

# FHCC: A SOFT HIERARCHICAL CLUSTERING APPROACH FOR COLLABORATIVE FILTERING RECOMMENDATION

Kaiman Zeng<sup>1</sup>, Nansong Wu<sup>1</sup>, Xiaokun Yang<sup>1</sup>, Lu Wang<sup>2</sup>, and Kang K. Yen<sup>1</sup>

Department of Electrical and Computer Engineering,  
Florida International University, Miami, USA  
School of Science and Technology, Saint Thomas University, Miami, USA

## ABSTRACT

*Recommendation becomes a mainstream feature in nowadays e-commerce because of its significant contributions in promoting revenue and customer satisfaction. Given hundreds of millions of user activity logs and product items, accurate and efficient recommendation is a challenging computational task. This paper introduces a new soft hierarchical clustering algorithm - Fuzzy Hierarchical Co-clustering (FHCC) algorithm, and applies this algorithm to detect user-product joint groups from users' behavior data for collaborative filtering recommendation. Via FHCC, complex relations among different data sources can be analyzed and understood comprehensively. Besides, FHCC is able to adapt to different types of applications according to the accessibility of data sources by carefully adjust the weights of different data sources. Experimental evaluations are performed on a benchmark rating dataset to extract user-product co-clusters. The results show that our proposed approach provide more meaningful recommendation results, and outperforms existing item-based and user-based collaborative filtering recommendations in terms of accuracy and ranked position.*

## KEYWORDS

*Collaborative Filtering, Recommender system, Hierarchical clustering, Co-clustering*

## 1. INTRODUCTION

Recommendation service is gaining increasing attention in the big data era and has brought great benefit in e-commerce. Amazon, as the largest and most influential e-commerce company in the world, has successfully applied the recommendation in their online retail business. The recommendation system helps lower transaction costs and improves revenues in different ways, such as promoting cross-selling and upselling, and reducing labor cost of customer assistance by providing this self-service tools. Besides, it is capable of analyzing the ongoing consumer panel data, support marketing research and support product management. On the other hand, the recommendation system also benefits the consumers in a way that it facilitates the process of product search and discovery. It mitigates, if not overcomes, the problem of information overload by aiding customers in search and exploring new, relevant and interesting items (e.g., media, product, or service) and helps them identify which items are worth viewing in detail. One of the major features for a recommendation system is its serendipity, i.e., to help the users make fortunate discoveries that they were not explicitly looking for, as well as make personalized recommendations.

Recommender systems identify recommendations for a user based on recommending services' content, users' previous purchases, searches and other behaviors. Most recommendation systems take either of two basic approaches: content-based filtering or collaborative filtering. Content-based filtering is derived from information retrieval with a specific focus on long-term information filtering [1]. It recommends products which are the most similar to that the user interests in, with regard to the inner attributes (e.g. audio feature [2], image feature [3, 4] and textual information [5]) of the products. Collaborative filtering is a sort of word-of-mouth advertisement. It assumes that if user A and user B have similar tastes on a specific item, such as similar behaviors of purchasing, rating, or watching the same item, they may act similarly on other items. Then, the user-item ratings data will be used to make predictions and recommendations. The ratings by users are explicit indications on a certain scale, normally 1 to 5, while purchases and click-throughs are implicit indications [6, 7]. Collaborative filtering highly relies on user historical usage data, and hence suffers from the new user problem and the new item problem. The new user problem represents the lacking of accuracy of the recommendations received by a new user when a significant number of votes are not yet fed to the recommendation system's collaborative filtering core. The new item problem is that the first user who rates the new item cannot benefit from the rating. The problem causes new items to be ignored until a substantial number of ratings are received for the same item [8]. Content-based filtering does not incorporate the similarity of preference across individuals. Additionally, hybrid approaches, such as the content-boosted content filtering algorithm and personality diagnosis, combine content-based filtering and collaborative filtering. Therefore the limitations of either approach are avoided, and the performance of recommendation is also improved [9].

