

Hybrid recommendation system combining collaborative filtering and content-based recommendation with keyword extraction

Shuting Zhang^{1,6}, Kechen Liu^{2,7}, Zekai Yu^{3,8}, Bowen Feng^{4,9} and Zijie Ou^{5,10}

¹Glasgow College, University of Electronic Science and Technology of China, Sichuan, 611700, China

²College of Computer and Information Technology, China Three Gorges University, Yichang, 443002, China.

³Ningbo Binhai International Cooperative School, Ningbo, 315800, China,

⁴Southland International High School Qingdao Campus, Qingdao, 266555, China

⁵Guangdong Experimental High School, Guangzhou, 510000, China

⁶2019190504007@std.uestc.edu.cn

⁷2019125132@ctgu.edu.cn

⁸2368484453@qq.com

⁹bowenfeng1026@gmail.com

¹⁰ouzijie2020@163.com

Abstract. With the development of recommendation systems, large amount of information collected from e-commerce could help customers to find the potential interesting products. Collaborative filtering and content-based recommendation systems are two common recommendation systems. While collaborative filtering has the problem of cold-start, content-based recommendation system could not explore the potential interests of users. Hybrid system combining these two techniques could achieve better results. This paper applies hybrid recommendation methods to the Amazon food reviews and evaluate the results in the aspects of precision, recall, diversity and novelty. It is found that the weighted hybrid recommendation system combining 0.95 weight of collaborative filtering and 0.05 content-based recommendation system achieves a good precision and diversity.

Keywords: recommendation systems, collaborative filtering, hybrid system.

1. Introduction

Various recommendation system techniques have been proposed since the mid-1990s [1], at the beginning, the recommendation system was more to help people filter emails, but with the development of technology, the recommendation system, now, focuses on helping customers to find what they need more efficiently and accurately.

The recommendation system has the ability to predict based on the consumer's purchase record-whether a particular consumer would prefer an item or not [2]. Collaborative filtering as one of the most common methods focuses on filtering information depending on correlative information between

products and consumer, products and products, consumer and consumer, like linking them according to the relevant evaluation [3].

Classic recommendation systems include collaborative filtering and content-based recommendations. Specifically, collaborative filtering is divided into user-based filtering and item-based filtering. User-Based Collaborative filtering systems collect human judgements (known as ratings) for objects in a specific domain and analyze some users who have similar information requirements or interests [4]. Item-based techniques based on the user-item matrix find relationships between different items to make recommendations to users indirectly [5]. Nowadays, there are more than 7 billion people in the whole world, it is difficult to compute every change of individuals. Compared to other recommendation techniques, Collaborative filtering systems require less computational resources. Classic recommendation systems include collaborative filtering and content-based recommendations. Specifically, collaborative filtering is divided into user-based filtering and item-based filtering. User-Based Collaborative filtering systems collect human judgements (known as ratings) for objects in a specific domain and analyze some users who have similar information requirements or interests [4]. Item-based techniques based on the user-item matrix find relationships between different items to make recommendations to users indirectly [5]. Nowadays, there are more than 7 billion people in the whole world, it is difficult to compute every change of individuals. Compared to other recommendation techniques, Collaborative filtering systems require less computational resources.

Item-based Collaborative filtering (Item-based CF) is more an available method than the user-based Collaborative filtering. The reason is that items are more static than users. Users may change their personal interests or actions in a short period of time. Whereas, products will not change that much, thus leading to lower compute times. Moreover, by using item-based CF, data can be computed offline and without frequent model re-training [6]. In this system, instead of finding a relationship between uses, items used such as vegetables or meat are compared with each other so that it will be easier to make recommendations.

Another Content-based filtering technique recommends items to users according to the interests of the users. The system first identifies the properties of the items to construct item profiles and then establishes the user profile of the interests based on one's past behaviors such as purchases and ratings. Then items with profiles similar to or contained in the user profile are recommended to users. Therefore, content-based filtering has the advantages that it caters to the preferences of the users [7].

However, the primary problem with content-based filtering is that it could not explore the potential interests of the users. With only past behaviors of the users as the source of information, content-based filtering could only recommend the items consistent with the detected interests of the users. Another problem is that the features of items might be hard to obtain [8,9].

Both content-based recommendation and collaborative filtering have strengths and weaknesses. A hybrid recommendation system combining content-based recommendation and collaborative filtering could recommend both novel and familiar products to uses [7].

Keywords, as a succinct representation of a document's contents, are extracted as the tags for the content-based recommendation [10]. Keyword Extraction, as a method that analyzes those keywords in a paragraph by using several functions such as textblob and tf-idf, is now used in many online platforms whose information is always short and have poor tags [11].

