Multi-Clustering in Fast Collaborative Filtering Recommender Systems

Urszula Kużelewska

Bialystok University of Technology,
15-351 Bialystok, Wiejska 45a, Poland
Email: u.kuzelewska@pb.edu.pl

***Abstract*—Searching the huge amount of information available on the Internet is undoubtedly a challenging task. A lot of new Web sites are created every day, containing not only text, but other types of resources: e.g., songs, movies or images. As a consequence, a simple search result list from search engines becomes insufficient. Recommender systems are the solution supporting users in finding items, which are interesting for them. These items may be information as well as products, in general. The main distinctive feature of recommender systems is taking into account the personal needs and tastes of users. Collaborative filtering approach is based on users' interactions with the electronic system. Its main challenge is generating on-line recommendations in reasonable time when coping with a large data size. Appropriate tools to support recommender systems in increasing time efficiency are clustering algorithms, which find similarities in off-line mode. Commonly, this causes a decrease in prediction accuracy of the final recommendations. This article presents a high time efficiency approach based on multi-clustered data, which avoids negative consequences. The input data is represented by clusters of similar items or users, where one item or user may belong to more than one cluster. When recommendations are generated, the best cluster for the user or item is selected. The best cluster means that the user or item is the most similar to the center of the cluster. As a result, the final accuracy is not decreased.**

***Index Terms*—Recommender systems; Multi-clustering; Collaborative filtering.**

I. Introduction

Recommender Systems (RS) are electronic applications with the aim to generate for a user a limited list of items from a large item set. The list is constructed basing on the active user's and other users' past behaviour. People interact with recommender systems by visiting web sites, listening to music, rating items, doing shopping, reading items' description, selecting links from search results. This behaviour is registered as access log files from Web servers, or values in databases: direct ratings for items, the numbers of song plays, content of shopping basket, etc. After each action users can see different, adapted to them, recommendation lists depending on their tastes [1].

Recommender systems are used for many purposes in various areas. They offer great opportunities for business, government, education, e-commerce, leisure activities and other domains, with successful developments in commercial applications [2]. Recommender systems are often used in e-shops proposing products, which are the most similar to the content of customers' shopping baskets. Some examples include: a shopping assistant on website Qwikshop.com [3] and a mobile personalized recommender system to suggest new products to supermarket shoppers [4]. A practical example is also What2Buy [5] which contains results of a recommender system deployed in an e-store. Multimedia services, such as Netflix [6] or Spotify [7], are places, where recommendations are extremely helpful. A music recommender is described in [8] and a method applied in MoveLens system in [9].

Scalability and performance are key metrics for deploying a recommender system in a real environment [10]. Although they are precise, CF techniques are not time effective, because they calculate items for suggestion by searching similar users or items in the whole archived data. They deal with large amount of dynamic data, however the time of results generation should be reasonable to apply them in real-time applications. A user reading news expects to see the next offer for him/her in seconds, after analysis of millions of archived news.

Clustering algorithms can be used to increase neighbour searching efficiency and thus to decrease the time of recommendations generation. A drawback is that the quality of predictions is usually slightly reduced in comparison to k-Nearest Neighbours (kNN) neighbourhood identification strategy [11] [12]. The reason is due to the way clustering algorithms work. The typical approach is based on one partitioning scheme, which is generated once and then not updated significantly. The neighbourhood of data located on borders of clusters is not modelled precisely (see Figure 1).

To improve the quality of the neighbourhood modelling one can use multiple clustering schemes and select the most appropriate one to the particular data object. As a result, multi-clustering approach eliminates the inconvenience of decreased quality of predictions while keeping a high time effectiveness. Figure 2 presents two different clustering results for the same dataset. For a particular data object, one can select the scheme with this object located closer to the cluster center, thus having more neighbours around.

