A Spatio-temporal Recommender System for On-demand Cinemas

Taofeng Xue[1,2], Beihong Jin[1,2*], Beibei Li[1,2], Weiqing Wang[3], Qi Zhang[4], and Sihua Tian[4]

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

ABSTRACT

On-demand cinemas are a new type of offline entertainment venues which have shown the rapid expansion in the recent years. Recommending movies of interest to the potential audiences in on-demand cinemas is keen but challenging because the recommendation scenario is totally different from all the existing recommendation applications including online video recommendation, offline item recommendation and group recommendation. In this paper, we propose a novel spatio-temporal approach called Pegasus. Because of the specific characteristics of on-demand cinema recommendation, Pegasus exploits the POI (Point of Interest) information around cinemas and the content descriptions of movies, apart from the historical movie consumption records of cinemas. Pegasus explores the temporal dynamics and spatial influences rooted in audience behaviors, and captures the similarities between cinemas, the changes of audience crowds, time-varying features and regional disparities of movie popularity. It offers an effective and explainable way to recommend movies to on-demand cinemas. The corresponding Pegasus system has been deployed in some pilot on-demand cinemas. Based on the real-world data from on-demand cinemas, extensive experiments as well as pilot tests are conducted. Both experimental results and post-deployment feedback show that Pegasus is effective.

CCS CONCEPTS

• **Information systems → Recommender systems**; **Collaborative filtering**.

KEYWORDS

Recommender system; On-demand cinema; Spatio-temporal effect

ACM Reference Format:

Taofeng Xue, Beihong Jin, Beibei Li, Weiqing Wang, Qi Zhang, and Sihua Tian. 2019. A Spatio-temporal Recommender System for On-demand Cinemas. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357888

* Corresponding author.

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357888

1 INTRODUCTION

With the increase of the diverse cultural and entertainment needs of the masses, on-demand cinemas, which combine advantages of online video websites and traditional offline cinemas, emerge and show the tendency of rapid expansion. Taking iQIYI (the video entertainment company that ranks first in China) as an example, more than 200 on-demand cinemas in the league of iQIYI are running in China, which attracts more investments in this type of cinemas.

The on-demand cinemas, as a new type of offline entertainment venues, are usually newly-built or adapted from Karaoke (also known as KTV) boxes. They aim at serving small groups of audiences (e.g., 1∼20 people) with fresh watching experiences. Specifically, these new offline on-demand cinemas offer different-sized private boxes equipped with rich hardware configurations, and high-quality digital movies whose copyrights are owned by online video websites. Compared with online video websites, offline on-demand cinemas can provide a full range of services to their target audiences, including friend reunion, family gathering, and couple dating. Compared with traditional offline cinemas, on-demand cinemas are able to offer flexible choices of movie resources and watching time.

As far as the normal operations of on-demand cinemas are concerned, a list of all the movies may be provided to potential audiences so that they browse and choose the movies to be watched. However, facing the overloading information, potential audiences have to take a lot of time to make a small decision. That would bring the negative user experience, even resulting in the loss of audiences. This motivates us to build an effective recommender system for on-demand cinemas, which recommends the movies that are of interest to potential audiences from numerous copyrighted candidates. With the recommendation lists in hand, staffs/hosts of on-demand cinemas are able to recommend them to audiences.

It is very challenging to build an effective recommender system for on-demand cinemas as the recommendation scenario is totally different from all the existing recommendation applications, including online video recommendation, offline item recommendation and group recommendation.

On-demand cinema recommendation quite differs from online video recommendation. First, the latter aims to recommend movies to individuals while the former to cinemas. Since the audiences of on-demand cinemas are unknown in advance and arrive in an anonymous and random manner, it is impossible to achieve accurate personalized recommendation directly to audiences. Second, movie-watching behaviors of on-demand cinemas show the characteristics of temporal dynamics and geographical aggregation while online users scatter at more various locations and have more choices of watching time. Specifically, the audiences of the on-demand cinemas usually go to cinemas along with other people, generally for the purpose of dating or gathering, and prefer to choose special times, such as evenings or weekends. As a result, the audience behaviors of watching movies will show more obvious temporal dynamics. In addition, the cinemas show the characteristics of geographical clustering due to the fact that audiences of on-demand cinemas usually consist of people who work or live nearby. In the meantime, cinemas have the characteristics of preference locality, that is, the preferences of audiences in different cinemas may vary a lot. If the location of a cinema is in a typical functional region (such as a central business district, a shopping center, or a residential area), then the cinemas also have the characteristics of category locality, i.e., the watching behaviors are probably related to the category of a region.

On-demand cinema recommendation also differs from both existing offline recommendation and group recommendation. The existing offline recommender systems are usually designed for high-frequent and personalized consumption scenarios [20]. For example, for a convenience store, a customer with a membership card may visit it frequently. In this scenario, a recommender system is able to track each user and also has enough data to build a profile for each user. Based on the learned profiles, recommendation is made to each user. In contrast, the on-demand cinemas belong to a low-frequent and anonymous consumption scenario by nature. Besides, the high mobility and anonymity of audiences in on-demand cinemas also make the recommendation differ from group recommendation [2]. Typically, a group recommender system focuses on modeling a fixed set of users with the goal to satisfy the preferences of all the members by aggregating preferences of group members. However, in a on-demand cinema recommender system, the audiences in one cinema are anonymous (i.e., it is unable to track each individual user) and not fixed (i.e., recommendation to a cinema is not equal to the recommendation to a specific group of users).

In this paper, we propose a novel movie recommendation method for on-demand cinemas, called Pegasus, which is able to simultaneously model the temporal dynamics and spatial influences of audience behaviors. Based on the method, we build a practical recommender system, i.e., the Pegasus system, for on-demand cinemas.

The main contributions of our work are summarized as follows.

- We propose a novel spatio-temporal approach called Pegasus. Pegasus exploits historical movie-watching records of on-demand cinemas, POI data around the cinemas and content descriptions of movies. By modeling temporal dynamics and spatial influences in the audience behaviors and further combining multiple collaborative filtering (e.g., global and neighboring collaborative filtering) to reveal the global and local effects of audience behaviors, Pegasus offers an effective and explainable way to recommend movies to on-demand cinemas.
- We collect the real-world records from on-demand cinemas to form datasets and conduct extensive experiments on the datasets. The experimental results show that Pegasus significantly outperforms the other state-of-the-art methods.
- So far, the Pegasus system has been deployed in multiple on-demand cinemas for a pilot. We have conducted post-deployment evaluation on dozens of cinemas. The feedback on recommendation lists shows that the Pegasus system is helpful in improving the operation performance of cinemas.
- To the best of our knowledge, the Pegasus system is the first recommender system for on-demand cinemas running in the real world. It provides a practical way of building recommender systems for service business at physical sites. For example, supermarkets have the demand of stocking fresh or dairy goods locally and wrong stocking choices will result in either a waste of resource or being unable to meet the needs of consumers. With Pegasus, we can accurately predict consumer needs to reduce operational risks and increase revenue.

The rest of the paper is organized as follows. Section 2 introduces the related work. Section 3 gives the formulation of the recommendation problem to be solved. Section 4 describes our Pegasus method in detail. Section 5 gives the experiment and post-deployment evaluation. Finally, the paper is concluded in Section 6.

2 RELATED WORK

In recent years, in order to improve the performance of recommender systems, a variety of contextual information has been introduced into recommendation methods [16]. For example, [13] incorporates social relationships and employs the regularization terms to constrain the representation of latent vectors of similar users. [21] adopts LDA (Latent Dirichlet Allocation) to infer the topics of items from their content descriptions and incorporates the topics of items into the collaborative filtering method.

Further, some research has incorporated temporal or spatial information into the recommendation methods.

In terms of temporal dynamic modeling, there are mainly three kinds of methods, that is, point analysis methods, local analysis methods and global analysis methods.

Point analysis methods usually calculate the temporal influences of data, and turn the original static methods to be time-aware methods. For example, [5] employs a time decay function to discount and re-weight user feedback data. Further, [26] studies a kernel-based time decay function to model the sudden change in interest and the short-term interest of a tag recommedation system. The drawback of point analysis methods is that they result in information loss and underestimation of historical data.

