pop}}$	$\mathrm{R}_{\mathrm{rat}}$	$\mathrm{R}_{\mathrm{pop}}$
0	1.56619	1.07552	1.76201	0.61958
300	1.62557	1.01311	1.65805	0.55522
800	1.04140	1.01134	0.94465	0.60817
1000	0.97168	1.00418	0.94995	0.59704
1200	1.09212	1.01272	1.13135	0.59666
1500	1.20946	1.01028	1.25788	0.58953
2000	1.30536	1.00101	1.42081	0.54787

TABLE 4. Sensitivity test for parameter λ: prediction RMSE of our method for cold-start item, on microblogging dataset.

λ	$\mathrm{R}_{\mathrm{rat}}$	$\mathrm{R}_{\mathrm{pop}}$
0	0.22199	1.27044
1	0.22207	1.26204
10	0.22183	1.26204
100	0.21972	0.76804
1000	0.27655	0.76804
5000	0.45452	0.75488

First, we discuss the effect of λ in the evaluation results. In Table 3 and 4, one can see that prediction error $\mathrm{R}_{\mathrm{rat}}$ and $\mathrm{R}_{\mathrm{pop}}$ are indeed affected by the varying λ value. As introduced in Section IV-A, movie ratings and their popularity are mapped into range 1–5 and 0–5. According to the item joint prediction loss equation shown in Eq.6, the former summing term sums all user-specific ratings' prediction error for one item while the latter popularity prediction error has only one value per item. Therefore, when the second term is scaled up by multiplying a proper valued λ during model training, the two summing terms can be comparable in their values, and the overall joint prediction performance is expected to be better. It can be found from the performance results shown in Table 3 that the best joint prediction performance is achieved when λ is around 1000, which number is close to the average rating counts per movie in the MovieLens-10M dataset. For microblogging dataset evaluation results shown in Table 4, the proper λ should be around 100. This suggests that λ

should be set accordingly with the varying dataset scales and value ranges.

Secondly, for each row in Table 3, $\mathtt{R}_{\mathtt{pop}}$ is lower in Scenario 2 than in Scenario 1. As the poorly-reviewed items are not included in the evaluation results in Scenario 2, it indicates that joint predictions performance will indeed improve when the quality of the sentinel user reviews are higher. Therefore, in real-world interview processes, the higher reward could be offered to encourage the selected sentinel users to give more accurate and meaningful review opinions on the cold-start items, so that the joint prediction performance can be improved.

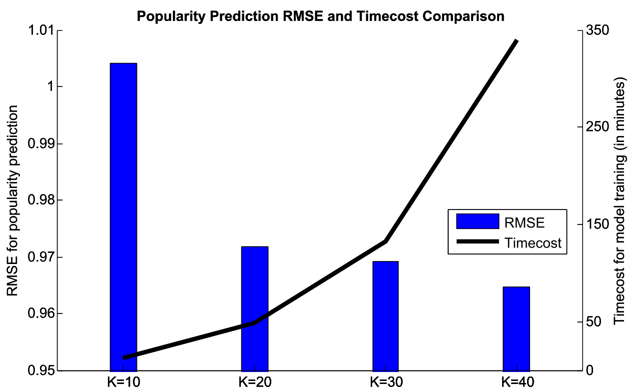


FIGURE 5. $\mathtt{R}_{\mathtt{pop}}$ **and time cost (unit is in minutes) comparison with different user/item profile dimension *K*. As *K* increases, the popularity prediction RMSE (the blue bar) drops with more training time (black curve) needed.**

Furthermore, the parameter settings in the experiments, such as the sentinel users number P and user/item profile dimension K, will also have impact on the prediction performance. As shown in Fig.5, when the user and item profile dimension K increases, the prediction error $\mathtt{R}_{\mathtt{pop}}$ drops. However, in the meanwhile the model training complexity is also increasing, thus much more training time is needed. So in practical usage, the trade-off between prediction accuracy and training complexity should be considered.

