

A Novel Recommendation Algorithm Based on Clustering Dissimilarity Measures

Liang Zhang^{1,*}, Xiaojing Liu¹ and Xue Zhou²

¹School of Economics and Management, Guizhou Normal University, Guiyang 550001, China

²School of management, University of Chicago, Chicago, IL 60637, United States

Received 3 January 2020; Accepted 11 April 2020

Abstract

With the rapid development of information technology and Internet computing, recommendation systems of various forms have been used in nearly all large-scale e-commerce platforms to different degrees. Besides accuracy, novelty also influences the satisfaction of users with these recommendation systems. To be considered novel, an item should show some differences from the preferences of users. To recommend diverse items under the precondition of guaranteeing accuracy, a clustering-based novel recommendation algorithm was proposed in this study. User preference was modeled by using global clustering and related-items clustering (RC), weighted distance was employed to calculate the dissimilarity of the to-be-recommended items from user preference, and item novelty was measured by combining satisfactory and unknown, followed by recommendation. The effectiveness of the proposed algorithm was verified by conducting an offline experiment. Results show that the recall rate of unknown items of RC increases by 15% whereas that of known items decreases by 60%, thereby indicating that the novelty of the recommendation result is remarkably improved. Meanwhile, the coverage rate increases by approximately 200%, thereby indicating that the proposed recommendation system demonstrates an improved ability to recommend long-tail items. This study provides references for improving the satisfaction of users with recommendation systems.

Keywords: Novelty, Dissimilarity, Clustering, Recommendation Algorithm

1. Introduction

As both the Internet and e-commerce are growing at unexpected rates, the Internet and online services (e.g., e-commerce and electronic news) have created an increasingly severe information overloading problem while simultaneously bringing convenience to users [1]. Personalized recommendation technologies can help users make wise decisions when choosing among a large number of products [2]. Under most circumstances, the recommendation lists generated by a recommendation system contains many similar items to guarantee accuracy. In recent years, studies on recommendation systems have begun to focus on how to correctly sort recommendation lists [3] to ensure that the items contained therein satisfy the diversified points of interest of users [4], and to guarantee the novelty of these items [5]. Diversity and novelty are also becoming increasingly important in the practical application of Top-N recommendation systems.

Both diversity and novelty require the items in a recommendation list to show a certain degree of difference from one another. Diversity emphasizes the satisfaction of diversified points of interest of users and requires that every two items in a recommendation list should be as different as possible. Meanwhile, novelty emphasizes going beyond the expectations of users. In this case, the items in a recommendation list should not only satisfy the preferences of users but should also differ from those items that are familiar to users. The most commonly used dissimilarity

(similarity) measures in recommendation algorithms include Jaccard, cosine distance, Pearson correlation, and their variants [6]. One commonly used method for measuring the difference between to-be-recommended items and user preference is the dissimilarity measure, which represents the average difference between a recommended item and the rated items [7]. Although measuring the preferences of users by taking the adopted items as a whole seems reasonable, the accuracy of such user preference modeling is reduced due to the inclusion of some behavioral data of sudden epidemic influence or temporary adoption [8-10]. Previous studies have mainly focused on improving the accuracy of recommendation systems, but how to select proper clustering algorithms to find novel items that diverge from user preference warrants further research.

To this end, user preference was modeled in this study by using global clustering (GC) and rated-items clustering (RC) algorithms. Minimum distance, average distance, and weighted distance were employed to measure the difference between the to-be-recommended items and user preference, and satisfactory and unknown attributes were combined to calculate the novelty of those items to be recommended to users in order to improve the novelty and coverage rate of recommendation lists while guaranteeing their reasonability and accuracy.

2. State of the art

Many scholars have examined the application of clustering algorithms to recommendation systems. For instance, Feng, L. et al. divided users and items into several subgroups, trained the traditional collaborative filtering model for each

*E-mail address: ftygyqz@vip.sina.com

ISSN: 1791-2377 © 2020 School of Science, IHU. All rights reserved.

doi:10.25103/jestr.133.10

cluster after obtaining the closely associated clusters of users and items, and remarkably improved the performance of several traditional collaborative filtering Top-N recommendations [8]. Wang, X. Y. et al. proposed a collaborative filtering recommendation model based on rough user clustering, allocated users to upper and lower approximations according to their similarities with the clustering center, formed an initial neighbor set for each user, searched the nearest neighbor of the target user online from its initial neighbor set, predicted item ratings, and offered recommendations to the target user, thereby effectively improving both recommendation accuracy and quality [9]. These studies have also performed clustering to find neighbor sets with high reference value and trained algorithms to improve the accuracy of recommendation systems. However, these processes can also influence the coverage rate of the recommendation results.