This paper contains results of experiments on a collaborative filtering recommender system, which is based on similarities among items identified a priori as multi-clusters. The aim of the experiments is to improve quality of recommendation systems (which is typically measured by Root Mean Squared

 ISBN: 978-1-61208-668-2

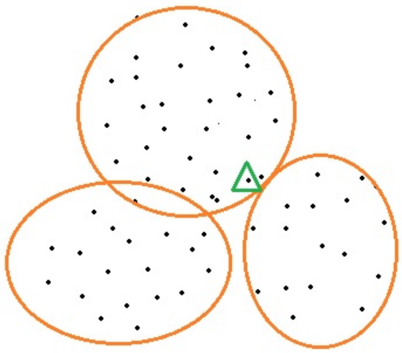

Fig. 1. Inadequate neighbourhood modelling for data located on cluster border in case of conventional k-means clustering

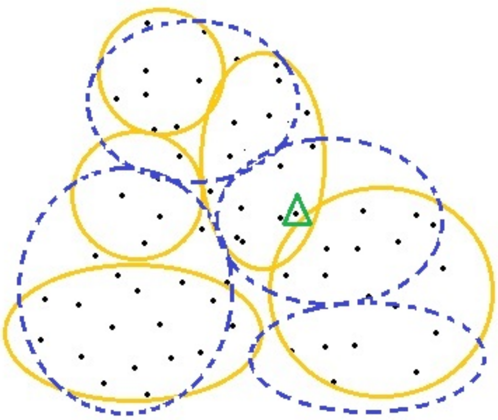

Fig. 2. Various neighbourhood modelling for particular data in case of k-means multi-clustering

Error - RMSE - of estimated vs real ratings) as well as to maintain a short time for recommendations generation (due to a real application - recommender systems in WWW services).

The set of clustering schemes was generated by k-means algorithm with the same values of their input parameters at every time. While searching the most similar items, every cluster is examined, and the one in which the appropriate items are the most similar to the center is selected.

The rest of the paper is organised as follows: Section II describes general aspects in recommendation systems, including problems in this domain and the role of clustering and multi-clustering algorithms. This section contains a description of related work, as well. The following section, Section III describes the proposed approach, whereas Section IV contains the results of the performed experiments. The last section concludes the paper.

II. Determining Neighbourhood in a Collaborative Filtering Domain - Related Work

Collaborative Filtering (CF) techniques search similarities among users or items based on archives of registered users' behaviour. As an example, similar users have mostly the same products in their baskets and similar items are bought by the same customers. Collaborative filtering methods can be classified into model-based and memory-based. The first category builds a model on the ratings, which is then used in the process of generating recommendations. The other category calculates recommendations by searching similar users or items in the whole archived dataset.

Recommender systems face many challenges and problems. The most important one, from the point of view of on-line recommendations, is scalability. Despite dealing with large amounts of dynamic data, recommender systems should generate results reasonably fast to be used in real-time applications. Nowadays, internet users are used to immediate displaying of each website. The most effective recommender systems are hybrid approaches, which combine at least two different methods that are complement to one another. Complementarity means that if one of the methods has a drawback or weakness in a certain area, then the other one has the considered features strong [1]. Clustering algorithms are good tools to analyze the neighborhood before a proper recommendation process, positively influencing its scalability [11].

A. Clustering Methods

Clustering has been and continues to be an important subject of research in the area of recommender systems [12]. The most often used method in memory-based collaborative filtering to identify neighbours is the kNN algorithm, which requires calculating distances between an active user and all the registered ones. In contrast, clustering (in model-based collaborative filtering) reduces computation time, due to the introduction of clusters models.

There are two approaches, which apply clustering in the recommender systems domain, namely: Cluster-based and Cluster-only [13]. In both, the computation efficiency of systems increases as the clustering phase is performed off-line. The first approach is the most common one and focuses only on time efficiency improvement; this is achieved by the application of clustering to find the neighbourhood of active users. Further generation of a recommendation list for the particular active user is performed by memory-based collaborative filtering methods. The process is executed on part of input data and identifies the most similar cluster. The final precision of recommendations can be lower in comparison with memory-based collaborative filtering. The second approach uses clustering as a main module of a recommender system. The partitioning applied on input data builds its model and further calculations are performed only on this model. The final precision of recommendations can be also lower in comparison with memory-based collaborative filtering methods.

