Mathematical and Computational Applications

MDPI

Article

A Fast Recommender System for Cold User Using Categorized Items

Hamid Jazayeriy [1], Saghi Mohammadi [2] and Shahaboddin Shamshirband [3,4,*]

[1] Department of Computer Engineering, Faculty of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Babol 47148-71167, Iran; Jhamid@nit.ac.ir

[2] Department of Computer Engineering, Faculty of Engineering, Azad University, Amol 61349-37333, Iran; saghimohamadi14@gmail.com

[3] Department for Management of Science and Technology Development, Ton Duc Thang University, Ho Chi Minh City, Vietnam

[4] Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam

[*] Correspondence: shahaboddin.shamshirband@tdt.edu.vn; Tel.: +6-014-626-6763

Received: 17 December 2017; Accepted: 14 January 2018; Published: 15 January 2018

Abstract: In recent years, recommender systems (RS) provide a considerable progress to users. RSs reduce the cost of a user's time in order to reach to desired results faster. The main issue of RSs is the presence of cold users which are less active and their preferences are more difficult to detect. The aim of this study is to provide a new way to improve recall and precision in recommender systems for cold users. According to the available categories of items, prioritization of the proposed items is improved and then presented to the cold user. The obtained results show that in addition to increased speed of processing, recall and precision have an acceptable improvement.

Keywords: recommender systems; collaborative filtering; k-nearest neighbor; cold user

1. Introduction

Nowadays, most people use the Internet and spend more time on social networks or e-commerce sites than in the past. The exponential growth of the amount of information on the Internet has made users face challenges in finding useful information [1–3]. Fortunately, studying the users' behaviors, their preferences can be analyzed. Recommender systems (RS) are used to do so [4]. Recommender systems adopt to provide recommendations to each user based on their activity, preferences, and behaviors which are consistent with the users' personal preferences and assist them in decision-making [5].

Recommender systems are implemented with three techniques including: content-based [6], knowledge-based [7], and collaborative filtering [8]. Collaborative filtering (CF) is one of the most commonly used techniques in RS [9–11]. In this study, we use collaborative filtering. In CF, opinions and ratings of similar users to the target user are used to provide recommendations. The target user is a user who should receive recommendations. In fact, the core of CF is to find the similarities between users. CF is possible in two ways: using memory-based and model-based approaches [12]; the combination of these two methods is also used [13,14]. The memory-based CF is used in this study. In the memory-based method, the similarity between users is calculated using one of three techniques of similarity algorithms [10,15], similarity measures [16–21], or heuristics methods [22]. Similarity measures are used this study. Then, a rating is predicted for the items that are not rated by the target user using rating prediction formulas. Finally, the items with the highest predicted ratings are recommended to the target user [10,17,23].

Cold starts are a challenging problem in collaborative filtering [15,24–26]. It arises in two cases: "cold user" and "cold item" cases. The first case happens when a new user registers on a web site and there is little information about the user. The final case happens when a new item is added to the system and there is no information about users rating on this item.

The aim of this study is to improve the output of the recommender systems with cold users. Cold users are those who have rated 2 to 20 items [4,17,24]. In this study, the proposed method improves the recall and precision of the recommender system with cold users. Although, the proposed method is much faster than the traditional recommender system, it only works with categorized items.

The structure of this article is as follows: In Section 2, some of recent studies related to similarity measures and prediction formulas in recommender systems are reviewed. Section 3 describes the methodology using categorized items and cold user in recommender systems. In Section 4, we simulate our proposed method and present the evaluation results. In Section 5, we discuss the conclusion and future directions for our work.

2. Related Work

In recent years, there have been studies which attempt to improve output of recommender systems [27–30]. The performance of recommender systems has a high impact on e-commerce [31–33].

In general, in a traditional recommender systems, proposing items to the target user is done in six steps [16,22,23,34]. The order of the steps is shown in Figure 1a. The first step is to calculate the similarity between users and the target user. In this study, the target users only include cold users. To find users similar to the cold user, there are many similarity measures [4,12–17]. Among the best of them is the cosine similarity measure [35], which is widely used to measure the similarity between users in CF. In cosine method, common rating among the users is considered to calculate the similarity between users. The equation for calculating the similarity between users using the cosine similarity is

$$sim(u,v)^{COS} = \frac{\sum_{p\in I}\left(r_{u,p} \times r_{v,p}\right)}{\sqrt{\sum_{p\in I} r_{u,p}^2} \times \sqrt{\sum_{p\in I} r_{v,p}^2}} \tag{1}$$

where p shows an item and I represents the set of common rating items by user u and v (user u is the cold user), respectively. $r_{u,p}$ and $r_{v,p}$ are the ratings that are given to item p by users u and v, respectively.