Here are the Contributions/outcomes :

- Keywords from both product names and comments were extracted as the tags of the product.
- A hybrid collaborative filtering and content-based recommendation approach was applied to Amazon food reviews, for better novelty and diversity
- Precision, recall, novelty and diversity were used to evaluate the performance of the system.

2. Literature review

2.1. Recommendation system

Most of the recommendation systems had been developed in the 1990s. Recommendation system pushed a myriad of fields and then improved life quality [12]. The recommendation system focuses on providing the most suitable products and services to various users by anticipating individual's preference depending on related messages about the products, users and the intersection between them [13]. With more fields and individuals relying on recommendation systems, there will be more and more different data to analyze. So, it requests faster and precise analysis on recommendation systems, and this is the most important development direction of the recommendation system now.

For businesses: The goal of every manufacturer is to increase their earning, and Amazon, as one of the most successful companies, also gained a lot from the recommendation system in the past few years. Amazon delved into the purchasing history of its users to find the tag (spending power and type of merchandise) to its consumers. Recommendation system led to the 29% sales increase for Amazon from 2012 to 2016 [14]. There was also inflation during this period to impact company sales, and the average inflation rate between 2012 to 2016 is 1.12% [15]. Then, the real output increased by about 27%.

For consumers: Recommendation system enhance the shopping experience of consumers. Cause recommendation system help consumers to save time picking up goods and provide consumers with what they want. When the platform has the personalized feed (built based on the recent purchases) and comprehends the consumers demand(from other similar consumers), it tends consumers to buy more which leads to the customers satisfaction increase [14].

2.2. Collaborative filtering

In the field of recommendation algorithms, collaborative filtering (CF) is the most widely adopted recommendation strategy [6].

1) **User-based collaborative filtering:** Based on the user's neighbors, User-based Collaborative Filtering (UCF) is a recommendation system that predicts items that a user might like on the basis of ratings given to that item by the other users who have similar tastes, with that of the target user. As a result, the recommendations are solely based on the preferences of the user's neighbors [16]. But it causes a main problem that some users may buy few items, in this case it is difficult to find the users with similar preferences to make recommendations.

2) **Item-based collaborative filtering:** Since it was reported in 2001 in the research of B. Sarwar, item-based filtering has been applied in many e-commerce platforms. Item-based CF (ICF) makes recommendations to a user based on their previously consumed items, determining the user-item pair's relevance by comparing the target item to previously consumed items [6,17].

To be specific, based on the set of items rated by the target user, this algorithm uses some similarity measures like Cosine, Pearson to calculate the similarity matrix to denote the most similar items. Once the most similar items have been identified, the prediction is calculated by taking a weighted average of the target user's ratings on these similar items [5].

Since it was reported in 2001 in the research of B. Sarwar, item-based filtering has been applied in many e-commerce platforms. Because there are some advantages of ICF over UCF:

- It is easier for the composability of ICF in user preference to implement online personalization [17].
- It can make the accuracy and interpretability of user preference modeling improved and recommend items more accurately owing to representing a user with more information from their consumed items [6,18].

2.3. Content-based recommendation

Content-based recommendation (CB) could be dated back to the ideas of "Selective Dissemination of Information" in 1960s, developed by Hensley, in which the item information is matched with the

information stored in user profile [19]. In the web age, CB could help users to find interesting websites [20] and could also be applied in the e-commercial field.

CB makes recommendations by measuring the similarities between the features of new items and the features of the items preferred by the user. Specifically, the features of the items are extracted based on the description of the items and are stored in the item profile, while the items preferred by the users are extracted according to their historical behavior and stored in the user profiles [7].

Table 1. The pros and cons of CB.

Pros	Cons
no cold-start	focus on past interests hard to extract features

Compared with CF that requires a large amount of data to start, CB has the advantage that there is no cold-start problem (new items without historical data could not be recommended) for a new item as long as its feature could be extracted and is sometimes used to enhancing CF [21].

The first problem of CB is that CB could only focus on the past interests of the user because it simply utilizes the information that is shown in user profile, while collaborative filtering could enlarge the range of the recommendations based on the users who share similar tastes [8,9]. Based on that, some hybrid recommendation methods combining

CF and CB have been developed [7].

Another problem of CB is that it could be hard to extract features. By contrast, CF is trained based on the purchase and ratings of the user and therefore does not have this problem.

In recent years, however, some methods have been developed to extract the features of the items. For example, McAuley trained a convolutional neural network on the images to obtain visual features [22].