B. Multi-Clustering Model of Data

Multi-clustering or alternative clustering is variously defined in literature. This term is usually used to describe the clustering, which is different than a typical partitioning based only on a single scheme. It can be a technique which tries to find different partitioning schemes on the same data, as well as a method that combines the results from clustering items' description with clustering users' demographic data. Bailey [14] provided a thorough survey on alternative clustering methods.

An example of multi-clustering algorithm is COALA (Constrained Orthogonal Average Link Algorithm) [15] that searches for alternative clusterings of better quality and dissimilarity with respect to the given clustering. It starts by treating each object as a single cluster and then iteratively merges a pair of the most similar clusters. The idea bases on $cannot - link$ constraints that guide the generation of a new, dissimilar clustering. Another example is MSC (Multiple Stable Clusterings) [16] that generates stable multiple clusterings. The advantage of this method is that it does not require to specify the number of clusters and provides users a feature subspace to understand each clustering solution.

The advantages of multi-clustering methods can be beneficial to the recommender systems domain. The better quality of the neighbourhood modelling leads to high quality of predictions keeping high time effectiveness provided by clustering methods. Despite of this, there are few publications describing application multi-clustering methods in recommendations.

The method described in [17] combines content-based and collaborative filtering approaches to recommendations. The system uses multi-clustering, however it is interpreted as a single scheme clustering on the following input data: items' description, users' information and item-user ratings matrix. It groups the items and the users based on their content, then uses the result, which is represented by the fuzzy set, to create an item group-rating matrix and a user group-rating matrix. As a clustering algorithm, it uses k-means combined with a fuzzy set theory to represent the level of membership (which is a number from the interval [0,1]) an object has to the cluster. Then, finally, the prediction rating matrix is calculated to represent the whole dataset. In the last step of this process, k-means is used again on the new rating matrix to find a group of similar users. The groups represent the neighbourhood of users in order to reduce a search space for collaborative filtering method.

Another algorithm is presented in [18]. The authors observed that users might have different interests over topics, thus might share similar preferences with different groups of users over different sets of items. The Co-Clustering For Collaborative Filtering (CCCF) method first clusters users and items into several subgroups, where each subgroup includes a set of like-minded users and a set of items in which these users share their interests. The groups are analysed by collaborative filtering methods and the resulting recommendations are aggregated over all the subgroups.

III. The Algorithm Used in the Experiments

The recommendation algorithm proposed in this article is composed of two steps. The first step (off-line) prepares the set of neighbourhoods of the most similar users in the form of a set of many clustering schemes. The method k-means is run several times with the same value of the k parameter. The results are stored as an input for the recommendation process.

In the second step, while calculating items for recommendation for a particular user, the most appropriate neighbourhood is selected for searching for the candidates. The level of adequacy is calculated as a value of similarity between the particular user and a cluster centre. As a similarity value (1) it can be used one of several common measures, e.g based on Euclidean distance, cosine value, Pearson correlation, LogLikehood based, Tanimoto, adopted from mathematical applications [19].

$$\mu_{x_i} = \frac{\sum_{q \in V(x_i)} r(x_{iq})}{|V(x_i)|} \tag{1}$$

An example similarity formula based on Pearson correlation is as follows (2):

$$sim_P(x_i, x_j) = \frac{\sum_{k \in V_{ij}} r_{ik} \cdot r_{jk}}{\sqrt{\sum_{k \in V_{ij}} (r_{ik})^2} \cdot \sqrt{\sum_{k \in V_{ij}} (r_{jk})^2}} \tag{2}$$

where $V_{ij} = V(x_i) \cap V(x_j)$ is a set of ratings present in both user's vectors: i and j, $r_{ik} = r(x_{ik}) - \mu_{x_i}$ and $r_{jk} = r(x_{jk}) - \mu_{x_j}$.

In the recommendation list generation process, a similarity measure is estimated in the same way like it was described above. Then, the candidate clusters are searched by the collaborative filtering item-based technique, but only within the cluster.

