

# Improving Recommendation via Inference of User Popularity Preference in Sparse Data Environment

Xiaoying TAN<sup>†</sup>, Nonmember, Yuchun GUO<sup>†a)</sup>, Member, Yishuai CHEN<sup>†</sup>, and Wei ZHU<sup>††</sup>, Nonmembers

**SUMMARY** The Collaborative Filtering (CF) algorithms work fairly well in personalized recommendation except in sparse data environment. To deal with the sparsity problem, researchers either take into account auxiliary information extracted from additional data resources, or set the missing ratings with default values, e.g., video popularity. Nevertheless, the former often costs high and incurs difficulty in knowledge transference whereas the latter degrades the accuracy and coverage of recommendation results. To our best knowledge, few literatures take advantage of users' preference on video popularity to tackle this problem. In this paper, we intend to enhance the performance of recommendation algorithm via the inference of the users' popularity preferences (PPs), especially in a sparse data environment. We propose a scheme to aggregate users' PPs and a Collaborative Filtering based algorithm to make the inference of PP feasible and effective from a small number of watching records. We modify a  $k$ -Nearest-Neighbor recommendation algorithm and a Matrix Factorization algorithm via introducing the inferred PP. Experiments on a large-scale commercial dataset show that the modified algorithm outperforms the original CF algorithms on both the recommendation accuracy and coverage. The significance of improvement is significant especially with the data sparsity.

**key words:** user popularity preference, video recommendation, sparse data

## 1. Introduction

As the online video services prevail in recent years, more and more researchers are devoted to developing recommendation algorithms in order to satisfy users' personal preferences. As predicted in report [1], the online video services, such as Youtube [2], will account for approximately 86% of the global consumer traffic by 2016. Driven by the invest revenue, the providers of these video services are devoted into providing personalized services, e.g., recommendation system, rather than just listing the Top- $N$  popular videos on their home pages.

Among all recommendation algorithms, the widely used Collaborative Filtering (CF) algorithms work quite well except in the cases of sparse data or cold-start conditions. The CF methods produce recommendations via finding the like-minded users (or similar items) based on the patterns of ratings without need for exogenous information about either items or users [3]. As they rely totally on past behavior of users, the recommendation accuracy degrades largely in the sparse or cold-start situations [4]. In such

situations, there are only a few historical ratings available, which results in the bias of the estimation of the similarities among users or items and in turn the bias of recommendation results.

To deal with the sparsity problems, there are basically two types of methods. One type of methods takes into account some auxiliary information, e.g., the sound, visual, textual or content information of videos [5]–[7] or the attribute of users extracted from cross network [8], [9]. Another type fills the missing ratings with default values [10], [11], e.g., the average of ratings or video popularity.

But these methods have their own drawbacks. The former methods require extra auxiliary information which are costly and non-scalable. The latter methods use the average ratings which are general instead of personal, leading to the degradation of the recommendation performance. For example, filling the missing ratings with the video popularity makes the recommendation results incline towards the popular videos and degrade the personality, the coverage and the novelty of recommendation. Moreover, such a recommendation impairs the interest of the service providers as well, as it decreases the recommendations of the videos on the “long tail”.

In fact, a few work have noticed that users have diverse preferences on video popularity [12], [13], referred to as *Popularity Preference (PP)* in this paper, but they have not noticed that the awareness of the users' PPs could help improve the recommendation performance. Only Oh *et al.* took advantages of such preferences to improve both the accuracy and the coverage of the recommendation [14]. However, they drew the PP for users with statistics of a sufficient training dataset (the training ratio is 80% in their experiment), which is not appropriate in the sparse data environment as the accuracy of the statistical PP relies on a number of sufficient historical records.

To fill this gap, in this paper, we first propose a scheme to aggregate users' PPs and a Collaborative Filtering based algorithm to make the inference of PP feasible and effective from a small number of watching records. We then modify a  $k$ -Nearest-Neighbor recommendation algorithm and a Matrix Factorization algorithm via introducing the inferred PP. We name the modified algorithms as *Popularity Preference Inference enhanced k-Nearest-Neighbor (PPI-NN)* and *Popularity Preference Inference enhanced Matrix Factorization (PPI-MF)* respectively. To evaluate the modified algorithm performance, we conduct experiment on a large scale dataset

Manuscript received June 28, 2017.

Manuscript revised November 8, 2017.

Manuscript publicized January 18, 2018.

<sup>†</sup>The authors are with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, 100044, China.

<sup>††</sup>The author participated in this work when he was with the PPTV Inc., Shanghai, 200120, China.

a) E-mail: ychguo@bjtu.edu.cn

DOI: 10.1587/transinf.2017DAP0024

from PPTV [15], one of the largest online video streaming providers in China. The experiment results prove that our PPI-NN algorithm could improve both the accuracy and coverage of the recommendation in the sparsity situation.

The remainder of this paper is organized as follows. In Sect. 2, we introduce the related work. In Sect. 3, we introduce the preliminary of this work including the dataset used in this work and the definition of user PP. In Sect. 4, we propose the PPI-NN recommendation algorithm and in Sect. 5 we evaluate the algorithm on our dataset. Finally, we conclude the paper in Sect. 6.

