Context-Aware Personalized Video Recommendation Algorithms in Mobile Environments (Postprint)

Authors: Luo Guoqian, Liu Zhiyong, Zhang Lin, Zhang Jiaxin, He Zhuoheng, Zhang Xin

Date: 2019-04-01T00:00:00+00:00

Abstract

To address the issue of low accuracy in personalized movie and TV recommendations when using traditional recommendation algorithms in mobile environments, a context-aware matrix factorization algorithm is proposed. Building upon the basic matrix factorization framework, this algorithm incorporates global bias and context bias to predict unknown ratings. Its advantages are twofold: firstly, the matrix factorization approach significantly reduces the matrix dimensions compared to the original rating matrix; secondly, it fully integrates the influence of contextual factors on ratings, thereby enhancing prediction accuracy. Experimental results on the LDOS-CoMoDa dataset demonstrate that the proposed algorithm achieves superior accuracy compared to user-based collaborative filtering, basic matrix factorization, and baseline prediction algorithms.

Full Text

Preamble

Vol. 37 No. 5
Application Research of Computers
ChinaXiv Partner Journal

Research on Personalized Film Recommendation Algorithm Based on Context-Awareness in Mobile Environment

Luo Guoqian[1] , Liu Zhiyong[1],[1] †, Zhang Lin[2], Zhang Jiaxin[1] , He Zhuoheng[1] , Zhang Xin[1]
(1. a. School of Information Science & Technology; b. Digital Learning

Engineering Research Center of China Educational Ministry, Northeast Normal University, Changchun 130117, China;
2. School of Software, Jilin University, Changchun 130012, China)

Abstract: Traditional recommendation algorithms exhibit low accuracy when applied to personalized film recommendation in mobile environments. To address this issue, this paper proposes a context-aware matrix factorization algorithm that incorporates global bias and context bias into the basic matrix factorization framework for predicting unknown ratings. The proposed algorithm offers two key advantages: first, matrix factorization significantly reduces the dimensionality compared to the original rating matrix; second, it fully integrates the influence of contextual factors on ratings, enabling more accurate predictions. Experimental results on the LDOS-CoMoDa dataset demonstrate that the proposed algorithm outperforms user-based collaborative filtering, basic matrix factorization, and Baseline prediction algorithms in terms of accuracy.

Keywords: film recommendation; matrix factorization; context-aware

0 Introduction

The advent of the mobile Internet era and the proliferation of mobile devices such as smartphones and tablets have led to an increasing number of users accessing film services via mobile platforms. According to the *China Film Audience Viewing Behavior Research Report 2014–2015*, nearly 57.3% of users watch film and television programs on mobile devices among non-theatrical viewing channels, indicating that mobile viewing has become increasingly popular. Meanwhile, the problem of information overload on mobile Internet has become more severe, and personalized recommendation represents an effective solution. It actively recommends valuable information to users based on their characteristics and needs [1]. However, personalized recommendation in mobile environments is highly context-sensitive—users' needs and interests vary with different contexts, particularly in film viewing. For instance, adventurous individuals may prefer crime or adventure films; males tend to favor war and action films, while females may be more interested in romance and family films. During major holidays such as Spring Festival, users may be more willing to watch New Year or comedy films. Unlike traditional personalized recommendations that consider only user and item dimensions, contextual factors must be fully incorporated. Context-aware personalized film recommendation in mobile environments can provide users with film information that matches their current context in real time, better satisfying personalized viewing needs and delivering more accurate services [2].

The integration of personalized recommendation services with contextual factors has attracted widespread scholarly attention, yielding various research outcomes across different application domains. In mobile personalized reading recommendation, Zeng Ziming et al. [3] proposed a context-aware mobile reading personalized information recommendation model by incorporating contextual conditional entropy into traditional collaborative filtering algorithms. In mobile electronic resource recommendation, Tian Xuewei [4] introduced contextual factors into personalized recommendation systems and proposed a hybrid algorithm combining context-based and content-based methods, leveraging users' historical context and preference information to provide personalized electronic resource services. In personalized health services, Kim et al. [5] utilized context models to extract missing user preference values during collaborative filtering, combining context-awareness with collaborative filtering for personalized health service systems. In mobile digital library resource recommendation, Hong Liang et al. [6] designed a context-aware personalized resource recommendation approach tailored to mobile digital library resource distribution and push characteristics.

