

Context Clustering-based Recommender Systems

Eyad Kannout

Institute of Informatics, University of Warsaw
Banacha 2, Warsaw, Poland
eyad.kannout@mimuw.edu.pl

Abstract—Recommender systems have gained lots of attention due to the rapid increase in the amount of data on the internet. Therefore, the demand for finding more advanced techniques to generate more useful recommendations becomes an urgent. The increasing need for generating more relevant recommendations led to the emergence of many novel recommendation systems, such as Context-aware Recommender System (CARS), which is based on incorporating the contextual information in recommendation systems. The goal of this paper is to propose new recommender systems that utilize the contextual information to find more relevant recommendations.

In this paper, we propose CoCl, a novel Context Clustering-based recommender system. We introduce two approaches which utilize the contextual information and KMeans clustering algorithm to generate new forms of user-item matrices. We show that the accuracy of CoCl which uses the new user-item matrices has been improved comparing with the accuracy of classical recommender system which uses the original user-item matrix.

Keywords: collaborative filtering, context-aware recommender system, contextual information, clustering, accuracy of predictions, quality of recommendations.

I. INTRODUCTION

THROUGHOUT the past decade, along with the rapid expansion of the online services, many e-commerce, e-tourism, e-resource services, social media, and retail companies started leveraging the power of data in order to boost their profits by assisting the customers in discovering interesting items/products in a huge amount of online data. For the sake of achieving this goal, a recommender system (RS) must be implemented, and therefore, the demand for recommender systems have been increased more than ever before.

In theory, traditional, or two-dimensional (2D), recommender systems tend to estimate user preferences or user's ratings based on the ratings given by the users to other items, and possibly on some other information, such as user demographics and item characteristics. However, early recommender systems emerged without taking into consideration any contextual information, such as time, location, and the company of other people when providing recommendations. During the past decade, the increasing need for generating more relevant recommendations led to the emergence of many novel recommendation systems, such as context-aware recommender systems (CARS), social recommender systems, and group recommender systems.

Recently, the field of context-aware recommender systems (CARS) has attracted a lot of attention due to its importance in many recommendation applications. Although an increasing

number of papers on context-aware recommender systems have been appeared recently, this field is still considered as relatively new and several challenges that need more attention by the current researchers still exist. Therefore, in this paper, we work on finding new methods that incorporate the contextual information in recommendation systems to generate more useful and user-related recommendations.

The main contributions of this paper are as follows: 1) proposing two methods which cluster the ratings and the users in user-item matrix using the contextual information; 2) producing new aggregated forms of user-item matrix based on previous grouping of ratings and users; 3) employing collaborative filtering model to predict missing preference of a user for an item using new aggregated user-item matrices.

The remainder of this paper is organized as follows. In Section 2, we provide background information for collaborative filtering and context-aware recommender systems. Section 3 describes the problem we study in this paper and reviews its related works. In Section 4, we present CoCl, a novel context clustering based recommender system. Section 5 evaluates and compares CoCl against traditional recommender system. Finally, in Section 6, we draw conclusions and make suggestions of possible future work.

II. LITERATURE REVIEW FOR RECOMMENDATION SYSTEMS

In this section, we briefly summarize the academic knowledge on collaborative filtering as well as context-aware recommender system.

A. Collaborative Filtering

The basic idea behind collaborative filtering is that the users who have similar preferences in the past tend to behave similarly in the future [1]. The recommendations made by this methodology are based on information about similar users and items. Collaborative filtering methods only rely on user ratings or user interactions. That means there is no need to have additional information about items or users. Moreover, the user's ratings can be acquired explicitly or implicitly (e.g., products bought, songs heard, movies watched, visited pages) [4], so collaborative filtering methods can be used even when the user does not explicitly provide ratings for the items. In the literature, collaborative filtering methods can be grouped in two general classes (i) memory-based techniques and (ii) model-based techniques.

