New Hybrid Recommendation System Based On C-Means Clustering Method

Mohammad Hamidi Esfahani
Department of Information Technology
K. N. Toosi University of Technology
Tehran, Iran
hamidi@nsec.ir

Farid Khosh Alhan
Department of Information Technology
K. N. Toosi University of Technology
Tehran, Iran
khoshalhan@kntu.ac.ir

Abstract— **Nowadays recommendation systems are widely used in E-Commerce. They can learn about user interests and automatically suggest the best product to the consumer. Most of these recommendation systems are using collaborative, content-based or knowledge-based method. Users and products can gather in some groups based on their similar features. Using these groups can improve their recommendations and help these systems to solve some problems (*for example cold start problem*). Many clustering methods used to in recommendation systems but a few of these methods are light or easy to use so they can make the recommendation process and user feedback faster, in the other hand, having a good recommendation is more useful than having too many recommendations that a few of them take the user attention. In this paper, a hybrid recommendation system with C-Means clustering method selected to have a better and faster recommendation system.**

Keywords-component; Recommendation system, fuzzy, C-Means, K-Means, Clustering

I. INTRODUCTION

Recommendation systems will gather too many information about users preferences and items, so this information can be used to have a better recommendation.

In this paper, first of all three recommendation methods are discussed then a hybrid approach based on C-Means clustering method described to combine the results of these three methods to decide which product should recommend to which user.

The proposed method has combine content-based, collaborative and knowledge-based methods. It has a collaborative filtering on the result of content-based filtering on items and users. Then extract some rules for rule engine. The knowledge-based part can recommend some items to users and get feedback to prioritize rules by user click counting.

Recommendation systems are using collaborative, content-based or knowledge-based method to have a better recommendation. Each recommendation systems have some strategy to recommend better, the most useful strategies are listed below.

A. Collaborative recommendation systems

The basic idea of these systems is that if users shared the same interests in the past, they will also have similar tastes in the future. So, for example, if user A and user B have a purchase history that overlaps strongly and user A has recently bought an item that B has not yet been, the basic rationale is to propose this item also to B.

Because this selection of hopefully interesting items involves filtering the most promising ones from a large set and because the users implicitly collaborate with one another, this technique is also called collaborative filtering (CF).

Typical questions that arise in the context of collaborative approaches include the following:

- How to find users with similar tastes to the user for whom we need a recommendation?
- How to measure similarity?
- What should be done with new users, for whom a buying history is not yet available?
- How to deal with new items that nobody has bought yet?
- What other techniques besides looking for similar users can use for making a prediction about whether a certain user will like an item?

Pure CF approaches do not exploit or require any knowledge about the items themselves. The recommender system, for instance, does not need to know what a book is about, its genre, or who wrote it. The obvious advantage of this strategy is that these data do not have to be entered into the system or maintained [1].

B. Content-Based Recommendation Systems

In general, recommender systems may serve two different purposes. On one hand, they can be used to stimulate users into doing something such as buying a specific book or watching a specific movie. On the other hand, recommender systems can also be seen as tools for dealing with information overload, as these systems aim to select the most interesting items from a larger set. Thus, recommender systems research is also strongly rooted in the fields of information retrieval and

978-1-4673-6490-4/13/$31.00 ©2013 IEEE

information filtering. In these areas, however, the focus lies mainly on the problem of discriminating between relevant and irrelevant documents. Many of the techniques developed in these areas exploit information derived from the documents' contents to rank them.

At its core, content-based recommendation is based on the availability of (manually created or automatically extracted) item descriptions and a profile that assigns importance to these characteristics. Similar to item descriptions, user profiles may also be automatically derived and "learned" either by analyzing user behavior and feedback or by asking explicitly about interests and preferences.

In the context of content-based recommendation, the following questions must be answered:

- How can systems automatically acquire and continuously improve user profiles?
- How to determine which items match, or are at least similar to or compatible with, a user's interests?
- What techniques can be used to automatically extract or learn the item descriptions to reduce manual annotation?

When compared with the content-agnostic approaches described above, content-based recommendation has two advantages. First, it does not require large user groups to achieve reasonable recommendation accuracy. In addition, new items can be immediately recommended once item attributes are available.

In some domains, such item descriptions can be automatically extracted or are already available in an electronic catalog.

