Postprint: Dynamic Correlation Weighting for Collaborative Filtering Recommendation

Authors: Wang Jian, Yu Qingsong

Date: 2018-08-13T00:00:00+00:00

Abstract

When employing traditional collaborative filtering (CF) algorithms for recommendation, the high sparsity of the user-item rating matrix renders directly computed similarities between users or items relatively unreliable. To address this issue, we propose a method that, building upon traditional collaborative filtering, introduces inter-item correlations. By constructing dynamic weighting factors between item category tags and bipartite graph approaches to fuse these two types of correlations, asymmetric correlation relationships are formed, enabling rating predictions for users on items and thereby mitigating the problem of excessive rating matrix sparsity. Experimental results demonstrate that compared to traditional methods employing symmetric similarity relationships and fixed-weight approaches, the proposed method of fusing correlations through dynamic weights to form asymmetric relationships better aligns with real-world scenarios and achieves superior recommendation performance.

Full Text

Preamble

Authors: Wang Jian, Yu Qingsong
Affiliation: College of Computer Science & Software Engineering, East China Normal University, Shanghai 200333, China

Abstract: Traditional collaborative filtering (CF) algorithms suffer from low credibility in similarity calculations between users or items due to the extreme sparsity of user rating matrices. To address this limitation, this paper introduces inter-item relevance into conventional collaborative filtering. By constructing dynamic weighting factors between item category labels and bipartite graph methods, we fuse these two types of associations to form a non-equivalent relevance relationship, enabling more accurate user rating predictions for items.

Experimental results demonstrate that compared to conventional methods using equivalent similarity relationships and fixed weights, our approach of forming non-equivalent relationships through dynamic weight fusion better reflects real-world scenarios and yields superior recommendation performance.

Keywords: collaborative filtering; rating matrix; sparsity; dynamic weighting factor; non-equivalent relationship

1 Related Work

Rapid advances in information technology have generated massive amounts of data on the Internet, creating significant challenges for users seeking relevant information. Recommendation systems have emerged as a solution, filtering existing information to provide personalized suggestions without requiring explicit search queries. The history of recommendation systems can be traced back to 1979 in connection with cognitive science research [1], and they are now widely applied across various domains, particularly in e-commerce where their importance is paramount.

Numerous recommendation approaches have been developed, including content-based methods that leverage user demographics and item attributes [2], collaborative filtering techniques that rely solely on user behavior data [3,4], tag-based approaches utilizing user-generated labels [5], and context-aware systems incorporating temporal and spatial information [6]. To improve recommendation performance, various enhanced models have been proposed. For instance, Zhou et al. [7] introduced a bipartite graph recommendation algorithm using a probabilistic propagation mechanism that constructs unidirectional, non-equivalent weight relationships between similar items, achieving significant improvements in accuracy and recall. Nilashi et al. [8] employed Singular Value Decomposition (SVD) to extract the most influential latent features for dimensionality reduction of sparse matrices, combined with ontology techniques to enhance recommendation accuracy. Fan et al. [9] utilized the KNN algorithm to identify user neighbors and predict missing values in sparse matrices based on neighbor ratings to improve collaborative filtering performance.

While these methods primarily model user behavior data with reasonable effectiveness, they possess certain limitations. Therefore, this paper proposes a method that dynamically fuses category and bipartite graph concepts (RD-WCF), integrating both content and behavioral data through dynamic weighting to achieve superior recommendation results.

2 Methodology

2.1 Collaborative Filtering

Collaborative filtering is a widely adopted and successful recommendation technique. Its fundamental principle involves recommending items to users with similar preferences based on the rating patterns of their nearest neighbors, either at the user or item level. The relationship between users and items is typically represented using a rating matrix, as illustrated in , where U_m denotes the m-th user, I_n denotes the n-th item, and R_{mn} represents the rating given by user U_m to item I_n, with "?" indicating unknown relationships requiring prediction.

