A Hybrid Recommendation Algorithm Based on User Interest Drift and Category Correlation: Postprint

Authors: Chen Hailong, Xie Sheng, Xue Yutong

Date: 2018-05-20T00:00:00+00:00

Abstract

Collaborative filtering algorithms currently represent the most prevalent personalized recommendation technique in recommendation systems. To address the limitations in similarity measurement methods of traditional algorithms, a hybrid recommendation algorithm is proposed that integrates user interest dynamics and category correlations. The algorithm categorizes items based on users' rating information, thereby uncovering users' preference levels for items across different categories. Simultaneously, a time-based interest weighting function is incorporated into item similarity computation to further enhance computational accuracy. Finally, the improved similarity computation method is integrated into the user clustering methodology; following user clustering, the resulting clusters substantially impact recommendation accuracy for users. Experimental results demonstrate that when executed on the Movielens-1k dataset, the algorithm exhibits improvements in both computational efficiency and accuracy.

Full Text

Hybrid Recommendation Algorithm Based on User Interest Change and Category Correlation

Chen Hailong, Xie Sheng, Xue Yutong
(School of Computer Science & Technology, Harbin University of Science & Technology, Harbin 150080, China)

Abstract

Collaborative filtering algorithms represent the most prevalent personalized recommendation technology in current recommender systems. To address the limitations of traditional similarity measurement methods, this paper proposes a

hybrid recommendation algorithm that integrates user interest changes and category correlation. The algorithm classifies items based on user rating information to uncover users' preference and attention levels toward different item categories. Simultaneously, a time-based interest weight function is introduced into item similarity calculations to further enhance computational accuracy. Finally, the improved similarity measurement method is incorporated into a user clustering approach, where the resulting clusters significantly impact recommendation accuracy. Experimental results on the Movielens-1k dataset demonstrate improvements in both operational efficiency and accuracy.

Keywords: collaborative filtering; clustering algorithm; class correlation; interest change; similarity

0 Introduction

With the continuous expansion of internet scale and coverage, online information data is growing at an explosive rate, overwhelming users with excessive information that makes it difficult to identify and obtain effective content, resulting in low information utilization and information overload. Recommender systems have emerged as a highly effective solution to this problem. Unlike search engines, recommender systems perform personalized calculations by studying user interest preferences, enabling the system to discover user interest points and guide users toward identifying their information needs. Recommendation algorithms based on user behavior data analysis are known as collaborative filtering algorithms, which operate on the fundamental principle that users with similar behaviors share similar preferences and needs. Consequently, collaborative filtering algorithms focus more on users' historical behaviors, are unaffected by new items, and deliver better recommendation accuracy.

To address issues such as low recommendation precision and dataset sparsity in traditional collaborative filtering algorithms, many scholars have proposed improved similarity algorithms and other approaches like clustering algorithms. For instance, Song Ruiping proposed the MSCF algorithm—a hybrid recommendation method based on user ratings and attributes combined with an improved k-nearest neighbor approach [1], which enhanced recommendation accuracy. While these methods improved algorithmic precision, they only considered user rating data when calculating user similarity, neglecting users' co-rated items (i.e., rating discrepancy) and overlooking item category preference and category attention issues. To resolve these problems, relevant researchers have introduced clustering techniques to optimize collaborative filtering algorithms, such as Yin Hang' s K-nearest neighbor collaborative filtering algorithm optimized with clustering [3].

User similarity measurement is closely related not only to the numerical values users assign to items but also to user interests. Literature [7] incorporated user trust and item attribute information, utilizing a time-based interest change strategy grounded in forgetting curves to generate neighbor set recommenda-

tions. In response to these challenges, this paper proposes a clustering-based collaborative filtering algorithm that incorporates user interest changes and category correlation. Since user interests may evolve over time, interest change curves are integrated into item similarity calculations. Category correlation—representing category preference or attention—reflects user preferences to a certain degree. Based on these two factors, this paper first clusters items, then performs improved similarity calculations based on the item clustering results, and subsequently clusters users according to these similarity measures. This approach significantly reduces the time complexity of clustering while making full use of available data information.

1 Traditional Collaborative Filtering Recommendation Algorithm

Collaborative filtering algorithms based on user behavior data analysis operate on the principle that users with similar behaviors share similar preferences and needs. Therefore, these algorithms focus on users' historical behaviors, are unaffected by new items, and provide better recommendation accuracy. Collaborative filtering-based algorithms primarily include user-based recommendation algorithms, item-based recommendation algorithms, model-based recommendation algorithms, and hybrid recommendation algorithms. Memory-based collaborative filtering algorithms encompass user-based and item-based methods, which consist of three main steps: (1) calculating similarity between users (or items) based on the user-item rating matrix; (2) selecting the top K most similar users (or items) as neighbors through inverse similarity sorting; and (3) predicting ratings for target users (or items) on unrated items based on neighbor ratings. The following sections detail the user-based collaborative filtering algorithm.

