An Efficient User Interest Extractor for Recommender Systems

Bilal Hawashin, Ahmad Abusukhon, Ayman Mansour

Abstract— **This paper proposes an efficient method to extract user interests for recommender systems. Although item-item content similarity has been widely used in the literature, it could not detect certain user interests. Our solution improves the current work in two aspects. First, it improves the current recommender systems by detecting actual user interests. Second, it considers many types of user interests such as single-term interest, time interval interest, multi-interests, and dislikes. This extractor would improve recommender systems in many aspects. Our experiments show that our proposed method is efficient in terms of accuracy and execution time.**

Index Terms— **User Interest Extraction, Content Based Filtering, Recommender Systems.**

I. INTRODUCTION

Recommender Systems are used to suggest items to users based on their interests. They have many applications in various domains. Examples of recommender systems include research papers recommenders, book recommenders, product recommenders, twitter follower recommenders, and much more. In this paper, our concentration is on movies recommenders, whereas the items are movies and the users are the store clients. In this paper, the terms movie and item are used interchangeably.

In order to provide recommendations, these recommender systems use user-item rates. When a user buys an item, the recommender system asks the user to rate the item on a scale, commonly one to five, one if the user did not like the item 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 based on that user.

Manuscript received July 09, 2015; revised July 22, 2015.

Bilal Hawashin is with the Department of Computer Information Systems, Alzaytoonah University of Jordan, Amman, 11733 Jordan, e-mail: b.hawashin@zuj.edu.jo.

Ahmad Abusukhon is with the Department of Computer Networks, Alzaytoonah University of Jordan, Amman, 11733 Jordan, e-mail: ahmad.abusukhon@zuj.edu.jo.

Ayman Mansour is with the Department of Communication and Computer Engineering, Tafila Technical University, Tafila, 66110, Jordan, e-mail: mansour@ttu.edu.jo.

Many current recommender systems use item-item content similarity(content based filtering). However, this similarity could not always detect the real user interests. For example, suppose that a user prefers movies of Tom Hanks. Such user would highly rate his films such as “Saving Private Rayan”, the war-related movie. Using this highly rated movie, item-item content similarity method would recommend other war related movies, such as “Schindler's List”, according to the movie plot similarity. However, such recommendations would not meet the user interest because it does not have the user’s preferred actor. Table I provides another example in details. The table contains four movies, three of which were produced in 2002, while the fourth was produced in 1960. Suppose that a user prefers movies produced in 2002. According to the item-item content similarity, the item-item cosine similarity between movie 1 and movie 3 is 0.37. In contrast, the item-item cosine similarity between movie 1 and movie 4 is 0.51, even though movie 4 was released in a different year. This similarity between movie 1 and movie 4 is due to the similarity in the genre between the two movies, as both are of genre comedy romance. Therefore, using item-item cosine similarity, the recommender system would give movie 4 the priority over movie 2, even though movie 4 does not meet the user interest, which is 2002 movies. To summarize, item-item content similarity would fail to detect certain user interests in many scenarios, and therefore, it is necessary to detect actual user interests to improve recommender systems.

Furthermore, user interests may take multiple types. For example, a user may prefer movies of a certain actor. Other user may prefer movies whose genre is both comedy and horror, and this user would not highly rate the movie if it is horror only or comedy only. Third example is when the user prefers movies of a certain period of time, such as movies of seventies or movies of the 21 century. Finally, a user may not have interests but, on the other hand, have negative opinions toward certain movies, such as horror movies.

In this paper, we propose an efficient method to extract user interests. This method is capable of detecting the actual user interests in its various types. Such method could be integrated into existing recommender systems to improve it in many aspects. We leave this step to future work. In order to evaluate the system, we use a synthesized dataset, because commonly used datasets in this domain do not give the actual user interests in the form of terms. To evaluate our method, we use both the rank given by our method to the real user interest and execution time. The contributions of this work are as follows.

ISBN: 978-988-14047-2-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

- Detecting actual user interests and returning them as terms.
- Considering many types of user interests such as single-term interests, time interval interest, multi-interests, and dislikes.

The rest of this paper is organized as follows. Section II is a literature review of the related works in this field. Section III describes our proposed method. Section IV elaborates on some extensions to the method necessary for certain user interests, Section V is the experimental part, and Section VI is the conclusion.