Recommending products require understanding large-scale data of user logs and product records, such as user purchase patterns, product attributes, price ranges, and product categories. Given hundreds of millions of user activity logs and product items, accurate and efficient recommendation is a challenging computational task. Data mining provides possible tools to tackle this problem. These algorithms include clustering, classification techniques, the generation of association rules, and the construction of similarity graphs. In this paper, a novel method called Fuzzy Hierarchical Co-Clustering (FHCC) is developed to build hierarchical co-clusters of users and products, and then perform recommendation based on the similarity between the information of a new user and the properties of the clustered user-product groups. Inspired by approaches in [10], we design FHCC as follows: first the algorithm begins from singleton clusters. In the next steps, every two nearest clusters are repeatedly merged into one until there is only one cluster remaining. In our case, FHCC can merge a subset of the users with a subset of the products based on their internal similarity measurement in each step of the merging process. In addition, since the users and products could be related to different clusters in practice, we extend the hard HCC with fuzzy set theory to support soft clustering, in which each object can belong to one or more clusters based on the similarity measurement. We formalize each user-product instance as a three-dimensional virtual vector. To calculate the similarities between different instances, we evaluate the similarity score for each component in the virtual vector, and finally combine them into a hybrid similarity measure.

The paper is organized as follows. In Section 2, we have a literature review of different recommendation techniques. Section 3 discusses the design of the proposed FHCC algorithm. Section 4 presents experimental results and analysis. Finally, we conclude with the paper and discuss future works in Section 5.

## **2. RELATED WORK**

One of the successful pioneers in recommender technologies is collaborative filtering (CF) [11-13]. CF starts by constructing a database based on individual consumers' preferences of products. Statistical data is used to find consumers, who have history similar to the target customer. For

example, they rate different products similarly and buy similar set of products. These set of customers are called neighbors. Once a neighborhood of customers is formed, these systems use algorithms to produce recommendations, and then recommend products to target customers based on the opinions of others.

In the recommender systems, clustering is one of the data mining techniques that are usually used to form neighborhoods. Compared with other CF techniques, such as methods based on correlation criteria, non-negative matrix factorization, or singular value decomposition, clustering methods has lower computational cost, and thus have received extensive research from a number of different recommender domains. With the purpose of discovering natural or meaningful groups, the unsupervised learning assigns items to groups based on similarity. The similarity is determined by a distance measure, such as Euclidean distance, Minkowski distance, Mahalanobis distance, Cosine similarity, Pearson correlation, and so on. The clustering algorithm can minimize intra-cluster distances and also maximize inter-cluster distances.

Partitional clustering and hierarchical clustering are two major types of clustering methods. Partitional clustering algorithms divide data items into a certain number of disjoint or overlapping clusters. Hierarchical clustering algorithms consecutively cluster items in the clusters it detected, and produce a set of nested clusters that are organized into a hierarchical tree. A classic partitioning algorithm called *K*-means clustering partitions a dataset into predefined *k* clusters by minimizing the distance between each data point. Ungar and Foster [14] presented a repeated *k*-means and Gibb sampling clustering technique to cluster users with similar items. Xue et al. [15] proposed a later commonly used clustering method in the context of recommendation systems. This method introduces the *k*-means algorithm as a pre-processing step to help forming the cluster neighborhood. The distances between the user and the centroids of different cluster are used as the pre-selection criterions for neighbors. They also suggested a cluster-based smoothing technique. In this technique, the missing values for users in a cluster are replaced by cluster representatives. Their method is reported of performing slightly better than classic *k*NN-based collaborative filtering. Similar to Xue's method, Sarwar et al. [16] applied the bisecting *k*-means algorithm to partition the user space into clusters in order to form neighborhood in the next step. Their approach proved a significant improvement in efficiency over traditional *k*NN collaborative filtering, and provides comparable recommendation quality. In literature [17] O'Connor and Herlocker performed clustering on items instead of users. They compared four algorithms: average link hierarchical agglomerative clustering, robust clustering algorithm for categorical attributes, *k*Metis, and *h*Metis using the Pearson Correlation similarity measure.

Hierarchical clustering generates a set of nested clusters organized as a hierarchical tree or dendrogram. Since any desired number of clusters can be obtained by selecting the hierarchical tree at the proper level, assuming the number of clusters beforehand is unnecessary. Hierarchical clusters can sometimes correspond to meaningful taxonomies. Traditional hierarchical algorithms merge or split one cluster at a time according to a similarity or distance matrix. Agglomerative hierarchical clustering and divisive hierarchical clustering are the two major approaches. The agglomerative hierarchical clustering starts with the points as individual clusters. Then it consecutively merges the closest pair of clusters until only one cluster is left. The divisive hierarchical clustering starts with a single all-inclusive cluster. Then it successively splits the clusters until each cluster contains only one point. Merialdo [18] proposed a hierarchical clustering algorithm to cluster users and items into two independent cluster hierarchies. In the cluster hierarchies, the author used the nodes on the path from the roots to the leaves. The recommendation is made by the calculated weighted sum of the defined centers of these nodes.

