Effect of Dataset Size on Efficiency of Collaborative Filtering Recommender Systems with Multi-Clustering as a Neighbourhood Identification Strategy

Urszula Kużelewska[0000−0003−4612−7640]

Faculty of Computer Science
Bialystok University of Technology,
Wiejska 45a, 15-351 Bialystok, Poland
u.kuzelewska@pb.edu.pl

Abstract. Determination of accurate neighbourhood of an active user (a user to whom recommendations are generated) is one of the essential problems that collaborative filtering based recommender systems encounter. Properly adjusted neighbourhood leads to more accurate recommendation generated by a recommender system. In classical collaborative filtering technique, the neighbourhood is modelled by kNN algorithm, but this approach has poor scalability. Clustering techniques, although improved time efficiency of recommender systems, can negatively affect the quality (precision or accuracy) of recommendations.
This article presents a new approach to collaborative filtering recommender systems that focuses on the problem of an active user's neighbourhood modelling. Instead of one clustering scheme, it works on a set of partitions, therefore it selects the most appropriate one that models the neighbourhood precisely. This article presents the results of the experiments validating the advantage of multi-clustering approach, $M-CCF$, over the traditional methods based on single-scheme clustering. The experiments particularly focus on the effect of great size of datasets concerning overall recommendation performance including accuracy and coverage.

Keywords: Multi-clustering · Collaborative filtering · Recommender systems.

1 Introduction

Recommender Systems (RSs) are solutions to cope with information overload that is observed nowadays on the Internet. Their goal is to provide filtered data to the particular user [12]. As stated in [25], RSs are a special type of information retrieval to estimate the level of relevance of unknown items to a particular user and to order them according to the relevance.

There are non-personalized recommenders based on e.g. average customers' ratings as well as personalized systems predicting preferences based on analysing

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_25

users' behaviour. The most popular RSs are collaborative filtering methods (CF) that build a model on users and the items which the users were interested in [1]. The model's data are preferences e.g. visited or purchased items, ratings [20]. Then CF search for the similarities in the model to generate a list of suggestions that fit users' preferences [20].

They are based on either user-based or item-based similarity to make recommendations. The item-based approach usually generates more relevant recommendations since it uses user's ratings [23] - there are identified similar items to a target item, and the user's ratings on those items are used to extrapolate the ratings of the target. This approach is more resistant to changes in the ratings, as well, because usually the number of users is considerably greater than the number of items and new items are less frequently added to the dataset [2].

During recommendations generation, a huge amount of data is processed. To improve time efficiency and make it possible to generate proposition lists in real time, RSs reduce the search space around an active user to its closest neighbourhood. A traditional method for this purpose is k Nearest Neighbours (kNN) [4]. It calculates all user-user or item-item similarities and identifies the most k similar objects (users or items) to the target object as its neighbourhood. Then, further calculations are performed only on objects from the neighbourhood improving the time of processing. The kNN algorithm is a reference method used in order to determine the neighbourhood of an active user for the collaborative filtering recommendation process [8]. Simplicity and reasonably accurate results are its advantages; its disadvantages are low scalability and vulnerability to sparsity in data [24].

Clustering algorithms can be an efficient solution to the disadvantages of kNN approach due to the neighbourhood is shared by all cluster members. The problems are: the results can be different as the most of clustering methods are non-deterministic and usually significant loss of prediction accuracy. Multi-clustering approach, instead of one clustering scheme, works on a set of partitions, therefore it selects the most appropriate one that models the neighbourhood precisely, thus reducing the negative impact of non-determinism.

The article is organised as follows: the first section presents problems with scalability occurring in collaborative filtering Recommender Systems with a solution based on clustering algorithms, including their advantages and disadvantages. Next section describes the proposed multi-clustering algorithm on the background of alternative clustering techniques, whereas the following section contains results of performed experiments to compare multi-clustering and single-clustering approaches. The last section concludes the paper.

2 Background and Related Work

Clustering is a part of Machine Learning domain. The aim of clustering methods is to organize data into separate groups without any external information about their membership, such as class labels. They analyse only the relationship among the data, therefore clustering belongs to Unsupervised Learning techniques [13].