TABLE I
A COMPARISON OF THE ITEM-ITEM SIMILARITIES BETWEEN MOVIE1, PRODUCED IN 2002, AND THREE OTHER MOVIES PRODUCED IN VARIOUS YEARS

Title	Genere	Year	Actor	Country	Cosine Similarity with Movie 1
Waking Up in Reno	Comedy Romance	**2002**	Penelope Cruz	USA	1
Spiderman	Romance Animation	2002	Tobey Maguire	USA	0.51
The Ring	Horror	**2002**	Naomi Watts	USA	**0.37**
The Apartment	Comedy, Romance	**1960**	Mathieu Kassovitz	USA	**0.51**

II. LITERATURE REVIEW

Many works have studied recommender systems. Content based filters[1],[4],[5],[6],[7] recommend items based on its description similarity to the previously highly rated items by the user. In details, [4],[6] used Bayesian classifiers to estimate the probability that a user likes an item based on its content, while [5] used the winnow algorithm that works well when many possible features exist. [7] used a threshold to decide whether the description match that of the highly rated items or not. Collaborative filtering [2],[8],[9],[10],[11],[12],[13] on the other hand, recommend items that were highly rated by similar users. In details, The Grundy system [8] is one of the earliest examples that proposed the use of user stereotypes. Tapestry system[9] demanded users to specify their similar users manually. Memory based methods[10],[11] in collaborative filtering use the previously rated items in finding similar users, in contrast with model based algorithms[12],[13] that learn a model from the previous rates, such as Baysian model[12] and maximum entropy model[13]. Hyprid methods[14],[15] combine both content and collaborative features together. Context aware recommender systems [16],[17],[18] are those that consider context information such as location [17], time[18], and user interests[19] in their recommendations.

As for interest-based recommender systems, [20] proposed a design framework for multi agent interest based system. [21] developed a reinforcement learning strategy for market based multi agent recommendation system to provide the best recommendation when many recommender systems are used. [22] used user movie genre interest to detect account hacks, as attacker would give random and different genre interests . Up to our knowledge, no work has extracted user interests as explicit terms and used them to improve recommender system performance.

III. EXTRACTING USER INTERESTS

Our basic user interest extraction method, which is presented in Algorithm 1, is described below. Some extensions would be presented in the next section. This basic algorithm is capable of extracting actual user interests as terms. Furthermore, it is the first part of many improvements to the recommender systems that are left to the future work.

The user interest extractor has four inputs and one output. The inputs are User Item rate file, which contains rates of each user on items bought by this user, Item Term matrix, where every row presents an item and every column presents a term from the item description file, a user U, and K, which represents the number of interests we want to extract for a user U. The output of the algorithm is a list of K terms that present the interests for user U. The algorithm starts by creating a submatrix of the Item Term matrix, M, for items rated by user U. Each row in the matrix presents an item and each column presents a term. As rated items by a certain user is a subset of the list of all items, the number of rows, N, would be a subset of the number of items I. Next, every row in this matrix M is assigned a label based on the user rate. If the rate of the user U to the item in row r is greater than three, row r is assigned a positive label, +1. In contrast, if the rate of the user U to item in row r is less than three, row r is assigned a negative label, -1. Later, every term in the matrix M is given a value indicating its importance, which reflects the degree of relation between the term and the positive label. The term importance is given for each term according to the following formula, which is the formula for χ^2 method.

$$\mathrm{Imp}(t) = \sqrt{\frac{(n_{pt+}+n_{nt+}+n_{pt-}+n_{nt-})(n_{pt+}n_{nt-}-n_{pt-}n_{nt+})^2}{(n_{pt+}+n_{pt\_})(n_{nt+}+n_{nt-})(n_{pt+}+n_{nt+})(n_{pt-}+n_{nt\_})}}, \quad (1)$$

Where n_{pt+} and n_{nt+} are the number of items highly rated by the user and not highly rated by the user respectively in which the term t appears at least once; n_{pt-} and n_{nt-} are the number of items highly rated by the user and not highly rated by the user respectively in which the term t doesn't occur.

Depending on K, this method would extract the K terms with the highest importance values. These terms present user interests.