In summary, ensuring the quality of personalized recommendation in mobile environments requires adequate consideration of contextual factors. However, most existing studies employ neighborhood-based collaborative filtering methods that primarily rely on user-item rating matrices. As is well known, user-item rating matrices can be extremely sparse in real-world scenarios. To extract more useful information, this paper adopts matrix factorization for dimensionality reduction. Moreover, neighborhood-based collaborative filtering is primarily a statistical method without a learning process, whereas matrix factorization is theoretically grounded as a machine learning approach that builds optimal models by optimizing a defined objective function [7].

Therefore, this paper proposes the Context-Aware Matrix Factorization (CAMF) algorithm, focusing on context-awareness in personalized film recommendation. Compared with user-based collaborative filtering and basic matrix factorization, CAMF possesses context-aware capability, thereby providing users with film information that matches their current context. This paper details the CAMF algorithm and conducts comparative experiments using the LDOS-CoMoDa dataset against user-based collaborative filtering, basic matrix factorization, and Baseline prediction algorithms, demonstrating its superior accuracy.

1.1 Context and Context-Awareness

The most widely accepted definition of context was proposed by Dey [8]: context is a combination of information that describes an entity in a situation, where an entity can be a user or an object associated with application interaction. This paper defines mobile viewing context as the combination of information affecting users' current viewing needs, including temporal information, geographical location information, weather information, etc.

The concept of context-awareness was first introduced by Schilit et al. [9] in 1994, who defined it as software' s ability to adapt its state according to the current context. Dey' s [10] definition is now widely accepted: context-awareness refers to the process of using contextual information to provide relevant information or services to users. In mobile environments, context-awareness for user viewing involves providing better viewing information and services based on users'current contextual information.

1.2 Context Elements

Different application domains or systems consider different context elements. For instance, mobile electronic resource recommendation systems primarily consider device information, location, network conditions, temporal information, and user preferences [4]; adaptive learning systems consider physical context, user context, and learning context [11]; search engine systems consider search intent, geographical location, time, and device information.

For personalized recommendation in the film domain, Li Sheng [12] categorized user viewing context into static and dynamic contexts based on the state dimension of contextual information. Static context includes age, gender, occupation, and personality, while dynamic context includes time, location, weather, mood, companion, etc.

Košir et al. [13] constructed the LDOS-CoMoDa dataset incorporating contextual elements for film recommendation research. The dataset includes 12 contextual elements: time, daytype, season, location, weather, social, endEmo, mood, dominantEmo, physical, decision, and interaction.

Drawing from Li Sheng' s ideas and incorporating elements from the LDOS-CoMoDa dataset, this paper classifies mobile viewing context elements into static and dynamic information. Static information refers to users' long-term, stable characteristics, including age, gender, occupation, education, and personality. Dynamic information refers to dynamically changing characteristics of users or their surroundings, including time, location, weather, date, companion, mood, physical condition, post-viewing emotion, viewing decision, and viewing frequency. A preliminary set of 15 context elements is proposed, as shown in Table 1.

Table 1 Classification of Context Elements in Mobile Movie Watching

Category	Context Element	Example Values
Static Information	Age	1-18, 19-40, 41-60, 60+ years
	Gender	Male, Female
	Occupation	Student, Teacher, Businessperson, etc.
	Education	Primary, Middle, High School, Bachelor, etc.
	Personality	Outgoing, Introverted, Steady, etc.
Dynamic Information	Time	Morning, Afternoon, Evening, etc.
	Location	Home, Company, Friend' s house, etc.
	Weather	Sunny, Cloudy, Rainy, Snowy, Windy, etc.
	Date	Working day, Weekend, Holiday
	Companion	Alone, Family, Friends, etc.
	Mood	Happy, Calm, Sad, etc.