Local analysis methods suggest that user preferences and item popularity can be learned by mining ratings of same time intervals. The corresponding data are divided by time intervals and time parameters for different time intervals are set to model temporal dynamics. The representative method is TimeSVD++ [10]. TimeSVD++ assigns a bias to each item at each time interval and utilizes different linear functions to capture the gradual changing of the user biases and user latent factors. Local analysis methods mainly explore the local temporal dynamics to model user preference changes. However, these methods ignore the relationship of user preferences across time intervals, which means that the overall evolution of user preferences is ignored. Therefore, they are only applicable to scenarios where there is no obvious global evolution of user preferences.

The methods of explicitly revealing the temporal dynamics by state space models or temporal regression models are classified as global analysis methods. In the state space models (Bayesian probability models), [19] employs Kalman filtering to model the changes of user preferences. Based on Poisson factorization, [7] utilizes a Poisson process to model the phenomenon of recurrent activities. In temporal regression models, [12] models the changes of the user latent vectors by means of a vector autoregressive model. The global analysis methods are good at discovering the global evolution patterns of user preferences. However, the common premise of these methods is that the evolution of user preferences has potential and global patterns to follow. That is to say, they are not suitable for scenarios where there is no obvious global evolution.

The goal of integration of spatial information is to model the spatial influences. In detail, [25] employs a power-law distribution to model the check-in probability to the distance between two POIs visited by the same person. [28] proposes the personalized kernel density estimation to predict the probability of a user visiting a certain location. [3] models the probability of a user’s check-in on a location as a Multi-center Gaussian Model (MGM). The common characteristics of these methods are that they use only the geographical location information of users or items, while ignoring the rest of the spatial information related to users or items.

The fusion of multiple contextual information into the recommendation methods is a recent research hotspot for recommender systems. [23] proposes a spatial-temporal sparse additive generative model for spatial item recommendation. [15] proposes a spatio-temporal context-aware and translation-based recommender framework to model the third-order relationship among users, POIs, and spatio-temporal contexts for large-scale POI recommendation. [30] incorporates temporal and spatial information, and proposes a spatio-temporal latent ranking method to explicitly model the interactions among users, POIs and times for successive POI recommendations. [27] proposes a method of integrating spatial information and description information of items in a unified way, which enables clustering of both content-similar and co-visited spatial items into the same topics with a high probability.

We note that deep learning techniques are being applied to recommender systems [29]. Some work utilizes neural networks to extract item presentations and apply them to the existing recommender systems [8, 9]. Some work combines collaborative filtering with neural networks to derive new recommendation methods [6, 17, 22]. Considering that deep learning methods cannot show the advantage while using limited amount of data as input (see Section 5.4), we do not employ deep learning techniques temporarily.

Different from the existing work, our Pegasus method incorporates a variety of contextual information, including time information of movie-watching records of cinemas, POI information around cinemas, and content description information of movies. In particular, by means of the temporal information of interactive data, we capture the temporal dynamics of the audience behaviors. Further, based on the POIs around cinemas, we learn the similarity metrics between cinemas with the aid of metric learning techniques. Last but not least, based on the content descriptions of movies, we infer the topic distributions of movies and apply them to measure the recent interest of cinemas.

3 FORMULATION

In this section, we give a formal description of the data we employ, formulate the problem of movie recommendation in on-demand cinemas, and outline our approach.

Definition 1. (**Cinema Activity**) A cinema activity is made of a 4-tuple (i, j, t, l), which indicates that at time t, audiences watch movie j at on-demand cinema i located at geographic location l.

Definition 2. (**Cinema Profile**) For a cinema i, its cinema profile P_i is a set of cinema activities associated with i, and its cinema profile at time interval τ, denoted as P_i^τ, consists of cinema activities occur at time interval τ. Further, historical records of all the on-demand cinemas, denoted as P, is a collection of P_i^τ. That is, $P = \{\cup P_i^\tau | i = 1,...,M; \tau = 1,...,m\}$, where M denotes the number of cinemas and m denotes the number of time intervals.

Definition 3. (**POIs around a Cinema**) Given a cinema i and its location l, the POIs around cinema i refer to the POIs within the region which centers on the location l with a radius of 1 km. For all the cinemas, the POIs around cinemas are organized as the matrix $X \in \mathbb{R}^{M\times\mathcal{D}}$, where $\mathcal{D}$ is the number of POI types and element x_{id} denotes the number of POIs with the type of d around cinema i. Since different cinemas have greatly varying POI types, a logarithmic transformation on x_{id} has been used to smooth the data.

Definition 4. (**Content Descriptions of a Movie**) Given a movie j, its content description refers to the text content of the movie (including title, introduction, released area, type, style and stars). For all the movies, their movie contents are organized as the matrix $Y \in \mathbb{R}^{N\times V}$, where N is the total number of movies, V is the total number of words in the dictionary, and element y_{jv} represents the times that the word v appears in the content description of movie j.

Definition 5. (**Rating of a Cinema on a Movie**) The rating of a cinema i on a movie j reflects the level of interest that the audiences of i have on j. Note that, we cannot get this rating directly from the original data. In this paper, we propose a percentile-based preprocessing method to calculate the ratings of cinemas on movies. The basic idea is to use the numbers of a cinema’s activities assigned to each movie to derive ratings of this cinema on each movie.

Considering that a movie may enjoy different popularity at different intervals, we conduct rating conversion by time intervals. Further, since there are obvious popularity differences between movies, we employ a smoothing strategy during conversion. The detailed procedure is as follows.

For a cinema i and a time interval τ, we first obtain the number of cinema activities associated with each movie j from cinema profile P_i^τ, denoted as c_{ij}^τ. Then, we can form a set of the numbers of different movies in cinema i at time interval τ, denoted as $C_i^\tau = \{c_{i1}^\tau, c_{i2}^\tau, \ldots, c_{in}^\tau\}$, where n is the total number of movies watched in cinema i at time interval τ.

Further, we transform the number set C_i^τ to a rating set. Given the specified p_1, p_2, p_3 and p_4 ($p_1 < p_2 < p_3 < p_4$), we calculate the p_1-th, p_2-th, p_3-th and p_4-th percentiles of C_i^τ, that is, $c_{ip_1}^\tau$, $c_{ip_2}^\tau$, $c_{ip_3}^\tau$ and $c_{ip_4}^\tau$, respectively. In general, we have $c_{ip_1}^\tau \leq c_{ip_2}^\tau \leq c_{ip_3}^\tau \leq c_{ip_4}^\tau$, however, to get a reasonable rating value, we need these 4 percentiles to be different. Thus, we traverse the 4 percentiles in order, and if the adjacent percentiles are equal, we subtract the previous percentile by a small number 0.1, i.e., $c^{\tau}_{ip_o} \leftarrow c^{\tau}_{ip_o} - 0.1$, if $c^{\tau}_{ip_{o+1}} = c^{\tau}_{ip_o}, \forall o = 1, 2, 3$. After the process, we can ensure that $c^{\tau}_{ip_1} < c^{\tau}_{ip_2} < c^{\tau}_{ip_3} < c^{\tau}_{ip_4}$. Note that p_1, p_2, p_3, and p_4 can be set according to the actual data distribution of C^{τ}_i. By default, $p_1 = 30\%$, $p_2 = 53\%$, $p_3 = 76\%$, and $p_4 = 99\%$ (i.e., nearly 30% movies only appear once, 1% movies appear significantly more than others, other percentiles are equally set between 30% ∼ 99%).