1.1 User-Item Rating Model

To generate recommendations for users, we focus on their rating data to mine their interests, preferences, and needs, ultimately recommending relevant products. The experimental dataset records the specific time when each user rated each movie, allowing us to fully utilize this data to discover recent preference changes. People' s preferences for different movie categories and specific movies change over time, and movies users have recently watched better predict their future interests. Inspired by forgetting curves and referencing the Ebbinghaus forgetting curve function, let $s(u,i)$ denote user u' s interest level in item i. Considering that users rate items at different times, let t_0 be the earliest rating time for user u, and $t(i)$ be the rating time for item i. Then $s(u,i)$ can be expressed as:

$$s(u,i) = e^{-\frac{t(i)-t_0}{T}}$$

If $t_0 = t_i$, then $s(u,i) = 1$.

Define a given user set U and item set S, where user ratings for items are represented as an $m \times n$ matrix R, as shown in Table . $R(i,j)$ represents user i' s rating for item j, reflecting user preferences for items. For example, in the MovieLens dataset, 1-5 points indicate user preference levels; if $R(i,j) = 0$, it means user i has not rated item j.

1.2 User Similarity Measurement Formulas

1) Cosine Similarity Based on the user-item rating matrix, we calculate user similarity. Similarity calculation methods include cosine similarity and Pearson correlation coefficient. Here we adopt the cosine similarity algorithm, treating user ratings as n-dimensional rating vectors where the k-th dimension value represents the rating for item k. Let user u and user v' s rating vectors be represented as vectors u and v, respectively. The similarity between user u and user v is:

$$\text{sim}(u,v) = \cos(u,v) = \frac{\sum_{s \in S} r_{u,s} \cdot r_{v,s}}{\sqrt{\sum_{s \in S} r_{u,s}^2} \cdot \sqrt{\sum_{s \in S} r_{v,s}^2}}$$

The similarity range is $[-1,1]$, where larger values indicate closer interests between users u and v.

2) Pearson Correlation Coefficient The cosine similarity measurement has certain accuracy issues as it fails to consider different users' rating scales, hence the Pearson correlation coefficient is introduced. Define the set of items co-rated by users u and v as $I(u,v)$, where $I(u)$ and $I(v)$ are the sets of items rated by users u and v, respectively. Their average ratings are $\bar{r}(u)$ and $\bar{r}(v)$, and user u' s rating for item s is $r(u,s)$. The similarity $\text{sim}(u,v)$ is:

$$\text{sim}(u,v) = \frac{\sum_{s \in I(u,v)} (r(u,s) - \bar{r}(u)) \cdot (r(v,s) - \bar{r}(v))}{\sqrt{\sum_{s \in I(u,v)} (r(u,s) - \bar{r}(u))^2} \cdot \sqrt{\sum_{s \in I(u,v)} (r(v,s) - \bar{r}(v))^2}}$$

The similarity range is $[-1,1]$, where larger values indicate closer interests. This paper uses the Pearson correlation coefficient to calculate similarity.

After calculating similarity, select the top k users with the highest similarity to user u as neighbors $G(U)$. Based on these k users' ratings for target item j, we predict user u' s rating for item j using the following formula, where $\bar{R}_u$ represents user u' s average rating and R_{uj} represents user u' s rating for item j:

$$P_{u,j} = \bar{R}_u + \frac{\sum_{n \in G(u)} \text{sim}(u,n) \cdot (R_{n,j} - \bar{R}_n)}{\sum_{n \in G(u)} |\text{sim}(u,n)|}$$

2 Time-Based Interest Weight

Each user' s interest changes at different speeds and patterns, and user interests exhibit various fluctuations and changes. Therefore, early access data should also be valued and fully utilized. The following function measures past user interest similarity $I(u,i)$. Let $I(u)$ be the set of items accessed by user u, and define a time period T. Let $I(u,t)$ be the set of items accessed by user u in the recent time period T, indicating the user' s recent interests. For an item i, if many items in u' s recent access set are highly similar to i, it indicates that item i is strongly associated with the user' s current interests, and the user' s future interests may remain similar to item i. Therefore, item i plays a key role in predicting user interests.

Calculate $I(u,i)$ through the overall similarity between i and items in $I(u,t)$:

$$I(u,i) = \frac{\sum_{j \in I(u,t)} \mathrm{sim}(i,j)}{|I(u,t)|}$$

where $|I(u,t)|$ is the number of resources accessed by user u in the recent time period T.