Another, newly introduced, similarity measure is NHSM composite [16]. This measure considers not only the local context information of the cold user ratings, but also the global preference of the behavior of this user. The local context considers the cold user ratings and the global context considers the ratings of similar user to the cold user. The NHSM measure is formed by a combination of three similarity measures of proximity-significance-singularity (PSS), modified Jaccard, and user rating preference (URP). NHSM is calculated as

$$Sim(u,v)^{NHSM} = Sim(u,v)^{JPSS} \times Sim(u,v)^{URP} \tag{2}$$

$JPSS$ has been created by PSS combined with modified Jaccard and is calculated as

$$Sim(u,v)^{JPSS} = Sim(u,v)^{PSS} \times Sim(u,v)^{Jaccard} \tag{3}$$

$$Sim(u,v)^{Jaccard} = \frac{|I_u \ \cap \ I_v|}{|I_u \ \cup \ I_v|} \tag{4}$$

where I_u and I_v are all the items rated by users u and v, respectively. PSS is the modified mode of similarity measures of proximity-impact-popularity (PIP). PSS is formed by a combination of three criteria of proximity-significance-singularity, which is calculated as

$$Sim(u,v)^{PSS} = \sum_{P \in I} PSS\ (r_{u,p}, r_{v,p}) \tag{5}$$

$$PSS\left(r_{u,p},\ r_{v,p}\right) = Proximity\ \left(r_{u,p},\ r_{v,p}\right) \times Significance\ \left(r_{u,p},\ r_{v,p}\right) \times Singularity\ \left(r_{u,p},\ r_{v,p}\right) \quad (6)$$

Proximity shows an arithmetic difference between two ratings. While, *Significance* shows the distance from the median rating.

$$Proximity\left(r_{u,p}, r_{v,p}\right) = 1 - \frac{1}{1 + exp\left(-\left|r_{u,p} - r_{v,p}\right|\right)} \quad (7)$$

$$\begin{aligned} Significance\left(r_{u,p}, r_{v,p}\right) &= \tfrac{1}{1+exp\left(-\left|r_{u,p}-r_{med}\right|\cdot\left|r_{v,p}-r_{med}\right|\right)} \\ r_{med} &= \tfrac{r_{max}+r_{min}}{2} \end{aligned} \quad (8)$$

where r_{med} indicates the median range of ratings in dataset. For example, if the ratings are in the range of 1 to 5, r_{med} = 3, and if the ratings are in the range of 1 to 7, r_{med} = 4. In the dataset used in this study, the ratings are in the range of 1 to 5. Thus, in these tests, r_{med} is always equal to 3.

$$Singularity\left(r_{u,p}, r_{v,p}\right) = 1 - \frac{1}{1 + exp\left(-\left|\frac{r_{u,p}+r_{v,p}}{2} - \mu_p\right|\right)} \quad (9)$$

where μ_p is the mean ratings given to item p by all users.

In the similarity measures of NHSM, different users' rating preferences are also considered. Because some users follow popular items, even if they have little interest in those items. To reflect the preferences of users, median and variance of the user's preferred model is calculated using the *URP* formula

$$\begin{aligned} Sim\ (u,v)^{URP} &= 1 - \tfrac{1}{1+exp\left(-\left|\mu_u-\mu_v\right|\times\left|\sigma_u-\sigma_v\right|\right)} \\ \mu_u = \tfrac{\sum_{p\in I_u} r_{u,p}}{\left|I_u\right|} \qquad \sigma_u &= \sqrt{\tfrac{\sum_{p\in I}\left(r_{u,p}-\overline{r_u}\right)^2}{\left|I_u\right|}} \end{aligned} \quad (10)$$

where μ_u and μ_v are the mean ratings of users u and v, respectively. σ_u and σ_v are the standard variances of users u and v, respectively. $\overline{r_u}$ and $\overline{r_v}$ are the total mean ratings of users u and v, respectively.

After calculating the similarity between the cold user and other users, in the second stage, some users with the most similarity to the cold user should be selected. k-nearest neighbor (kNN) is a popular algorithm to select k similar users.

In the third stage, a number of items that the cold user has not yet rated are chosen randomly. This number can be variable. For example, in the MovieLens dataset 500, 1000, 1500 and 2000 items may be selected randomly [28,36].

In the fourth stage, ratings are predicted for the random items of the third stage. Three standard formulas of (12), (13) or (14) are used usually to predict the ratings of the unrated items of the cold user. These formulas are known as the average, weighted sum (WS), and deviation from mean (DFM), respectively. Mentioned formulas are calculated as

$$G_{u,i} = \{v\ \in\ k_u\ |\exists\ i \quad v(i) \neq\ \varnothing\} \quad (11)$$

where k_u is a set of k users similar to the user u and $G_{u,i}$ indicates a set of users, v, who are similar to the cold user u that rated item i.

$$p_{u,i} = \frac{\sum_{v\epsilon G_{u,i}} r_{v,i}}{\#G_{u,i}} \ \Leftrightarrow\ G_{u,i} \neq \varnothing \quad (12)$$

$$p_{u,i} = \frac{\sum_{v\epsilon G_{u,i}} sim(u,v)\ r_{v,i}}{\sum_{n\epsilon G_{u,i}} sim(u,v)} \ \Leftrightarrow\ G_{u,i} \neq \varnothing \quad (13)$$

$$p_{u,i} = \overline{r_u} + \frac{\sum_{v\epsilon G_{u,i}} sim(u,v)\left(r_{v,i} - \overline{r_v}\right)}{\sum_{n\epsilon G_{u,i}}\ sim(u,v)} \ \Leftrightarrow\ G_{u,i} \neq \varnothing \quad (14)$$

All three of the above-mentioned formulas have very close responses. However, Formulas (13) and (14), are more suitable for dataset MovieLens [7,19]. In most of the related studies, Formula (14) or DFM is used for this dataset. Therefore, in our study, DFM is used to predict rating for all the unrated items of the cold user. In order to predict rating for item p, DFM uses the ratings given to item p by the similar users to the cold user [4,12,17].