2.4. Hybrid method

To exploit the advantages of collaborative filtering (CF), several types of hybrid approaches are proposed to improve the accurateness of results which are computed by the collaborative filtering (CF) [8]:

- 1) Weighted Approach: Each kind of recommendation approaches give results and then these results are combined together to give one final prediction.
- 2) Switching: In certain condition, the most suitable type of recommendation system is selected and give final prediction. The condition had to be set before doing recommendation.
- 3) Mixed: The prediction of multiple recommendation approaches will be provided to customers.
- 4) Cascade: The prediction of one type of recommendation system is improved by another.
- 5) Feature combination: Different recommendation methods give their own features to one single recommendation technique. Then this technique provides final prediction.
- 6) Feature augmentation: The output of one specific recommendation method is provided to another one.
- 7) Metal-level: It is that one kind of recommendation method creates one type of recommendation model, then provided to another kind of recommendation method to get final prediction.

2.5. Keyword extraction

Semantic analysis is very important in a recommendation system. In a recommendation system, keyword extraction (KE) is indispensable. KE is described as a task that automatically identifies a group of words that best explain the most important information in a text: key phrases, key segments, key terms, or simply characterize the themes mentioned in the document [23]. Nevertheless, the rise of social media platforms like Twitter and Facebook has paved the way for more casual data to be shared online. These messages are often shorter than web pages, particularly on Twitter, where content is limited to 140 characters. The language is also more informal, with many communications incorporating typos, slang (e.g. cday), and abbreviations, among other domain-specific artifacts. Existing datasets and models in many applications, such as parts-of-speech (POS) tagging, Machine Translation, Named Entity

Recognition, Information Retrieval, and Summarization, tend to perform much poorer on these domains [24].

Typically, people create such text to share their current status, recent news, or fascinating occurrences with their friends or followers. Text social tidbits are what we call them. Facebook status updates and Twitter messages are two instances of social snippets [25]. The Automatic Keyword Detection include four parts, and the different methodologies and the algorithms used under each methodology are shown in Fig.1 below [26].

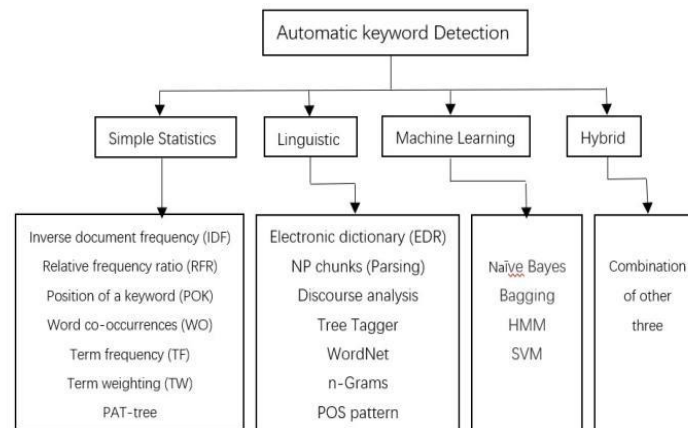


Figure 1. Keyword detection methods.

Keyword extraction and keyword assignment are the two existing techniques to automated keyword indexing. In keyword extraction, words found in the document are examined for attributes such as frequency and length to discover those that appear to be relevant. Keywords are picked from a regulated vocabulary of phrases, and documents are categorized into classes based on their content, which match to aspects of the lexicon [27]. Starting with the former one, tfidf is one of the functions that are widely used in keyword analysis. Hulth et al. mentions that the document itself is, without a doubt, the most significant knowledge source for locating relevant descriptors for a document [28]. Term frequency or the tf metric can be used to choose the most central terms in a document: Frequent words can be deemed to be relevant since they allow for document length normalization. The language context is a second significant source of information regarding the comparative value of descriptors: a frequent phrase is only relevant if it is often infrequent. The standard collection frequency or idf metric may be used to estimate this insight: calculating the percentage of documents in which a word appears. To ensure the tags obtained by tfidf, wordnet is a good choice to determine the characteristics of these words.

3. Data

The dataset is named Amazon food and extracted from Kaggle. The detailed description of fields is shown in the Table 2.

Table 2. The fields and detailed descriptions of the dataset.

Field	Description
Product ID	the unique ID for products in Amazon
User ID	unique ID for each user on Amazon
Product name	title for each unique product ID
Evaluation	rating given for the product on a scale of 1-5
Summary	summarized review of the users
Comment	review given for the product

Table 3. The detailed values of item.

Item	Value
Total number of reviews	41710
Number of products	36624
Number of users	34605
Average number of reviews per product	1.0583
Maximum number of reviews per product	632
Minimum number of reviews per product	1
Number of reviews with score 5	26180
Number of reviews with score 4	5999
Number of reviews with score 3	3316
Number of reviews with score 2	2328
Number of reviews with score 1	3882

4. Methodology

4.1. Keyword extraction

Food titles may contain words that reference the quantity of food, such as “5 ounce”, “3 oz”, and “15 lbs”. For this research, we will assume quantity has no influence on a review, so we remove all words that reference quantities from the analysis. We gather a list of common words used in referencing quantities, like ounce, oz, gram, grams, etc () and use this list to filter out these words from the dataset.