Clustering algorithms have also been applied in user preference modeling. Han, H. M. et al. clustered users and items into multiple clustering subsets according to a user-item evaluation matrix and the item attributes derived by domain experts, mined candidate items by using the item attributes selection method, and showed improvements in mean absolute error and root mean square error [10]. Jiang, M. Y. et al. introduced information entropy and blustering into collaborative filtering and obtained better results than the benchmark algorithm in terms of precision and calculation cost [11]. Raja, D. R. K. et al. proposed an individualized mobile multimedia application recommendation system based on user comments, comment texts, application descriptions, and application popularity, used the latent dirichlet allocation (LDA) topic model to analyze user comments and descriptions of application programs, and applied k-means clustering in all extracted application characteristics via principal component analysis while reserving only the optimal potential characteristics; their proposed system was able to guarantee the correlation and diversity of the generated recommendation list [12]. Das, J. et al. used two hierarchical spatial division data structures (k-d tree and Quadtree), clustered or divided user spaces in the system according to user location, and used these clusters to predict the ratings of target users, thereby reducing the operation time while maintaining an acceptable recommendation quality [13]. Sarwat, M. et al. classified three location-based ratings within the same framework [14]. Mezni, H. et al. proposed a context-aware web service recommendation method that pays special attention to the time dimension and combined the k-means clustering method with multi-population variants of the famous particle swarm optimization algorithm to exclude those similar users who share few public web services with active users in a specific context [15]. Selvi, C. proposed a supervised adaptive genetic neural network method based on the improved k-means clustering algorithm to localize the data points that are liked the most by users in order to provide effective recommendations [16]. These studies have integrated clustering technology with several algorithms, including LDA and the genetic neural network, and fully utilized spatial and contextual information to accurately describe user preferences. However, most of these studies are restricted to recommendation accuracy and ignored the improvements in diversity and novelty.

Clustering algorithms are often used to solve the data sparsity problem in recommendation systems. Zhang, D. et al. proposed the concepts of popular items and frequent raters and used the k-means algorithm to cluster users and

products to reduce the dimensionality of the UI matrix [17]. Yuan, H. N. et al. introduced the characteristic attributes of item contents, constructed a multi-instance-based user rating information expression model, and took full advantages of the content features of items and user rating information to improve rating prediction accuracy and effectively address the data sparsity problem [18]. Sun, Y. C. improved the traditional collaborative filtering algorithm by introducing a clustering algorithm, solving its “cold boot” and “data sparsity” problems, and enhancing its recommendation quality [19]. Rupasingha, A. H. M. et al. solved these problems by proposing an ontology-based clustering method that generates the ontology by virtue of field specificity and service similarity. This clustering method assumes that the user preference is empty according to the history of the user preference domain and can conveniently and effectively improve the data density of the user service dataset [20]. The above analyses have used clustering algorithms to solve the sparsity and cold boot problems of recommendation systems and improve their recommendation quality.

The fuzzy clustering algorithm can effectively realize user interest and preference modeling and be effectively integrated with scenario information. Liu, Z. B. et al. introduced models of user stable interest and current interest and used the fuzzy clustering algorithm to solve the data sparsity and equal consideration problems of user interests at different periods in the traditional collaborative filtering recommendation algorithm [21]. Mohanty, S. N. et al. proposed a rough-fuzzy clustering technology-based user access behavior recommendation framework [22]. The above studies have mainly applied scenario information in the fuzzy clustering algorithm to dynamically analyze user preferences, thereby improving recommendation quality.

Scenario information is generally used in clustering algorithms to improve the performance of recommendation systems. Zeng, Q. et al. proposed a collaborative filtering recommendation algorithm based on similarity propagation and scenario clustering, clustered users according to their calculated scenario similarities, identified additional nearest neighbors for the target user by following the similarity propagation principle, and offered recommendations by predicting the item ratings given by the target user [23].

Clustering algorithms can effectively find the points of interest of users and the correlation between them. However, in the novel recommendation, the recommended items should not only conform to user preference but should also be different from the items that are already known by the user. In this case, user preference can be modeled according to the cluster where the items already known by the user are located, and those items that show the greatest differences from these clusters are sent for recommendation to guarantee the diversity of the recommendation results.

The rest of this study is organized as follows. Section 3 applies three modeling methods, namely, weighted distance, global clustering, and rated-items clustering, to ensure the dissimilarity of the novel recommendation. Section 4 verifies the change degrees of various algorithm indexes, including accuracy and novelty, by conducting an offline experiment. Section 5 summarizes and concludes the paper.

3. Methodology

For users, novel items should have three key features, namely, satisfactory, unknown, and dissimilarity. Dissimilarity highlights the difference between the novel

item and the already known item and is often measured by using the average distance (Eq. (1)) or minimum distance (Eq. (2)) between the to-be-recommended item and rated-item sets based on user-item data, where $dis(i,u)$ denotes the distance between the to-be-recommended item i and the preference of user u , and I_u is the item rated by user u . The definitions of average and minimum distance can be easily understood at the intuitive level with a simple calculation and easy operation, but both of these methods have limited capability in judging dissimilarity.

$$dis(i,u) = \min_{j \in I_u} d(i,j) = \min_{j \in I_u} (1 - \cosine(i,j)) \quad (1)$$

$$dis(i,u) = \text{mean}_{j \in I_u} d(i,j) = \text{mean}_{j \in I_u} (1 - \cosine(i,j)) \quad (2)$$

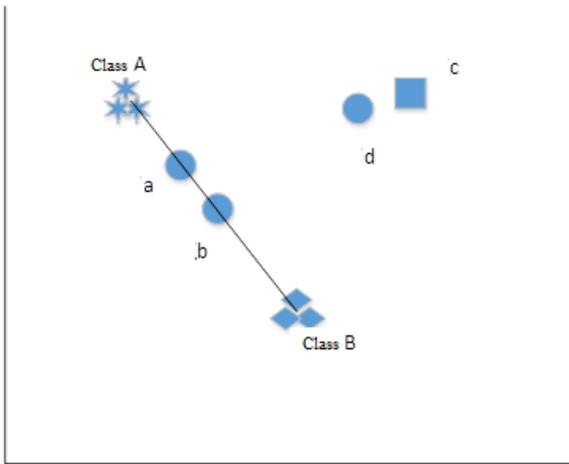


Fig. 1. Schematic diagram of the dissimilarity measure

3.1 Weighted distance method

The points in Fig. 1 denote the items. When the rated items contain classifications A and B, if average distance is used, then the dissimilarities of points a and b are equal. However, given that the dissimilarity of point b is greater than that of point a, minimum distance can be used to measure such dissimilarity. When the rated items include classifications A and B and point c, point d exhibit the maximum dissimilarity, whereas point c may be an outlier that represents a novel item previously found by the user. In this case, point d is more novel than point c.