In the fifth stage, items are arranged in a descending order based on predicted ratings in the previous stage.

Then in the final stage, $TopN$ number of items are selected from the list sorted which have the highest predicted ratings, and are recommended to the cold user.

The focus of our study is on methods of selecting items for recommendation to the cold user. In other words, some changes are made in the selection process of the proposed item. The similarity of our study to [16,37] is in using similarity measures of cosine and NHSM, but the difference is in the method of selection of items proposed for recommendation to the cold user.

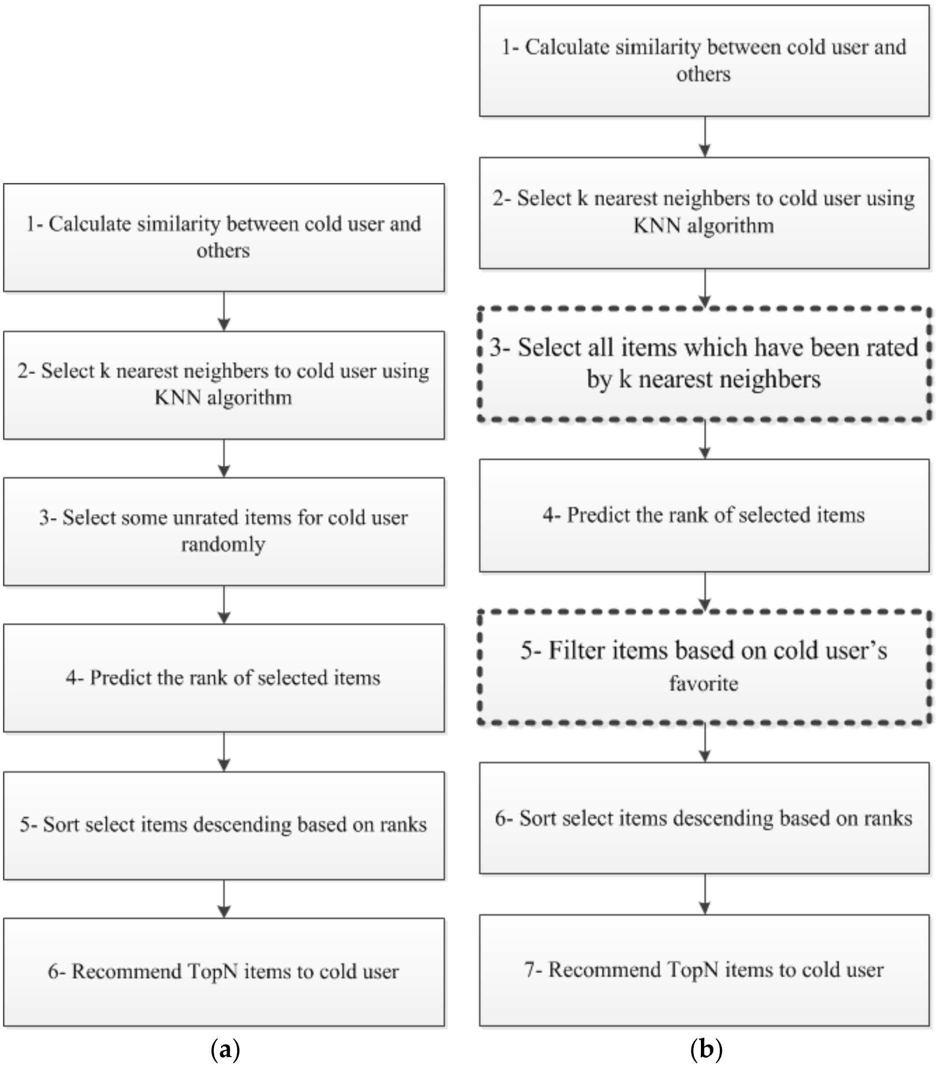

1- Calculate similarity between cold user and others
2- Select k nearest neighbors to cold user using KNN algorithm
3- Select some unrated items for cold user randomly
4- Predict the rank of selected items
5- Sort select items descending based on ranks
6- Recommend TopN items to cold user
(a)
1- Calculate similarity between cold user and others
2- Select k nearest neighbors to cold user using KNN algorithm
3- Select all items which have been rated by k nearest neighbors
4- Predict the rank of selected items
5- Filter items based on cold user's favorite
6- Sort select items descending based on ranks
7- Recommend TopN items to cold user
(b)

Figure 1. (**a**) Block diagram of a recommender system; (**b**) Block diagram of the proposed recommender system for cold users.

3. Proposed Method

In this study, we propose a new method for the selection of items for recommendation to the cold user. In this method, some changes have been made in the steps of a generic recommender system. The proposed method is shown in Figure 1b and the changed or added steps are identified by dotted box.

As shown in Figure 1b, the proposed method consists of seven steps. The differences between the proposed method and the generic recommender systems are only in the third and the fifth steps. In the third stage, the unrated items of the cold user are not selected randomly. However, all the items rated by k similar users to the cold user, which are not rated by the cold user, are selected for rating prediction.

By doing so, the likelihood that the cold user will be interested in those items increases. In fact, it makes the items appear to be more likely for recommendation to cold users to be selected for rating prediction.

Also, another advantage is the savings in processing time. This is because only the items that are rated by similar users to the cold user are selected for rating prediction. As a result, there is no need to predict a rating for 2000 or a specific number of random items.