List of quantity words removed: ounce, oz, gram, grams.

We use the regex API (re) from Python to perform text preprocessing. To delete numbers, we use re.sub to delete everything which is not a letter or space. This function is used to replace occurrences of a particular substring with another substring [29]. In our program, we use space to replace the elements which are not English letters and space. That ensures the text we use does not contain something the keyword extraction methods are not able to analyze.

After preprocessing the elements in the dataset, the program starts to roughly extract keywords from the text. We use rake-nltk to analyze first. Rake-nltk is a domain independent keyword extraction algorithm which tries to determine key phrases in a body of text by analyzing the frequency of word appearance and its co-occurrence with other words in the text [30]. There are three main steps which rake uses to extract keywords, which is lemmatize Text, select potential phrases, and score each phrase. Lemmatize text is a process that finds the root of each word and then summarizes words which have the same meaning into one. Select potential phrases means that stop-words that influence the processing must be removed from the text in order to connect letters before and after the stop-words together to analyze the meaning. The third step is scoring each phrase. Just like its name said. After analyzing every part of the sentence input, rake would give a rank to each keyword to find which one is the most important [31]. Whole process of keyword extraction was shown in Fig.2.

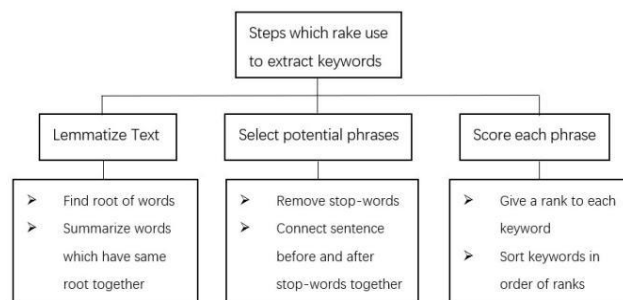


Figure 2. The process of keyword extraction.

4.2. Content-based recommendation

The first and second step of the content-based recommendation system is to construct the item profile and user profile, representing the features of the item and the user separately. The final step is to make recommendations by measuring the similarities between established item profiles and user profiles, as shown in Fig. 3.

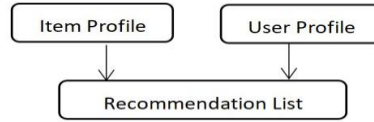


Figure 3. The procedure of recommendation system.

To construct the item profiles, the product name and comment of the product was processed by eliminating the punctuation, numbers and stop words (frequent and trivial words such as “we”) which could not represent the features of the products. Then the key words of product name and comment were extracted, illustrated in the keyword extraction section. The item profile was constructed in a vector space, in which each of the key words occupies one dimension and has a score calculated by tf-idf (Formula 1)

$$w_i = tf_i \cdot \log\left(\frac{n}{df_i}\right) \quad (1)$$

tf is the number of times the word i appears in the key word set of the item divided by the total number of keywords of the item. n is the total number of items and df_i is the number of items whose profile contains word i. The ifidf score measures how each key word could represent the features of one item. The higher the score is, the more likely the item could be represented by it.

The user profile was calculated by the weighted sum of the profile vectors in historical items. Items with a higher rate will obtain a higher weight. The similarity between one user (D) and one item (P) is calculated by the cosine similarity measure [32] (Formula 2)

$$sim(D, P) = \frac{D \cdot P}{||D|| \cdot ||P||} = \frac{\sum_k u_k \cdot w_k}{\sqrt{\sum_k u_k^2 \cdot \sum_k w_k^2}} \quad (2)$$

4.3. Item-based collaborative filtering

B. Sarwar summarized the process of item-based collaborative filtering in his research in 2001: We extract the similarity between items by analyzing “User-Item Rating Matrix” to find a collection of neighbor items most similar to the recommended user’s favorite item set. Then we sort the user’s list of interested programs for these items according to the possible score given by the user.

1) **Step 1: Similarity Measures:** We compute the similarity between each two items on the dataset using Cosine-based Similarity.

Cosine-based Similarity- In this case, Cosine-based Similarity is to evaluate the similarity of two vectors by calculating the cosine of the Angle between them. Consuming that there are item u and item v, we need to calculate the Cosine-based Similarity between them (Formula 3)

$$W_{\mu\nu} = \frac{|N(\mu) \cap N(\nu)|}{\sqrt{|N(\mu)| |N(\nu)|}} \quad (3)$$

$W_{\mu\nu}$ denotes Cosine similarity of ν and μ , $|N(\mu)|$ is the number of users who like item, $|N(\mu) \cap N(\nu)|$ is the number of users who like both good μ and good ν .

