Scheme Selection based on Clusters' Quality in Multi-Clustering $M - CCF$ Recommender System

Urszula Kużelewska
Faculty of Computer Science/ Bialystok University of Technology
Bialystok, Poland *u.kuzelewska@pb.edu.pl*

Abstract

Identifying a neighbourhood based on multi-clusters was successfully applied to recommender systems, increasing recommendation accuracy and eliminating divergence related to differences in clustering schemes generated by traditional methods. Multi-Clustering Collaborative Filtering algorithm was developed for this purpose, which was described in the author's previous papers. However, the solutions involving many clusters face substantial challenges around memory consumption and scalability. Differently, some groups are not useful due to their high similarity to other ones. Selection of the clusters to provide to the recommender system's input, without deterioration in recommendation accuracy, can be used as a precaution to address these problems. The article describes a solution of a clustering schemes' selection based on internal indices evaluation. The results confirmed its positive impact on the system's overall recommendation performance. They were compared with baseline recommenders' outcomes.

Keywords: Multi-clustering, Collaborative filtering, Evaluation of clustering schemes.

1. Introduction

Recommender systems (RSs) are electronic applications to help users to reach the information or resource they are interested in. Usually, their outcome is collected in the form of a list of recommender items, typically ranked, which is [9]. Collaborative filtering techniques (CF) are a very attractive solution in the domain of RSs [3]. They generally search for similarities among users or items with an underlining supposition that users with comparable activity select the same items. They consider users' data, e.g. visited websites, rated items. As a consequence, recommenders are able to evaluate the level of interest of those users on different items which are new for them. Collaborative filtering methods particularly generate accurate recommendations [9].

Vertical scalability corresponds to real-time delivery of recommendation lists despite data size [19]. Clustering algorithms are attractive tools for this problem [21]. They identify groups of similar items (or users) that can contribute to recommender systems for $a\ priori$ identification of neighbourhood objects. On the other hand, they have their weak points as well. Most of them have input parameters, which different values highly influence the final results. Moreover, even though the values remain the same, the outcomes can differ. It is related to how they work - their purpose is not to find a globally optimal partition, but a local one, starting with different initial points [10].

The challenges identified above can be accompanied by new methods described as: multiple clusterings, multi-clustering, or ensemble clustering [2], [15]. Their common idea is to implement multiple runs of clustering algorithms or to apply multiple applications of a partitioning process on different input data. Algorithm Multi-Clustering Collaborative Filtering (described in [12]) works on a set of several clustering schemes, which come from several runs of a clustering method with different values of an input parameter. Such operation fulfills ensemble clustering purpose, which is "to find a combined clustering result based on multiple clusterings of the dataset" [15]. The first experiments, which were described in [13], validated

$M-CCF$ against baseline predictors: an item-based and single-clustering recommender systems. The item-based approach ($IBCF$) identifies neighbourhoods using k Nearest Neighbours algorithm, and the single-clustering techniques ($SCCF$) utilize only one partitioning scheme for this purpose. The advantage of $M-CCF$ was gained in terms of recommendation quality. Unfortunately, other elements of performance have deteriorated: the time taken to generate recommendations as additional time required to select one of the clustering schemes.

The selection of clustering schemes handles this problem, reducing the number of clusters to forward on $M-CCF's$ input. As selection tools, internal indices are proposed, to measure the compactness and separability of the clustering schemes. Our main contributions are as follows:

- Selection of clusters to forward to $M-CCF$ input is beneficial for its performance in terms of recommendation accuracy and coverage,
- Criteria based on internal indices for evaluation of quality of clustering schemes is a suitable approach to identify valuable clusters for $M-CCF$ input.

The article is organised as follows: the following section presents the background of the clustering algorithms in the field of RSs. The next section, Section 3, describes the proposed algorithm, $M-CCF$. Section 4 is devoted to a cluster selection procedure. The following section contains the results of the performed experiments. The last section concludes the paper.

2. Background and Related Work

To achieve high-quality clustering, several problems need to be considered. First of all, a final partitioning is determined by the values of parameters given to the algorithms' input. Moreover, the evaluation of clusters to select for the recommendation process is also challenging. The additional problem connected with using clusters in recommender systems is decreasing prediction accuracy. It results from incorrect neighbourhood identification of the data located on the borders of groups. In these circumstances, the objects from other groups located close to the border data, may appear to be more similar to the active user. It is discussed in detail in [13].