IV. EXTENDING EXTRACTOR TO INCLUDE MULTI-INTERESTS, INTERVALS, AND DISLIKES

Algorithm 1 is designed to retrieve user interests as single terms. For example, it would return Horror, Hanks, and so on. However, some user interests may appear in different formats. For example, a user may like a movie if it is both Horror AND Comedy movie, other user may like movies of seventies. Furthermore, a user may does not have specific likes but dislikes animation movies. In all these cases, our method can apply with small extensions. These extensions are presented in the following subsections.

ISBN: 978-988-14047-2-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Algorithm 1: USER INTEREST EXTRACTOR
Input: User Item rate file.
Item Term matrix IT of *I* Items and *T* Terms.
A user *U*.
Desired number of interests *K*.
Output: The top *K* interests for user *U*.
Algorithm:
01 Create a submatrix of matrix IT , named *M*, for items
02 rated by user *U*. *M* is *NXT* , where $N < I$.
03 For each record in *M* with item *x*
04 //If the rate of User *U* to item *x* >3
05 Add +1 label to the record
06 Else
07 Add -1 label to the record
08 Use χ^2 method to assign a value for each term in *M*
09 based on the labels
10 Likes ← The *K* terms of the highest values.
11 Return Likes

A. Multi-Interests

To detect multi-interests, the same algorithm applies after the following extension. In Algorithm 1, line 01, the columns of the matrix IT must contain not only all single terms from the item description file, but also all the n-combinations of these single terms. Each column would present either a single term or an n-combination of the terms. The value of n depends on the domain. For example, it could be dual combination, triple combination, … etc. The rest of the algorithm is the same. In line 10 of the algorithm 1, each single term or a combination of terms would be given a value of importance, and the algorithm would detect whether the user interests are single-term interests or combined interests.

B. Time Interval Interests

Certain user interests have the form of time intervals. For example, some users like movies of seventies, others may like movies of the 20 century, and so on. In these cases, we add the designated interval(s) as column(s) to the matrix IT in line 01. Each interval column is populated as follows. If the movie record is in the interval, we add 1, otherwise, we add 0. For example, we can add to the matrix IT the columns Classic, Seventies, Eighties, Nineties, and Recent. If a user has an interest to the movies of interval, the algorithm would give it a high value of importance and return it among the top interests. If the user does not have a special interest to any of these intervals, none of them would be given a high importance and none would be returned among the top user interests.

C. Dislikes

If we want to extract user dislikes, the sign for rates greater than zero must be -1, and for rates less than zero must be +1. This way, our algorithm would extract user dislikes.

V. EXPERIMENTS

In order to evaluate the user interest extraction method, we need a dataset that provides the actual user interests, along with user-item ratings and item description file. We could not find such dataset among the commonly used ones; therefore, we collected a dataset similar to MovieLens with the actual user interests. Below are details of this dataset.

A. The Synthesized Dataset

This 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 II. Each user rated at least 10 movies. First, we manually specified various interests for users. Table III illustrates this part. Later, we filled the user-item rates according to these likes. It should be noted that if a user has a certain interest, such as Comedy movies for example, it does not mean that the user would highly rate all comedy movies. Therefore, we evaluated our method using various values for interest degree, which is the percentage of highly rated movies that have the user interest in their description. Also, if a user likes comedy movies, it does not mean that the user would not highly rate any non-comedy movie. Therefore, we used various degrees noise, which is the percentage of highly rated movies that do not have the user interest in their description. Item description file has the description of 100 movies, each contains its Title, year, genre(multiple), main actor, main actress, director, and country of production.

For our experiments, we used an Intel® Xeon® server of 3.16GHz CPU and 2GB RAM, with Microsoft Windows Server 2003 Operating System. Also, we used Microsoft Visual Studio 6.0 to read the dataset and execute the methods.

TABLE II
THE SYNTHESIZED DATASET

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

TABLE III
THE ACTUAL USER INTERESTS IN THE SYNTHESIZED DATASET

User Number	Description
1	Likes Tom Hank's movies
2	Likes Indian movies
3	Likes Horror movies
4	Likes movies produced in 2002
5	Likes movies plots about Dead
6	Likes Angelina Jolie's movies
7	Likes movies directed by James Cameron
8	Likes movies of seventies
9	Dislikes Horror Movies
10	Likes movies that are both horror and comedy

In order to evaluate our method, we used rank and execution time measurements. They are defined as follows.

