Quality of Recommendations and Cold-start Problem in Recommender Systems based on Multi-Clusters

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. This article presents a new approach to collaborative filtering recommender systems that focuses on the problem of an active user's (a user to whom recommendations are generated) neighbourhood modelling. Precise identification of the neighbours has a direct impact on the quality of the generated recommendation lists. Clustering techniques are the solution that is often used for neighbourhood calculation, however, they negatively affect the quality (precision) of recommendations.
In this article, a new version of the algorithm based on multi-clustering, $M-CCF$, is proposed. Instead of one clustering scheme, it works on a set of multi-clusters, therefore it selects the most appropriate one that models the neighbourhood most 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 focus on the overall recommendation performance including accuracy and coverage as well as a cold-start problem.

Keywords: Multi-clustering · Collaborative filtering · Recommender systems · Cold-start problem.

1 Introduction

With the rapid development of the Internet, a large expansion of data is observed. To help users to cope with the information overload, Recommender Systems (RSs) were designed. They are computer applications with the purpose to provide relevant information to a user and as a consequence reduce his/her time spent on searching and increase personal customer's satisfaction. The form of such relevant information is a list (usually ranked) of items that are interesting and useful to the user [8], [16].

Collaborative filtering methods (CF) are the most popular type of RSs [4], [8]. They are based on users' past behaviour data: search history, visited web sites, and rated items, and use them for similarity searching, with an assumption that users with corresponding interests prefer the same items. As a result, they predict the level of interest of those users on new, never seen items [19],

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_6

[4]. Collaborative filtering approach has been very successful due to its precise prediction ability [18].

Although many complex algorithms to generate recommendations were proposed by scientists, it is still an open research challenge to build a universal system which is accurate, scalable, and time efficient [19].

During recommendation generation, a great amount of data is analysed and processed, whereas the generation outcome in real time is an issue. Ideally, an algorithm should produce entirely accurate propositions, that is, suggest items that are highly rated by users. At the same time, the method should be both vertically and horizontally scalable. Vertical scalability is related to the remaining real time of recommendation generation regardless of data size, whereas the horizontal scalability problem occurs when the data is sparse (when few items are connected by users, e.g. rated by them) [16].

The article is organised as follows: the first section presents the background of the neighbourhood identification problem in the field of Recommender Systems. This section discusses common solutions with their advantages and disadvantages as well. Next section describes the proposed multi-clustering algorithm, $M-CCF$ on the background of alternative clustering techniques, whereas the following section contains the results of the performed experiments to compare multi-clustering and single-clustering approaches. The algorithm $M-CCF$ is executed on different types of multi-clusters: when they come from the algorithm with different values of input parameters. The last section concludes the paper.

2 Background and Related Work

Generation of recommendation lists is connected with processing a large amount of input data. The input data is usually the ratings of users on a set of items. If a set of users is denoted as $X = \{x_1, \ldots, x_n\}$ and a set of items as $A = \{a_1, \ldots, a_k\}$, the matrix of the input data can be represented by a matrix $U = (X, A, V)$, where $V = \{v_1, \ldots, v_c\}$ and is a set of ratings values.

The main part of the processing of data is a similarity calculation of every pair of users, and with awareness of the fact that the number of ratings can reach millions values, the real time of recommendation generation appears as a challenge. A common solution to this problem is to reduce the search space around an active user to its closest neighbours [4]. A domain of CF focused on neighbourhood identification is still under intensive research [10], [23].

2.1 Neighbourhood Identification Techniques

The traditional method for neighbourhood calculation is k Nearest Neighbours (kNN) [18]. 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 the objects from the neighbourhood, improving the time of processing. The kNN algorithm is a

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_6

reference method used to determine the neighbourhood of an active user for the collaborative filtering recommendation process [4].

The neighbourhood calculated by kNN technique can be noted as follows:

$$N_{knn}(y_i) = \forall_{y \in Y} \underset{p}{sim}(y_i, y) \tag{1}$$

where p is a number of the neighbours determined by k factor in kNN algorithm. This formula can be related to both users (X) or items (A), therefore the set is denoted generally as Y. Every object in N set is different from y_i.