## 2. Related Work

### 2.1 Collaborative Filtering Algorithms

Collaborative Filtering recommendation algorithms have been well summarized in literatures [16]–[18]. They are usually categorized into memory-based (e.g., Neighbor-based) and model-based (e.g., Bayesian belief nets, latent factor analysis) techniques. The model-based techniques, especially Matrix Factorization (MF) models [19]–[21], are popular in video systems for good scalability. The memory-based ones based on nearest-neighbors enjoy a huge amount of popularity for the simplicity, justifiability, efficiency and stability [4].

The CF algorithms work well in most common recommendation scenarios, but their performances degrade largely in the sparse data environment where the rating records are not enough.

### 2.2 Sparsity and Cold-Start Problem in Recommendation

Sparsity and cold-start situations [22] are common in recommendation systems, especially when there are only limited watching records for inactive users or for users at the very beginning of a system. There are two types of mainstreaming methods to tackle with the sparsity or cold-start problem in recommendation.

One type of the methods integrates auxiliary information extracted from extra data sources. For example, Yan et. al., [8], [9] introduced users' information on Twitter into Youtube video recommendation. Li, et. al., [7] extracted sound and visual information of items to build a multiple kernel SVM recommendation algorithm. Schein, et. al., [5] developed generative probabilistic models by mixing content data with collaborative data. Roy, et. al., [6] utilized textual information of each item as auxiliary information for learning latent factor representations.

The other type of methods fills the missing ratings with default values, e.g., the average of a user's ratings or video popularity, in order to generate recommendations with these default ratings. For example, Xia et. al., [10] combined the popularity and the mean rating value of item for recommendations for new users. Ahn [11] combined the popularity-class information with genre information to generate the recommendations under the sparse or cold-starting conditions.

These methods have their drawbacks. The first type methods have to extract extra data source which is usually expensive and is likely to lead to the difficulty of the knowledge transference and associations. The second type methods might degrade users' personal interest and the coverage and novelty of the recommendations.

### 2.3 User Popularity Preference

Video popularity has been characterized and predicted in many literature [23]–[25], but, to our best knowledge, only a few works have studied users' personalized popularity preferences till now. In fact, users have been proved to have diverse PPs [12], [13]. Oh, et. al. [14] took advantages of the statistical user PPs to help recommendations. The statistic method, however, is likely to be biased to capture users' PPs, especially for the users having only quite limited records.

As far as we know, we are the first in our earlier work [13] to propose the CF based algorithms to estimate user PP that is more accurate than the statistic method in other works.

## 3. Dataset

We introduce the dataset used for training and evaluation. We collected a large-scale dataset from clients of PPTV, one of the largest typical online video streaming systems in China. The dataset collected spans from March 23rd to 28th in 2011, including more than 10 million user watching sessions, 20 thousand unique videos and 1 million users. Each session record includes anonymous user ID, video ID, user watching time and session start time. The watching time is the valid playback time exclusive of the time spent on advertisement. The watching time normalized by the length of the video watched is regarded to be the user's interest rating towards the video. Uniformly, the rating is scaled in the range 1 to 10.

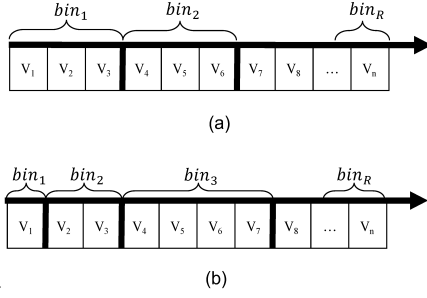
To avoid the effects of video types on user preference, we focus our analysis on the *movie* type and we do not distinguish between videos and movies hereinafter. We filter out the sessions shorter than 30 seconds which are considered as surfing rather than purposeful watching.

We filter out the users with less than 20 records to ensure that we have enough data to evaluate the accuracy of our inference algorithm. To this end, we need a users' real preference with the support of enough data. The threshold of 20 is also used in [14] for the same goal.

The resulted dataset includes more than 20 thousands of movies, 90 thousands of users and more than 2 million of sessions.

## 4. Aggregation of User Popularity Preferences

Original CF algorithms make the prediction of user preference based on the easy-to-obtained user-video ratings or behavior records without any extra expensive information. But



packing.pdf

**Fig. 1** The sketches of the schemes (a) linear packing assignment and (b) log-like packing assignment to pack videos into several bins according to the video popularity. The videos are sorted by descending order of popularity.

when only a few number of the ratings or records are available, the CF algorithms can hardly find the video that interests the user most, since the user-video matrix for training is too sparse to contribute to finding the similar neighbors or the pattern of the user's preference.

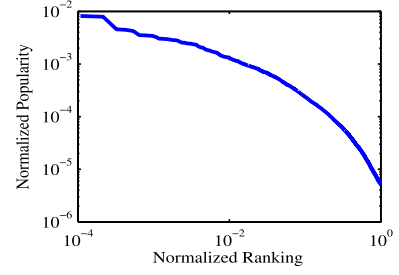
To provide more information of user preference effectively and efficiently in the sparsity situations, we propose an aggregation scheme to identity user Popularity Preferences (PP).

