Cold-Start Recommendation for On-Demand Cinemas

Beibei Li[1,2], Beihong Jin[1,2](✉), Taofeng Xue[1,2], Kunchi Liu[1,2], Qi Zhang[3], and Sihua Tian[3]

[1] State Key Laboratory of Computer Sciences, Institute of Software, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
Beihong@iscas.ac.cn
[3] Beijing iQIYI Cinema Management Co., Ltd.

Abstract. The on-demand cinemas, which has emerged in recent years, provide offline entertainment venues for individuals and small groups. Because of the limitation of network speed and storage space, it is necessary to recommend movies to cinemas, that is, to suggest cinemas to download the recommended movies in advance. This is particularly true for new cinemas. For the new cinema cold-start recommendation, we build a matrix factorization framework and then fuse location categories of cinemas and co-popular relationship between movies in the framework. Specifically, location categories of cinemas are learned through LDA from the type information of POIs around the cinemas and used to approximate cinema latent representations. Moreover, a SPPMI matrix that reflects co-popular relationship between movies is constructed and factorized collectively with the interaction matrix by sharing the same item latent representations. Extensive experiments on real-world data are conducted. The experimental results show that the proposed approach yields significant improvements over state-of-the-art cold-start recommenders in terms of precision, recall and NDCG.

Keywords: Recommendation system · On-demand cinema · Cold-start Problem · Matrix factorization · Location category · Co-popular relationship.

1 Introduction

The on-demand cinemas are a new type of offline entertainment venues for individuals and small groups of audiences. At such a cinema, audiences can choose the favorite movies from local copies of copyrighted movies and watch them. In the recent two years, the number of on-demand cinemas has been increasing

This work was supported by the National Natural Science Foundation of China (No. 61472408) and the joint project with iQIYI (No. LUM18-200032).

rapidly. In China, on-demand cinemas have covered 29 provinces and 92 core cities.

The movies played at on-demand cinemas are ones with ultra-clear HD pictures and HiFi sounds, which usually consume considerable storage space. For example, a Blu-ray HD movie may take up 40-50GB. In order to ensure the watching experience and to avoid the display lag phenomena caused by the network bandwidth limitation, on-demand cinemas usually download movies in advance and store movies locally. However, due to the limited local storage space, on-demand cinemas must choose some potential popular movies from numerous candidates, and then download them for audiences in advance. That is, the local movie libraries of on-demand cinemas will directly influence choices and consumption trends of audiences.

For on-demand cinemas, Xue et al. [19] establish a spatial-temporal recommendation system, which utilizes the history watching records of on-demand cinemas and recommend the movies to the staffs/hosts of on-demand cinemas to download the movies into local movie libraries. However, the above system is only suitable for cinemas that are already in operation and have history watching records. For newly-opened on-demand cinemas, the initial movie libraries will directly affect the operation of the cinemas at the beginning, which further involves the survival of on-demand cinema business. Therefore, it is important to provide an initial movie recommendation list for newly-opened on-demand cinemas.

Recommending movies to new cinemas, i.e., cinema cold-start problem, can be analogous to the user cold-start problem in the traditional recommendation systems. That is, the recommendation objects are on-demand cinemas in our scenarios, and recommendation items are movies.

We notice that in the on-demand cinema scenarios, the surrounding environment of a cinema will affect the composition of audiences. For example, those who live or work around a cinema tend to visit the cinema. Moreover, if the cinema is in the business district, the colleagues are more likely to watch movie together; if the cinema is at living areas, then it is more likely that the family members watch movie together. Therefore, we can explore the point of interest (POI) type information around each new cinema to predict the popularity of each movie in each new cinema. In addition, high-quality movie representations are helpful in dealing with the cinema cold-start problem. Mining the correlation over popularity of two movies is a feasible way.

Based on the above considerations, we establish a basic framework of cold-start model based on matrix factorization, and then fuse side information to serve the cold-start of the on-demand cinemas from two aspects. On the one hand, from the POI type information around the cinemas, we learn location category probability distribution of the cinemas through the latent Dirichlet allocation (LDA), and further apply the linear transformation to the learned distribution to obtain the latent representations of cinemas. On the other hand, we define the movie co-popular relationship and find co-popular contexts of movie pairs, from which we build the shifted positive pointwise mutual information (SPPMI) matrix. Finally, we collectively factorize the SPPMI matrix and the cinema-movie interaction matrix factorization by sharing the latent representations of movies to further improve the quality of movie representations. By conducting extensive experiments on real-world datasets of on-demand cinemas, we demonstrate that our proposed approach yields significant improvement compared with other state-of-art cold-start recommenders in terms of precision, recall, and NDCG.

Our contributions are summarized as follows:

First, we devise a matrix factorization-based model to tackle the cold-start problem of on-demand cinemas, which explores the influence of cinema location category and co-popular relationship between movies. To the best of our knowledge, it is the first work to attempt to solve the on-demand cold-start problem.

Second, extensive experiments performed on real-world datasets demonstrate the effectiveness of our proposed approach. Besides, we evaluate the performance contributions of key components of the proposed approach, including the SPPMI matrix, the location category probability distribution vector learned by LDA and the frequency-based weights.

Third, the proposed approach is also applicable to the warm-start recommendation of on-demand cinemas and the cold-start scenarios of offline stores. Furthermore, the SPPMI matrix component reflecting co-popular relationship between movies can be generalized to any scenario with co-popular relationship between items and serve as a separate module adding to other matrix factorization-based recommendation approaches.