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_25

Due to independent *á priori* clusters identification, clustering algorithms are an efficient solution to the problem of RSs scalability, providing for recommendation process a pre-defined neighbourhood [21]. Recently, clustering algorithms have drawn much attention of researchers and there were proposed new algorithms, particularly developed for recommender systems application [6], [16], [22]. The efficiency of clustering techniques is related to the fact, that a cluster is a neighbourhood that is shared by all the cluster members, in contrast to kNN approach determining neighbours for every object separately [2]. The disadvantage of this approach is usually loss of prediction accuracy.

The explanation for decreasing recommendations accuracy is in the way how clustering algorithms work. A typical approach is based on a single partitioning scheme, which is generated once and then not updated significantly. There are two major problems related to the quality of clustering. The first is the clustering results depend on the input algorithm parameters, and additionally, there is no reliable technique to evaluate clusters before on-line recommendations process. Moreover, some clustering schemes may better suit to some particular applications, whereas other clustering schemes perform better in other solutions [28]. The other issue addressed to decreasing prediction accuracy is imprecise neighbourhood modelling of the data located on borders of clusters [14], [18].

Popular clustering technique is $k-means$ due to its simplicity and high scalability [13]. It is often used in CF approach [21]. A variant of $k-means$ clustering, bisecting $k-means$, was proposed for privacy-preserving applications [7] and web-based movie RS [21]. Another solution, ClustKNN [19] was used to cope with large-scale RS applications. However, the $k-means$ approach, as well as many other clustering methods, do not always result in clustering convergence. Moreover, they require input parameters e.g. a number of clusters, as well.

The disadvantages described above can be solved by techniques called alternate clustering, multi-view clustering, multi-clustering or co-clustering. They include a wide range of methods which are based on widely understood multiple runs of clustering algorithms or multiple application of clustering process on different input data [5].

Multi-clustering or co-clustering have been applied to improve scalability in the domain of RSs. Co-clustering discovers samples that are similar to one another with respect to a subset of features. As a result, interesting patterns (co-clusters) are identified unable to be found by traditional one-way clusterings [28]. Multiple clustering approaches discover various partitioning schemes, each capturing different aspects of the data [3]. They can apply one clustering algorithm changing values of input parameters or distance metrics, as well as they can use different clustering techniques to generate a complementary result [28].

The role of multi-clustering in the recommendations generation process that is applied in the approach described in this article, is to determine the most appropriate neighbourhood for an active user. It means that the algorithm selects the best cluster from a set of clusters prepared previously (see the following Section).

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_25

A method described in [18] combines both content-based and collaborative filtering approaches. The system uses multi-clustering, however, it is interpreted as clustering of a single scheme on both techniques. It groups the ratings, 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 of an object to the cluster. Then a final prediction rating matrix is calculated to represent the whole dataset. In the last step of pre-recommendation 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 to limit the search space for a collaborative filtering method. It is difficult to compare this approach to the other techniques including single-clustering ones because the article [18] describes the experiments on the unknown dataset containing only 1675 ratings.

The other solution is presented in [26]. The authors observed, that users might have different interests over topics, thus they might share similar preferences with different groups of users over different sets of items. The method $CCCF$ (Co-Clustering For Collaborative Filtering) first clusters users and items into several subgroups, where the each subgroup includes a set of like-minded users and the set of items in which these users share their interests. The groups are analysed by collaborative filtering methods and the result recommendations are aggregated over all the subgroups. This approach has advantages like scalability, flexibility, interpretability and extensibility.

Other applications are: accurate recommendations of tourist attractions based on a co-clustering and bipartite graphs theory [27] and $OCuLaR$ (Overlapping co-CLuster Recommendation) [11] - an algorithm for processing very large databases, detecting co-clusters among users and items as well as providing interpretable recommendations.

There are some other methods, which can be generally called as multi-view clustering, that find partitioning schemes on different data (e.g. ratings and text description) combining results after all ([5], [17]). The main objective of a multi-view partitioning is to provide more information about the data in order to understand them better by generating distinct aspects of the data and searching for the mutual link information among the various views [10]. It is stated, that single-view data may contain incomplete knowledge while multi-view data fill this gap by complementary and redundant information [9]. It is rather useful in interpretability aspect developing in Recommender Systems [11].

