An Efficient Cold Start Solution Based on Group Interests for Recommender Systems

Bilal Hawashin
Department of Computer Information Systems
Alzaytoonah University of Jordan,
Amman, Jordan
b.hawashin@zuj.edu.jo

Ayman Mansour
Department of Communication and Computer Engineering
Tafila Technical University
Tafila, Jordan
mansour@ttu.edu.jo

Tarek Kanan
Department of Computer Science
Alzaytoonah University of Jordan,
Amman, Jordan
tarek.kanan@zuj.edu.jo

Farshad Fotouhi
Department of Computer Science
Wayne State University
Detroit, MI, USA
fotouhi@wayne.edu

ABSTRACT

This paper proposes an efficient solution for the cold start problem in recommender systems. This problem occurs with new users who do not have sufficient information in their records. This would cause the recommender system to fail in providing recommendations to these users. This problem is one of the common and important problems in recommender systems. Although some solutions have been proposed to solve it in the literature, these solutions would not work properly in some scenarios because they do not concentrate on finding the actual interests of the users and the hidden motives behind their behavior. Our proposed solution uses the hidden interests of the group to which the target user belongs to provide recommendations for that user. The experiments show that our proposed solution is efficient in terms of searching time and space consumption.

CCS Concepts

• **Artificial Intelligence, Machine Learning, Natural Language Processing, Feature Selection.**

Keywords

User Interest, Cold Start Problem, Content Based Filtering, Group Interest, Recommender Systems, Machine Learning.

1. INTRODUCTION

Recommender Systems are used to suggest items to users based on their interests. They have been used in various domains such as research papers recommenders, book recommenders, product recommenders, and many more. In this paper, our concentration is on movie recommenders, whereas the items are movies and the users are online store clients. In this paper, the terms movie and item are used interchangeably.

In order to provide recommendations, user-item rates are used. When a user buys an item, the recommender system asks the user to rate the item on a scale, commonly from one to five, one if the user found the item not interesting at all and five if the user found the item very interesting. Later, the recommender system predicts the user rates on non-rated items using existing rates and/or other supporting information. Finally, when a user login to the store, the recommender system would recommend the items whose predicted rates are the highest for that user.

One example of recommender systems can be found in Amazon website. The recommender system would ask the user who already bought an item to rate that item. Based on these rates, the recommender system provides suggestions to the user whenever he visits any page. These recommendations are given under the following statements: "Customers who bought this bought also", "Recommended for you", "Frequently bought together", and so on. Figure1 depicts an example of such recommendations.

One of the commonly known problems in recommender systems is the cold start problem, which occurs with new users who do not have sufficient rating history. In this case, the widely used user-user similarity would fail. While many solutions have been proposed to solve this issue [13][15][20][29], these solutions did not concentrate on extracting the actual user interests and the hidden motives behind their actions. For example, suppose that our target user *X* is a young female user. Assume that most similar user to *X* is user *Y* , who highly rated the movie "Mission Impossible". Using the widely used Item-Item similarity method would falsely assume that the user *Y* is interested in action movies. However, the user *Y* could have highly rated the movie not because of its genre but because of his main actor Mill Gibson. Such factor, although important, would be ignored in the commonly used item-item similarity when other factors such as movie genre and movie plot exist. This would lead to the false recommendation of action movies to our target user *X*. From this example, we can conclude that it is necessary to find the real interests and the actual motivations behind the user rates.

On the other hand, many existing solutions concentrate on finding the most similar users to the target user. This would not

DATA '18, October 1–2, 2018, Madrid, Spain © 2018 Association for Computing Machinery. ACM ISBN 978-1-4503-6536-9/18/10…$15.00
https://doi.org/10.1145/3279996.3280022

provide correct recommendations in some cases. For example, suppose that the most similar user to the target user highly rated animation movies, while the majority of the other similar users highly rated action movies. In this case, the most similar user does not represent the general interest of the user group. This user would contribute more in the final recommendations even though it could be a noisy user. Therefore, it would be more appropriate to consider the general interest of the group to which the user belongs instead of placing high importance on the interests of certain individuals.

In this paper, the authors propose a solution to the cold start problem that uses the actual interests of the group to which the target user belongs. First, it finds the users that belong to the same group like the target user. Next, it extracts the interests of this group. As the rating history of the target user could be insufficient, contextual data is used instead to find similar users. It should be noted that the proposed solution is based on extracting the hidden interests of users, which is done using the User Interest Extractor[10]. In order to evaluate this method, the authors used an extended subset of MovieLens. Unfortunately, the current version of the MovieLens does not provide actor/acress names, director name, country, and other important information. Therefore, the authors extended the item descriptions of a subset of items. To evaluate the method, the authors used the searching time and space consumption. This work is a preliminary work that insists on the importance of considering the actual user interests to increase the recommender system accuracy. The contributions of this work are as follows.