2 Related Work

This work is broadly related to the research in two nonorthogonal categories: cold-start recommendation, embedding techniques.

2.1 Cold-Start Recommendation

The existing methods to solve cold-start recommendation for users can be roughly divided into two categories: interview-based methods and side information based methods [13].

The interview-based methods [13] need to design some representative questions to ask cold-start users and then recommend them based on their feedback. This type of methods consumes a lot of time and manpower, which are not suitable for on-demand cinemas.

Besides directly inquiring cold-start users, side-information including social network information, user profile information or other cross-domain information can be used to solve the problem of cold-start recommendation for users. These methods are called side information based methods. These methods can be further divided into three kinds [9]: similarity based methods, matrix factorization methods, and feature mapping methods.

Recommending based on the similarities between the cold-start user and the other users is the common idea in the similarity based methods. However, the method of calculating the similarity is multifarious. For example, Suvash et al. [15] measure the similarity between users by using cosine similarity of user side information vectors. Peng et al. [11] construct a parameterized Mahalanobis distance metric using user side information, and learns the parameters of the Mahalanobis distance from the training data, finally, use the learned Mahalanobis distance to measure the similarity between users.

Matrix factorization methods integrate side information on the basis of the interaction matrix factorization. For example, the CMF (Collective Matrix Factorization) model [14] factorizes the interaction matrix and side information jointly by sharing the user hidden vector U and learns the linear mapping function from the user latent representations to the side information. The CTR (Collaborative Topic Regression) model [17] first learns the topic distribution from the content of the article through the LDA [1] model, and then adds Gaussian error to the latent space representations of the articles to solve the cold-start problem of article recommendation.

The core of the feature mapping methods is to learn the transfer function of side information to user hidden space. The main difference between it and collective matrix factorization is that the latter is to learn user latent vector and mapping function simultaneously, while the former has a separate process of learning feature mapping. For example, the BRP-linMap algorithm proposed by Zeno et al. [3] learns a linear feature mapping function from side information to user latent vectors. For a variety of methods to solve the cold-start problem with side information, Sedhain et al. [12] unify them in a linear transformation framework, and propose a low-rank linear regression model LOCO using the side information matrix and the interaction matrix.

2.2 Embedding Techniques

In a recommendation system, the embedding techniques can be used to learn the hidden vectors of users and items, thereby improving the performance of the recommendation results [10, 18, 20].

One way is to express relationships between users or items or otherwise via a graph and then apply graph embedding techniques to learn the latent vectors of users or items. For example, Xie et al. [18] construct four bipartite graphs such as POI-POI, POI-region, POI-time slots and POI-content words for POI recommendation and the learned latent vectors characterize the four effects, i.e., sequential effect, geographical influence, temporal cyclic effect, and semantic effect in POIs. Yu et al. [20] establish a heterogeneous information network based on the potential friend relationships between users and the purchase relationships between user and item to learn the user latent representations. Then, recommendation are made based on the similarity between user latent representations.

Word embedding techniques utilize the contextual information between words to learn the low-dimension representations of words. The techniques have been successful in the field of natural language processing [4] and have recently been introduced into recommendation systems. Grbovic et al. [4] regard the product in a product recommendation system as a word, the product list consumed by a user as a sentence, and propose prov2vec algorithm based on word2vec. Levy et al. [8] demonstrate that the skip-gram negative sampling (SGNS) model for word embedding is equivalent to decomposing the shifted positive pointwise mutual information (SPPMI) matrix. Inspired by this, Liang et al. [10] propose a collective matrix factorization method, which combines the user-item interaction matrix and the SPPMI matrix of the item on the basis of the shared item latent vectors. For calculating the SPPMI matrix, the item lists that have been interacted with by the same user can be regarded as a sentence, each item in the list is regarded as a word, and each item in the list is regarded as context for each other. Subsequently, Cao et al. [2] adopt the similar method to generate the SPPMI matrix. Thus, while the decomposition is being performed, the embeddings of items and lists are simultaneously learned, and the recommendation of item and item list are solved. In addition, for watching video scenarios, Than et al. [16] first define user context, co-like context and co-dislike context, and then obtain the corresponding three SPPMI matrices and perform regularized matrix factorization while fusing with three SPPMI matrices to recommend videos.

Different from the existing work, we combine matrix factorization and item embedding techniques for cold-start recommendation. On the one hand, we utilize location category distribution vectors of cold-start cinemas learned by LDA to obtain the latent representations of cinemas. On the other hand, we exploit co-popular relationship between movies and build the SPPMI matrix. Then, we collectively factorize the SPPMI matrix with the cinema-movie interaction matrix by sharing the latent vectors of the movies.

3 Approach

3.1 Formulation

Recommending movies to startup on-demand cinemas is the user cold-start recommendation problem, where we take a cinema as a user. We split the cinema set S into two disjoint parts: the set of warm-start cinemas S_{tr} whose elements have at least one watching record in the past t timeslots and the set of cold-start cinemas S_{te}, where $S_{tr} \cap S_{te} = \emptyset$. Let $R_{tr} \in \{0,1\}^{|S_{tr}| \times n}$ denote the interaction between warm-start cinemas in S_{tr} and n movies, where $|S_{tr}|$ is the number of warm-start cinemas, and if there is interaction between $i-$th cinema and $j-$th movie, then the element in R_{tr}, i.e., $r_{ij} = 1$. Otherwise $r_{ij} = 0$. Let $X_{tr} \in \mathbb{R}^{|S_{tr}| \times d}$ indicate d-dimension side information of warm-start cinemas and $X_{te} \in \mathbb{R}^{|S_{te}| \times d}$ indicate d-dimension side information of cold-start cinemas. Our goal is to predict the interaction matrix $\hat{R}_{te} \in \mathbb{R}^{|S_{te}| \times n}$ between cinemas in S_{te} and n movies based on R_{tr},X_{tr} and X_{te}.