B. PERFORMANCE COMPARISON WITH PEER METHODS

In order to compare the joint prediction performance of the proposed algorithms with existing collaborative filtering methods, experiments with two peer methods are carried out. They are namely SVD++ and ALS[7] that implemented in GraphLab Toolkit,[8] where 100 iterations and regularization coefficient 0.001 are used for both methods. As these two collaborative filtering methods are not natively designed for cold-start item rating prediction, we run them in a 'warm-start' situation: The *Known Set* and the whole *Candidate Answering Set* are merged together, and then the whole merged set $\Omega_K \cup \Omega_C$ is regarded as the known set in these two peer algorithms. Afterwards, the testing items' ratings are predicted and evaluated with *Testing Set* Ω_T, and then the extracted latent item profiles by these two methods are also used as feature vectors for item popularity prediction. It can be seen from the above procedures that the two peer collaborative methods need more user ratings data than the proposed methods to make joint predictions.

[7]The detailed description of these methods can be found in GraphLab Toolkit user manual. http://select.cs.cmu.edu/code/graphlab/pmf.html

[8]Available at http://select.cs.cmu.edu/code/graphlab/download.html

TABLE 5. **Prediction RMSE of peer CF methods in warm-start settings, on MovieLens-10M dataset.**

Warm-start Method	$\mathtt{R}_{\mathtt{rat}}$	$\mathtt{R}_{\mathtt{pop}}$
SVD++	1.6083	1.0844
ALS	1.5550	1.1106

The experimental results of the two peer methods on MovieLens-10M dataset are shown in Table 5. By comparing the results with Table 3, it can be found that our proposed method outperforms the two peer methods in $\mathtt{R}_{\mathtt{rat}}$ for most cases except when $\lambda < 300$. In terms of popularity prediction performance $\mathtt{R}_{\mathtt{pop}}$, the two peer methods both underperform our method. In general, the proposed method that exploits the selected sentinel user ratings are indeed reducing the cold-start problem, and it's effective in making joint prediction on both item rating and popularity as the two aspects are both taken into account during model training and sentinel user selection.

C. SELECTED SENTINEL USER AND FREQUENT USER

In all previous experiments, sentinel users are selected among all users in the system. However in real-world the long-tail phenomenon is common in both item popularity distributions and user rating/participation frequency distributions, which means a large portion of users in the system are not actively involved in participating and rating for a relatively large portion of items, thus these items are less popular. One interesting question arises for how well will the joint prediction perform if the sentinel users are selected only from the more frequent users rather than the users with less item involvement counts? We explore the answer by training multiple models with split Netflix datasets and then testing different movie sets with these models. Specifically, denoting $\mathtt{RC}_i (i \in \mathcal{N})$ as each user i's total rating counts of all items, we can split all users in Netflix dataset into five disjoint sets, denoted as U1 through U5, based on the ratio of each user's $\mathtt{RC}_i$ to the maximum valued $\mathtt{RC}_{\max} = \max\{\mathtt{RC}_i | i \in \mathcal{N}\}$. So user set U1 contains users whose rating counts are less than $0.2^*\mathtt{RC}_{\max}$, and set U2 through set U5 contain the users whose rating counts fall in range $(0.2^*\mathtt{RC}_{\max}, 0.4^*\mathtt{RC}_{\max}]$, $(0.4^*\mathtt{RC}_{\max}, 0.6^*\mathtt{RC}_{\max}]$, …, etc. Besides the split on users, all movies in Netflix dataset are also split into five disjoint sets M1 through M5 based on its popularity b with split interval of 1. That is to say, movie set M5 contains the movies with its popularity b in range (4,5], whose corresponding real box-office data range from $\$10^8$ to $\$10^9$ (explained in Section IV-A); movie set M4 contains the movies with b in range (3,4], …, etc.

As a result, the five U $\times$ five M sets partition the original dataset into 25 pieces, so 25 experiments are carried out for each piece of dataset, using the dataset setting up operations described in Section IV-C and the evaluation operations described in Section V. The evaluation results are reported in Fig.6.