A selection procedure of a clustering algorithm is also essential. Simplicity and high scalability make $k-means$ one of the most popular clustering techniques [10], particularly useful in collaborative filtering recommender systems. The authors in [8] combined $k-means$ to cluster items in a movie recommender with online learning automata-based user profiling. Another method, ClustKNN [17] was used to handle large-scale RS applications. Two-stage clustering was applied in [6] to implement a concept of so called $RatingBubbles$. They appear when users and items are grouped into homogeneous clusters. In [11], a biclustering approach, with clusters' overlap, is used for neighbourhood formation. The authors obtained a strong partial similarity with active user's preferences. One of the recent solutions [7] applies hierarchical clustering to extract clusters from a hierarchy of candidates automatically. It can be applied as a preprocessing step in an arbitrary recommender system.

3. Presentation of M-CCF Algorithm

The proposed method $M-CCF$ is implemented in the following way (for the original version, with one type of clustering scheme, check in [13], [12]).

Step I. Multiple clustering

The first stage of the $M-CCF$ algorithm is identifying clusters in the input data. The process is repeated several times, and all outcomes are saved in order to transfer them to $M-CCF$. In the experiments described in this paper, $k-means$ was selected as a clustering algorithm.

Step II. Building M-CCF RS system

In the case of items clustering, every item needs to have the most appropriate cluster identified. The term $the\ most\ appropriate$ is related to the cluster, in which center object is the most similar to the particular input data. Then, when all input objects have their associated clusters, traditional separate CF systems are built on these clusters, which provide input data sets for them. As a consequence, $M-CCF$ system is built - an aggregate of recommender algorithms created on particular clusters.

Step III. Recommendation generation

First, a relevant RS from $M-CCF$ is selected when recommendations for a target user are generated. It also utilizes the similarity between the target user's and cluster centers' ratings. Then, the recommendation generation process is executed as it is implemented in the traditional CF approach. However, searching for similar items is restricted to the group connected to the particular recommender.

When a single-clustering scheme represents a neighbourhood, the items located on the border of clusters have fewer neighbours in their nearby area than the ones located in the middle of a group. Moreover, if the clusters are located close to one another, more similar neighbours can be the ones belonging to the other clusters. The multi-clustering avoids such situations, as it recognizes clusters where particular users or items are very close to its center. However, some schemes are not used in the recommendation process as long as they do not contain the optimal location of objects.

4. Cluster Selection Techniques

A concept of $cluster\ ensemble$ or $clustering\ aggregation$ emerged to integrate several partitionings into a final outcome [20]. One of the approaches to this concept that generates a set of base clustering schemes is to run a single clustering algorithm with different initial sets of parameters several times [1]. Then, a cluster selection procedure can be applied to determine the relevant ones to a particular problem.

Cluster ensembles are widely used in data mining tasks, including recommendation generation. In [1], a recommender system is proposed, which uses $k-means$-based method, called $KMCE$, to select a final result from many base clustering schemes. The authors used the Rand index as one of the evaluation criteria. Recommendation accuracy was raised in [21] by applying a combination of PCA and $k-means$ methods. $Dunn$ index was used to evaluate clusterings.

Evaluation of clustering results is challenging due to the lack of group labels. In this case the only option is to use internal measures [10]. In the experiments described below, the following indices were applied: Silhouette (SH) (1) [18], Davies-Bouldin (DB) (2) [5] and Calinski-Harabasz (CH) (3) [4].

$$\begin{aligned}
&SH = \forall_{y_i \in Y}\overline{SH(y_i)}\\
&SH(y_i) = \frac{a(y_i)-b(y_i)}{\max(a(y_i),b(y_i))}\\
&a(y_i) = \frac{1}{\|C_i\|-1}\sum_{j\in C_i, i\neq j} d(i,j)\ b(y_i) = \min_{k\neq i}\frac{1}{\|C_k\|}\sum_{j\in C_i} d(i,j)
\end{aligned} \tag{1}$$

The component $a(y_i)$ calculates an average distance between y_i object and all other objects in the same cluster, whereas $b(y_i)$ is an average distance between y_i object and all other objects in the nearest cluster. The range of SH is [-1,1], with naturally correct clusters was identified by higher values.

$$\begin{aligned}
&DB = \frac{1}{|C|}\sum_{C_i,C_j\in C}\max_{i\neq j}\frac{diam_{C_i}+diam_{C_j}}{D(C_i,C_j)}\\
&diam_{C_i} = \forall_{k\in C_i}\overline{d(c_i,k)},\ diam_{C_j} = \forall_{m\in C_j}\overline{d(c_j,m)},\ D(C_i,C_j) = d(c_i,c_j)
\end{aligned} \tag{2}$$