Despite its effectiveness, collaborative filtering faces several challenges in practical applications [12]. As the scale of recommendation system deployments expands, both user and item populations grow exponentially, degrading the performance of traditional collaborative filtering algorithms. The primary issue is data sparsity—while systems may contain millions of users and items, each user typically interacts with only a limited number of items, resulting in user-item rating matrices that are approximately 1-2% populated. This sparsity inevitably compromises the accuracy of similarity calculations between users or items. For example, two users with few ratings may share high similarity based on limited overlapping items that do not truly reflect their interests. Additionally, traditional collaborative filtering suffers from cold-start problems, struggling to generate recommendations for new users and items due to insufficient data.

2.2 Relevance

Similarity is typically symmetric—if the similarity between A and B is S, then the similarity between B and A is also S. This paper introduces the concept of *relevance* to capture asymmetric relationships. Empirical evidence suggests that relationships are inherently non-equivalent; measuring from different perspectives yields different relationship weights. Therefore, incorporating non-equivalent relevance between items provides a more realistic representation.

2.2.1 Category Tag-Based Relevance Items often possess category attributes—movies may belong to genres such as comedy, thriller, or romance, while books have analogous category labels. The item-category matrix is shown in , where I_n represents the n-th item and G_k represents the k-th category, with values of 1 indicating item-category membership and 0 indicating non-membership. The Jaccard formula can be applied to establish relevance between two items I_i and I_j:

$$rel_g(I_i, I_j) = \frac{|S(I_i) \cap S(I_j)|}{|S(I_i) \cup S(I_j)|}$$

where $S(I_i)$ denotes the set of category labels for item I_i. This relationship, derived from intrinsic item attributes, constitutes an equivalent (symmetric) relationship.

2.2.2 Bipartite Graph-Based Relevance In recommendation systems, relationships exist between users, between items, and between users and items, naturally lending themselves to graph representations. By treating users and items as nodes and their relationships as edges weighted by ratings, we can construct a bipartite graph to model these connections, as illustrated in [Figure 1: see original paper]. The fundamental bipartite graph model shows user nodes (X) and item nodes (Y), enabling the construction of user-user relationships [FIGURE:1(b)] or item-item relationships [FIGURE:1(c)], where edge values represent the number of shared connection nodes.

Bipartite graphs in recommendation systems employ a probabilistic propagation mechanism. As shown in [Figure 2: see original paper], resources propagate from initial nodes (A, B, C) to lower-layer nodes with equal probability [FIGURE:2(b)], and subsequently return to upper-layer nodes [FIGURE:2(c)]. While this "equal probability" approach distributes resources uniformly, non-equal probabilistic propagation can also be implemented, where rating values determine the proportion of resources allocated to lower-layer nodes. This mixing process creates non-equivalent relationships among similar nodes, with varying resource mixing degrees for each node, thereby facilitating personalized recommendations.

2.2.3 Dynamic Weighting of Relevance We consider fusing the two aforementioned relevance calculation methods—content-based and behavior-based—by introducing dynamic weight factors λ_1 and λ_2 that dynamically adjust the proportion of each relevance measure based on their relative strengths, yielding more accurate and stable predictions. The dynamic weight factors are defined as:

$$\lambda_1 = \frac{\sum_{i,j} rel_b(I_i, I_j)}{\sum_{i,j} rel_b(I_i, I_j) + \sum_{i,j} rel_g(I_i, I_j)}$$

$$\lambda_2 = \frac{\sum_{i,j} rel_g(I_i, I_j)}{\sum_{i,j} rel_b(I_i, I_j) + \sum_{i,j} rel_g(I_i, I_j)}$$

where $\lambda_1 + \lambda_2 = 1$, $rel_b(I_i, I_j)$ represents bipartite graph-based relevance, and $rel_g(I_i, I_j)$ represents category tag-based relevance. The fused relevance is computed as:

$$rel(I_i, I_j) = \lambda_1 \cdot rel_b(I_i, I_j) + \lambda_2 \cdot rel_g(I_i, I_j)$$

This formulation captures a non-equivalent relevance relationship. For items without any user interaction, category-based relevance dominates, enabling the system to establish connections through item attributes and mitigate the cold-start problem for new items.