In memory-based technique, the rating history is directly used to predict rating of items that the user has not yet seen. This can be done in two ways: (i) user-based collaborative filtering and (ii) item-based collaborative filtering. In the former method, a set of neighbor users will be selected based on similarity in their rating history to the targeted user. Then, the recommendations will be produced based on top-rated products liked by neighbor users. The item-based collaborative filtering is just an analogous procedure to the previous method. Here, for each item, a set of k -nearest items will be selected. Then, for every product that the target user has not rated before, we estimate the rating using the closest neighbors which are rated previously by the target user. It is important to note that every neighbor has a weight, which reflects the degree of similarity, that will be used in process of rating estimation. However, the most popular metrics used to calculate the similarity between different users, or items, are cosine similarity and Pearson correlation.

In contrast to memory-based technique, which uses the stored ratings directly in the prediction, the model-based technique use these ratings to learn a predictive model. Basically, the learning process is based on matrix factorization which uses the rating history to learn the latent preferences of users and items in order to make a prediction for the missing ratings. Matrix factorization is an unsupervised learning method for dimensionality reduction. The most popular techniques applied for dimensionality reduction are Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Probabilistic Matrix Factorization (PMF), Matrix Completion Technique, Latent Semantic methods, and Regression and Clustering [2] [5].

B. Context-aware Recommender System

The basic idea behind context-aware recommender systems (CARS) is to incorporate the contextual information into recommendation process in order to recommend more relevant items to users under certain circumstances [2] [3]. Many researchers and practitioners have recognized that it is very important to consider relevant contextual information, such as weather, time, location and mood, when providing recommendations. For example, the vacation packages proposed by a travel agency in the winter can be very different from the one proposed in the summer. Thus, the main goal of context-aware recommender system is to consider the contextual information when providing recommendations [6]. After gathering the relevant context, explicitly, implicitly or by inferring, the following question arises: how can we incorporate the context in the recommendation process?. However, Adomavicius and Tuzhilin [7] identified three different approaches to achieve this goal as follows:

- Contextual pre-filtering: here the context information is used to select only the most relevant data from the data set. In other words, information about the current context is used for selecting the relevant set of data records (i.e., ratings). Then, ratings can be predicted using

any traditional 2D recommender system on the selected data [8].

- Contextual post-filtering: the context information is ignored during the recommendation process, only the resulting set is contextualized. In this approach, the contextual information is initially ignored, and the ratings are predicted using any traditional 2D recommender system on the entire data. Then, the resulting set of recommendations is adjusted (contextualized) for each user using the contextual information [8].
- Contextual modeling: the recommendation algorithm is altered to include the context and consider it when calculating recommendations. In other words, the contextual information is used directly in the modeling technique as part of rating estimation [8] [9].

III. RELATED WORKS

Before presenting CoCl, we review some of the research literature related to contextual collaborative filtering approaches that utilize contextual information to improve recommendation quality.

Over the past decade, a lot of research concerned with context-aware recommender systems has been presented. Palmisano et al. [10] has proved that contextual information, such as age, time, and location, is very useful when predicting customer behavior. Recently, many researches started focusing on the use of context for user-item sub-grouping. Zhong et al. [11] and Liu et al. [12] used decision trees to partition the original rating matrix hierarchically by grouping similar users and items together, and then using the matrix factorization technique to predict missing preference of a user for an item using the partitioned matrix. The previous sub-grouping methods can only handle categorical contexts, and to mitigate this problem, Xiaolin Zheng et al. [13] proposed the use of spectral clustering for user-item sub-grouping, which can handle both categorical and continuous contexts. A new recommender system which is based on matrix factorization has been proposed by Xiaoyao et al. [14]. They considered many factors while building the recommender system, such as contextual information, user ratings and item feature. Using K-modes algorithm, they optimized the process of building the recommender system by clustering user-item dataset which eventually reduces the computation complexity. However, they performed extensive experiments to demonstrate that their method improves the accuracy of generated recommendations.

According to previous findings, sub-grouping has been proved to be valuable for better performance in collaborative filtering methods, but we believe there is space left for further improvements by discovering more advanced grouping approaches.

IV. COCL RECOMMENDER SYSTEM

In this section, we present CoCl, a Context Clustering based recommender system. We first formalize the context-aware recommendation problem. Then, we describe our proposed contextual clustering model that is used in CoCl.