We first define the popularity of a video to be the total number of the sessions referred to this video during the measured period. In other words, the video popularity varies with the measured period. In our dataset, the popularity distributes extremely uneven among videos, even at the shortest measured period that contains 200 thousand of sessions (10% of the whole dataset for training, for example).

Facing with data sparsity, we pack the videos into several discrete ranking bins according to their popularity. In fact, excessively fine-grained ranking is unnecessary especially for the video on the tail whose popularities are quite the same. Moreover, the statistic of a users' preference on such a "fine-grained" popularity may be inaccurate, as this popularity changes against time quite frequently and the user is likely not been informed just at the time she was watching the video. Instead, the assess of users' preferences on "coarser" popularity - measured in a packed bin granularity, would be more reliable.

An extreme packing scheme is to linearly assign ranking  $r$  to bin  $b$ , as illustrated in Fig. 1 (a). That is,  $b(r) = \text{round}(r/\text{bin\_length})$  where  $\text{bin\_length}$  is the video number assigned in each bin, and  $\text{round}(x)$  denotes the rounded integer value of  $x$ . An obvious disadvantage of this scheme is that the click numbers distribute imbalanced extremely between bins. As the highly-skewed popularity distribution shown in Fig. 2, in this scheme, the first two bins will dominate most of all the clicks and it hardly differentiates between users' PPs.

Instead, we propose an efficient bin assignment scheme inspired by the power-law-like popularity distribution shown in Fig. 2. Specifically, we assign ranking  $r$  to bin  $b$  as  $b(r) = \text{round}(c \log(r + 1))$ ,  $1 \leq b \leq R$ , where  $c$  is a parameter chosen to be 3.322 here so that the Top 1 and



**Fig. 2** Popularity distribution in PPTV.

Top 2 videos, i.e., the hottest two videos, that have a huge popularity gap can be separated into two bins, i.e.,  $b(1) = \text{round}(c \log(1 + 1)) = 1$  and  $b(2) = \text{round}(c \log(2 + 1)) = 2$ . Accordingly, all the rankings are packed into 15 bins here, i.e.,  $R = 15$ . In this way, the resulted bins, as schematically illustrated in Fig. 1 (b), contain sessions much more evenly than those in the linear assignment. Specifically, the variance of the popularity distribution between bins in our log-assignment scheme is 0.05 while that in the linear-assignment scheme is 0.12.

Accordingly, we assign each user a  $R$ -dimension vector of *Popularity Preference (PP)*,  $\mathbf{p}_u = \langle p_{u,1}, \dots, p_{u,i}, \dots, p_{u,R} \rangle$ , where  $p_{u,i}$  denotes the proportion of the videos in bin  $i$  preferred by user  $u$ . If user  $u$  has not watched any videos in bin  $i$ , the probability  $p_{u,i} = 0$ . For  $n$  users, we get a PP matrix  $\mathbf{P}$  of  $n \times R$  dimensions where a row represents the PP vector of a user. The resulted *user-bin* PP matrix is much denser and more reliable than the original *user-video* rating matrix in the sparsity situations, so that it can be used as auxiliary information to make up the bias of traditional CF recommendation.

Besides, our proposal to utilize user PP matrix as auxiliary information is practical as it does not require any extra data sources. On the other hand, compared with the default popularity filling schemes, the utilization of user PP is more beneficial to meet the personal interest of the users.

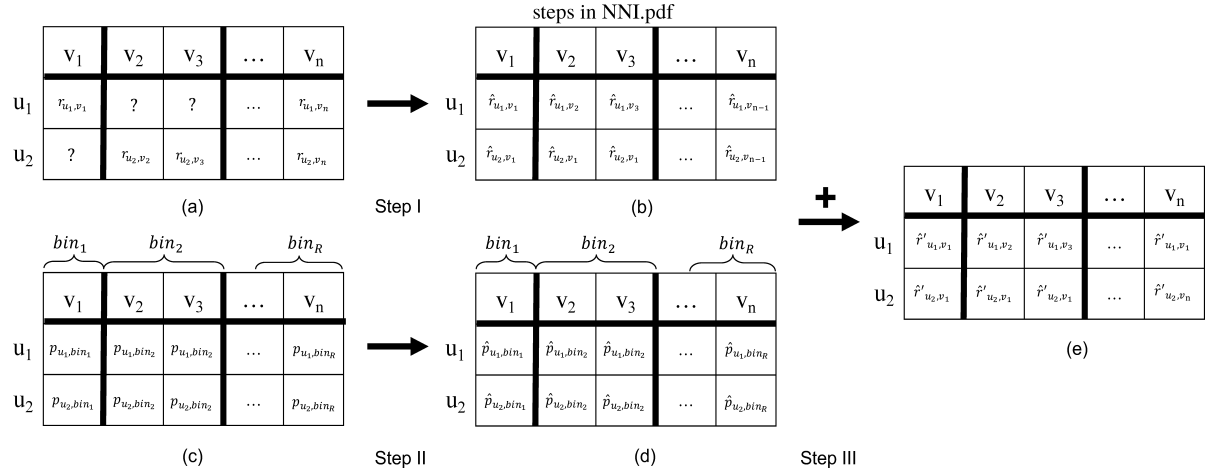
## 5. Popularity Preference Inference Enhanced Recommendation Algorithm

