A Time-Context-Dependent Resource Diffusion Algorithm Based on User Splitting

Hao FAN¹, Xiao-yan Yu¹, De-qin SHU*, Hongbin CAO¹

¹College of Information Science and Engineering, Shandong Agricultural University, Tai’an, 271018, China

*Corresponding author’s e-mail: shudeqin@126.com, fanhao@sdau.edu.cn

Abstract. Personalized recommendation for smartphone applications is studied, and a time-context-dependent resource diffusion algorithm based on user splitting is proposed. The algorithm introduces time context information. According to this idea, the algorithm can make full use of user preference data information, cluster mobile applications to get more user preference categories and optimize recommendation results. In the personalized recommendation process of mobile applications, because users and projects are dynamic changes, this paper analyses the dynamic change process of the algorithm in detail, and enhances the applicability of the algorithm. Experimental results show that the proposed algorithm improves the accuracy of recommendation results compared with the previous resource diffusion algorithm.

1. Introduction

Personalized recommendation was proposed as an independent concept in the 1990s [1-2]. Tapestry, the first recommendation system named Tapestry was proposed by Goldberg et al [3]. Recommendation systems are bridges between users and applications or projects of interest. It can be divided into three categories according to the different ways of recommendation: user behaviour-based recommendation, content-based recommendation and mixed recommendation [4]. Analyse users' favourite projects and recommend similar projects. Find user B who has similar interests with user A and recommend the project that user B likes to A. Analyse the user data to get the characteristics that users like, and recommend the projects that contain these characteristics. According to the principle of recommendation system, it can be divided into demographic recommendation, knowledge-based recommendation and graph-based recommendation. Recently, collaborative filtering algorithms are also commonly used as recommendation techniques.

Researchers have proposed many algorithms for massive data in mobile applications. Sun J.L, et al presents a CBR recommendation system, which first establishes a multi-case database of users and their preferences, and then combines expert system analysis to analyze the potential preferences of users, or to analyze users' thinking patterns, so as to form personalized cases of different users' preferences for recommendation [5]. A recommendation algorithm using cloud as an intermediate node for recommendation is proposed-DocCloud Recommendation System by Vera-Del-Campo J. By adding the recommendation method of intermediate nodes, the user's factors are well protected. The recommender's recommendation and the recommender's acceptance of the recommendation can be anonymously carried out. The system introduces many security mechanisms into the recommendation algorithm [6]. Chang H.Y, et al propose an Internet TV recommendation system, which successfully used cloud-assisted channel recommendation and adopted association rules to
solve the problem of slow channel switching. A large number of users' history logs are processed in parallel by distributed processing on multiple computers. The experimental results show that this method can help users find their favorite program channels in a short time superior to COV and MFS algorithms [7]. In literature [8], a system of promoting e-learning system for learners is presented. The fuzziness and uncertainty of user's learning activities, i.e. the uncertain category of a learning activity, are studied. The priority relationships of multiple learning activities that users are interested in are given. The problem of individuation in the process of learning recommendation is solved. Liu H et al propose a recommendation system model based on collaborative filtering. The model not only considers the local context information of user ratings, but also the global preference of user behavior [9]. Wang L.C et al proposed a fashion recommendation system for consumers to choose the most relevant fashion design scheme and provide personalized fashion products. Fuzzy knot and fuzzy relation are used to formally describe the perceptual data of consumers and designers, and fuzzy cognitive map is used to model [10]. In addition, there are many recommendation algorithms and systems based on different applications [11-13].

These methods greatly improve the quality of recommendations, but these algorithms require multi-dimensional contextual information, such as installation information, detailed records, implicit feedback, and so on. First, a large amount of user information needs to be collected, which consumes more time and space resources. Second, the current mobile system does not have relevant data collection and processing modules, which is not conducive to the promotion and application. Considering the above problems, this paper proposes a time-context-based resource diffusion algorithm, which improves the accuracy of the recommendation algorithm under the condition of the allowable resources of mobile devices (time and space resources).

2. Analysis of common resource diffusion algorithm

Literature [14] gives the basic principle of current common resource diffusion algorithm- Bipartite network projection, as shown in figure 1.

