



HAL
open science

Towards better representation of context into recommender systems

Jinfeng Zhong, Elsa Negre

► **To cite this version:**

Jinfeng Zhong, Elsa Negre. Towards better representation of context into recommender systems. The International Conference on Information and Knowledge Systems (ICIKS 2021), Jun 2021, virtual conference, France. hal-03529597

HAL Id: hal-03529597

<https://hal.science/hal-03529597>

Submitted on 17 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards better representation of context into recommender systems

Jinfeng ZHONG and Elsa NEGRE

Paris-Dauphine University, PSL Research University
CNRS UMR 7243, LAMSADE 75016 Paris France

Abstract. Context-aware recommender systems (CARSs) are attracting more and more attention from both the academic community and from industry. Users' contextual situations (e.g. location, time, companion, etc) which can influence their ratings on items, are taken into consideration. Therefore, more accurate and personalized recommendations can be generated. The integration of contextual information in recommender system to better model users' preferences under different contextual situations is a key research topic. In this paper, we propose a new method for representing contextual situations in recommender systems based on the influence of contextual conditions on ratings using Pearson Correlation Coefficient. We show the effectiveness of the proposed method compared to state-of-art methods by experiments on three different datasets widely used in CARSs research community.

Keywords: Context-aware recommender systems · Decision support.

1 Introduction

With the large amount of information and services available online, recommender systems have become an important tool for helping people find more relevant information or services. Traditional recommender techniques can be classified into three types [3]: content-based techniques, collaborative filtering techniques and hybrid techniques. These techniques have reached great success in numerous domains of application especially in developing decision support systems. The intuition behind these techniques is that people will prefer similar items (e.g. information, services) that they like in the past or similar users will have similar preferences. However, they do not consider users' context and suppose that users' preferences are static, ignoring the impacts of context on users' preferences.

Context-aware recommender systems (CARSs), on the contrary, can generate more accurate and personalized recommendations by leveraging users' contextual situations [2]. Users' contextual situations are composed of several contextual conditions. For example, a contextual situation of a target user can be: (Morning, With Friends, At home), indicating that the target user is at home with his/her friends when the time is morning. Representing contextual situations to better model users' preferences is a key challenge in CARSs research. State-of-art methods include key-value [20], ontology-based [15], etc. Key-value methods

are simple but are less expressive than ontology-based methods [15]. Meanwhile, ontology-based methods require domain-specific ontology and it is hard to model a generic ontology that can fit all applications. Besides, modeling such ontologies can be time consuming. Recently, a data-driven method for representing contextual situations and measuring their similarities in recommender system was proposed by [10]. The authors represented contextual situations based on the influence of contextual conditions on ratings using Pearson Correlation Coefficient (PCC) [7]. Based on their framework, we propose a new method for representing contextual situations in recommender system.

The remainder of this paper is structured as follows: In Section 2 we review the state-of-art works related to CARSs; we present our method to represent contextual situations in recommender system in Section 3; we present the experiment set up in Section 4; results of experiments on three CARS datasets are presented in Section 5; lastly we propose potential future work in Section 6.

2 Related work

2.1 Context in recommender system

Context-aware recommender systems (CARSs) assume that users’ preferences are dynamic and change over different contexts. While the notion “context” has been studied by researchers of various domains (e.g. psychology, artificial intelligence, information retrieval), no common consensus about the definition of it has been achieved. Researchers have studied context in recommender systems from different points of views, which varies from application domains. [1] defined that context is any information that can be used to characterize the situation of an entity. Based on this definition, [11] proposed a hierarchical categorization of contextual factors in recommender system. According to [11], contextual factors can be classified into physical context, personal context and technical context, which will be the definition adopted in this paper. For example, in a movie recommender system, time (e.g. workday, weekend), companion (e.g. friends, lovers or alone), mood (happy, sad or neutral) can influence a user’s choice of movies; in a trip recommender system, weather (sunny, rainy), season (spring, summer), distance (near, far) can influence a user’s choice of destination. The “time”, “companion” and “season” are contextual factors and for each contextual factor, it can have several possible values, these values are contextual conditions and the contextual conditions compose contextual situations. Incorporating such information in recommender systems helps improve recommendation quality since more personalised propositions can be generated [2].

