Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 4, April 2015, pg.106 – 113

RESEARCH ARTICLE

COLLABORATIVE FILTERING USING FUZZY NEIGHBOUR IDENTIFICATION

Ms.Ramya.L
M.E.,CSE.,(PT)
SCSVMV
Kanchipuram
ramyanaidu2811@gmail.com

Mrs.D.Thamaraiselvi M.E
Assistant Professor
Department of CSE
SCSVMV, Kanchipuram
Thamaraiselvi17@gmail.com

Abstract: Collaborative filtering (CF) is an important and popular technology for recommender systems. However, current CF methods suffer from such problems as data sparsity, recommendation inaccuracy, and big-error in predictions. In this paper, we borrow ideas of object typicality from cognitive psychology and propose a novel typicality-based collaborative filtering recommendation method named TyCo. A distinct feature of typicality-based CF is that it finds "neighbors" of users based on user typicality degrees in user groups (instead of the corated items of users, or common users of items, as in traditional CF). To the best of our knowledge, there has been no prior work on investigating CF recommendation by combining object typicality. TyCo outperforms many CF recommendation methods on recommendation accuracy (in terms of MAE) with an improvement of at least 6.35 percent in Movielens data set, especially with sparse training data (9.89 percent improvement on MAE) and has lower time cost than other CF methods. Further, it can obtain more accurate predictions with less number of big-error predictions.

Index Terms—Recommendation, typicality, collaborative filtering

1. INTRODUCTION

Collaborative filtering (CF) is an important and popular technology for recommender systems. There has been a lot of work done both in industry and academia. these methods are classified into user-based CF and item-based CF. The basic idea of user-based CF approach is to find out a set of users who have similar favor patterns to a given user (i.e., "neighbors" of the user) and recommend to the user those items that other users in the same set like, while the item-based CF approach aims to provide a user with the recommendation on an item based on the other items with high correlations (i.e., "neighbors" of the item). In all collaborative filtering methods, it is a significant step to find users' (or items') neighbors, that is, a set of similar users (or items).

For instance, Raymond is a very typical member of the concept "users who like war movies" while not so typical in the concept "users who like romance movies." The typicality of users in different user groups can indicate the user's favor or preference on different kinds of items. The typicality degree of a user in a particular user group can reflect the user's preference at a higher abstraction level than the rated items by the user.

2. BACKGROUND AND RELATED WORK

Prototype View and Typicality

In cognitive psychology, object typicality is considered as a measure of the goodness degree of objects as exemplars in concepts

It depends on salient properties shared by most of the objects of that concept, which generally include salient properties that are not necessary for defining the concept as well as those that are in the prototype view of concepts. Concept is represented by a best prototype abstracted by the property list that consists of the salient properties of the objects that are classified into this concept. The salient properties defining the prototype include both necessary and unnecessary properties. It has been found that typicality of an instance can be determined by the number of its properties which it shares with the concept prototype. For example, the property "can-fly" will probably appear in the proto-type of the concept "bird" because most birds can fly. So birds that can fly will be judged as more typical than those that cannot. A prototype of a concept is considered as the best example of the concept, and is abstracted to be a feature list. Although the prototype view can explain many different aspects of how concepts and properties are represented in human's mind, there are also situations in which it fails to give a thorough explanation. For example, there is virtually no prototype to represent the concept "animal."

Recommender Systems

There have been many works on recommender systems and most of these works focus on developing new methods of recommending items to users (e.g., works in [13], [14]). The objective of recommender systems is to assist users to find out items which they would be interested in. Items can be of any type, such as movies, jokes, restaurants, books, news articles, and so on. Currently, recommendation methods are mainly classified into collaborative filtering (CF), content based (CB), and hybrid methods [2]. For the reason that we are focusing on proposing a new CF method, we will introduce the related works about CF methods in more details.

Content-Based Recommender Systems

The inspiration of this kind of recommendation methods comes from the fact that people have their subjective evaluations on some items in the past and will have the similar evaluations on other similar items in the future.

Collaborative Filtering

CF recommendation methods predict the preferences of active users on items based on the preferences of other similar users or items. For the reason that CF methods do not require well-structured item descriptions, they are more often implemented than CB methods [2], and many collaborative systems are developed in academia and industry. There are two kinds of CF methods, namely user-based CF approach and item-based CF approach [2].