In many domains, however, the more subjective characteristics of an item (such as "ease of use" or "elegance of design") would be useful in the recommendation process. These characteristics are hard to acquire automatically, however, meaning that such information must be manually entered into the system in a potentially expensive and error-prone process [1].

C. Knowledge-Based recommendation systems

According to other application domains, such as consumer electronics, many involve large numbers of one-time buyers. This means that we cannot rely on the existence of a purchase history, a prerequisite for collaborative and content-based filtering approaches. However, more detailed and structured content may be available, including technical and quality features take, for instance, a recommender system for digital cameras that should help the end user find a camera model that fits his or her particular requirements.

Typical customers buy a new camera only once every few years, so the recommender system cannot construct a user profile or propose cameras that others liked, which – as a side note – would result in proposing only top-selling items.

Thus, a system is needed that exploits additional and means end knowledge to generate recommendations. In such knowledge-based approaches, the recommender system typically makes use of additional, often manually provided, information about both the current user and the available items. Constraint-based recommenders are one example of such systems, which we will consider in our discussion of the different aspects of knowledge-based approaches. In addition, explicit constraints may be used to describe the context in which certain features are relevant for the customer. Simply presenting products that fulfill a given set of requested features is not enough, as the aspect of personalization is missing, and every user with the same set of requested features will get the same set of recommendations. So, constraint-based recommender systems also need to maintain user profiles.

The other aspect is "user interaction", as in many knowledge-based recommender systems, the user requirements must be elicited interactively. Therefore, more complex types of interaction are required to determine the user's needs and preferences, mostly because no purchase history is available that can be exploited. A simple approach would be to ask the user directly about his or her requirements. Such an approach, however, not only requires detailed technical understanding of the item's features but also generates additional cognitive load in scenarios with a large number of item features. More elaborate approaches, therefore, try to implement more conversational interaction styles, in which the system tries to incrementally ascertain preferences within an interactive and personalized dialog [1].

Overall, the questions in knowledge-based recommender systems include the following:

- What kinds of domain knowledge can be represented in a knowledge base?
- What mechanisms can be used to select and rank the items based on the user's characteristics?
- How to acquire the user profile in domains in which no purchase history is available, and how to take the customer's explicit preferences into account?
- Which interaction patterns can be used in interactive recommender systems?
- Finally, in which dimensions can personalize the dialog to maximize the precision of the preference elicitation process?

II. RELATED WORKS

David W. McDonald and Mark S. Ackerman have made a flexible recommendation system by locating the expertise necessary to solve difficult problems. They said, in organizations, some people assist others in locating expertise by making referrals. People who make referrals fill key organizational roles that have been identified by CSCW and affiliated research. Expertise locating systems are not designed to replace people who fill these key organizational roles. Instead, expertise locating systems attempt to decrease workload and support people who have no other options. They use collaborative recommendation system that describes a general recommendation architecture that is grounded in the field of expertise locating [4].

Qing Li and Byeong Man Kim use a clustering method on an item-based collaborative filtering framework to solve the cold start problem [5], but they don't use it on users.

III. PROPOSE METHOD

In this paper, we describe a new hybrid recommendation system to combines the content-based, collaborative and knowledge-based methods to capitalize on their respective strengths, and thereby achieve a good performance. Figure 1 shows the structure of new recommender system.

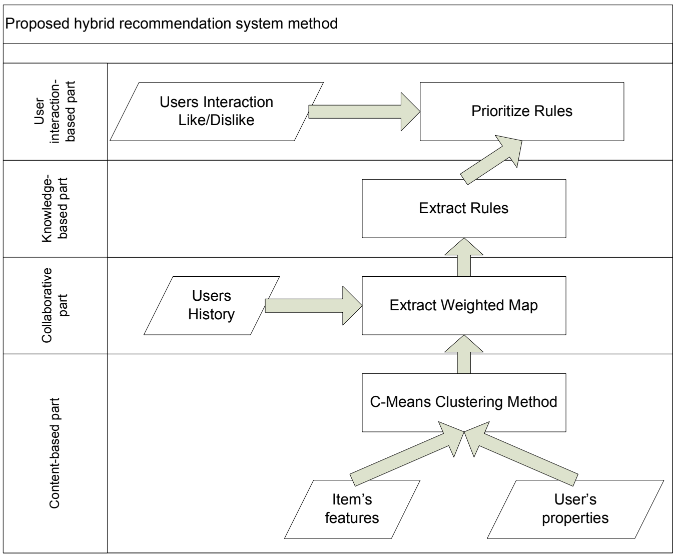

Figure 1 : Proposed Hybrid Recommendation Method

The key idea of content-based filtering approach is based on similarity of item's features. The best idea in content-based recommendation system, is to recommend similar items to user that buy the first item previously. This similarity factor can extract from two matrix show in Figure 2.