### 5.1 Problem Statement

The basic idea of our algorithm is to introduce users' PPs as auxiliary information into the traditional CF algorithm to enhance the performance of the personalized recommendations for users. Formally, the proposed algorithm can be expressed to be

$$\hat{r}'_{u,i} \leftarrow f(\hat{r}_{u,i}, p_{u,\text{bin}(i)}) \quad (1)$$

where,  $\hat{r}'_{u,i}$  denotes the final predicted rating of video  $i$  assigned by user  $u$ ,  $\hat{r}_{u,i}$  is the original rating predicted by the traditional CF algorithm, and  $p_{u,\text{bin}(i)}$  is the user's preference rating to the bin of video  $i$ .  $f(\cdot)$  is a general function, e.g., the average or polynomial function, that integrates the two parts of the ratings.



**Fig. 3** The three steps in PPI-NN algorithm, including I) original rating prediction: produce (b) a predicted rating matrix  $\hat{R}_{u,i}$  based on (a) the historical ratings matrix  $\tilde{R}_{u,i}$ , II) PP rating inference: produce (d) an inferred PP matrix  $\hat{P}_{u,bin(i)}$  based on (c) the statistical PP matrix  $\hat{P}_{u,bin(i)}$ , and III) rating integration: produce (e) the integrated prediction rating matrix  $\hat{R}'_{u,bin(i)}$ .

To make the algorithm effective as well in the face of quite limited historical records, e.g., for the inactive users or most users at the very beginning of a video streaming system, recommendation more accurate, we need also an algorithm to correct the statistical users' PPs. In these cases, the statistics of users' PPs are likely to be biased, which would limit the significance of our proposed algorithm. Thus, it is necessary to propose an inference algorithm to improve the accuracy of the users' PPs from the observed ones.

## 5.2 PPI-NN/PPI-MF Algorithm

Specifically, we propose two recommendation algorithms including the following three steps: 1) original rating prediction, 2) PP rating inference, and 3) rating integration, as illustrated in Fig. 3.

### • Step I. Original rating prediction

In Step I, we predict users' original ratings (as referred to the matrix in Fig. 3 (b)) based on the limited historical ratings (as referred to the matrix in Fig. 3 (a)) using two typical CF algorithms,  $k$ -Nearest Neighbor (kNN) and Matrix Factorization (MF). The former algorithm is commonly used in reality due to its simplicity, efficiency, and stability [4], and the later one is good at dealing with data sparsity and performs quite well in the Netflix prize.

Similar to the typical kNN algorithm, we first select  $k$  neighbor users,  $N_u$ , who have the largest similarities with the object user  $u$ . To this end, we apply a Pearson's correlation together with Case Amplification to measure the similarity, as defined in literature [28]. This similarity metric has been proved to outperform other typical similarity measures like Jacard similarity, cosine similarity, and adjusted cosine similarity [29].

Moreover, we refine the similarity by multiplying the Pearson's correlation by a Significance Weighting factor [30] to value the trust of the correlations. The Sig-

nificance Weighting factor is set to be the number of the commonly-watched videos between users.

Then, we select  $k$  users with the largest similarities with user  $u$  to be the neighborhood to user  $u$ . We set  $k$  to be 20 according to experiment experience. Thus, the rating is predicted to be the weighted average rating of the neighbor users. The weights are the similarity defined above. That is, the rating of video  $i$  for user  $u$ ,  $\hat{r}_{u,i}$ , is predicted to be

$$\hat{r}_{u,i} = \frac{\sum_{v \in N_{u,i}} \omega_{u,v} \tilde{r}_{v,i}}{\sum_{v \in N_{u,i}} \omega_{u,v}} \quad (2)$$

where, the  $N_{u,i}$  are the neighbors of user  $u$  who have watched video  $i$ ,  $\tilde{r}_{v,i}$  is the observed rating of video  $i$  by user  $v$  and  $\omega_{u,v}$  is the similarity of the ratings between user  $u$  and  $v$ .

In term of the MF algorithm, we directly apply the typical algorithm elaborated in literature [19]. Due to the space limit, we would not elaborate this mature algorithm which can be referred to the literature [19].

### • Step II. PP rating inference

In order to get more accurate users' PPs (as referred to the matrix in Fig. 3 (d)) from their quite limited historical data, we proposed a NN-based PP inference algorithm, as illustrated in our earlier work [13], to make up the bias of the statistical users' PPs (as referred to the matrix in Fig. 3 (c)).

In our PP inference algorithm, the similarity between users is measured by the cosine similarity [3] of their PPs. With a similarity larger than a threshold, two users will be regarded to be neighbors of each other and the inferred PP of user  $u$  is the weighted average of the neighbor users and user  $u$  herself, as

$$\hat{p}_{u,bin(i)} = \frac{\sum_{v \in N_{u,i}^{(p)}} \omega_{u,v}^{(p)} \tilde{p}_{v,bin(i)} + \tilde{p}_{u,bin(i)}}{\sum_{v \in N_{u,i}^{(p)}} \omega_{u,v}^{(p)} + 1} \quad (3)$$



where,  $N_{u,i}^{(p)}$  are the neighbors of user  $u$  who have similar PPs with user  $u$  in the statistic of their historical records. The similarity threshold for neighbor selection,  $\tau$ , is determined by experiment.  $\tilde{p}_{v,bin(i)}$  is the observed PP of user  $v$  and  $w_{u,v}^{(p)}$  is the PP similarity between user  $u$  and  $v$ . For clarity, the PP rating is linearly scaled to be in the range of (0, 10) in according with the scale of the *user-video* interest rating.