A. Problem Definition

The main problem we address in this paper is to improve the traditional collaborative filtering approach by incorporating the context in the process of building the recommender system. The main idea is to produce new forms of user-item matrices, also known as utility matrices, by clustering, aggregating and splitting the records in this matrix. However, two approaches will be provided for grouping or clustering the rating records in utility matrix. In the first approach, called RateClust, the ratings in the utility matrix will be grouped in such a way that the ratings with similar contextual information will be together. In the second approach, called UserClust, the users in the utility matrix will be grouped based on their ratings in dedicated contexts.

B. Contextual Clustering Model

In this section, we introduce the reader to our proposed recommendation model. As mentioned before, CoCl proposes a hybrid model that utilizes the contextual information and KMeans clustering algorithm to create new forms of user-item matrices. Then, we apply the traditional collaborative filtering approach on these new matrices to get many recommender systems which give us more accurate results than building one recommender system using the original dataset. In this experiment, we present two approaches to utilize the contextual information for the purpose of clustering the data in user-item matrix: (i) RateClust, and (ii) UserClust.

In RateClust, we aim to group the ratings that are given in similar contexts. As we mentioned in previous section, the contextual information in our dataset describes the situation in which the user consumed the item. In this approach, the contextual information space is represented by an array of vectors where every vector, i.e. $C = (c_1, c_2, \dots, c_l)$ represents the contexts associated with one rating in user-item matrix. The values in this vector describes the situation for every context variable. For instance, the first context variable in our dataset is time which is represented by five values as follows: morning = 1, afternoon = 2, evening = 3, night = 4, missing value = -1. After creating this array of vectors, we passed it to KMeans algorithm that helps us to cluster the ratings, and subsequently divide the user-item matrix into smaller parts.

On the other hand, our goal in UserClust, is to group the users that share the same behaviour in similar contexts. In other words, we aim to cluster the users based on their ratings in particular context. For instance, in our experiment, we select the mood context which is represented by positive, neutral, negative and missing (unknown). Then, for each user, we calculate the average rating given in every mood possible value. So, for every user, we have average rating for positive, neutral, negative and missing values. The result of applying previous step for all users is an array of vectors which is used to cluster the users. Finally, the output of users' clustering is utilized to split the user-item matrix into smaller groups which contain the users who rate the items similarly in the same context.

After clustering the records in the original user-item matrix using previous approaches, we generate two new versions of user-item matrix by aggregating the ratings given for each movie in each cluster. For instance, if the same movie has been rated by two users who belong to the same cluster, then the ratings given by both users will be replaced by their average. The new generated matrices can be utilized in different ways while building the recommender system. One way is to divide the aggregated user-item matrix into smaller matrices based on the cluster the records belong to. Then, many recommender systems can be build using these smaller matrices. However, this approach is useful when we have enough number of records belong to each cluster. Another way is to just build one recommender system without dividing the aggregated user-item matrix. More information about these approaches will be provided in the next section while comparing the performance of CoCl models with traditional collaborative filtering model.

V. EVALUATION FOR CoCl

In this section, we conduct comprehensive experiments to evaluate the performance of CoCl by comparing the recommendations accuracy with classical collaborative filtering recommender system.

A. Dataset

In our experiments, we used LDOS-CoMoDa¹ dataset which is presented by KOŠIR et al. [15]. LDOS-CoMoDa is a context rich movie recommender dataset that consists of 200 users, who gave 2296 ratings for 4138 movies in twelve pieces of contextual information. However, the contextual information is explicitly acquired from the users directly after watching the movies. Moreover, this dataset is collected from real user-item interaction and not from hypothetical situation or user's memory of past interactions. The context variables in LDOS-CoMoDa dataset are presented in Table I. The values of context variables in this dataset is represented by numerical values. For example, in daytype variable, Working day is represented by 1, Weekend by 2, Holiday by 3.