2.2 Representation of contextual situation in CARSs

In traditional recommender methods, rating function is bi-dimensional ($Rating_{RS} : users \times items \rightarrow rating$), while in CARSs it is multidimensional ($Rating_{CARS} : users \times items \times context \rightarrow rating$). Depending on the way contextual information

is integrated, CARSs can be classified into three categories [4]: pre-filtering, contextual modeling, post-filtering. Pre-filtering methods first select relevant data according to target users’ contextual situations, then a context-free method is applied to generate recommendations; contextual modeling methods, integrate contextual information directly in recommender algorithms; post-filtering methods first apply a context-free method to generate recommendation candidates, then contextual information is used to reorder the candidates. Among the three methods mentioned as above, there is no single winner [4]. The advantage of pre-filtering and post-filtering methods compared to contextual filtering is that existing system can be easily adapted to integrate contextual information. Existing system can be reused, which saves lots of effort. However the effectiveness depends on the real-world datasets and application domain.

As we discussed in Section 1, ontology-based methods for representing contextual information are more expressive but ontologies are usually domain specific and can not be used in deployment. Key-value methods are simple but less expressive. Therefore, we turn to data-driven methods to represent contextual situations. This is also the strategy adopted by [10]. The authors used a pre-filtering approach and represented users’ contextual situation based on the influence of contextual conditions on ratings using Pearson Correlation Coefficient (PCC) [7], which is called CBPF (Correlation-Based Pre-Filtering). PCC is chosen since it can measure the strength of linear association between two variables and therefore can catch the influence of context conditions on rating [10]. In CARSs, the PCC between contextual conditions and ratings can be used to measure the impacts of contextual conditions on ratings. Based on the similarity of target users’ contextual situation and the existing contextual situations in dataset, a relevant local dataset is selected and Biased Matrix Factorization algorithm proposed by [14] is applied to generate recommendation. The intuition behind is that people will rate an item similarly under similar contextual situations and people will have similar preferences under similar contextual situations. The authors compared their method with the state-of-art methods in CARSs on three real world datasets and proved its effectiveness in terms of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), calculated by Equation 1 and Equation 2 respectively, where N is the number of observations in a dataset, r_k is the actual rating of the k^{th} observation and \hat{r}_k is the predicted rating of the k^{th} observation. They measure the differences between actual ratings and predicted ratings, the smaller they are, the better the performances are.

$$MAE = \frac{1}{N} \sum_{k=1}^{k=N} |r_k - \hat{r}_k| \quad (1)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^{k=N} (r_k - \hat{r}_k)^2} \quad (2)$$

The advantages of the method proposed by [10] is the following: (1) No external resources like domain specific ontology is required to represent contextual

Table 1. List of key symbols

Symbol	Meaning
$U = \{u_1, u_2 \dots u_m\}$	Set of users
$I = \{i_1, i_2 \dots i_n\}$	Set of items
$C = \{c_1, c_2 \dots c_p\}$	Set of contextual conditions
$CF = \{cf_1, cf_2 \dots cf_q\}$	Set of contextual factors
$X_k = \{u_k, i_k, r_k, c_{1k}, c_{2k} \dots c_{pk}\}$	The k^{th} observation in dataset
$w_{c_j} = (w_{u_{cluster1}, c_j}, w_{u_{cluster2}, c_j}, \dots)$	Representation of contextual condition c_j
$CS = (c_1, \dots c_q)$	A target contextual situation

situations; (2) The method belongs to pre-filtering methods, therefore, it can be easily integrated into existing systems, reducing the engineering costs. In this paper we aim to improve CBPF by adopting more relevant representation of contextual information. More details will be presented in Section 3.

3 Methodology

We will first introduce the CBPF method proposed by [10] and then present our modification, the symbols used are presented in Table 1. To be more concrete, in a CARS, suppose there are m users; n items; q contextual factors (e.g. time, companion, location, etc); suppose we have $|c_k|$ (e.g. possible condition for companion is family, friend, colleague, etc.) possible conditions for contextual factor c_k , then $\sum_{k=1}^{k=q} |c_k| = p$, where p is the total number of contextual conditions and q is the total number of contextual factors; X_k is the k^{th} observation in the dataset, r_k is the rating given by user u_k to item i_k , $c_{tk} = 1$ means contextual condition c_{tk} appears in k^{th} observation and $c_{tk} = 0$ means absent. For example, in a movie recommender system where ratings range from 1 to 5, $X_k = \{Mike, HarryPotter, 4, DaytypeWeekday = 1, DaytypeWeekend = 0, CompanionFamily = 0, CompanionFriends = 1, CompanionAlone = 0, MoodHappy = 1, MoodDisappointed = 0, MoodAngry = 0\}$ means that Mike watched the movie Harry Potter with his friends when he was happy in a weekday and he gave a score 4 out of 5 for this movie. A contextual condition c_j can be represented by $w_{c_j} = (w_{u_{cluster1}, c_j}, w_{u_{cluster2}, c_j}, \dots)$, the values are user-based (or item-based) PCC between contextual condition c_j and ratings calculated by Equation 3; a target contextual situation $CS = (c_1, \dots c_q)$ is composed of q contextual conditions, which means that for each of the q contextual factors, only one contextual condition occurs in a target contextual situation. Considering a contextual factor “time”, it can not be “morning” and “afternoon” at the same time.