Different from original kNN algorithm, the observed PP of user  $u$ 's own is also considered in the predicted PP and the corresponding weight  $w_{u,u}^{(p)}$  is set to be 1.

- *Step III. Rating integration*

In Step III, we generate an integrate predicted rating as referred to the matrix in Fig. 3 (e), using the original rating in Step I and the inferred PP rating in Step II. The integration function  $f(\cdot)$  in Eq. (1) can be chosen to be like polynomial function or even complex machine learning algorithm. As an example, we apply an weighted additive function which is simple and interpretable. Accordingly, the integrated rating  $\hat{r}'_{u,i}$  is predicted to be

$$\hat{r}'_{u,i} = (1 - \gamma) \hat{r}_{u,i} + \gamma \hat{p}_{u,bin(i)} \quad (4)$$

where,  $\gamma$  is the weight of the PP rating to be determined by experiments. Finally, the Top- $N$  videos sorted by the predicted ratings on descending order will be recommended to this user.

For clarity, when the original rating in Step I is predicted by KNN and MF respectively, we name the proposed recommendation algorithms as *Popularity Preference Inference enhanced Nearest Neighbor (PPI-kNN)* algorithm and *Popularity Preference Inference enhanced Matrix Factorization (PPI-MF)* algorithm respectively.

Even if some other alternative forms of the integration function might make the prediction more accurate, our algorithms have strengths in interpretation and computational effect. If necessary, any other forms of function could easily apply as well.

## 6. Evaluation

In this section, we evaluate our proposed Top- $N$  recommendation algorithms compared with some baseline methods in terms of both accuracy and coverage under various recommend video number.

### 6.1 Performance Metrics

**Accuracy.** We evaluate the accuracy of the algorithms by using the metric Precision at  $N$  [27]. Precision at  $N$  for each user is defined to be the ratio of the properly recommended videos (that have been recommended and really been watched by the user) to the recommended video number  $N$  for the user. For the whole system, we will calculate the average precision at  $N$  for users in the system.

**Coverage.** As another important performance metric related to the interest of content providers, the term coverage

can refer to several distinct properties of the system [27]. We here define the term coverage at  $N$  to be the total number of the videos recommended in the whole system when  $N$  videos are recommended to each user. For clarity, we further normalize it by the total video number in the whole system. Obviously, the larger coverage, the more opportunities for more videos to be shown to potential users.

### 6.2 Baseline Methods

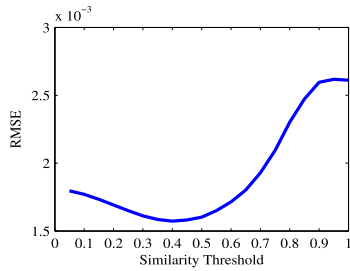
We evaluate our proposed algorithms together with some baseline methods. Besides original kNN and MF algorithms which make recommendations directly depending to the original rating in Step I, we propose the following four baseline algorithms as well.

- *Statistical Popularity Preference enhanced k-Nearest Neighbor (SPP-kNN) recommendation algorithm.* It predicts ratings using the original ratings predicted by KNN and the user PP ratings directly based on their statistical history but without PP inference.
- *Global Popularity Preference enhanced k-Nearest Neighbor (GPP-kNN) recommendation algorithm.* Besides the original ratings predicted by KNN, it considers the users generally have the same popularity preference. In other words, it takes into account the average popularity preference in the global system instead of the users' own popularity preferences.
- *Statistical Popularity Preference enhanced Matrix Factorization (SPP-MF) recommendation algorithm.* Similar to SPP-kNN, it makes recommendations using the original ratings predicted by MF and the statistical user PP ratings without PP inference.
- *Global Popularity Preference enhanced k-Nearest Neighbor (GPP-kNN) recommendation algorithm.* Similar to GPP-kNN, besides the original ratings predicted by MF, it uses the average popularity preference of all the users in the global system.

### 6.3 Experiments Settings

We randomly sample certain ratio of our dataset for training and the rest for test. To evaluate the algorithms under diverse sparsity scenarios, we vary the proportion for training in separate experiment rounds. To address the capacity of our algorithms in the quite early or sparse situation, we discuss the evaluation results under the training ratio below 50% in this paper.

We first determine the optimal similarity threshold  $\tau$  in Step II to optimize the PP inference accuracy which is measured by Root Mean Square Error (RMSE). In Fig. 4, we present the changes of the accuracy against different threshold values when the training ratio is set to be 20%. As shown, the accuracy curve fluctuates and gets highest at  $\tau$  of 0.4. Such a fluctuation is rational since a too small  $\tau$  might introduce the irrelevant information, while a too large one could not maintain the neighborhood size big enough.