Finally, we introduce the binning function f shown in Eq. (1), which consults the neighborhood of values and perform local smoothing. f is applied to each value c^{τ}_{ij} of C^{τ}_i to obtain the rating of cinema i on movie j at time interval τ.

$$r^{\tau}_{ij} = f(c^{\tau}_{ij}) = \begin{cases} 1, & \text{if } c^{\tau}_{ij} \le c^{\tau}_{ip_1} \\ 2, & \text{if } c^{\tau}_{ij} > c^{\tau}_{ip_1} \;\&\&\; c^{\tau}_{ij} \le c^{\tau}_{ip_2} \\ 3, & \text{if } c^{\tau}_{ij} > c^{\tau}_{ip_2} \;\&\&\; c^{\tau}_{ij} \le c^{\tau}_{ip_3} \\ 4, & \text{if } c^{\tau}_{ij} > c^{\tau}_{ip_3} \;\&\&\; c^{\tau}_{ij} \le c^{\tau}_{ip_4} \\ 5, & \text{if } c^{\tau}_{ij} > c^{\tau}_{ip_4} \end{cases} \tag{1}$$

After the above conversion, the original P is transformed into $\mathcal{R}$, a 5-tuple set of cinema activities. That is, $\mathcal{R} = \{(i, j, \tau, l, r)\}$, where i denotes a cinema, j denotes a movie, τ denotes a time interval, l denotes the geographic location of cinema i, and $r_{ij}(\tau, l)$ denotes the rating of cinema i located at l on movie j at time interval τ, that is, $r_{ij}(\tau, l) = r^{\tau}_{ij}$.

Problem. (**Movie Recommendation in On-demand Cinemas**) Given M on-demand cinemas and their historical records $P = \{\cup P^{\tau}_i | i = 1,...,M; \tau = 1,...,m\}$ at continuous m time intervals, the problem to be solved is to recommend a list of movies from N movies that may be of interest to potential audiences of cinema i at time interval $m + 1$.

Approach. (**Movie Recommendation in On-demand Cinemas**) Given a target cinema i with its location l, our approach is to mine the rating set $\mathcal{R}$ originated from historical records P, POI matrix X of cinemas, and content description matrix Y of movies to predict rating $r_{ij}(\tau, l)$ of different movies at the subsequent time interval τ and then, based on the predicted ratings, to form a recommendation list of movies that may be of interest to potential audiences of cinema i at the subsequent time interval τ, where $\tau = m + 1$.

4 PEGASUS APPROACH

Pegasus is based on the idea of collaborative filtering [18]. It models temporal dynamics (including periodic effect, recency effect and audience crowd drifting effect) and spatial influences (including spatial neighboring of cinemas and spatial popularity of movies). Pegasus decomposes the rating of cinema i located at l on movie j at time interval τ, denoted as $\hat{r}_{ij}(\tau, l)$, as follows, which makes ratings explainable.

$$\hat{r}_{ij}(\tau, l) = \hat{r}^{\mathrm{B}}_{ij} + \hat{r}^{\mathrm{T}}_{ij}(\tau) + \hat{r}^{\mathrm{S}}_{ij}(l) \tag{2}$$

$$\hat{r}^{\mathrm{B}}_{ij} = b_i + b_j + \mathbf{q}^T_j \mathbf{p}_i \tag{3}$$

$$\hat{r}^{\mathrm{T}}_{ij}(\tau) = \hat{r}^{\mathrm{Period}}_{ij}(\tau) + \hat{r}^{\mathrm{Recency}}_{ij}(\tau) + \hat{r}^{\mathrm{Crowd}}_{ij}(\tau) \tag{4}$$

$$\hat{r}^{\mathrm{S}}_{ij}(l) = \hat{r}^{\mathrm{Neighbor}}_{ij}(l) + \hat{r}^{\mathrm{Popularity}}_{ij}(l) \tag{5}$$

Eq. (2) denotes how to derive the overall rating $\hat{r}_{ij}(\tau, l)$. In this equation, $\hat{r}^{\mathrm{B}}_{ij}$ denotes the rating mined from the cinema activity profiles based on collaborative filtering. The detailed inference of $\hat{r}^{\mathrm{B}}_{ij}$ is given in Eq. (3). In Eq. (3), $\mathbf{p}_i$ and $\mathbf{q}_j$ represent the k-dimension latent vectors of cinema i and movie j, respectively. b_i and b_j denote the preference biases of cinema i and movie j, respectively. $\hat{r}^{\mathrm{T}}_{ij}(\tau)$ and $\hat{r}^{\mathrm{S}}_{ij}(l)$ in Eq. (2) represent the overall effects of temporal dynamics and spatial influences on the ratings. Their detailed information are given in Eq. (4) and (5). $\hat{r}^{\mathrm{Period}}_{ij}(\tau)$, $\hat{r}^{\mathrm{Recency}}_{ij}(\tau)$ and $\hat{r}^{\mathrm{Crowd}}_{ij}(\tau)$ represent the influences of periodic effect, recency effect, and audience crowd drifting effect, respectively. $\hat{r}^{\mathrm{Neighbor}}_{ij}(l)$ and $\hat{r}^{\mathrm{Popularity}}_{ij}(l)$ denote the influences of spatial neighboring effect and spatial popularity effect, respectively. Note that, the Eq. (3) is designed following the standard collaborative filtering approach. Thus, we will focus on the detailed introduction to the modeling of temporal dynamics and spatial influences (i.e., Eq. (4) and Eq. (5)) in the following subsections.

4.1 Modeling Temporal Dynamics

This subsection describes the influences of periodic effect, recency effect and audience crowd drifting effect on ratings. Note that $r_{ij}(\tau, l)$ used in this subsection can be simplified to $r_{ij}(\tau)$ because l is irrelevant to modeling temporal dynamics.

4.1.1 Periodic Effect. In the on-demand cinema scenario, the audience watching behaviors will show strong periodicity. For example, audiences may have similar watching behaviors at the same time of different days (e.g., night), on the same day of different weeks (e.g., weekend) or in the same season (e.g., winter) of different years.

Considering that the cinema-level interactive data are usually denser than the personal data in traditional recommender systems, latent factor factorization is used to capture the periodic effect of the cinemas, that is,

$$\hat{r}^{\mathrm{Period}}_{ij}(\tau) = \mathbf{q}^T_j \mathbf{p}^{\mathrm{Period}}_i(\tau) \tag{6}$$

$$\mathbf{p}^{\mathrm{Period}}_i(\tau) = \sum\nolimits_{s=1}^{S} \alpha_{i,s} \times \mathbf{p}^{\mathrm{Period}}_{i,s} \times h_s(\tau) \tag{7}$$

where $\mathbf{p}^{\mathrm{Period}}_i(\tau)$ is the overall periodic latent vector of cinema i at time interval τ, $\mathbf{p}^{\mathrm{Period}}_{i,s}$ is the s-th periodic latent vector of cinema i, $\alpha_{i,s}$ is the adaptive coefficient of the influence of $\mathbf{p}^{\mathrm{Period}}_{i,s}$ ($s \in \{1, 2 \ldots, S\}$, where S denotes the number of period types we consider), and $h_s(\tau)$ is an indicator function for period s at time interval τ, i.e., if τ falls into a period such as the s-th period, then $h_s(\tau)$ returns 1, otherwise 0.

Specifically, we can customize different periods based on the unit of time interval required by the recommendation problem. In practice, if our goal is to recommend movies to cinema once a month, then we take a season (e.g., winter) as a period. If our goal is to recommend once a week, then we treat any long holiday (e.g., National Day) as a period, besides 4 seasonal periods. If our goal is to recommend once a day, then we treat each day of a week (e.g., Sunday), any short/long holiday, and each season as a period.

4.1.2 Recency Effect. The recency effect of on-demand cinemas means that the recent social hotspots will contribute to a positive correlation of the recent behaviors that audiences watch movies. For example, a recently released blockbuster movie will attract a lot of audiences to go to the on-demand cinemas, and related movies, e.g., movies belonging to the same series or movies of the same stars, will enjoy the great popularity, too.

Specifically, we think that cinema preference bias (denoted as $b_{i,\tau}$) and movie preference bias (denoted as $b_{j,\tau}$) are associated with time intervals. In addition, the more the topic distribution $\boldsymbol{\theta}_j$ of movie j is similar to the recent interest distribution of cinema i, the more the movie j meets the recent interest of audiences of cinema i, and the higher rating of cinema i on movie j is expected. Here, the recency refers to the time interval τ.