The dataset used in this study is MovieLens. It is one of the most popular datasets in the field of recommender systems. In this dataset, movies belongs to categories (or genres) such as comedy, horror, drama, and so on. We thought that the categorized items may help to find favorite items for cold users.

Thus, in the fifth stage, it is checked that which of the items in the list obtained from stage four, have the cold user's favorite genres.

To detect the cold user's favorite genres, first, the average rating of the cold user, $\overline{r_u}$, is calculated. Then, the mean of the rating assigned to any genre by the cold user, $r_{u,G}$, is calculated. Then, all the genres are examined individually by the comparison

$$\begin{cases} \text{if } r_{u,G} \geq \overline{r_u} \Leftrightarrow & \text{user } u \text{ like genre } G \\ \text{otherwise} & \text{user } u \text{ dosen't like genre } G \end{cases}$$

where $r_{u,G}$ is rating given by cold user u to genre G.

By doing so, the genre whose mean rating is higher or equal to the total mean rating of the cold user is located in the cold user's favorite list. Also, if the rating is less than the total mean rating of the cold user, it indicates a lack of user interest in the genre.

After the cold user's favorite genres are determined, a list is provided from the items rated in the fourth stage, having one or more favorite genres of the cold user. The new list is arranged in descending order based on the predicted ratings in stage four. Finally, a $TopN$ number of items from the beginning of the new list is recommended to the cold user.

In the proposed method, it has been tried to recommend items to the cold user that are rated by similar users and also have the cold user's favorite genres. The results obtained indicate that this will improve the quality of recommendations.

4. Implementation and Evaluation of the Results

In this study, MovieLens is used as the standard dataset to examine the quality of the proposed method in recommender systems [38]. It contains user recommendations about movies over a period of seven months in 1997. This dataset exists in three different sizes (100 KB, 1 MB, and 10 MB ratings). In this study, 1 MB dataset is used for testing [24,39]. MovieLens dataset 1 M has 1,000,209 ratings on 3883 films by 6040 users. In experiments conducted in this study, the dataset is divided into three parts of validation, testing, and training data [12].

Validation data and testing data include only cold users. In fact, the users of validation data and testing data are the same and only the items rated by users in these two datasets are different. The datasets are divided in a way that all cold users of MovieLens dataset will be placed in the validation data. In the MovieLens dataset, the minimum number of users' ratings is 20 items, so there are no cold users. To create cold users in this dataset, the following steps can be followed [1,12]:

First, all users who rated between 20–30 items are located in validation data. The number of users is equal to 809. Then, randomly, 5 to 20 rated items are located in testing data and are deleted from validation data. It is again examined that no user in the validation data has rated more than 20 and less than 2 items. Doing so, all 809 users of the validation dataset will be cold users.

The training data is the remaining part of the total MovieLens dataset. In fact, all remaining users of the total MovieLens dataset, who are not cold users, are in training data. In this study, the number of users in training data is 5231.

In this study, data are presented by vector or matrix data structure and processed in MATLAB software. The *top-n* is set to 100 in this study. All experiments were conducted on a computer with Intel core i3, 4 GB RAM, and 1TB HDD.

4.1. Selecting k Similar Users

At this stage, using similarity measures mentioned in the previous sections, for each cold user in the validation dataset, the similarity of the cold user to the users of training data is calculated. Then, using the kNN algorithm, k more similar users to the cold user are selected.

4.2. Selecting Items for Recommendation

After finding the most similar users to the cold user, the k similar users' rated items are used to provide recommendations to the cold user. In fact, at this stage, an attempt is made to offer the user's favorite items. This reduces the users' time cost spent searching for items of interest.

4.3. Evaluation Criteria

Various parameters can be evaluated in RS. However, recall and precision are the most important parameters usually evaluated. These criteria are used in this study to assess the quality of the proposed method. The processing time of the proposed method has also been evaluated.

The Recall criterion is the mean ratio of items from testing data that can be observed in the rated list of training data [40]. Recall criterion is calculated as

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

In the equation above, TP is the number of cold user's favourite items that are recommended correctly, whereas FN is the number of cold user's favourite items that are not recommended.

The Precision criterion is ratio of the recommended items to the items that, in fact test data users are really interested in to those items [22]. The Precision criterion is calculated as

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

where FP is the number of inacceptable items for cold user which have wrongly been recommended to the user.