Co-clustering is the clustering of multiple types of data. The co-clustering technique performs better to handle the sparse and high-dimensional matrices than traditional clustering [19-22]. Dhillon [23] presented a method to co-cluster words and documents based on bipartite spectral

graph partitioning. Long et al. [24] introduced a general principled model, called Relation Summary Network, for co-cluster heterogeneous data presented as a k-partite graph. Hierarchical clustering deals with only one type of data and co-clustering produces just one level of data organization. Hierarchical co-clustering aims at simultaneously construction of two or more hierarchies [25, 26]. These approaches are typically preferences in biological and medical sciences [27, 28]. Co-clustering appears under the term ‘bi-clustering’ in these disciplines. Vlachos et al. [29] proposed a co-clustering algorithm based on *k*-means and agglomerative hierarchical clustering approaches. Their approach analyzed and visualized the connections between users and items for ranked product recommendations.

### 3. METHOD

#### 3.1. Problem Formulation

Generally speaking, the rating data from users contains three types of resources: users, products, and products’ ratings reviewed by users. An example of rating data is depicted in Table 1.

Table 1. An example of rating data

| User \ Product | p <sub>1</sub> | p <sub>2</sub> | p <sub>3</sub> | p <sub>4</sub> | p <sub>5</sub> |
|----------------|----------------|----------------|----------------|----------------|----------------|
| u <sub>1</sub> | 5              |                | 5              |                | 4              |
| u <sub>2</sub> |                | 4              |                | 3              |                |
| u <sub>3</sub> |                |                | 4              |                | 4              |
| u <sub>4</sub> | 3              |                |                |                | 1              |
| u <sub>5</sub> |                | 4              | 5              | 4              |                |

With such data structure, we can formalize the recommendation problem as follows: assume we are given a set of *m* users  $U = \{u_1, u_2, \dots, u_m\}$ , and a set of *n* products  $P = \{p_1, p_2, \dots, p_n\}$ .

We are also given  $m \times n$  rating matrix  $R = (r_{ij}) \in \mathbb{S}^{m \times n}$ , where  $r_{ij}$  is the rating score that the *i*th user in *U* assigned to the *j*th product in *P*. Our objective is to design a recommendation model that could perform the following functions:

- 1) Simultaneously we generate a hierarchical clustering of *U* and of *P* based on matrix *R*. Each cluster contains a subset of these two data sources, and can be regarded as a potential community group underlying the users rating data.
- 2) When a new user joins, along with his/her profile  $u'$  and his/her interested product  $p'$ , the system should be capable of recommending a series of products. The recommended products are based on the similarities between  $u'$  and *U*,  $p'$  and *P*.

We design and propose the following method to solve the problem and achieve the goals mentioned above:

- 1) We propose a soft agglomerative clustering algorithm – Fuzzy Hierarchical Co-clustering to recover the hidden co-clusters by recognizing the bipartite user-product rating matrix;

- 2) Based on the recovered co-clusters, hybrid similarity measurement is used to obtain the average similarities between the new user, his/her interested product and the co-clusters. Then we extract the potential recommendations from the most similar co-cluster.
- 3) We rank these recommendation candidates from high to low according to the product rating predications.

### 3.2. Fuzzy Hierarchical Co-Clustering

Traditional hierarchical co-clustering [10, 29] is a hard clustering technique, in which each item can only be assigned into one co-cluster. In the real world, people would be interested in various different categories of products, which can be revealed via their behaviors like search, click-on, purchase and rate. Therefore, users should be grouped to more than one interest community. It is a non-trivial task to decide which group a user belongs to, and it is not reasonable to assign a user showing interests in different products to only one group. To address this issue, a possible solution is to utilize soft clustering techniques on users rating data so that after clustering, a user might belong to multiple groups. In this study, we propose a novel hierarchical co-clustering algorithm with fuzzy set theory to support the soft clustering, in which each entity belongs to multiple groups, for our recommendation purpose. Membership levels that indicate the strength of the association between the data element and the cluster are assigned to each entity. Specifically, we first represent the users rating data as three-dimension virtual vectors, and then perform hierarchical clustering on these vectors by virtue of a hybrid similarity measurement. The basic steps of the FHCC are designed as follows:

Step 1: The initial co-cluster is formed by assigning each user to his/her rated product. If there are  $N$  rating scores, then there are  $N$  co-clusters, each containing just one user and one product. Let the similarities between the co-clusters defined as a hybrid similarity of three components that a co-cluster entity contains.