Item	Attr. X	Attr. Y	Attr. Z
A	X_A	Y_A	Z_A
B	X_B	Y_B	Z_B

User	Prop. X	Prop. Y	Prop. Z
A	X_A	Y_A	Z_A
B	X_B	Y_B	Z_B

Figure 3 : Content-Based recommendation Method

In collaborative filtering approach a user will prefer those items that like-minded people prefer. Therefore, a collaborative filtering recommender system (CFRS), makes prediction for a user based on the similarity between the interest profile of that user and those of other users. Suppose we have a database of user ratings for items, where users indicate their interest in an item using a numerical scale. Then, it is possible to define similarity factor between two user profiles (Figure 2).

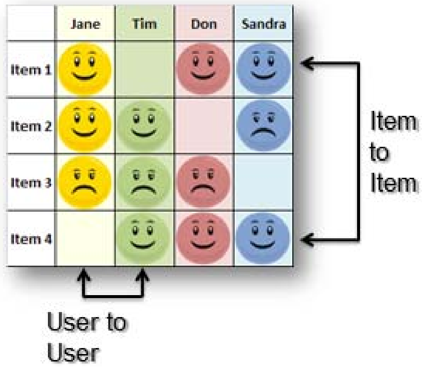

Figure 2 : Collabrative Filter Sample

Most recommendation systems use some methods to make groups of users or items or both, to recommend better. To make these groups, recommendation systems use classifying or clustering method. The main difference between clustering and classifying, is that in classifying methods, the classes are exist before items come but in clustering methods clusters create on items come. Of course this method should be a dynamic filtering, so it should use clustering method to group items.

The K-Means clustering method is one of the most famous and also most light clustering methods. This method can cluster items and users but after clustering by K-Means each item or users just belong to one cluster.

In fuzzy clustering, each item has a degree of belonging to clusters, as in fuzzy logic, rather than belonging completely to just one cluster. Thus, items on the edge of a cluster may be in the cluster to a lesser degree than items in the center of cluster. An overview and comparison of different fuzzy clustering algorithms is available [3].

A new fuzzy clustering method based on K-Means is C-Means method. The difference between K-Means and C-Means is that after clustering by C-Means, each item has a belonging degree to each cluster. So each item is belongs to each cluster with some probability degree. Thus, the system can overcome the cold start problem to recommend items that no one in the community has yet buy or rated [2]. If an item never buy but it is in some clusters with some relations to some clusters in users and if a user never buy anything but he/she is in some clusters on users have some relations with some clusters on items. So the system can recommend this item to this user (overcome the cold start problem).

Any item x has a set of coefficients giving the degree of being in the K_{th} cluster $W_k(x)$. With fuzzy C-Means, the centroid of a cluster is the mean of all items, weighted by their degree of belonging to the cluster:

$$C_k = \frac{\sum_x W_k(x)x}{\sum_x W_k(x)}$$

The degree of belonging $W_k(x)$ is related inversely to the distance from x to the cluster center as calculated on the previous pass. It also depends on a parameter m that controls how much weight is given to the closest center.

Each new item have some features enter while adding to system, although each user have some other features manually set in his/her profile.

First of all, the new method uses these features to cluster items and users using C-Means method (content-based part).

Now, it is some clusters on items and some clusters on users. To figure out the relations between these clusters, it uses users buy history.

Consider item I_a belongs to cluster C_i by $B(C_{ia})$ degree and belong to cluster C_i by $B(C_{ib})$ degree. User U_a belongs to cluster C_u by $B(C_{ua})$ degree and U_b belongs to cluster C_u by $B(C_{ub})$ degree. Consider user U_a buy I_a and user U_b don't buy any item and item I_b never buy by any users.
Now, cluster C_i on items has a relation to cluster C_u on users by $M(C_{iu})$ while:

$$M(C_{iu}) = B(C_{ua}) * B(C_{ia})$$

So, after generalization the formula described above:

$$M(C_{iu}) = \frac{\sum_x^{items} \sum_y^{users} B(C_{ux}) * B(C_{iy})}{m * n}$$

While, m represents the count of users and n represents the count of items.