An example similarity (between items a_i and a_j) formula based on Pearson correlation is as follows:

$$sim_P(a_i, a_j) = \frac{\sum_{k \in V_{ij}} (r(a_{ik}) - \mu_{a_i}) \cdot (r(a_{jk} - \mu_{a_j})}{\sqrt{\sum_{k \in V_{ij}} (r(a_{ik}) - \mu_{a_i})^2} \cdot \sqrt{\sum_{k \in V_{ij}} (r(a_{jk}) - \mu_{a_j})^2}} \tag{2}$$

where $r(a_{ik})$ is a rating of the item a_i given by the user x_k, μ_{a_i} is an average rating of the item a_i given by all users who rated this item, V is a vector of possible ratings $V = \{v_1, \ldots, v_c\}$ and $V_{ij} = V(a_i) \cap V(a_j)$ - a set of ratings present in both item's vectors: i and j.

This equation can be used in item-item CF systems, however, it is possible to build an analogous equation for the calculation of a similarity between users in the case of a user-user recommender. Other similarity measures are: Euclidean-, and CityBlock-based Similarity Measures, Cosine Index, or Tanimoto Similarity [16]. They can be applied in both types of collaborative filtering recommender systems: item-item as well as user-user.

Simplicity and reasonably accurate results are the advantages of kNN approach; its disadvantages are low scalability and vulnerability to sparsity in data [19].

Clustering algorithms can be an efficient solution to the disadvantages of kNN approach due to the neighbourhood being shared by all cluster members. The neighbourhood calculated by clustering techniques can be described by (3).

$$N_{cl}(y_i) = C_j, \Rightarrow C_j = \{y_1, \ldots, y_{cj}\}, C_j \in C \tag{3}$$

where C_j is j-th cluster from one clustering scheme C and c is the number of objects in this cluster. Note that the object y_i is a member of the j-th cluster, as well. In this case, the metric of classification a particular object into a particular cluster is different from the similarity used in recommender systems - usually it is Euclidean distance.

The following problems may arise when one applies clustering algorithms to neighbourhood identification: significant loss of prediction accuracy and different every recommendation outcome. The diversity of results is related to the fact that most of the clustering methods are non-deterministic and therefore several runs of the algorithms can effect obtaining various clustering schemes. The following section, Section 2.2 is devoted to the clustering domain and methods.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_6

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. Section 2.3 presents a background of multi-clustering and describes selected solutions and applications.

2.2 Clustering Methods Used in RS Domain

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 [9].

Due to the independent *á priori* clusters identification, clustering algorithms are an efficient solution to the problem of RSs scalability, providing a predefined neighbourhood for the recommendation process [17]. The efficiency of clustering techniques is related to the fact that a cluster is a neighbourhood that is shared by all cluster members, in contrast to kNN approach determining neighbours for every object separately [17]. The disadvantage of this approach is usually the loss of prediction accuracy.

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 recommendation process. Moreover, some clustering schemes may better suit to some particular applications [22]. The other issue addressed to decreasing prediction accuracy is the imprecise neighbourhood modelling of the data located on the borders of clusters [11], [12].

Popular clustering technique is $k-means$ due to its simplicity and high scalability [9]. It is often used in CF approach [17]. A variant of $k-means$ clustering, bisecting $k-means$, was proposed for privacy-preserving applications [3] and web-based movie RS [17]. Another solution, ClustKNN [15] 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.

2.3 Multi-Clustering Approach to Recommendations

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 applications of a clustering process on different input data [2].

Multi-clustering or co-clustering has 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 clustering [22].

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_6

Multiple clustering approaches discover various partitioning schemes, each capturing different aspects of the data [1]. They can apply one clustering algorithm changing the values of input parameters or distance metrics, as well as they can use different clustering techniques to generate a complementary result [22].

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

The method described in [14] uses a multi-clustering method, however, it is interpreted as clustering of a single scheme for 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. In the last step of the pre-recommendation process, $k - means$ is used again on the new rating matrix to find groups of similar users that represent their neighbourhood with the goal to limit the search space for a collaborative filtering method. It is difficult to compare this approach with other techniques including single-clustering ones, because the article [14] describes the experiments on the unknown dataset containing only 1675 ratings.