Step 2: The pair of co-clusters with highest similarity is discovered and merged into a single co-cluster. Therefore the number of co-clusters is reduced by one.

Step 3: The similarities between the new co-cluster and each of the old co-clusters are computed. In our study, we use single-link clustering, also known as connectedness method, to represent the new similarities. The new similarity between two clusters is equal to the greatest similarity of any two individual members from each of the two clusters.

Step 4: Step 2 and Step 3 are repeated until initial co-clusters are entirely clustered into a single co-cluster. It requires  $N-1$  rounds of iterations.

The algorithm is described in Algorithm 1. A sample of FHCC clustering is visualized as a dendrogram in Figure 1. The horizontal lines represent the merging of clusters. The y-coordinate represents the similarity of the two co-clusters that are merged. In general, there are various types of measures to evaluate the similarity of two clusters, such as Euclidean Distance, Cosine Distance, Jaccard Similarity and so on. However, none of them can evaluate the pairwise similarity of multi-source entities. To address this issue, we propose a hybrid similarity measurement, consisting of three different similarity computations for the corresponding components. Given two three-dimensional vectors  $e_i(u, p, r)$  and  $e_j(u', p', r')$ , the similarity between them can be calculated as

$$Sim(e_i, e_j) = w_1 \cdot Sim^U + w_2 \cdot Sim^P + w_3 \cdot Sim^R$$

Here, the weights  $w_1$ ,  $w_2$ , and  $w_3$  are the three control parameters, indicating how much we trust the corresponding components. In general they can be tuned appropriately to different applications.  $Sim^U$  represents the similarity between two user sets,  $Sim^P$  denotes the similarity between two product sets, and  $Sim^R$  evaluates how similar the two user sets rates the corresponding product sets.

---

**Algorithm 1** The FHCC Algorithm Description.

---

**Input:**  $m$  users set  $U$ ,  $n$  products set  $P$ , and rating matrix  $R$ .

**Output:** Dendrogram of clusering results.

**Initialization:**

Create an empty hierarchy  $H$

**for**  $i = 1$  to  $m$  **do**

**for**  $j = 1$  to  $n$  **do**

**if**  $r_{ij} > 0$ , **then**

            Create node  $e = \{u_i, p_j\}$

**end if**

**end for**

**end for**

$List \leftarrow \{e \in E\}$

$N \leftarrow \text{sizeof}[E]$

Add  $List$  to  $H$  as the bottom layer

**Hybrid similarity:**

**for**  $i = 1$  to  $N$  **do**

**for**  $j = 1$  to  $N$  **do**

$$Sim(e_i, e_j) = w_1 \cdot Sim^U + w_2 \cdot Sim^P + w_3 \cdot Sim^R$$

**end for**

**end for**

Sort  $E$

**Clustering:**

**for**  $i = 1$  to  $N$  **do**

    Choose the pair of nodes that of highest similarity

    Merge them into a new node  $e = Merge(u, p)$

    Remove  $\{u, p\}$  from  $List$  and add  $e$  to  $List$

    Add  $List$  to  $H$  as the next layer

**end for**

---

$Sim^U$ : To evaluate the similarity between two user sets, we need to analyze what types of information in the users' profiles we can utilize. In general, the user profile may provide a couple of optional contents, such as self-definition, interests, job titles and so on. With such information, we treat the similarity computation as a multi-attribute comparison problem. For each attribute, we analyze the overlapping between two values and get a similarity score restricted in  $[0, 1]$ . The similarity between two different user profiles can be obtained by assigning different weights on different attribute comparisons, and sum them up. If the user sets contain multiple users,  $Sim^U$  is finally calculated as the average similarity among all the user pairs.