2.3 Rating Prediction

Based on the relevance calculation method, we obtain a list of related items $M = \{I_1, I_2, \dots, I_n\}$ for item I (where $I \notin M$), sorted in descending order by relevance strength. The predicted rating for user u on unrated item I is given by:

$$P_{u,I} = \bar{R}_I + \frac{\sum_{I_n \in M} rel(I, I_n) \cdot (R_{u,I_n} - \bar{R}_{I_n})}{\sum_{I_n \in M} |rel(I, I_n)|}$$

where $\bar{R}_I$ and $\bar{R}_{I_n}$ are the average ratings for items I and I_n, respectively, and R_{u,I_n} is user u' s rating for item I_n.

As evident from the formulas, both λ values and neighbor count influence experimental outcomes. However, since λ is computed using global data (formulas (2) and (3)), neighbor selection does not affect λ, making them independent variables. For set M, we select an appropriate size in experiments to balance computational efficiency and prediction accuracy.

2.4 Algorithm Steps

a) Input data and construct user-item rating matrix and item-category matrix.

b) Calculate content-based relevance $rel_g(I_i, I_j)$ using item-category matrix (Formula (1)).

c) Calculate behavior-based relevance $rel_b(I_i, I_j)$ using user-rating matrix with bipartite graph method.

d) Compute dynamic weight factors λ_1 and λ_2 (Formulas (2), (3)).

e) Fuse the two relevance measures using Formula (4) to obtain $rel(I_i, I_j)$.

f) Predict ratings for unrated items $P_{u,I}$ using Formula (5).

g) Evaluate results using test set.

3 Experimental Results and Analysis

3.1 Experimental Data

To validate the effectiveness of our proposed algorithm, we conducted experiments using the MovieLens film rating dataset, which contains user ratings on a scale of 1-5 (higher values indicating greater preference). The dataset includes genre information for each movie across 18 categories: Action, Adventure, Animation, Children' s, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, and Western, with each movie potentially belonging to multiple genres.

Specifically, we used the MovieLens-100k dataset containing 943 users, 1,682 movies, and approximately 100,000 ratings, resulting in a sparsity of approximately 93.70% . This extreme sparsity underscores the necessity of our proposed solution. In experiments, we randomly selected 80% of the data as training set and the remaining 20% as test set, performing multiple rounds of cross-validation to avoid 偶然性 and enhance model reliability.

3.2 Evaluation Metric

We employ Mean Absolute Error (MAE) to evaluate prediction accuracy, as it measures the average deviation between predicted and actual ratings:

$$MAE = \frac{\sum_{i=1}^{n} |p_i - q_i|}{n}$$

where n is the number of predicted ratings, p_i is the predicted score, and q_i is the actual user rating. Lower MAE values indicate better algorithm performance. To ensure result reliability, we conduct multiple experiments and report average MAE values.

3.3 Experimental Results

We first evaluate the non-dynamic weighting method from literature [11] for fusing relevance measures, with λ_1 and λ_2 in range [0,1] and $\lambda_1 + \lambda_2 = 1$. We vary λ_1 from 0.1 to 1.0 (with λ_2 decreasing from 0.9 to 0) and use all items with which users have interacted as the neighbor set. Results are shown in [Figure 3: see original paper].

The non-dynamic weighting method achieves its lowest MAE when $\lambda_1 = 0.9$. In contrast, our dynamic weighting approach [Figure 4: see original paper] consistently produces MAE values below this minimum across multiple experimental runs, with the mean MAE substantially lower than the best non-dynamic result. The bar chart represents MAE values from multiple dynamic weighting experiments, the dashed line indicates their mean, and the solid line shows the minimum value achieved by the non-dynamic method.

