RAPARE: A Generic Strategy For Cold-Start Rating Prediction Problem

Veena Vasudevan, V. Vishnu Priya, Ms Anjana Devi J M.E.
CSE, RMKCET

ABSTRACT:
The recommender system is one of indispensable components in many e-commerce websites. One of the major challenges that largely remains open is the cold-start problem, which can be viewed as a barrier that keeps the cold-start users/items away from the existing ones. In this paper, we aim to break through this barrier for cold-start users/items by the assistance of existing ones. In particular, inspired by the classic Elo Rating System, which has been widely adopted in chess tournaments, we propose a novel rating comparison strategy (RAPARE) to learn the latent profiles of cold-start users/items. The center-piece of our RAPARE is to provide a fine-grained calibration on the latent profiles of cold-start users/items by exploring the differences between cold-start and existing users/items. As a generic strategy, our proposed strategy can be instantiated into existing methods in recommender systems. To reveal the capability of RAPARE strategy, we instantiate our strategy on two prevalent methods in recommender systems, i.e., the matrix factorization based and neighborhood based collaborative filtering. Experimental evaluations on five real data sets validate the superiority of our approach over the existing methods in cold-start scenario.

INTRODUCTION

Since the concept of recommender systems was proposed in 1997 [1], both industry and academia have pro-vided their contribution to the improvement of quality and efficiency for recommender systems. As one of the major components of e-commerce and social websites, recommender system has become an inalienable part of these web-sites. During the last Decade, many mainstream e-commerce companies have reported significant profit growth by integrating recommender systems into their applications [2], [3], [4], [5], [6].

Despite the success of existing recommender systems all over the world, the cold-start problem [7], [8], i.e., how to make proper recommendations for cold-start users or cold-start items largely remains a daunting dilemma. On one hand, cold-start users (e.g., who have rated no more than 10 items) and cold-start items (e.g., which have received no more than 10 ratings) occupy a large proportion in many real applications such as Netflix [9]. On the other hand, the effectiveness of the existing recommendation approaches (e.g., collaborative filtering) largely depends on the sufficient amount of historical ratings, and hence these approaches might quickly become ineffective for cold-start users/items that only have few ratings.

To date, many collaborative filtering methods have been proposed to mitigate the cold-start problem, and these efforts can be divided into three classes. In the first class, a well designed interview process is introduced for cold-start users [10]. During this interview process, a set of items are provided for the cold-start users to express their opinions. The main disadvantage of methods in this class is the additional burdens incurred by the interview process. Methods in the second class resort to side information such as the user/item attributes [11] and social relationships [12] for the cold-start problem. The advantage is that these methods could be applicable for a new user/item with not rating at all. However, they rely on the access of such side information. These methods are inapplicable when the in-formation is not available due to some reasons (e.g., privacy issue, user's social network structure not existing [12]), and has a higher computational cost compared with its side information.
free counterpart. In the third class, the cold-start problem is tackled in a dynamic manner. The intuition is that, compared to existing users/items, ratings for cold-start users/items may be more valuable to improve the accuracy of recommendation for these cold-start users/items; consequently, methods in this class aim to provide fast recommendations for cold-start users/items specifically, and then dynamically and efficiently adjust their latent profiles as they give/receive new ratings. Existing methods in this class include the incremental singular value decomposition (ISVD) method [13] and the incremental matrix factorization method [14], [15], etc. However; methods in the third class cannot serve users with no rating in the recommender system.

Compared with methods in the first two classes, the methods with dynamic view of the cold-start problem do not incur additional interview burden or rely on the access of side information, and thus become the focus of this paper.

In particular, we make the following analogy, i.e., to view the cold-start problem as a barrier between the cold-start users/items and the existing ones, and such a barrier could be broken with the assistance of existing users/items. To this end, we propose a novel rating comparison strategy (RAPARE) which can calibrate the latent profiles for cold-start users/items. Take cold-start user as an example, when a cold-start user gives a rating on an item, we first compare this rating with the existing ratings (which are from existing users) on this item. Then, we adjust the profile of the cold-start user based on the outcomes of the comparisons. Our rating comparison strategy (RAPARE) is inspired by the Elo Rating System [16] which has been widely used to calculate players’ ratings in many different types of match systems, such as chess tournaments, FIFA, ATP, MLB and even some online competition sites (e.g., TopCoder).