2) **Step 2: Generate a collection of neighbor items:** After calculating the similarity between items, Item-Based collaborative filtering algorithm needs to find the nearest neighbor item set of the

target item. In this neighbor set, a nearest neighbor set M, which is arranged in descending order according to the size of similarity, is generated for the target item.

3) **Step 3: The user's prediction scoring method for the target item:** User U's prediction of item I is calculated by summing user ratings for items similar to I. Each rating is weighted by the corresponding similarity between item I and item J(Formula 4)

$$P_{u,i} = \frac{\sum_{\text{all similar items } N} (s_{i,N} * R_{u,N})}{\sum_{\text{all similar items } N} (|s_{i,N}|)} \quad (4)$$

4.4. Hybrid recommendation system

Two hybrid recommendation systems combining collaborative filtering and content-based recommendation techniques were applied to the Amazon food dataset.

1) **System 1:** Collaborative filtering and content-based recommendation were implemented independently, after which the items on the recommendation list were sent to the content-based recommendation system to obtain the supplement of the recommendation items.

2) **System 2:** The collaborative filtering and content-based recommendation were implemented independently. The final similarities were the weighted sum of the similarities resulting from collaborative filtering and content-based recommendation. The items with high similarities to the items purchased by the user were recommended. Different values of weights between [0,1] were tested.

4.5. Evaluation

To evaluate and select recommendation system algorithms [33], recommendation system properties including prediction accuracy, novelty and diversity can be calculated. Prediction accuracy including precision (Formula 5) and recall (Formula 6) evaluates to what extent the items in recommendation lists are similar to the items purchased by users.

Table 4. Possible results of the recommendation.

	Recommended	Not recommended
Used	True-Positive (tp)	False-Negative (fn)
Not used	False-Positive (fp)	True-Negative (tn)

$$\text{Precision} = \frac{\#tp}{\#tp + \#fp} \quad (5)$$

$$\text{Recall(} \text{TruePositiveRate)} = \frac{\#tp}{\#tp + \#fn} \quad (6)$$

Novelty (Formula 7) calculates how dissimilar the items in the item list are to the marked items of users. #Z_u is the set of items purchased by the user. Z_u is the set of items on the recommendation list.

$$\text{novelty}_i = \frac{1}{\#Z_u} \sum_{j \in \#Z_u} [1 - \text{sim}(i, j)], \quad i \in Z_u \quad (7)$$

Diversity (Formula 8) evaluates the dissimilarity between items in the item list [35]. R[u] is the set of items being recommended.

$$\text{Diversity}(R(u)) = \frac{\sum_{i,j \in R(u), i \neq j} (1 - \text{sim}(i, j))}{\frac{1}{2}|R(u)|(|R(u)| - 1)} \quad (8)$$

5. Results

5.1. ICF and content-based recommendation system

Fig. 4 and Fig. 5 compare the performance of ICF and content-based recommendation system based on precision, recall, diversity and novelty when the recommended number increases from 2 to 9. It is apparent from Fig. 4 and Fig. 5 that the downward trend in precision in ICF and content-based recommendation system can be seen following the addition of the recommended number, whereas recall in ICF and content-based recommendation system increases gradually.

It can be seen from the results that precision is reported 0.23 in ICF and recall in ICF is 0.35 when the recommended number is 2. As seen from Fig. 4 and Fig. 5, these values are greater as compared to those of content-based recommendation system. Overall, content-based recommendation system performs better than ICF in terms of diversity and novelty, but worse than ICF in terms of precision and recall.

5.2. Hybrid recommendation systems

For the weighted recommendation systems, two different weighted systems are tested-0.95 weighted CF with 0.05 CB and 0.2 CB with 0.8 CF. The performance of two systems in the aspect of precision, recall, diversity and novelty when the recommend number is 3 is shown in Fig.6. The similarity used to calculation the parameters of evaluation is chosen to be of CB to make fair comparisons. Obviously, the 0.95 weighted CF + 0.05 weighted CB performs better in all the aspects of precision, recall, diversity and novelty.

In meta-level hybrid system tested, two items in the recommendation lists of eight are generated from the CF. Each of the two items is fed into CB and generates three results separately from CB. The results of meta-level hybrid system are compared with the CF, CB, 0.95CF with 0.05CB systems when the recommend number is eight and the result is shown in Fig.7. As it could be observed, the performance of meta level hybrid systems are the worst among the four systems, while 0.95CF with 0.05CB weighted systems has the best performance in the aspects of recall, diversity and novelty.