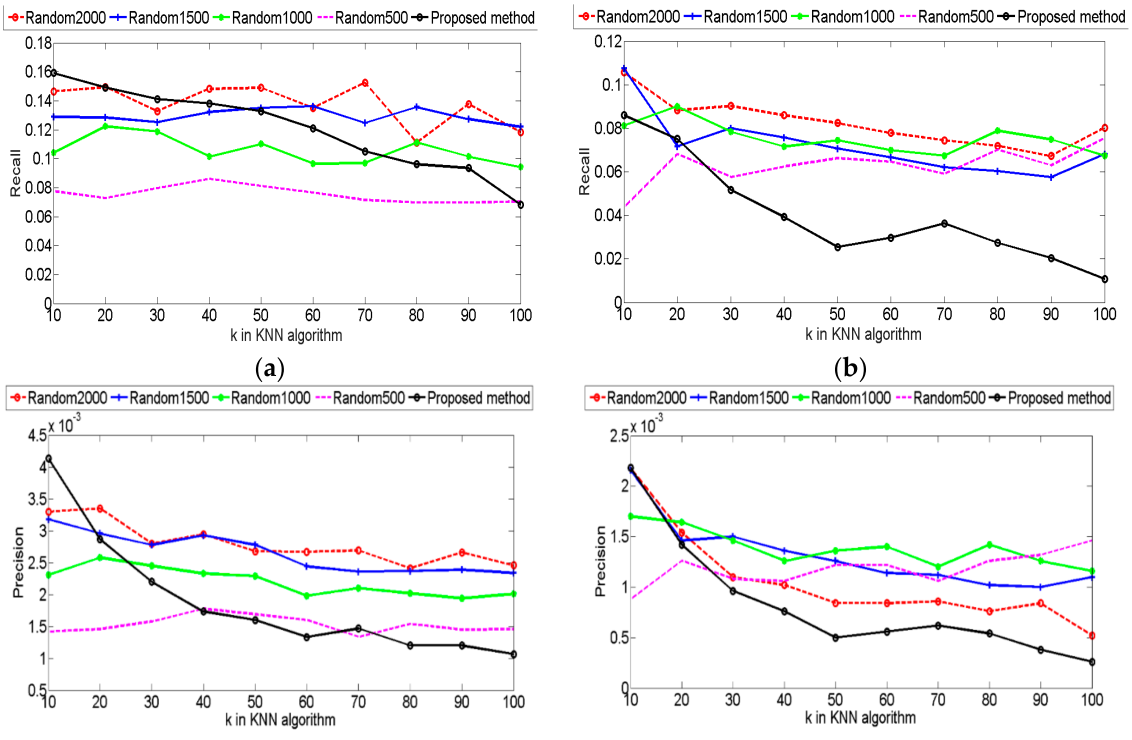
The other criteria evaluated is the processing time. The purpose of processing time in this study is the time of the selection of the proposed item, which is done after calculating the similarity between users. In fact, the processing time of the second stage to the last stage is calculated in Figure 1a,b. It worth mentioning that the first stage in traditional RS (Figure 1a) and proposed RS for cold users (Figure 1b) are the same.

4.4. Experiments

To evaluate the efficiency of the proposed method, a series of experiments is conducted. In the following experiment, the DFM method (Formula (14)) has been used to predict the ratings of the non-rated items.

As it can be seen in step 3 of Figure 1b, item selection in the proposed method is based on *k* similar users some while in traditional recommender systems, Figure 1a, random selection is used. The first experiment evaluates the effect of the step 3 in proposed method without considering the cold users' favorite genre.

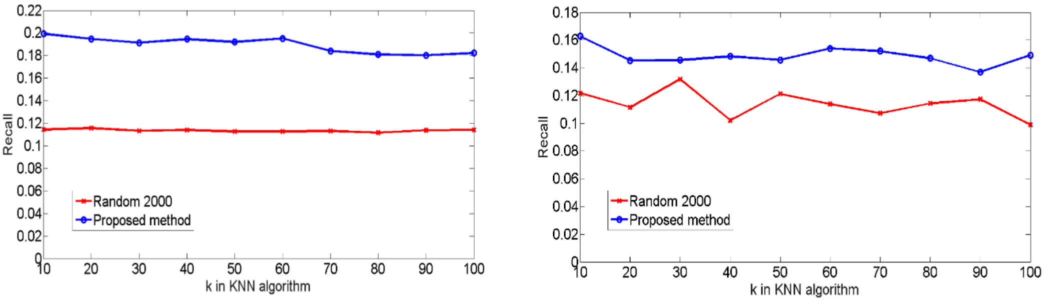
Figure 2 shows the recall and precision of proposed and tradition recommender systems. It can be seen that traditional recommender systems with random selection have better recall and precision. Therefore, so far, the proposed method without considering the cold user's favorite genre is not efficient. Moreover, this experiment shows that having more random items improves recall and precision. Therefore, from now on, we just consider 2000 random items. Another observation is that, the cosine similarity (Formula (1)) has a better result than NHSM (Formula (2)).



(**a**) (**b**) (**c**) (**d**)

Figure 2. (**a**) Recall criterion with similarity measures of cosine; (**b**) Recall criterion with similarity measures of NHSM; (**c**) Precision criterion with similarity measures of cosine; (**d**) Precision criterion with similarity measures of NHSM.

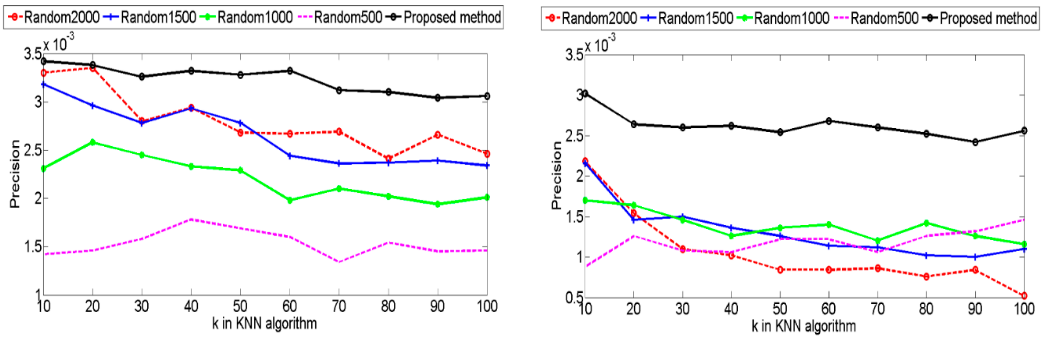
In the next experiment, we want to evaluate the effect of the users' favorite genres between the traditional system and the proposed system (step 5 in Figure 1b). To do so, we have considered the cold users' favorite genres to filter items and rank them. Figure 3 shows the results. It can be seen that the proposed method has better recall than traditional recommender system in both cosine and NHSM similarity. It shows the amount of recall is dramatically improved using the items of the *k* similar users to the cold user. Another observation is that, in our proposed method, the changes on *k* value does not have too much effect on the recall criterion.