TABLE I: Context variables in LDOS-CoMoDa dataset

time	Morning, Afternoon, Evening, Night
daytype	Working day, Weekend, Holiday
season	Spring, Summer, Autumn, Winter
location	Home, Public place, Friend's house
weather	Sunny / clear, Rainy, Stormy, Snowy, Cloudy
social	Alone, My partner, Friends, Colleagues, Parents, Public, My family
endEmo	Sad, Happy, Scared, Surprised, Angry, Disgusted, Neutral
dominantEmo	Sad, Happy, Scared, Surprised, Angry, Disgusted, Neutral
mood	Positive, Neutral, Negative
physical	Healthy, Ill
decision	User decided which movie to watch, User was given a movie
interaction	first interaction with a movie, n-th interaction with a movie

¹<https://www.lucami.org/en/research/ldos-comoda-dataset/>

Also, it is worth to note that the entire dataset is splitted into training, testing and validation sets where:

- Training set is used to build the recommender systems using several algorithms, where each algorithm has pre-defined set of input parameters. The optimal values of these parameters is discovered using the validation set.
- Testing set is used to compare the accuracy of different recommender systems.
- Validation set is used to discover (i) the optimal parameters settings for each recommender system algorithm and (ii) the optimal number of clusters. More details about these two steps will be provided in next subsection.

B. Parameters Selection

In order to evaluate our model, various collaborative filtering algorithms have been applied to generate recommendations based on user-item rating matrix. We use Surprise library which provides various implementations of collaborative filtering algorithms [16]. These algorithms are (i) matrix factorization-based algorithms, such as singular value decomposition (SVD) and non-negative matrix factorization (NMF) and (ii) k-nearest neighbors-based algorithms, such as KNNBasic, KNNBaseline, KNNWithMeans and KNNWithZScore and (iii) other types of algorithms, such as CoClustering, SlopeOne, NormalPredictor and BaselineOnly. The main challenge here is to determine the optimal values of hyper-parameters for every algorithm. This is extremely important since the performance of the recommender system will be impacted based on those values. Moreover, selecting optimal parameters settings for every algorithm is also important to conduct fair and reliable experiments. To tackle this challenge, we use grid search with cross-validation (GridSearchCV) tool which automates the process of tuning the hyper-parameters for every algorithm mentioned before.

On the other hand, we used silhouettes score to select the optimal number of clusters for both versions of CoCl, RateClust and UserClust. The silhouettes score is calculated for each instance based on below formula:

$$SilhouetteScore = (x - y) / \max(x, y)$$

where, y is the mean distance to the other instances in the same cluster (mean intra-cluster distance), x is mean distance to the instances of the next closest cluster (mean nearest-cluster distance).

The silhouettes score, or silhouettes coefficient, varies between 1 and -1 . A value close to 1 implies that the instance is far away from the neighboring clusters; hence, it is a part of the right cluster. Whereas, a value close to -1 indicates that the value is assigned to the wrong cluster. A value close to 0 means that the instance is very close to the decision boundary between two neighboring clusters [17].

Fig 1 shows that 8 is the optimal number of clusters that needs to be selected to group the users based on their ratings in given context.

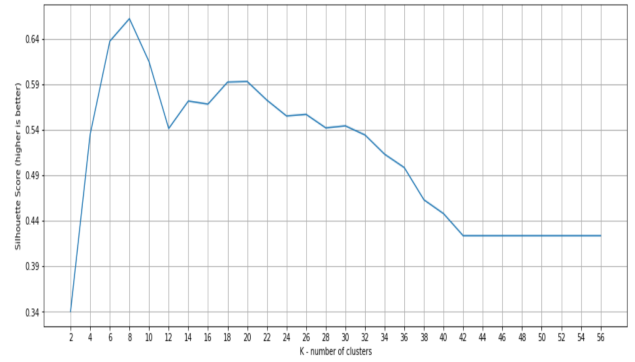


Fig. 1: Calculating mean silhouettes score over all samples for different number of clusters

C. Performance Comparison and Analysis

We use the standard Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) metrics to measure and compare the performance of various recommendation models. The RMSE imposes a penalty over the larger errors:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - x'_i)^2} \quad (1)$$

While MAE measures the average magnitude of the errors in a set of predictions, without considering their direction:

$$MAE = \frac{1}{N} \sum_{i=1}^N |x_i - x'_i| \quad (2)$$

After generating the new utility matrices, aggregating the ratings, and calculating the weights, we evaluate the recommendation accuracy of context clustering based models by comparing it with classical collaborative filtering model. For the sake of more accurate evaluation, we use different approaches in our comparisons.