- Proposing an efficient solution to the cold start problem.
- Insisting on the importance of considering the hidden user interests and group interests in recommender systems.

The advantage of using this method lies in the increase in the recommendation accuracy by concentrating on the hidden user interests and behavior instead of merely concentrating on item-item or user-user similarity, which could be misleading. Furthermore, the accuracy is increased by concentrating on the group instead of individuals, which would prevent noisy users from affecting the results.

The rest of this paper is organized as follows. Section 2 is a literature review of the related works in this field. Section 3 describes our proposed method. Section 4 is the experimental part. Section 5 is the discussion, and Section 6 provides the conclusions and the future work.

2. LITERATURE REVIEW

Many works have studied recommender systems. Content based filters[18], [19], [23] recommend items based on its description similarity to the previously highly rated items by the user. In details, [18], [19] used Bayesian classifiers to estimate the probability that a user likes an item based on its content, while [23] used a threshold to decide whether the description match that of the highly rated items or not. Collaborative filtering [5], [17], [22], [24] on the other hand, recommend items that were highly rated by similar users. In details, Memory based methods [22], [24] in collaborative filtering use the previously rated items in finding similar users, in contrast with model based algorithms[5], [17] that learn a model from the previous rates, such as Baysian model [5] and maximum entropy model [17]. Hybrid methods [26], [27] combine both content and collaborative features together. Context aware recommender systems [1], [6], [28] are those that consider context information such as location [28] and time [6] in their recommendations. [10] proposed the extract of user interests from their rates, while [11] proposed a method to extract the interests of a group of users.

As for the cold start problem, many solutions in the literature have been proposed to solve it. Examples include [29] which proposed bi-clustering and fusion (BiFu) under cloud computing setting. They also used popular items and frequent users in their attempt to identify the rating sources for recommendations. [20] used three social factors, mainly personal interest, interpersonal interest similarity, and interpersonal influence in order to create a recommendation model based on probabilistic matrix factorization. Both interpersonal interest similarity and interpersonal influence were used for cold start users. [15] used collaborative-filtering recommender system using interest expansion via personalized ranking named iExpand. [13] proposed a Bayesian model that links collaborative filtering with topic model. They used the review text information to solve the cold start problem. [25] was among the early works who tried to propose a solution for this problem. They proposed to use the ascpect model latent variable method for cold start recommending. This model combines both collaborative and content information. Furthermore, they proposed the use of CROC curve as an evaluation measurement. They suggested to use heuristic recommenders with CROC curves. [8] stated that many previously proposed solutions did not concentrate on the privacy issue in their proposed solutions, and they proposed a privacy protected cold start solution. In their work, they suggested two types of recommendations; node recommendations and batch recommendations, and they compared their work with three existing methods.; Triadic Aspect Method, Naïve Filterbot Method, and MediaScout Stereotype Method. Other recommender methods include [2][4][7][9][12][14][16][21]

Clearly, while many solutions have been proposed to solve the cold start problem, they did not take into consideration the hidden user interests, and they placed more importance on the interests of the individuals instead of the group interests, which would affect the accuracy of the recommendations.

3. THE PROPOSED COLD START SOLUTION

Our solution to the cold start problem is the following. First, the domain needs to specify the set of factors that affect the user interests. For example, for the domain of movies, user interests could be affected by the user age, gender, time of the year, marital status, country, job, and so on. These factors are domain dependent and we leave the specification of these factors to the domain. Such factors would provide one layer of abstraction and eliminate some noisy similarities. For example, the user of age 14 is not that different from the user of age 16. They should be considered similar as they belong to the same age group, even though they are considered different using the traditional similarities. After specifying these factors, and upon creating an account, each user needs to enter his information. Next, the algorithm finds similar users to this user based on these factors. As previously mentioned, each domain can use its own factors and its own values for each factor. In this work, the authors used the following set of factors and values for an online movie store.

- Age: Children, Young, Adult, and Senior.
- Gender: Male or Female
- Time of the year: Each month is a value.
- Marital Status: Single, Married.