The main steps of CBPF include:

- **Step 1:** Calculate the user-based (or item-based) PCC between contextual condition c_j and ratings for each user (item) using Equation 3. Note that for user-based PCC, the contextual conditions and ratings are selected according

to users; for item-based PCC, the contextual conditions and ratings are selected according to items.

$$w_{u,c_j} = PCC_u(r, c_j) = \frac{\sum_{k \in K} (r_k - \bar{r}_u)(c_{jk} - \bar{c}_u)}{\sqrt{\sum_{k \in K} (r_k - \bar{r}_u)^2} \sqrt{\sum_{k \in K} (c_{jk} - \bar{c}_u)^2}} \quad (3)$$

where K is the set of observations $X_k = \{u, i_k, r_k, c_{1k}, c_{2k} \dots c_{pk}\}$ with user u , \bar{c}_u is the mean value of the context condition c_j over observations for user u , \bar{r}_u is the mean of the ratings given by the user u . Based on the calculated w_{u,c_j} , each contextual condition c_j can be represented by a vector whose size is the total number of users (or items) and whose values are the corresponding w_{u,c_j} , all between -1 and 1. Considering the large number of users and items, [10] proposed to first cluster the users and items into limited groups to reduce computation costs. For example, the characteristics such as age and/or sex, can be used to cluster users. Table 2 is an example of this representation method. To be more clear, the values correspond to the PCC between the contextual conditions on the left and the ratings of a user cluster. For example, the first row in Table 2 is the contextual condition ‘‘DaypeWeekday’’, its PCC with ratings for user cluster1, user cluster2 and user cluster3 is 0.18, -0.13 and -0.54 respectively.

- **Step 2** Represent contextual target situation based on its composing context conditions. In this step [10] used 2 methods:
 - Aggregation: The mean of corresponding representation vector of contextual conditions which compose a target contextual situation:

$$w_{CS} = \frac{\sum_{k=1}^{k=q} w_{c_k}}{q} \quad (4)$$

- Concatenation: The concatenation of the composing contextual conditions:

$$w_{CS} = \text{concat}(w_{c_k}) \quad (5)$$

- **Step 3** Calculate the similarity between the target contextual situation CS^* and the contextual situations CS existing in the original dataset, the similarity here is the cosine similarity between the vectors that represent them.

$$\text{sim}(CS^*, CS) = \text{cosine}(w_{CS^*}, w_{CS}) \quad (6)$$

- **Step 4** Select a local dataset whose contextual situation is similar to the target contextual situation according to a similarity threshold. (Set as 0.5)
- **Step 5** A traditional 2D recommender technique is applied in the local dataset to obtain recommendation.

The steps above have been proved to be more effective than state-of-art methods [10]: DSPF (Distributional Semantic Pre-Filtering) [9], Deviation-based CAMF (Context-Aware Matrix Factorization) [6], DCM (Differential Context Modeling) [22].

Table 2. Examples of representation of contextual conditions (User-based and user clustered) (Step 1)

	Cluster1	Cluster2	Cluster3
DaytypeWeekday	0.18	-0.13	-0.54
DaytypeWeekend	0.45	0.56	0.62
CompanionFamily	0.67	0.12	-0.34
CompanionFriends	0.51	0.45	0.12
CompanionAlone	-0.18	-0.34	0.33
MoodHappy	0.45	0.67	0.65
MoodDisappointed	-0.45	-0.32	-0.31
MoodAngry	-0.24	-0.34	-0.12