The other solution is presented in [20]. The method $CCCF$ (Co-Clustering For Collaborative Filtering) 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 result recommendations are aggregated over all subgroups. This approach has advantages like scalability, flexibility, interpretability, and extensibility.

Other applications are: accurate recommendation of tourist attractions based on a co-clustering and bipartite graph theory [21] and $OCuLaR$ (Overlapping co-CLuster Recommendation) [7] - 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 ([2], [13]). 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 [6]. It is stated that single-view data may contain incomplete knowledge while multi-view data fill this gap by complementary and redundant information [5].

2.4 Contribution of Proposed Work

A novel recommender system with neighbourhood identification based on multi-clustering - $M - CCF$ - is described in this paper. The following are the major contributions of $M - CCF$:

1. Neighbourhood of an active user is modelled more precisely due to the fact that the system's overall neighbourhood is formed by a set of cluster schemes

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_6

and the most similar cluster can be selected in every case, thereby improving the recommendation accuracy.
2. Precise neighbourhood increases the system's horizontal scalabililty, therefore a cold-start problem occurs rarely.
3. Clustering schemes obtained from different runs of a clustering algorithm with different values of input parameters model the neighbourhood better than the schemes obtained from clustering algorithms with the same parameter's values on their input, thereby improving the recommendation accuracy.

The last statement refers to a version of $M-CCF$ described in [12], in which the input data come from multi-clustering approach, however a value of an input parameter in $k-means$ was the same when building one $M-CCF$ RS system.

3 Description of M-CCF Algorithm

The novel solution consists of multiple types of clustering schemes that are provided for the method's input. It is implemented in the following way (for the original version, with one type of a clustering scheme, check in [11], [12]).

Step I. Multiple clustering
The first step of the $M-CCF$ is to perform clustering on the input data. The process is conducted several times and all results are stored in order to deliver them to the algorithm. In the experiments described in this paper, $k-means$ was selected as a clustering method, which was executed for $k = 10, 20, 50$ to generate input schemes (denoted by C set) for one $M-CCF$ RS system. This is illustrated in Figure 1.

Step II. Building M-CCF RS system
It is a vital issue to have precise neighbourhood modelling for all input data. In $M-CCF$ it is performed by iterating every input object and selection of the best cluster from C set for it. The term *best* refers to the cluster which center is the most similar to the particular input object. Then, when all input data have their connected clusters, a traditional CF systems are built on these clusters. As a result, $M-CCF$ algorithm is created - a complex of recommender systems formed on their clusters as recommender data.

A general formula of a neighbourhood calculated by $M-CCF$ method can be described by (4).

$$N_{mcl}(y_i) = C_j(t), \Rightarrow C_j(t) \in C, C = \{C_1(1), \ldots, C_j(1), C_1(2), \ldots, C_g(h)\} \quad (4)$$

where $C_j(t)$ is j-th cluster from t-th clustering scheme, and C are all clustering schemes generated by a clustering algorithm in several runs of different values of its input parameters. In this case, the metric of classification a particular object into a particular cluster is different than the similarity used in recommender systems, as well.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_6

Step III. Recommendation generation
When generating recommendations for an active user, first of all, a relevant RS from $M - CCF$ is selected. It is also based on the similarity between the active user's and cluster centers' ratings. Then, the process of recommendation generation is performed as it is implemented in the traditional collaborative filtering approach, however, searching for similar objects is limited to the cluster connected to the particular recommender in $M - CCF$ algorithm.

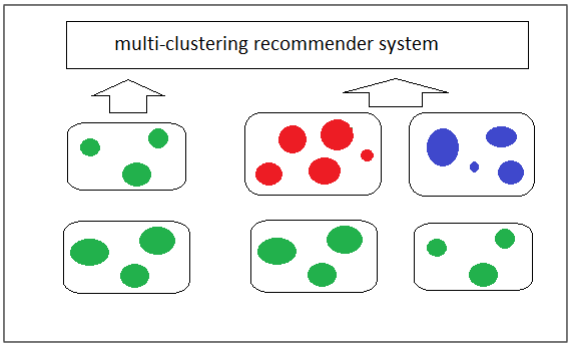

Fig. 1. Comparison of different inputs in $M - CCF$ algorithm

When a neighbourhood is modelled by a single-clustering method, the border objects have fewer neighbours in their closest area than the objects located in the middle of a cluster. The multi-clustering prevents such situations, as it identifies clusters in which particular users are very close to its center. A major advantage of $M - CCF$ algorithm is a better quality of an active user's neighbourhood modelling, therefore resulting in high precision of recommendations, including highly sparse cases.

