Collaborative Filtering Recommendation Algorithm Based on Item Probability Distribution (Postprint)

Authors: Wang Yong, Deng Jiangzhou, Deng Yongheng, Zhang Pu

Date: 2017-10-11T00:00:00+00:00

Abstract

Objective: To address the limitations of traditional item similarity measurement methods that must rely on co-rated items and suffer from low prediction accuracy in sparse datasets. Method: KL divergence from the signal processing domain is introduced into item similarity computation. By utilizing the probability density distribution of rating values to calculate item similarity, similar neighbor items for target items can be more effectively identified. Results: Experimental results on the MovieLens dataset demonstrate that the algorithm' s comprehensive recommendation metric F1 exceeds 0.65, and its performance in terms of prediction effectiveness, prediction error, and recommendation accuracy significantly outperforms currently prevalent item similarity methods. Limitations: The method only considers the ratio of item rating values and does not fully exploit the absolute rating values of items. Conclusion: The algorithm effectively leverages rating information within the dataset, satisfactorily overcomes the data sparsity problem, and exhibits considerable application value.

Full Text

A Collaborative Filtering Recommendation Algorithm Based on Item Probability Distribution

Wang Yong[1], Deng Jiangzhou[1], Deng Yongheng[1], Zhang Pu[2]
[1](Key Laboratory of Electronic Commerce and Modern Logistics, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)
[2](College of Computer Science, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract

[**Objective**] This study addresses the limitations of traditional item similarity measurement methods that must rely on co-rated items, and improves prediction accuracy in sparse datasets. [**Methods**] We introduce the Kullback-Leibler (KL) divergence from signal processing into item similarity calculation, leveraging the probability density distribution of rating values to compute item similarity, which enables more effective discovery of similar neighbor items for target items. [**Results**] Experimental results on the MovieLens dataset demonstrate that the proposed algorithm achieves an F1 measure exceeding 0.65, with evaluation results in prediction effectiveness, prediction error, and recommendation accuracy all significantly outperforming commonly used item similarity methods. [**Limitations**] The method only considers the ratio of item rating values and does not fully utilize absolute rating values. [**Conclusions**] The algorithm effectively leverages rating information within the dataset, better overcoming data sparsity issues, and demonstrates strong practical application value.

Keywords: Item similarity; Collaborative filtering; KL divergence; Recommendation algorithm
Classification Number: TP391; G350

---

1. Introduction

With the popularization and deep application of the Internet and mobile Internet, information volume has surged dramatically. Addressing information overload and meeting users' personalized needs has become a current research hotspot. Collaborative filtering-based recommendation algorithms represent a highly effective approach in this domain and have garnered increasing attention. Goldberg et al. [**?**] first proposed the concept of collaborative filtering in 1992, and currently, collaborative filtering recommendation systems have been widely applied in social networks, e-commerce, and other fields [**?**]. Collaborative filtering recommendation algorithms primarily identify users similar to the current user within a large user community, and use these similar users' preferences as a basis to recommend products or services to the current user.

Currently, mainstream collaborative filtering recommendation algorithms are divided into two categories: user-based collaborative filtering [**?**] and item-based collaborative filtering [**?**]. In recommendation systems, the employed user (or item) similarity measurement method directly affects recommendation quality. Traditional user-based similarity measurement methods [**?**], such as cosine similarity and Pearson correlation coefficient, have achieved great success. However, with changes and deepening of application environments, sparsity and cold-start problems have become increasingly prominent. To address these issues, several new similarity methods have been proposed. Luo et al. [**?**] combined two similarity calculation methods to solve sparse dataset problems, proposing local user similarity and global user similarity based on surprise vectors. Ahn [**?**] proposed

a heuristic similarity calculation method called PIP. Although the PIP method alleviates the cold-start problem to some extent, it produces inaccurate results in sparse datasets due to few co-rated items between users. Bobadilla et al. [**?**] proposed the JMSD method, which combines Jaccard [**?**] and MSD [**?**]. This method compensates for Jaccard' s failure to consider absolute rating values and MSD' s neglect of the proportion of co-rated items. Arwar et al. [**?**] proposed a series of item-based collaborative filtering recommendation algorithms that achieved considerable success in practice. However, when users have few co-rated items, all the above methods suffer from low recommendation quality —that is, the sparsity problem. To fully utilize each item' s ratings, Patra et al. [**?**] proposed a similarity measurement method based on the Bhattacharyya coefficient. This method calculates item similarity from the perspective of probability density distribution, compensating for traditional similarity measurement methods' reliance on co-rated items and positively contributing to solving the sparsity problem.