Considering that a row of matrix Y represents the content of a movie, we can regard the set of words (i.e., the non-zero elements in a row of Y matrix) as a document, and apply LDA (Latent Dirichlet Allocation) algorithm to learn the topic distribution of the movie. Let $\boldsymbol{\theta}_j$ denote the learned topic distribution of movie j. We use the ratings to weight the topic distributions of all the recent on-demand movies of cinema i, and regard the weighted topic distribution $\boldsymbol{\theta}_{i,\tau}$ as the recent interest distribution of cinema i as follows.

$$\boldsymbol{\theta}_{i,\tau} = \frac{\sum_{j' \in \mathcal{J}_{i,\tau}} r_{ij'}(\tau) \times \boldsymbol{\theta}_{j'}}{\sum_{j' \in \mathcal{J}_{i,\tau}} r_{ij'}(\tau)} \tag{8}$$

In Eq. (8), $\mathcal{J}_{i,\tau}$ is the set of on-demand movies of cinema i at time interval τ.

Therefore, the influence of recency effect on the ratings can be expressed as follows.

$$\begin{aligned} \hat{r}_{i,j}^{\text{Recency}}(\tau) &= \boldsymbol{\theta}_j^T \boldsymbol{\theta}_{i,\tau} + b_{i,\tau} + b_{j,\tau} \\ &= \boldsymbol{\theta}_j^T \left(\frac{\sum_{j' \in \mathcal{J}_{i,\tau}} r_{ij'}(\tau) \times \boldsymbol{\theta}_{j'}}{\sum_{j' \in \mathcal{J}_{i,\tau}} r_{ij'}(\tau)}\right) + b_{i,\tau} + b_{j,\tau} \end{aligned} \tag{9}$$

In Eq. (9), $\boldsymbol{\theta}_j^T \boldsymbol{\theta}_{i,\tau}$ measures the effect of similarity between the recent interest distribution of cinema i and the topic distribution of movie j.

4.1.3 Audience Crowd Drifting Effect. In addition to the drifting of overall audiences, the proportions of different audience crowd types (e.g., couples, families or friends) also drift over time. Thus, we employ a generative model to obtain the distribution of audience crowd types at different time intervals, and then obtain the corresponding rating drifting $\hat{r}_{ij}^{\text{Crowd}}(\tau)$.

In detail, we infer the distribution of audience crowd types of the cinemas at each time interval based on $\mathcal{R} = \{(i, j, \tau, l, r)\}$. The generative process of $\mathcal{R}$ of all cinemas is as follows.

For each cinema $i \in \{1, 2, \ldots, M\}$, and each time interval $\tau \in \{1, 2, \ldots, m\}$,

(1) Draw the distribution of audience crowd types of cinema i at current time interval τ, $\boldsymbol{\gamma}_{i,\tau} \sim Dir(\boldsymbol{\alpha})$,
(2) Draw the interest distribution of the audience crowd type k, $\boldsymbol{\phi}_k \sim Dir(\boldsymbol{\beta})$,
(3) For each movie in the set of on-demand movies of cinema i at time interval τ, i.e., $a_{i,j} \in \mathcal{J}_{i,\tau}$,
  (a) Draw an audience crowd type according to the distribution of audience crowd types, $z_{i,j} \sim Mult(\boldsymbol{\gamma}_{i,\tau})$,
  (b) Draw a movie according to the interest distribution of the audience crowd type, $a_{i,j} \sim Mult(\boldsymbol{\phi}_{z_{i,j}})$.
  (c) Repeat the above two steps $r_{ij}(\tau)$ times.

In our method, the rating $r_{ij}(\tau)$ is approximated by the frequency of the appearance of movie j in $\mathcal{J}_{i,\tau}$. Through the above generative process and the variational inference method in [1], we can infer the distribution of audience crowd types of cinema i at different time intervals $\tau = \{1, 2, \ldots, m\}$, denoted as $\boldsymbol{\gamma}_{i,\tau}$. Each dimension of $\boldsymbol{\gamma}_{i,\tau}$ represents the ratio of corresponding crowd type to all crowd types of cinema i at time interval τ. By default, the dimension of $\boldsymbol{\gamma}_{i,\tau}$ is 50. With $\boldsymbol{\gamma}_{i,\tau}$, the rating drifting caused by changes in the audience crowd types is as follows.

$$\hat{r}_{ij}^{\text{Crowd}}(\tau) = \mathbf{g}_i^T \boldsymbol{\gamma}_{i,\tau} \tag{10}$$

In Eq. (10), $\mathbf{g}_i$ represents the weights of different audience crowd types, which needs to be learned to capture the shared interest of different audience crowds of cinema i across all time intervals.

4.2 Modeling Spatial Influences

This subsection describes the spatial influences of spatial neighboring effect of cinemas and the spatial popularity effect of movies.

4.2.1 Spatial Neighboring Effect. Considering that cinemas with similar POI distribution around may attract similar audiences, there may be a tight correlation between the watching preferences of the two cinemas. Therefore, based on POI matrix $\mathbf{X}$, we can obtain the similarity metrics between cinemas through metric learning [24, 31], and capture the influence of spatial neighboring effect on ratings through KNN (K-Nearest Neighbors). The formula is as follows.

$$\hat{r}_{ij}^{\text{Neighbor}}(l) = \rho_i \times \frac{\sum_{u \in \mathcal{N}_i} w_{iu} \times (\bar{r}_{uj} - \bar{r}_u)}{\sum_{u \in \mathcal{N}_i} w_{iu}} \tag{11}$$

In Eq. (11), ρ_i denotes the degree of influence of spatial neighboring effect, $\bar{r}_{uj}$ denotes the average rating of cinema u on movie j over all time intervals, $\bar{r}_u$ denotes the average rating of cinema u on all movies, $\mathcal{N}_i$ denotes the set of k-nearest neighbors to cinema i, which can be obtained in terms of the Mahalanobis distance calculated by the POI vectors of the cinemas, and w_{iu} denotes the similarity metric between cinema i (located at l) and cinema u, which is computed by the Gaussian kernel function from Mahalanobis distance $dist_{iu}$ of the two cinemas.

$$w_{iu} = \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{(dist_{iu})^2}{\sigma^2}\right) \tag{12}$$

$$\begin{aligned} (dist_{iu})^2 &= ||(\mathbf{x}_i - \mathbf{x}_u)^T \mathbf{E}(\mathbf{x}_i - \mathbf{x}_u)|| \\ &= ||(\mathbf{x}_i - \mathbf{x}_u)^T \mathbf{A}^T \mathbf{A}(\mathbf{x}_i - \mathbf{x}_u)|| \\ &= ||\mathbf{A}(\mathbf{x}_i - \mathbf{x}_u)||^2 \end{aligned} \tag{13}$$

In above equations, σ denotes the parameter of the Gaussian kernel function, $\mathbf{x}_i$ and $\mathbf{x}_u$ denote the POI vectors in the POI matrix $\mathbf{X}$ corresponding to the cinema i and u respectively, and $\mathbf{E}$ is a symmetric positive semi-definite matrix, i.e., $\mathbf{E} = \mathbf{A}^T\mathbf{A}$, $\mathbf{A} \in \mathbb{R}^{D \times G}$, which can be regarded as a projection matrix that projects the original D-dimensional vector of POI data into the G-dimensional space. We will incorporate the neighboring effect and matrix factorization in a unified learning framework to learn $\mathbf{A}$ in Section 4.3.

4.2.2 Spatial Popularity Effect. Movies may have entirely different popularity in different regions. For example, Cantonese movies may be more popular in Pearl River Delta (e.g., Guangdong and Hong Kong), China. Obviously, the popularity of the movies will directly affect their on-demand probabilities. Thus, we infer the spatial popularity of the movies and estimate the corresponding influence on ratings.

We adopt a generative process similar to that in the crowd drifting effect (but not considering drifting) to interpret $\mathcal{R}$. For each movie j, we first draw its spatial popularity distribution $\boldsymbol{\xi}_j$, then draw a specific region from the distribution, and finally based on the functional distribution of this region, draw a cinema located at some location l to play the movie.