3.2 Basic Framework

Matrix factorization is a typical technique for recommendation. Its basic idea is to leverage correlations between users and items with a k-dimension latent space.

The interaction matrix $R \in \mathbb{R}^{m \times n}$ between m users and n items is factorized into two low-rank matrices as $U \in \mathbb{R}^{k \times m}$ and $V \in \mathbb{R}^{k \times n}$, which are the latent representations of users and items respectively. The prediction of user-item pairs can be calculated by the inner dot of $U \in \mathbb{R}^{k \times m}$ and $V \in \mathbb{R}^{k \times n}$. Since learning latent representations rely on history interaction data, cold-start problem occurs, which is pervasive in real-world recommendation applications.

Our key point to solve cinema cold-start problem is to learn the cinema latent representations from side information of cinemas. Suppose that there is a function $f(\cdot)$ which can transform the side information of cinemas into their latent representations U_{tr}, which indicates $U_{tr} \approx f(X_{tr})$, then we can replace cinema latent representations U_{tr} by $f(X_{tr})$ in matrix factorization. Since there is side information X_{te} of cold-start cinemas U_{te}, latent representations of cold-start cinemas can be obtained directly with side information matrix X_{te} and transformation function $f(X_{te})$. Further, interaction matrix of cold-start cinemas can be predicted to solve the cold-start problem.

In order to recommending movies to cold-start cinemas, we construct the loss function as following:

$$L = C \odot (R_{tr} - \sigma(f(X_{tr})^T V))^2 + \lambda_f \|\Theta_f\|_F^2 + \lambda_v \|V\|_F^2 \tag{1}$$

Here $\odot$ denotes element-wise matrix multiplication, Θ_f is the parameters of function $f(\cdot)$. λ_f and λ_v are regularization coefficients, σ is sigmoid function, which transforms the prediction values into $(0, 1)$. In Equation (1), $C \in \mathbb{R}^{m \times n}$ is the confidence coefficient matrix, where larger c_{ij} indicates higher confidence. In on-demand cinema cold-start problem, interaction frequencies of each cinema-movie pair differ a lot from each other. Thus, c_{ij} is set by $c_{ij} = \log(freq_{ij} + 2)$ to reflect how popular movie j is for cinema i, where $freq_{ij}$ is the frequency that movie j is played at cinema i.

As for warm-start cinemas, we assume that their latent representations are close to $f(X_{tr})$ but could diverge from it if they have to, which can be learned from history data. Therefore, we add an offset $\epsilon \sim N(0, \lambda_\epsilon^{-1} I_k)$ on $f(X_{tr})$ as latent representations. That is to say, for a cinema i with side information vector X_i, its latent vector can be calculated as $U_i = f(X_i) + \epsilon_i$, for which the loss function turns into Equation (2),

$$L = C \odot (R_{tr} - \sigma(U_{tr}^T V))^2 + \lambda_\epsilon \|U_{tr} - f(X_{tr})\|^2 + \lambda_v \|V\|_F^2 + \lambda_f \|\Theta_f\|^2 \tag{2}$$

where λ_ϵ is the hyperparameter for the distribution of the offset.

The optimization objective of the problem is to minimize the loss value in Equation (2). After solving the objective function, we can obtain the parameters of transforming function $f(\cdot)$, and latent representations of movies V. Furthermore, as for cold-start cinemas, the offsets can be set as the expectation i.e., 0. Thus, we have $U_{te} = f(X_{te})$. Finally, we use Equation (3) to predict the interaction matrix, in which the larger value in $\hat{R}_{te}$ is, the higher the watching probability is.

$$\hat{R}_{te} = \sigma(f(X_{te})^T V) \tag{3}$$

During solving the objective function, latent representations of warm-start cinemas and latent representations of movies are learned, which can be used to predict the future interaction matrix of warm-start cinemas by $\hat{R}_{tr} = \sigma(U_{tr}^T V)$.

The above is our basic framework to solve cold-start cinema problem with side information. Its main idea is to learn cinema latent representations from their side information, thereby solving the cold-start problem. Meanwhile, the basic framework can recommend movies to warm-start cinemas, too. In particular, $f(X)$ is a mapping function to transform cinema side information into the latent space, which can be constructed according to actual conditions. The simplest way to construct $f(X)$ is to directly apply linear transformation on the side information matrix.

3.3 Location Categories of Cinemas

In on-demand cinema scenarios, the main audiences of a cinema come from the people who live or work around the cinema, for which the surrounding environment of the cinema and its location category is closely related to the profile of the potential audiences. POI type information around a cinema reflects its location category, so it can be taken as side information of cinemas. Further, we learn a probability distribution vector of cinema location category by LDA algorithm and transform it into the latent space with linear transformation.