The recommendation step of the algorithm is described in Algorithm 1. The input set contains data of n users, who rated a subset of items - $A = \{a_1, \ldots, a_k\}$. The set of possible ratings - V - contains values $v_1, \ldots, v_c$. The input data are clustered ncs times into nc clusters every time giving as a result a set of clustering schemes CS. The algorithm generates a list of recommendations R_{x_a} for the active user.

IV. Experiments

This section contains the results of experiments with multi-clustering recommender system with respect to quality of recommendations and time effectiveness. Quality of recommendations was calculated with the Root Mean Square Error (RMSE) measure in the following way. For every user from the input set, their ratings were divided into training (70%) and testing parts. The values from the testing parts were removed and estimated with the selected recommender system. The difference between the original and the estimated number is taken for calculations. The time effectiveness is measured as the average time of generating recommendations list composed of 5 elements for every of 100 users. The tests were performed on a computer with Windows 7 OS, running on Intel Core i7 3.40 GHz with 8 GB of RAM.

 ISBN: 978-1-61208-668-2

Algorithm 1: A general algorithm of a recommender system based on multi-clustering used in the experiments

Data:
- $U = (X, A, V)$ - matrix of clustered data, where $X = \{x_1, \ldots, x_n\}$ is a set of users, $A = \{a_1, \ldots, a_k\}$ is a set of items and $V = \{v_1, \ldots, v_c\}$ is a set of ratings values,
- $\delta : v \in V$ - a similarity function,
- $nc \in [2, n]$ - a number of clusters,
- $ncs \in [2, \infty]$ - a number of clustering schemes,
- $CS = \{CS_1, \ldots, CS_{ncs}\}$ - a set of clustering schemes,
- $CS_i = \{C_1, \ldots, CS_{nc}\}$ - a set of clusters for a particular clustering scheme,
- $CS_r = \{c_{r,1}, \ldots, c_{r,nc \cdot ncs}\}$ - the set of cluster centres,

Result:
- R_{x_a} - a list of recommended items for an active user x_a,

begin

$\delta_1..\delta_{ncs} \longleftarrow$ calculateSimilarity(x_a, CS_r, δ);
$C \longleftarrow$ findTheBestCluster($\delta_1..\delta_{ncs}, CS$);
$R_{x_a} \longleftarrow$ recommend(x_a, C, δ);

The clustering algorithm as well as the recommendation system were created using Apache Mahout library [20]. The methods were tested with various similarity measures implemented in Apache Mahout: Euclidean based ($Eucl$), cosine coefficient (Cos), Pearson correlation measure (Prs), CityBlock (CBl), Tanimoto (Ta) and loglikehood (LL) similarity.

Recommendations were executed on benchmark LastFM music data [21]. The whole set contains over 16 million ratings: 345 652 users who rated 158 697 songs. The data was split into several smaller sets presented in Table I.

TABLE I. DESCRIPTION OF DATA USED IN THE EXPERIMENTS

Name of dataset	Number of users	Number of items	Number of ratings
100k	2032	22 174	99 998
500k	10 236	49 602	499 992
1M	20 464	66 798	999 981
2M	40 914	86 348	1 999 960
3M	61 367	98 924	2 999 945

First, the data was used as input to the traditional collaborative filtering item-based system. Tables II and III contain results of RMSE values and time (in s) of execution while generating 5 recommendation elements.

The next experiment compared the previous results with the results of the recommender system with modelling of neighbourhood by k-means clusters from a single clustering scheme. Tables IV and V contain results of RMSE values and time (in s) of execution while generating 5 recommendation elements. It can be noticed that in every case of the second experiment RMSE is greater than in the first one, regardless of type of similarity measure. However, the time needed to generate 5 recommendation elements is a few hundred times lower than in the first experiment.