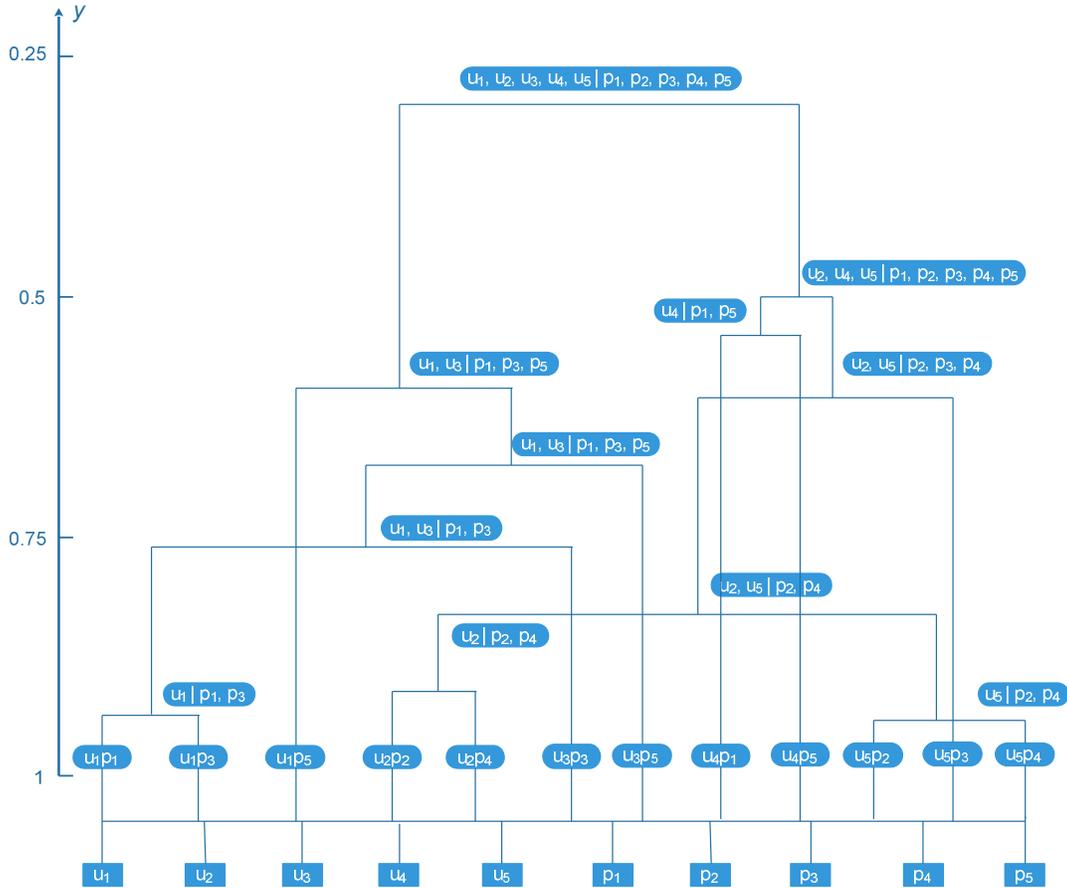


Figure 1. A sample of FHCC clustering dendrogram

$Sim^P$ : The similarity between two product sets also relies on what product information is applicable. In general, a product could contain information of product name, brand, category, description and so on. Similar to the  $Sim^U$ , similarity of product is computed as a multi-attribute overlapping measure. Also, if there are multiple products in the product sets, the average similarity score is regarded as the value of  $Sim^P$ .

$Sim^R$ : To evaluate how similar rate sets are, we simply use the normalized difference of two rating scores.

Each similarity mentioned above is restricted in the range  $[0, 1]$ , and the weights  $w_1, w_2$ , and  $w_3$  are also normalized in the range  $[0, 1]$ .

### 3.3. Personalized Recommendation

Assume we have detected several user-product groups by adopting the approaches mentioned above, say  $C = \{C_1, C_2, \dots, C_n\}$ . It is obtained by cutting the clustering dendrogram at a pre-specified level of similarity. For example, we cut the dendrogram at 0.4 if we want clusters with a minimum similarity of 0.4. The higher the similarity is chose, the less the cluster groups achieves. Given a new user with a new interested product  $\langle u', p' \rangle$ , we need to compare  $\langle u', p' \rangle$  with user profile sets and product sets in each co-cluster group. To do so, we adopt the hybrid

similarity measurement only with component  $U$  and  $P$  proposed in Section 3.2. We can obtain the closest group for the new user. Then we rank the products within the closest group by the predicted rating scores and finally select the top  $k$  products as the recommendation result.