An easy-to-operate processing method for the abovementioned phenomenon is to use the weighted average of average and minimum distance to measure the dissimilarity of the items for the target user as shown in Eq. (3), where λ denotes weight, $\lambda=0$ denotes average distance, and $\lambda=1$ denotes minimum distance. To investigate the differences between the algorithms λ is set to 0.5 in the experiment.

$$dis(i,u) = \lambda \min_{j \in I_u} d(i,j) + (1-\lambda) \text{mean}_{j \in I_u} d(i,j) \quad \lambda \leq 1 \quad (3)$$

3.2 Dissimilarity measure based on k-means global clustering

In Fig. 1, classifications A and B denote user preference, whereas outlier c denotes novelty. The dissimilarity of point d is smaller than that of point b if the weighted distance method is used, but the deviation of point d from the user preference is very large. Therefore, when the rated items do not completely represent user preference, calculating the

distance between the to-be-recommended item and each rated item to measure dissimilarity is somewhat irrational. To identify the classifications that represent the user preference among the rated items, the k-means algorithm can be used to classify all items. Assume that the quantity of items containing classification C_j among the rated items is w_j , then a greater value of w_j indicates a greater ability of classification C_j to represent user preference. This classification is called the preference classification (PC) of user u , and those items close to this classification have a low dissimilarity. On the contrary, if w_j value is very small, then classification C_j is probably an outlier that should be excluded in the dissimilarity measure because this classification does not represent user preference. Assume that the quantity w_j of items containing classification C_j among the rated items is greater than threshold value δ , then C_j belongs to the PC of user and represents user preference. The dissimilarity of item i for user u is defined in Eq. (4), where c_j denotes the centroid of classification C_j .

$$dis(i,u) = \lambda \min_{C_j \in PC} d(i,c_j) + (1-\lambda) \text{mean}_{C_j \in PC} d(i,c_j) \quad \lambda \leq 1 \quad (4)$$

3.3 Dissimilarity measure based on rated-items clustering

Assume that the points in Figure 2 represent rated items. The PC of users after global clustering includes classifications A and B, but the figure shows that points a and b are combined into one classification, whereas point c is not suitable for representing user preference. Therefore, clustering based on rated items is most appropriate for describing user preference. For the target user, the item cluster that can represent the user preference among rated items can be found while neglecting the outliers. Therefore, using the DBSCAN algorithm [24] is suitable. This algorithm divides the data points into the core, boundary, and noise points, with each point representing an item. The core point is located in the density-based cluster, the boundary point is located within the neighborhood of one or multiple core points, and the noise point can be any point except for core and boundary points. As previously mentioned the noise point is an outlier and cannot represent user preference, and the DBSCAN algorithm can be used to delete these points. When the rated items are divided by using the DBSCAN algorithm, the dissimilarity $dis(i,u)$ between the item and user preference is calculated by using Eq. (4). The concrete idea behind this algorithm is explained as follows:

- (1) All points are marked as core, boundary, and noise points. The number of points in the neighborhood of a core point should exceed the preset threshold δ .
- (2) All noise points are deleted.
- (3) A side with a distance of within ϵ is assigned among all core points.
- (4) Each group of interconnected core points forms a cluster.
- (5) Each boundary point is designated to its associated cluster of core points.

The DBSCAN algorithm needs to determine two parameters, namely, threshold value δ and radius ϵ . Obviously, the threshold value δ should be related to the quantity of rated items and refers to the quantity of data points for the DBSCAN algorithm. If many data points are available, then δ is partially large; otherwise, δ takes a small value. For a recommendation system, the quantity of rated items (i.e., user activeness) changes frequently and is determined by using a simple strategy. First, the accumulative distribution diagram of user activeness is drawn, the threshold value δ for the most active users is set to 7 and progressively declines from that point, and the threshold value for the least active users is set to 2. The radius ϵ is determined according to the threshold value δ , and the δ -distance of all points is calculated. δ -distance refers to the average distance from this data point to the nearest δ points. Afterward, sorting is performed in a progressively increasing order, and the distance value corresponding to the point showing the greatest change is determined as radius ϵ .

Three methods are used to measure the dissimilarity of an item from the target user preference, namely, the weighted distance (WD) dissimilarity algorithm, k-means for global clustering (global clustering or GC) dissimilarity algorithm, and the DBSCAN (rated-items clustering or RC) algorithm.