3 Description of M-CCF Algorithm

The approach presented in this article has a name Multi-Clustering Collaborative Filtering ($M - CCF$) and defines a multi-clustering process as generation of a set of clustering results obtained from an arbitrary clustering algorithm with the same data on its input. The advantage of this approach is a better quality of the neighbourhood modelling, leading to the high quality of predictions, keeping real-time effectiveness provided by clustering methods. The explanation is in

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_25

imprecise neighbourhood modelling of the data located on borders of the clusters. The border objects have fewer neighbours in their closest area than the objects located in the middle of a cluster. The multi-clustering technique selects the most appropriate cluster to the particular data object. The most appropriate means the one that includes the active object in the closest distance to the cluster's center, thus delivering more neighbours around it. A more detailed description of this phenomenon is in [14], [15].

The general algorithm $M-CCF$ is presented 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. Finally, the algorithm generates a list of recommendations R_{x_a} for the active user.

Algorithm 1: A general algorithm $M-CCF$ 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:

- A_{Rx_a} - a list of recommended items for an active user x_a,

begin

$\delta_1..\delta_{ncs} \longleftarrow$ `calculateSimilarity`(CS_r, CS_i, δ);
$C_{best_{x_a}} \longleftarrow$ `findTheBestCluster`$(x_a, CS_r, \delta_1..\delta_{ncs \cdot ncs}, CS_r, CS_i)$;
$R_{x_a} \longleftarrow$ `recommend`$(x_a, C_{best_{x_a}}, \delta_1..\delta_{nc \cdot ncs})$;

The set of groups is identified by the clustering algorithm which is run several times with the same or different values of its input parameters. In the experiments described in this article, $k-means$ was used as a clustering method. The set of clusters provided for the collaborative filtering process was generated with the same parameter k (a number of clusters). This step, although time-consuming, has a minor impact on overall system scalability, because it is performed rarely and in an off-line mode.

After the neighbourhood identification, the following step, appropriate recommendation generations, is executed. This process requires, despite great pre-

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_25

cision, high time effectiveness. The multi-clustering approach satisfies these two conditions because it can select the most suitable neighbourhood area of an active user for candidates searching and the neighbourhood of all objects is already determined, as well.

One of the most important issues of this approach is to generate a wide set of input clusters that is not very numerous in the size, thus providing a high similarity for every user or item. The other matter concerns matching users with the best clusters as their neighbourhood. It can be obtained in the following ways. The first of them compares the active user's ratings with the cluster centers' ratings and searches for the most similar one using a certain similarity measure. The other way, instead of the cluster centers, compares the active user with all cluster members and selects the one with the highest overall similarity. Both solutions have their advantages and disadvantages, e.g. the first one works well for clusters of spherical shapes, whereas the other one requires higher time consumption. In the experiments presented in this paper, the clusters for active users are selected based on their similarity to the centers of groups (see Algorithm 2).

Algorithm 2: Algorithm of cluster selection of $M-CCF$ recommender system used in the experiments

Data:

- $U = (X, A, V)$ - matrix of clustered data, where x_a is an active user, $A = \{a_1, \ldots, a_k\}$ 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,
- $CS = \{CS_1, \ldots, CS_{ncs}\}$ - 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:

- $C_{best_{x_a}}$ - the best cluster for an active user x_a,
- δ_{best} - a matrix of similarity within the best cluster

begin

$\delta_1..\delta_{ncs \cdot ncs} \longleftarrow$ `calculateSimilarity`(x_a, CS_r, δ);
$\delta_{best} \longleftarrow$ `selectTheHighestSimilarity`$(\delta_1..\delta_{ncs})$;
$C_{best_{x_a}} \longleftarrow$ `findTheBestCluster`$(\delta_{best}, CS, CS_i)$;

Afterwards, a recommendations generation process works as a typical collaborative filtering approach, although the candidates are searched only within the selected cluster of the neighbourhood.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_25

4 Experiments

Evaluation of the performance of the proposed algorithm M-CCF was conducted on the MovieLens dataset [30]. The original set is composed of 25 million ratings; however two subsets were used in the experiments: a small dataset - $100k$ and a big dataset - $10M$. The parameters of the subsets are presented in Table 1.

Table 1. Description of the datasets used in the experiments.