The main contributions of this paper are summarized as follows:

- We propose a novel and generic rating comparison strategy RAPARE to serve for the cold-start problem. We formulate the strategy as an optimization problem. The key idea of RAPARE is to exploit the knowledge from existing users/items to help calibrate the latent profiles of cold-start users/items.

- We instantiate the proposed generic RAPARE strategy on both matrix factorization based (RAPARE-MF) and neighborhood based (RAPARE-KNN) collaborative filtering, together with algorithms to solve them.

- We present the algorithm analysis for RAPARE strategy and its instantiations on aspects of effectiveness and efficiency.

- We conduct extensive experimental evaluations on five real data sets, showing that our approach (1) out-performs several benchmark collaborative filtering methods and online updating methods in terms of prediction accuracy for cold-start scenario; (2) earns better quality-speed balance while enjoying a linear scalability.

**Scope of the project:**

In this Project the Concept of RAPARE strategy is proposed which eliminates the the cold start problem. The cold start will make the system halt available. The inactive products are eliminated with the help of existing users, products by studying the characteristics of existing products and inactive products. Now the products are recommended uniformly to the users which make the flow of products to recommend uniformly by making the cold products list go high and the active product no longer.

**PROBLEM STATEMENT**

In this section, we present the notations and the problem statement of recommending items to cold-start users and recommending cold-start items to users.

We list the main notations that are used throughout the paper in Table 1. Suppose we have sets of users U, items I and ratings R. Let u, v represent the users, and i, j represent the items, respectively. Then, r_{ui} \in R is the rating of user u for items i accordingly. For the scenario of cold-start user problem, we call the users who have given more than a certain quantity of ratings (e.g., 10 ratings) as existing users and the rest as cold-start users. Similarly, we call the items that have received more than a certain quantity of ratings (e.g., 10 ratings) as existing items and the rest as cold-start items. Meanwhile, R_u and R_v denote the sets of ratings that belong to cold-start users and existing users, respectively. We use R_w to
represent the set of ratings on item i from existing users.

Table 1

Symbols.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>u, v</td>
<td>Users</td>
</tr>
<tr>
<td>i, j</td>
<td>Items</td>
</tr>
<tr>
<td>r_{ui}</td>
<td>real rating from user u to item i</td>
</tr>
<tr>
<td>r^{*}_{ui}</td>
<td>the predicted rating from user u to item i</td>
</tr>
<tr>
<td>U, I, R</td>
<td>the set of users, items, and ratings, respectively</td>
</tr>
<tr>
<td>I(u)</td>
<td>the set of items that have been rated by user u</td>
</tr>
<tr>
<td>U(i)</td>
<td>the set of users who have rated item i</td>
</tr>
<tr>
<td>R_c</td>
<td>the set of ratings by cold-start users</td>
</tr>
<tr>
<td>R_w</td>
<td>the set of ratings by existing users</td>
</tr>
<tr>
<td>w_n</td>
<td>the set of ratings on item i from existing users</td>
</tr>
<tr>
<td>T</td>
<td>the set of observed ratings</td>
</tr>
<tr>
<td>E</td>
<td>the set of ratings for evaluation</td>
</tr>
<tr>
<td>p_u, q_i</td>
<td>the latent profiles for user u and item i, respectively</td>
</tr>
<tr>
<td>P_c</td>
<td>the set of latent profiles for cold-start users</td>
</tr>
<tr>
<td>Q</td>
<td>the set of latent profiles for items</td>
</tr>
</tbody>
</table>

EXISTING SYSTEM:

In the existing system the recommendation engine only recommend products for the user which are active products, the users will follow the recommendation products and buy those products. There are many underlying probability that those inactive products will never be recommended and unsold.

Disadvantages:

- If a product is remained UN-recommended the product provider and the product will become inactive.
- The product provider will face loss because his products might become inactive and his product might become unsold.
- If the list of inactive product increases then there will be no active products available to sell.

PROPOSED SYSTEM:

The proposed system is aimed at eliminating the inactive products from recommendation system and makes every product active with the help of proposed RAPARE strategy. The RAPARE strategy overcomes the inactive products by proposing the ELO rating system. If a product is cold product the admin will rate the product by giving a review, now the product will activated and it will flow into the recommendation system, If a user buys the particular product and reviews the product Now the average of his
and previous review of admin will calculated. The average will get updated in the review to the product.