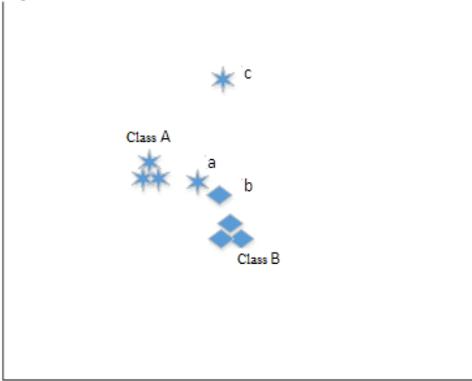


Fig. 2. Schematic diagram of the dissimilarity measure based on global clustering

Following the idea of the novelty algorithm based on diffusion of innovation (DI) [25], the k-means clustering method is used to divide all items into K classifications, and the number of classifications is determined according to silhouette coefficient.

Second, a model is established to measure user positivity in innovation adoption, and the probability $AP(i,u)$ for user u to adopt item i is calculated.

Third, the traditional algorithm is used to predict user preference for the item, and N items that are most probably liked are included in an alternative set.

Fourth, Eq. (5) is used to calculate the novelty of each item for recommendation, where $dis(i,u)$ is calculated by using the three previously mentioned algorithms. These algorithms are integrated with the traditional collaborative filtering (CF) and content-based recommendation (CBR) to generate six algorithms, namely, WD-DI-CF, GC-DI-CF, RC-DI-CF, WD-DI-CBR, GC-DI-CBR, and RC-DI-CBR.

$$Novelty(i,u) = p(i | like, u) \times \frac{AP(i,u)}{\log(\sum_{k=1}^n (n-k+1)i_k)} \times dis(i,u) \quad (5)$$

4. Result Analysis and Discussion

4.1 Experiment and evaluation metrics design

In Top-N recommendation, the traditional experimental method hides some item ratings in the existing datasets and then measures algorithm performance in terms of recall rate and accuracy. Here, the assumption that the hidden items cater to user preferences is reasonable, while the hidden items are already known by the user based on the training dataset. The novelty of each item is relative to specific users at a specified time. A user behavior dataset with a timestamp can be used and divided into two subsets by setting a time point. Those items with high ratings at this time point are deemed novel for the user. Following this idea, a detailed offline experimental scheme for novelty recommendation is designed as shown in Fig. 3 [26]. The experimental results are comprehensively evaluated by using three metrics in the recommendation list, namely, novelty (Eq. (6)), average popularity (Eq. (7)), and coverage (Eq. (8)).

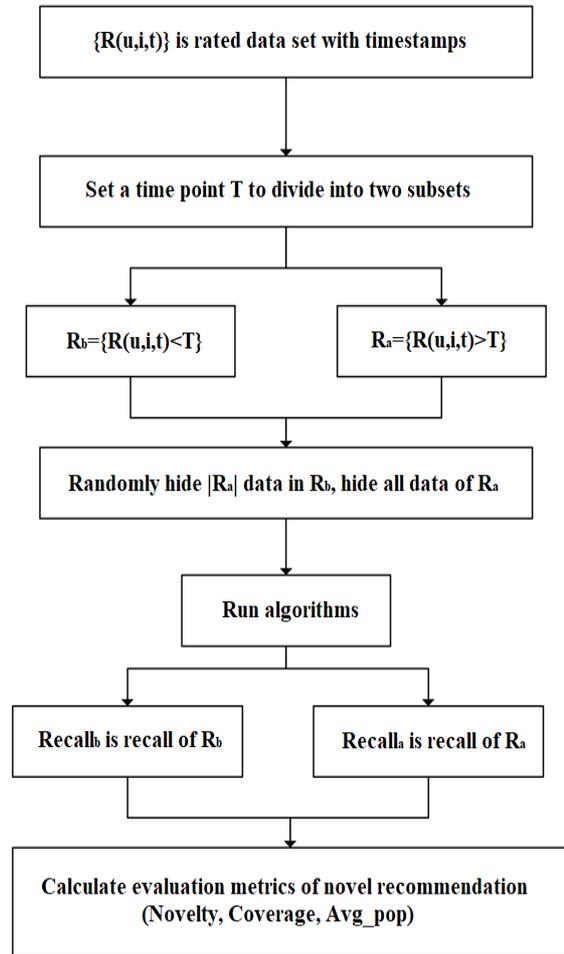


Fig. 3. Offline experimental scheme of novel recommendation

$$Novelty = \frac{1}{|U|} \sum_{u \in U} (2 \text{Recall}_{au} - \text{Recall}_{bu})$$

$$= \frac{1}{|U|} \sum_{u \in U} (2 \frac{|Rec_u \cap Hide_{au}|}{|Hide_{au}|} - \frac{|Rec_u \cap Hide_{bu}|}{|Hide_{bu}|}) \quad (6)$$

$$Avg_pop = \frac{1}{|U|} \sum_{u \in U} \frac{\sum_{i \in Rec_u} RN_i}{|Rec_u|} \quad (7)$$

$$Coverage = \frac{|\bigcup_{u \in I} Rec_u|}{|I|} \quad (8)$$

where $Hide_{au}$ and $Hide_{bu}$ are the sets of items hidden by user u in R_a and R_b , respectively, Rec_u is the set of recommended items for user u , RN_i represents the number of evaluation times of item i , and U and I represent the user and item sets, respectively. As shown in the Eq. (6), $Recall_b$ is the recall rate in the traditional experiment, and

$Recall_a$ is the accuracy measurement for the prediction of future user demand by the recommendation system. The quantity of recommendation lists is extremely limited. Thus, each recommendation list is expected to contain items meeting future user demands and reduce the items already known by the user, thereby accurately defining the meaning of novelty metrics. Average popularity and coverage refer to the measurement of the ability of the recommendation algorithm to extract long-tail products and also reflect the algorithm's ability in novelty recommendation.

