An Improved Information Kernel Extraction Method in Collaborative Filtering Recommendation (Postprint)

Authors: Zhang Wenjing, Li Jinping, Yang Jun

Date: 2018-11-29T00:00:00+00:00

Abstract

Recommender systems (RS) assist users in discovering interesting information within massive data resources and provide accurate personalized recommendations. Information kernel-based recommendation algorithms can substantially reduce the temporal cost of the recommendation process. To address the scalability issues inherent in collaborative filtering recommendation algorithms, this paper proposes improved information kernel extraction methods, namely IFB (IFrequency-based) and IRB (IRank-based), built upon the original frequency-based (FB) and rank-based (RB) approaches. Furthermore, we introduce the concept of an optimization set in the neighbor-finding stage, wherein the most similar neighbors are identified for each user within this set. Experimental results demonstrate that the proposed methods achieve more accurate recommendation outcomes, effectively reduce Mean Absolute Error (MAE), and attain higher precision and recall rates, thereby delivering superior recommendation performance.

Full Text

Preamble

Improved Extraction Method of Information Core in Collaborative Filtering Recommendation

Wenjing Zhang, Jinping Li†, Jun Yang
(School of Electronics & Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China)

Abstract: Recommender systems (RS) help users discover interesting information within vast data resources and provide accurate personalized recommendations. Recommendation algorithms based on information core can

significantly reduce time costs during the recommendation process. To address scalability issues in collaborative filtering recommendation algorithms, this paper proposes improved information core extraction methods—IFB (Improved Frequency-based) and IRB (Improved Rank-based)—building upon existing frequency-based (FB) and rank-based (RB) approaches. The concept of an optimization set is introduced for identifying the most similar neighbors, enabling neighbor selection for each user within this set. Experimental results demonstrate that the proposed methods yield more accurate recommendations, effectively reduce Mean Absolute Error (MAE), and achieve higher precision and recall rates, thereby delivering superior recommendation performance.

Keywords: recommender systems; collaborative filtering; information core

0 Introduction

With the rapid development and popularization of the Internet, data resources have increased exponentially, making it difficult for users to efficiently select useful information from massive resources, thus creating the information overload problem. Consequently, personalized recommender systems have emerged as a solution [1]. Existing personalized recommendation methods typically fall into three categories: content-based filtering (CBF) [2–4], collaborative filtering (CF) [5–7], and hybrid filtering (HF) that combines CBF and CF [8–10]. Among these, collaborative filtering has proven the most widely applied and successful [11].

Collaborative filtering generates personalized recommendations by analyzing user-item interactions captured in collected rating pairs. Since it does not require preprocessing of item or user features, it is not dependent on any specific domain. However, this advantage introduces scalability issues: when the number of users or items is large, time consumption grows exponentially with data scale. To address this problem, we assume that certain "expert" users possess deep understanding of objects in specific domains. By referencing these experts, recommender systems can provide satisfactory recommendations for ordinary users. Additionally, some malicious online users attempt to bias system outputs [12]. Therefore, by investigating user roles in recommendations, we can exclude irrelevant and unreliable users, thereby improving algorithm efficiency and robustness [13]. The information core-based collaborative filtering algorithm presented in this paper effectively solves this problem. We refer to expert users as core users, and the collection of core users as the information core. Core users constitute approximately 20% of the entire system, yet their recommendation accuracy can reach 90% of that achieved by all users. Since core users represent only about 20% of total users, they significantly reduce the number of users involved in recommendation, thereby alleviating the scalability issues inherent in traditional collaborative filtering algorithms.

Wu et al. [14] proposed a trapezoidal fuzzy rating model to address problems

such as discrete ratings' inability to reasonably express user opinions and the sparsity issues in traditional collaborative filtering. This algorithm demonstrates outstanding performance when data is sparse and user count far exceeds item count. Rong et al. [15] introduced user similarity concepts to tackle efficiency and accuracy degradation when collaborative filtering is applied to social networks, providing evaluation methods for recommendation quality and user satisfaction that effectively improve recommendation accuracy and efficiency in social networks. Huang et al. [16] proposed a hybrid recommendation algorithm combining LDA_MF, LDA_CF, and traditional item-based collaborative filtering to address the problem of existing methods focusing on accuracy while neglecting diversity, making recommendations more diverse while maintaining high accuracy. In this paper, we target the scalability problem in collaborative filtering algorithms and propose improved information core extraction methods –IFB (Improved Frequency-based) and IRB (Improved Rank-based)–building upon the frequency-based and rank-based methods proposed by Zeng et al. [17], thereby substantially improving recommendation system performance.