We start evaluating CoCl recommender system which utilizes the contexts to build two types of recommendation systems by clustering the ratings (RateClust), and clustering the users (UserClust) in utility matrix. The recommendation systems produced by CoCl will be evaluated using four methods.

In the first one, we use cross-validation method to check how well the model is able to make new predictions for data which has not seen before. Using k-fold cross-validation is very useful in such scenario when the size of dataset is considered as small. In this method, we use the entire aggregated dataset to build each recommender system in CoCl. we split this data into two parts, training, and testing. The training, which is equal to 85 percent of entire dataset, is used to evaluate every model using repeated cross-validation method. While the rest of the data is used as testing set for final general evaluation. The same criteria is applied to split the original, not aggregated, dataset which is used to evaluate classical recommender system. With the objective of

conducting fair and reliable comparison, we ensure that the same records have been used in every fold while evaluating all models using cross-validation method. The only difference between the folds is that the aggregated ratings have been used to evaluate CoCl models while the original ratings have been used to evaluate traditional model. We compare the accuracy between CoCl and classical models by calculating the average of RMSE and MAE which are generated in each fold. However, we repeat the same comparison by using different algorithms to build the recommender system. The comparison results of this evaluation methodology for RateClust and UserClust are shown in Tables II and III respectively. Moreover, Fig 2 shows the generalization assessment of final models fit on entire training set. It is important to note that the performance of CoCl models outperforms the performance of traditional model in every iteration of cross-validation method. In conclusion, this experiment shows that CoCl models reduce MAE/RMSE by around 6% on average compared to classical collaborative filtering model.

In the second evaluation methodology, we use holdout evaluation method where the entire aggregated dataset is splitted into training and testing sets. The training and testing records have been selected in a way such that from each cluster we select 85 percent of data as training and 15 as testing. In this way, we guarantee that records from all clusters have been included in training and testing sets. For classical recommender system, we select the same records which are selected before as training and testing but from original user-item matrix without any aggregation in ratings. After that, we compare the accuracy by calculating RMSE and MAE for each recommender system. However, we repeat the same comparison by using different algorithms to build the recommender system. The comparison results of this evaluation methodology are shown in Tables IV and V. This experiment demonstrates that using contextual information improves recommendation quality.

In the third evaluation methodology, we split the records in aggregated user-item matrix into smaller matrices based on the clustering results. Then, we build a clustering based recommender systems for each one of them. These recommender systems will be compared with classical recommender system which is created based on original dataset without any splitting or aggregation. The comparison results of this evaluation methodology are shown in Tables VI and VII. It is important to note that the recommender systems which are generated using the small aggregated utility matrices perform better than the one which is generated based on entire aggregated matrix.

In the fourth method, we create ensemble recommender systems for RateClust, UserClust and classical models. The main idea is to aggregate the ratings produced by each algorithm in order to produce the final ratings in the target recommender system. For the sake of improving the results, we select the best three algorithms that produce the most accurate results in previous evaluation methods. These algorithms are SVD, KNNBaseline and BaselineOnly. However, while building the clustering-based recommender systems; the entire aggregated

dataset is used without any splitting. The comparison results of this evaluation methodology are shown in Tables VIII and IX. The results indicate that the clustering based recommender systems achieve better accuracy than the one which is produced using original dataset. However, RateClust models has slightly better accuracy than UserClust models.

We notice that in all experiment scenarios, CoCl models outperform traditional collaborative filtering model. Moreover, the experiment results demonstrate the advantage of considering the contextual information in the area of recommender systems.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed CoCl, a novel context clustering based recommender system, which methodically incorporates the context in the process of generating the recommendations. Two versions of CoCl have been introduced, RateCust, rating-based clustering recommender system, and UserClust, user-based clustering recommender system. We proposed to use KMeans clustering algorithm to cluster the data in user-item matrix in order to produce new forms of utility matrices which can achieve better accuracy using collaborative filtering techniques. To evaluate our proposed models, we conducted comprehensive experiments on LDOS-CoMoDa dataset using Surprise library which provides various implementations of collaborative filtering algorithms that can be used for building and analyzing recommender systems. Moreover, multiple evaluation methodologies have been proposed to compare between models. The experimental results can reveal the answer for our research question stated above. The results illustrate that CoCl accuracy outperforms classical collaborative filtering approach in all experiments. However, experiments also indicate that RateClust approach has slightly better performance than UserClust approach.