Let bipartite graph \( G = (P,Q,R) \), where \( P = \{p_1, p_2, \ldots, p_s\} \) is the set of items, \( Q = \{q_1, q_2, \ldots, q_u\} \) is the set of users, and \( R \) is the edge set representing the preference relationship between users and items.

![Figure 1. Calculation process of bipartite network projection.](image)

The calculation process of the user's preference for the items in the algorithm based on bipartite network projection is shown in figure 1.

Let \( f(p_i) \) an integer greater than 0, which denote the initial resource of the first node in set \( P \), \( f(q_i) \) is the resource value of the first element in \( Q \), and the resource is diffused from \( P \) to \( Q \). \( M \) is a resource transfer matrix. \( m_{i,j} \) represents the proportion of resources transferred from \( p_i \) to \( p_j \). The elements \( i \) in set \( P \) are redistributed by two-step diffusion.

The resource transfer matrix of figure 1 is

\[
\begin{bmatrix}
3/7 & 1/7 & 6/21 \\
1/12 & 4/9 & 7/21 \\
5/12 & 7/18 & 4/9
\end{bmatrix}
\]. When computing a user's preference vector, the resource transfer matrix is multiplied by the user's item selection vector. The
calculation process of the item preference vector of the first user (black solid circle) in figure 1 is as follows. The value of the element in the result vector reflects the user's preference degree for the item.

\[
\begin{bmatrix}
3/7 & 1/7 & 6/21 & 1 \\
1/12 & 4/9 & 7/21 & 0 \\
5/12 & 7/18 & 4/9 & 0
\end{bmatrix} \times \begin{bmatrix}
1 \\
0 \\
0
\end{bmatrix} = \begin{bmatrix}
3/7 \\
1/12 \\
5/12
\end{bmatrix}
\]

Previous resource diffusion algorithms simulated the process of material diffusion in physics and obtained the recommended results by using simple two-step resource allocation in the user-item bipartite graph. The results of this kind of resource diffusion algorithm are better than the classical collaborative filtering algorithm. However, the research of Neal et al. shows that this kind of algorithm has a good performance in the test set, but the effect is not good in practical application, because this kind of resource diffusion algorithm does not consider the time factor, that is, it directly obtains the recommendation results from the historical records. Therefore, the algorithm proposed in this paper divides users whose preferences change with time into multiple users. In other words, time context information is introduced from the perspective of user splitting to improve the recommendation accuracy of the algorithm.

3. Time-context-dependent resource diffusion algorithm based on user splitting

The algorithm proposed in this paper is named as time-context-dependent resource diffusion algorithm based on user splitting. The mathematical model and algorithm steps of the algorithm are described in detail in this section.

3.1. Mathematical model of Algorithm

The mathematical model is expressed as a four partite graph with \( r + r + n + s \) nodes, \( G = (P, U, I, T, E) \), as shown in figure 2. Assuming that the number of users is \( r \), the number of items is \( n \), the number of tags is \( s \). \( U = (u_1, u_2, \ldots, u_r) \) is the set of users, \( P = (p_1, p_2, \ldots, p_r) \) is the set of time property to each user, each user can choose different items at different times. \( I = (i_1, i_2, \ldots, i_n) \) is the set of items. \( T = (t_1, t_2, \ldots, t_r) \) is the label classification of items. \( E \) is the edge set representing the preference relationship between users and items.

![Mathematical model of time-context-dependent resource diffusion algorithm based on user splitting.](image)

Figure 2. Mathematical model of time-context-dependent resource diffusion algorithm based on user splitting.

3.2. Algorithm description

In literature [15-16], we have done some basic work, and proposed the recommendation algorithm of time context and user splitting algorithm. Here we give the step description of time-context-dependent resource diffusion algorithm based on user splitting.