(**a**) (**b**)

Figure 3. (**a**) Recall criterion with similarity measures of cosine; (**b**) Recall criterion with similarity measures of NHSM.

Figure 4 shows the results of the precision criterion when comparing the proposed method and the tradition recommender systems while considering the category of items. It can be seen that the proposed has higher precision with both cosine and NHSM similarity. By considering the favorite genre of the cold users, the precision criterion of the proposed recommendation has improved dramatically. Another observation is that cosine similarity has higher precision than NHSM and NHSM similarity is very sensitive to the number of random items in traditional recommender systems.



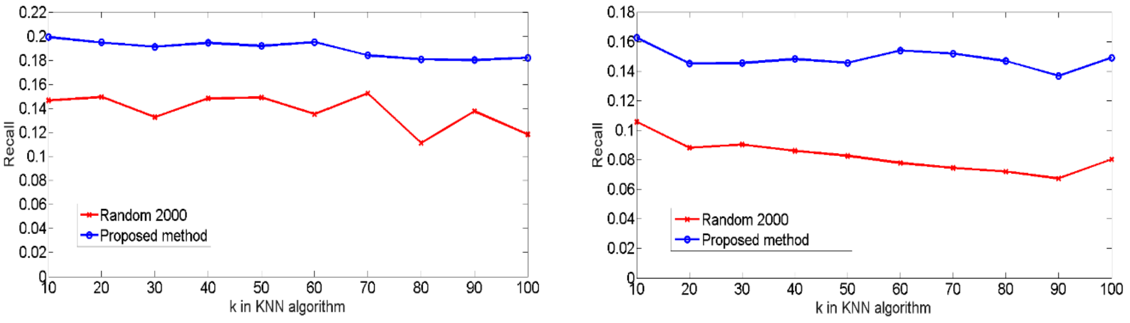
(**a**) (**b**)

Figure 4. (**a**) Precision criterion with similarity measures of cosine; (**b**) Precision criterion with similarity measures of NHSM.

Therefore, this experiment shows that considering categorized items does not have too much of an effect on traditional recommender systems, while it can dramatically improve the recall and precision of the proposed method.

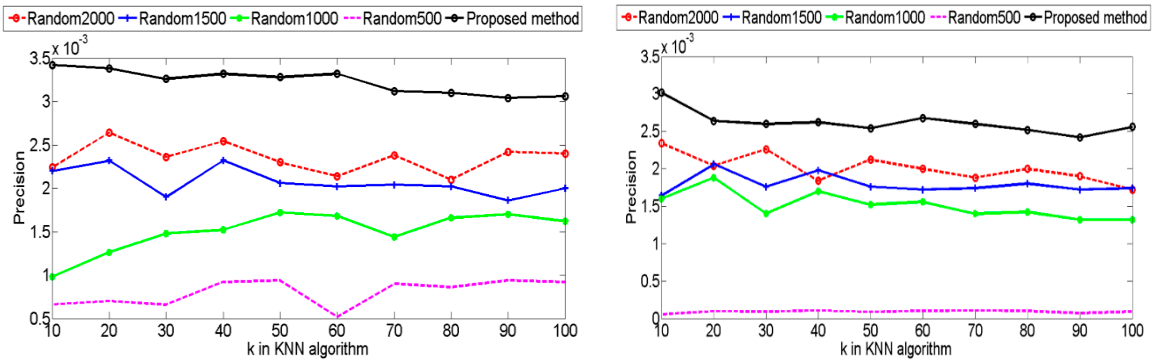
In the final experiment, we consider the effect of items from k nearest users (step 3 of Figure 1b) together with favourite category of the cold users (step 5 of Figure 1b) and compare it with a traditional recommender system (Figure 1a).

Figure 5 shows the recall criterion. It can be seen that the proposed method has higher recall than the traditional recommender system for cold users. Figure 6 shows the precision criterion. It can be seen that the proposed method has higher precision than the traditional recommender system for cold users.



(**a**) (**b**)

Figure 5. (**a**) Recall criterion with similarity measures of cosine; (**b**) Recall criterion with similarity measures of NHSM.



(**a**) (**b**)

Figure 6. (**a**) Precision criterion with similarity measures of cosine; (**b**) Precision criterion with similarity measures of NHSM.

These experiments show that the traditional recommender system has higher recall when no filter is provided for categories while having categorized items may improve its precision.

The processing time in the proposed method is less than the traditional recommender systems. The reason is that, in step 3, the number of items selected from *k* nearest users is less than the traditional recommender system. Therefore, in the following steps less time is consumed. Figure 7 show the processing time of the proposed method and traditional recommender system with 500, 1000, 1500 and 2000 random items. As it is shown in this figure, the proposed method is many times faster than a traditional recommender system. It is almost six times faster than a traditional recommender system with 2000 random items.