Advantages:

- The inactive list of products will get eliminated.
- The product vendor and the seller will get benefited as every product is recommended uniformly.

Matrix Factorization: matrix factorization based collaborative filtering has been one of the most dominate methods in recommender systems. In detail, matrix factorization (MF) [17] assumes that users’ opinions to items are based on the latent profiles for both users and items. With this assumption, MF projects both users and items into a joint latent factor space. The latent factors in the latent space can be seen as the latent profiles for users/items.

K-Nearest –Neighbors Method:

K-Nearest-Neighbors method (KNN) is one of popular approaches in neighbor-hood based collaborative filtering. The key of K-Nearest-Neighbors method is to calculate the similarities between users or items. There are two kinds of KNN (i.e., user-based KNN and item-based KNN) in recommender systems based on the type of similarity calculation.

User-based KNN: The key intuition of user-based KNN is that users with similar tastes may give the similar ratings to the same item. Calculating the similarity between each pair of the given users is the key part of this method. We choose the adjusted cosine similarity [18] from existing similarity measurements in recommender systems.

Item-based KNN: The key of item-based KNN is on the similarity calculation of items.

ELO Rating System: ELO Rating System, which is first adopted in chess tournament, can be used to measure the relative skill levels between players in a certain competition. The basic idea Elo Rating System is that a player’s rating is behind determined by the competition outcomes against her opponents and the ratings of these opponents. For example, a player’s rating w-l be −w if she wins an opponent whose rating is much higher or if she loses to an opponent who has a much lower rating. In other words, the system implicitly aims to minimize the difference between the expected actual outcome of competitions.

To this end, we pay special attention to the cold –start users/items, and aim to calibrate the latent profiles for cold –start users/items with the help of the existing users/items.

THE RATING COMPARISON STRATEGY:

In this section, we first present the core idea of our rating comparison strategy RAPARE, and then formulate the strategy as an optimization formulation. The probabilistic interpretation of RAPARE presents in Appendix.

The Core Idea of RaPare Strategy

Our goal is to break through the barrier between cold-start users and existing users by the assistance of existing users. Specifically, we achieve this goal by borrowing the idea of rating comparison from Elo Rating System. That is, we use the difference between the expected result and the actual result from the rating comparison strategy to calibrate the latent profiles of cold-start users. To start the calibration, we need to first create a competition between a cold-start user and a selected existing user over a given item. Suppose that u is a cold-start user who has just rated item i, and v is an existing user who rated item i in the past. Then, user u and user v have a competition in terms of item, there could be multiple existing users who have rated the same item. We then need to create multiple competitions/comparisons and update the latent profiles of the cold-start user multiple times. Item i. Next, we need to compare the expected result and the actual result of the competition. Still using the example above, the expected result of this competition can be calculated as the difference between user u (which is also the parameters that we aim to estimate). In the meanwhile, with the actual rating rui, we can have the actual result of the competition, which is the difference between rui and rvi. Finally, based on the expected result and actual result of the competition, we may update the latent profiles of user u by following a similar strategy as in Elo Rating System. That is, the
farther the expected result of competition deviates from its actual result, the larger the latent profiles of user $u$ will be changed. In reality, when a cold-start user gives a rating on an item, the RAPARE strategy is equivalent to minimize the difference between the expected result and actual result by learning/calibrating the latent profiles of cold-start users.

**CONCLUSION**

In this paper, we have proposed a generic rating Comparison strategy (RAPARE) to make proper recommendations for cold-start problem. In particular, the RAPARE strategy provides a special, fine-grained treatment for cold-start users and cold-start items. This generic strategy can be instantiated to many existing methods for recommender systems. We proposed the RAPARE-MF (instantiating with matrix factorization method) and RAPARE-KNN (instantiating with nearest neighborhood method) models as well as algorithms to solve them. Experimental evaluations on five real data sets show that our approach outperforms several benchmark collaborative filtering and online updating methods in terms of prediction accuracy, and RAPARE-MF can provide fast recommendations with linear scalability.

**REFERENCES**


