Abstract—Collaborative filtering (CF) is one of the most successful technologies in recommender systems, and widely used in many personalized recommender applications, such as digital library, e-commerce, news sites, and so on. However, most collaborative filtering algorithms suffer from data sparsity problem which leads to inaccuracy of recommendation. This paper is with an eye to missing data imputation strategy in nearest-neighbor CF. We propose an effective CF framework based on missing data imputation before conducting CF process, which utilizes item’s genre information. In the experimental evaluations, 19 item’s genres are employed in the imputation stage. The results show that the proposed approaches effectively alleviate the negative impact of data sparsity, and perform better prediction accuracy than traditional widely-used CF algorithms.

Keywords—collaborative filtering; recommender system; missing data imputation; sparsity problem

I. INTRODUCTION

With the rapidly increasing web information and content, we have to spend much time on searching the interesting topics we just need. Information filtering has emerged as a necessary technology to address “information overload” problem. Recommender system utilizing information filtering technology is popular in various applications, such as e-commerce, digital library, news sites, entertainment sites and IPTV. Collaborative filtering (CF) is one of the most successful technologies in recommender system [1, 2], which adopts information retrieval and data mining techniques to provide recommendations based on suggestions of users with similar preferences. There are many famous sites using CF to develop their personalized systems, such as Tapestry [3], WebWatcher [4], GroupLens [5], Firefly [6], SELECT [7], SiteSeek [8], Amazon [9], douban (www.douban.com), and LikeMinds (www.macromedia.com).

The main idea of CF is that the target user is likely to enjoy the items which other users with common interests like. Therefore, the accuracy of similarity is the key to the success of the recommendation, both for selecting neighborhoods and computing predictions. It’s important to accurately measure user’s or item’s similarity. However, the computed similarities between users or items are somewhat inaccurate due to data sparsity, which consequently frustrate the quality of prediction and recommendation. Presently, there is little work on rating data imputation aspect, it’s necessary to examine it as a pretreatment stage before CF process. Based on the non-sparse rating data, similarity computation will be more accurate and result in better prediction.

In this paper, we focus on the rating imputation strategy before nearest-neighbor CF process and propose a new CF framework to improve prediction accuracy based on missing data filling up, which utilizes item’s genre information. And to validate and evaluate the advantage of this imputation method, user-based CF and item-based CF depending on the modified rating data are respectively conducted. The results of comparison experiments show the superiority of our proposed method, achieving more than 10% improvements in terms of precision over widely-used CF algorithms, especially on the extremely sparse data.

The rest of this paper is arranged as follows: In the following section, the background of CF is briefly introduced. In section 3, we present our proposed framework and imputation strategy in detail. The experimental evaluations are given in section 4. Finally, section 5 concludes the paper and provides directions for future research.

II. BACKGROUND

In this section, we review traditional collaborative filtering algorithms and related work on alleviating data sparsity problem, and then clarify our contribution in CF process on this issue.

A. Collaborative Filtering

In 1992, the Tapestry system [3] first introduced Collaborative Filtering (CF). It is a technique of using peer opinions to predict the interests of others. CF algorithms are based on the assumption that similar users prefer similar items, or that a user expresses similar preferences for similar items. They fall into two general classes: user-based CF (UBCF) [3] and item-based CF (IBCF) [9]. Both of them employ two fundamental steps as follows:

1) Formation of user (or item) neighborhood

Based on the user-item rating data matrix, similarity between users (or items) are calculated, and some users (or items), known as neighbors, are selected based on their similarity to the active user (or to under-predict item). Similarity computation is the key step in CF. Whether the similarity is measured accurately directly affects the neighbor selection, and then affects the quality of recommendation.

978-1-4244-4520-2/09/$25.00 ©2009 IEEE
Two commonly used similarity measures are Pearson Correlation Coefficient (PCC) [10] and Adjusted Cosine [9]. The equations are given in (1) and (2). As described in [11], the PCC algorithm generally achieves higher performance in user-based CF, while Adjusted Cosine has a clear superiority in item-based CF.

\[
sim(u, v) = \frac{\sum_{i \in I_u} (r_{ui} - \bar{r}_u) \cdot (r_{vj} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u} (r_{ui} - \bar{r}_u)^2} \cdot \sqrt{\sum_{j \in I_v} (r_{vj} - \bar{r}_v)^2}}
\]