The components $diam_C$ stand for cluster diameters, which are average distances between the cluster's center and all other cluster's points, whereas D is the distance between clusters, calculated by the distance between their centers. The desired value is around 0, which is related to well-separable clusters.

$$CH = \frac{Sep(C)}{Coh(C)}$$
$$Sep = \frac{\sum_{C_i \in C} \|C_i\| \cdot d(c_i - c)}{\|C\| - 1}, \; Coh = \frac{\sum_{C_i \in C} \sum_{j \in C_i} d(i, c_i)}{\sum_{C_i \in C} \|C_i\| \|C\|} \tag{3}$$

CH value is related to the similarity of objects to their own clusters (cohesion - Coh) compared to other clusters (separation - Sep). The highest values mean better result partitions.

The indices described above were applied to evaluate clustering schemes in order to eliminate the useless ones. The detailed idea is described in Section 5.1.

5. Experiments

The experiments were divided into 2 phases: clustering with clusters' evaluation and generating recommendations with a measurement of their accuracy. In the first phase, $k - means$ was taken as the most common clustering algorithm and was successfully deployed in the previous version of $M - CCF$ approach [13].

A subset of MovieLens dataset [23] was taken for this purpose. Originally, the data contained 25 million ratings; however, randomly selected samples were taken in the experiments. The set consisted of 100 000 ratings (549 users and 11 024 items) and was split into training and testing parts in the proportion of about 100 to 1.

5.1. Clustering and Evaluation of Clustering Schemes

The clustering process was executed several times with the following values of k: 5, 20, 50 and 100. The range and the particular numbers of parameters were selected after the execution of many experiments as the values which highly influence the recommendation accuracy. Furthermore, it used various distance measures: cosine-based, Euclidean, CityBlock and Tanimoto-based. It was decided to cluster the items (movies).

Every run of $k - means$ was repeated 6 times, and each result (a clustering scheme) was evaluated in terms of compactness and separability. The implementation of indices in Python's Scikit Learn library was applied [16]. Figure 1 presents evaluation results for each scheme.

The values of internal indices were analysed during the evaluation of clustering schemes. Although all of them can detect well separable and compact clusters, the evaluation process was not a straightforward task. In numerous cases, the evaluation values of particular indices were not considerably diversified to imply an appropriate result, and additionally, the indices' optimum were not coherent. In the definitive selection, the following rules were applied: the importance of a level of difference in every particular index's value and voting of the indices in the case of inconsistency.

The left top figure's data was clustered with cosine-based distance. The solid line (identically on all graphs) denotes an assessment of 5 groups and clearly identifies all indices with the 2nd and 5th scheme as the best results. The dashed line denotes the evaluation of 20 groups, and both CH and SH indices indicate their 5th scheme as the best result. In contrast, DB index has the lowest value for the 6th clustering, but the difference between the 5th and 6th results is very slight. Both evaluation lines - CH and SH - for 50 group schemes (dotted lines) are relatively flat, which means they were not able to distinguish any result, as opposed to DB, which indicates its 5th clustering as the best one. A similar situation is observed in the case of 100