1 Preliminary Knowledge

This paper employs user-based collaborative filtering [18] with cosine similarity as the similarity metric between users. We assume the rating matrix contains historical ratings from m users on n items. The CF algorithm proceeds as follows:

a) Similarity Calculation. Based on rating vectors of target user u and other user v in the rating matrix, their similarity is calculated using Equation (1):

$$sim(u,v) = \frac{\sum_{i=1}^{n} r_{ui} \cdot r_{vi}}{\sqrt{\sum_{i=1}^{n} r_{ui}^2} \cdot \sqrt{\sum_{i=1}^{n} r_{vi}^2}}$$

where $sim(u,v)$ denotes the similarity between target user u and user v, r_{ui} represents the rating of target user u on item i, r_{vi} represents the rating of user v on item i, and n denotes the number of items.

b) Neighbor Selection. Based on the similarity matrix S, k most similar users are selected for each target user u as the neighbor set N_u.

c) Rating Prediction. Using target user u and its neighbor set N_u, the predicted rating $\hat{r}_{ui}$ of target user u for unrated item i is calculated using Equation (2):

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in N_u} sim(u,v) \cdot (r_{vi} - \bar{r}_v)}{\sum_{v \in N_u} sim(u,v)}$$

where $\bar{r}_u$ denotes the average rating of target user u, $sim(u, v)$ represents the similarity between target user u and user v, r_{vi} denotes user v' s rating on item *i$, and* $\bar{r}_v$ *represents user* v*' s average rating.

d) Effectiveness Evaluation. Mean Absolute Error (MAE), precision, and recall are used as metrics to evaluate recommendation effectiveness.

(a) MAE Metric. MAE, the most commonly used metric, is defined in Equation (3):

$$MAE = \frac{1}{|T_u|} \sum_{u \in T_u} \frac{1}{|I_u|} \sum_{i \in I_u} |\hat{r}_{ui} - r_{ui}|$$

where r_{ui} represents the actual rating of user u on item i in the test set, I_u denotes the set of items rated by user u, $|I_u|$ represents the number of items in the set, T_u denotes the set of all users in the test set, and $|T_u|$ represents the number of users in the set. MAE measures the average difference between predicted and actual ratings in the test data matrix; smaller MAE values indicate better algorithm performance.

(b) Precision Metric. Precision for target user u is calculated using Equation (4):

$$P_u = \frac{R_u}{N}$$

where R_u represents the number of items in the recommendation list that match items rated by target user u in the test set, and N represents the length of the recommendation list.

System precision P is calculated using Equation (5):

$$P = \frac{1}{|U|} \sum_{u \in U} P_u$$

where U denotes the user set and $|U|$ represents the number of users. Higher precision indicates better algorithm performance.

(c) Recall Metric. Recall for target user u is calculated using Equation (6):

$$Re_u = \frac{R_u}{W}$$

where R_u represents the number of items in the recommendation list that match items rated by target user u in the test set, and W represents the number of items rated by target user u in the test set.

System recall Re is calculated using Equation (7):

$$Re = \frac{1}{|U|} \sum_{u \in U} Re_u$$

where U denotes the user set and $|U|$ represents the number of users. Higher recall indicates better algorithm performance.

As evident from the above process, CF algorithms spend most of their time on similarity computation—first calculating similarities between the target user and all other users, then selecting the L most similar users as neighbors. If we can identify fewer but more reliable users, similarity computation can be performed on a smaller user set, thereby reducing online recommendation time. The method proposed in this paper aims to find such a smaller yet more effective user set.