TABLE II. RMSE OF ITEM BASED COLLABORATIVE FILTERING RECOMMENDATIONS

Number of cases in data	*Similarity Measure*					
	LL	*Cos*	*Prs*	*Eucl*	*CBl*	*Ta*
100k	0.58	0.58	0.61	0.48	0.58	0.58
500k	0.58	0.58	0.57	0.51	0.58	0.58
1M	0.58	0.58	0.56	0.52	0.58	0.58
2M	0.58	0.58	0.56	0.52	0.58	0.58
3M	0.58	0.58	0.56	0.53	0.58	0.58

TABLE III. TIME [S] OF ITEM BASED COLLABORATIVE FILTERING RECOMMENDATIONS

Number of in data	*Similarity Measure*					
	LL	*Cos*	*Prs*	*Eucl*	*CBl*	*Ta*
100k	0.090	0.125	0.127	0.121	0.071	0.077
500k	0.71	1.02	1.03	1.03	0.663	0.677
1M	1.774	2.718	2.791	2.800	1.761	1.789
2M	4.587	10.186	10.250	7.954	5.782	5.788
3M	6.516	16.021	16.820	8.020	6.210	6.272

TABLE IV. RMSE OF ITEM BASED COLLABORATIVE FILTERING RECOMMENDATIONS WITH NEIGHBOURHOOD DETERMINED BY K-MEANS

Number of clusters	*Similarity Measure*					
	LL	*Cos*	*Prs*	*Eucl*	*CBl*	*Ta*
20	0.65	0.65	0.64	0.64	0.65	0.66
50	0.67	0.67	0.67	0.66	0.67	0.68
100	0.68	0.67	0.67	0.66	0.68	0.67
400	0.65	0.65	0.64	0.63	0.63	0.65
1000	0.66	0.64	0.65	0.62	0.65	0.66

TABLE V. TIME [S] OF ITEM BASED COLLABORATIVE FILTERING RECOMMENDATIONS WITH NEIGHBOURHOOD DETERMINED BY K-MEANS

Number of clusters	*Similarity Measure*					
	LL	*Cos*	*Prs*	*Eucl*	*CBl*	*Ta*
20	0.017	0.019	0.018	0.019	0.015	0.016
50	0.026	0.027	0.027	0.027	0.024	0.024
100	0.011	0.012	0.011	0.011	0.010	0.010
400	0.036	0.041	0.035	0.040	0.035	0.035
1000	0.010	0.02	0.020	0.020	0.010	0.010

The following experiment was based on 3 clustering schemes generated for 20 and 200 clusters. The dataset used for this experiment contained 100 000 ratings ($100k$). Tables VI and VII contain RMSE and time of recommender system executed separately for every scheme. It is visible that the obtained values differ in all cases of schemes generated for the same number of clusters. Different values of RMSE indicate that, by selecting a suitable clustering scheme, particularly for each active user, it is possible to decrease that value.

The last experiment concerns generating recommendation based on a set of 3 clustering schemes (multi-clustering) generated for 20 and 200 clusters. The dataset used for this experiment is the same - $100k$. Tables VIII and IX contain RMSE and time of recommender system executed for multi-clustering. The time is slightly greater than in the experiments where the neighbourhood was modelled by a single clustering scheme, however the value of RMSE is tremendously lower.

TABLE VI. RMSE OF ITEM BASED COLLABORATIVE FILTERING RECOMMENDATIONS WITH NEIGHBOURHOOD DETERMINED BY K-MEANS PERFORMED FOR SELECTED SCHEMES

Number of clusters	*LL*	*Cos*	*Prs*	*Eucl*	*CBl*	*Ta*
20 (1)	0.637	0,640	-	0.480	0.660	0.660
20 (2)	0.644	0.642	3.36	0.486	0.672	0.672
20 (3)	0.638	0.637	-	0.483	0.663	0.663
200 (1)	0.963	0.954	0.500	0.870	1.004	0.995
200 (2)	0.717	0.716	4.876	0.565	0.753	0.754
200 (3)	0.682	0.682	-	0.545	0.724	0.727

TABLE VII. TIME [S] OF ITEM BASED COLLABORATIVE FILTERING RECOMMENDATIONS WITH NEIGHBOURHOOD DETERMINED BY K-MEANS PERFORMED FOR SELECTED SCHEMES