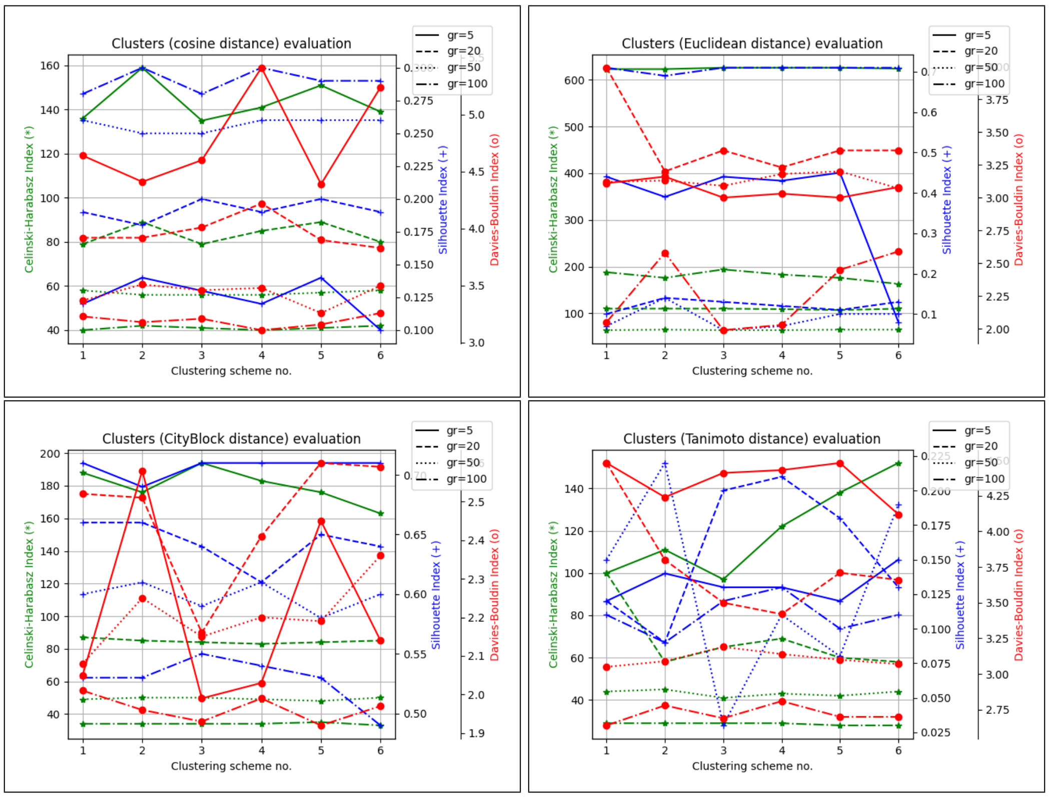

Fig. 1. Evaluation of clustering schemes

group schemes (dash-dotted lines) in which all indices' graphs are equal with extremely slight fluctuations on both schemes: the 2nd and 4th. Finally, the following schemes were selected for a recommendation phase: the 2nd, the 5th (5 groups), the 5th, the 6th (20 groups), and the 5th (50 groups).

The top right graph presents analogous results; however, a distance measure during the clustering process was Euclidean. These schemes are more difficult to select due to the lack of distinct points on the evaluation graphs. However, in order to make further research in this space complete, some schemes were selected in this case, as well: the 3rd, the 5th (5 groups), the 2nd, the 4th (20 groups), the 6th (50 groups), the 1st, the 2nd, the 3rd (100 groups) due to slight better indices' performance.

Corresponding results, but for a CityBlock distance as a clustering metric, are shown on the left bottom graph in the exact figure. The results were evaluated unambiguously for the 5 group clusterings, with the 3rd and 4th schemes specified as the best. In the following case, the results of the 20 group clusterings were relatively precise with the 1st and 3rd schemes selected. For the 50 group clusterings, only the 1st scheme was taken; however, the values of indices were somewhat unambiguous. The last case, the set of 100 group clusterings, was evaluated unanimously, and the following schemes were chosen: the 3rd, and the 5th.

5.2. Evaluation of Recommendations

The best clustering schemes were forwarded to $M-CCF$ recommender system. Then the calculated values were compared to the original ones in order to determine a difference in precision and completeness of recommendation lists. Evaluation criteria were related to the following standard main metrics:

- Root Mean Squared Error ($RMSE$) a baseline way to measure the error in model evaluation studies. It is a square root of an arithmetic mean of the squares of the predictions between the model and the observations. The lower value of $RMSE$ refers to a better prediction ability.

- $Coverage$ measures the system's responsiveness to a required length of a recommendation list. It is a portion of generated predictions to the needed size. If a system covers all positions in the recommendation list, its $Coverage$ is 100%. During the evaluation process, there were cases in which rating estimation was impossible. It often occurs in both $SCCF$ and $M-CCF$ when the item for which the calculations are performed, is not present in the same cluster to which the already rated items belong. In every experiment, it was assumed that $RMSE$ is significant if the value of $Coverage$ is greater than 90%.

Table 1 reports results of the system's evaluation. The following measures were used to calculate similarities between items: $Cosine-based$, $LogLikelihood$, $Euclidean$ distance-based. To have a compact view of the obtained results, without reducing the general concept to confirm, only selected results are reported in the experiments, which were generated by both $SCCF$ and $M-CCF$ recommender systems with remaining all $IBCF$ outcomes presented. If the $Coverage$ was below 90% the result was not displayed in the tables - instead, there is a mark '-'.