## 4. EXPERIMENT

In this section, we empirically analyze the proposed methods using a benchmark dataset to assess the performance of them. We compare the recommendation performance of the proposed soft hierarchical co-clustering algorithm FHCC with the recommendations derived via traditional techniques based on  $k$ -means CF and association rules, which illustrates the ability of the proposed algorithm to reveal patterns hidden in the data. MovieLens 100K [30] - a benchmark dataset for recommendation systems is employed to evaluate the performance of our algorithm. This data set contains 943 users who have rated at least 20 movies on the scale of 1 to 5, a total of 1682 movies and 100,000 ratings.

### 4.1. Metric

In this research, the recommendation system is designed to recommend the most likely high-rated products to users. In the experiment, we mask 20% of the actual scores in the rating matrix to evaluate our extracted data model, and use the remaining 80% for training. For each user, we examined top-5 and top-10 recommendations. We define the performance evaluation as a binary class problem, and make some assumption on the experiment dataset. The movies that are actually rated in the recommendation list will be considered as true positive results. Otherwise, if the user does not rate a movie, we assume that he/she is not showing interests in it. The averaged F1-score and averaged NDCG is calculated for the testing users as comparisons. F1-score is defined as

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

And  $NDCG$  at position  $k$  is defined as

$$NDCG @ k = \frac{\sum_{i=1}^k \frac{2^{rel_i} - 1}{\log_2(i+1)}}{IDCG}$$

In our scenario,  $rel_i = 1$  if the user has rated the recommended movie and 0 otherwise.

### 4.2. Behavior of the Recommendation System

Table 2. Performance comparison of different collaborative filtering methods

| Method               | User based CF | Item based CF | FHCC CF |
|----------------------|---------------|---------------|---------|
| <b>prec@5</b>        | 0.35136       | 0.31157       | 0.43853 |
| <b>prec@10</b>       | 0.31948       | 0.28493       | 0.35011 |
| <b>recall@5</b>      | 0.14074       | 0.10936       | 0.23909 |
| <b>recall@10</b>     | 0.21633       | 0.16243       | 0.21272 |
| <b>F1-score</b>      | 0.22947       | 0.18440       | 0.28705 |
| <b>Averaged NDCG</b> | 0.59215       | 0.55374       | 0.63379 |

Table 2 presents the recommendation performance of proposed method and the traditional user-based and item-based CF methods. We compare the recommendation precision and recall at the top-5 and top-10 results, denoted by  $\text{prec}@5$ ,  $\text{prec}@10$ ,  $\text{recall}@5$ , and  $\text{recall}@10$ , respectively. Precision reflects the ability of recommendation system to eliminate irrelevant items, and recall measures the ability of recommendation system to return all items that user may find interesting.

As we can see, at the top-5 results, proposed FHCC CF achieves 0.43853 of precision, better than 0.35136 of user based CF, and 0.31157 of item based CF. The recall of proposed FHCC CF is also superior to the other two methods at top-5. At the top-10 results, FHCC CF still outperforms the user based and item based CF in terms of precision. With respect to recall, user based CF gains the best result of 0.21633, while the proposed FHCC CF performs 0.21272, which is comparative to the best. Then, we use F1-score to evaluate the balance of precision and recall.