Specifically, first, we count the POI types around cinemas. Given a cinema i and its latitude and longitude pair (lat_i, lng_i), we crawl POI information within 1km of the cinema from the Amap, group it by POI types and count the quantity of each type. Then we get the vector $X_i^{\text{raw}} \in \mathbb{R}^{m \times d}$, where d is the number of different POI types. After processing all cinemas in the same way, we obtain a POI type matrix $X^{\text{raw}} \in \mathbb{R}^{m \times d}$ in which x_{ij}^{raw} is the number of POIs around cinema i belonging to type j. We find that different types of POIs have orders of magnitude differences in quantity. Therefore, we smooth the POI type matrix with logarithmic function and obtain $X^{\log} \in \mathbb{R}^{m \times d}$, where $x_{ij}^{\log} = log_2 {x_{ij}}^{\text{raw}}$.

As an analogy, we regard a POI type as a word j, POIs around cinema i as a sentence and $x_{ij}^{\log}$ in matrix $X^{\log} \in \mathbb{R}^{m \times d}$ as the frequency of occurrence of word j in sentence i. Then, we learn location category distribution of cinemas reflected by POI type vectors via the LDA. We assume the following generative process:

For each cinema $i \in \{1, 2, \ldots, m\}$:

1. Draw the distribution of location category of cinema i, $\theta_i \sim Dirichlet(\alpha)$.
2. Draw the POI type distribution of the location category k, $\phi_k \sim Dirichlet(\beta)$.
3. For each POI around cinema i,i.e., w_{ij}:
   (a) Draw location category $z_{ij} \sim Mult(\theta_i)$.
   (b) Draw POI type $w_{ij} \sim Mult(\phi_{z_{ij}})$.

Based on the above generative process, we obtain the location category probability distribution of cinemas $X^{\text{lda}} \in \mathbb{R}^{m \times l}$, where m is the number of cinemas and l is the number of location category, which is a hyperparameter. The distribution matrix of training cinemas is denoted as X_{tr}^{lda} and the distribution matrix of cold-start cinemas is denoted as X_{te}^{lda}. We transform X_{tr}^{lda} into the cinema latent space with linear function $f(\cdot)$, where f is defined as Equation (4),

$$f(X) = \Theta_f^T X_{tr}^{\text{lda}} \tag{4}$$

In Equation (4), $\Theta_f \in \mathbb{R}^{l \times k}$, in which Θ_f is the parameters to learn, k is the dimension of the latent space.

3.4 Co-Popular Relationship between Movies

In matrix factorization based recommendation methods, regularizing with the SPPMI matrix when learning item embeddings can improve performance of the model [10, 16]. Inspired by this, we exploit co-popular patterns between two movies and form the corresponding SPPMI matrix. Then we use the SPPMI matrix to learn movie embeddings together with the interaction matrix to improve quality of movie embeddings. After adding SPPMI matrix to the basic framework, the optimization objective is as Equation (5).

$$\begin{aligned} \min_{\Theta_f, U_{tr}, V, Y} C \odot \| \left\| R_{tr} - \sigma\left(U_{tr}^T V\right) \right\|^2 + \left\| U_{tr} - f\left(X_{tr}^{\text{lda}}\right) \right\|^2 \\ + \alpha \left\| M^{\text{SPPMI}} - V^T Y \right\|^2 + \Omega\left(V, Y, \Theta_f\right) \end{aligned} \tag{5}$$

Where Y is the context embedding matrix of movies, in which ith column vector denotes the context embedding of movie i. Besides, $\Omega(V, Y, \Theta_f) = \lambda_v \|V\|^2 + \lambda_y \|Y\|^2 + \lambda_f \|\Theta_f\|^2$.

Specifically, during generating the SPPMI matrix, how to define two movies as the context of each other is a key point. We mine the co-popular relationship between two movies from history on-demand data and construct co-popular context among movies.

In detail, the co-popular relationship between movies is mainly considered from two aspects. On the one hand, we observe that the popularity of a movie has time effect. That is to say, audiences may have similar watching behaviors in a certain period, which may be caused by recent hotspot of the society. For example, if a popular cinema movie is released recently, audiences of on-demand cinemas will get more interested in the old movies related to the popular one, such as movies with the same director, in the same series or of the same type, etc. Therefore, related movies are more likely watched in the same period. On the other hand, in a certain period, the closer the frequencies of two movies are watched, the closer the popularities of the two movies are. For example, two movies watched with a similar frequency in the same period may be both hotpot movies in that period. Taking the above two factors into consideration, if two movies A and B are watched with close frequency in the same time window, A and B have co-popular relationship. If movie A and movie B have co-popular relationship and movie A is watched by audiences of a cinema, then we can infer that audiences of the cinema may also like movie B.

In order to get co-popular context of movies and form SPPMI matrix, we sort the different movies that are watched in the same time period in decreasing order according to watching times. In the sorted sequence, the similarly ranked movie pairs form co-popular contexts, from which we build the SPPMI matrix that reflects the movie co-popular relationship. Consider that the movie watched frequency shows a strong periodicity in weeks, we use a week as a time window to generate the frequency sorted sequence. The construction steps of co-popular context is as following, which is also presented in Figure 1.