Building upon the idea from literature [**?**] of calculating item similarity from a probability density distribution perspective, this paper introduces the KL divergence from information theory into similarity calculation and proposes a collaborative filtering recommendation algorithm based on item probability distribution. Using KL divergence to calculate similarity between different items effectively avoids the limitation of existing methods that must rely on co-rated items in the dataset. We use similarity to predict users' ratings for unrated items and generate recommendation datasets based on predicted values. This method can effectively address the sparsity problem in collaborative filtering.

The advantage of KL divergence lies in its ability to distinguish objects that geometric distance cannot. As shown in [Figure 1: see original paper], suppose Object 1 and Object 2 follow a normal distribution and uniform distribution respectively, with substantial overlap between the two objects' sample points. Clearly, geometric distance struggles to distinguish the two objects. However, from a probability distribution perspective, KL divergence can efficiently differentiate them.

3. Methodology

3.1 KL Divergence for Item Similarity (1) Smoothing Processing

To ensure KL divergence can be applied to user rating matrices—that is, to guarantee that the probability density function $\rho(x)$ is always greater than 0—we perform smoothing correction as follows:

$$\hat{\rho}(x) = \rho(x) + \delta$$

where $0 < \delta < 1$, and D represents the number of all possible values in the discrete domain.

After smoothing processing, error analysis is as follows:

$$\hat{D}(x) = \hat{\rho}(x) - \rho(x) = \delta$$

When the δ value is sufficiently small, smoothing processing can provide arbitrary precision.

(2) Symmetry Correction

As shown in formula (7), KL distance is not symmetric, i.e., $D(\rho_i||\rho_j) \neq D(\rho_j||\rho_i)$. When representing the distance between two items, symmetry is required. Therefore, we perform symmetry correction on KL distance as follows:

$$D_s(i,j) = \frac{D(i,j) + D(j,i)}{2}$$

(3) Similarity Calculation

KL Similarity

In the user rating matrix, for any two items i and j, we treat all users' ratings for them as two variable sequences. The KL distance $D(i,j)$ between items i and j is calculated as follows:

$$D(i,j) = \sum_{v=1}^{r} \rho_i(v) \log \frac{\rho_i(v)}{\rho_j(v)}$$

where ρ_i is the probability density function of item i, r is the maximum rating value, $\rho_i(v) = \frac{\#v}{\#i}$ is the ratio of rating value v in item i, $\#i$ is the total number of ratings for item i, and $\#v$ is the number of ratings with value v for item i.

Based on KL distance, we provide the KL-based similarity calculation formula as follows:

$$\text{sim}_{\text{KL}}(i,j) = \frac{1}{1 + D(i,j)}$$

where smaller KL distance indicates higher similarity between items.

The KL-based similarity calculation method does not rely on co-rated items and applies to scenarios where traditional similarity methods cannot be used. We illustrate the advantage of our method with an example. Suppose the ratings for items i and j are: $i = (1,0,2,0,3,0)^T$ and $j = (0,3,0,2,0,1)^T$, with a rating range of 1-3. Since no user rated both items simultaneously, existing methods (such as cosine similarity) cannot calculate similarity between the two items. However, using formula (8), we can obtain the KL similarity between items i and j as follows:

When calculating KL similarity between items, we replace $D(i,j)$ in formula (8) with $D_s(i,j)$.