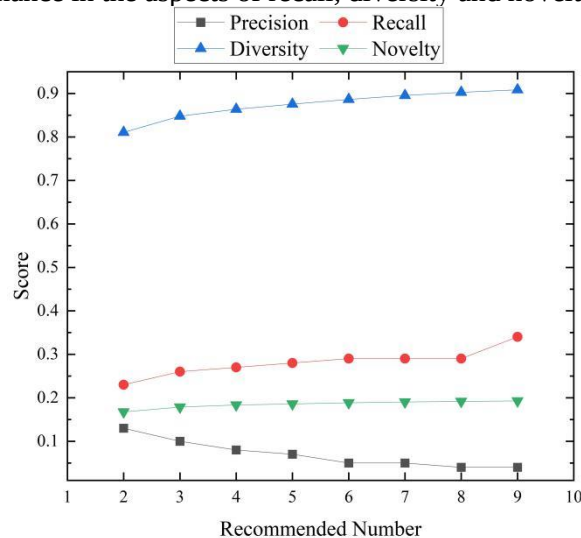


Figure 4. Results of item-based collaborative filtering recommendation system.

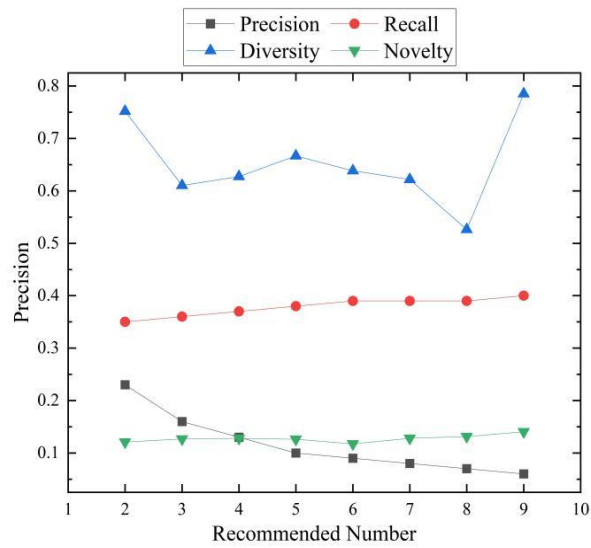


Figure 5. Results of item-based collaborative filtering recommendation system.

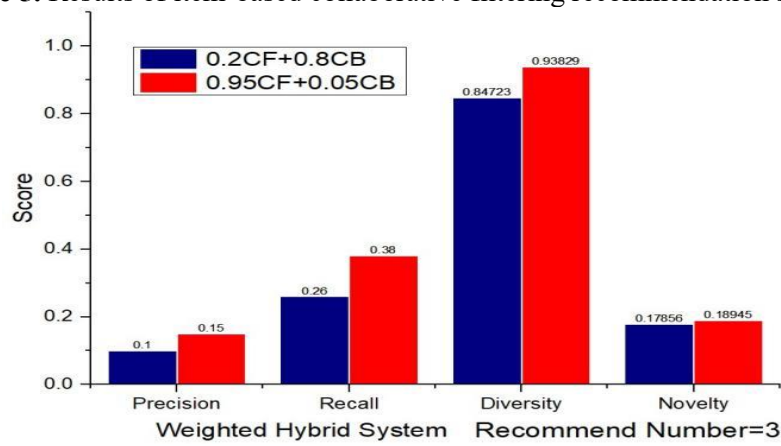


Figure 6. Results of two weighted hybrid recommendation systems.

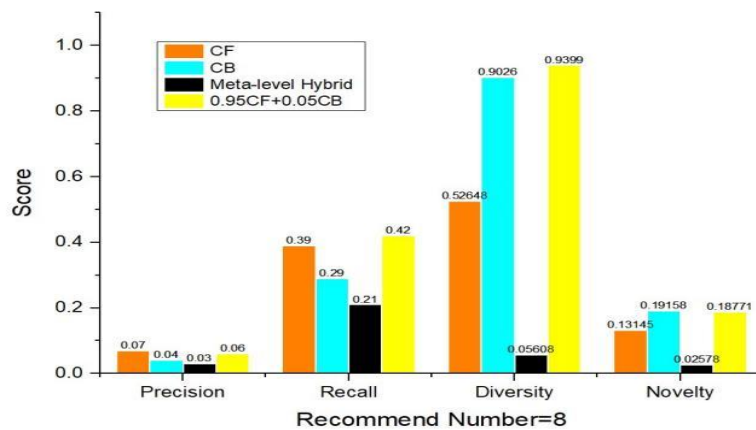


Figure 7. Results of four systems when the recommend number is 8.