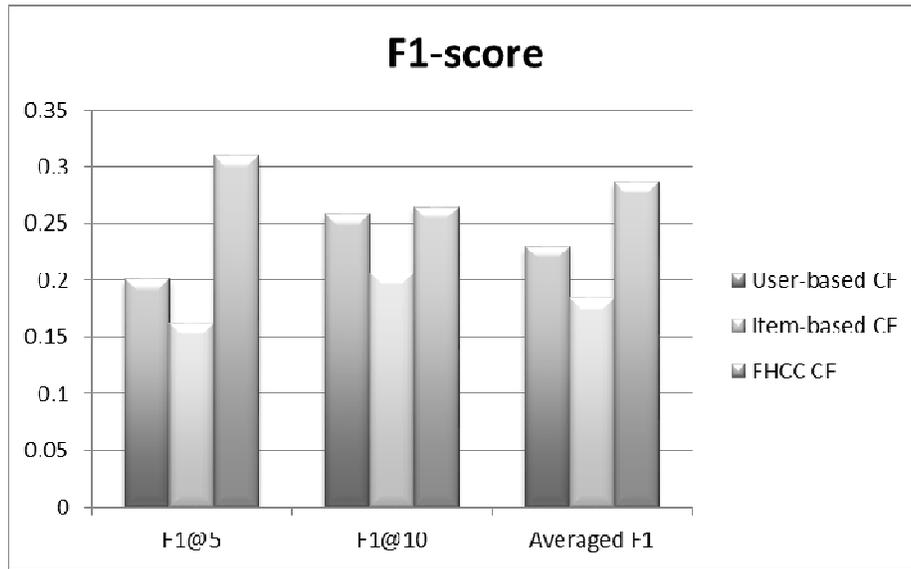


Figure 2. F1-score comparison

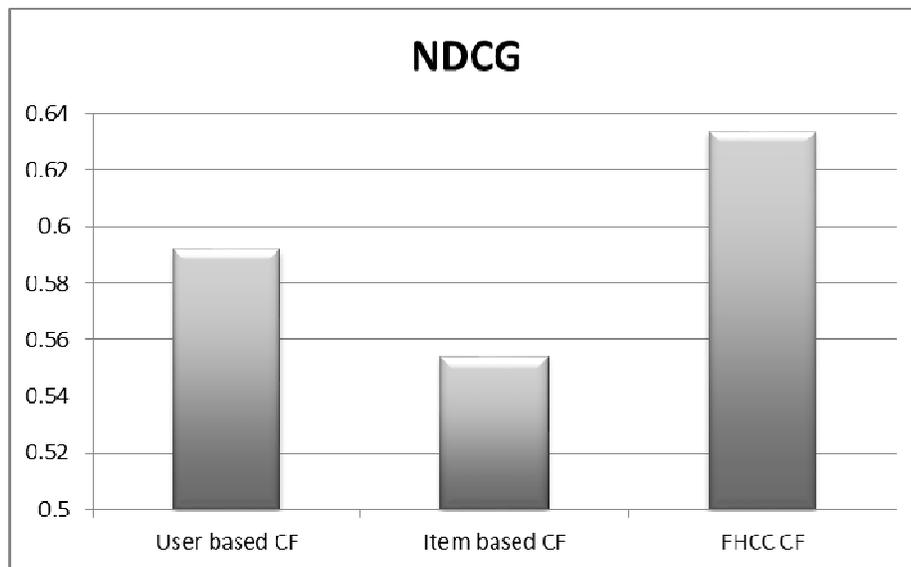


Figure 3. Averaged NDCG comparison

To sum up, it is apparently that the FHCC CF outperforms other two traditional CF methods from both accuracy and ranking perspectives. This is because in the proposed model, two different data sources, users and product items, are hierarchically clustered simultaneously, which discovers the

relationship between the users and products within the same cluster, leads to a more meaningful group detection than other approaches, and hence makes the recommended results more accurate. Besides, the proposed recommendation framework is more flexible than the traditional recommenders. In our method, a set of control parameters is provided to adjust the weights of user and product components according to different datasets. For example, under the scenario that users provide very limited personal information, we could degrade the weight of user profile similarity computation in the total hybrid similarity computations. It depends on what kind of business dataset we are working on, and how much information is accessible and available to be analyzed for the recommendation purpose, which makes our method more flexible to different applications.

## 5. CONCLUSIONS

In this paper, we have introduced a soft hierarchical co-clustering technique for binary matrices. We address the issue of detecting user-product groups by taking advantage of clustering techniques. Inspired by agglomerative hierarchical clustering approaches, our method is extended with fuzzy set theory to achieve higher flexibility. We have explicitly shown how our method can be coupled with a recommendation system that merges derived co-clusters and individual customer information, as well as the product properties for ranked recommendations. Therefore the method can serve as an interactive tool for examining hypotheses on product offerings. In addition, our approach can assist in the visual identification of market segments to which specific focus should be given, e.g., co-clusters with high propensity for buying emerging products, or products with high profit margin. Empirical experiments conducted on MovieLens 100K dataset demonstrate the effectiveness of our proposed recommendation framework.

Future work will focus on enriching the content of different data sources in order to understand their inner correlations more comprehensively. For instance, we could take into consideration of document similarity measure for product descriptions and user comments. Besides, with the increase of the amount of user rating data, especially documental data, the pairwise similarity comparisons become computational expensive. We will also work on techniques to accelerate the similarity computation and to handle the largest-scale issue in the future.