---

2 Algorithm Description

Building upon Zeng et al.' s algorithm, this paper introduces the concept of an optimization set for finding the most similar neighbors, enabling neighbor selection for each user within this set.

2.1 FB-Based Algorithm Model

The FB-based information core identification method utilizes similarity information between users. First, it calculates the similarity matrix S for all user pairs. Based on S, it identifies each user' s k most similar neighbors, generating a top-K neighbor matrix M. Elements in M are user IDs, with each row representing a user' s top-K most similar users (i.e., the user' s top-K nearest neighbor list). For example, element m_{31} in matrix M represents the user ID most similar to user 3, and m_{32} represents the second most similar user to user 3. Then, it counts the occurrence frequency c of each user in matrix M—i.e., how many times each user appears in other users' top-K nearest neighbor lists. A higher occurrence count c indicates greater similarity with other users, implying more effective recommendation information carried by the user and thus higher importance to the system. Finally, users with the highest occurrence counts are selected to form the information core of the recommender system. The specific process is illustrated in [Figure 1: see original paper].

2.2 RB-Based Algorithm Model

The RB-based algorithm model is similar to the FB-based model. The FB method only counts how many times a user appears in other users' top-K neighbor lists without considering the position. However, the top-K list represents a descending order of similarity, where earlier positions indicate higher similarity. The RB method incorporates position information: if user i belongs to user j' s

top-K neighbors at position p, user i receives a weight of $\frac{1}{p}$. If user i appears in multiple users' top-K neighbor lists, the sum of these weights becomes the user' s final weight $\sum \frac{1}{p}$. Users with the highest total position weights are selected as the information core. The specific process is also illustrated in [Figure 1: see original paper].

2.3 Improved IFB and IRB Algorithm Models

Both FB and RB algorithm models select neighbor lists based solely on user similarity magnitudes without fully utilizing user-item rating information. This paper proposes a novel approach to neighbor list selection that leverages rating information more effectively to achieve better recommendation results.

The new neighbor selection process works as follows:

a) Randomly split user ratings into a training set (80% of ratings) and an optimization set (20% of ratings).

b) Select a rating r_{ai} from the optimization set, where r_{ai} represents user a' s rating on item i.

c) In the training set, select the top n users who have rated item i and are most similar to user a as the neighbor list top-N_{ai} for rating r_{ai}.

d) Find the corresponding neighbor list top-N_{ji} for each rating in the optimization set, where top-N_{ji} represents the neighbor list for user j' s rating r_{ji} on item i.

After generating neighbor lists, the IFB algorithm model proceeds identically to the FB model: it constructs a neighbor matrix W from all rating neighbor lists, counts each user' s occurrence frequency in matrix W (i.e., how many times each user appears in nearest neighbor lists for ratings in the optimization set), and selects users with the highest occurrence frequencies to form the information core. The IRB algorithm model similarly follows the original RB model' s procedure after neighbor matrix generation. By finding neighbor lists for every rating in the optimization set, our approach fully utilizes rating information to generate neighbor lists. The specific process is illustrated in [Figure 2: see original paper].

To concretely illustrate the information core extraction process for FB, RB, IFB, and IRB algorithms, consider the following example using ratings from five users on five items, split into training and optimization sets. The training and optimization set ratings are shown in and , respectively. shows five users (u1-u5) and their ratings on five items (I1-I5) in the training set. shows the same five users' ratings on the same five items in the optimization set. presents the cosine similarities between these five users.

The FB and RB algorithm models select information cores as shown in [Figure 1: see original paper]. For each user, the top two most similar users are selected as neighbors based on similarity values from . For instance, for user u1, the two

most similar users are u5 and u4, making u1' s neighbor list {u5, u4}. With information core size set to 2, the FB algorithm extracts the information core as {u1, u4}, while the RB algorithm (where u5 and u1 both have weights of 1.5) randomly selects one to combine with u4, resulting in {u4, u5}.