Dataset	Number of ratings	Number of users	Number of items
small dataset - $100k$	100 415	534	11109
big dataset - $10M$	1 000 794	4537	16767

The results obtained with the algorithm $M-CCF$ were compared with the recommender system whose neighbourhood identification is based on a single-clustering. The attention was paid to the precision and completeness of recommendation lists generated by the systems. The evaluation criteria were related to the following baselines: Root Mean Squared Error ($RMSE$) described by (1) and $Coverage$ described by (2) (in %). The symbols in the equations, as well as the method of calculation are characterised in details below.

$$RMSE = \frac{\sum_{i=1}^{N} |r_{real}(x_i) - r_{est}(x_i)|}{N} \tag{1}$$

$$Coverage = \frac{\sum_{i=1}^{N} r_{est}(x_i) \in \mathbb{R}_+}{N} \cdot 100\% \tag{2}$$

where $\mathbb{R}_+$ stands for a set of positive real numbers. The performance of both approaches was evaluated in the following way. Before the clustering step, the whole input dataset was split into two parts: training and testing. In the case of $100k$ subset, the parameters of a testing part were as follows: 393 ratings, 48 users, 354 items, whereas the case of $10M$ subset: 432 ratings, 44 users and 383 items. This step provides the same testing data during experiments and makes the comparison more objective.

In the evaluation process, the values of ratings from the testing part were removed and estimated by the recommender systems. The difference between the original and the calculated value (represented respectively as $r_{real}(x_i)$ and $r_{est}(x_i)$ for user x_i and a particular item i) was taken for $RMSE$ calculation. The number of ratings is denoted as N in the equations. The lower value of $RMSE$ stands for a better prediction ability.

During the evaluation process, there were the cases in which estimation of ratings was not possible. It occurs when the item for which the calculations are performed, is not present in the clusters which the items with existing ratings belong to. It is considered in $Coverage$ index (2). In every experiment, it was

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_25

assumed that the $RMSE$ is significant if the value of $Coverage$ is greater than 80%. It means that if the number of users to whom the recommendations were calculated was 48 and for each of them it was expected to estimate on average 5 ratings, therefore at least 192 values should be present in the recommendation lists.

The clustering method, similarity and distance measures were taken from Apache Mahout environment [29]. To achieve the comparable time evaluation, in implementation of the multi-clustering algorithm, data models (FileDataModel) and structures (FastIDMap, FastIDSet) derived from Apache Mahout were taken, as well. The following data models were implemented: ClusteringDataModel and MultiClusteringDataModel that implement the interface of DataModel. The appropriate recommender and evaluator classes were implemented, as well.

The first experiment was performed on $100k$ dataset, that was clustered independently five times into 10 groups. The clustering algorithm was $k-means$ and a distance measure - *cosine* value between the vectors formed from data points. The number of groups (10) was determined experimentally as an optimal value that led to the highest values of $Coverage$ in the recommendations. In every case, a new recommender system was built and evaluated. Table 2 contains evaluation of the systems' precision that was run with the following similarity indices: $Cosine-based$, $LogLikelihood$, $Pearson\ correlation$, $Euclidean$ distance-based, $CityBlock$ distance-based and $Tanimoto$ coefficient. In the tables below they are represented by the following shortcuts respectively: $Cosine$, $LogLike$, $Pearson$, $Euclidean$, $CityBlock$, $Tanimoto$. The $RMSE$ values are presented with a reference value in brackets that stands for $Coverage$ in this case.

Table 2. RMSE of item based collaborative filtering recommendations with the neighbourhood determined by a single (5 different runs of $k-means$ algorithm) as well as multi-clustering ($k-means$ with $cosine-based$ distance measure) for a small dataset. The best values are in bold.