4 Experiments

Evaluation of the performance of $M - CCF$ algorithm was conducted on two MovieLens datasets: a small one containing 534 users, 11 109 items and 100 415 ratings ($100k$), and a big dataset consisting of 4537 users, 16767 items and 1 000 794 ratings ($10M$) [24]. Note, that the small set is more sparse (contains fewer ratings per user and per item) than the big one.

The results obtained with $M - CCF$ were compared with the recommender system whose neighbourhood modelling is based on a single-clustering ($SCCF$). 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 (5) and $Coverage$ described by (6). The symbols in the equations, as well as the method of calculation are characterised in detail below.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_6

$$RMSE = \sqrt{\frac{1}{n \cdot k} \sum_{i=1}^{nk} \left(r_{real}(v_i) - r_{est}(v_i)\right)^2}, r_{real} \in [2,3,4,5], r_{est} \in \mathbb{R}_+ \quad (5)$$

$$Coverage = \frac{\sum_{i=1}^{N} r_{est}(x_i) \in \mathbb{R}_+}{N} \cdot 100\% \quad (6)$$

where $\mathbb{R}_+$ stands for the 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 $100k$ set, the parameters of the testing part were as follows: 393 ratings, 48 users, 354 items, whereas the case of $10M$: 432 ratings, 44 users and 383 items. This step provides the same testing data during all the experiments presented in this paper, therefore making the comparison more objective.

In the evaluation process, the values of ratings from the testing part were removed and estimated by the recommender system. 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 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 (6). In every experiment, it was assumed that $RMSE$ is significant if the value of $Coverage$ is greater than 90%. It means that if the number of users for whom the recommendations were calculated was 48 and for each of them it was expected to estimate 5 ratings, therefore at least 192 values should be present in the recommendation lists.

The experiments started from the precision evaluation of RS in which neighbourhood was determined by single-scheme $k-means$ algorithm. Table 1 contains evaluation results on $100k$ dataset, whereas Table 2 - on $10M$ data. In both cases, data were clustered independently six times into a particular number of groups to examine the influence of a non-determinism of $k-means$ results.

A clustering measure was one of the following: Euclidean, Cosine-based and Chebyshev. Although Euclidean and Chebyshev distances usually generate accurate partitions, the measure that is selected most often is Cosine-based due to its low complexity, especially for sparse vectors. It needs to be admitted that formally it is not a proper distance metric as it does not have the triangle inequality property. The number of groups was equal 10, 20, or 50. Finally, this experiment was performed 54 times (and repeated 5 times to decrease randomness) per input dataset.

The results concern similarity measures as well. The following indices were taken: $Cosine-based$, $LogLikelihood$, $Pearson\ correlation$, both $Euclidean$ and $CityBlock$ distance-based and $Tanimoto$ coefficient. There are minimal and maximal values of $RMSE$ to present a wide range of precision values which are

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_6

Table 1. RMSE of SCCF evaluated on $100k$ dataset. The best values are in bold.

Similarity Measure	Clustering Distance Measure					
	Euclidean		Cosine-based		Chebyshev	
	min	max	min	max	min	max
Cosine-based	**0.83(99%)**	0.86(99%)	0.85(97%)	0.90(99%)	0.88(85%)	0.91(89%)
LogLikelihood	0.84(99%)	0.86(99%)	0.86(97%)	0.90(98%)	0.88(90%)	0.91(92%)
Pearson corr.	1.18(97%)	5.97(97%)	-	-	-	-
Euclidean	**0.81(99%)**	0.86(99%)	**0.84(94%)**	0.89(95%)	**0.87(86%)**	0.90(91%)
CityBlock	0.84(99%)	0.87(99%)	**0.84(92%)**	0.89(97%)	**0.87(86%)**	0.97(92%)
Tanimoto	**0.83(99%)**	0.85(99%)	**0.84(92%)**	0.89(97%)	0.88(91%)	0.91(95%)

Table 2. RMSE of SCCF evaluated on $10M$ dataset. The best values are in bold.