Category	Context Element	Example Values
	Physical Condition	Healthy, Sick
	Post-viewing Emotion	Sad, Calm, Excited, etc.
	Viewing Decision	Self-selected, Recommended by others
	Viewing Frequency	First time, More than once

1.3 User-Based Collaborative Filtering Algorithm

User-based collaborative filtering is one of the most widely applied and successful collaborative filtering recommendation algorithms. Its main idea is to identify similar users through their ratings on items and generate recommendations based on the assumption that similar users share similar interests. The algorithm consists of two main steps: (a) calculating user similarity based on their item ratings—various similarity metrics exist, including cosine similarity and Pearson correlation coefficient, with this paper adopting the latter; (b) finding nearest neighbors for the current user based on calculated similarities, predicting the user' s ratings for unrated items based on these neighbors, and generating recommendations accordingly.

1.4 Basic Matrix Factorization Algorithm

The fundamental idea of basic matrix factorization is to decompose the user-item rating matrix R into two low-dimensional latent feature matrices: a user latent feature matrix P and an item latent feature matrix Q, such that their product approximates R [14]:

$$R_{m\times n} \approx P_{m\times d} Q_{d\times n}^T = \begin{pmatrix} p_1^T q_1 & p_1^T q_2 & \cdots & p_1^T q_n \\ p_2^T q_1 & p_2^T q_2 & \cdots & p_2^T q_n \\ \vdots & \vdots & \ddots & \vdots \\ p_m^T q_1 & p_m^T q_2 & \cdots & p_m^T q_n \end{pmatrix}$$

where $R \in \mathbb{R}^{m\times n}$ denotes the user-item rating matrix; $P \in \mathbb{R}^{m\times d}$ denotes the user latent feature matrix; $Q \in \mathbb{R}^{n\times d}$ denotes the item latent feature matrix; p_u represents the latent feature vector of user u; q_i represents the latent feature vector of item i; and $d \ll \min(m, n)$ denotes the dimensionality of feature vectors. The rating of user u on item i is predicted using the inner product of p_u and q_i:

$$\hat{r}_{ui} = p_u^T q_i$$

To obtain the feature vectors p_u and q_i, the matrix factorization algorithm defines an objective function to minimize the squared error:

$$L = \min_{P,Q} \frac{1}{2} \sum_{(u,i)\in K} (r_{ui} - p_u^T q_i)^2 + \frac{\lambda}{2}(\|p_u\|^2 + \|q_i\|^2)$$

where K is the set of observed (user, item) rating pairs; r_{ui} is the actual rating of user u on item i; and λ is the regularization coefficient to prevent overfitting. The optimal solution can be obtained using stochastic gradient descent (SGD) [15].

1.5 Baseline Prediction Algorithm

The Baseline prediction algorithm posits that user ratings on items are influenced by inherent characteristics of users or items themselves [16]. For instance, lenient users tend to give ratings higher than the average, while strict users give lower ratings. Similarly, high-quality film items generally receive higher ratings, whereas low-quality items receive lower ratings. These user- or item-independent influencing factors are termed global bias. The Baseline rating prediction model incorporating global bias is:

$$\hat{r}_{ui} = \mu + b_u + b_i$$

where $\hat{r}_{ui}$ represents the predicted rating of user u on item i; μ denotes the global average rating of the training set; b_u denotes the bias term for user u (user-independent influence); and b_i denotes the bias term for item i (item-independent influence).