Through this generative process, we can infer the distribution of spatial popularity of movie j, denoted as ξ_j. Each dimension of ξ_j represents the popularity of movie j in the corresponding region. By default, the dimension of ξ_j is 50. With ξ_j, we can capture the influence of spatial popularity effect on ratings as follows.

$$\hat{r}_{ij}^{\text{Popularity}}(l) = \mathbf{d}_j^T \xi_j \tag{14}$$

In Eq. (14), $\mathbf{d}_j$ represents the weights of different regions, which needs to be learned to capture the popularity of different regions for movie j.

4.3 Two-stage Learning Algorithm

The final rating prediction formula is summarized as follows.

$$\begin{aligned}\hat{r}_{ij}(\tau, l) &= \hat{r}_{ij}^{\text{B}} + \hat{r}_{ij}^{\text{T}}(\tau) + \hat{r}_{ij}^{\text{S}}(l)\\ &= b_i + b_j + \mathbf{q}_j^T \mathbf{p}_i + \hat{r}_{ij}^{\text{Period}}(\tau) + \hat{r}_{ij}^{\text{Recency}}(\tau) + \hat{r}_{ij}^{\text{Crowd}}(\tau)\\ &\quad + \hat{r}_{ij}^{\text{Neighbor}}(l) + \hat{r}_{ij}^{\text{Popularity}}(l)\end{aligned} \tag{15}$$

In order to learn parameters, we need to minimize the regularized squared error, i.e.,

$$\begin{aligned}L(\Theta) &= \sum_{(i,j,\tau,l,r)\in\mathcal{R}} (r_{ij}(\tau, l) - \hat{r}_{ij}(\tau, l))^2 + \lambda\Omega(\Theta)\\ &= \sum_{(i,j,\tau,l,r)\in\mathcal{R}} \big(r_{ij}(\tau, l) - \hat{r}_{ij}(\tau, l)\big)^2 + \lambda\big(b_i^2 + b_j^2 + b_{i,\tau}^2 + b_{j,\tau}^2 + \rho_i^2\\ &\quad + \sum_{s=1}^{S} \alpha_{i,s}^2 + ||\mathbf{p}_i||^2 + ||\mathbf{q}_i||^2 + ||\mathbf{g}_i||^2 + ||\mathbf{d}_j||^2 + \sum_{s=1}^{S} ||\mathbf{p}_{i,s}^{\text{Period}}||^2\big)\end{aligned} \tag{16}$$

where $\Theta = \{b_i, b_j, b_{i,\tau}, b_{j,\tau}, \alpha_{i,s}, \rho_i, \mathbf{d}_j, \mathbf{g}_i, \mathbf{q}_j, \mathbf{p}_i, \mathbf{p}_{i,s}^{\text{Period}}, \mathbf{A}\}$ (we do not impose the regularization on $\mathbf{A}$), λ is the regularization coefficient.

The above parameters can be learned using the Stochastic Gradient Descent (SGD) method. Let $e_{ij}(\tau, l)$ denote the prediction error of a training sample (i, j, τ, l, r), i.e.,

$$e_{ij}(\tau, l) = r_{ij}(\tau, l) - \hat{r}_{ij}(\tau, l) \tag{17}$$

Then the gradient descent update formula of one training sample for parameters except $\mathbf{A}$ is as follows, where η denotes the learning rate.

$$\begin{aligned}b_i &\leftarrow b_i + \eta(e_{ij}(\tau, l) - \lambda \times b_i)\\ b_j &\leftarrow b_j + \eta(e_{ij}(\tau, l) - \lambda \times b_j)\\ b_{i,\tau} &\leftarrow b_{i,\tau} + \eta(e_{ij}(\tau, l) - \lambda \times b_{i,\tau})\\ b_{j,\tau} &\leftarrow b_{j,\tau} + \eta(e_{ij}(\tau, l) - \lambda \times b_{j,\tau})\\ \mathbf{p}_i &\leftarrow \mathbf{p}_i + \eta(e_{ij}(\tau, l) \times \mathbf{q}_j - \lambda \times \mathbf{p}_i)\\ \mathbf{q}_j &\leftarrow \mathbf{q}_j + \eta(e_{ij}(\tau, l) \times (\mathbf{p}_i + \textstyle\sum_{s=1}^{S} \alpha_{i,s} \times \mathbf{p}_{i,s}^{\text{Period}} \times h_s(\tau)) - \lambda \times \mathbf{q}_j)\\ \mathbf{p}_{i,s}^{\text{Period}} &\leftarrow \mathbf{p}_{i,s}^{\text{Period}} + \eta(e_{ij}(\tau, l) \times h_s(\tau) \times \alpha_{i,s} \times \mathbf{q}_j - \lambda \times \mathbf{p}_{i,s}^{\text{Period}}),\ s = 1,...,S\\ \mathbf{g}_i &\leftarrow \mathbf{g}_i + \eta(e_{ij}(\tau, l) \times \gamma_{i,\tau} - \lambda \times \mathbf{g}_i)\\ \mathbf{d}_j &\leftarrow \mathbf{d}_j + \eta(e_{ij}(\tau, l) \times \xi_j - \lambda \times \mathbf{d}_j)\\ \alpha_{i,s} &\leftarrow \alpha_{i,s} + \eta(e_{ij}(\tau, l) \times h_s(t) \times \mathbf{q}_j^T \mathbf{p}_{i,s}^{\text{Period}} - \lambda \times \alpha_{i,s}),\ s = 1,...,S\\ \rho_i &\leftarrow \rho_i + \eta(e_{ij}(\tau, l) \times \frac{\sum_{u\in N_i} w_{iu} \times (\bar{r}_{uj} - \bar{r}_u)}{\sum_{u\in N_i} w_{iu}} - \lambda \times \rho_i)\end{aligned} \tag{18}$$

Considering that the time complexity of optimizing $\mathbf{A}$ is high (i.e., $O(D^2M^2)$), the convergence speed will be slow and the running time will be too long if SGD is used to learn $\mathbf{A}$.

In order to accelerate the optimization, we take apart the spatial influence items, i.e., $\hat{r}_{ij}^{\text{Popularity}}(l)$ and $\hat{r}_{ij}^{\text{Neighbor}}(l)$, and rewrite the loss function as follows.

$$L(\Theta) = \sum\nolimits_{i,j} (r_{ij}(\tau, l) - \hat{r}_{ij}^{\text{B}} - \hat{r}_{ij}^{\text{T}}(\tau) - \hat{r}_{ij}^{\text{Popularity}}(l) - \hat{r}_{ij}^{\text{Neighbor}}(l))^2 + \lambda\Omega(\Theta) \tag{19}$$

To achieve an efficient joint learning, we design a two-stage learning method. The basic idea of this learning method is to separate the updating of $\hat{r}_{ij}^{\text{Neighbor}}(l)$ (i.e., related to $\mathbf{A}$) from the learning of other parts in Eq. (19).

At the first stage, we fix $\mathbf{A}$ and compute the Mahalanobis distance matrix of cinemas, denoted as $\mathcal{D} = \{dist_{iu}\}, i, u \in \{1, \ldots, M\}$, so $\frac{\sum_{u\in N_i} w_{iu} \times (\bar{r}_{uj} - \bar{r}_u)}{\sum_{u\in N_i} w_{iu}}$ is fixed at this stage. Then, all parameters except $\mathbf{A}$ are updated based on Eq. (18) using SGD on the full dataset. Besides, we denote as δ_{ij} the average prediction error over all time intervals except $\hat{r}_{ij}^{\text{Neighbor}}(l)$, i.e.,

$$\delta_{ij} = \mathbb{E}[r_{ij}(\tau, l) - \hat{r}_{ij}^{\text{B}} - \hat{r}_{ij}^{\text{T}}(\tau) - \hat{r}_{ij}^{\text{Popularity}}(l)] \tag{20}$$

At the second stage, the learning task is equivalent to minimizing the loss function in Eq. (21).

$$\begin{aligned}L(A) &= \sum\nolimits_j \sum\nolimits_i (\delta_{ij} - \hat{r}_{ij}^{\text{Neighbor}}(l))^2\\ &= \sum\nolimits_j \sum\nolimits_i (\delta_{ij} - \rho_i \times \frac{\sum_{u\in N_i} w_{iu} \times (\bar{r}_{uj} - \bar{r}_u)}{\sum_{u\in N_i} w_{iu}})^2\end{aligned} \tag{21}$$

At this stage, the remaining parameters other than $\mathbf{A}$ are fixed. After ϵ is set to $\frac{\sum_{u\in N_i} w_{iu} \times (\bar{r}_{uj} - \bar{r}_u)}{\sum_{u\in N_i} w_{iu}}$, the gradient of $\mathbf{A}$ over the full dataset is calculated as follows.