As an implementation, the authors used a bitmap matrix, which proved its efficiency in the literature in many fields. The bitmap description is explained as follows. When a user creates an account, and after providing his personal information, (s)he will be represented as a vector of *n* digits, where *n* is the number of values in all the previously mentioned factors. Therefore, if the total number of values in all factors is 10, each user vector would contain 10 digits. In each user vector, the value of the digit would be 1 if the user belongs to this value, otherwise, it would be 0. Later, when the user rates an item, the vector of the user, the item vector, and the rate are inserted in a table, Enhanced_R. It should be noted that the item vector has the frequencies of terms in that item.

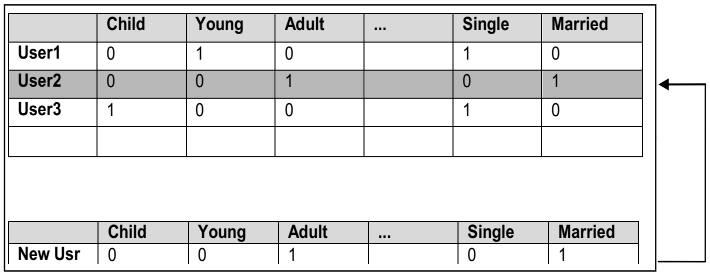
Finally, when a new user creates an account, and after filling his information, the vector of that user is used to search for identical or similar users, as depicted in Figure 1, and both item vectors and rates of these users are extracted and passed to the User Interest Extractor, explained in [10], to extract the top interests of these users as a group. Finally, the top interests are used to search for items, and these items are suggested to the new user. In this work, the authors concentrate on extracting identical users. Extracting similar users is left to future work. Our proposed solution to the cold start problem is presented in Algorithm 1.

4. EXPERIMENTS AND RESULTS

4.1 Dataset

One of the widely used datasets to evaluate recommender systems is MovieLens, which includes the rates of online customers to a set of movies. This dataset has been used extensively in the literature, and therefore, we adopted this dataset for our experiments. Unfortunately, MovieLens does not include many important data in the movie description that could reflect the real user interests. In order to evaluate the recommender system using the new method, the authors used an expanded subset of items. A detailed description of the dataset is given next.

The Expanded1 MovieLens dataset has the same domain of MovieLens dataset, i.e. movies. It is composed of 1000 ratings of 10 users and 100 movies, as illustrated in Table 1. Item description file has the descriptions of movies. Each description contains the movie title, the year, the genre(multiple), the main actor, the main actress, the director, and the country of production. As an example, the description of the movie Toy Story is the following. " Toy Story , 1995, Jan , animation , children, comedy, TomHanks, TimAllen, JohnLasseter, USA".



	Child	Young	Adult	...	Single	Married
User1	0	1	0		1	0
User2	0	0	1		0	1
User3	1	0	0		1	0

	Child	Young	Adult	...	Single	Married
New Usr	0	0	1		0	1

Figure 1: Finding similar users to a new user based on profile similarity.

```
Algorithm 1: COLD START SOLUTION
Input: The vector V representing a new user U.
       Desired Number of recommendations K.
Output: The top K recommendations for user U.
Algorithm:
Find records of identical/similar users to user U using the
    table Enhanced_R.
For each record, extract both the item vector and the label for
    each rate and add them to S.
Pass S and K to User Interest Extractor to
    find top  interests.
```

Table 1. Datasets

Dataset	Number of Ratings	Number of Users	Number of Items
Expanded1	1000	10	100

4.2 Experimental Settings

For our experiments, we used an Intel® Xeon® server of 3.16GHz CPU and 8GB RAM, with Microsoft Windows Server Operating System. Also, we used Microsoft Visual Studio 6.0 to read the dataset and execute the methods.

4.3 Evaluation Metrics

In order to evaluate the recommender system, we used Searching Time and Space consumption. They are defined as follows.

- Searching Time: it is the time used to find users who are identical/similar to the target user.
- Space Consumption: The space needed to represent the users using the bitmap method.

4.4 Experimental Results

As for the searching time, the authors used various lengths of bitmap vectors per user and various number of users. The searching time was negligible and less than one second in the worst case. Apparently, the time required to find identical/similar users to the new user is O(n), were n is the number of users. Table 2 gives the search time using various number of users and bitmap sizes.

Table 2. The searching time using various number of users and bitmap sizes.

Number of Users	Bitmap Size(Bit)	Searching Time(Second)
3000	20	2
3000	40	2
6000	15	4
6000	30	4
12000	10	8
12000	20	8

As for the space consumption, it was an issue when both the bitmap length and the number of users were large. Table 3 illustrates the maximum number of values allowed for various number of users using our system settings. It should be noted that many domains do not need large bitmap size. For example, 20 bit bitmap size is sufficient for many domains, and such bitmap size can handle relatively large number of users.