The basic idea of user-based CF approach is to provide recommendation of an item for a user based on the opinions of other like-minded users on that item. The user-based CF approach first finds out a set of nearest "neighbors" (similar users) for each user, who share similar favorites or interests. Then, the rating of a user on an unrated item is predicted based on the ratings given by the user's "neighbors" on the item. The basic idea of item-based CF approach is to provide a user with the recommendation of an item based on the For both user-based CF and item-based CF, the measurement of similarity between users or items is a significant step. Pearson correlation coefficient, cosine-based similar-ity, vector space similarity, and so on is widely used in similarity measurement in CF methods [2].

There are some hybrid methods such as [13]. Besides, Huang et al. [1] try to apply associative retrieval techniques to alleviate the sparsity problem. Hu et al. [19] explore algorithms suitable for processing implicit feedbacks. Umyarov and Tuzhilin [20] propose an approach for incorporating externally specified aggregate ratings information into CF methods.

Hybrid Recommender Systems

Several recommender systems (e.g., [28] and [29]) use a hybrid approach by combining collaborative and content-based methods, so as to help avoid some limitations of content-based and collaborative systems. Naive hybrid approaches is to implement collaborative and CB methods separately, and then combine their predictions by a combining function, such as a linear combination of ratings or a voting scheme or other metrics. Melville et al. [28] use a CB method to augment the rating matrix and then use a CF method for recommendation.

Some hybrid recommender systems combine item-based CF and user-based CF.

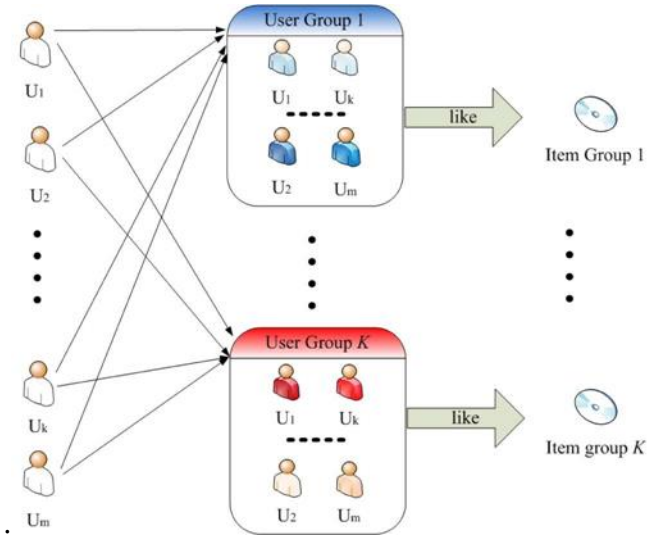
3. TYPICALITY-BASED COLLABORATIVE FILTERING

In this section, we propose a typicality-based collaborative filtering approach named TyCo, in which the "neighbors" of users are found based on user typicality in user groups instead of co-rated items of users. We first introduce some formal definitions of concepts in TyCo in Section 3.1. The mechanism of TyCo is then described in Section 3.2. We introduce its technical details in Sections 3.3, 3.4, 3.5, and 3.6.

3.1 Preliminaries

Assume that in a CF recommender system, there are a set U of users, and a set O of items. Items can be clustered into several item groups and an item group is intuitively a set of similar items. For example, movies can be clustered into action movies, war movies, and so on. Each movie belongs to different movie groups to different degrees. The choice of clustering method is application domain dependent, and is out of the scope of this paper. For instance, based on the keyword descriptions of movies, we can use Topic Model-based clustering [30], [31] for the task of obtaining movie groups and the degrees of movies belonging to movie groups. In other application domains, other clustering approaches (such as [32], [33]) can also be used. In this paper, we will not discuss clustering methods further.

The formal definition of an item group is given in the following



The relations among users, user groups, and item groups are as shown in Fig. 1. Users possess different typical degrees in different user groups: the darker a user is in Fig. 1, the more typical it is in that user groups. For examples, U_1 and U_k are typical in user group g_k but not typical in g_1, while U_2 and U_m are typical in g_1 but not typical in g_k.

For the reason that users have different typicality degrees in different user groups, we represent a user by a user typicality vector defined below

3.2 Mechanism of TyCo

The mechanism of TyCo is as follows: given a set O ¼ fO$_1$; O$_2$; . . . ; O$_h$g of items and a set U ¼ fU$_1$; U$_2$; . . . ; U$_m$g of users, a set K ¼ fk$_1$; k$_2$; . . . ; k$_n$g of item groups is formed. For each item group k$_i$, there is a corresponding user group ^{gi}.