The two datasets, namely, MovieLens and LastFM, are adopted in the offline experiment. The time interval is set to 14 days. Five experimental points are uniformly selected from the datasets of the previous year, previous rating data of each experimental point is set as R_b , the data within 14 days after these experimental data are used as the test data, namely R_a , and random rating data concealed by each user in R_b with the same quantity as that in R_a are taken as the training set. Each user is recommended with 20 items on the basis of this training set. The data obtained through the five repeated experiments are averaged to compare the differences of the two commonly used algorithms, namely, CF and CBR. The reasons are analyzed on the basis of various evaluation metrics. Considering that the differences of metrics are mainly considered, the change amplitude of experimental data is calculated using Eq. (9), where M_n and

M_s represent the new experimental and experimental data for reference, respectively, which can be expressed as:

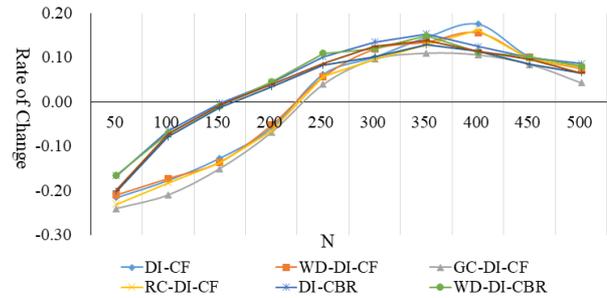
$$M = \frac{M_n - M_s}{|M_s|} \quad (9)$$

4.2 Novel recommendation based on the dissimilarity algorithm

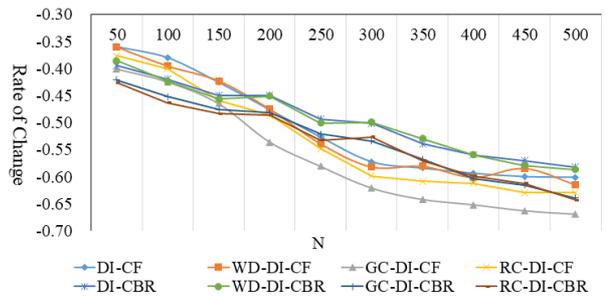
The novelty of the recommendation based on the dissimilarity algorithm is determined by observing the influence of dissimilarity calculation on the accuracy, popularity, and coverage of novel recommendation lists. The main algorithm parameter is the number (N) of items in the alternative set. Figs. 4 and 5 present the experimental results for the MovieLens and LastFM datasets, respectively.

(1) WD algorithm

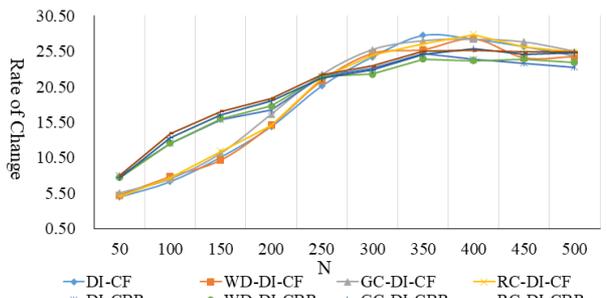
Given that the evaluation indexes of the WD algorithm do not show many changes, the weighted method cannot effectively improve the novelty of recommendation lists.