The tables contain an evaluation of $IBCF$ in the first row and selected configurations of $SCCF$ and $M-CCF$ systems that are formatted in the following way:

- SCCF-$distance$-x - x clusters generated by $k-means$ using cosine-based (cos) or Euclidean (eu) distance,

- M-CCF-$distance$-x-[y]-[z] - clusters generated by $k-means$ with $k=x,y,z$ (y and z are optional) in 6 runs, using one of the following distance measures: cosine-based (cos), Euclidean (eu), CityBlock (cb) or Tanimoto (tan),

- M-CCF-$distance$-x-[y]-[z]-s - clusters generated by $k-means$ on the conditions described above, however, the procedure of clustering schemes was applied.

In the case of $IBCF$ algorithm, the results are not very good: $RMSE$ ranges from 0.91 to 0.94 with relatively high $Coverage$ from 95% to 99%. $SCCF$ method obtained better values in some configurations: for data split into 5 groups for both clustering metrics: Euclidean and cosine-based: $RMSE$ ranges from 0.88 to 0.93 with $Coverage$ comparable to the previous results. Note that in this case a range of values is presented. It refers to the characteristic of $k-means$ clustering algorithm, which often generates different results even if the values of its input parameters remain the same. Hence, it was launched 6 times and every particular clustering scheme was evaluated individually.

The outcomes are comparable or frequently better in the case of $M-CCF$ algorithm. For instance, the configuration $M-CCF-eu-5$ and Euclidean-based similarity obtained $RMSE$=0.88, and the configuration $M-CCF-cos-5-20-50-s$ and the same similarity obtained $RMSE$=0.91. It must be admitted that the values of $Coverage$, although over 90%, are slightly lower than in the previous cases. The most crucial issue is that this experiment shows that the cluster schemes selection in terms of compactness and separability for $M-CCF$ algorithm mainly contributed towards the model's performance - usually, the values $RMSE$ as well as $Coverage$ were improved. As an example, the configuration $M-CCF-cos-5-20-50-s$ can be presented, in which $RMSE$ value decline was 0.03 with simultaneous progress in $Coverage$.

Table 1. RMSE of the algorithms on 100k dataset. The best values are in bold.

Algorithm	Similarity Measure		
	Cosine-based	LogLikelihood	Euclidean
IBCF	0.94 (95%)	0.94(95%)	**0.92(95%)**
SCCF-cos-5	0.93-0.95 (95%)	0.93-0.94(95%)	**0.91-0.93(95%)**
SCCF-cos-20	0.94-1.00 (93%)	0.93-0.99(93%)	0.93-0.98(93%)
SCCF-cos-50	0.98-1.00 (91%)	0.98-1.00(95%)	0.97-0.99(97%)
SCCF-cos-100	-	0.96-1.03(94%)	0.95-1.02(97%)
SCCF-eu-5	**0.90-0.91 (95%)**	**0.90-0.91(97%)**	**0.89-0.90(98%)**
SCCF-eu-20	1.00-1.08 (93%)	0.99-1.08(96%)	0.99-1.08(98%)
SCCF-eu-50	-	1.00-1.03(95%)	1.02-1.03(97%)
SCCF-eu-100	-	0.99-1.00(94%)	0.99-1.02(96%)
M-CCF-eu-5	**0.91 (95%)**	**0.91 (95%)**	**0.88 (95%)**
M-CCF-eu-5-20	0.94 (93%)	0.98 (92%)	**0.92 (93%)**
M-CCF-eu-5-20-s	0.94 (93%)	0.98 (92%)	0.93 (93%)
M-CCF-cos-5-20-50	0.96 (91%)	-	0.94 (90%)
M-CCF-cos-5-20-50-s	0.93 (92%)	0.99 (90%)	**0.91 (91%)**
M-CCF-cb-5-20-50	0.95 (90%)	0.95 (90%)	0.93 (90%)
M-CCF-cb-5-20-50-s	0.94 (93%)	0.94 (93%)	**0.92 (93%)**
M-CCF-tan-5-20-50	0.97 (90%)	-	-
M-CCF-tan-5-20-50-s	0.96 (91%)	0.96 (90%)	0.93 (91%)

6. Conclusions

This paper presents a recommender system based on multi-clustering to model the neighbourhood of a target user with internal indices-based clustering scheme selection. The concept of $M - CCF$ algorithm is to store multiple clustering schemes on its input and dynamically match every item that takes part in the recommendation generation process with the most appropriate cluster. As accuracy advances, it faces substantial challenges around time efficiency.