Table 3. The maximum number of Bitmap values allowed using various number of users using our system settings.

Bitmap Size	Max Users Allowed
20	12900
30	8700
40	6000

5. DISCUSSION

From the previous experiments, it was clear that the proposed solution is efficient in term of time. As for the space, such solution would be efficient for many domains. The domains that need larger bitmap size and larger number of users can adopt various hardware and software solutions. As for the hardware solutions, increasing the main memory would provide more capability to the system with relatively low cost. As for the software solutions, many space-efficient data structures can be used. For example, in many scenarios, only one value in the factor is applicable to the user. A user can be a male or a female, a child, a young, an adult, or a senior, and so on. Therefore, there is no need to reserve space for all the values. A linked-list can be used for this sake instead of a bitmap array, whereas only the applicable values are stored in the nodes. This software solution would increase the maximum number of users handled by the system.

The experiments can be extended in the future to include the accuracy. The accuracy is commonly measured using the Mean Absolute Error, which is the average difference between the predicted rate and the actual rate of all user-item testing records. Mean Absolute Error is given in Equation 1.

$$\text{MAE} = \sum_{Testing=1}^{n} \frac{|actual\_rate(U,I) - predicted\_rate(U,I)|}{n}, \qquad (1)$$

Whereas *U* represents the target user, *I* represents an item, and *n* is the number of testing records.

Another interesting research direction is to propose new similarity methods for recommender systems using the hidden user interests. Such similarities could contribute in further increasing the accuracy of the recommender systems. Unfortunately, most of the existing similarities do not consider these hidden interests. Instead, they consider the user rates, the item plot similarity, and so on. Such similarities can be delusive. The similarity in the user rates does not always guarantee the similarity in user behavior. Two users can have the same rates but different interests and behaviors. Therefore, it is important to concentrate on the meaning of the rate, not on the rate itself. This part is left for the future work as well.

6. CONCLUSIONS AND FUTURE WORK

This paper proposes a new solution to the cold start problem. This solution considers the hidden user interests and the behavior of the user. Furthermore, it considers the interests of the group to which the user belongs instead of considering the interests of similar individuals. Upon conducting the experiments, the searching time to find similar users proved to be very fast, basically few seconds, even with large number of users and large number of used factors. Space consumption was relatively efficient for many domains and it could be further improved using hardware and software solutions.

This work is a preliminary work that can be extended in many directions. One proposed extension is to compare the accuracy of the proposed method with the other methods. Another research direction is to propose new similarity methods for recommender systems using hidden user interests. Such similarities could contribute in further increasing the accuracy of the recommender systems.

7. ACKNOWLEDGMENTS

The authors would like to thank Alzaytoonah University of Jordan for its support.

8. REFERENCES

[1] Adomavicius, G., and Tuzhilin, A. 2011. Context-Aware Recommender Systems. Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners, Rokach, L., Shapira, B., Kantor, P., and Ricci, F., eds. 217-250.

[2] Aljawarneh, S. A. and Vangipuram, R. 2018. GARUDA: Gaussian dissimilarity measure for feature representation and anomaly detection in Internet of things. *The Journal of Supercomputing*, 1-38.

[3] Aljawarneh, S. A., Vangipuram, R., Puligadda, V. K., and Vinjamuri, J. 2017. G-SPAMINE: An approach to discover temporal association patterns and trends in internet of things. *Future Generation Computer Systems*, *74*, 430-443.

[4] AlZubi, S., Sharif, MS, and Abbod, M. 2011. Efficient Implementation and Evaluation of Wavelet Packet for 3d Medical Image Segmentation. In: *IEEE International Workshop On Medical Measurements and Applications Proceedings*. 619-622.

[5] Chien, YH and George, EI. 1999. A Bayesian Model for Collaborative Filtering. In: *International Workshop on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, USA.

[6] Dey, A., Abowd, G. and Salber, D. 2001. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *HumAN-Computer Interaction*. 16, 97-166.

[7] Elbes, M, Al-Fuqaha, A. 2013. Design of a Social Collaboration and Precise Localization Services for the Blind and Visually Impaired. *Procedia Computer Science*. 21, 282-291.