STEP1: Resource diffusion, as shown in figure 2. The resource allocation results of elements in item \( I \) are obtained by using the resource diffusion algorithm for user \( U \) and item \( I \), which are stored as the resource transfer matrix \( M \).
Let the distance between elements in item $I$ be as formula (2).

$$
\text{dis}_i = \begin{cases} 
\log m_{i,j} \times \log m_{j,i} & m_{i,j} \neq 0 \text{ and } m_{j,i} \neq 0 \\
+\infty & \text{otherwise}
\end{cases}
$$

STEP3: User splitting. The users whose preferences change with the time dimension are divided into multiple users by using the splitting idea. When making item recommendation to users with user splitting, new users obtained by using the splitting idea can participate in the algorithm operation as historical users as shown in figure 3(b) to 3(c). The method is as follows: Let the ordered set $U$ be the list of user $U_a$ preference items sorted in chronological order, and $U_1$ be the ordered set of the initial $m$ items. Let $U_2 = U - U_1$. The difference ratio between $U_1$ and $U_2$ in various tags is defined as formula (3).

$$
\text{Ratio}_{a_1-a_2} = \frac{1}{2|U|} \sum_{i=1}^{\text{tags}} \text{differs}
$$

Where $|\text{tags}|$ the module of the tag is set and $\text{differs}$ is the difference of the number of items in each tag. $|U|$ is module of the set of users.

$T$ is the threshold set by the system. When $\text{Ratio} \geq T$, it is believed that user preference has changed and new users $U_a$ are obtained, that is, user $U_a$ is split into $U_a$ and set $U_1$ is the list of user $U_a$ preference items. When $\text{Ratio} \leq T$, the user preference is considered unchanged. Let $U_1 = U - U_1 - U_2$. $\text{Ratio}_1$ is the difference ratio between $U_1$ and $U_3$. When $\text{Ratio}_1 \geq T$, we get new user $U_a$, set $U_1 \cup U_2$ is the list of user $U_a$ preference items, and when $\text{Ratio}_1 < T$, we get new user $U_a$ by splitting, set $U_1 \cup U_2 \cup U_1$ is the list of user $U_a$ preference items. The $U_a$ split ends when the module of the remaining preference item set $U_{last}$ is less than $m$. $U_{last}$ is the last user set split from $U_a$.

STEP4: Production of user tag preference vector. As shown in figure 4, let the edge between each user-item have a weight of 1, distribute the items evenly to relevant tags, sum and normalize the user preference vector. For example, the user $u_1$ preference vector is $V_{u_1} = [1/3, 1/19, 5/18, 1/21, 1/3]$.

STEP5: Items additions and deletions. Let the total downloads of the mobile application item since its release be $x$, and when the download times of nearly a month are less than $m$% of the total downloads, the item is considered to have entered a declining period. Delete the item and re-execute the algorithm. Add the items, when the number of mobile application item uploads times above $m$% of the current item set $|I|$.
STEP6: User additions and deletions. When a new user $U_{\text{new}}$ joins the set $U$ and the new user downloads application item $I_j$, add the mapping $U_{\text{new}} - I_j$. When a user's related items are deleted, the user is deleted.

STEP7: Generate a list of recommendations. Based on the user tag preference vector, the items within each tag are recommended in proportion.

4. Experimental verification and analysis

4.1. Data sets and measurement standard

The data set was obtained by crawling the 360 app circle in December, 2018, including 1,175 Android mobile apps and their downloaders and download date. Each mobile app has 1 to 4 tags, which are selected from 37 system tags provided by the app circle. The data set is divided into 5 data subsets in ascending order according to download time. The first m ($1 \leq m \leq 4$) data subsets are taken as training sets, and the latter 5-m data subsets are taken as test sets for algorithm verification. Take the top 10 average accuracy as the measurement standard.

4.2. Experimental results and analysis

Set $R=0.16$ as the diffusion ratio in the experiment [17]. Tag clustering is performed when the number of mobile apps in the tag is greater than 60. The commonly used resource diffusion algorithm [14] and the algorithm (time-context-dependent resource diffusion algorithm based on user splitting) proposed in this paper are tested. The comparison of experimental results is shown in table 1 and figure 5. The number of test subsets in the table 1 is the first m subsets sorted in chronological order, as described in section 4.1. Common algorithms refer to commonly used resource diffusion algorithms, and
time-content-dependent algorithm refers to the time-content-dependent resource diffusion algorithm based on user fission algorithm proposed in this paper.

Table 1. Comparison of average precision of algorithm.

| Number of test subsets | Common algorithms | time-context-dependent algorithm |
|------------------------|-------------------|---------------------------------|
| 1                      | 18.2%             | 21.3%                           |
| 2                      | 23.9%             | 28.2%                           |
| 3                      | 26.2%             | 32.1%                           |
| 4                      | 27.6%             | 36.4%                           |

Figure 5. Comparison of algorithm accuracy.