To solve for b_u and b_i, an objective function L is defined and optimized using stochastic gradient descent [15]:

$$L = \min_{b_u,b_i} \frac{1}{2} \sum_{(u,i)\in K} (r_{ui} - \mu - b_u - b_i)^2 + \frac{\lambda}{2}(b_u^2 + b_i^2)$$

For example, to predict user $user1$' s rating on film $movie1$, suppose user1' s preference for movie1 is 2.5 points, movie1' s reputation is 0.5 points lower than other films (i.e., $b_i = -0.5$), user1 is a lenient rater who tends to rate 0.3 points higher than average (i.e., $b_u = 0.3$), and user1 in the current context tends to rate this type of film 0.2 points higher (i.e., $b_c = 0.2$). Then user1' s predicted rating for movie1 would be $2.5 - 0.5 + 0.3 + 0.2 = 2.5$ points.

2.1 Rating Prediction Model

The basic matrix factorization algorithm learns latent feature vectors of users and items for prediction, where item latent feature vectors represent the degree of each item characteristic, and user latent feature vectors represent user preferences for each characteristic. Their inner product indicates the user' s preference for the item.

As the Baseline prediction algorithm demonstrates, user ratings are susceptible to global bias from inherent user or item characteristics. Therefore, this paper first incorporates global bias into the user-item preference modeling (i.e., basic matrix factorization), denoted as:

$$\hat{r}_{ui} = \mu + b_u + b_i + p_u^T q_i$$

where parameters μ, b_u, and b_i share the same meanings as in the Baseline prediction algorithm.

However, in real life, especially in mobile environments, user ratings exhibit context sensitivity. For instance, user mood and post-viewing experience influence film ratings. To incorporate contextual influence, Baltrunas et al. [17] proposed in their CAMF-CC model that each context element exerts the same influence on ratings for items within the same category. This paper adopts Baltrunas' concept, defining the influence of context elements on a specific item category as context bias, denoted as $b_{c_j,t}$:

$$b_{c_j,t} = \sum_{j=1}^{k} b_{c_j,t}$$

where t represents the item category; c_j denotes context elements, with k elements in total; and $b_{c_j,t}$ represents the influence of context element c_j on category t film items.

The final CAMF rating prediction model incorporating both global bias and context bias is:

$$\hat{r}_{ui} = \mu + b_u + b_i + p_u^T q_i + \frac{1}{k}\sum_{j=1}^{k} b_{c_j,t}$$

where γ is the learning rate (step size for gradient descent), optimized through experiments.

2.2 Training Model

The rating prediction model aims to make predicted ratings as close as possible to actual values. The following objective function is defined for model training:

$$L = \min_{P,Q,b_u,b_i,b_{c_j,t}} \frac{1}{2}\sum_{(u,i)\in K}\left(r_{ui} - \mu - b_u - b_i - p_u^T q_i - \frac{1}{k}\sum_{j=1}^{k} b_{c_j,t}\right)^2 + \frac{\lambda}{2}\left(\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2 + \sum_{j=1}^{k} b_{c_j,t}^2\right)$$

where r_{ui} is the actual rating in the training set; K is the set of observed (user, item) pairs; and λ is the regularization coefficient to prevent overfitting. Other parameters share the same meanings as in the basic matrix factorization algorithm and the rating prediction model above.

To optimize the objective function L, stochastic gradient descent [15] is employed. For each rating r_{ui} in the training set, parameters are iteratively updated following the negative gradient direction of L to obtain the optimal solution. The calculation formulas are as follows: Equation (10) is the error formula; Equations (11)-(15) calculate the negative gradient directions (via partial derivatives); and Equations (16)-(20) are parameter update formulas.

$$e_{ui} = r_{ui} - \hat{r}_{ui} = r_{ui} - \mu - b_u - b_i - p_u^T q_i - \frac{1}{k}\sum_{j=1}^{k} b_{c_j,t}$$