[8] Embarak, O.H.. 2011. A method for solving the cold start problem in recommendation systems. In *International Conference on Innovations in Information Technology (IIT)*, Abu Dhabi, UAE, pp. 238-243

[9] Haffar, N., Maraoui, N., Aljawarneh, S., Bouhorma, M., Alnuaimi, AA, Hawashin, B. 2017. Pedagogical Indexed Arabic Text in Cloud E-Learning System. In: *International Journal of Cloud Applications and Computing*. 7(1), 32-46.

[10] Hawashin, B., Abusukhon, A., and Mansour, A. 2015. An Efficient User Interest Extractor for Recommender Systems. In: *The World Congress on Engineering and Computer Science* , San Francisco, CA, USA, 791-795.

[11] Hawashin, B. and Mansour, A. 2016. An Efficient Agent-Based System to Extract Interests of User Groups. In: *The World Congress on Engineering and Computer Science* , San Francisco, CA, USA.

[12] Shadi A. Aljawarneh, Muneer Bani Yassein, and We'am Adel Talafha. 2017. A resource-efficient encryption algorithm for multimedia big data. Multimedia Tools Appl. 76, 21 (November 2017), 22703-22724. DOI: https://doi.org/10.1007/s11042-016-4333-y

[13] Jiang, M., Song, D., Liao, L., et al. 2015. A Bayesian Recommender Model for User Rating and Review Profiling. *Tsinghua Science and Technology*. 20(6), 634-643.

[14] Kanan, T. and Fox, E. 2016. Automatic Arabic Text Classification with P-Stemmer. *Machine Learning and a Tailored News Article Taxonomy.* 67(11), 2667.

[15] Liu, Q., Chen, E., Xiong, H., et al. 2012. Enhancing Collaborative Filtering by User Interest Expansion via Personalized Ranking. *Transactions on Systems, Man, and Cybernetics.* 42(1), 218-233.

[16] Mansour, AM, Obaidat, MA, and Hawashin, B. 2014. Elderly People Health Monitoring System Using Fuzzy Rule Based Approach. *International Journal of Advanced Computer Research.* 4, 904.

[17] Pavlov, D., and Pennock, D. 2002. A Maximum Entropy Approach to Collaborative Filtering in Dynamic, Sparse, High-Dimensional Domains. In: *16th Annual Conference on Neural Information Processing System*, Vancouver,Canada, 1441-1448.

[18] Pazzani, MJ and Billsus, D. 2007. Content-Based Recommendation Systems. *The AdaptiAdaptive Web.* 4321, 325-341.

[19] Pazzani, M. and Billsus, D. 1997. Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning*. 27, 313-331.

[20] Qian, X., Feng, H., Zhao, G., et al. 2014. Personalized Recommendation Combining User Interest and Social Circle. *IEEE Transactions on Knowledge and Data Engineering* 26(7), 1763-1777.

[21] Radhakrishna, V., Aljawarneh, S. A., Kumar, P. V., and Janaki, V. 2018. A novel fuzzy similarity measure and prevalence estimation approach for similarity profiled temporal association pattern mining. *Future Generation Computer Systems*, 83, 582-595.

[22] Resnick, P., Iakovou, N., Sushak, M., et al. 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In: *Computer Supported Cooperative Work Confrerence* , Chapel Hill, NC, USA, 175-186.

[23] Robertson, S. and Walker, S. 2000. Threshold Setting in Adaptive Filtering. *Journal of Documentation*; 56, 312-331.

[24] Sarwar, B., Karypis, G., Konstan, J., et al. 2001. Item-Based Collaborative Filtering Recommendation Algorithms. In: *World Wide Web Conference,* Hong Kong, 285-295.

[25] Schien, A., Popescul, A., Ungar, L., and Pennock, D. 2002. Methods and Metric for Cold Start Recommendations. In: *ACM SIGIR Conference on Research and Development in Information Retrieval*. Tampere, Finland, pp. 253-260.

[26] Soboroff, I. and Nicholas, C. 1999. Combining Content and Collaboration in Text Filtering. In: *International Joint Conference on Artificial Intelligence Workshop: Machine Learning for Information Filtering*, 86-91.

[27] Ungar, LH and Foster, DP. 1998. Clustering Methods for Collaborative Filtering. *Proc. Recommender Systems* Technical Report WS-98-08.

[28] Wang, YK. 2004. Context Awareness and Adaptation in Mobile Learning. In: *IEEE Second International Workshop on Wireless and Mobile Technologies in Education (WMTE '04)*, 154-158.

[29] Zhang, D., Hsu, CH, Chen, M., et al. Cold-Start Recommendation Using Bi-Clustering and Fusion for Large-Scale Social Recommender Systems. *Transactions on Emerging Topics in Computing*. 2(2), 239-250.