- Rank is the order given by our method to the actual user interest. As mentioned before, our method gives a value of importance for each term(single or combined) and later, it ranks terms based on their importance and retrieve those of the highest importance. The ideal user interest extraction method would always place the actual user interest in rank number one.

ISBN: 978-988-14047-2-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

- Execution time is the time required to run the algorithm. It includes reading dataset and applying χ^2 method.

As mentioned earlier, we used various values of interest degrees and noise. They are defined as follows.

- Interest Degree: The percentage of movies that have the user interest in its description and were highly rated by the user. For example, a user may like horror movies in general but did not like certain horror movie.

- Noise: The percentage of movies that do not have the user interest in its description and were highly rated by the user. For example, a user may like horror movies in general but also liked a movie that is not a horror movie.

According to Table IV, interest degree and noise are formally defined as follows.

Interest Degree = $A/(A+C)$ (2)

Noise = $B/(B+D)$ (3)

TABLE IV
INTEREST DEGREE AND NOISE ELEMENTS

	User Interest Term Appeared in Movie Description	**User Interest Term did not Appear in Movie Description**
Movie was Highly Rated	*A*	*B*
Movie was not Highly Rated	*C*	*D*

First, we retrieved the rank of the term Hanks using interest degree values ranging from 50% to 100%, and noise values ranging from 0 to 20%. Table V illustrates the results. Clearly, our method succeeded to rank the actual interest term in the first order when the interest degree was 60% or more. When the interest degree becomes 50%, the rank of the actual interest shifted down gradually. When applying the method on the term Horror, as shown in Table VI, our method succeeded even with interest degree 50% and noise up to 15%.

TABLE V
THE RANK OF THE TERM HANKS USING VARIOUS INTEREST DEGREES AND NOISE VALUES

	Interest Degree = 100%	**80%**	**60%**	**50%**
Noise = 0%	1	1	1	43
5%	1	1	1	43
10%	1	1	1	43
15%	1	1	1	63
Noise = 20%	1	1	57	64

TABLE VI
THE RANK OF THE TERM HORROR USING VARIOUS INTEREST DEGREES AND NOISE VALUES

	Interest Degree = 100%	**90%**	**80%**	**70%**	**60%**	**50%**
Noise = 0%	1	1	1	1	1	1
5%	1	1	1	1	1	1
10%	1	1	1	1	1	1
15%	1	1	1	1	1	1
20%	1	1	1	1	1	66

Table IX presents the noise percentage when the actual user interest rank shifted from the first rank when interest degree was 60%. From this Table, we notice that when the interest degree is 60%, which is a rational minimum degree if we want to claim that a user has an interest, then for all the users, when the noise was below the average of 20%, the interest would be in the first order. When noise becomes around 20%, the interest order shifted. We also noted that the shift depends on the terms appearing in the movie descriptions for the movies highly rated by the user. For example, if a user likes many noisy movies that have common terms, these terms would have high order that would push down the actual interest order. In contrast, if the noisy movies do not have common terms, then the actual user interest would maintain its high order.

TABLE IX
THE NOISE PERCENTAGE VALUE WHEN THE ACTUAL INTEREST RANK WAS SHIFTED FROM THE FIRST ORDER WHEN USER INTEREST DEGREE = 60%

	Noise that shifted term below rank 1	**New Rank**
Hanks	20%	57
Indian	10%	2
Horror	25%	68
2002	15%	78
Dead	20%	3
Jolie	20%	2
Cameron	20%	5
Seventies	25%	85
Dislike Horror	30%	106
Comedy & Horror	15%	40

As for execution time, the main algorithm needed 1.56 seconds to read the dataset and extract the user interests. As for the extensions, extracting user dislikes needed the same time as it applies the same algorithm with flipping the signs. Adding time intervals increased slightly the execution time as in Table X. Regarding extracting multi-interests, we leave it for future work because including all n-combinations is time consuming.

ISBN: 978-988-14047-2-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE X
THE EXECUTION TIME FOR THE ALGORITHM AND ITS EXTENSIONS

	Average Extraction Time(s)	Dataset Reading Time(s)
Single-Term Interest/Dislike	0.26	1.3
Adding One Interval		1.5
Adding Two Intervals		1.7
Adding Three Intervals		2

VI. Conclusion

This paper proposes an efficient method to extract user interests. This method can extract the user interests as explicit terms. Besides, it can extract user multi-interests, time interval interests, and dislikes. Experiments showed that the method retrieval order for user interests is accurate as long as the user interest is 60% and with noise up to 20%. As for execution time, the method and both extracting dislikes and intervals were time efficient.