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{A}} &= -4\mathbf{A} \sum\nolimits_j \sum\nolimits_i \frac{\rho_i \times (\delta_{ij} - \hat{r}_{ij}^{\text{Neighbor}}(l))}{\sum_{u\in N_i} w_{iu}} \times\\ &\quad \sum\nolimits_{u\in N_i} w_{iu} \times [\epsilon - (\bar{r}_{uj} - \bar{r}_u)] \times (\mathbf{x}_i - \mathbf{x}_u)(\mathbf{x}_i - \mathbf{x}_u)^T\end{aligned} \tag{22}$$

The above loss function and the gradient calculation formula will be used as input to the L-BFGS [11] algorithm to approximate the Hessian matrix, which will improve the convergence speed.

An initialization of $\mathbf{A}$ is required to start above two-stage learning. We use PCA (Principal Components Analysis) to transform the POI data, and the learned projection matrix of PCA is used to initialize the matrix $\mathbf{A}$.

5 EVALUATION

In this section, we evaluate Pegasus extensively. Firstly, we give a case study to observe whether Pegasus is able to capture the changes in audience behaviors. Secondly, we conduct an ablation study on Pegasus to evaluate the contribution of three temporal components and two spatial components. Then we evaluate the recommendation performance of Pegasus, and compare it with several state-of-the-art recommendation methods. Finally, we demonstrate post-deployment evaluation for Pegasus.

5.1 Experimental Setting

Evaluation Metric. Four popular metrics in recommender systems have been used in our experiments to conduct an extensive evaluation: RMSE (Root Mean Square Error), MAE (Mean Absolute Error), NDCG (Normalized Discounted Cumulative Gain), and recall. Note

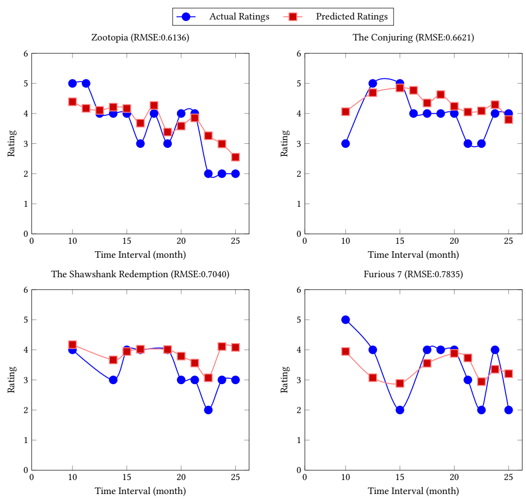

Figure 1. Actual and predicted ratings of four movies of the Zibo Cinema

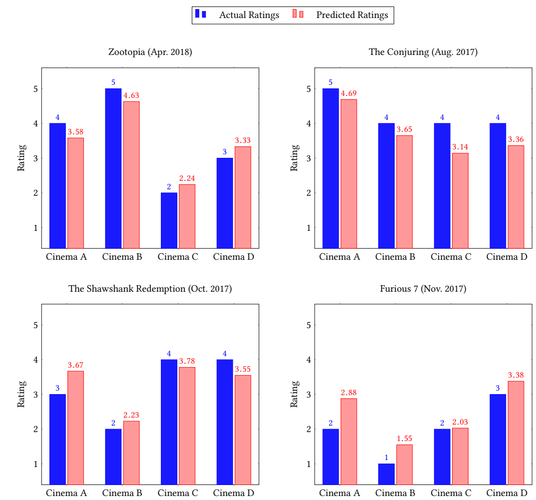

Figure 2. Actual and predicted ratings of movies of four cinemas at the same time intervals

that RMSE and MAE are rating-based metrics while NDCG and recall are ranking-based metrics.

Dataset. The data used in our evaluation are real-world data from iQIYI. We collect related data between July 1, 2016 and Sept. 1, 2018 from 207 on-demand cinemas. In this dataset, there are 743,558 consumption records of 5,879 movies in these 207 cinemas. We also collect content descriptions of these 5,879 movies. Specifically, we utilize the open APIs to crawl the data of POIs within 1 km of each on-demand cinema and employ the logarithmic transformation on the number of POIs in each type.

We evaluate three scenarios: recommending movies to on-demand cinemas in next month, next week and next day. By setting the time interval to one month, one week and one day respectively, we get three cinema profiles. Hereinafter they are referred to as the monthly dataset, weekly dataset and daily dataset, respectively. On each of these three datasets, we calculate the ratings of cinemas on movies by the method in Section 3. Note that, for a cinema, a movie may not be watched (no rating) in most days and mutations on watching behaviors may occur from time to time. Therefore, given a single time interval, the data in the daily dataset is sparser and more irregular than data in other datasets.

For experiments in Section 5.3 and Section 5.4, we use the data at continuous m time intervals of each cinema as training data, and the data at the next time interval as testing data. When the unit of time interval is month, week or day, m is set to 10, 50, or 300, respectively. Based on this rule, we form five groups of training data and testing data for each dataset (i.e., the monthly dataset, weekly dataset and daily dataset), respectively.

5.2 Case Study

In this subsection, we give the case study to observe whether Pegasus can follow the time- or space-varying features of audience behaviors in on-demand cinemas.

From the monthly dataset, we choose on-demand records of the four movies (i.e., Zootopia, The Conjuring, The Shawshank Redemption, Furious 7) of the on-demand cinema named "Watching together with iQIYI, Zibo Cinema". We plot the movie ratings between June 2017 and Sept. 2018, which are shown by blue lines with solid circles in Figure 1. Then, we run Pegasus and forecast the ratings of the same interval for these movies. Specifically, for predicting ratings of τ month (let $\tau \in [10, 25]$, where 10 denotes June 2017 and 25 denotes Sept. 2018), we employ the 10-month (i.e., $[\tau - 10, \tau - 1]$) data before τ month for training Pegasus and then run trained Pegasus to obtain the rating of the τ month. The predicted ratings are shown by red lines with solid squares in Figure 1. As shown in Figure 1, Pegasus can follow the prevailing trends of these movies over time.

Next, we observe at the same time interval the prevailing trends of movies in different cinemas. We choose four on-demand cinemas subordinate to "Watching together with iQIYI", i.e., Zibo Cinema (A), Changsha Desheng Recreational Centre Cinema (B), Shantou Jinping Cinema (C), Shanghai International Square Cinema (D). From the monthly dataset, we observe the 1-month ratings of four movies, i.e., Zootopia in Apr. 2018, The Conjuring in Aug. 2017, The Shawshank Redemption in Oct. 2017, and Furious 7 in Nov. 2017. We also use the 10-month data before the predicted month to train Pegasus and then predict the ratings of different movies of the above four cinemas. The actual and predicted ratings are shown by the dark blue and light red bars in Figure 2, respectively. The differences between the real ratings and the predictions are at most 0.88, with an average of 0.43.

5.3 Ablation Study

In this subsection, we conduct an ablation study on Pegasus to analyze the individual contribution of each component (i.e., three temporal components and two spatial components) and joint contributions of multiple component compositions. Specifically, due to the limited number of pages, we only list the average results of the ablation study on the monthly dataset in Table 1.