6. Discussion

Firstly, the precision and recall are relatively low in all four recommendation systems with the value of lower than 0.5. This is for the reason that the data of the dataset chosen is extremely sparse—the average reviews made for each product is 1.0583, which means the advantages of CF could not be maximized. Even so, the performance of CF in the aspects of precision and recall is the best when the recommend number is 3.

In Fig.7, the diversity of CB is somewhat high and this may be caused by the differences in the similarities of each product. When the 0.95 weighted CF is combined with 0.05 CB, the recall, diversity and novelty are all improved. Its diversity is higher than CB while its recall is higher than CF. It proves that the weighted recommendation system could combine the advantages of both CF and CB.

However, the result from meta-level system is unsatisfactory in four aspects of precision, recall, diversity and novelty, which is resulted from the low proportion of the recommend items generated by CF. In conclusion, the recall, diversity and recall could all be improved by enhancing the weight of CF, while the cold-start problem could be handled by the CB.

Future analysis could be explored by applying the hybrid methods to a less sparse dataset and evaluates the performance of the recommend systems. Furthermore, the time could be added to the system as another parameter for the interests of people may change over time.

7. Conclusion

In this paper, in order to reduce the impact of the cold-start and the sparsity problems, we apply a hybrid recommendation system combining the respective strengths of ICF and content-based recommendation systems to the dataset. We also compare the performance of the hybrid recommendation system with that of the content-based recommendation system and ICF in terms of precision, recall, novelty, and diversity. The results show that the weighted hybrid recommendation system combining 0.09 weight of collaborative filtering and 0.05 content-based recommendation system has better accuracy and diversity compared to the content-based recommendation system and ICF. This model can be a good way to support users in discovering their potential interest preferences. At the same time, it makes a positive impact in solving user cold-start problems and item cold-start problems. But the major limitation of this model is that its performance on sparse datasets is not high when using precision to evaluate the recommendations. Our data suggests that we still have a long way to go to improve the quality of recommendations. Here we list two potential directions for making better recommendation results. First, it is worth exploring how to combine the features of products and items in the process of similarity computation to improve the performance of the model. In addition, we can import more context-based information including time and location to analyze the interests of people more precisely.

Acknowledgement

Shuting Zhang and Kechen Liu contributed equally to this work and should be considered co-first authors.

References

- [1] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, “Recommender System Application Developments: A Survey,” p. 38.
- [2] Rathi, P. Keni, and J. Sisodia, “Project Topic Recommendation by Analyzing User’s Interest Using Intelligent Conversational System,” in *Innovative Data Communication Technologies and Application*, Singapore, 2022, pp. 277–287. doi: 10.1007/978-981-16-7167-821.
- [3] “Collaborative Filtering: A Simple Introduction | Built In.” <https://builtin.com/data-science/collaborative-filtering-recommendersystem> (accessed Mar. 19, 2022).
- [4] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, ‘An Algorithmic Framework for Performing Collaborative Filtering’, *ACM SIGIR Forum*, vol. 51, no. 2, pp. 227–234, Aug. 2017, doi: 10.1145/3130348.3130372.