Future work can be done to improve the execution time for extracting user multi-interests. Furthermore, this work serves as the first step in improving recommender systems. Future work can be done to integrate the extraction method into recommender systems to give interesting statistics and further improve its performance.

Acknowledgment

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

References

[1] M.J. Pazzani and D. Billsus, "Content-Based Recommendation Systems," *The Adaptive Web*, P. Brusilovsky, A. Kobsa, and W. Nejdl, eds., vol. 4321, pp. 325-341, Springer-Verlag, 2007.

[2] J.L. Herlocker, J.A. Konstan, L.G. Terveen, and J.T. Riedl, "Evaluating Collaborative Filtering Recommender Systems," *ACM Trans. Information Systems*, vol. 22, no. 1, pp. 5-53, 2004.

[3] R. Burke, "Hybrid Web Recommender Systems," *The Adaptive Web*, P. Brusilovsky, A. Kobsa, and W. Nejdl, eds., vol. 4321, ch. 12, pp. 377-408, Springer, 2007.

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

[5] N. Littlestone and M. Warmuth, "The Weighted Majority Algorithm," *Information and Computation*, vol. 108, no. 2, pp. 212-261, 1994.

[6] R.J. Mooney, P.N. Bennett, and L. Roy, "Book Recommending Using Text Categorization with Extracted Information," Proc. Recommender Systems Papers from 1998 Workshop, Technical Report WS-98-08, 1998.

[7] S. Robertson and S. Walker, "Threshold Setting in Adaptive Filtering," J. Documentation, vol. 56, pp. 312-331, 2000.

[8] E. Rich, "User Modeling via Stereotypes," *Cognitive Science*, vol. 3, no. 4, pp. 329-354, 1979.

[9] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry, "Using Collaborative Filtering to Weave an Information Tapestry," Comm. ACM, vol. 35, no. 12, pp. 61-70, 1992.

[10] P. Resnick, N. Iakovou, M. Sushak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," Proc. 1994 Computer Supported Cooperative Work Conf., 1994.

[11] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-Based Collaborative Filtering Recommendation Algorithms," Proc. 10th Int'l WWW Conf., 2001.

[12] Y.-H. Chien and E.I. George, "A Bayesian Model for Collaborative Filtering," Proc. Seventh Int'l Workshop Artificial Intelligence and Statistics, 1999.

[13] D. Pavlov and D. Pennock, "A Maximum Entropy Approach to Collaborative Filtering in Dynamic, Sparse, High-Dimensional Domains," Proc. 16th Ann. Conf. Neural Information Processing Systems (NIPS '02), 2002.

[14] I. Soboroff and C. Nicholas, "Combining Content and Collaboration in Text Filtering," Proc. Int'l Joint Conf. Artificial Intelligence Workshop: Machine Learning for Information Filtering, Aug. 1999.

[15] L.H. Ungar and D.P. Foster, "Clustering Methods for Collaborative Filtering," Proc. Recommender Systems, Papers from 1998 Workshop, Technical Report WS-98-08 1998.

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

[17] Y.-K. Wang, "Context Awareness and Adaptation in Mobile Learning," Proc. IEEE Second Int'l Workshop Wireless and Mobile Technologies in Education (WMTE '04), 2004, pp. 154-158.

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

[19] L. Nguyen and P. Do, "Learner Model in Adaptive Learning," *World Academy of Science Eng. and Technology*, vol. 45, pp. 395-400, 2008

[20] P. Vashisth and P. Bedi, "Interest-Based Personalized Recommender System," World Congress on Information and Communication Technologies, 2011.

[21] Y. Z. Wei, L. Moreau, and N. R. Jennings, "Learning Users' Interests by Quality Classification in Market-Based Recommender Systems," *IEEE Trans. Knowledge and Data Engineering*, vol. 17, no. 12, pp. 1678-1688, Dec. 2005.

[22] G. Aghili, M.. Shajari, S. Khadivi, and M. A. Morid, "Using Genre Interest of Users to Detect Profile Injection Attacks in Movie Recommender Systems," Proc. 10th Ann. Conf. Machine Learning and Applications, 2011.