For various combinations (each based on the baseline predictor), Table 1 lists their performance in terms of RMSE, MAE and NDCG@50, along with their performance differences with the baseline predictor. As shown in Table 1, individually adding each component proposed in this paper to the original baseline would contribute

Table 1. Ablation study over spatio-temporal components in Pegasus

	RMSE	RMSE %Improv.	MAE	MAE %Improv.	NDCG@50	NDCG@50 %Improv.
Baseline	0.8969	0	0.7170	0	0.7302	0
Baseline+Period	0.8657	3.48%	0.6843	4.56%	0.7537	3.22%
Baseline+Recency	0.8762	2.31%	0.6943	3.17%	0.7393	1.25%
Baseline+Crowd	0.8617	3.92%	0.6842	4.57%	0.7519	2.97%
Baseline+Neighbor	0.8781	2.10%	0.6969	2.80%	0.7384	1.12%
Baseline+Popularity	0.8780	2.11%	0.6967	2.83%	0.7369	0.92%
Baseline+Temporal: (Period+Recency+Crowd)	0.8235	8.18%	0.6442	10.15%	0.7805	6.89%
Baseline+Spatial: (Neighbor+Popularity)	0.8538	4.81%	0.6797	5.20%	0.7449	2.01%
Pegasus: (Baseline+ Temporal +Spatial)	0.8017	10.61%	0.6274	12.50%	0.7947	8.83%

to recommendation performance. Specifically, the crowd drifting effect leads to the most significant improvement, i.e., reducing 3.92% in RMSE, 4.57% in MAE and increasing 2.97% in NDCG@50. Moreover, the combination of the three temporal dynamic components and the combination of the two spatial influence components also lead to the significant improvement. The former shows nearly double performance improvement than the latter.

In addition, we conduct the one-tailed t-test on the improvements of all metrics of Pegasus where p-value is set to 0.05. Taking RMSE as an example, the t-test result shows that at a 95% confidence level, the hypothesis that the improvement of Pegasus compared to baseline is not larger than 10% is rejected, which means that the improvement of RMSE is larger than 10%. The same procedure is also adapted to obtain t-test results for the improvements of MAE (> 12%) and NDCG@50 (> 8%).

These results illustrate that Pegasus is effective in modeling the spatio-temporal effects embedded in audience behaviors, and each component in the additive rating is reasonable besides explainable.

5.4 Performance Comparison

We choose eight methods to conduct comparative experiments. Two of them, i.e., UCF (User-based Collaborative Filtering) and PMF (Probabilistic Matrix Factorization) [14], belong to collaborative filtering methods that do not model contextual information. The left six are methods that can model contextual information. They are listed as follows.

- T-UCF (Time weight User-based Collaborative Filtering) [5], which uses an exponential decay function to indicate the decaying of old data over time.
- TimeSVD++ [10]. We remove the modeling of implicit feedback in TimeSVD++ because of the high time complexity and no improvement in our scenario. Hereinafter, the modified method is referred to as TimeSVD.
- CE (Collaborative Evolution) [12], a time-aware matrix factorization method, which employs the temporal regression to deal with the dynamic evolution of the user latent vectors over time.
- CTR (Collaborative Topic Regression) [21], which fuses the content of items through LDA and considers that there is a linear relationship between the item latent vector and the item topic distribution.
- CDL (Collaborative Deep Learning) [22], a hierarchical Bayesian model, which jointly performs deep representation learning for the content information and collaborative filtering for the rating (feedback) matrix. In our scenario, we incorporate both the content descriptions of movies and the POI information around cinemas into CDL.
- GRU (GRU-based RNN) [4], which employs GRUs to capture long-term data features and can interpret the global dynamic evolution of item ratings.

For adapting to our scenario, we make some adjustments to the comparative methods. For methods that do not consider temporal information (UCF, PMF, CTR, CDL), since a cinema has different ratings on a movie over different time intervals, we perform the exponentially weighted moving average on the ratings in the training data for the m time intervals before the prediction interval. That is, the aggregation rating of the cinema i on the movie j in training data, i.e., $\bar{r}_{ij}(m)$, can be calculated by $\bar{r}_{ij}(\tau) = \zeta \cdot r_{ij}(\tau) + (1-\zeta) \cdot \bar{r}_{ij}(\tau-1)$, where the decay coefficient ζ is set to 0.9 by default. For the GRU-based RNN, we concatenate the ratings on a movie from a cinema at m time intervals into a sequence of length m. Since the RNN requires more training data, we construct multiple sequences using a sliding window of size $m/2$ over a sequence. Finally, each sequence s_{ij} represents the sequence of ratings of the cinema i on the movie j over $m/2$ time intervals. Further, due to the differences between cinema preferences, we add cinema latent vector p_i and movie latent vector q_j as the context, concat them with the rating at each time interval to form the input of GRU-based RNN.

For PMF, TimeSVD, CTR, CDL and Pegasus, we set the values of latent dimensions k to 100, 50, 20 for the monthly dataset, weekly dataset and daily dataset, respectively, and use the grid search technique to choose the appropriate values for other parameters.

For each dataset (i.e., the monthly dataset, weekly dataset and daily dataset), we conduct each method over five groups of training data and testing data, and record the average results of five experiments on each dataset in Table 2. Specially, the last column of Table 2 gives the improvement proportions of Pegasus relative to the best method of the other eight methods.

Table 2. Performance comparison of different methods

Dataset	Metrics	UCF	PMF	T-UCF	TimeSVD	CE	CTR	CDL	GRU	Pegasus	% Improv.
Monthly Dataset	RMSE	0.9597	0.9003	0.9494	0.8512	0.9653	0.9031	0.8630	0.9510	**0.8017**	5.82%
	MAE	0.7731	0.7176	0.7598	0.6897	0.7560	0.7169	0.6842	0.7515	**0.6274**	8.30%
	NDCG@50	0.6984	0.7261	0.7094	0.7775	0.7438	0.7406	0.7333	0.7158	**0.7947**	2.21%
Weekly Dataset	RMSE	0.9151	0.9021	0.9064	0.7707	1.0567	0.9329	0.8709	0.8918	**0.7158**	7.12%
	MAE	0.7521	0.7311	0.7445	0.6193	0.8235	0.7458	0.7091	0.7299	**0.5466**	11.74%
	NDCG@50	0.6137	0.6585	0.6378	0.7527	0.7300	0.6712	0.6516	0.6985	**0.7861**	4.44%
Daily Dataset	RMSE	1.0584	1.0885	1.0535	0.9004	1.4019	1.1456	1.0495	1.1552	**0.7173**	20.34%
	MAE	0.8958	0.9063	0.8945	0.7186	1.1090	0.9420	0.8813	0.9314	**0.5476**	23.80%
	NDCG@50	0.6114	0.6222	0.6158	0.6865	0.6669	0.6384	0.6199	0.6772	**0.7847**	14.30%

As shown in Table 2, all the metrics of Pegasus on the three datasets are superior to other methods. Similar to Section 5.3, we conduct the t-test ($p = 0.05$) for the improvements of all metrics of Pegasus compared to other methods. Taking the daily dataset as an example, the t-test results show that the improvements of Pegasus are significant in terms of RMSE (> 20%), MAE (> 20%), and NDCG@50 (> 10%). Further, we can obtain the following observations and inferences:

- The methods which can model contextual information perform better especially in terms of NDCG@50. For example, T-UCF with a time decay function is better than UCF. CTR models the movie content information and behaves better than PMF in terms of NDCG@50. However, CTR performs worse than Pegasus due to its lack of the ability to model temporal information. Pegasus fuses the time, space and content information and is superior to other methods.
- CE shows the poor performance, perhaps because the overall preference evolution in on-demand records is not obvious. Further, the good performance of TimeSVD and Pegasus may come from their advantage in the modeling of local temporal dynamics. Such dynamics are common in the on-demand cinema scenario, e.g., audiences of on-demand cinemas may have a certain time-related pattern in the watching behaviors, and may be affected by the recent social hotspots, such as the newly-released movies.
- The performance of deep learning methods, i.e., CDL and GRU is not satisfactory in our scenario. In the current situation, the actual amount of data in the on-demand cinemas is limited, which makes it hard to show the advantages of deep learning methods.
- The unit of time interval has an influence on the recommendation performance. The smaller the unit is, the greater the advantage of Pegasus over others is. Most of the methods perform best on the weekly dataset, which may be due to the consistency of partition of records and weekly routines among actual human activities. However, on the daily dataset, the performance of TimeSVD, T-UCF, CE and GRU drops sharply, although Pegasus still performs well. Obviously, the data sparseness and irregularity in daily dataset increase the difficulty of accurately predicting the rating of the next day. These comparative methods fail to deal with the data sparseness and irregularity. To some extent, our method is also affected by data sparseness, such as the audience crowd distribution and the bias of a cinema in each time interval. However, the performance can be compensated by other parts of our method. For example, the recency effect based on the movie content information considers the temporal locality of the data. For the audience crowd effect, although the learning of the audience crowd distribution in each interval is not accurate enough, the crowd drifting effect can be fine tuned by the coefficient g_i shared over all time intervals of cinema i. In addition, the neighboring effect between cinemas makes it possible to refer to the watching behavior of other cinemas. In other words, the fact that Pegasus incorporates a variety of contextual information beyond the on-demand records helps Pegasus to alleviate data sparseness and irregularity.