(1)

\[
sim(i, j) = \frac{\sum_{u \in U_i} (r_{ui} - \bar{r}_i) \cdot (r_{uj} - \bar{r}_j)}{\sqrt{\sum_{u \in U_i} (r_{ui} - \bar{r}_i)^2} \cdot \sqrt{\sum_{u \in U_j} (r_{uj} - \bar{r}_j)^2}}
\]

(2)

Here, \(r_{ui}\), \(r_{vj}\) denotes the rating of user \(u\) on item \(i\), \(j\); \(\bar{r}_u\), \(\bar{r}_v\) denote mean rating value for user \(u\), \(v\); \(I_u\), \(I_v\) denotes set of items both rated by user \(u\) and \(v\); \(U_i\), \(U_j\) denotes the set of users rated both item \(i\) and \(j\); \(U\) denotes the set of users.

2) Prediction of missing data and Top-N list generation

In predicting the rating value of a given item for a particular (active) user based on the ratings from other similar users (or this user’s ratings on similar items), a weighted average of the neighbor’s ratings is used to predict the rating value, with the weights proportional to the similarity between each neighbor and the active user (or the predicting item). And then select the items with high scores as recommendation output. Here CF comes to an end.

B. Sparsity Problem and Related Work

In many real cases, users rate only a very small percentage of items. With the magnitude of users and items, almost all CF algorithms are confronted with sparsity problem. For example, in many large-scale e-commerce systems, the percentage of a user’s rated items is always under 1% [9], and even less of common items of every two users. Due to lack of sufficient information, it’s difficult to measure the similarity accurately, which leads to a serious degradation of recommender quality.

To alleviate the sparsity problem, [12] proposes an algorithm in which all unrated items are given a default rating. The shortcoming of this algorithm is based on a truth that the ratings of the unrated items given by user are impossibly same. So the result of the algorithm is not reliable. Reference [13] proposed to reduce the dimensionality of recommender system database using the method called Singular Value Decomposition (SVD) and then users have ratings on every item on reduced dimension. But [14] indicated that this method may lead to loss of information and its effects are very data dependent. In high dimension domain, dimensionality reduction can’t achieve good performance. A novel collaborative filtering algorithm based on item rating prediction is proposed in [15]. This method predicts item ratings that users have not rated by the similarity of items. It can effectively improve the extreme sparsity of user rating data, and provide better recommendation results than traditional collaborative filtering algorithms. But the double-computation similarity significantly adds time cost. Reference [16] proposed a generative probabilistic framework to exploit more of the data available in the user-item matrix by fusing all ratings with a predictive value for a recommendation to be made. This complete model is more robust to data sparsity, because the different types of ratings are used in concert, while additional ratings from similar users towards similar items are employed as a background model to smooth the predictions. However, the efficiency is a bottleneck. Therefore, methods mentioned above all confront with their own flaws.

C. Our Contribution

To smooth the data sparsity problem, we propose a new CF framework, which uses missing data imputation as a pretreatment stage before similarity computation. In the imputation, we combine item’s genre information and original ratings to generate the proper value to fill up the user-item rating matrix. Here according to the datasets, 19 item’s genres are employed in the calculation. Based on the filled non-sparse data, user-based and item-based CF are conducted, respectively called imputation user-based CF (IUBCF) and imputation item-based CF (IIBCF). With the adequate data, the similarity calculation will be well-founded and more accurate, which will result in better quality of prediction.

III. METHODOLOGY

The framework of CF based on missing data imputation is given in Fig.1.

![Figure 1. Framework of missing data imputation-based CF](image)

We introduce an additional missing data imputation step in the traditional CF process. In the imputation stage, item’s genre information is employed in the calculation, fusing with the original user-item rating data. Based on the imputed data, similarity computation and missing data prediction are
conducted subsequently. Note that similarity computation, either user-user or item-item, is based on the imputed rating data matrix; and prediction is based on the similarity matrix produced from the former step and the original rating data.

A. Algorithm

1) access the original user-item rating data document and item information document, generating an original user-item rating data matrix \( A(m,n) \);
2) utilize \( A(m,n) \) and item information document to calculate the imputed value of missing data in \( A(m,n) \), forming a modified full rating matrix \( B(m,n) \);
3) calculate the similarity between users (or items) based on \( B(m,n) \) and generate K nearest neighbors;
4) predict the missing data in \( A(m,n) \) based on the neighbors’ ratings in the original rating data \( A(m,n) \) and their similarity values with the active user (or the under-predicting item), generating recommendation for the active user.