Similarity Measure	Clustering Distance Measure					
	Euclidean		Cosine-based		Chebyshev	
	min	max	min	max	min	max
Cosine-based	**0.94(99%)**	0.98(84%)	0.95(97%)	0.99(97%)	0.95(95%)	1.00(99%)
LogLikelihood	**0.94(99%)**	1.00(98%)	0.95(97%)	0.99(98%)	0.93(95%)	1.00(98%)
Pearson corr.	1.02(77%)	2.52(99%)	0.96(94%)	2.52(98%)	0.98(94%)	2.52(98%)
Euclidean	0.95(99%)	0.99(99%)	**0.94(99%)**	0.97(99%)	0.92(91%)	0.99(98%)
CityBlock	0.95(99%)	0.99(99%)	**0.94(97%)**	0.99(99%)	**0.92(96%)**	0.99(98%)
Tanimoto	**0.93(99%)**	0.99(92%)	**0.93(97%)**	0.95(99%)	**0.92(98%)**	0.97(99%)

a result of the non-determinism mentioned above. It means that there is no guarantee that the scheme selected for the recommendation process is optimal. The values are presented with a reference value in brackets that stands for $Coverage$.

Recommendation quality is definitely worse for $10M$ dataset. The best $RMSE$ values are 0.92 with $Coverage$=98%, whereas for $100k$ input data - 0.81 with $Coverage$=99%. However, for the big set, the range of values is smaller, regardless of a similarity or distance measure. The $Coverage$ is higher when the number of ratings increases as well.

The following experiments were performed on both input data described above. However, as a recommender, $M-CCF$ method was taken. Table 3 contains the results, containing $RMSE$ and $Coverage$ (they are average values from 5 different runs of $M-CCF$). Similarity and distance measures were the same as in the previous tests. Both input datasets were prepared as follows. All $k-means$ clustering schemes, regardless of a number of clusters, that were obtained in the previous experiments were placed as input data for $M-CCF$ algorithm.

Recommendation precision of $M-CCF$ was also worse in the case of $10M$ dataset. In comparison to the best $RMSE$ values for the single clustering algorithm, they were comparable or slightly worse. However, note that there are not any value ranges, but explicit numbers in every case. It means that the multi-clustering approach has eliminated the ambiguity of clustering scheme selection. Additionally, $Coverage$ was higher for both datasets. It means that $M-CCF$ is able to generate recommendations more often than the recommender system with neighbourhood strategy based on single-scheme clustering.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_6

In the experiments described above, a big impact on $RMSE$ has a similarity measure, however only the best values are discussed there. For similarity as well as distance effect evaluation, see [12].

Table 3. RMSE of $M - CCF$ algorithm evaluated on both datasets: 100k and 10M. The best values are in bold.

Similarity Measure	Euclidean Clustering Distance Measure	
	100K dataset	10M dataset
Cosine-based	0.84(95%)	0.99(98%)
LogLikelihood	0.87(98%)	0.99(99%)
Pearson corr.	-	6.73(98%)
Euclidean	**0.84**(99%)	0.98(99%)
CityBlock	0.89(99%)	**0.97**(99%)
Tanimoto	0.86(99%)	**0.97**(99%)
Similarity Measure	Cosine Clustering Distance Measure	
	100K dataset	10M dataset
Cosine-based	**0.83(95%)**	0.94(97%)
LogLikelihood	0.89(98%)	0.96(99%)
Pearson corr.	4.98(96%)	1.14(99%)
Euclidean	**0.82(98%)**	**0.93**(99%)
CityBlock	**0.84**(98%)	**0.94**(99%)
Tanimoto	**0.82(98%)**	**0.93**(99%)
Similarity Measure	Chebyshev Clustering Distance Measure	
	100K dataset	10M dataset
Cosine-based	0.87(93%)	**0.98**(97%)
LogLikelihood	0.88(98%)	**0.98**(99%)
Pearson corr.	2.41(96%)	1.08(99%)
Euclidean	**0.84**(98%)	**0.97**(99%)
CityBlock	0.87(98%)	0.99(99%)
Tanimoto	0.87(98%)	**0.98**(99%)