In the future work, we are interested in applying CoCl to some real world application scenarios. For instance, CoCl can be integrated into a website where the recommender system can generate some recommendations in real time based on the current context. Also, we need to take into account the dynamics of evolving user preference by periodically updating the user-item matrices based on recent recommendations. Moreover, in the scenario of splitting the user-item matrix into smaller ones based on clusters, we need to determine in real time the proper recommender system that can produce the best recommendations for specific user and in dedicated context. Also, LDOS-CoMoDa is considered to be rather small dataset, and hence, another suggestion for future work is to evaluate CoCl against larger and more complex datasets.

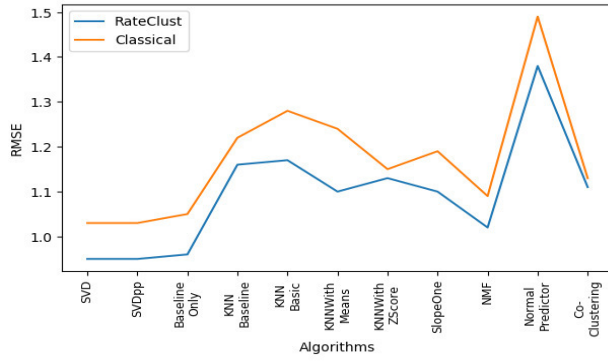
On the other hand, in our research we focused on memory based techniques in collaborative filtering, so another important aspect to consider is to evaluate CoCl using model based techniques. Furthermore, we are particularly interested in using more advanced machine learning techniques to incorporate the contexts in recommendation systems. Also, another direction of future work is to use distributed stream processing engines, like Apache Flink, to examine parallel

TABLE II: Cross Validation - Rating-based clustering VS Classical

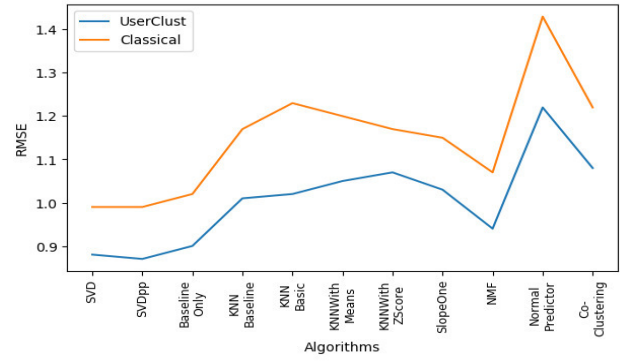
Model	Metric	SVD	SVDpp	Baseline Only	KNN Baseline	KNN Basic	KNN WithMeans	KNN WithZScore	Slope One	NMF	Normal Predictor	Co-Clustering
RateClust	RMSE	0.96	0.95	0.95	1.07	1.08	1.08	1.09	1.06	1.03	1.36	1.10
	MAE	0.76	0.76	0.76	0.78	0.79	0.82	0.82	0.82	0.83	1.08	0.83
Classical	RMSE	1.01	1.01	1.02	1.14	1.18	1.16	1.12	1.14	1.09	1.41	1.14
	MAE	0.81	0.81	0.82	0.87	0.89	0.89	0.85	0.89	0.88	1.13	0.88

TABLE III: Cross Validation - User-based clustering VS Classical

Model	Metric	SVD	SVDpp	Baseline Only	KNN Baseline	KNN Basic	KNN WithMeans	KNN WithZScore	Slope One	NMF	Normal Predictor	Co-Clustering
UserClust	RMSE	0.95	0.94	0.96	1.13	1.13	1.08	1.08	1.09	1.01	1.33	1.13
	MAE	0.75	0.74	0.76	0.84	0.84	0.80	0.81	0.84	0.81	1.04	0.86
Classical	RMSE	0.99	0.99	1.03	1.17	1.23	1.20	1.17	1.15	1.06	1.53	1.18
	MAE	0.80	0.80	0.84	0.90	0.93	0.93	0.91	0.91	0.87	1.21	0.91