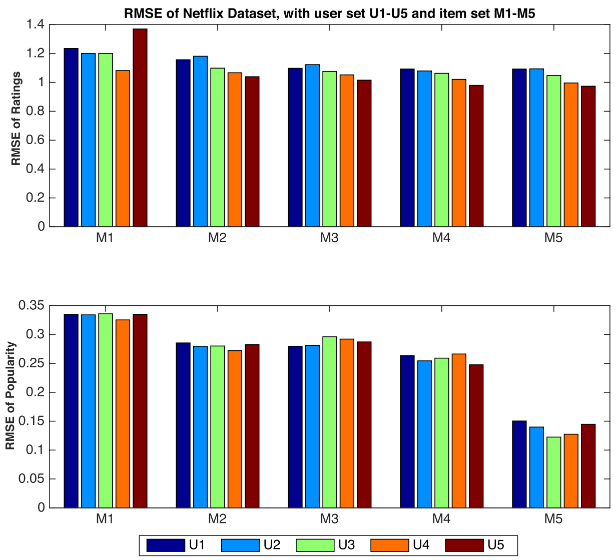


FIGURE 6. **Prediction RMSE comparison for items in set M1–M5 with sentinel users selected from set U1–U5, on the Netflix Dataset.**

First, as shown in the upper sub-figure in Fig.6, item ratings prediction RMSE R_{rat} for the items in set M2 through M5 becomes gradually lower when their sentinel users are selected from user set U1, set U2, ... , and set U5 respectively, thus the performance is better. This is because the users in U5 are more willing to give high quality reviews than the users in U1. So it seems that the selection range for candidate sentinel users can be narrowed to high frequent users when the predicted items have higher popularity. However, one exception exists that RMSE R_{rat} for the items in set M1 is apparently higher when their sentinel users are selected only from user set U5, which means the selected frequent users are not doing well in predicting items with lower popularity. The possible explanation is that some audiences of less popularized items (e.g., items in M1 set) have different tastes and preferences from the other users, so the existence of diversity in user-item preferences makes it an improper choice to predict these items' ratings by only referring to the review opinions from the users with high rating frequency.

In terms of item popularity prediction RMSE R_{pop} shown in the lower sub-figure in Fig.6, the performance gradually improves for the items in M1 through M5 set when sentinel users are selected from the same user set. This shows that when an item's popularity and ratings count arise, it will have more chance to receive high quality reviews by the sentinel users, and thus its popularity prediction results will be more accurate. For the items within the same M set and their sentinel users are selected from different U sets, the value of R_{pop} has no obvious change pattern.

In summary, due to the preference diversity that each item can have its own fans group, selecting sentinel users only from more frequent users may not always have the highest rating prediction accuracy among all items in the system, though it may speed up the user selection process.

VI. CONCLUSION

Recommending items that satisfy individual user's taste and meanwhile have high potential popularity is vital to keep user active on social networks and other online services.

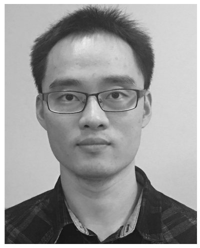
In this paper, we propose a method to overcome the cold-start problem in new-arrival item's rating prediction and popularity prediction, by exploiting the sentinel users' reviews on the new items to elicit their latent profiles. We formulate the joint prediction model as a two-stage optimization task, and extend the functional matrix factorization method by integrating regression-based popularity prediction into the procedure of decision tree model building and optimizing for sentinel user selection. Cold-start item's latent profile can then be discriminately extracted by the reviews of selected sentinel users, hence its popularity and user-specific ratings are jointly predicted. The future work will involve exploring the content-based features for item popularity prediction.