1. Split the training data by weeks. Then, count the total watched frequency of each movie in each week and form $Q \in \mathbb{R}^{p \times n}$, where p is the number of weeks of training data, n is the number of movies, q_{ij} is the total watched frequency of movie j in $i-$th week.
2. Sort movies by the total watched frequency in decreasing order and record the corresponding movie number. That is to sort each row in Q decreasingly and form $G \in \mathbb{R}^{p \times n}$, where g_{ij} indicates that movie g_{ij} ranks at jth position in week i, e.g. g_{i1} denotes that movie g_{i1} is watched with most frequency in week i.
3. If we consider the movie as a word, the sorted movie list in week i can be considered as a sentence. The length of the sentence is the number of distinct movies watched in week i, denoted by $a_i = \Sigma_{j=1}^{n} I(q_{ij})$, where if $q_{ij} > 0$ then $I(q_{ij}) = 1$, otherwise $I(q_{ij}) = 0$. After processing data of p weeks in the same way, we form a vector indicating the number of movies watched in each week $a = (a_1, a_2, \ldots, a_p)$.
4. Set the context window size to $2t + 1$, where t is a hyperparameter. As for movie g_{ij} which ranks j in week i, movies ranking within its context window are its co-popular context, which are movies in $(g_{i(j-t)}, \ldots, g_{i(j-1)}, g_{i(j+1)}, \ldots, g_{i(j+t)})$. It is obvious that movies ranking in the first t and last t have less than $2t$ context movies.

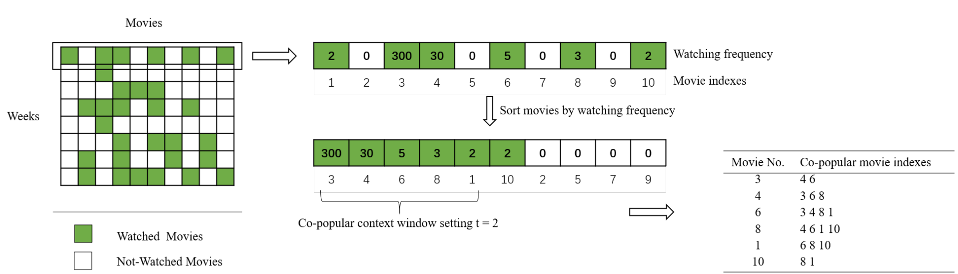

Fig. 1: Illustration of the co-popular relationship among movies, where we assume $t = 2$ and the fixed context window size is $2t + 1 = 5$.

According to the above steps to construct the co-popular context, we can calculate the PMI value between two movies, which is the logarithm of the ratio between their joint probability (the frequency they are co-popular contexts) and their marginal probabilities (the frequency they are watched independently). Then we can form PMI matrix M^{PMI}, where $m_{ij}^{PMI} = \log \frac{\#(i,j)D}{\#(i)\#(j)}$, $\#(i,j)$ is the frequency in which they are co-popular, $\#(i) = \Sigma_j \#(i,j)$ and $\#(j) = \Sigma_i \#(i,j)$, D is the amount of co-popular movie pairs. With PMI, we can obtain SPPMI with Equation (6).

$$M^{SPPMI}(i,j) = \max\left(M^{PMI}(i,j) - \log(h), 0\right) \tag{6}$$

In Equation (6), h is is a hyperparameter. The larger h is, the sparser SPPMI matrix is.

Finally, we add the SPPMI matrix obtained into Equation (5). Then we optimize the objective function with Adam algorithm and finally learn the parameters.

4 Experiments

In this section, we conduct extensive experiments on real-world on-demand data to answer the following five research questions:

RQ1: How does our approach perform compared with other state-of-art cold-start recommenders?

RQ2: Does the SPPMI matrix enhance the perfomance when handling cold-start problem?

RQ3. Do location category distribution vectors learned by LDA outperform log-smoothed POI type vectors?

RQ4. We use frequency-based weights in our loss function. Does it affect the performance?

RQ5. Our approach is also suitable for warm-start applications. So, how does it perform on warm-start applications?

Among the above questions, RQ1 focuses on performance of our model for cold-start problem. RQ2-RQ4 evaluate the contributions of several key components in our model. RQ5 explores performance of our approach for warm-start problem.

4.1 Experimental Settings

Datasets The data used in experiments are from iQIYI, which is one of the largest online video platforms in China. We collect real-world on-demand records between July 1, 2016 and Sept 1, 2018. There are total 798,886 valid records, covering 6,160 different movies, 207 cinemas and 29 provinces. We split the data by weeks, group the data of 24 weeks to be a dataset and construct three datasets with different periods as shown in Table 1.

For each dataset, data in the first 22 weeks are used to construct training sets while data of the last 2 weeks are used to construct testing sets. The following is the strategy to construct cold-start datasets. Taking the balance of cinema geographical distribution into consideration, we select cold-start cinemas randomly

Tab. 1: Timespans of datasets

Number	Start time	End time
Dataset 1	4/3/2017	9/17/2017
Dataset 2	9/25/2017	3/18/2018
Dataset 3	3/19/2018	9/2/2018

at each province, where the proportion of cold-start cinemas in each province is not less than 10%. Then we eliminate the on-demand records for these chosen cinemas, thus construct cold-start datasets. For each dataset, we split cinemas into three folds randomly and obtain 9 cold-start training sets and 9 testing sets, respectively. Finally, we collect POI information of cinemas within 1km.

Experiments for cold-start recommendation are designed to predict the interaction matrix of testing cinemas in the future 2 weeks based on the first 22 weeks of warm-start cinemas and the side information of all cinemas.