Users have different typicality degrees in each g^i. Then, a user typicality vector U $_i$ is built for each user, from which user-typicality matrix M_ is obtained. After obtaining users similarity based on their typicality degrees in user groups, a set N $_i$ of "neighbors" is obtained for each user. Then, we predict the rating of an active user on an item based on the ratings by "neighbors" of that user on the same item.

	g$_1$	g$_2$	g$_3$	g$_4$	g$_5$	g$_6$
U$_1$	0. 87	0. 75	0. 92	0. 12	0. 32	0. 28
U$_2$	0. 34	0. 21	0. 38	0. 89	0. 85	0. 94
⋮	...	...	...	...	...	...
U$_k$	0. 81	0. 79	0. 89	0. 15	0. 29	0. 31
⋮	...	...	...	...	...	...
U$_m$	0. 41	0. 22	0. 35	0. 90	0. 88	0. 92

Fig. 2. An example of user-typicality matrix in TyCo.

3.3 Item Typicality Measurement

As introduced above, the typicality of an object in a concept depends on the central tendency of the object for the prototype of the concept.

3.4 Neighbors Selection

We select a fuzzy set of "neighbors" of user U^j, denoted by
$\dot{\bar{N}}_j$, by choosing users who are sufficiently similar to U$_j$, i.e.,

$$N_j \text{ ¼ } fU_i \text{ j } Sim\eth U_i; U_j\text{Þ } \_ \_g;$$

where SimðU$_i$; U$_j$Þ is the similarity of U$_i$ and U$_j$ and _ is a threshold to select users who are qualified as "neighbors" of user U$_j$. represented by a set of properties, which, following our previous work [34], we shall call item property vector. For example, keywords, actors, directors, and producers are properties of a movie and these properties can form an item property vector to represent a movie. For each item group k$_j$, we can extract a prototype to represent the item group.
The prototype of k^j is represented by a set of properties

denoted by the prototype property vector of k$_j$ t $_{kj}$, as follows:

$$^{t}k_j \text{ ¼ } \eth p k_j;1 : {}^{r}k_j;1;{}^{p}k_j;2 : {}^{r}k_j;2;\cdots;{}^{p}k_j;m : {}^{r}k_j;m\text{Þ};$$

where m is the number of the properties of the prototype of concept (item group) k$_j$, and r$_{kj;i}$ is a real number (between 0 and 1), which indicates the degree the prototype of concept k$_j$ possesses the property p$_{kj;i}$.[3] The typicality of an item O$_y$ in an item group k$_j$, denoted by w$_{j;y}$, depends on the similarity between the item O$_y$ and the prototype of k$_j$.

	i_1	i_2	...	i_k	...	i_n
U_1	5	?	...	3	...	4
U_2	?	?	...	4	...	5
...	...	...	...	...	...	...
U_k	2	5	...	?	...	3
...	...	...	...	...	...	...
U_m	5	4	...	2	...	?

3.5 Prediction

Having obtained the set of "neighbors" of each user, we can predict the rating of an active user Ui on an item Oj, denoted by RðUi; OjÞ, based on the ratings of all "neigh-bors" of Ui on Oj, as follows:
where Ux is a user in the set of "neighbors" of Ui, RðUx; OjÞ is the rating of user For TyCo, we first use the Topic Ux on item Oj, and SimðUx; UiÞ is the similarity between Ux and Ui. This function calculates a weighted sum of all ratings given by the "neighbors" of Ui on Oj.

4. EXPERIMENTS

We report the results obtained from experiments conducted to compare the typicality-based CF (TyCo) with other current CF methods. We want to address the following questions:

1. How does TyCo compare with other current CF methods?
2. Does TyCo have a good performance with sparse training data?
3. Can TyCo obtain a good result with less big-error predictions?
4. Is TyCo more efficient than other existing methods?
5. How does the number of user groups affect the recommendation quality?
6. How do the similarity function and threshold _ affect the recommendation results?

4.1 Experimental Setting

4.1.1 Data Set Description

To evaluate our recommendation method, we use the MovieLens data set in the experiments, as this data set has been widely used in previous papers such as [17], [39]. From the MovieLens data set, we obtain 100,000 ratings, assigned by 943 users on 1,682 movies. Each user has rated at least 20 movies, and the ratings follow the 1 (bad) to 5 (excellent) numerical scale. The sparsity level of the data set. To measure statistical accuracy, we use the mean absolute error (MAE) metric, which is defined as the average absolute difference between predicted ratings and actual ratings [2]. MAE is a measure of deviation of recommendations from real user-rated ratings, and it is most commonly used and very easy to interpret. It is computed by averaging the all sums of the absolute errors of the n corresponding ratings-prediction pairs, and can be formally defined as follows:

$$MAE = \frac{\sum_{i=1}^{n} |f_i - h_{ij}|}{n}$$

where n is the number of rating-prediction pairs, f_i is an actual

4.1.2 Experiment Process

Model-based clustering [30] to cluster the movies described by keywords. Based on the clustering results for item

groups, we then form a corresponding user group for each item group and build a prototype for each user group. To evaluate TyCo thoroughly. Similar to previous works such as [40], [14] and [13], we conduct two experiments.In the second experiment, we compare TyCo with some state-of-the-art methods to further demonstrate theadvantages of TyCo on improving the recommendation quality. The compared state-of-the-art methods include a cluster-based Pearson Correlation Coefficient method (SCBPCC) [40], weighted low-rank approximation (WLR) [41], a transfer learning-based collaborative filtering (CBT) [42], and SVD++ [43]. In this experiment, similar to previous works [40], [42], we extract a subset of 500 users from the data set, and select the first 100, 200, and 300 users to form the training sets, named as ML100, ML200, and ML300, respectively (the remaining last 200 users are used as test users). As to the active (test) users, we vary the number of rated items provided by the test users in training from 5, 10, to 20, and give the name Given5, Given10, and Given20, respectively. To evaluate TyCo furthermore, we also compare TyCo with SVD++ using Netflix data.

The experiment is conducted using a PC with a Pentium 4 3.2-GHz CPU, 2-GB memory, Windows XP Professional operating system, and Java J2SE platform

4.2 Experimental Results

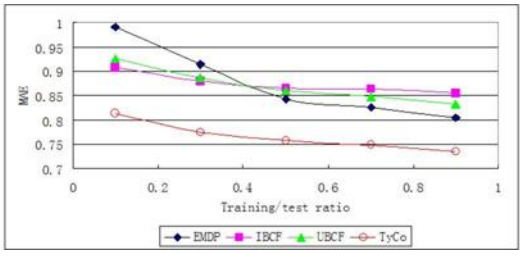
4.2.1 Impact of Number of User Groups

In TyCo, the number of user groups is the same as the number of item groups. To test the sensitivity of different number of user groups (i.e., n), we run experiments for various n from 5 to 30, and the best results (with the most suitable parameter _) on MAE and coverage are shown in Tables 1 and 2, respectively. According to Tables 1 and 2, we find that the number of user groups has little effect on the recommendation results. Although the MAE values for some n (e.g., n ¼ 25) are a little bigger than those for other n (e.g., n ¼ 20), their coverage values are still bigger. Thus, we regard recommendation quality under different n as stable by setting an appropriate.

4.2.2 Comparison on Recommendation Quality

As mentioned in Section 4.1.3, we adopt three existing recommendation methods as baselines, which are user-based collaborative filtering with Pearson Correlation Coefficient, item-based collaborative filtering with Pearson Correlation Coefficient, and the EMDP method [13], to compare with the novel typicality-based CF method. Figs. 4 and 5 show the comparison results of TyCo (we set the number of user groups n ¼ 20 and _ ¼ 0:6) with the baseline methods with different train/test ratios on MAE and coverage, respectively.

From Fig. 4, we can find that TyCo outperforms all other three baseline methods in all train/test ratios on MAE.



4.2.3 Performance on Sparse Data

Another interesting finding is that using TyCo can obtain good MAE and coverage values even with low train/test ratio _. For instance, according to Figs. 4 and 5, the MAE value is 0.8125 and coverage value is 0.9685 using TyCo, even with a very low train/test ratio _ ¼ 0:1. Such results are comparable to those results obtained by EMDP method with a high train/test ratio _ ¼ 0:7. For the baseline methods in our evaluation, the best MAE is 0.9073 obtained by IBCF with _ ¼ 0:1, while the best coverage is 1 obtained by EMDP. From Fig. 4, it is clear that TyCo can produce more accurate recommendations than the compared meth-ods with sparse data sets. According to Fig. 5, TyCo and EMDP can predict much more ratings for users on unrated items than other compared methods.

4.2.4 Comparison on Prediction Errors (PEs)

Predictions with big error may reduce the users' trust and loyalty of recommender systems. Therefore, in our experiments, we also evaluate the prediction errors of the TyCo recommendation method. Fig. 6 compares TyCo with distance-based similarity (we set n ¼ 20 and _ ¼ 0:6) and other current methods on prediction error, with the train/test ratio _ ¼ 0:9. For other train/test ratios (i.e., _ < 0:9), the comparison results are similar and are not shown here.