Finally, the last experiment concerned a cold-start problem occurrence. The results are in Tables 4, 5, 6 and 7. A precision of recommendations generated for sparse data was tested, that is, for users who rated at most 1, 2, or 3 items. There users were taken from the test set, however, if a particular user had more ratings present, they were removed from the vectors. The ratings for the purge were selected randomly. All previously mentioned aspects (similarity, distance measure, the size of dataset) were taken into consideration.

Analysing the results, it can be noted that it is a common situation, when the recommendation precision increases if users rate more items. However, it is not always a rule. There were a few cases in which the generation of propositions was more difficult. If they appeared in the test set, the final precision was affected. Better quality of recommendations generated for sparse data is visible in the case of big dataset $10M$. This situation is common for both examined recommendation algorithms. However, when the input data were not sparse, the high

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_6

Table 4. Results of cold-start testing (RMSE) - SCCF evaluated on $100k$ dataset. The best values are in bold.

Similarity Measure	Clustering Distance Measure					
	Euclidean			Cosine-based		
	1 rating	2 ratings	3 ratings	1 rating	2 ratings	3 ratings
Cosine	0.97-1.03	0.92-0.97	0.87-0.97	**0.90-0.97**	0.90-0.96	**0.91-0.97**
LogLike	**0.96-1.03**	**0.91-0.97**	0.87-0.97	**0.90-0.98**	0.90-0.96	**0.91-0.96**
Pearson	1.02-2.92	1.04-1.99	1.01-3.07	0.99-2.12	1.01-2.44	0.99-8.83
Euclidean	0.97-1.03	**0.91-0.97**	**0.87-0.96**	**0.90-0.97**	**0.89-0.95**	**0.91-0.96**
CityBlock	1.00-1.03	0.94-0.97	0.90-0.96	0.93-0.97	0.92-0.97	0.93-0.96
Tanimoto	**0.95-1.03**	**0.91-0.97**	**0.86-0.96**	**0.90-0.97**	**0.89-0.95**	**0.91-0.96**

number of ratings affected the precision negatively. The explanation is related to sparsity of data. The $10M$ data, although bigger, were more dense.

Table 5. Results of cold-start testing (RMSE) - SCCF evaluated on $10M$ dataset. The best values are in bold.

Similarity Measure	Clustering Distance Measure					
	Euclidean			Cosine-based		
	1 rating	2 ratings	3 ratings	1 rating	2 ratings	3 ratings
Cosine	0.90-0.93	**0.89-0.93**	0.88-0.94	**0.89-0.94**	0.91-0.93	0.89-0.92
LogLike	0.90-0.93	**0.89-0.93**	0.88-0.95	**0.89-0.94**	**0.90-0.93**	**0.89-0.91**
Pearson	2.17-5.72	1.08-2.98	2.16-5.31	1.07-1.81	2.11-2.86	2.29-2.56
Euclidean	**0.89-0.92**	**0.88-0.93**	0.88-0.95	**0.89-0.93**	0.91-0.93	**0.89-0.91**
CityBlock	0.93-0.97	0.93-0.98	0.92-0.98	0.94-0.98	0.95-0.96	0.93-0.96
Tanimoto	**0.89-0.92**	**0.89-0.93**	**0.86-0.95**	**0.89-0.93**	**0.90-0.93**	**0.89-0.91**

The goal of the last experiment was to evaluate which algorithm better succeeded in managing a cold-start problem. In the case of recommendations with single-clustering based on neighbourhood modelling, the best values obtained for $10M$ dataset equal 0.89-0.92 - in the case of 1 rating, 0.87-0.91 - in the case of 2 ratings, and 0.86-0.95 - in the case of 3 ratings present in users' vectors. For $100k$ dataset, the values were more scattered - 0.90-0.97 - in the case of 1 rating, 0.89-0.95 - in the case of 2 ratings present, and 0.86-0.96 - in the case of 3 ratings present in users' vectors.