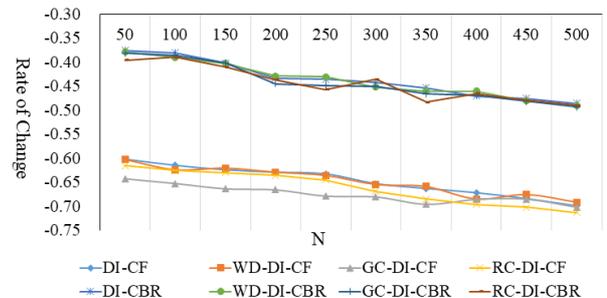
(a) Recall_a



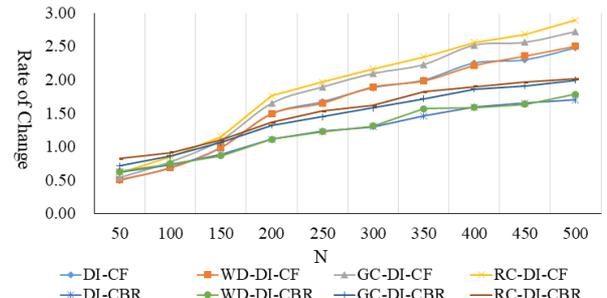
(b) Recall_b



(c) Novelty



(d) Avg_pop



(e) Coverage

Fig. 4. Experimental results for the MovieLens dataset

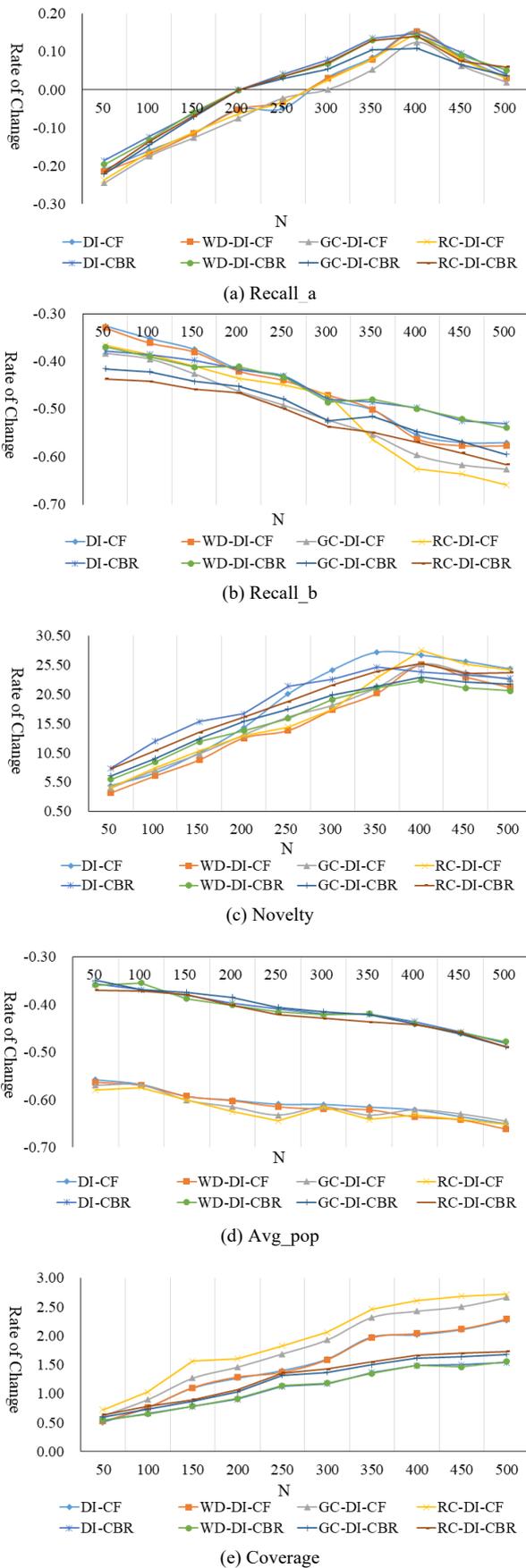


Fig. 5. Experimental results for the LastFM dataset

(2) GC algorithm

The GC algorithm is used to divide the items based on global clustering and to calculate the item dissimilarity for a user. Based on the CF results, Recall_a declines to a certain degree and shows the greatest decline among the three dissimilarity algorithms. Meanwhile, the novelty index only declines by a small margin because Recall_b simultaneously decreases at an amplitude equivalent to that of Recall. As shown in the CBR results, given that the decline amplitude of Recall_a is very evident, novelty obviously decreases compared with the DI algorithm. Moreover, the GC algorithm is not suitable to be fused with the CBR algorithm because the latter offers recommendations by calculating the similarities between items that carry the ratings and rated-items set. Although a certain accuracy of the alternative set is guaranteed, the GC algorithm still remarkably influences such accuracy. From index Avg_pop, four algorithms have not shown any obvious changes relative to the DI algorithm, thereby suggesting that dissimilarity has an insignificant influence on the average evaluation times of the recommendation list. In this case, for each user, those items showing a great dissimilarity not only include popular items but also unpopular items, which will not be analyzed in this paper. The coverage of the GC algorithm significantly increases and is only second to that of the RC algorithm.

(3) RC algorithm

According to the CF results, the novelty of the RC algorithm is obviously improved compared with that of the DI algorithm. The changes in the two indexes Recall_a and Recall_b reveal that the decline amplitude of Recall_a is obviously smaller than that of Recall_b in the RC algorithm, thereby proving the novelty of the recommendation system. In other words, the DBSCAN algorithm can effectively exclude rated items and has a subtle effect on the accuracy of the recommendation system. Meanwhile, according to the CF results, similar to the GC algorithm, both novelty and accuracy are degraded relative to the DI algorithm. The RC algorithm has the highest coverage, and the ability of the recommendation system to recommend long-tail items is remarkably improved.

4.2 Influence of weight

To investigate the influence of weight λ on the three algorithms, the MovieLens and LastFM datasets are used in the offline experiment, and the number of items in the alternative set is fixed at 400. Fig. 6 shows the experimental results for the MovieLens dataset. When λ changes from 0 to 1, this parameter does not exert much effect on both the WD and GC algorithms. Meanwhile, for the RC algorithm, all indexes are improved to some extent when the weight λ is 0 because the DBSCAN algorithm has deleted the noise points of the user, and the formed cluster can reflect user preference very well. Therefore, using minimum distance to calculate dissimilarity is reasonable. The experimental results for the LastFM dataset are similar to those for the MovieLens dataset, that is, not many differences are observed when the CBR algorithm is used. Therefore, the experimental results are not listed in Fig. 5.