The IFB and IRB algorithm models select information cores as shown in [Figure 2: see original paper]. For the optimization set rating r_{12} (user u1' s rating on item I2), users u2, u3, and u4 have rated item I2 in the training set. Based on similarities in , the top two users most similar to u1 among these are u4 and u3, which are selected as the neighbor list for r_{12}. Following this process for all optimization set ratings, the IFB algorithm extracts the information core as {u1, u3}, while the IRB algorithm (where u3 and u4 both have total weights of 2) randomly selects one to combine with u1, yielding {u1, u4}.

---

3 Experiments

3.1 Dataset and Experimental Environment

We conduct experiments using the MovieLens-100K and MovieLens-1M datasets, which are commonly used benchmarks. MovieLens-100K contains 100,000 explicit ratings from 943 anonymous users on 1,682 movies (items). MovieLens-1M contains 1,000,000 explicit ratings from 6,040 anonymous users on 3,952 movies (items). Ratings range from 1 (dislike) to 5 (like), with each user rating at least 20 movies.

The MovieLens-100K dataset is partitioned into three subsets: the final test set (Test) contains 20,000 randomly selected ratings (20% of total data); the optimization set (Optim) contains another 20,000 random ratings (20%); and the training set (Train) comprises the remaining ratings (60%). Similarly, MovieLens-1M is partitioned using the same 3:1:1 ratio. The information core size is set to 20% of dataset users. The training set (Train) and optimization set (Optim) are used for information core extraction, while the test set (Test) evaluates the performance of the final selected core users.

The experimental environment runs on Windows 7 64-bit OS with an Intel(R) Core(TM) i3-CPU 550U @ 3.20 GHz processor, 4 GB RAM, and MATLAB 2017a.

3.2 Experimental Design and Results Analysis

To validate the performance of our proposed IFB and IRB algorithms for information core extraction, we implement FB-based, RB-based, and random selection methods for comparison. The random method simply selects users randomly from the user set as the information core. We compare IFB and IRB against these baselines using Mean Absolute Error (MAE) as the primary performance metric, where smaller MAE values indicate better performance.

For experimental robustness, we split the rating information in MovieLens-100K and MovieLens-1M five times to obtain five original datasets (u1-u5) with varying sparsity levels. Each original dataset contains training, optimization, and test sets in a 3:1:1 ratio, allowing us to compare algorithm performance under different data sparsity conditions.

[Figure 3: see original paper] and [Figure 4: see original paper] show MAE values obtained by Random, FB, RB, IFB, and IRB algorithms on the five initial datasets (u1-u5) of MovieLens-100K and MovieLens-1M, respectively, with neighbor counts of 10, 15, and 20. Across all bar charts in both figures, IFB and IRB consistently achieve lower MAE than the other three methods under equivalent conditions, demonstrating the effectiveness and feasibility of our improved approaches.

[Figure 5: see original paper] and [Figure 6: see original paper] show average precision and recall curves for Random, FB, RB, IFB, and IRB algorithms on MovieLens-100K and MovieLens-1M datasets, respectively, with recommendation list lengths varying from 10 to 20. The line charts in both figures demonstrate that our two improved IFB and IRB methods achieve higher precision and recall rates than other comparison algorithms, confirming their superiority.

Information core-based recommendation algorithms use only 20% of total users for recommendations, whereas traditional algorithms utilize all users. Since our improved information core extraction methods further enhance quality, our algorithm' s time complexity is approximately one-fifth that of traditional recommendation algorithms, substantially reducing time consumption.

compares online recommendation time between traditional CF and information core-based algorithms, showing average results from 10 independent runs on MovieLens-100K and MovieLens-1M. The results confirm that information core-based recommendation consumes significantly less time than traditional CF, demonstrating that identified information cores greatly reduce online recommendation time and lower time complexity, thereby alleviating scalability issues.

---

4 Conclusion

Recommendation technology represents an effective means to address big data challenges. However, as time progresses and data scales continue to grow, recommendation algorithms face increasingly higher performance demands, particularly regarding real-time capabilities. Information core-based recommendation algorithms have emerged as a promising direction in recent years, with a critical challenge being how to accurately identify the information core. Currently, understanding of information cores remains in the exploratory stage, and no definitive definition exists.