## REFERENCES

- [1] Pazzani, M.J. and D. Billsus, Content-based recommendation systems, in *The adaptive web*. 2007, Springer. p. 325-341.
- [2] Li, T. and L. Li, Music data mining: An introduction. *Music data mining*, 2011: p. 1.
- [3] Datta, R., J. Li, and J.Z. Wang. Content-based image retrieval: approaches and trends of the new age. 2005. ACM.
- [4] Zeng, K., N. Wu, and K.K. Yen, A Color Boosted Local Feature Extraction Method for Mobile Product Search. *Int. J. on Recent Trends in Engineering and Technology*, 2014. 10(2): p. 78-84.
- [5] Mooney, R.J. and L. Roy. Content-based book recommending using learning for text categorization. in *Proceedings of the fifth ACM conference on Digital libraries*. 2000. ACM.
- [6] Miller, B.N., J.A. Konstan, and J. Riedl, PocketLens: Toward a personal recommender system. *ACM Transactions on Information Systems (TOIS)*, 2004. 22(3): p. 437-476.
- [7] Ansari, A., S. Essegaiier, and R. Kohli, Internet recommendation systems. *Journal of Marketing research*, 2000. 37(3): p. 363-375.

- [8] Adomavicius, G. and A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 2005. 17(6): p. 734-749.
- [9] Burke, R., Hybrid web recommender systems, in *The adaptive web*. 2007, Springer. p. 377-408.
- [10] Li, J., et al., Hierarchical co-clustering: a new way to organize the music data. *Multimedia, IEEE Transactions on*, 2012. 14(2): p. 471-481.
- [11] Herlocker, J.L., et al. An algorithmic framework for performing collaborative filtering. in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. 1999. ACM.
- [12] Konstan, J.A., et al., GroupLens: applying collaborative filtering to Usenet news. *Communications of the ACM*, 1997. 40(3): p. 77-87.
- [13] Shardanand, U. and P. Maes. Social information filtering: algorithms for automating “word of mouth”. in *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1995. ACM Press/Addison-Wesley Publishing Co.
- [14] Ungar, L.H. and D.P. Foster. Clustering methods for collaborative filtering. in *AAAI workshop on recommendation systems*. 1998.
- [15] Xue, G.-R., et al. Scalable collaborative filtering using cluster-based smoothing. in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. 2005. ACM.
- [16] Sarwar, B.M., et al. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. in *Proceedings of the fifth international conference on computer and information technology*. 2002. Citeseer.
- [17] O’Connor, M. and J. Herlocker. Clustering items for collaborative filtering. in *Proceedings of the ACM SIGIR workshop on recommender systems*. 1999. Citeseer.
- [18] Merialdo, A.K.-B., Clustering for collaborative filtering applications. *Intelligent Image Processing, Data Analysis & Information Retrieval*, 1999. 3: p. 199.
- [19] Zhang, D., et al., Cold-start recommendation using bi-clustering and fusion for large-scale social recommender systems. *Emerging Topics in Computing, IEEE Transactions on*, 2014. 2(2): p. 239-250.
- [20] Leung, K.W.-T., D.L. Lee, and W.-C. Lee. CLR: a collaborative location recommendation framework based on co-clustering. in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 2011. ACM.
- [21] Chen, G., F. Wang, and C. Zhang, Collaborative filtering using orthogonal nonnegative matrix tri-factorization. *Information Processing & Management*, 2009. 45(3): p. 368-379.
- [22] Deodhar, M. and J. Ghosh. A framework for simultaneous co-clustering and learning from complex data. in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2007. ACM.
- [23] Dhillon, I.S. Co-clustering documents and words using bipartite spectral graph partitioning. in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. 2001. ACM.
- [24] Long, B., et al. Unsupervised learning on k-partite graphs. in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006. ACM.

- International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.6, No.3, May 2016
- [25] Hosseini, M. and H. Abolhassani, Hierarchical co-clustering for web queries and selected urls, in *Web Information Systems Engineering–WISE 2007*. 2007, Springer. p. 653-662.
  - [26] Ienco, D., R.G. Pensa, and R. Meo, Parameter-free hierarchical co-clustering by n-ary splits, in *Machine Learning and Knowledge Discovery in Databases*. 2009, Springer. p. 580-595.
  - [27] Madeira, S.C. and A.L. Oliveira, Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 2004. 1(1): p. 24-45.
  - [28] Tanay, A., R. Sharan, and R. Shamir, Biclustering algorithms: A survey. *Handbook of computational molecular biology*, 2005. 9(1-20): p. 122-124.
  - [29] Vlachos, M., et al. Improving Co-Cluster Quality with Application to Product Recommendations. in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. 2014. ACM.
  - [30] MovieLens Dataset. Available from: <http://grouplens.org/datasets/movielens/>.