Similarity Measure	Single Clustering					Multi-Clustering
Cosine	0.88(83%)	0.9(87%)	0.88(81%)	0.89(85%)	**0.87(85%)**	0.87(83%)
LogLike	0.89(87%)	0.88(81%)	0.89(85%)	0.88(81%)	**0.86(85%)**	0.9(83%)
Pearson	-	-	-	-	-	-
Euclidean	0.89(87%)	0.87(81%)	0.88(85%)	0.87(81%)	**0.87(85%)**	**0.85(83%)**
CityBlock	0.87(85%)	0.89(87%)	0.88(81%)	0.89(85%)	**0.86(87%)**	0.88(81%)
Tanimoto	**0.87(87%)**	0.86(81%)	**0.87(85%)**	0.87(81%)	**0.85(85%)**	-

It is visible that the values are different for different input data, although the number of clusters is the same in every result. As an example, the recommender system with $Cosine-based$ similarity has $RMSE$ in the range from 0.87 to 0.9. The difference in values may seem to be small, but the table contains only values

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_25

whose *Coverage* was high enough. Different values of $RMSE$ mean that the precision of a recommender system depends on the quality of a clustering scheme. There is no guarantee that the scheme selected for recommendation process is optimal. Table 2 contains performance results of the recommender system that has the neighbourhood determined by the multi-clustering approach. There is a case where the precision is better (for the *Euclidean* distance based similarity), but in the majority of cases it is slightly worse. Despite this, the multi-clustering approach has eliminated the ambiguity of clustering scheme selection.

The goal of the other experiment was to examine the influence of a distance measure used in the clustering process on a final recommender system performance. The dataset, as well as the similarity measures or a number of clusters, remained the same; however, the distance between the data points was measured by the *Euclidean* distance. The results are presented in Table 3. In this case, one can observe the same values of $RMSE$ regardless of the similarity measure. Note, that the $M - CCF$ algorithm generated results identical to the values from the single-clustering approach.

Table 3. RMSE of item based collaborative filtering recommendations with the neighbourhood determined by a single (5 different runs of $k - means$ algorithm) as well as multi-clustering ($k - means$ with the *Euclidean* distance measure) for a small dataset. The best values are in bold.

Similarity Measure	Single Clustering					Multi-Clustering
Cosine	0.85(83%)	0.85(83%)	0.85(83%)	0.85(83%)	0.85(83%)	0.85(83%)
LogLike	**0.84(83%)**	**0.84(83%)**	**0.84(83%)**	**0.84(83%)**	**0.84(83%)**	**0.84(83%)**
Pearson	-	-	-	-	-	-
Euclidean	**0.84(83%)**	**0.84(83%)**	**0.84(83%)**	**0.84(83%)**	**0.84(83%)**	**0.84(83%)**
CityBlock	0.85(83%)	0.85(83%)	0.85(83%)	0.85(83%)	0.85(83%)	0.85(83%)
Tanimoto	**0.84(83%)**	**0.84(83%)**	**0.84(83%)**	**0.84(83%)**	**0.84(83%)**	**0.84(83%)**

The following experiments were performed on the big dataset ($10M$). By an analogy to the previous ones, the influence of a distance measure was examined, as well. In the first of them, the cosine value between data vectors was taken as a distance measure. The results, for all the similarity indices, are presented in Table 4. The overall performance is worse, although the size of the dataset is considerably greater. There are more cases with insufficient *Coverage* related to the great size of the data, as well. However, the phenomenon of different precision for various clustering schemes in the case of the single-clustering approach remained and the performance of the $M - CCF$ method improved. The table has bold values only for the multi-clustering column.

Finally, the last experiment was performed on the big dataset ($10M$) which was clustered based on the *Euclidean* distance. Table 5 contains the results of $RMSE$ and *Coverage*. *Coverage* values are visibly higher in this case, even for the *Pearson correlation* similarity index. The performance of the multi-

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_25

Table 4. RMSE of item based collaborative filtering recommendations with the neighbourhood determined by a single (5 different runs of $k-means$ algorithm) as well as multi-clustering ($k-means$ with $cosine-based$ distance measure) for a big dataset. The best values are in bold.

Similarity Measure	Single Clustering					Multi-Clustering
Cosine	0.99(98%)	-	0.97(93%)	0.98(91%)	-	-
LogLike	0.99(98%)	-	0.97(93%)	0.98(91%)	-	**0.95(95%)**
Pearson	-	-	-	0.98(91%)	-	-
Euclidean	0.98(98%)	-	0.96(93%)	0.97(91%)	-	**0.93(91%)**
CityBlock	0.98(98%)	0.96(95%)	0.98(91%)	0.96(93%)	-	0.97(91%)
Tanimoto	0.97(98%)	0.95(95%)	-	0.96(93%)	0.96(91%)	0.97(91%)

clustering approach is still better than the method based on single-clustering - the $RMSE$ values are lower in the majority of cases, in the case of the single-clustering, there is only one scheme that slightly outperforms the $M-CCF$ method.