Comparisons with methods from literature [13-16] and traditional collaborative filtering [Figure 5: see original paper] demonstrate our approach' s advantages. Values represent averages across multiple experiments, resulting in slight fluctuations but overall strong performance. Our method (RDWCF) reaches its lowest MAE at approximately 45 neighbors, after which performance stabilizes. These results confirm that dynamic weighting and relevance calculations significantly improve recommendation system effectiveness.

---

4 Conclusion

This paper investigates the impact of dynamic weighting factors and relevance measures on recommendation systems. We systematically reviewed related research, introduced our methodology, and conducted comprehensive comparative experiments. Our findings indicate that: (1) bipartite graph-based relevance is stronger than tag-based relevance, suggesting user behavior is more influential than item attributes; (2) dynamic weighting factors outperform fixed weights; and (3) the combination of dynamic weighting with relevance measures enhances recommendation system performance beyond traditional fusion methods.

References

[1] Rich E. User modeling via stereotypes [J]. Cognitive Science, 1979, 3(4): 329-354.
[2] Pazzani M J, Billsus D. Content-based recommendation systems [M]. Berlin: Springer-Verlag, 2007: 325-341.
[3] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering [C]// Proc of the 14th Conference on Uncertainty in Artificial Intelligence. San Francisco: Morgan Kaufmann Publishers Inc., 1998: 43-52.
[4] Zhang Baofu, Yuan Baoping. Improved collaborative filtering recommendation algorithm of similarity measure [C]// Proc of International Conference on Materials Science. New York: AIP Publishing LLC, 2017: 109-132.
[5] Vig J, Sen S, Riedl J. Tagsplanations: explaining recommendations using tags [C]// Proc of International Conference on Intelligent User Interfaces. New York: ACM Press, 2009: 47-56.
[6] Adomavicius G, Tuzhilin A. Context-aware recommender systems [C]// Proc of ACM Conference on Recommender Systems. New York: ACM Press, 2008: 335-336.
[7] Zhou Tao, Ren Jie, Medo M, et al. Bipartite network projection and personal recommendation [J]. Physical Review E: Statistical Nonlinear & Soft Matter Physics, 2007, 76(2): 046115.
[8] Nilashi M, Ibrahi O, Bagherifard K, et al. A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques

[J]. Expert Systems with Applications, 2017, 92.
[9] Fan Jiaqi, Pan Weimin, Jiang Lisi. An improved collaborative filtering algorithm combining content-based algorithm and user activity [C]// Proc of International Conference on Big Data and Smart Computing. Piscataway, NJ: IEEE Press, 2014: 88-91.
[10] Liu Jian, Zhang Kun, Chen Xuan. Personalized recommendation algorithm based on tag and collaborative filtering [J]. Computer and Modernization, 2016(2): 62-65.
[11] Zhang Jinglong, Huang Mengxing, Zhang Yu, Wu Qingzhou. Collaborative Filtering Recommendation Algorithm Based on Label Optimization [J/OL]. Application Research of Computers, 2018, 35(10). [2017-09-27]. http://www.arocmag.com/article/02-2018-10-023.html.
[12] Sun Xiaohua. Research on sparsity and cold start of collaborative filtering system [D]. Hangzhou: Zhejiang University, 2005.
[13] Wang Mingjia, Han Jingti. Collaborative filtering algorithm based on user preference for project attributes [J]. Computer Engineering and Applications, 2017, 53(6): 106-110.
[14] Wu Yueping, Zheng Jianguo. A collaborative filtering recommendation algorithm based on improved similarity measure method [C]// Proc of IEEE International Conference on Progress in Informatics and Computing. Piscataway, NJ: IEEE Press, 2011: 246-249.
[15] Liu Jianping, Wang Yong, Yan Fenghua. An improved collaborative filtering recommendation algorithm [C]// Proc of the 1st International Conference on Networking and Distributed Computing. Piscataway, NJ: IEEE Press, 2010: 204-208.
[16] Pan Taotao, Wen Feng, Liu Qinrang. Collaborative filtering algorithm based on matrix filling and item predictability [J]. Acta Automatica Sinica, 2017, 43(9): 1597-1606.

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