From the above analysis, it is necessary to combine recent user interest changes with long-term interest data for analysis. Users with frequently changing interests pay more attention to recent interests, where the weight of recently liked items should be greater than that of long-term items. The long-term interest measurement function operates on historical data to make full use of information, avoid missing key characteristics of early data, and truly capture the recurring nature of user interests. Finally, combine the recent interest measurement function with the long-term interest measurement function:

$$\mathrm{fu}(u,i) = \alpha \times s(u,i) + (1-\alpha) \times I(u,i)$$

where α is a balancing factor with a range of $[0,1]$.

3 Similarity-Improved User Clustering Collaborative Filtering Algorithm

This paper employs the K-means clustering method for user and item clustering. The traditional clustering algorithm steps are:

Input: K user categories.

a) Randomly select K users from user set U as the initial cluster centers for the K clusters.

b) Calculate the similarity between each user $U(i)$ and the K cluster centers, assigning $U(i)$ to the category with maximum similarity.

c) Recalculate the centers of the K clusters by computing the arithmetic mean of all users' ratings for items as the new cluster center.

d) For all users in U, repeat steps b) and c) iteratively until the clustering results remain unchanged, at which point iteration ends; otherwise, continue iterating.

After completing item clustering (based on categories defined in the dataset), items are grouped into K clusters, where items in each cluster share similar properties. Since each user has different preferences, after item clustering we can obtain users' preference levels for categories. When users enter a website to select movies without knowing specific content but only the category, they exhibit certain biases. Even after watching a movie they may not like it, so we define user i' s category preference for movie category j as $\text{fav}(i, j)$:

$$\text{fav}(i, j) = \frac{\sum_{c \in I_j} r_{i,c}}{|I_j|}$$

where I_j is the set of items in category j.

Based on the above formula, we can calculate the clustering similarity $\text{simc}(I, j)$ between users I and J:

$$\text{simc}(i, j) = \frac{\sum_{k=1}^{K} \text{fav}(i, k) \cdot \text{fav}(j, k)}{\sqrt{\sum_{k=1}^{K} \text{fav}(i, k)^2} \cdot \sqrt{\sum_{k=1}^{K} \text{fav}(j, k)^2}}$$

The improved similarity calculation method is:

$$\text{simc}_{\text{improved}}(i, j) = \beta \times \text{fu}(i, j) + (1 - \beta) \times \text{simc}(i, j)$$

where β is a balancing factor with a range of $[0, 1]$.

3.1 Similarity-Improved User Clustering Collaborative Filtering Algorithm

After incorporating all the above elements into user similarity calculation, we cluster users with the goal of grouping highly similar users together. This prevents recommendation bias from using data from users with completely different interests during recommendation while also reducing unnecessary computations and resource waste. This paper uses the K-means clustering method for user clustering.

Input: K user categories.

a) Randomly select K users from user set U as the initial cluster centers.

b) Calculate the similarity between each user $U(i)$ and the K cluster centers, assigning $U(i)$ to the category with maximum similarity.

c) Recalculate the centers of the K clusters by computing the arithmetic mean of all users' ratings for items as the new cluster center.

d) For all users in U, repeat steps b) and c) iteratively until the clustering results remain unchanged, at which point iteration ends; otherwise, continue iterating.

After user clustering is completed, when generating recommendations for a specific user, we only need to identify the top-N most similar users within his/her cluster and predict ratings using the rating prediction formula, finally recommending the top-K items with highest predicted scores.

4 Experimental Results and Analysis

4.1 Experimental Dataset

This experiment uses the MovieLens-1m dataset, which contains 100,000 ratings from 6,640 users on 4,000 items. The dataset is split with 80% as the training set and 20% as the test set. Mean Absolute Error (MAE) is adopted as the metric to evaluate recommendation algorithm performance. Let the predicted rating set be $\{p_1, p_2, \ldots, p_C\}$ and the corresponding actual rating set be $\{r_1, r_2, \ldots, r_C\}$. MAE is defined as:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{C} |p_i - r_i|}{C}$$

4.2.1 Selection of α

The improved similarity calculation consists of two parts: recent interest changes and long-term interest changes. The balancing factor α is used to balance the proportion between these two components, with a range of $[0, 1]$, incrementing by 0.1 each time to compare MAE variations. As shown in Figure [Figure 1: see original paper], when $\alpha = 0.2$, MAE reaches its minimum, indicating optimal recommendation performance. This demonstrates that users' recent preferences for movie categories significantly influence their future movie selections. In reality, preferences for movie categories are stage-based, and interests typically remain relatively stable within a certain period.