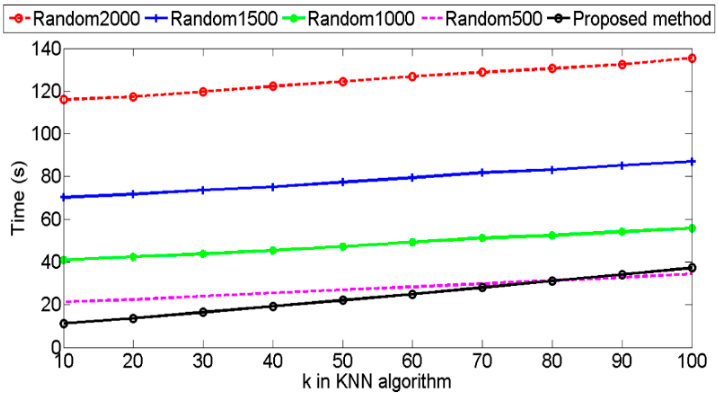


Figure 7. Processing time of the random methods and the proposed method.

5. Conclusions

Although the similarity measures have a great impact on collaborative filtering in recommender systems, this study shows that having categorized items can improve the recall and precision of the recommender system for cold users.

Although having categorized items improves the recall and precision of the proposed method for cold users, it may have a negative effect on the recall criterion of the traditional recommender systems. The processing time of the proposed method is also much less than traditional recommender systems. In this study, we have used the MovieLens dataset which is already categorized. However, categorizing of items is still a problem in recommender systems. Finding a fast and effective method for categorizing items can be considered for future work.

Author Contributions: Hamid Jazayeriy, Saghi Mohammadi and Shahaboddin Shamshirband had contribution in designed and analysis of the experiments; Saghi Mohammadi performed the experiments. Hamid Jazayeriy and Shahaboddin Shamshirband revised the paper in term of writing.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Baker, T.; Mackay, M.; Randles, M.; Taleb-Bendiab, A. Intention-oriented programming support for runtime adaptive autonomic cloud-based applications. *Comput. Electr. Eng.* **2013**, *39*, 2400–2412. [CrossRef]
2. Baker, T.; Taleb-Bendiab, A.; Randles, M. Auditable intention-oriented Web applications using PAA auditing/accounting paradigm. In Proceedings of the 2009 Conference on Techniques and Applications for Mobile Commerce (TAMoCo 2009), Mérida, Spain, 16–17 September 2009; IOS Press: Tepper Drive Clifton, VA, USA, 2009.
3. Karam, Y.; Baker, T.; Taleb-Bendiab, A. Intention-oriented modelling support for socio-technical driven elastic cloud applications. In Proceedings of the IEEE International Conference on Innovations in Information Technology (IIT), Abu Dhabi, United Arab Emirates, 18–20 March 2012.
4. Bobadilla, J.; Ortega, F.; Hernando, A.; Gutirrez, A. Recommender systems survey. *Knowl. Based Syst.* **2013**, *46*, 109–132. [CrossRef]
5. Chesnevar, C.I.; Maguitman, A.G.; Simari, G.R. A first approach to argument-based recommender systems based on defeasible logic programming. In Proceedings of the 10th International Workshop on Non-Monotonic Reasoning (NMR 2004), Whistler, BC, Canada, 6–8 June 2004; pp. 109–117.
6. Pazzani, J.M.; Billsus, D. Content-Based Recommendation Systems. In *The Adaptive Web*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 325–341.
7. Lu, J.; Shambour, Q.; Xu, Y.; Lin, Q.; Zhang, G. BizSeeker: A hybrid semantic recommendation system for personalized government-to-business e-services. *Internet Res.* **2010**, *20*, 342–365. [CrossRef]
8. Breese, S.J.; Heckerman, D.; Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, Madison, WI, USA, 24–26 July 1998; pp. 43–52.
9. Kim, H.-N.; Ji, A.-T.; Ha, I.; Jo, G.-S. Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation. *Electron. Commer. Res. Appl.* **2010**, *9*, 73–83. [CrossRef]
10. Bobadilla, J.; Hernando, A.; Ortega, F.; Bernal, J. A framework for collaborative filtering recommender systems. *Expert Syst. Appl.* **2011**, *38*, 14609–14623. [CrossRef]
11. Bobadilla, J.S.; Ortega, F.; Hernando, A.; Bernal, J.S. Generalization of recommender systems: Collaborative filtering extended to groups of users and restricted to groups of items. *Expert Syst. Appl.* **2012**, *39*, 172–186. [CrossRef]
12. Cacheda, F.; Carneiro, V.; Fernandez, D.; Formoso, V. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Trans. Web* **2011**, *5*. [CrossRef]
13. Koren, Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008.