**Fig. 4** PP inference accuracy against the varying similarity threshold,  $\tau$ , in the PP rating inference.

**Table 1** Optimal similarity threshold,  $\tau$ , in Step II changes against training ratios.

Training Ratio (%)	5	10	20	30	40	50
$\tau$	0.01	0.2	0.4	0.6	0.8	1

**Table 2** The optimal parameter  $\gamma$  for the proposed PPI-kNN algorithm and PPI-MF algorithm, together with the baseline SPP-kNN, SPP-MF, GPP-kNN and GPP-MF, respectively changes against the training ratio. The recommend number  $N$  is set to be 20 as an example.

Algorithm \ Training Ratio (%)	10	20	30	40
PPI-kNN	0.08	0.1	0.08	0.06
SPP-kNN	0.07	0.08	0.05	0.01
GPP-kNN	0.001	0.001	0.001	0.001
PPI-MF	0.04	0.03	0.03	0.02
SPP-MF	0.03	0.015	0.009	0.005
GPP-MF	0.001	0.001	0.001	0.001

We further list the optimal  $\tau$  under different training ratios below 50% in Table 1, while when training ratio is above 50%, the optimal  $\tau$  stays at 1. As shown, with a larger training ratio, the optimal  $\tau$  gets larger, suggesting a stricter restriction to select the neighbors when we can get more historical records. In our earlier work in [13], more detailed evaluation results of the PP inference under optimal  $\tau$  are elaborated. We have proved that the proposed NN-based PP inference algorithm outperforms the statistical methods significantly when the training ratio is set below 48%, and performs as well as the statistical one when the training ratio is larger than 50%.

With the optimal  $\tau$ , we have tuned the weight  $\gamma$  in Eq. (5) for the proposed PPI-kNN algorithm and PPI-MF algorithm, together with the baseline SPP-kNN, SPP-MF, GPP-kNN and GPP-MF respectively. We vary the weight from 0.001 to 1 in experiment rounds. A larger weight means more significance of the improvement on the recommendation accuracy could be brought by users' popularity preference (or the global popularity preference in the GPP-kNN algorithm). Its optimum value changes against the recommended video number  $N$  and the training ratio. We list the optimum value at  $N$  of 20 in Table 2. As shown, the weight decreases with the training ratio growing, which means the knowledge of user PP is more useful in a more sparse dataset.

**Table 3** The average of the accuracy improvements (%) made by the proposed algorithms, i.e., PPI-kNN and PPI-MF, over their corresponding baseline algorithms when  $N$  varies from 10 to 200.

Algorithms \ Training Ratio (%)	10	20	30	40
PPI-kNN vs. kNN	5.8	4.0	3.1	2.5
PPI-kNN vs. SPP-kNN	1.2	2.6	3.2	3.0
PPI-kNN vs. GPP-kNN	6.8	31.5	51.1	63.5
PPI-MF vs. MF	2.5	3.2	3.3	2.7
PPI-MF vs. SPP-MF	0.7	1.3	2.0	2.2
PPI-MF vs. GPP-MF	4.8	24	34	40.9

## 6.4 Evaluation Results

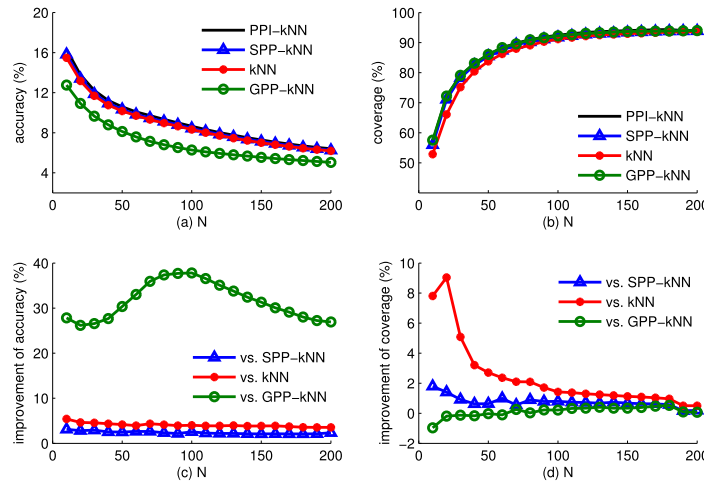
With the optimal parameters, our proposed PPI-kNN and PPI-MF algorithms outperform all the baseline methods in terms of the recommendation accuracy when the training ratio is below 48%. They also have high coverage than the baseline method except the GPP-kNN and GPP-MF respectively.

As an example, we plot the accuracy and coverage of the proposed algorithm, PPI-kNN, together with the baseline algorithms, i.e., kNN, GPP-kNN and SPP-kNN, under the training ratio of 20%. As shown in Fig. 5 (a), the PPI-kNN algorithm has improvement over the other baseline methods when the recommended number  $N$  changes from 10 to 200. Generally, the accuracy decreases with a larger  $N$ , since the majority of the users actually have not viewed as many videos as the large recommended number  $N$ . For clarity, we plot the improvements of the PPI-kNN algorithm on the baseline methods in Fig. 5 (c). On average, the accuracy improvement achieves 2.6% over the SPP-kNN algorithm, 31.5% over the GPP-kNN algorithm and 4.0% over the original kNN algorithm.