4.2.5 Impact of Similarity Functions

In Section 3.5, we presented three similarity functions that are distance-based similarity, cosine-based similarity function, and Pearson correlation similarity function. Besides, we set a similarity threshold _ in our "neighbors" selection. We conduct experiments here to evaluate the effect of different similarity functions and that of different thresh-olds on recommendation quality.

Fig. 8 shows the best MAE results (with the best setting of _) of different similarity functions with different train/test ratios. According to Fig. 8, the distance-based similarity can achieve smallest MAE values in all train/test ratio. Thus, we consider that the distance-based similarity is more appropriate for typicality-based CF and adopt it in the comparison experiments.

To evaluate the effect of similarity threshold _, we conduct experiments by setting _ from 0.1 to 0.9 on different n. Fig. 9 shows the comparison of different _ for n ¼ 10, n ¼ 20, and n ¼ 30. For each n, there is a best similarity threshold _. With the increase of n, the best _ decreases for obtaining the best result. For example, the best value for _ is 0.7 when n ¼ 10 and it decreases to 0.5 when n ¼ 30. We think that the reason is as follows: for smaller n (e.g., n ¼ 5), users belong to user groups with higher degrees because the prototype of each cluster is not so specific; while for larger n (e.g., n ¼ 35), users belong to user groups with a lower degrees because the prototype of each cluster is more specific.

When we increase _ to a value that is much greater than the best value, MAE increases quickly. The reason is that the requirement to be "neighbors" of a user is more strict as _ increases and there is not enough qualified "neighbors" for each user with large. A small set of "neighbors" is not good enough to assist predicting unknown ratings for users and may cause big-error predictions.

5. CONCLUSIONS AND FUTURE ENHANCEMENT

In this paper, we investigate the collaborative filtering recommendation from a new perspective and present a novel typicality-based collaborative filtering recommendation method named TyCo. In TyCo, a user is represented by a user typicality vector that can indicate the user's preference on each kind of items. A distinct feature of TyCo is that it selects "neighbors" of users by measuring users' similarity based on their typicality degrees instead of corated items by users. Such a feature can overcome several limitations of traditional collaborative filtering methods. It is the first work that applies typicality for collaborative filtering. We conduct experiments to evaluate TyCo and demonstrate the advantages of TyCo. In TyCo, there are some preprocessing procedures, such as constructing user prototype by clustering and measuring user typicality in user groups. The cost of these preprocessing procedures depends on the particular clustering method used. In real-life applications, these procedures can be processed offline. While users' prototypes are constructed, the remained recommendation process which is based on user typicality will be efficient. For large scale applications, we can also first conduct the above preprocessing offline, and then adopt some parallel computing methods (e.g., MapReduce) to speed up the computing.

There are several possible future extensions to our work. In TyCo, we do not specify how to cluster resources so as to find out item groups and the corresponding user groups. One possible future work is to try different clustering methods and see how the recommendation results are affected. How to using parallel computing methods (e.g., MapReduce) to handle the large scale applications is also one of the possible future works.

ACKNOWLEDGEMENTS

The authors are grateful to the anonymous reviewers and Mr. Hao Han. The research was supported by National Natural Science Foundation of China (Grant No. 61300137), Guangdong NSFC (S2011040002222 and S2013010013836), the Fundamental Research Funds for the Central Universities, SCUT (2014ZZ0035).

REFERENCES

[1] Z. Huang, H. Chen, and D. Zeng, "Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering," ACM Trans. Information Systems, vol. 22, no. 1, pp. 116-142, 2004.
[2] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 6, pp. 734-749, June 2005.
[3] K.M. Galotti, Cognitive Psychology In and Out of the Laboratory, third ed. Wadsworth, 2004.
[4] G.L. Murphy, The Big Book of Concepts. MIT Press, 2002.
[5] L.W. Barsalou, Cognitive Psychology: An Overview for Cognitive Scientists. Lawrence Erlbaum Assoc., 1992.
[6] S. Schiffer and S. Steele, Cognition and Representation. Westview Press, 1988.
[7] D.L. Medin and E.E. Smith, "Concepts and Concept Formation," Ann. Rev. of Psychology, vol. 35, pp. 113-138, 1984.
[8] W. Vanpaemel, G. Storms, and B. Ons, "A Varying Abstraction Model for Categorization," Proc. Cognitive Science Conf. (CogSci '05), pp. 2277-2282, 2005.
[9] L.W. Barsalou, "Ideals, Central Tendency, and Frequency of Instantiation as Determinants of Graded Structure in Categories,"