To gain insights into the watching records, we perform some statistical analysis. We plot the number of movies with respect to the movie watched frequency in Figure 2, from which we observe that the movie frequency distribution show a long tail distribution. Further, we find that interactions related to the 20% most popular movies occupy 86.9% of total interactions. This is consistent with most recommendation benchmarks, such as Nexflix [6] and Yelp [5] datasets, which highlights the sparsity challenge faced by recommendation systems. Besides, the watched frequency of a movie reflects its popularity among audiences. Huge differences in the watched frequencies indicate that different movies have huge difference of popularity for the same cinema.

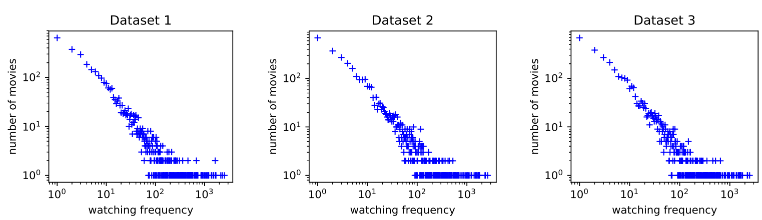

Fig. 2: Long tail distribution of movie watched frequency in each dataset

Metrics In this paper, we select three common ranking metrics to evaluate our approach. They are precision@k, recall@k and NDCG@k(Normalized Discounted Cumulative Gain), in which k ranges in $[1, 5, 10, 20, 50]$.

Hyperparameter Settings Default hyperparameter settings are listed influence Table. 2.

Tab. 2: Hypeparameter settings

Hypeparameters	settings
Initial Learning Rate	0.05
Dimension of Latent Space	50
α	0.5
λ_v,λ_y,λ_f	0.0001
h	1

4.2 Performance Evaluation for Cold-Start Recommendation

In this section, to justify the effectiveness of our proposed approach, we compare with the following state-of-art algorithms:

1. Most popular: This is a non-personalized recommendation approach, which selects the top-N movies with the highest watched frequency in the past t timeslots as the recommendation in the next timeslot. This approach generates the same recommendation for all cinemas.
2. Cos-Cos [15]: This is a similarity based method. It obtains K-nearest neighbor cinemas of cold-start cinemas according to the cosine similarity of cinema side information and generates the recommendation with the history data of neighbor cinemas.
3. CMF [14]: This is a matrix factorization based method. It collectively factorizes interaction matrix and the side information matrix by sharing user latent representations. Then the latent representations of cold-start users can be obtained with their side information. In this paper, we improve the original CMF model with frequency-based weights.
4. MetricRec [11]: This is a similarity based method based on Mahalanobis distance. It constructs the parameterized Mahalanobis distance with the cinema side information, whose parameters can be learned from training data. As for cold-start cinemas, we get K-nearest-neighbor users based on learned Mahalanobis distances and generate the recommendation.
5. LOCO [12]: This is a low-rank linear regression method. It regards the interaction matrix as a linear regression of cinema side information, in which the linear transformation matrix is low-rank. Similar to CMF, we improve the originsl LOCO model with frequency-based weights.

Side information inputted into competitors is a log-smoothed POI type matrix $X^{\log}$. Comparison results of precision, recall and NDCG are listed in Table 3. We observe: 1) Our approach significantly outperforms other competitors in all metrics with different hyperparameter k. 2) Although, our approach and CMF are both matrix factorization based model, our approach significantly outperforms CMF. There are mainly two reasons. On the one hand, we jointly factorized the interaction matrix and the co-popular SPPMI matrix by sharing item embeddings, which enhances item embeddings by exploiting co-popular patterns among movies. On the other hand, location category vectors learned by LDA are used to learn user latent representations in our approach, while log-smoothed POI type vectors are used in CMF. The latter is weaker to reflect cinema location categories. 3) As an unexpected finding, Most Popular approach is a very competitive approach in different metrics. It mainly results from the long tail distribution of movie watched frequency, which means only a small number of popular movies meet needs of most audiences. 4) Metric-Rec and Cos-Cos are all similarity based cold-start models. However Metric-Rec learns similarity parameters from training data while Cos-Cos adopts cosine similarity directly, which may be the main reason why Cos-Cos performs worse. This section answers RQ1.

Tab. 3: Performance comparison for cold-start recommendation

	Most Popular			Cos-Cos			CMF		
	Precision	Recall	NDCG	Precision	Recall	NDCG	Precision	Recall	NDCG
@1	0.401	0.004	0.401	0.123	0.001	0.172	0.492	0.005	0.492
@5	0.518	0.024	0.531	0.288	0.014	0.269	0.472	0.023	0.480
@10	0.541	0.050	0.529	0.267	0.024	0.275	0.443	0.043	0.457
@20	0.523	0.096	0.515	0.239	0.044	0.239	0.403	0.077	0.423
@50	0.449	0.206	0.440	0.254	0.119	0.249	0.342	0.161	0.369
Avg	0.486	0.076	0.483	0.234	0.040	0.241	0.430	0.062	0.444
	MetricRec			**LOCO**			**Ours**		
	Precision	Recall	NDCG	Precision	Recall	NDCG	Precision	Recall	NDCG
@1	0.456	0.004	0.456	0.682	0.007	0.682	**0.806**	**0.008**	**0.806**
@5	0.491	0.023	0.487	0.651	0.032	0.656	**0.765**	**0.038**	**0.777**
@10	0.497	0.046	0.492	0.610	0.059	0.624	**0.755**	**0.075**	**0.766**
@20	0.464	0.087	0.47	0.575	0.110	0.594	**0.693**	**0.134**	**0.719**
@50	0.406	0.189	0.425	0.479	0.222	0.514	**0.548**	**0.260**	**0.600**
Avg	0.463	0.070	0.466	0.560	0.086	0.614	**0.713**	**0.103**	**0.734**

4.3 Rationality Evaluation

In this section, we conduct an ablation study to illustrate the design rationalization of our approach, in which we verify contributions to performance of several main components of our approach, including the SPPMI matrix reflecting co-popular patterns among movies, location category probability distribution vectors learned by LDA and the frequency-based weights. This section answers RQ2-RQ4.