14. Pennock, D.M.; Horvitz, E.; Lawrence, S.; Giles, C.L. Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach. In Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, Stanford, CA, USA, 30 June–3 July 2000; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2000.
15. Kim, H.-N.; El-Saddik, A.; Jo, G.-S. Collaborative error-reflected models for cold-start recommender systems. *Decis. Support Syst.* **2011**, *51*, 519–531. [CrossRef]
16. Liu, H.; Hu, Z.; Mian, A.; Tian, H.; Zhu, X. A new user similarity model to improve the accuracy of collaborative filtering. *Knowl. Based Syst.* **2014**, *56*, 156–166. [CrossRef]
17. Bobadilla, J.S.; Ortega, F.; Hernando, A.; Bernal, J.S. A collaborative filtering approach to mitigate the new user cold start problem. *Knowl. Based Syst.* **2012**, *26*, 225–238. [CrossRef]
18. Herlocker, L.J.; Konstan, A.J.; Terveen, G.L.; Riedl, T.J. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* **2004**, *22*, 5–53. [CrossRef]
19. Shardanand, U.; Maes, P. Social information filtering: Algorithms for automating "word of mouth". In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95), Denver, CO, USA, 7–11 May 1995; ACM Press: New York, NY, USA, 1995.
20. Koutrika, G.; Bercovitz, B.; Garcia-Molina, H. FlexRecs: Expressing and combining flexible recommendations. In Proceedings of the 35th SIGMOD International Conference on Management of Data (SIGMOD '09), Providence, RI, USA, 29 June–2 July 2009; ACM Press: New York, NY, USA, 2009.
21. Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P.; Riedl, J. GroupLens: An open architecture for collaborative filtering of netnews. In Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, Chapel Hill, NC, USA, 22–26 October 1994; ACM Press: New York, NY, USA, 1994.
22. Ahn, H.J. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Inf. Sci.* **2008**, *178*, 37–51. [CrossRef]
23. Bobadilla, J.S.; Ortega, F.; Hernando, A. A collaborative filtering similarity measure based on singularities. *Inf. Process. Manag.* **2012**, *48*, 204–217. [CrossRef]
24. Patra, K.B.; Launonen, R.; Ollikainen, V.; Nandi, S. A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data. *Knowl. Based Syst.* **2015**, *82*, 163–177. [CrossRef]
25. Leung, W.-K.C.; Chan, C.-F.S.; Chung, F.-L. An empirical study of a cross-level association rule mining approach to cold-start recommendations. *Knowl. Based Syst.* **2008**, *21*, 515–529. [CrossRef]
26. Zhang, M.; Tang, J.; Zhang, X.; Xue, X. Addressing cold start in recommender systems: A semi-supervised co-training algorithm. In Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '14), Gold Coast, Australia, 6–11 July 2014; ACM Press: New York, NY, USA, 2004.
27. Zheng, X.; Da Xu, L.; Chai, S. QoS Recommendation in Cloud Services. *IEEE Access* **2017**, *5*, 5171–5177. [CrossRef]
28. Cremonesi, P.; Koren, Y.; Turrin, R. Performance of recommender algorithms on *top-n* recommendation tasks. In Proceedings of the Fourth ACM Conference on Recommender Systems, Barcelona, Spain, 26–30 September 2010; ACM Press: New York, NY, USA, 2010.
29. Massa, P.; Avesani, P. Trust-aware recommender systems. In Proceedings of the 2007 ACM Conference on Recommender Systems, Minneapolis, MN, USA, 19–20 October 2007; ACM Press: New York, NY, USA, 2007.
30. Rosaci, D.; Sarné, G.M.; Garruzzo, S. MUADDIB: A distributed recommender system supporting device adaptivity. *ACM Trans. Inf. Syst.* **2009**, *27*. [CrossRef]
31. Brodén, B.; Hammar, M.; Nilsson, B.J.; Paraschakis, D. Bandit Algorithms for e-Commerce Recommender Systems. In Proceedings of the Eleventh ACM Conference on Recommender Systems, Como, Italy, 27–31 August 2017; ACM Press: New York, NY, USA, 2017.
32. Jazayeriy, H.; Azmi-Murad, M.; Sulaiman, N.; Udzir, N.I. Generating Pareto-Optimal Offers in Bilateral Automated Negotiation with One-Side Uncertain Importance Weights. *Comput. Inf.* **2013**, *31*, 1235–1253.
33. Palopoli, L.; Rosaci, D.; Sarné, G.M. A distributed and multi-tiered software architecture for assessing e-Commerce recommendations. *Concurr. Comput. Pract. Exp.* **2016**, *28*, 4507–4531. [CrossRef]
34. Pirasteh, P.; Hwang, D.; Jung, J.J. Exploiting matrix factorization to asymmetric user similarities in recommendation systems. *Knowl. Based Syst.* **2015**, *83*, 51–57. [CrossRef]
35. Adomavicius, G.; Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 734–749. [CrossRef]

36. Rossetti, M.; Stella, F.; Zanker, M. Contrasting offline and online results when evaluating recommendation algorithms. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016.
37. Xie, F.; Chen, Z.; Shang, J.; Fox, C.G. Grey Forecast model for accurate recommendation in presence of data sparsity and correlation. *Knowl. Based Syst.* **2014**, *69*, 179–190. [CrossRef]
38. Tsai, C.-F.; Hung, C. Cluster ensembles in collaborative filtering recommendation. *Appl. Soft Comput.* **2012**, *12*, 1417–1425. [CrossRef]
39. Da Silva, Q.E.; Camilo-Junior, G.C.; Pascoal, L.L.M.; Rosa, C.T. An evolutionary approach for combining results of recommender systems techniques based on collaborative filtering. *Expert Syst. Appl.* **2016**, *53*, 204–218. [CrossRef]
40. Fouss, F.; Pirotte, A.; Renders, J.-M.; Saerens, M. Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation. *IEEE Trans. Knowl. Data Eng.* **2007**, *19*, 355–369. [CrossRef]