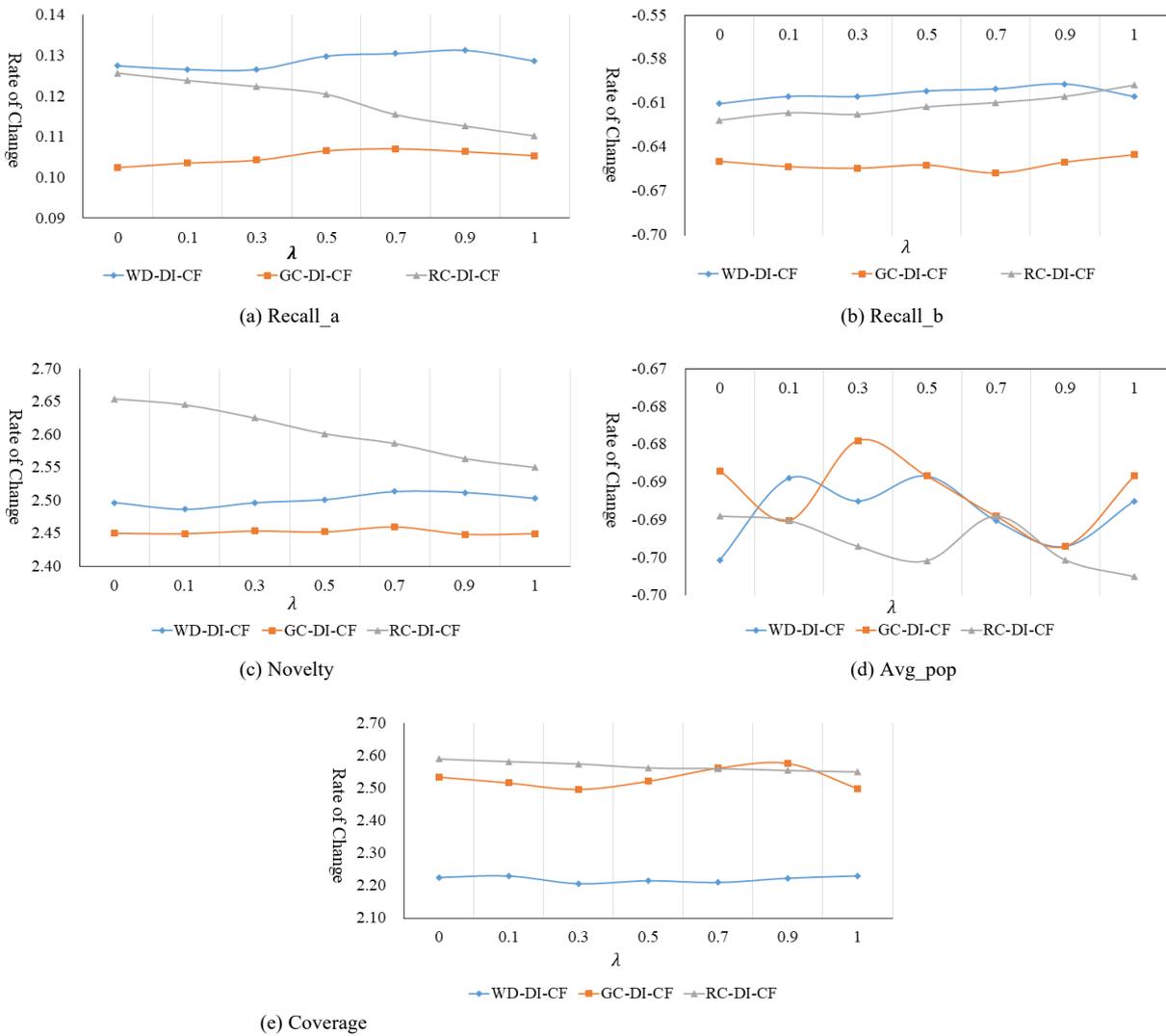


Fig. 6. Influence of weight on the three algorithms

5. Conclusions

Novel items have three attributes, namely, satisfactory, unknown, and dissimilarity, where dissimilarity means that these items should differ from user preferences to a certain degree. Weighted distance, global clustering, and rated-items clustering are used in this work to accurately model user preferences. The recommendation idea of the DI algorithm is used to design the WD, GC, and RC algorithms. The following conclusions are drawn:

- (1) Given that the evaluation indexes of the WD algorithm do not show many changes, the weighted method fails to improve the novelty of recommendation lists.
- (2) The GC algorithm shows an obvious decline in accuracy yet its coverage increases to a certain degree.
- (3) The RC algorithm improves the coverage of the recommendation system while ensuring that its accuracy and novelty are not obviously degraded.

The clustering algorithm was used in this study to model user preference, and the DBSCAN algorithm was proposed to cluster the rated items and to accurately calculate the difference between the to-be-recommended items and user preference. The coverage of the recommendation system was increased without sacrificing accuracy and novelty.

Therefore, this algorithm has some value in improving the satisfaction of users with recommendation systems. For a user, the novelty of to-be-recommended items is closely related to time, user preference demonstrates time-dependent changes, and user acceptance and understanding degrees for items are also changing along with time. Therefore, future studies should conduct an in-depth analysis while taking time information into account to improve the novelty of recommendation systems.

Acknowledgements

This work was supported by Guizhou normal university funded doctoral research projects (program number: GZNUD[2017]36), Innovation and Entrepreneurship Project of Guizhou Normal University, Major Education Reform Project of Guizhou Normal University in 2018 (Project Name: Innovation and Practice of E-commerce Talent Cultivation Mode with Deep Integration of Industry and Education in Guizhou province, China), and 2019 Provincial First-class Major in Marketing in Guizhou province, China.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License