In the case of $M-CCF$ recommender which was given on its input the clustering schemes obtained using $k-means$ algorithm for $k = 10, 20, 50$, the best values were obtained for $10M$ dataset equal 0.81 - in the case of 1 rating, 0.84 - in the case of 2 ratings, and 0.91 - in the case of 3 ratings present in users' vectors. For $100k$ dataset, the values were as follows - 0.86 - in the case of 1 rating, 0.85 - in the case of 2 ratings, and 0.89 - in the case of 3 ratings present in users' vectors. The values are more advantageous, that is, $M-CCF$ outperforms the compared method. Moreover, the final value is always unambiguous, the recom-

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_6

Table 6. Results of cold-start testing (RMSE) - SCCF evaluated on both datasets. The best values are in bold.

Similarity Measure	Chebyshev Clustering Distance Measure					
	$100k$ dataset			$10M$ dataset		
	1 rating	2 ratings	3 ratings	1 rating	2 ratings	3 ratings
Cosine	0.90-1.15	0.89-1.03	0.93-1.01	**0.90-0.93**	**0.87-0.91**	0.88-0.93
LogLike	**0.90-1.12**	0.88-1.01	0.92-1.00	**0.90-0.93**	**0.87-0.91**	**0.87-0.93**
Pearson	1.27-5.47	1.18-2.86	2.17-3.82	2.64-5.58	2.34-5.59	2.05-3.04
Euclidean	0.90-1.17	0.88-1.04	0.92-1.02	0.91-0.94	**0.87-0.91**	0.88-0.93
CityBlock	0.95-1.18	0.91-1.05	0.94-1.02	0.95-0.98	0.92-0.95	0.92-0.96
Tanimoto	**0.91-1.07**	**0.88-0.97**	**0.91-0.94**	**0.90-0.93**	**0.88-0.90**	**0.87-0.92**

mender system is able to generate the most optimal propositions. The results for Cosine and Pearson coefficients were not presented due to low *Coverage*.

Taking into consideration all the experiments presented in this article, it can be observed, that the method based on multi-clustering used for neighbourhood modelling in RS is more successful in precise recommendation generation. Moreover, the recommendation lists are more covered, and the results are free from the ambiguousity present in the case of RS in which the neighbourhood is modelled by single-clustering approach.

5 Conclusions

A new developed version of a collaborative filtering recommender system, $M - CCF$, was described in this paper. To improve RS scalability, a search space is limited to variously defined users' neighbourhoods. The presented method models the neighbourhood using a multi-clustering algorithm. It works as follows: $M - CCF$ dynamically selects the most appropriate cluster for every user to whom recommendations are generated. Properly adjusted neighbourhood leads to more accurate recommendations generated by a recommender system. The algorithm eliminates a disadvantage appearing when the neighbourhood is modelled by a single-clustering method - dependence of the final performance of a recommender system on a clustering scheme selected for the recommendation process. Additionally, the preparation of input data was improved. Data come from many clustering schemes obtained from $k - means$ algorithm, however, its input parameters (k) were diversified.

The experiments which are described in this paper confirmed the better performance of $M - CCF$ over the traditional method based on single-clustering. The recommendations have greater precision (lower $RMSE$ values) and are not ambiguous due to working with a mixture of clustering schemes instead of a single one, which may appear not optimal. $M - CCF$ improved better accuracy in the case of a cold-start problem as well.

Next experiments will be performed to prepare a mixture of clustering schemes that is more adjusted to the input data. The clusters will be evaluated and only

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_6

Table 7. Results of data sparsity testing (RMSE) - $M - CCF$ evaluated on both datasets. The best values are in bold.