Table 5. RMSE of item based collaborative filtering recommendations with the neighbourhood determined by a single (5 different runs of $k-means$ algorithm) as well as multi-clustering ($k-means$ with the $Euclidean$ distance measure) for a big dataset. The best values are in bold.

Similarity Measure	Single Clustering					Multi-Clustering
Cosine	0.96(93%)	0.97(91%)	0.96(93%)	0.96(93%)	0.95(93%)	**0.94(91%)**
LogLike	0.96(93%)	0.97(91%)	0.96(93%)	0.96(93%)	0.95(93%)	0.96(91%)
Pearson	1.42(93%)	1.09(91%)	0.96(93%)	2.7(91%)	0.99(93%)	**0.93(91%)**
Euclidean	0.95(93%)	0.96(91%)	0.95(93%)	0.95(93%)	**0.94(93%)**	**0.92(93%)**
CityBlock	0.96(95%)	0.96(93%)	0.96(95%)	0.95(95%)	**0.94(95%)**	0.96(93%)
Tanimoto	0.95(93%)	0.95(91%)	0.95(93%)	**0.94(93%)**	**0.93(93%)**	0.95(91%)

Taking into consideration all the experiments presented in this article, it can be observed, that the performance of a recommender system depends on the quality of a clustering scheme provided to the system by a clustering algorithm. In the case of the single-clustering and several schemes generated by this approach, the final precision of recommendations can differ. It means that in order to build a good neighbourhood model for a recommender system, a single run of a clustering algorithm is insufficient. A multi-clustering recommender system and the technique of dynamic selection the most suitable clusters, offers valuable results, particularly in the case of a great size of datasets.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_25

5 Conclusions

In this paper, a developed version of a collaborative filtering recommender system based on multi-clustering neighbourhood modelling is presented. The algorithm $M - CCF$ dynamically selects the most appropriate cluster for every user whom recommendations are generated to. Properly adjusted neighbourhood leads to more accurate recommendations generated by a recommender system. The algorithm eliminates a disadvantage appeared in the case of the neighbourhood determination by a single-clustering method - dependence of the final performance of a recommender system on a clustering scheme selected for the recommendation process.

The experiments described in this paper validated the better performance of the recommender system when the neighbourhood is modelled by the $M - CCF$ algorithm. It was particularly evident in the case of the great dataset containing 10 million ratings. The experiments showed good scalability of the method and increased the competitiveness of the $M - CCF$ algorithm relative to a single-clustering approach in the case of the bigger dataset. Additionally, the technique is free from the negative impact on precision provided by selection of an inappropriate clustering scheme.

The future experiments will be performed to validate the proposed algorithm on different datasets, particularly focused on its great size. It is planned to check the impact of a type of a clustering method on the recommender system's final performance and a mixture of clustering schemes instead of one-algorithm output on an input of the recommender system.

Acknowledgment

The work was supported by the grant from Bialystok University of Technology and funded with resources for research by the Ministry of Science and Higher Education in Poland

References

1. Tuzhilin, A., Adomavicius.G.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge And Data Engineering **17**(6), 734–749 (2005)
2. Aggrawal, C.C.: Recommender Systems. The Textbook. Springer (2016)
3. Dan, A. and Guo, L.: Evolutionary Parameter Setting of Multi-clustering. In: Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, pp. 25–31 (2007)
4. Gorgoglione, M., Pannielloa. U., Tuzhilin. A.: Recommendation strategies in personalization applications. Information&Management **56**(6), 103143 (2019)
5. Bailey, J.: Alternative Clustering Analysis: a Review. Intelligent Decision Technologies: Data Clustering: Algorithms and Applications, pp. 533–548. Chapman and Hall/CRC (2014)