REFERENCES

[1] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 2065–2073, Mar. 2014.
[2] M. Saveski and A. Mantrach, "Item cold-start recommendations: Learning local collective embeddings," in *Proc. 8th ACM Conf. Recommender Syst.*, 2014, pp. 89–96.
[3] K. Zhou, S.-H. Yang, and H. Zha, "Functional matrix factorizations for cold-start recommendation," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2011, pp. 315–324.
[4] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, pp. 1–19, Jan. 2009.
[5] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl. Syst.*, vol. 46, pp. 109–132, Jul. 2013.
[6] H.-R. Zhang and F. Min, "Three-way recommender systems based on random forests," *Knowl.-Based Syst.*, vol. 91, pp. 275–286, Jan. 2016.
[7] H. Ma, I. King, and M. R. Lyu, "Effective missing data prediction for collaborative filtering," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2007, pp. 39–46.
[8] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.
[9] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 89–115, Jan. 2004.
[10] D. Agarwal and B.-C. Chen, "Regression-based latent factor models," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 19–28.
[11] W. Chu and S.-T. Park, "Personalized recommendation on dynamic content using predictive bilinear models," in *Proc. 18th Int. Conf. World Wide Web*, 2009, pp. 691–700
[12] S.-T. Park and W. Chu, "Pairwise preference regression for cold-start recommendation," in *Proc. 3rd ACM Conf. Recommender Syst.*, 2009, pp. 21–28.
[13] G. Capan and O. Yilmazel, "Joint features regression for cold-start recommendation on videolectures.Net," in *Proc. ECML/PKDD Discovery Challenge Workshop*, vol. 770. 2011, pp. 103–109.
[14] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Comput.*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[15] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proc. KDD Cup Workshop*, 2007, pp. 39–42.
[16] D. Liang, J. Altosaar, L. Charlin, and D. M. Blei, "Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence," in *Proc. 10th ACM Conf. Recommender Syst.*, 2016, pp. 59–66.
[17] F. Abel, E. Herder, G.-J. Houben, N. Henze, and D. Krause, "Cross-system user modeling and personalization on the social Web," *User Model. User-Adapt. Interact.*, vol. 23, nos. 2–3, pp. 169–209, 2013.
[18] I. Barjasteh, R. Forsati, D. Ross, A. H. Esfahanian, and H. Radha, "Cold-start recommendation with provable guarantees: A decoupled approach," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 6, pp. 1462–1474, Jun. 2016.
[19] S. Roy and S. C. Guntuku, "Latent factor representations for cold-start video recommendation," in *Proc. 10th ACM Conf. Recommender Syst.*, 2016, pp. 99–106.
[20] N. N. Liu, X. Meng, C. Liu, and Q. Yang, "Wisdom of the better few: Cold start recommendation via representative based rating elicitation," in *Proc. 5th ACM Conf. Recommender Syst.*, 2011, pp. 37–44.
[21] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *Inf. Retr.*, vol. 4, no. 2, pp. 133–151, 2001.
[22] A. M. Rashid *et al.*, "Getting to know you: Learning new user preferences in recommender systems," in *Proc. 7th Int. Conf. Intell. Interfaces*, 2002, pp. 127–134.
[23] N. Golbandi, Y. Koren, and R. Lempel, "On bootstrapping recommender systems," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2010, pp. 1805–1808.
[24] A. M. Rashid, G. Karypis, and J. Riedl, "Learning preferences of new users in recommender systems: An information theoretic approach," *ACM SIGKDD Explorations Newslett.*, vol. 10, no. 2, pp. 90–100, Dec. 2008.
[25] N. Golbandi, Y. Koren, and R. Lempel, "Adaptive bootstrapping of recommender systems using decision trees," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, 2011, pp. 595–604.
[26] M. Sun, F. Li, J. Lee, K. Zhou, G. Lebanon, and H. Zha, "Learning multiple-question decision trees for cold-start recommendation," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, 2013, pp. 445–454.
[27] R. Karimi, M. Wistuba, A. Nanopoulos, and L. Schmidt-Thieme, "Factorized decision trees for active learning in recommender systems," in *Proc. IEEE 25th Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2013, pp. 404–411.
[28] R. Karimi, A. Nanopoulos, and L. Schmidt-Thieme, "A supervised active learning framework for recommender systems based on decision trees," *User Model. User-Adapted Interaction*, vol. 25, no. 1, pp. 39–64, 2015.
[29] H. Steck, "Item popularity and recommendation accuracy," in *Proc. 5th ACM Conf. Recommender Syst.*, 2011, pp. 125–132.
[30] H. Huang, Y. Zhao, J. Yan, and L. Yang, "Improve the 'long tail' recommendation through popularity-sensitive clustering," Dept. Comput. Sci. Technol., Tsinghua Univ., Beijing, China, Tech. Rep., 2015.
[31] D. Jannach, L. Lerche, I. Kamehkhosh, and M. Jugovac, "What recommenders recommend: An analysis of recommendation biases and possible countermeasures," *User Model. User-Adapted Interaction*, vol. 25, no. 5, pp. 427–491, 2015.
[32] C. M. Chen, H. P. Chen, M. F. Tsai, and Y. H. Yang, "Leverage item popularity and recommendation quality via cost-sensitive factorization machines," in *Proc. IEEE Int. Conf. Data Mining Workshop*, Dec. 2014, pp. 1158–1162.
[33] N. Koenigstein, G. Dror, and Y. Koren, "Yahoo! Music recommendations: Modeling music ratings with temporal dynamics and item taxonomy," in *Proc. 5th ACM Conf. Recommender Syst.*, 2011, pp. 165–172.
[34] J. Liebig and A. Rao, "Predicting item popularity: Analysing local clustering behaviour of users," *Phys. A, Statist. Mech. Appl.*, vol. 442, pp. 523–531, Jan. 2016.
[35] Z. Xu *et al.*, "Crowdsourcing based description of urban emergency events using social media big data," *IEEE Trans. Cloud Comput.*, to be published.
[36] Z. Xu *et al.*, "From latency, through outbreak, to decline: Detecting different states of emergency events using Web resources," *IEEE Trans. Big Data*, to be published.
[37] G. Thore, J. Q. Candela, B. Thomas, and H. Ralf, "Web-scale Bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine," in *Proc. 27th Int. Conf. Mach. Learn.*, Jun. 2010, pp. 13–20.