(a) Rating-based clustering VS Classical



(b) User-based clustering VS Classical

Fig. 2: Generalization assessment using testing set

TABLE IV: Performance Comparison - Rating-based clustering (one recommender system) VS Classical

Model	Metric	SVD	SVDpp	Baseline Only	KNN Baseline	KNN Basic	KNN WithMeans	KNN WithZScore	Slope One	NMF	Normal Predictor	Co-Clustering
RateClust	RMSE	0.86	0.85	0.86	1.02	1.03	1.02	1.06	0.98	0.92	1.31	0.99
	MAE	0.69	0.68	0.68	0.72	0.73	0.78	0.79	0.75	0.73	1.07	0.76
Classical	RMSE	0.94	0.94	0.95	1.11	1.08	1.06	1.08	1.05	1.00	1.38	1.11
	MAE	0.76	0.77	0.76	0.83	0.83	0.80	0.83	0.82	0.80	1.12	0.86

TABLE V: Performance Comparison - User-based clustering (one recommender system) VS Classical

Model	Metric	SVD	SVDpp	Baseline Only	KNN Baseline	KNN Basic	KNN WithMeans	KNN WithZScore	Slope One	NMF	Normal Predictor	Co-Clustering
UserClust	RMSE	0.89	0.89	0.89	0.95	0.96	1.00	1.05	0.96	0.98	1.34	0.97
	MAE	0.69	0.70	0.70	0.68	0.68	0.75	0.78	0.75	0.76	1.07	0.73
Classical	RMSE	1.04	1.03	1.05	1.12	1.18	1.15	1.14	1.17	1.14	1.49	1.18
	MAE	0.84	0.84	0.85	0.87	0.89	0.89	0.88	0.92	0.92	1.22	0.93

TABLE VI: Performance Comparison - Rating-based clustering (Multiple recommender systems) VS Classical

Model	Metric	SVD	SVDpp	Baseline Only	KNN Baseline	KNN Basic	KNN WithMeans	KNN WithZScore	Slope One	NMF	Normal Predictor	Co-Clustering
RateClust	RMSE	0.78	0.79	0.78	0.82	0.83	0.80	0.80	0.86	0.82	1.17	0.86
	MAE	0.61	0.62	0.60	0.61	0.62	0.56	0.56	0.66	0.62	0.95	0.67
Classical	RMSE	0.94	0.93	0.95	1.08	1.12	1.08	1.06	1.05	1.00	1.35	1.06
	MAE	0.77	0.76	0.76	0.83	0.83	0.83	0.80	0.82	0.80	1.06	0.83

TABLE VII: Performance Comparison - User-based clustering (Multiple recommender systems) VS Classical

Model	Metric	SVD	SVDpp	Baseline Only	KNN Baseline	KNN Basic	KNN WithMeans	KNN WithZScore	Slope One	NMF	Normal Predictor	Co-Clustering
UserClust	RMSE	0.88	0.89	0.88	0.89	0.90	0.88	0.89	0.93	1.04	1.26	0.91
	MAE	0.68	0.69	0.69	0.64	0.64	0.60	0.61	0.71	0.82	1.00	0.68
Classical	RMSE	1.04	1.03	1.05	1.12	1.18	1.15	1.14	1.17	1.14	1.39	1.15
	MAE	0.85	0.83	0.85	0.87	0.89	0.89	0.88	0.92	0.92	1.12	0.92

TABLE VIII: Rating-based clustering VS Classical (Ensemble Recommender System)

model	Ensemble Recommender System	
	RMSE	MAE
RateClust	0.95	0.74
Classical	1.03	0.79

TABLE IX: User-based clustering VS Classical (Ensemble Recommender System)

model	Ensemble Recommender System	
	RMSE	MAE
UserClust	1.00	0.80
Classical	1.08	0.86

implementations of CoCl, in order to make them scalable to infinite streams or large-scale datasets.