We further list the accuracy improvements made by the proposed PPI-kNN algorithm and the PPI-MF algorithm over their corresponding baseline algorithms in Table 3. The improvements change against different training ratio from 10% to 40%, while with a training ratio above 50% the baseline algorithms except GPP-kNN and GPP-MF performs comparable to the proposed algorithms. The results listed are the average accuracy when  $N$  varies from 10 to 200.

As shown, the improvements change against the training ratio. With a larger training ratio, the improvement made by the PPI-MF algorithm decreases over the original kNN algorithm, which accords with the earlier suppose that PP rating plays more important role in a sparser context. Provided with more training data, the original CF algorithms is more reliable and the superiority brought by user PP is less significant. However, compared with GPP-kNN, the improvement gets more significant. It is rational because with more training data the PP inference brings much more enhancement in prediction accuracy.

Furthermore, there is also an improvement made by the PPI-MF algorithm although it is not as large as that by PPI-kNN algorithm under a training ratio below 50%. It may



**Fig. 5** The performances of the proposed PPI-kNN algorithm and the corresponding three baseline methods, i.e., original kNN, SPP-kNN and GPP-kNN, in terms of (a) accuracy and (b) coverage under training ratio of 20%. The improvements made by the PPI-kNN algorithm over the other baseline algorithms in terms of (a) accuracy and (b) coverage.

because the original MF algorithm itself has an advantage in face with the sparsity. However, it would not degrade the advantages of our algorithm in dealing with the sparsity problem.

When it comes to coverage, the PPI-kNN algorithm outperform most of the baseline algorithms except the GPP-kNN one, as shown in Fig. 5 (b). It may because users' own preferences usually concentrate on one or several popularity bins, while the global preferences disperse in every bin, so that the GPP-kNN is likely to recommend more diverse videos.

We further present the improvement of the PPI-kNN algorithm over other baseline methods in Fig. 5 (d). As shown, the improvement over the original kNN algorithm is as significant as around 8% at the recommended number  $N$  below 20. It means that, with the knowledge of user PP, only a few recommendations could cover more videos over the baselines, which satisfies the interest of the providers. On the above, the positive results prove that the knowledge of users' PPs could help handle with the sparsity problem in recommendation systems in terms of both accuracy and coverage, and that the accurate inference of users' PPs brings more significant benefit when we have only a few historical records.

## 7. Conclusion

In this paper, we modify two traditional CF recommendation algorithm via introducing the inferred PP to tackle with the sparsity problem. In the modified algorithm, we utilize the aggregated information of user preference on video popularity to make up the inaccuracy of the CF algorithms caused by the insufficiency of the training data. To enhance the accuracy, we propose a CF based algorithm to make the inference of users' PPs effective and efficient.

Our experiment on a large-scale dataset from PPTV

proves that, facing with the sparsity problem, the proposed PPI-kNN algorithm and the PPI-MF algorithm could improve both the recommendation accuracy and coverage. For example, when training ratio is 20%, the accuracy improvement made by PPI-kNN achieves 2.3% over the SPP-kNN algorithm, 34.1% over the GPP-kNN algorithm and 4.3% over the original kNN algorithm.

Different from some other typical methods that requires auxiliary information from extra data, e.g., video content and users' behavior in other applications, to deal with the sparsity problem, our algorithm is more simple, efficient and practical. Furthermore, compared with the default popularity filling schemes, the utilization of user PP is more beneficial to meet the personal interest of the users.

## Acknowledgments

This work was supported in part by the National Science Foundation of China under Grant No. 61572071, 61271199 and 61301082.

## References

- [1] "Cisco Visual Networking Index: Forecast and Methodology," 2011-2016, May 30, 2012.
- [2] <http://www.youtube.com>
- [3] X. Amatriain and J.M. Pujol, Data Mining Methods for Recommender Systems, Recommender Systems Handbook, Springer US, pp.39–71, 2015.
- [4] X. Ning, C. Desrosiers, G. Karypis, "A Comprehensive Survey of Neighborhood-Based Recommendation Methods," Recommender Systems Handbook, Springer US, pp.107–144, 2015.
- [5] A.I. Schein, et al., "Generative Models for Cold-Start Recommendations," Proceedings of the Sigir Workshop on Recommender Systems, 2001.
- [6] S. Roy and S.C. Guntuku, "Latent Factor Representations for Cold-Start Video Recommendation," ACM Recsys ACM, pp.99–106, 2016.
- [7] Z. Li, J.-Y. Peng, G.-H. Geng, X.-J. Chen, and P.-P. Zheng,