This paper analyzes existing information core extraction methods and proposes

improved approaches—IFB and IRB—enhancing FB and RB methods. Experimental results demonstrate that IFB and IRB can more accurately identify information cores. Information core-based recommendation algorithms can reduce time complexity while maintaining good recommendation quality, but what characteristics define suitable core users remains an open question. From the perspective of reducing recommendation time complexity, this represents a valuable research direction.

References

[1] Zhang Hui, You Fei. A novel fuzzy clustering recommendation algorithm based on PSO [J]. Cybernetics & Information Technologies, 2014, 14(5): 45-56.

[2] Son J, Kim S B. Content-based filtering for recommendation systems using multiattribute networks [J]. Expert Systems with Applications: An International Journal, 2017, 89(15): 404-412.

[3] Zhang Kaiwen, Muthusamy V, Sadoghi M, et al. Subscription covering for relevance-based filtering in content-based publish/subscribe Systems [C]//Proc of IEEE International Conference on Distributed Computing Systems. Atlanta, GA, USA: IEEE Press, 2017: 2039-2044.

[4] Thotharat N. Thai local product recommendation using ontological content based filtering [C]//Proc of International Conference on Knowledge and Smart Technology. Chonburi, Thailand: IEEE Press, 2017: 45-49.

[5] Resnick P, Iacovou N, Suchak M, et al. GroupLens: an open architecture for collaborative filtering of netnews [C]//Proc of ACM Conference on Computer Supported Cooperative Work. USA: ACM Press, 1994: 175-186.

[6] Maltz D, Ehrlich K. Pointing the way: active collaborative filtering [C]//Proc of Sigchi Conference on Human Factors in Computing Systems. USA: ACM Press/Addison-Wesley Publishing Co, 2016: 202-209.

[7] He R, Mcauley J. Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering [C]//Proc of International Conference on World Wide Web. Montreal, Canada: International World Wide Web Conferences Steering Committee, 2016: 507-517.

[8] Wang Haiming, Zhang Peng, Lu Tun, et al. Hybrid recommendation model based on incremental collaborative filtering and content-based algorithms [C]//Proc of International Conference on Computer Supported Cooperative Work in Design. Wellington, New Zealand: IEEE Press, 2017: 98-103.

[9] Chughtai M W, Selamat A, Ghani I, et al. Retracted: E-learning recommender systems based on goal-based hybrid filtering [J]. International Journal of Distributed Sensor Networks, 2014, 10(7): 252-260.

[10] Ghazanfar M A, Prugelbennett A. An improved switching hybrid recommender system using naive Bayes classifier and collaborative filtering [J]. Lecture Notes in Engineering & Computer Science, 2010, 1(2180): 45-52.

[11] Jannach D, Zanker M., Felferni A, et al. Recommender systems [M]. Jiang Fan, Translate. Beijing: People' s Post and Telecommunications Press, 2013: 8-90.

[12] Ricci F, Rokach L, Shapira B, et al. Recommender systems handbook [M]. New York: Springer, 2011: 73-182.

[13] Zhou Yanbo, Lei Ting, Zhou Tao. A robust ranking algorithm to spamming [J]. Europhysics Letters Association 2011, 94(4): 1034-1054.

[14] Wu Yitao, Zhang Xingming. User fuzzy similarity-based collaborative filtering recommendation algorithm [J]. Journal on Communication, 2016, 37(1): 198-206.

[15] Rong Huigui, Huo Shengxu, Hu Chunhua, et al. User similarity-based collaborative filtering recommendation algorithm [J]. Journal on Communication, 2014, 35(2): 16-24.

[16] Huang Lu, Lin Chuanjie, He Jun, et al. Diversified mobile App recommendation combining topic model and collaborative filtering [J]. Journal of Software, 2017, 28(3): 708-720.

[17] Zeng Wei, Zeng An, Liu Hao, et al. Uncovering the information core in recommender systems [J]. Scientific Reports, 2014, 4(6140): 1-8.

[18] Ekstrand M D, Riedl J T, Konstan J A. Collaborative filtering recommender systems [M]//The Adaptive Web. Berlin: Springer-Verlag, 2011: 291-324.

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