3.2 Generating Recommendations (1) Forming Nearest Neighbor Set

Based on formula (8), we calculate similarity values between any items to obtain the item similarity matrix $[S_{i,j}]_{n\times n}$ as follows:

$$S = \begin{bmatrix} S_{1,1} & S_{1,2} & \cdots & S_{1,n} \\ S_{2,1} & S_{2,2} & \cdots & S_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ S_{n,1} & S_{n,2} & \cdots & S_{n,n} \end{bmatrix}$$

where $S_{i,j}$ $(1 \le i \le n,\ 1 \le j \le n)$ represents the similarity between items i and j.

Based on the item similarity matrix S, we can obtain the set of N nearest neighbor items for item i, denoted as $N_i = \{i_1, i_2, \ldots, i_N\}$, with elements sorted internally by similarity.

(2) Prediction Calculation

Using ratings from items in set N_i, we calculate the predicted rating $P_{u,i}$ for target user u on item i as follows:

$$P_{u,i} = \frac{\sum_{j\in N_i} \mathrm{sim}_{\mathrm{KL}}(i,j)\cdot r_{u,j}}{\sum_{j\in N_i} \mathrm{sim}_{\mathrm{KL}}(i,j)}$$

where $r_{u,j}$ is user u' s rating for item j.

(3) Top-N Recommendation

Based on predicted values, we can perform Top-N recommendation by selecting the top N items with highest predicted values as the user' s recommendation set.

4. Algorithm Analysis

(1) Sparsity Analysis

Traditional similarity calculation methods such as ACOS, PCC, and CPC have the limitation of "must rely on co-rated items" —that is, at least one user must rate both items i and j. Once there are no co-rated items, traditional methods cannot calculate similarity between these two items. This situation is particularly prominent in sparse datasets. Our proposed method calculates similarity using the probability distribution of rating values for items i and j, without relying on co-rated items and without requirements on the number of users who rated the items. Therefore, even in sparse datasets, our method can obtain necessary information to complete item similarity calculation. Consequently, the proposed method better addresses the data sparsity problem commonly present in recommendation algorithms.

(2) Applicability Analysis
Our proposed similarity method is based on the probability density of rating values and calculates item similarity through KL divergence. This method makes no assumptions about data distribution in the dataset. However, some traditional similarity calculation methods typically assume linear relationships between variables, limiting their applicability. For user rating datasets, the data is discrete and often lacks linear relationships. Predictions based on linear assumptions inevitably yield poor results. Our method has no requirements regarding whether linear relationships exist between data, thus offering better adaptability for both linear and non-linear data relationship problems.

(3) Information Utilization Analysis
Our method is not constrained by co-rated items. When calculating rating probability density, it uses all user rating information in the rating matrix. Therefore, the proposed algorithm has higher information utilization than other similarity calculation methods. High information utilization can prevent one-sidedness in prediction results and avoid large fluctuations, thereby improving the overall performance of our algorithm.

5. Experiments

5.1 Dataset We adopt the public MovieLens dataset[1] as our test and validation dataset, which includes 706 users' ratings for 8,570 movies with 100,023 rating records total. From this dataset, we selected 59,775 ratings as our experimental dataset, containing 706 users and 813 movies, with rating range 1-5 and each movie rated at least 25 times. The sparsity of the experimental dataset is 10.4%. To test recommendation performance, we divided the dataset into 80% training set and 20% test set.

[1]http://www.grouplens.org

5.2 Evaluation Metrics Recommendation algorithm evaluation primarily includes three aspects: prediction accuracy, recommendation accuracy, and computational effectiveness [**?**]. Commonly used prediction accuracy metrics are Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), with formulas as follows:

$$\mathrm{MAE} = \frac{\sum_{u,i} |r_{ui} - \hat{r}_{ui}|}{n}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{u,i} (r_{ui} - \hat{r}_{ui})^2}{n}}$$

where r_{ui} and $\hat{r}_{ui}$ are the actual and predicted rating values of user u for item i, respectively, and n represents the number of items to be predicted. Smaller values for these metrics indicate higher prediction accuracy.