Then, a rule generated for recommending items in cluster C_i on items to users in cluster C_u on users. So, user U_b have a relation with item I_b :

$$Rec(U_b, I_b) = M(C_{iu}) * B(C_{ib}) * B(C_{ub})$$

While, $Rec(U_b, I_b)$ it between *[0 , 1]* and create a weighted map. If $Rec(U_b, I_b)$ is bigger that a threshold the system recommend item I_b to user U_b. This threshold should figure out in user interaction interface (knowledge-based part).

At the start point of this system the recommendation threshold is zero. So, every $Rec(U_b, I_b)$ make a recommendation with a *like/dislike* button. If user U_b liked item I_b, it is inferred that this recommendation was good but if user U_b disliked item I_b, it is inferred that the recommendation is not good and the threshold should be bigger than $Rec(U_b, I_b)$, so the threshold updates to have better recommendations later (user interaction-based part).

Because of differences in user's preference this threshold should be measures from all users' opinions. The simplest method is to measure an average on every disliked $Rec(U_b, I_b)$ to have a better threshold.

After all, the knowledge base updates rules and prioritizes rules by the last recommendation result make by user interaction-based part.

IV. Conclusion

In this paper, we suggest a new hybrid recommendation system that combines content-based, collaborative and knowledge-based recommendation systems. We use C-Means Fuzzy clustering method to cluster the items and user profiles. After clustering, we use a rule generator engine to make rules from the clusters and take it to a knowledge-based part.

After all, we have an interactive user interface to see what user want and customize what the users want to see and make the most successful suggestions.

Knowledge-based recommendation systems key idea is to use rules to recommend different items to different users. Perhaps some rules has conflict or include others rules, so it is good to have different weights on these rules to prioritize them. One of the most reliable rating and weighting method is to interact with users to find which rule is more important. Now the importance of user interaction is magnified.

V. Future Works

Because of our automatic rule generation, these new rules may have some bugs that user can edit them. The future work for this paper is to make the rule rating and rule selection more intelligent. To do this, we need some data mining methods like association rules to find which rules fire more than the others and find the rule that make most successful suggestions.

Some other clustering methods can be used in this system to make the clusters more exact and perhaps smaller. It can help the recommendation system but it has more time complexity.

VI. Results

The method proposed in this paper, has used in a recommendation book system (http://www.flazx.ir). After two month, the website has 213 active users with 1734 books.

Table 1: FLAZX.IR Results

Table *1*shows the system results.
Where LF(Like Factor), stands for percentage of books which recommend to users and they like it. So it is proved that C-Means clustering method can work better than none-fuzzy clustering methods.

VII. Acknowledgements

Thanks to Mrs. Leila Nazari and Dr. Farbod Noorbehbahani for all their advices.
Of course, thanks to FLAZX.IR team that help us to test our method.

VIII. REFERENCES

1. JANNACH, DEITMAR; FELFERING, ALEXANDER; FRIEDRICH, GERHARD; ZANKER, MARKUS;. (2010). *RECOMMENDER SYSTEMS: AN INTRODUCTION.* CAMBRIDGE UNIVERSITY PRESS.

2. SCHEIN, A., POPESCUL, A., UNGAR, L.H., & PENNOCK, D.M. (2002). METHODS AND METRICS FOR COLD-START RECOMMENDATIONS. *PROCEEDINGS OF THE 25TH ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL* (PP. 253-260). ACM.

3. Nock, R. and Nielsen, F. (2006) "On Weighting Clustering", IEEE Trans. on Pattern Analysis and Machine Intelligence, 28 (8), 1–13

4. David W. McDonald and Mark S. Ackerman. (2000) " a flexible recommendation system and architecture", CSCW '00 Proceedings of the 2000 ACM conference on Computer supported cooperative work, 213-240

5. Qing Li. and Byeong Man Kim. (2003) "Clustering approach for hybrid recommender system", Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference Proceeding, 33-38

Table 1: FLAZX.IR Results

	proposed method with c-means	proposed method with k-means	random recommendation
Number of clusters of users	14	10	-
Number of clusters of books	56	48	-
LF (Like Factor)	78%	53%	32%