Effect of the SPPMI Matrix We collectively factorize SPPMI matrix formed by co-popular relationship among movies and the interaction matrix to enhance item embeddings. In order to evaluate the effect of SPPMI matrix, we remove SPPMI matrix but keep other settings the same. The results are shown in Table 4. We observe that the recommendation performance improves a lot with SPPMI matrix, especially in recall, in which recall@10 improves by 31.58%, recall@20 improves by 24.07%. This section answers RQ2.

Tab. 4: Effect of the SPPMI matrix

	Without SPPMI matrix			With SPPMI matrix			Improvement		
	Precision	Recall	NDCG	Precision	Recall	NDCG	Precision	Recall	NDCG
@1	0.741	0.007	0.741	0.806	0.008	0.806	8.77%	14.29%	8.77%
@5	0.666	0.032	0.680	0.765	0.038	0.777	14.86%	18.75%	14.26%
@10	0.607	0.057	0.633	0.755	0.075	0.766	24.38%	31.58%	21.01%
@20	0.576	0.108	0.602	0.693	0.134	0.719	20.31%	24.07%	19.44%
@50	0.472	0.218	0.513	0.548	0.260	0.600	16.10%	19.27%	16.96%
Avg	0.612	0.084	0.634	0.713	0.103	0.734	16.89%	21.59%	16.09%

Effect of the LDA-based Location Category Vectors In our approach, location category probability distribution vectors are learned by LDA. In order to evaluate the contribution of this module, we conduct a control experiment by directly replacing the X^{lda} with X^{log}, where we keep other settings the same. Performance comparison results are listed in Table 5. We observed that X^{lda} performs better than X^{log} in all metrics, which answers RQ3.

Tab. 5: Effect of the LDA-based location category vectors

	Log-smoothed vectors			LDA-based vectors			Improvement		
	Precision	Recall	NDCG	Precision	Recall	NDCG	Precision	Recall	NDCG
@1	0.680	0.007	0.680	0.806	0.008	0.806	18.53%	14.29%	18.53%
@5	0.679	0.033	0.680	0.765	0.038	0.777	12.67%	15.15%	14.26%
@10	0.673	0.066	0.676	0.755	0.075	0.766	12.18%	13.64%	13.31%
@20	0.616	0.120	0.635	0.693	0.134	0.719	12.50%	11.67%	13.23%
@50	0.498	0.236	0.539	0.548	0.260	0.600	10.04%	10.17%	11.32%
Avg	0.629	0.092	0.642	0.713	0.103	0.734	13.18%	12.98%	14.13%

Effect of the frequency-based weights In order to evaluate the effect of frequency-based weights, we conduct a control experiment by replacing the frequency-based weights with traditional fixed weights, in which we set $w_2 = 1$ for all movie-cinema pairs with $r_{ij} = 1$ and $w_2 = 0.5$ for all movie-cinema pairs with $r_{ij} = 0$. The hyperparameters are selected via the cross validation to achieve the best performance. Besides, other settings are kept the same with our approach. The comparison results are listed in Table 6. We observe that our performance has improved a lot, in which NDCG and precision are improved by 23.43% on average and recall has improved by 15.33% on average. This section answers RQ4.

Tab. 6: Effect of the frequency-based weights

	Fixed Weight			Frequency Weight			Improvement		
	Precision	Recall	NDCG	Precision	Recall	NDCG	Precision	Recall	NDCG
@1	0.617	0.007	0.617	0.806	0.008	0.806	30.63%	14.29%	30.63%
@5	0.649	0.034	0.643	0.765	0.038	0.777	17.87%	11.76%	20.84%
@10	0.628	0.065	0.632	0.755	0.075	0.766	20.22%	15.38%	21.20%
@20	0.567	0.114	0.588	0.693	0.134	0.719	22.22%	17.54%	22.28%
@50	0.448	0.221	0.491	0.548	0.260	0.600	22.32%	17.65%	22.20%
Avg	0.582	0.088	0.594	0.713	0.103	0.734	22.75%	15.33%	23.43%

4.4 Performance Evaluation for Warm-Start Recommendation

Though our approach solves the cinema cold-start problem, it is suitable for warm-start recommendation as well. In order to evaluate the performance for warm-start application, we compare our approach with WMF (Weighted Matrix Factorization) [7], which is a typical approach for warm-start recommendation.

To construct warm-start datasets, we use data of the first 22 weeks from each dataset and side information of corresponding cinemas as the warm-start training sets and form the interaction matrix R. At the same time, we obtain weight matrix W based on movie-cinema interaction frequency. Then, we predict the interaction matrix in the future two-weeks $\hat{R}$ with cinema latent representations U and movie latent representations I learned from training sets and select the top N movies as recommendation. We adopt frequency-based weights for WMF. Average results of three datasets are shown in Table 7.