Figure 2. CF algorithm based on missing data imputation

Here, \( m \) (or \( n \)) is the number of users (or items). In the algorithm, our attention will be concentrated on the data imputation strategy to alleviate data sparsity, which answers the question of how to make the proper imputed value based on the existing information in the data set. To illustrate the advantage of this item’s genre based data imputation, we respectively perform user-based and item-based CF afterwards, which are named as IUBCF and IIBCF. The performances of them two will be presented in the experiments part.

B. Missing Data Imputation

This imputation approach is based on the hypothesis that the given user probably ranks closely on the items in the same genre, which is also accordant with common sense. For example, Elma likes romance movies, while dislikes horror ones, well then she is likely to give high ratings on her unrated romance movies and low ratings on the horror ones. It suggests that making use of item’s genre information and user’s original rating to fill up the missing data is an appropriate way to smooth the sparsity problem in CF, so we try to calculate user’s mean rating on every item genre as the imputation values turning the original sparse rating into a full rating matrix. The following algorithm in Fig. 3 shows the imputation process in detail.

In the step 2)-b), there are two exceptional circumstances: One is when \( i \) belongs to multiple genres, the imputed value is substituted by \( u \)’s mean rating on all items in these genres; Another is user \( u \) has no ratings on the items in \( i \)’s genre, in other words, \( T \) is an empty set. On this occasion, the imputed value is substituted by \( u \)’s mean rating on the all items.

is more, when \( u \) has no rating on all items, we use the median rating as the imputation value. We know at this moment, \( u \) is a new user in the system.

1) clone the original rating data matrix \( A(m,n) \) to \( B(m,n) \), \( B \) is used to record the imputed data and to calculate the similarity, \( A \) is used in the imputation stage and last prediction stage;
2) for each null unit \( r_{ui} \) in \( A(m,n) \)
   a) search item \( i \)’s genre information, denoted as \( g_i \);
   b) access the item set in which user \( u \) has rating on each element which has the same genre attribute as item \( i \), the set is denoted as \( T \), that is to say, \( \forall t \in T, g_t = g_i \).
   c) calculate \( u \)’s mean rating on the item set \( T \) as the imputed value of \( r_{ui} \), denoted as \( \text{impute}_{ui} = \sum_{j \in T} r_{uj} / \|T\| \);
   d) fill up the corresponding unit \( r_{ui} \) in \( B(m,n) \) with this value \( \text{impute}_{ui} \);
   end for

Figure 3. Missing data imputation algorithm using item’s genre

C. Similarity Computation

In IUBCF approach, similarities between users are computed for selecting neighbors. As presented in [11], we choose PCC formula to measure user’s similarity. While in IIBCF, similarities between items are employed for neighbors’ generation, so Adjusted Cosine method turns into a better way to measure item’s similarity empirically. What’s worth mentioning is that either in IUBCF or in IIBCF, similarity computation is based on the imputed data, not on the original sparse data anymore. As we know, compared with computing with insufficient and sparse data, both PCC and Adjusted Cosine are more accurate under the full rating data condition.

D. Prediction of Missing Data

The last stage of CF is to generate the recommendations in terms of prediction based on neighbors’ view. Because in the two new approaches, similarity is user-based and item-based respectively, so there are two corresponding equations to predict the missing rating for target user. The predicting ratings of user \( u \) on item \( i \) are given as follows:

\[
P_{ui} = \bar{r}_u + \frac{\sum_{v \in \text{KNN}(u)} \text{sim}(u,v) (r_{vi} - \bar{r})}{\sum_{v \in \text{KNN}(u)} \text{sim}(u,v)} (3)
\]

\[
P_{ui} = \bar{r}_u + \frac{\sum_{j \in \text{KNN}(i)} \text{sim}(i,j) (r_{uj} - \bar{r})}{\sum_{j \in \text{KNN}(i)} \text{sim}(i,j)} (4)
\]
Here, $KNN(u)$ is user $u$'s K nearest neighbors set, in which the users are top-K most similar with $u$; $KNN(i)$ is item $i$'s K nearest neighbors set, in which the items are top-K most similar with $i$. Note that neighbors’ ratings are from the original data matrix to avoid the inaccuracy attributed to the imputed data.