Number of clusters	*LL*	*Cos*	*Prs*	*Eucl*	*CBl*	*Ta*
20 (1)	0.05	0.051	0.053	0.052	0.050	0.049
20 (2)	0.048	0.049	0.052	0.049	0.057	0.047
20 (3)	0.047	0.049	0.052	0.050	0.048	0.046
200 (1)	0.0002	0.0001	0.0002	0.0002	0.0001	0.0002
200 (2)	0.027	0.025	0.026	0.025	0.025	0.024
200 (3)	0.024	0.049	0.052	0.050	0.048	0.046

The experiments proved that the application multi-clustering in recommender systems and dynamic selection the most suitable clusters is very promising and worthy of further research. Figures 3 and 4 depict the summary of values from our experiments. The charts compare all of the examined methods to determine a neighbourhood: k-Nearest Neighbours (IB), k-means single clustering ($IBSC$), k-means multi-clustering ($IBMC$). The multi-clustering approach, even though it takes additional time for dynamic selection of the most suitable clusters, is very valuable due to its extremely low value of RMSE (green and yellow columns in 3 and 4). In case of greater number of clusters (200) the error is bigger, but the processing time is lower.

TABLE VIII. RMSE OF ITEM BASED COLLABORATIVE FILTERING RECOMMENDATIONS WITH NEIGHBOURHOOD DETERMINED BY MULTI-CLUSTERING K-MEANS

Number of clusters	*LL*	*Cos*	*Prs*	*Eucl*	*CBl*	*Ta*
20	0.15	0.15	-	0.11	0.15	0.15
200	0.18	0.18	-	0.16	0.19	0.19

TABLE IX. TIME [S] OF ITEM BASED COLLABORATIVE FILTERING RECOMMENDATIONS WITH NEIGHBOURHOOD DETERMINED BY MULTI-CLUSTERING K-MEANS

Number of clusters	*LL*	*Cos*	*Prs*	*Eucl*	*CBl*	*Ta*
20	0.0633	0.0707	0.0673	0.0668	0.0717	0.0693
200	0.0316	0.0600	0.0411	0.0323	0.0545	0.0404

V. CONCLUSION

Clustering algorithms support recommender systems in increasing time efficiency and scalability. This benefit usually involves decreased accuracy of prediction while generating recommendations. It results from inaccurate modelling of object neighbourhood in case of data located on borders of clusters. Multi-clustering approach eliminates the inconvenience of decreased quality of predictions while maintaining a high time effectiveness.

This article presented an approach based on multi-clustered data, which prevents the negative consequences, keeping high time efficiency. The neighbourhood is modelled by multiple clustering schemes and the most appropriate one to the particular data object is selected for recommendations. The results confirmed a significant reduction of RMSE without an increase in time.

Future work will concern deeper examination of the multi-clustering technique, as well as testing it in various types of recommender systems and on other benchmark datasets.

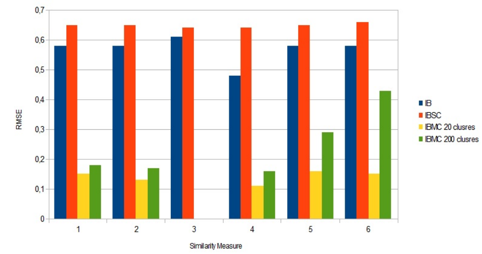

Fig. 3. RMSE of item based collaborative filtering recommendations based on different methods to determine neighbourhood

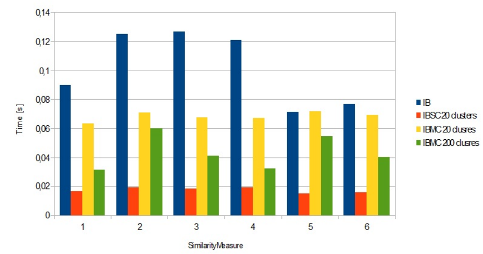

Fig. 4. Time [s] of item based collaborative filtering recommendations based on different methods to determine neighbourhood

ACKNOWLEDGMENT

The present study was supported by a grant S/WI/1/2018 from Bialystok University of Technology and founded from the resources for research by the Ministry of Science and Higher Education of Poland.