- [5] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, 'Item-based collaborative filtering recommendation algorithms', in Proceedings of the tenth international conference on World Wide Web - WWW '01, Hong Kong, Hong Kong, 2001, pp. 285–295. doi: 10.1145/371920.372071.
- [6] B. Smith and G. Linden, 'Two Decades of Recommender Systems at Amazon.com', IEEE Internet Computing, vol. 21, no. 3, pp. 12–18, 2017.
- [7] M. J. Pazzani and D. Billsus, 'Content-Based Recommendation Systems', in The Adaptive Web: Methods and Strategies of Web Personalization, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Heidelberg: Springer, 2007, pp. 325–341. doi: 10.1007/978-3-54072079-910.
- [8] B. Barragáns-Martínez, E. Costa-Montenegro, J. C. Burguillo, M. Rey-López, F. A. Mikic-Fonte, and A. Peleteiro, 'A hybrid contentbased and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition', Inf. Sci., vol. 180, no. 22, pp. 4290–4311, 2010, doi: 10.1016/j.ins.2010.07.024.
- [9] M. Balabanovic and Y. Shoham, 'Fab: content-based, collaborative' recommendation', Commun. ACM, vol. 40, no. 3, pp. 66–72, Mar. 1997, doi: 10.1145/245108.245124.
- [10] S. Rose, D. Engel, N. Cramer and W. Cowley, "Automatic keyword extraction from individual documents," applications and theory, p. 1, 20 1 2010.
- [11] textJ. Davidson, D. Liebal, J. Liu, P. Nandy and T. V. Vleet, "The YouTube Video Recommendation System," Proceedings of the fourth ACM conference on Recommender systems, pp. 293-294, 9 2010.
- [12] Liling, LIU. "Summary of Recommendation System Development." Journal of Physics: Conference Series, vol. 1187, Apr. 2019, p. 052044, <https://doi.org/10.1088/1742-6596/1187/5/052044>.
- [13] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowledge-Based Systems, 46 (2013) 109-132.
- [14] Team, "Recommendation Systems: Benefits And Development Process Issues," Azati: Uniting experts to fulfil important projects, Apr. 08, 2020. <https://azati.ai/recommendation-systems-benefits-and-issues/> (accessed Mar. 19, 2022).
- [15] "In2012B2016| Inflation Calculator,j [https : //www.in2013dollars.com/us/inflation/2012?endYear=2016&amount=1](https://www.in2013dollars.com/us/inflation/2012?endYear=2016&amount=1)(accessedMar.19,2022).
- [16] S. Gong, 'A collaborative filtering recommendation algorithm based on user clustering and item clustering.', J. Softw., vol. 5, no. 7, pp. 745–752, 2010.
- [17] X. He, Z. He, J. Song, Z. Liu, Y.-G. Jiang, and T.-S. Chua, 'NAIS: Neural Attentive Item Similarity Model for Recommendation', IEEE Transactions on Knowledge and Data Engineering, vol. 30, no. 12, pp. 2354–2366, 2018.
- [18] E. Christakopoulou and G. Karypis, 'Local Item-Item Models For TopN Recommendation', in Proceedings of the 10th ACM Conference on Recommender Systems, Boston Massachusetts USA, Sep. 2016, pp. 67–74.
- [19] C. B. Hensley, 'Selective Dissemination of Information (SDI): State of the Art in May, 1963', Proceedings of the May 21-23, 1963, Spring Joint Computer Conference, Detroit, Michigan, 1963, . 257–262.
- [20] M. Pazzani, J. Muramatsu, D. Billsus, 'Syskill Webert: Identifying interesting web sites', Proceedings of the Thirteenth National Conference on Artificial Intelligence, AAAI'96, 1996, . 54–61.
- [21] P. Lops, D. Jannach, C. Musto, T. Bogers, and M. Koolen, 'Trends in content-based recommendation', User Model. User-Adapt. Interact., vol. 29, no. 2, pp. 239–249, Apr. 2019, doi: 10.1007/s11257-01909231-w.
- [22] J. McAuley, C. Targett, Q. Shi, A. van den Hengel, 'Image-Based Recommendations on Styles and Substitutes', Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 2015, . 43–52.

- [23] S. Beliga, Keyword extraction: a review of methods and approaches, University of Rijeka, Department of Informatics, Rijeka, 2014.
- [24] L. Marujo, L. Wang, I. Trancoso, C. Dyer, A. W. Black, A. Gershman, D. M. d. Matos, J. P. Neto and J. Carbonell, "Automatic Keyword Extraction on Twitter," in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), Beijing, 2015.
- [25] Z. Li, D. Zhou, Y. Juan and J. Han, "Keyword Extraction for Social Snippets," in Proceedings of the 19th international conference on World wide web, 2010.
- [26] S. K. Bharti, K. S. Babu S. K. Jena, Automatic Keyword Extraction for Text Summarization: A Survey, arXiv preprint arXiv:1704.03242, 2017.
- [27] S. Rose, D. Engel, N. Cramer and W. Cowley, "Automatic keyword extraction from individual documents," Text Mining: Applications and Theory, p. 3, 2010.
- [28] Hulth, J. Karlgren, A. Jonsson, H. Boström and L. Asker, "Automatic Keyword Extraction Using Domain Knowledge," in International Conference on Intelligent Text Processing and Computational Linguistics, Springer, Berlin, Heidelberg, 2001.
- [29] iveBoy, "How to use RegEx in Python," Educative, Inc., [Online]. Available: <https://www.educative.io/edpresso/how-to-use-regex-in-python>. [Accessed 19 3 2022].
- [30] csurfer, "rake-nltk 1.0.6," python Software Foundation, [Online]. Available: <https://pypi.org/project/rake-nltk/>. [Accessed 19 3 2022].
- [31] N. Saxena, "Extracting Keyphrases from Text: RAKE and Gensim in Python," Medium, [Online]. Available: <https://towardsdatascience.com/extracting-keyphrases-from-textrake-and-gensim-in-python-eefd0fad582f>. [Accessed 19 3 2022].
- [32] R. van Meteren, 'Using Content-Based Filtering for Recommendation', 2000.
- [33] G. Shani and A. Gunawardana, 'Evaluating Recommendation Systems', in Recommender Systems Handbook, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston, MA: Springer US, 2011, pp. 257–297. doi: 10.1007/978-0-387-85820-38.