Tab. 7: Performance comparison with WMF for warm-start recommendation

	WMF			Our Approach			Improvement		
	Precision	Recall	NDCG	Precision	Recall	NDCG	Precision	Recall	NDCG
@1	0.608	0.007	0.608	0.806	0.008	0.806	32.57%	14.29%	32.57%
@5	0.624	0.035	0.628	0.765	0.038	0.777	22.60%	8.57%	23.73%
@10	0.559	0.061	0.581	0.755	0.075	0.766	35.06%	22.95%	31.84%
@20	0.519	0.113	0.546	0.693	0.134	0.719	33.53%	18.58%	31.68%
@50	0.423	0.228	0.467	0.548	0.260	0.600	29.55%	14.04%	28.48%
Avg	0.547	0.089	0.566	0.713	0.103	0.734	30.66%	15.69%	29.66%

We observe that our approach outperforms WMF in all metrics. The results can be explained from two aspects. On the one hand, we model the cinema location category, which reflects the preference of the potential audiences. On the other hand, the SPPMI matrix models co-popular relationship among movies, which enhances movie embeddings and improves the recommendation performance.

5 Conclusion

In order to solve the on-demand cinema cold-start problem, we propose a matrix factorization based approach which fuses location categories of cinemas and co-popular relationship among movies. We collect POI type information within 1km of the cinema as the cinema side information. Then we learn the location category probability distribution vector by LDA, which is used to approximate cinema latent representations. Besides, we construct the co-popular context of movies and form the SPPMI matrix, which is factorized collectively with the interaction matrix by sharing the same item latent representations. We conduct extensive experiments on real on-demand records. Firstly, we compare with other state-of-art cold-start approaches, which prove our approach has a significant improvement of performance. Secondly, we conduct the ablation study to verify the contributions to performance of several major components. Thirdly, we find that our approach outperforms weighted matrix factorization for warm-start recommendation.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993-1022 (2003)
2. Cao, D., Nie, L., He, X., Wei, X., Zhu, S., Chua, T.S.: Embedding factorization models for jointly recommending items and user generated lists. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 585-594. ACM (2017)
3. Gantner, Z., Drumond, L., Freudenthaler, C., Rendle, S., Schmidt-Thieme, L.: Learning attribute-to-feature mappings for cold-start recommendations. In: 2010 IEEE International Conference on Data Mining. pp. 176-185 (2010)
4. Grbovic, M., Radosavljevic, V., Djuric, N., Bhamidipati, N., Savla, J., Bhagwan, V., Sharp, D.: E-commerce in your inbox: Product recommendations at scale. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1809-1818. ACM (2015)
5. He, X., Zhang, H., Kan, M.Y., Chua, T.S.: Fast matrix factorization for online recommendation with implicit feedback. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 549-558. ACM (2016)
6. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 426-434. ACM (2008)

7. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Computer 42(8), 30-37 (2009)
8. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Advances in neural information processing systems. pp. 2177-2185 (2014)
9. Li, J., Lu, K., Huang, Z., Shen, H.T.: Two birds one stone: On both cold-start and long-tail recommendation. In: Acm on Multimedia Conference (2017)
10. Liang, D., Altosaar, J., Charlin, L., Blei, D.M.: Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In: Proceedings of the 10th ACM conference on recommender systems. pp. 59-66. ACM (2016)
11. Peng, F., Lu, X., Lu, J., Yi-Da Xu, R., Luo, C., Ma, C., Yang, J.: Metricrec: Metric learning for cold-start recommendations. In: International Conference on Advanced Data Mining and Applications. pp. 445-458. Springer (2016)
12. Sedhain, S., Menon, A.K., Sanner, S., Xie, L., Braziunas, D.: Low-rank linear coldstart recommendation from social data. In: AAAI. pp. 1502-1508 (2017)
13. Shi, L., Zhao, W.X., Shen, Y.: Local representative-based matrix factorization for cold-start recommendation. ACM Transactions on Information Systems 36(2), 22 (2017)
14. Singh, A.P., Gordon, G.J.: Relational learning via collective matrix factorization.In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 650-658 (2008)
15. Suvash, S., Scott, S., Darius, B., Lexing, X., Jordan, C.: Social collaborative filtering for cold-start recommendations. In: Proceedings of the ACM Conference on Recommender Systems, RecSys. vol. 14
16. Tran, T., Lee, K., Liao, Y., Lee, D.: Regularizing matrix factorization with user and item embeddings for recommendation. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. pp. 687-696. ACM (2018)
17. Wang, C., Blei, D.M.: Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 448-456. ACM (2011)
18. Xie, M., Yin, H., Wang, H., Xu, F., Chen, W., Wang, S.: Learning graph-based poi embedding for location-based recommendation. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. pp. 15-24. ACM (2016)
19. Xue, T., Jin, B., Li, B., Wang, W., Zhang, Q., Tian, S.: Spatial-temporal recommendation for on-demand cinemas. In: International Conference on Database Systems for Advanced Applications. Springer (2019)
20. Yu, J., Gao, M., Li, J., Yin, H., Liu, H.: Adaptive implicit friends identification over heterogeneous network for social recommendation. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. pp. 357-366. ACM (2018)