REFERENCES

[1] D. Jannach, *Recommender Systems: an Introduction.* Cambridge University Press, 2010.

[2] L. Jie, W. Dianshuang, M. Mingsong, W. Wei, and Z. Guangquan, “Recommender system application developments: A survey,” *Decision Support Systems*, vol. 74, pp. 12–32, 2015.

[3] R. Lawrence, G. Almasi, V. Kotlyar, M. Viveros, and S. Duri, “Personalization of supermarket product recommendations,” *Data Mining and Knowledge Discovery*, vol. 5, pp. 11–32, 2001.

[4] K. McCarthy, J. Reilly, L. McGinty, and B. Smyth, “Thinking positively-explanatory feedback for conversational recommender systems,” in *Proceedings of the European Conference on Case-Based Reasoning (ECCBR-04)*, 2004, pp. 115–124.

[5] U. Kużelewska, *Contextual Modelling Collaborative Recommender System—Real Environment Deployment Results*. Springer, 2016, pp. 119–126.
[6] “Netflix Website,” URL: (https://www.netflix.com/) [accessed: 2018-10-02].
[7] “Spotify Website,” URL: (https://www.spotify.com/) [accessed: 2018-10-02].
[8] A. Nanopoulos, D. Rafailidis, P. Symeonidis, and Y. Manolopoulos, “Musicbox: personalized music recommendation based on cubic analysis of social tags,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, pp. 407–412, 2010.
[9] J. Recio-Garcia, G. Jimenez-Diaz, A. Sanchez-Ruiz, and B. Diaz-Agudo, “Personality aware recommendations to groups,” in *Proceedings of the Third ACM Conference on Recommender Systems*, 2009, pp. 325–328.
[10] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender Systems Survey,” *Knowledge-Based Systems*, vol. 46, pp. 109–132, 2013.
[11] B. Sarwar, “Recommender Systems for Large-Scale E-Commerce: Scalable Neighborhood Formation Using Clustering,” in *Proceedings of the 5th International Conference on Computer and Information Technology*, 2002.
[12] U. Kużelewska, “Clustering algorithms in hybrid recommender system on MovieLens data,” *Studies in logic, grammar and rhetoric*, vol. 37, pp. 125–139, 2014.
[13] J. Rongfei, “A new clustering method for collaborative filtering,” in *Proceedings of the International IEEE Conference on Networking and Information Technology*, 2010, pp. 488–492.
[14] J. Bailey, *Alternative clustering analysis: a review*. Chapman and Hall/CRC, 2014, pp. 533–548.
[15] E. Bae and J. Bailey, “COALA: a novel approach for the extraction of an alternate clustering of high quality and high dissimilarity,” in *Proceedings of the IEEE international conference on data mining*, 2006, pp. 53–62.
[16] J. Hu, Q. Qian, J. Pei, R. Jin, and S. Zhu, “Finding multiple stable clusterings,” *Knowledge and Information Systems*, vol. 51, pp. 991–1021, 2017.
[17] S. Puntheeranurak and H. Tsuji, “A Multi-clustering Hybrid Recommender System,” in *Proceedings of the 7th IEEE International Conference on Computer and Information Technology*, 2007, pp. 223–238.
[18] Y. Wu, X. Liu, M. Xie, M. Ester, and Q. Yang, “Cccf: Improving collaborative filtering via scalable user-item co-clustering,” in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 2016, pp. 73–82.
[19] F. Ricci, L. Rokach, and B. Shapira, “Recommender systems: Introduction and challenges,” in *Recommender systems handbook*. Springer, 2015, pp. 1–34.
[20] “Apache Mahout,” URL: (http://mahout.apache.org/) [accessed: 2017-07-14].
[21] “A Million Song Dataset,” URL: https://labrosa.ee.columbia.edu/millionsong/lastfm/ [accessed: 2017-11-02].