- "Video recommendation based on multi-modal information and multiple kernel," *Multimedia Tools and Applications*, vol.74, no.13, pp.4599–4616, 2015.
- [8] M. Yan, J. Sang, and C. Xu, "Unified YouTube Video Recommendation via Cross-network Collaboration," *ACM on International Conference on Multimedia Retrieval*, pp.19–26, 2015.
- [9] M. Yan, J. Sang, C. Xu, and M.S. Hossain, "A Unified Video Recommendation by Cross-Network User Modeling," *ACM Transactions on Multimedia Computing Communications and Applications*, vol.12, no.4, pp.1–24, 2016.
- [10] W. Xia, L. He, L. Ren, M. Chen, and J. Gu, "A new collaborative filtering approach utilizing item's popularity," *IEEE International Conference on Industrial Engineering and Engineering Management IEEE*, pp.1480–1484, 2008.
- [11] H. Ahn, "Utilizing Popularity Characteristics for Product Recommendation," *International Journal of Electronic Commerce*, vol.11, no.2, pp.59–80, 2006.
- [12] S. Goel, A. Broder, E. Gabrilovich, and B. Pang, "Anatomy of the long tail: ordinary people with extraordinary tastes," in *proceedings of the third ACM international conference on Web search and data mining*, pp.201–210, 2010.
- [13] X. Tan, Y. Guo, Y. Chen, and W. Zhu, "Accurate inference of user popularity preference in a large-scale online video streaming system," *Science China Information Sciences*, vol.61, no.1, pp.1–3, 2018.
- [14] J. Oh, S. Park, H. Yu, M. Song, and S.-T. Park, "Novel recommendation based on personal popularity tendency," in *proceedings of the 11th international conference on data mining (ICDM)*, pp.507–516, 2011.
- [15] <http://www.pptv.com>
- [16] X. Su and T.M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol.2009, pp.1–19, 2009.
- [17] M. Deshpande and G. Karypis, "Item-based top-N recommendation algorithms," *Acm Transactions on Information Systems*, vol.22, no.1, pp.143–177, 2004.
- [18] L. Si and R. Jin, "Flexible Mixture Model for Collaborative Filtering," *Machine Learning, Proceedings of the Twentieth International Conference DBLP*, pp.704–711, 2003.
- [19] Y. Koren, R. Bell, and C. Volinsky, *Matrix Factorization Techniques for Recommender Systems*, *Computer*, vol.42, no.8, pp.30–37, 2009.
- [20] P. Arkadiusz, "Improving regularized singular value decomposition for collaborative filtering," *Proceedings of Kdd Cup and Workshop*, 2007.
- [21] <http://www.netflixprize.com>.
- [22] L. Iaquinta, G. Semeraro, and P. Lops, *Cold Start Problem: A Lightweight Approach, New Challenges in Distributed Information Filtering and Retrieval*, Springer Berlin Heidelberg, vol.439, pp.15–32, 2013.
- [23] S. Ouyang, C. Li, and X. Li, "A Peek Into the Future: Predicting the Popularity of Online Videos," *IEEE Access*, vol.4, pp.3026–3033, 2016.
- [24] Y. Zhou, L. Chen, C. Yang, and D.M. Chiu, "Video popularity dynamics and its implication for replication," *IEEE Trans. Multimedia*, vol.17, no.8, pp.1273–1285, 2015.
- [25] J. Yue, B. Yang, C. Chen, X. Guan, and W. Zhang, "Femto-caching in Video Content Delivery: Assignment of Video Clips to Serve Dynamic Mobile Users," *Computer Communications*, vol.51, pp.60–69, 2014.
- [26] M. Bouchard, A.-L. Jousselme, and P.-E. Doré, "A proof for the positive definiteness of the Jaccard index matrix," *International Journal of Approximate Reasoning*, vol.54, no.5, pp.615–626, 2013.
- [27] G. Shani and A. Gunawardana, "Evaluating Recommendation Systems," *Recommender Systems Handbook*, pp.257–297, 2011.
- [28] D. Lemire and A. Maclachlan, "Slope One Predictors for Online Rating-Based Collaborative Filtering," *Computer Science*, pp.21–

23, 2012.

- [29] J.S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," *Uncertainty in Artificial Intelligence*, vol.98, no.7, pp.43–52, 2013.
- [30] P. Melville, R.J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," *Eighteenth National Conference on Artificial Intelligence*, pp.187–192, 2002.



**Xiaoying Tan** received her B.S. in Communications Engineering from Central South University in 2008. She is currently a Ph.D student in the School of Electronics and Information Engineering at Beijing Jiaotong University.



**Yuchun Guo** received her M.S. degree from Beijing Normal University in 1993, and her Ph.D. degree from the Beijing Jiaotong University in 2008. She is currently a professor at the Beijing Jiaotong University. She spent an academic year at Delft University of Technology, the Netherlands, from 2001 to 2002, as a visiting scientist.



**Yishuai Chen** received his B.S., M.S., and Ph.D. degree from School of Electronics and Information Engineering at the Beijing Jiaotong University, in 1998, 2001, and 2010, respectively. He is currently a lecturer at the School of Electrical and Information Engineering of the Beijing Jiaotong University. From 2010 to 2012, he was a Postdoctoral Fellow with the Research Center of Ubiquitous Sensor Networks at the Chinese Academy of Sciences, Beijing, China. From 2001 to 2007, he worked at Lucent Bell-

labs (China) on intelligent network systems as a Technical Staff member. He has served as a TPC member for IEEE GLOBECOM.



**Wei Zhu** received his B.S. degrees from Huazhong University of Science and Technology in 2007. He was a manager at PPTV Inc. when he participated in this work. He is currently a senior expert at Alibaba Inc.