ACKNOWLEDGMENT

This paper was partially supported by the grant nr POIR.01.01.01-00-1307/17 entitled by "Development of a hybrid recommendation system for the fashion & cosmetics branch operating, among others based on contextual data using machine-learning mechanisms" co-financed by the European Regional Development Funds, a part of the Smart Growth Operational Programme.

REFERENCES

- [1] J.S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI1998), 1998.
- [2] F. Ricci, L. Rokach, and B. Shapira, Eds., "Recommender Systems Handbook," Springer New York Heidelberg Dordrecht London, 2015, doi: 10.1007/978-0-387-85820-3.
- [3] A. Lommatzsch, B. Kille, and S. Albayrak, "Incorporating context and trends in news recommender systems," In Proceedings of the International Conference on Web Intelligence (WI '17). ACM, New York, NY, USA, 1062-1068, 2017, doi: 10.1145/3106426.3109433.
- [4] S.K. Lee, Y.H. Cho, and S.H. Kim, "Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations," Information Sciences 180 (11) (2010) 2142–2155, doi: 10.1016/j.ins.2010.02.004.
- [5] L.E.M. FERNÁNDEZ, "Recommendation System for Netflix," VRIJE UNIVERSITEIT AMSTERDAM, 2018.
- [6] F. Shi, C. Ghedira, and J.-L. Marini, "Context Adaptation for Smart Recommender Systems," IEEE Computer Society 1520-9202/15/31.00 © 2015 IEEE, doi: 10.1109/MITP.2015.96.
- [7] G. Adomavicius, and A. Tuzhilin, "Chapter 6: Context-Aware Recommender Systems," in Recommender Systems Handbook, F. Ricci, L. Rokach and B. Shapira, Eds., Springer, Boston, MA, 2015, doi: 10.1007/978-1-4899-7637-6_6.
- [8] U. Panniello, A. Tuzhilin, M. Gorgoglione, C. Palmisano, and A. Pedone, "Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems," Proceedings of the 2009 ACM Conference on Recommender Systems, 2009, doi: 10.1145/1639714.1639764.
- [9] Y. Shen, Y. Deng, A. Ray, and H. Jin, "Interactive Recommendation via Deep Neural Memory Augmented Contextual Bandits," In Proceedings of RecSys 2018 – the ACM Conference Series in Recommendation systems, Vancouver, 2018, doi: 10.1145/3240323.3240344.
- [10] C. Palmisano, A. Tuzhilin, and M. Gorgoglione, "Using context to improve predictive modeling of customers in personalization applications," Knowledge and Data Engineering, IEEE Transactions on 20(11):1535–1549, 2008, doi: 10.1109/TKDE.2008.110.
- [11] E. Zhong, W. Fan, and Q. Yang, "Contextual collaborative filtering via hierarchical matrix factorization," In Proceedings of the SIAM International Conference on Data Mining, 744–755, 2012, doi: 10.1137/1.9781611972825.64.
- [12] X. Liu, and K. Aberer, "Soco: a social network aided context-aware recommender system," In Proceedings of the 22nd international conference on World Wide Web, 781–802, 2013, doi: 10.1145/2488388.2488457.
- [13] C. Chen, X. Zheng, Y. Wang, F. Hong, and Z. Lin, "Context-ware Collaborative Topic Regression with Social Matrix Factorization for Recommender Systems," In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. 9-15, 2014.
- [14] X. Zheng, Y. Luo, L. Sun, and F. Chen. 2016. "A New Recommender System Using Context Clustering Based on Matrix Factorization Techniques," Chinese Journal of Electronics. Vol.25, No.2, 2016, doi: 10.1049/cje.2016.03.021.
- [15] A. Kosir, A. Odic, M. Kunaver, M. Tkalcic, and J. F. Tasic, "Database for contextual personalization," Elektrotehnikski Vestnik/Electrotechnical Review, vol. 78, pp. 270–274, 2011.
- [16] N. Hug, "Home," Surprise. [Online]. Available: <http://surpriselib.com/>.
- [17] P.J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," Journal of Computational and Applied Mathematics 20, pp. 53-65, 1987, doi: 10.1016/0377-0427(87)90125-7.