It can be seen from table 1 and figure 5 that the accuracy of the algorithm proposed in this paper is obviously better than the traditional resource diffusion algorithm. This is because the algorithm uses the idea of user splitting to introduce time context information, makes full use of the time attribute of data, and improves the performance of the algorithm in practical applications. With the increase of test subset data, the average precision of both algorithms increases, but the algorithm proposed in this paper increases more obviously. It indicates that with the increase of data set, the user's interest list increases and the proportion of users participating in user splitting increases. This algorithm can make more effective use of the time context attribute, thus improving the accuracy of the algorithm.

5. Conclusion
This paper proposes a time-context-dependent resource diffusion algorithm based on user splitting. The algorithm improves the accuracy by introducing time context information through user splitting. User splitting avoids the emergence of super-large users and the excessive influence of super-large users on the recommendation results of other users. To make it easier for users to upload applications, many practical application systems provide more granular tags for users to select and mark. This algorithm gets more refined tags by tag clustering, and solves the problem of less tag number and larger granularity in the system. In fact, the structure of user-item bipartite graph is decomposed, the operation scale of the algorithm is reduced, and the efficiency of the algorithm is improved.

The common resource diffusion algorithm is compared with the algorithm proposed in this paper. The data set was obtained by crawling The 360 app circle in 12, 2018, including 1,065 Android mobile apps and their downloaders and download date. Experimental results show that the accuracy of the proposed algorithm is better than the common resource diffusion algorithm.
Acknowledgments
This work is supported by Natural Science Foundation of Shandong Province (ZR2016FB19, ZR2019MF026 and ZR2017MD018), National Natural Science Foundation of China (41471299) and Technology Innovation Fund Project of Shandong Agricultural University (23659).

References
[1] Resnick, P., Iacovou, N., Suchak, M., et al. (1994) GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of the ACM conference on Computer supported cooperative work. New York, pp.175-186.
[2] Hill, W., Stead, L., Rosenstein, M., et al. (1995) Recommending and evaluating choices in a virtual community of use. In: Sigchi Conference on Human Factors in Computing Systems. ACM Press/Addison-Wesley Publishing Co. San Diego, California. pp. 194-201.
[3] Goldberg, D. (1992) Using collaborative filtering to weave an information tapestry. Communications of the Acm, 35(12):61-70.
[4] Xiang, L. (2012) The practice of recommendation system. Posts and Telecommunications Press, Beijing.
[5] Vera-Del-Campo, J., Pegueroles, J., Hernández-Serrano, J., et al. (2014) DocCloud: a document recommender system on cloud computing with plausible deniability. Information Sciences, 258(3):387-402.
[6] Chang, H.Y. Huang, S. C., Lai, C. C. (2014) A personalized IPTV channel-recommendation mechanism based on the MapReduce framework. Journal of Supercomputing, 69(1):225-247.
[7] Wu, D., Lu, J., Zhang, G. (2015) A fuzzy tree matching-based personalized E-Learning recommender system[J]. IEEE Transactions on Fuzzy Systems, 23(6):2412-2426.
[8] Liu, H., Hu, Z., Mian, A., et al. (2014) A new user similarity model to improve the accuracy of collaborative filtering. Knowledge-Based Systems, 56(3):156-166.
[9] Wang, L. C., Zeng, X.Y., Koehl, L. et al. (2015) Intelligent fashion recommender system: fuzzy logic in personalized garment design. IEEE Transactions on Human-Machine Systems, 45(1):95-109.
[10] Zhou, T., Ren, J., Medo, M. (2007). Bipartite network projection and personal recommendation. Physical Review E, 76(4):046115.
[11] Cao, H.B. (2017) Research and application of mobile application recommendation system based on time context. Master's thesis, Shandong agricultural university.
[12] Cao, H.B., Zhang, L. (2017) Integrated diffusion-based recommendation algorithm based on user division. Journal of china academy of electronics and information technology, 12(2):159-163.
[13] Dakkel, A.M., Malazi, H.T., Mahdavi, M. (2018) A social recommender system using item asymmetric correlation. Applied Intelligence, 48(3):527–540.