$$\frac{\partial L}{\partial q_i} = -e_{ui} p_u + \lambda q_i$$

$$\frac{\partial L}{\partial p_u} = -e_{ui} q_i + \lambda p_u$$

$$\frac{\partial L}{\partial b_i} = -e_{ui} + \lambda b_i$$

$$\frac{\partial L}{\partial b_u} = -e_{ui} + \lambda b_u$$

$$\frac{\partial L}{\partial b_{c_j,t}} = -e_{ui} + \lambda b_{c_j,t}$$

$$q_i \leftarrow q_i + \gamma\left(e_{ui} p_u - \lambda q_i\right)$$

$$p_u \leftarrow p_u + \gamma\left(e_{ui} q_i - \lambda p_u\right)$$

$$b_i \leftarrow b_i + \gamma\left(e_{ui} - \lambda b_i\right)$$

$$b_u \leftarrow b_u + \gamma\left(e_{ui} - \lambda b_u\right)$$

$$b_{c_j,t} \leftarrow b_{c_j,t} + \gamma\left(e_{ui} - \lambda b_{c_j,t}\right)$$

2.3 Algorithm Flow and Flowchart

This section presents the CAMF algorithm flow and flowchart for computing the rating prediction function of target user u on item i.

Input: User-item rating matrix R, learning rate γ, regularization coefficient λ, iteration count n, and feature vector dimension d.

Initialization: a) Split data into training set, validation set, and test set. Initialize parameters: fill user latent feature matrix P and item latent feature matrix Q with random numbers proportional to $\sqrt{d}$; initialize b_u and b_i as zero vectors; initialize $b_{c_j,t}$ for category t items as zero vectors of dimension contextValueNum (number of context variable values).

Iteration: b) For each rating record (u, i, r_{ui}) in the training set: (a) Calculate negative gradient directions using Equations (11)-(15); (b) Update parameters using Equations (16)-(20).

Calculate validation set MAE (see Section 3.2) until convergence or reaching iteration count n.

Output: Parameters of the rating prediction function for target user u on item i.

The algorithm flow is illustrated in Figure 1 [Figure 1: see original paper].

2.4 Time Complexity Analysis

The time complexity of user-based collaborative filtering is primarily in similarity calculation: $O(n^2)$ for n users and m items. For basic matrix factorization with k rating records, feature dimension f, and s iterations, the complexity is $O(k \cdot f \cdot s)$. For Baseline prediction, the complexity is $O(k \cdot s)$. For CAMF with c context elements, the complexity is $O(k \cdot (f + c) \cdot s)$.

Thus, CAMF' s time complexity is similar to basic matrix factorization, while Baseline prediction is less complex. If $k \cdot (f + c) \cdot s > n^2$, CAMF' s complexity exceeds that of user-based collaborative filtering. Generally, matrix factorization algorithms are slightly more complex than user-based collaborative filtering due to multiple iterations, but there is no qualitative difference [7].

3 Experiments and Analysis

This chapter conducts comparative experiments to validate the effectiveness of the proposed CAMF algorithm against user-based collaborative filtering, basic matrix factorization, and Baseline prediction algorithms. We first introduce the dataset, experimental environment, and evaluation method, then present experimental design and comparative analysis.

3.1 Dataset and Experimental Environment

LDOS-CoMoDa is a film rating dataset that is open-source and rich in contextual elements, making it widely used by context-aware recommendation researchers. This dataset is selected for our experiments. It contains 2,296 rating records from 121 users on 1,232 films, with ratings ranging from 1 to 5. Detailed dataset parameters are shown in Table 2 .

The LDOS-CoMoDa dataset includes 30 variables, among which time, daytype, location, weather, social, mood, endEmo, physical, decision, and interaction belong to dynamic context information, while age and sex belong to static context information. Therefore, 12 context elements are considered in the experiments. All elements take positive integer values, with missing values set to -1. For convenience, the data is preprocessed as follows: (a) specific age values are categorized into four groups: 1-18, 19-40, 41-60, and 60+ years; (b) missing age values are filled with the mean, while missing values of other elements are filled with the mode (most frequent value) of that element. Descriptions of these context elements are provided in Table 3 .