ZHONGCHEN MIAO received the B.S. degree from Shanghai Jiao Tong University, China, in 2010, where he is currently pursuing the Ph.D. degree with the Department of Electronic Engineering. His research interests are data mining applications and user behavior analysis.



JUNCHI YAN received the Ph.D. degree with the Department of Electronic Engineering, Shanghai Jiao Tong University, China. He is currently with the School of Computer Science and Software Engineering, East China Normal University, and IBM Research—China. His research interests are computer vision and machine learning. His first-author papers have appeared in the ICCV, the ECCV, the CVPR, the IJCAI, the AAAI, the IEEE T-IP, and the IEEE T-PAMI. He is a recipient of the IBM Research Division Award in 2014 and 2015, and the ACM China Doctoral Dissertation Nomination Award in 2015. He serves as a Reviewer for these major venues and also the *JMLR*, the *PR*, the *NIPS*, the *ACM-MM*, and the *UbiComp*. He is also an IBM Master Inventor.

KAI CHEN received the Ph.D. degree from Shanghai Jiao Tong University, China, in 2003. He is currently an Associate Professor with the Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University. His major research includes information retrieving, object recognition, and big data mining. He is the key member of his institute on network engineering research. He has been with the several key nation projects and hosted lots of Industry-Academia-Research projects.

XIAOKANG YANG (M'00–SM'04) received the B.S degree from Xiamen University, Xiamen, China, in 1994, the M.S. degree from the Chinese Academy of Sciences, Shanghai, China, in 1997, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, in 2000. He is currently a Distinguished Professor with the School of Electronic Information and Electrical Engineering, and the Deputy Director of the Institute of Image Communication and Information Processing, Shanghai Jiao Tong University. His current research interests include visual signal processing and communication, media analysis and retrieval, and pattern recognition. He serves as an Associate Editor of the IEEE Transactions on Multimedia and the IEEE Signal Processing Letters.

HONGYUAN ZHA received the PhD degree in scientific computing from Stanford University in 1993. He is a professor with East China Normal University and the School of Computational Science and Engineering, College of Computing, Georgia Institute of Technology. He has been working on information retrieval, machine learning applications, and numerical methods. He received the Leslie Fox Prize (1991, second prize) of the Institute of Mathematics and its Applications, the Outstanding Paper Awards of the 26th International Conference on Advances in Neural Information Processing Systems (NIPS 2013), and the Best Student Paper Award (advisor) of the 34th ACM SIGIR International Conference on Information Retrieval (SIGIR 2011). He once served as an associate editor of IEEE Transactions on Knowledge and Data Engineering.

WENJUN ZHANG received the B.S., M.S., and Ph.D. degrees in electronics engineering from Shanghai Jiao Tong University, Shanghai, China, in 1984, 1987, and 1989, respectively. As the Project Leader, he successfully developed the first Chinese HDTV prototype system in 1998. He was one of the main contributors of the Chinese DTTB Standard in 2006. He is currently a Full Professor of Electronic Engineering with Shanghai Jiao Tong University. He holds more than 40 patents and authored over 110 papers in international journals and conferences. He is the Chief Scientist of the Chinese Digital TV Engineering Research Centre, an industry/government consortium in DTV technology research and standardization, and the Director of the Cooperative MediaNet Innovation Center, an excellence research cluster affirmed by the Chinese Government. His main research interests include digital video coding and transmission, multimedia semantic processing, and intelligent video surveillance.