Commonly used recommendation accuracy metrics are Precision, Recall, and F1 measure, with corresponding formulas:

$$\text{Precision} = \frac{n(I_a \cap I_p)}{n(I_p)}$$

$$\text{Recall} = \frac{n(I_a \cap I_p)}{n(I_a)}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where I_p is the number of predicted recommended items, and I_a is the number of actual recommended items. The F1 measure is a comprehensive evaluation metric combining precision and recall; higher values indicate better overall recommendation performance. In our experiments, we use ratings above the user' s average rating as the criterion for item recommendation to determine the recommended item list.

Additionally, computational effectiveness evaluation metrics are effective prediction count and perfect prediction count. Effective prediction count refers to the total number of successfully calculated prediction values from the user rating dataset according to the prediction formula. Perfect prediction count refers to the number of calculated prediction values that exactly match the actual rating values.

5.3 Results Analysis To compare with our algorithm, we conducted comparative tests with ACOS, PCC, and CPC methods. Additionally, since different neighbor counts K affect test results, we also consider this factor in our experiments.

(1) Effective Prediction Count and Perfect Prediction Count Analysis
On the experimental dataset, the total number of predicted items is 12,017. As shown in [Figure 2: see original paper], regardless of how the neighbor count K varies during calculation, our algorithm achieves the highest effective prediction count and perfect prediction count. This indicates that our algorithm has better adaptability than ACOS, PCC, and CPC methods, can calculate effective prediction values under more data conditions, and achieves higher accuracy.

(2) MAE and RMSE Analysis
MAE and RMSE primarily reflect the deviation between predicted rating values and actual rating values. As shown in [Figure 3: see original paper], our algorithm' s MAE and RMSE outperform traditional similarity methods, with both error values overall lower than other similarity methods. As K increases, MAE and RMSE both decrease slowly, with overall ranges of $0.739 \leq \text{MAE} \leq 0.779$ and $0.974 \leq \text{RMSE} \leq 1.049$, indicating good recommendation accuracy for our algorithm.

(3) Precision, Recall, and F1 Analysis
In Figure 4: see original paper, the PCC method achieves the highest precision, followed by our method, with little difference between them. In Figure 4: see original paper, regardless of how K varies, the KL similarity method' s recall is significantly better than other methods. The F1 measure comprehensively considers precision and recall. As shown in Figure 4: see original paper, our algorithm' s F1 measure is significantly better than other methods. Comprehensive analysis demonstrates that our algorithm has better recommendation performance.

6. Conclusion

This paper introduces KL divergence from information theory into similarity calculation for collaborative filtering algorithms and proposes a collaborative filtering recommendation algorithm based on KL similarity. This method calculates similarity between items using the probability density distribution of rating values. Its advantage lies in having no requirements on the number of items rated by users or on users rating multiple items simultaneously. Relaxing these constraints means our method can find more rating data that satisfies its calculation conditions, enabling effective prediction value calculation and item recommendation even in sparse datasets. Therefore, compared with traditional similarity calculation methods, our method better solves the data sparsity problem. Experiments on the public MovieLens dataset demonstrate that our KL similarity-based collaborative filtering algorithm outperforms other similar methods and effectively improves overall recommendation quality.

References

[1] Goldberg D, Nichols D, Oki B M, et al. Using Collaborative Filtering to Weave an Information Tapestry [J]. Communications of the ACM, 1992, 35(12): 61-70.
[2] Zheng N, Li Q, Liao S, et al. Which Photo Groups Should I Choose a Comparative Study of Recommendation Algorithms in Flickr [J]. Journal of Information Science, 2010, 36(6):
[3] Brynjolfsson E, Hu Y J, Smith M D. Consumer Surplus in the Digital Economy: Estimating the Value of Increased Product Variety at Online Booksellers [J]. Management Science, 2003, 49(11): 1580-1596.
[4] Breese J, Hecherman D, Kadie C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering [C]. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, 1998.
[5] Xu C, Xu J, Du X. Recommendation Algorithm Combining the User-based Classified Regression and the Item-based Filtering [C]. In: Proceedings of the International Conference on Electronic Commerce: The New E-commerce-Innovations for Conquering Current Barriers, Obstacles and Limitations to Conducting Successful Business on the Internet, Fredericton, New Brunswick, Canada. 2006: 574-578.