4.3 Selection of β

The clustering information component in the improved similarity consists of two parts: category correlation and user interest temporal changes. The balancing

factor β balances these two components, with a range of $[0, 1]$, incrementing by 0.1 each time while fixing $\alpha = 0.1$ to compare MAE variations. As shown in Figure [Figure 2: see original paper], when $\beta = 0.5$, MAE reaches its minimum, indicating optimal recommendation performance. The β value reflects the relative importance of category correlation and user interest change factors. A β value of 0.5 indicates no obvious bias between the two factors. Therefore, for the final similarity calculation formula, we select $\alpha = 0.2$ and $\beta = 0.5$.

4.4 Performance Comparison of Improved Algorithm

We compare the similarity-improved user clustering collaborative filtering algorithm with traditional collaborative filtering (user-based and item-based) and simple user clustering algorithms. Simple user clustering performs clustering directly, while similarity-improved user clustering uses $\alpha = 0.1$ and $\beta = 0.4$ based on experimental results. The results are shown in Figure [Figure 3: see original paper].

The experimental results demonstrate that the improved clustering algorithm achieves smaller MAE than traditional collaborative filtering algorithms and traditional clustering algorithms, with MAE reaching its minimum when the nearest neighbor size increases to 25. The experimental data proves that users' recent interests indeed affect the results and reduce errors, and the clustering algorithm implementation outperforms traditional collaborative filtering algorithms.

5 Conclusion

Recommender systems help users solve information overload problems and have been widely applied across various domains. Currently common recommendation methods include collaborative filtering, content-based recommendation, matrix-based recommendation, and hybrid recommendation. This paper primarily uses user ratings for movie items and rating timestamps to perform item clustering, discover user category preference and attention levels, and simultaneously uncover users' recent interests to improve inter-user similarity calculation rules. By reviewing relevant literature, we incorporate both user recent preferences and historical interest factors. Final experimental results demonstrate that the improved clustering algorithm indeed yields smaller errors. Recommendation algorithms continue to evolve and progress, with data sparsity, overfitting, scalability, and multimedia information feature extraction remaining primary challenges. Existing technologies and methods cannot fundamentally solve these problems. As application domains continue to expand, recommender systems will face new demands and issues. The development of recommender systems is inseparable from these problems and challenges, and research on recommendation methods addressing these issues remains a hot topic in intelligent information processing fields such as information retrieval, data mining, and machine learning.

References

[1] Song Ruiping. Research on Hybrid Recommendation Algorithms [D]. Lanzhou: Lanzhou University, 2014.
[2] Wang Xiaojun. Improving Collaborative Filtering Recommendation Using Item Attributes and Preferences [J]. Journal of Beijing University of Posts and Telecommunications, 2014, 37(6): 68-71.
[3] Yin Hang, Chang Guiran, Wang Xingwei. K-Nearest Neighbor Collaborative Filtering Algorithm Optimized by Clustering [J]. Small Microcomputer Systems, 2013, 34(4): 806-809.
[4] Wang Mingjia, Han Jingti, Han Songqiao. Collaborative Filtering Algorithm Based on Fuzzy Clustering [J]. Small Microcomputer Systems, 2012, 38(24): 50-52.
[5] Sun Hui, Ma Yue, Yang Haibo, et al. A Similarity-Improved User Clustering Collaborative Filtering Recommendation Algorithm [J]. Small Microcomputer Systems, 2014, 35(9): 1967-1970.
[6] Xing Chunxiao, Gao Fengrong, Zhan Sinan, et al. Collaborative Filtering Recommendation Algorithm Adapting to User Interest Changes [J]. Journal of Computer Research and Development, 2007, 44(2): 296-301.
[7] Chen Zhimin, Li Zhiqiang. Collaborative Filtering Recommendation Algorithm Based on User Characteristics and Item Attributes [J]. Computer Applications, 2011, 31(7): 1748-1751.
[8] Zhang J, Lin Y, Lin M, et al. An effective collaborative filtering algorithm based on user preference clustering [J]. Applied Intelligence, 2016, 45(2): 1-13.
[9] Li S, Yuan X, Han H. A kind of collaborative filtering algorithm based on user clustering and time stamp [C]// Proc of International Symposium on Advances in Electrical, Electronics and Computer Engineering. 2016.
[10] Chen Q, Li W, Liu J. Collaborative Filtering Algorithm Based on Item Attribute and Time Weight [C]// Proc of International Conference on Automatic Control and Information Engineering. 2016.
[11] Yao Z, Zhang Q. Item-based clustering collaborative filtering algorithm under high-dimensional sparse data [C]// Proc of International Joint Conference on Computational Sciences and Optimization. 2009: 787-790.
[12] Yu Y, Zhu S, Chen X. Collaborative filtering algorithms based on Kendall correlation in recommender systems [J]. Wuhan University Journal: Natural Science English Edition, 2015, 20(3): 204-210.
[13] Li J, Liu X. An Important Aspect of Big Data: Data Usability [J]. Journal of Computer Research & Development, 2013, 50(6): 1147-1162.

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