References

1. Jannach, D., Resnick, P., Tuzhilin, A., et al., "Recommender Systems: Beyond Matrix Completion". *Communications of the ACM*, 59(11), 2016, pp.94-102.
2. Zhang, L., Luo T., Zhang, F., and Wu, Y., "A recommendation model based on deep neural network". *IEEE Access*, 6, 2018, pp.9454–9463.
3. Shi, L., "Trading-off among accuracy, similarity, diversity, and long-tail: A graph-based recommendation approach". In: *Proceedings of the 7th ACM Conference on Recommender Systems*, New York, USA: ACM, 2013, pp. 57–64.
4. Vargas, S., Baltrunas, L., Karatzoglou, A., et al., "Coverage, redundancy and size-awareness in genre diversity for recommender systems". In: *Proceedings of the 8th ACM Conference on Recommender Systems*, California, USA: ACM, 2014, pp.209–216.
5. Ziegler, C.N., McNee, S.M., Konstan, J.A., et al., "Improving recommendation lists through topic diversification". In: *Proceedings of the 14th international conference on World Wide Web*, Chiba, Japan: A CM, 2005, pp.22–32.
6. Anelli, V.W., Noia, T.D., et al., "The importance of being dissimilar in Recommendation". In: *The 34th ACM/SIGAPP Symposium on Applied Computing*, New York, USA: ACM, 2019, pp.112-123.
7. Kaminkas, M., Bridge, D., "Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems". *ACM Transactions on Interactive Intelligent Systems*, 7(1), 2017:2.
8. Feng, L., Zhao, Q.Ch.,Zhou, C.Q., "Improving performances of Top-N recommendations with co-clustering method". *Expert Systems With Applications*, 143, 2020, Doi: 10.1016/j.eswa.2019.113078.
9. Wang, X.Y., Qian, L., Huang, S.Y., "Collaborative filtering recommendation model based on rough user clustering". *New Technology of Library and Information Service*, 31(1), 2015, pp.45-51.
10. Han, H.M., Dong, X.H., Zuo, C.H., "A weighted recommendation algorithm based on multiview clustering of use"r". *Journal of Intelligent & Fuzzy Systems*, 38 (1), 2020, pp.441-451.
11. Jiang, M. Y., Zhang, Z.F., Jiang, J.Q., "A collaborative filtering recommendation algorithm based on information theory and bi-clustering". *Neural Computing & Applications*, 31(12), 2019, pp.8279-8287.
12. Raja, D. R. K, Pushpa, S., "Diversifying personalized mobile multimedia application recommendations through the Latent Dirichlet Allocation and clustering optimization". *Multimedia Tools and Applications*, 78(17), 2019, pp.24047-24066.
13. Das, J., Majumder, S., Gupta, P., "Collaborative Recommendations using Hierarchical Clustering based on K-d Trees and Quadtrees". *International Journal of Uncertainty Fuzziness and Knowledge-based Systems*, 27(4), 2019, pp.637-668.
14. Sarwat, M., Levandoski, J.J., Eldawy, A., et al., "LARS: an efficient and scalable location-aware recommender system". *IEEE Transactions on Knowledge and Data Engineering*, 26 (6), 2014, pp.1384–1399.
15. Mezni, H., Ait A., Benslimane, D., et al., "An evolutionary clustering approach based on temporal aspects for context-aware service recommendation". *Journal of Ambient Intelligence and Humanized Computing*, 11(1), 2020, pp.119-138.
16. Selvi, C., Sivasankar, E., "A novel Adaptive Genetic Neural Network (AGNN) model for recommender systems using modified k-means clustering approach". *Multimedia Tools and Applications*, 78(11), 2019, pp.14303-14330.
17. Zhang, D., Hsu, C.-H., Chen, M., et al. "Cold-start recommendation using bi-clustering and fusion for large-scale social recommender systems". *IEEE Transactions on Emerging Topics in Computing*, 2(2), 2014, pp.239–250.
18. Yuan, H. N., Zhou, T., Han, Y. N., et al., "Collaborative recommendation algorithm based on MI clustering". *Journal of Wuhan University information science edition*, 40(2), 2015, pp.253-257.
19. Sun, Y. C., "Study of book recommendation system based on clustering analysis algorithm". *Library Theory and Practice*, 5, 2015, pp.76-79.
20. Rupasingha, A. H. M, Paik, I., "Alleviating sparsity by specificity-aware ontology-based clustering for improving web service recommendation". *IEEE Transactions on Electrical and Electronic Engineering*, 14(10), 2019, pp.1507-1517.
21. Liu, Z. B., Xiao, S. F., "Collaborative filtering algorithm based on fuzzy clustering of user interests". *New Technology of Library and Information Service*, 31(11), 2015, pp.12-17.
22. Mohanty, S. N., Parvin, J. R., Kumar, K. V., et al., "Optimal rough fuzzy clustering for user profile ontology based web page recommendation analysis". *Journal of Intelligent & Fuzzy Systems*, 37(1), 2019, pp.205-216.
23. Zeng, Q., Cheng, X., "Study of network collaborative filtering recommendation algorithm based on similarity propagation and scenario clustering". *Journal of Modern Information*, 36 (11), 2016, pp.50-54.
24. Hermawati, R., Sitanggang, I.S., "Web-Based Clustering Application Using Shiny Framework and DBSCAN Algorithm for Hotspots Data in Peatland in Sumatra". *Procedia Environmental Sciences*, 33, 2016, pp.317-323.
25. Zhang, L., Qian, X.S., Lv, P., et al., "A Novel Recommendation Algorithm Based on Diffusion of Innovation Theory". *Journal of Digital Information Management*, 12(6), 2019, pp.87-95.
26. Zhang, L., Peng, L. F., Phelan, C.A., "Novel recommendation of user-based collaborative filtering". *Journal of Digital Information Management*, 12(3), 2014, pp.165-175.