An exclusive set of partitions benefits $M - CCF$ algorithm's performance - $RMSE$ and $Coverage$. The results of the executed experiments confirmed that the performance of $M - CCF$ algorithm is usually better when it works on a reduced set of input clusters. Additionally, the technique still becomes free from the negative impact on the precision provided by the selection of an inappropriate clustering scheme as it occurs in the case of recommender systems in which the neighbourhood of objects is identified by single-clustering schemes.

Additional experiments will be executed to verify the proposed approach on datasets of greater size, e.g. 10 million ratings. Moreover, it is planned to check the impact of different types of clustering algorithms on the overall performance of the recommender system. The other characteristics of $M - CCF$, such as diversity, serendipity, and novelty, will be also evaluated.

Acknowledgment

Work supported by the grant from Bialystok University of Technology WZ/WI-IIT/3/2023 and funded with resources for research by the Ministry of Education and Science in Poland.

References

1. Bai, L., Liang, Y., Cao, F.: A Multiple K-means Clustering Ensemble Algorithm to Find Nonlinearly Separable Clusters, Information Fusion 61, 36-47 (2020)
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. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender Systems Survey. Knowledge-Based Systems **46**, 109–132 (2013)
4. Caliński, T., Harabasz, J.: A Dendrite Method for Cluster Analysis, Communications in Statistics - Theory and Methods 3, 1-27 (1074)
5. Davies, D. L., Bouldin, D. W.: A Cluster Separation Measure, PAMI-IEEE Transactions on Pattern Analysis and Machine Intelligence 2 (1), 224-227 (1979)
6. Divyaa, L. R., Pervin, N.: Towards Generating Scalable Personalized Recommendations: Integrating Social Trust, Social Bias, and Geo-spatial Clustering, Decision Support Systems 122, 1-17 (2019)
7. de Aguiar Neto, F.S., da Costa, A.F., Manzato, M.G., Campello, R.J.G.B: Preprocessing approaches for collaborative filtering based on hierarchical clustering, Information Sciences 534, 172-191 (2020)
8. Farahani, M.G., Torkestani, J.A., Rahmani, M: Adaptive personalized recommender system using learning automata and items clustering, Information Systems 106, 101978 (2022)
9. Jannach, D.: Recommender Systems: an Introduction. Cambridge University Press (2010)
10. Kaufman, L.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons (2009)
11. Kant, S., Mahara, T.: Nearest biclusters collaborative filtering framework with fusion, Journal of Computational Science 25, 204-212 (2018)
12. Kużelewska, U.: Performance of Multi-Clustering Recommender System after Selection of Clusters based on V-Measures, In: Information Systems Development: Artificial Intelligence for Information Systems Development and Operations (ISD2022 Proceedings), pp. 1-8 (2022)
13. Kużelewska, U.: Effect of Dataset Size on Efficiency of Collaborative Filtering Recommender Systems with Multi-clustering as a Neighbourhood Identification Strategy", In: International Conference on Computational Science' 2020, pp. 342-354 (2020)
14. Latifi, S., Mauro, N., Jannach, D.: Session-aware recommendation: A surprising quest for the state-of-the-art, Information Sciences 573, 291-315 (2021)
15. Li, T., Ogihara, M., Ma, S.: On Combining Multiple Clusterings: An Overview and a New Perspective, Applied Intelligence 2 (33), 207-219 (2010)
16. Pedregosa, F.: Scikit-learn: Machine Learning in Python, JMLR 12, 2825-2830 (2011)
17. Rashid, M., Shyong, K. L., Karypis, G., Riedl, J.: ClustKNN a Highly Scalable Hybrid Model - Memory-based CF Algorithm, In: Proceeding of WebKDD (2006)
18. Rousseeuw, P. J.: Silhouettes a Graphical Aid to the Interpretation and Validation of Cluster Analysis, Computational and Applied Mathematics 20, 53–65 (1987)
19. Ricci, F., Rokach, L., Shapira, B.: Recommender Systems: Introduction and Challenges. Recommender systems handbook, pp. 1–34. Springer (2015)
20. Strehl, A., Ghosh, J.: Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions, Journal of Machine Learning Research 3, 583-617 (2002)
21. Yadav1, V., Shukla, R., Tripathi, A., Maurya, A.: A New Approach for Movie Recommender System using K-means Clustering and PCA, Journal of Scientific & Industrial Research 80, 159-165 (2021)
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. MovieLens Dataset, https://grouplens.org/datasets/movielens/25m/.