Table 2 Description of LDOS-CoMoDa Movie Rating Dataset

Parameter	Value
Rating Range	[1, 5]
Total Ratings	2,296
Number of Users	121
Number of Items	1,232
Total Variables	30
Sparsity	98.45%

Sparsity: Ratio of unrated entries to total entries in the rating matrix

Table 3 Description of Context Elements

Element	Description	Values
time	Time of day	Morning(1), Afternoon(2), Evening(3), Night(4)
daytype	Day type	Working day(1), Weekend(2), Holiday(3)
location	Location	Home(1), Public place(2), Friend’ s house(3)

Element	Description	Values
weather	Weather condition	Sunny/clear(1), Rainy(2), Stormy(3), Snowy(4), Cloudy(5)
social	Companion	Alone(1), My partner(2), Friends(3), Colleagues(4), Parents(5), Public(6), My family(7)
mood	Mood	Positive(1), Neutral(2), Negative(3)
endEmo	Emotion after viewing	Sad(1), Happy(2), Scared(3), Surprised(4), Angry(5), Disgusted(6), Neutral(7)
physical	Physical condition	Healthy(1), Ill(2)
decision	Viewing decision	User’ s choice(1), Given by other(2)
interaction	Viewing frequency	First(1), n-th(2)
age	Age group	1-18(1), 19-40(2), 41-60(3), 60+(4)
sex	Gender	Male(1), Female(2)

Experimental Environment: Windows 10 OS, 8 GB RAM, Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz. Experimental code is implemented in Python 3.

3.2 Evaluation Method

Mean Absolute Error (MAE) is adopted to evaluate recommendation accuracy. MAE measures the deviation between predicted and actual ratings, where smaller values indicate higher accuracy and recommendation quality:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|P_i - R_i|$$

where N is the test set size; P_i is the predicted rating; and R_i is the actual rating.

3.3 Experimental Results and Analysis

To ensure robust results, 10-fold cross-validation is employed. For user-based collaborative filtering, the dataset is split into 10 folds, with 1 fold as test set and 9 folds as training set in each iteration. For basic matrix factorization, Baseline prediction, and CAMF algorithms, the dataset is split into 10 folds, with 1 fold as test set, 1 fold as validation set, and 8 folds as training set in each iteration. Final results are averaged across 10 runs.

User-Based Collaborative Filtering Experiment: The neighbor count k is varied from 5 to 50 with step size 5. As shown in Figure 2 [Figure 2: see original paper], MAE fluctuates between 0.96 and 0.98, reaching the minimum value of 0.959818 at $k = 30$, which yields optimal recommendation performance.

Basic Matrix Factorization Experiment: Regularization coefficient λ and learning rate γ are both varied from 0.01 to 0.05 with step size 0.01. With iteration count set to 100 and feature dimension to 10, results in Figure 3 [Figure 3: see original paper] show MAE fluctuating between 1.94 and 2.04, achieving the minimum value of 1.95012 at $\lambda = 0.03$ and $\gamma = 0.01$.

Baseline Prediction Algorithm Experiment: Using the same parameter settings as basic matrix factorization, results in Figure 4 [Figure 4: see original paper] show MAE fluctuating between 0.8 and 0.84, reaching the minimum value of 0.806097 at $\lambda = 0.03$ and $\gamma = 0.04$.

CAMF Algorithm Experiment: With identical parameter settings, results in Figure 5 [Figure 5: see original paper] show MAE fluctuating between 0.74 and 0.78, achieving the minimum value of 0.746469 at $\lambda = 0.04$ and $\gamma = 0.01$.

3.3.5 Comparative Analysis

The optimal and average MAE values for all algorithms are summarized in Figure 6 [Figure 6: see original paper]. In terms of optimal values: user-based collaborative filtering achieves MAE = 0.959818 at $k = 30$; basic matrix factorization achieves MAE = 1.95012 at $\lambda = 0.03, \gamma = 0.01$; Baseline prediction achieves MAE = 0.806097 at $\lambda = 0.03, \gamma = 0.04$; and CAMF achieves MAE = 0.746469 at $\lambda = 0.04, \gamma = 0.01$, representing improvements of 22.22%, 61.72%, and 7.4% respectively. In terms of average values: user-based collaborative filtering has average MAE = 0.964228; basic matrix factorization has average MAE = 1.967378; Baseline prediction has average MAE = 0.816832; and CAMF has average MAE = 0.773913, representing improvements of 19.74%, 60.66%, and 5.25% respectively. These results demonstrate CAMF' s superior accuracy and recommendation quality.