Similarity Measure	Euclidean Clustering Distance Measure					
	$100k$ dataset			$10M$ dataset		
	1 rating	2 ratings	3 ratings	1 rating	2 ratings	3 ratings
LogLike	**0.91**	0.97	0.98	**0.87**	**0.84**	0.97
Euclidean	1.04	0.94	0.92	1.06	0.96	1.01
CityBlock	1.06	1.04	0.97	0.94	**0.89**	**0.96**
Tanimoto	0.96	0.99	0.93	0.95	0.90	0.97
Similarity Measure	Cosine Clustering Distance Measure					
	$100k$ dataset			$10M$ dataset		
	1 rating	2 ratings	3 ratings	1 rating	2 ratings	3 ratings
LogLike	0.94	0.95	1.02	0.84	0.94	0.97
Euclidean	0.91	**0.92**	1.02	**0.78**	**0.88**	**0.94**
CityBlock	**0.88**	**0.92**	**0.92**	0.83	0.92	**0.94**
Tanimoto	**0.86**	**0.91**	0.94	**0.81**	**0.90**	**0.93**
Similarity Measure	Chebyshev Clustering Distance Measure					
	$100k$ dataset			$10M$ dataset		
	1 rating	2 ratings	3 ratings	1 rating	2 ratings	3 ratings
LogLike	1.01	0.93	1.01	**0.89**	**0.87**	**0.91**
Euclidean	**0.92**	**0.85**	**0.89**	0.92	0.95	1.00
CityBlock	0.95	0.89	0.93	0.91	0.96	1.03
Tanimoto	**0.93**	**0.87**	0.93	0.91	0.97	1.03

the best ones will be given to RS input. Finally, it should improve the overall quality of recommendation: precision as well as scalability.

Acknowledgment

The work was supported by the grant from Bialystok University of Technology WZ/WI-IIT/2/2020 and funded with resources for research by the Ministry of Science and Higher Education in Poland

References

1. Dan, A., 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)
2. Bailey, J.: Alternative Clustering Analysis: a Review. Intelligent Decision Technologies: Data Clustering: Algorithms and Applications, pp. 533–548. Chapman and Hall/CRC (2014)
3. Bilge A., Polat, H.: A Scalable Privacy-preserving Recommendation Scheme via Bisecting K-means Clustering. Information Process Management **49**(4), 912–927 (2013)
4. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender Systems Survey. Knowledge-Based Systems **46**, 109–132 (2013)

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_6

5. 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)
6. 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)
7. 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)
8. Jannach, D.: Recommender Systems: an Introduction. Cambridge University Press (2010)
9. Kaufman, L.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons (2009)
10. Kumar, R., Bala, P.K., Mukherjee, S.: A new neighbourhood formation approach for solving cold-start user problem in collaborative filtering, International Journal of Applied Management Science (IJAMS) **12**(2) (2020)
11. Kużelewska, U.: Dynamic Neighbourhood Identification Based on Multi-clustering in Collaborative Filtering Recommender Systems. International Conference on Dependability and Complex Systems, pp. 410–419. Springer (2020)
12. Kużelewska, U.: Effect of Dataset Size on Efficiency of Collaborative Filtering Recommender Systems with Multi-clustering as a Neighbourhood Identification Strategy. pp. 342–354. Springer (2020)
13. Mitra, S., Banka, H., Pedrycz, W.: Rough-fuzzy Collaborative Clustering. IEEE Trans on Systems, Man, and Cybernetics. Part B (Cybern.) **36**(4), 795–805 (2006)
14. 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)
15. Rashid, M., Shyong, K.L., Karypis, G., Riedl, J.: ClustKNN: a Highly Scalable Hybrid Model - &Memory-based CF Algorithm. In: Proceeding of WebKDD (2006)
16. Ricci, F., Rokach, L., Shapira, B.: Recommender Systems: Introduction and Challenges. Recommender systems handbook, pp. 1–34. Springer (2015)
17. 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)
18. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative Filtering Recommender Systems, pp. 291–324. The Adaptive Web (2007)
19. Singh, M.: Scalability and sparsity issues in recommender datasets:a survey. Knowledge and Information Systems, pp. 1–43. Springer (2018)
20. 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)
21. 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)
22. 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)
23. Zhang, L., Li, Z., Sun, X.: Iterative rating prediction for neighborhood-based collaborative filtering. Appl Intell (2021)
24. MovieLens Datasets, https://grouplens.org/datasets/movielens/25m/. Last accessed 10 Oct 2020

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77964-1_6