IV. EXPERIMENTAL ANALYSIS

We perform extensive experiments to examine the effectiveness of our new approaches. The results are to illustrate the following issues: (1) How does the neighborhood size affect the accuracy of the proposed approaches? (2) How do the proposed methods compare with traditional widely-used CF approaches? (3) How do the proposed methods work on the sparse data? We don’t design experiment to evaluate time cost because the new algorithms won’t have any great impact on the online efficiency, the missing data imputation step can be done offline periodically.

A. Data set

We use data from the web-based recommender system MovieLens (http://www.cs.umn.edu/Research/Grouplens/) in the experiments. This data set consists of 100,000 ratings (1-5) from 943 users on 1682 movies; each user had rated at least 20 movies; and information about the items (movies) is included. 5 pairs of training and test sets are provided as u1.base and u1.test through u5.base and u5.test, which are 80%/20% splits of all rating data into training and test data. Item’s genres (unknown, Action, Adventure, Animation, Children’s, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western) in the data set are employed to fill up the original user-item rating data.

B. Evaluation Metric

We use the Mean Absolute Error (MAE) metrics to measure the prediction quality of our proposed approaches. If prediction ratings set of $N$ users is $\{m_{i_1}, m_{i_2}, \ldots, m_{i_N}\}$, and corresponding true ratings set is $\{n_{i_1}, n_{i_2}, \ldots, n_{i_N}\}$, MAE is defined as (5).

$$MAE = \frac{1}{N} \sum m_i - n_i$$  \hspace{1cm} (5)

We use MAE as evaluation metric to report prediction experiments. The lower the MAE, the more accurately the algorithm predicts user ratings.

C. Experimental Results

1) Neighborhood size effect

The neighborhood size has a significant effect on the prediction quality. We performed experiments where we varied the number of neighbors by step of 20 from 20 to 200. The results of the two proposed approaches are shown in Fig. 4, from which we can observe that in the beginning, as the number of neighbors is increasing, the MAE is dropping considerably, but after 80 neighbors are selected, there are no obvious improvements in quality, so we select 80 as the optimal choice of neighborhood size.

On the other hand, we found as the neighborhood size is increasing, the results of method IIBCF exceed IUBCF’s gradually. The reason is item genre information is utilized in the data imputation, so the similarity computation employing item-based method is more reasonable and accurate.

Figure 4. Neighborhood size impacts on MAE in IUBCF and IIBCF

2) Comparison on full ratings

The given five pairs of data sets are adopted to compare the performance of our proposed algorithms with the traditional CF methods. Here in UBCF, similarity computation follows PCC formula, and in IBCF follows Adjusted Cosine formula. The comparison results are shown in Fig. 5. We observe that our new approaches improve the prediction quality of CF system, and outperform UBCF and IBCF, achieving more than 10% improvements in terms of prediction precision.

Figure 5. MAE comparison on 5 pairs datasets with full ratings

3) Comparison on sparse ratings

Especially, we conduct the comparison experiment on the sparse data to validate the superiorities of our proposed methods, in which some ratings are wiped off from the training set and the given ratings of each user are from 5 to 25. Because the item size is close twice as user size, item-
based methods have poorer performances than user-based ones on sparse data, we divide the comparison experiment as two pairs: UBCF vs. IUBCF and IBCF vs. IIBCF. In the first pair, we set neighborhood size $k=80$, while in the second one, we set $k=160$. The results are shown in Fig. 6 and Fig. 7. Remarkably, our proposed IUBCF and IIBCF respectively outperform the other two methods in every given rating condition. This comparison proves our novel imputation measures are outstanding even though the rating data is extremely sparse.

**Figure 6.** MAE comparison on sparse data (UBCF vs. IUBCF)

**Figure 7.** MAE comparison on sparse data (IBCF vs. IIBCF)

### V. CONCLUSIONS

In this paper, we propose a new CF framework based on missing data imputation, which alleviate the sparsity problem by filling up the original rating matrix. The imputation values are calculated by fusing original ratings and item’s genre information. Then two new algorithms are introduced, named as IUBCF and IIBCF, the difference between them is similarity computation is user-based or item-based. Through the experimental analysis, our approaches prove to be effective ways in CF. They both outperform other traditional CF methods in accuracy of recommendation. Especially, they go well under the extremely sparse data circumstances. For future work, we will conduct more research on capturing more users’ information and context in real applications, and adopt them in prediction.

**ACKNOWLEDGMENT**

This work is supported by the grants from “National Key Technology R&D Program of China during the 11th Five-Year Plan Period” (No.2007BAH09B04) and “Shanghai Science and Technology Elite Program” (No.07QB14036).

**REFERENCES**