4 Conclusion

This paper proposes the Context-Aware Matrix Factorization (CAMF) algorithm to improve recommendation accuracy for personalized film recommendation in mobile environments. By incorporating global bias and context bias into the basic matrix factorization framework, CAMF reduces rating matrix dimensionality while fully integrating contextual influences on user ratings, resulting in more accurate predictions and better recommendation performance. Experimental comparisons with user-based collaborative filtering, basic matrix factorization, and Baseline prediction algorithms show average MAE reductions of approximately 19.7%, 60.7%, and 5.3% respectively, validating the algorithm' s effectiveness. A limitation of this work is that all context elements are assigned equal weight, ignoring their varying influence on recommendation results—an issue to be addressed in future research.

References

[1] Hong L, Zhai C, Wu Z. Research on situational perception information recommendation based on common interest in mobile environment [J]. Information Theory & Practice, 2014, 37(11): 124-128.

[2] Guo S, Li X. Construction of user information demand model for mobile library based on context awareness [J]. Information Theory & Practice, 2014, 37(8): 64-68.

[3] Zeng Z, Chen B. Research on personalized reading recommendation based on situational perception in mobile environment [J]. Information Theory & Practice, 2015, 38(12): 31-36.

[4] Tian X. Research on recommendation technology of mobile electronic resources based on context awareness [J]. Information Theory & Practice, 2015, 38(5): 86-89.

[5] Kim J, Lee D, Chung K Y. Item recommendation based on context-aware model for personalized u-healthcare service [J]. Multimedia Tools & Applications, 2014, 71(2): 855-872.

[6] Hong L, Qian C, Fan X. Research on situational perception personalization recommendation method of mobile digital library resources [J]. Modern Library and Information Technology, 2016, 32(7): 110-119.

[7] Xiang L. Recommended system practice [M]. Beijing: People' s Posts and Telecommunications Press, 2012.

[8] Dey A K. Providing architectural support for building context-aware applications [J]. PhD Thesis, Georgia Institute of Technology, 2000, 25(2): 106-111.

[9] Schilit B, Adams N, Want R. Context-aware computing applications [C]// Proc of Workshop on Mobile Computing Systems & Applications. 1994: 85-90.

[10] Dey A K. Understanding and using context personal and ubiquitous computing journal [J]. Personal & Ubiquitous Computing, 2001, 5(1): 4-7.

[11] Jiang Y, Li W, Yang J. Context-aware-driven adaptive personalized learning and interaction [J]. Journal of Intelligent Systems, 2014(1): 60-68.

[12] Li S. Personalized film recommendation based on context perception [D]. Beijing: Beijing University of Posts and Telecommunications, 2012.

[13] Košir A, Odić A, Kunaver M, et al. Database for contextual personalization [J]. Elektrotehniski Vestnik/Electrotechnical Review, 2011, 78(5): 270-274.

[14] Yu Y, Gao Y, Wang W, et al. Recommendation algorithm for integrating user social status and matrix decomposition [J]. Journal of Computer Research and Development, 2018, 55(1): 113-124.

[15] Salakhutdinov R, Mnih A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo [C]// Proc of International Conference on Machine Learning. ACM Press, 2008: 880-887.

[16] Wang J, Zhang P, Liu Y. Research on decomposition algorithm based on improved band offset probability matrix [J]. Journal of Computer Applications, 2017, 34(5): 1397-1400.

[17] Baltrunas L, Ludwig B, Ricci F. Matrix factorization techniques for context aware recommendation [C]// Proc of ACM Conference on Recommender Systems. 2011.

Note: Figure translations are in progress. See original paper for figures.