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_25

6. Berbague, C.E., Karabadji, N., Seridi, H.: An Evolutionary Scheme for Improving Recommender System Using Clustering. Computational Intelligence and Its Applications, pp. 290–301. Springer (2018)
7. Bilge A., Polat, H.: A Scalable Privacy-preserving Recommendation Scheme via Bisecting K-means Clustering. Information Process Management **49**(4), 912–927 (2013)
8. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender Systems Survey. Knowledge-Based Systems **46**, 109–132 (2013)
9. Ye, Z., Hui, Ch., Qian, H., Li, R., Chen, Ch., Zheng, Z.: New Approaches in Multi-View Clustering, Recent Applications in Data Clustering. InTechOpen (2018)
10. Guang-Yu, Z., Chang-Dong, W., Dong, H., Wei-Shi, Z.: Multi-View Collaborative Locally Adaptive Clustering with Minkowski Metric. Expert Systems With Applications **86**, 307–320 (2017)
11. Heckel, R., Vlachos, M., Parnell, T., Duenner, C.: Scalable and interpretable product recommendations via overlapping co-clustering. In: IEEE 33rd International Conference on Data Engineering, pp. 1033–1044 (2017)
12. Jannach, D.: Recommender Systems: an Introduction. Cambridge University Press (2010)
13. Kaufman, L.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons (2009)
14. Kużelewska, U.: Collaborative Filtering Recommender Systems Based on $k-means$ Multi-clustering. Advances in Intelligent Systems and Computing, pp. 316–325. Springer (2018)
15. Kużelewska, U.: Multi-clustering Used as Neighbourhood Identification Strategy in Recommender Systems. Engineering in Dependability of Computer Systems and Networks. pp. 293–302. Springer (2019)
16. Pireva, K., Kefalas, P.: A Recommender System Based on Hierarchical Clustering for Cloud e-Learning. Intelligent Distributed Computing XI, pp. 235–245. Springer (2018)
17. Mitra, S., Banka, H., Pedrycz, W.: Rough-fuzzy Collaborative Clustering. IEEE Transactions on Systems, Man, and Cybernetics. Part B (Cybernetics) **36**(4), 795–805 (2006)
18. Puntheeranurak, S., Tsuji, H.: A Multi-clustering Hybrid Recommender System. In: Proceedings of the 7th IEEE International Conference on Computer and Information Technology, pp. 223–238 (2007)
19. Rashid, M., Shyong, K.L., Karypis, G., Riedl, J.: ClustKNN: a Highly Scalable Hybrid Model - &Memory-based CF Algorithm. In: Proceeding of WebKDD (2006)
20. Ricci, F., Rokach, L., Shapira, B.: Recommender Systems: Introduction and Challenges. Recommender systems handbook, pp. 1–34. Springer (2015)
21. Sarwar, B.: 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)
22. Selvi, C., Sivasankar, E.: A novel Adaptive Genetic Neural Network (AGNN) model for recommender systems using modified $k-means$ clustering approach. Multimedia Tools and Applications, pp. 1–28. Springer (2018)
23. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative Filtering Recommender Systems, pp. 291–324. The Adaptive Web (2007)
24. Singh, M.: Scalability and sparsity issues in recommender datasets:a survey. Knowledge and Information Systems, pp. 1–43. Springer (2018)

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_25

25. Vargas, S.: Novelty and diversity enhancement and evaluation in recommender systems and information retrieval. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval - SIGIR '14, pp. 1281–1281 (2014)
26. Wu, Y., Liu, X., Xie, M., Ester, M., Yang, Q.: CCCF: Improving Collaborative Filtering via Scalable User-Item Co-clustering. In: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, pp. 73–82 (2016)
27. Xiong, H., Zhou, Y., Hu, C., Wei, X., Li, L.: A Novel Recommendation Algorithm Frame for Tourist Spots Based on Multi - Clustering Bipartite Graphs. In: Proceedings of the 2nd IEEE International Conference on Cloud Computing and Big Data Analysis, pp. 276–282 (2017)
28. Yaoy, S., Yuy, G., Wangy, X., Wangy, J., Domeniconiz, C., Guox, M.: Discovering Multiple Co-Clusterings in Subspaces. In: Proceedings of the 2019 SIAM International Conference on Data Mining, pp. 423–431 (2019)
29. Apache Mahout, http://mahout.apache.org/. Last accessed 24 Aug 2019
30. MovieLens 25M Dataset, https://grouplens.org/datasets/movielens/25m/. Last accessed 18 Jul 2019

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50420-5_25