[6] Arwar B, Karypls G, Konstan J, et al. Item-based Collaborative Filtering Recommendation Algorithms [C]. In: Proceedings of the 10th International World Wide Web Conference. 2001.
[7] Kim B M, Li Q, Park C S, et al. A New Approach for Combining Content-based and Collaborative Filters [J]. Journal of Intelligent Information System, 2006, 27(1): 79-91.
[8] Karypis G. Evaluation of Item-based Top-N Recommendation Algorithms[C]. In: Proceedings of the 10th International Conference on Information and Knowledge Management.
[9] Deng A, Zhu Y, Shi B. A Collaborative Filtering Recommendation Algorithm Based on Item Rating Prediction [J]. Journal of Software, 2003, 14(9): 1621-1628.
[10] Luo H, Niu C, Shen R, et al. A Collaborative Filtering Framework Based on both Local User Similarity and Global User Similarity [J]. Machine Learning, 2008,72(3): 231-245.
[11] Ahn H J. A New Similarity Measure for Collaborative Filtering to Alleviate the New User Cold-Starting Problem [J]. Information Sciences, 2008, 178 (1): 37-51.
[12] Bobadilla J, Ortega F, Hernando A, et al. A Collaborative Filtering Approach to Mitigate the New User Cold Start Problem [J]. Knowledge-Based Systems, 2012, 26: 225-238.
[13] Koutrica G, Bercovitz B, Garcia H. FlexRecs: Expressing and Combining Flexible Recommendations [C]. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. 2009.
[14] Cacheda F, Carneiro V, Fernández D, et al. Comparison of Collaborative Filtering Algorithms: Limitations of Current Techniques and Proposals for Scalable, High-Performance Recommender System [J]. ACM Transactions on the Web, 2011, 5(1): 1-33.
[15] Patra B K, Launonen R, Ollikainen V, et al. Exploiting Bhattacharyya Similarity Measure to Diminish User Cold-start Problem in Sparse Data [A]. // Discovery Science [M]. Springer International Publishing, 2014: 252-263.
[16] Kullback S, Leibler R A. On Information and Sufficiency [J]. The Annals of Mathematical Statistics, 1951, 22(1): 79-86.
[17] Huang A. Similarity Measures for Text Document Clustering [C]. In: Proceedings of the 6th New Zealand Computer Science Research Student Conference. 2008.

Author Contributions Statement

Wang Yong: Determined research objectives and technical route, proposed paper revision suggestions, and modified the paper.
Deng Jiangzhou: Designed algorithms, drafted and modified the paper.
Deng Yongheng: Collected data and performed experimental analysis.
Zhang Pu: Assisted with algorithm design and algorithm performance improvement.

Conflict of Interest Statement

All authors declare no conflict of interest.

Supporting Data

Supporting data is available in the journal's online version at http://www.infotech.ac.cn:
[1] Wang Yong, Deng Jiangzhou. ACOSsim.xlsx. Adjusted cosine similarity calculation results.
[2] Wang Yong, Deng Jiangzhou. PCCsim.xlsx. Pearson correlation coefficient calculation results.
[3] Wang Yong, Deng Jiangzhou. CPCsim.xlsx. Constrained Pearson correlation coefficient calculation results.
[4] Wang Yong, Deng Jiangzhou. KLsim.xlsx. Item similarity calculation results based on KL divergence.
[5] Wang Yong, Deng Jiangzhou. MAE_{RMSE}.xlsx. Related error calculation results.
[6] Wang Yong, Deng Jiangzhou. F1.xlsx. F1 measure calculation results.
[7] Wang Yong, Deng Jiangzhou. Effective prediction count and perfect prediction count.xlsx. Effective and perfect prediction count calculation results.

Received Date: 2016-01-26
Revised Date: 2016-03-23

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