5.5 Post-deployment Evaluation

Currently, in the actual on-demand cinemas, limited by the human costs, it is unfeasible to split the audience flow into different groups to give them different recommended lists. In other words, conducting the post-deployment evaluation for all the comparative methods is unpractical. Therefore, we only conduct post-deployment evaluation for our own method Pegasus. Our purpose is to qualitatively show that Pegasus is effective in the real-world environment.

We have developed a customized WeChat applet on smart phones, which is the client of the Pegasus system and can receive the recommended list from Pegasus. By the applet, the staffs/hosts of on-demand cinemas can browse the recommended list, initiate the download for movies not in the local storage, and recommend the movies on the list to audiences. Since Oct. 2018, the staffs/hosts of 30 on-demand cinemas have been equipped with the applet.

The statistics reveal that, for a cinema, the number of on-demand records per day is relatively limited while the average number of on-demand records per week is 150 or so. Therefore, we recommend 150 movies to a cinema once a week. At the end of each week, we collect actual on-demand records as feedback, and then compare the recommendation list with the actual on-demand list to evaluate the recommendation performance. Finally, the Pegasus system is incrementally updated with the new on-demand records and produces a new recommendation list for the next week. The above

Table 3. Post-deployment evaluation results of 30 cinemas

	Average Recall@50	Average Recall@100
2018.10.22-2018.10.28	36.48%	47.02%
2018.10.29-2018.11.04	41.21%	51.91%
2018.11.05-2018.11.11	37.68%	47.64%
2018.11.12-2018.11.18	42.21%	51.28%
2018.11.19-2018.11.25	35.73%	45.37%
2018.11.26-2018.12.02	38.67%	47.16%
2018.12.03-2018.12.09	36.34%	48.02%
2018.12.10-2018.12.16	36.71%	49.71%
Average over 8 Weeks	**38.13%**	**48.51%**

process is carried out for eight consecutive weeks. We calculate the recall@K for cinemas, where K is set to 50 or 100. Further, we record the average recall@50 and average recall@100 for each week and show the results in Table 3.

As we can see from Table 3, during the 8-week post-deployment evaluation, the average recall@50 and average recall@100 are 38.13% and 48.51%, respectively. This result is identical with expectation, indicating that the recommended movies can meet the partial interest of potential audiences of the cinemas. Cinema staffs/hosts are suggested to download the recommended movies in advance for convenience of audiences.

6 CONCLUSION

This paper proposes the Pegasus method for recommending movies to on-demand cinemas. Pegasus models the influence of time-varying and space-varying features in the audience behaviors. More importantly, Pegasus is easily explainable. Both experimental results and actual operational feedback show that Pegasus can provide effective recommendation results, which can support the precise marketing for on-demand cinema operations.

As our future work, we intend to collect the missed movies in the actual scenarios, that is, the choices of audiences but not in the recommendation list or the movie library of cinemas. We will treat the missed movies as a type of strong feedback to improve the system performance.

ACKNOWLEDGMENTS

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

REFERENCES

[1] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. 2017. Variational inference: A review for statisticians. *J. Amer. Statist. Assoc.* 112, 518 (2017), 859–877.

[2] Da Cao, Xiangnan He, Lianhai Miao, Yahui An, Chao Yang, and Richang Hong. 2018. Attentive group recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 645–654.

[3] Chen Cheng, Haiqin Yang, Irwin King, and Michael R Lyu. 2012. Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks.. In *Aaai*, Vol. 12. 17–23.

[4] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[5] Yi Ding and Xue Li. 2005. Time weight collaborative filtering. In *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 485–492.

[6] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 173–182.

[7] Seyedabbas Hosseini, Ali Khodadadi, Keivan Alizadeh, Ali Arabzadeh, Mehrdad Farajtabar, Hongyuan Zha, and Hamid RR Rabiee. 2018. Recurrent poisson factorization for temporal recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2018).

[8] Wang-Cheng Kang, Chen Fang, Zhaowen Wang, and Julian McAuley. 2017. Visually-aware fashion recommendation and design with generative image models. In *Data Mining (ICDM), 2017 IEEE International Conference on*. IEEE, 207–216.

[9] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 233–240.

[10] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 447–456.

[11] Dong C Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical programming* 45, 1-3 (1989), 503–528.

[12] Zhongqi Lu, Sinno Jialin Pan, Yong Li, Jie Jiang, and Qiang Yang. 2016. Collaborative Evolution for User Profiling in Recommender Systems.. In *IJCAI*. 3804–3810.

[13] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 287–296.

[14] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*. 1257–1264.

[15] Tieyun Qian, Bei Liu, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2019. Spatiotemporal representation learning for translation-based poi recommendation. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 18.

[16] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. 2011. *Recommender Systems Handbook*. Springer US. 1–35 pages.

[17] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 111–112.

[18] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009).

[19] John Z Sun, Dhruv Parthasarathy, and Kush R Varshney. 2014. Collaborative Kalman Filtering for Dynamic Matrix Factorization. *IEEE Trans. Signal Processing* 62, 14 (2014), 3499–3509.

[20] Mengting Wan, Di Wang, Jie Liu, Paul Bennett, and Julian McAuley. 2018. Representing and Recommending Shopping Baskets with Complementarity, Compatibility and Loyalty. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1133–1142.

[21] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 448–456.

[22] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 1235–1244.

[23] Weiqing Wang, Hongzhi Yin, Ling Chen, Yizhou Sun, Shazia Sadiq, and Xiaofang Zhou. 2017. ST-SAGE: A spatial-temporal sparse additive generative model for spatial item recommendation. *ACM Transactions on Intelligent Systems and Technology (TIST)* 8, 3 (2017), 48.

[24] Kilian Q Weinberger and Gerald Tesauro. 2007. Metric learning for kernel regression. In *Artificial Intelligence and Statistics*. 612–619.

[25] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 325–334.

[26] Dawei Yin, Liangjie Hong, Zhenzhen Xue, and Brian D Davison. 2011. Temporal dynamics of user interests in tagging systems. In *Twenty-Fifth AAAI conference on artificial intelligence*.

[27] Hongzhi Yin, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen. 2013. LCARS: a location-content-aware recommender system. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 221–229.

[28] Jia-Dong Zhang, Chi-Yin Chow, and Yanhua Li. 2015. iGeoRec: A personalized and efficient geographical location recommendation framework. *IEEE Transactions on Services Computing* 8, 5 (2015), 701–714.

[29] Shuai Zhang, Lina Yao, and Aixin Sun. 2017. Deep learning based recommender system: A survey and new perspectives. *arXiv preprint arXiv:1707.07435* (2017).

[30] Shenglin Zhao, Tong Zhao, Haiqin Yang, Michael R Lyu, and Irwin King. 2016. STELLAR: Latent Ranking for Successive Point-of-Interest Recommendation.. In *AAAI*. 315–322.

[31] Guanjie Zheng, Susan L Brantley, Thomas Lauvaux, and Zhenhui Li. 2017. Contextual spatial outlier detection with metric learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2161–2170.