The modification we made is mainly on **Step 3**, we believe that the representation of a target contextual situation should be personalized to select more relevant local dataset and generate more accurate recommendations. Therefore, instead of aggregating or concatenating the representation of contextual conditions to represent a target contextual situation, we represent a target contextual situation by directly using the contextual conditions that occur in this contextual situation, in the following of this paper we call this method Relevance. For example, the target contextual situation for a user Mike is $CS = (DaytypeWeekday = 1, CompanionFriends = 1, MoodHappy = 1)$ and we suppose that according to his characteristic information (e.g. sex, age) he is in cluster2. By referring to Figure 3, contextual condition “DaytypeWeekday” is represented by a vector $w_{DaytypeWeekday} = (0.18, -0.13, -0.54)$, contextual condition “CompanionFriends” is represented by a vector $w_{CompanionFriends} = (0.51, 0.45, 0.12)$, contextual condition “MoodHappy” is represented by a vector $w_{MoodHappy} = (0.45, 0.67, 0.65)$.

- If the Aggregation method is adopted, according to Equation 4, then the representation of this target contextual situation will be $w_{CS-Agg} = (\frac{0.18+0.51+0.45}{3}, \frac{-0.13+0.45+0.67}{3}, \frac{-0.54+0.12+0.65}{3}) = (0.38, 0.33, 0.08)$,
- If the Concatenation method is adopted then $w_{DaytypeWeekday}, w_{CompanionFriends}$ and $w_{MoodHappy}$ are simply concatenated to represent this target contextual situation, the result will be $w_{CS-Con} = (0.18, -0.13, -0.54, 0.51, 0.45, 0.12, 0.45, 0.67, 0.65)$
- If the Relevance method is adopted, the PCC between “DaytypeWeekday” and ratings for cluster2 is -0.13 ; the PCC between “CompanionFriends” and ratings for cluster2 is 0.45 ; the PCC between “MoodHappy” and ratings for cluster2 is 0.67 , then the representation of this target contextual situation will be $w_{CS-Rel} = (-0.13, 0.45, 0.67)$.

4 Experiments setup

In this section, we present the experiment set up, we will evaluate the Relevance, Aggregation and Concatenation methods on three real-world context-aware rec-

Table 3. Descriptions of datasets

Characteristics	CoMoDa	CarMusic	TijuanaRestaurant
#ratings	2296	4012	1422
#users	121	42	50
#items	1197	139	40
rating scale	1-5	1-5	1-5
sparsity	98.41	31.27	28.90
User cluster	5	0	0
Item cluster	4	10	0
Contextual factors	12	8	2

ommendation scenarios: movie, music and restaurant recommendations. We also compare the context-aware methods with the context-free methods to prove the benefits of integrating contextual information in recommender system.

4.1 Datasets

We utilize the following datasets for context-aware movie, restaurant and music recommendation. Note that we did not choose **STS** (a context-aware travel recommendation dataset by [8]) as [10] did. Because in **STS** dataset, the number of contextual conditions vary from one observation to another, when adopting the Relevance method, the local dataset contains fewer observations, therefore we obtain worse results.

- (1) **CoMoDa**, a well known context-aware movie recommendation dataset collected by [16]. According to the definition of [11], users’ contextual situation is composed of 12 contextual factors: Temporal context: “time”, “daytype”, “season”; Spatial context: “location”; Environmental context: “weather”; Social context: “social”; Psychophysiological context: “end emotion”, “dominant emotion”, “mood”, “physical”; Cognitive context: “decision”, “interaction”.
- (2) **CarMusic**, a context-aware music recommendation when driving car, collected by [5]. Following definition of [11], user’s contextual situation is composed of 8 context factors, Spatial context: “landscape”, “road type”, “road conditions”; Environmental context: “weather”; Psychophysiological: “mood”, “sleepiness”; Cognitive context: “driving style”. It should be noted that for each observation in the dataset, only one contextual factor is known.
- (3) **TijuanaRestaurant** by [18], each contextual situation is composed of 2 contextual factors, namely Temporal context: “time”; Social context: “social”.

Table 3 summarizes the main characteristics of the three datasets. In order that we can compare the proposed modification to the original methods proposed by [10], we followed the same pre-treatment of the three datasets to get the cluster information of users and items. The clustering method used by [10] was Hierarchical Clustering. The detailed information is shown in Table 3.

We have 0 cluster for users in **CarMusic**, and for users and items in **TijuanaRestaurant**, because corresponding information related to users and items is not available. The traditional 2D (context-free) method in **Step 5** will be Biased Matrix Factorization model [14], implemented through Surprise library [13]. This context-free method is also used by [10] in **Step 5**.

5 Results and discussion

We will first compare the proposed Relevance method and the Aggregation method, Concatenation method in terms of MAE (Equation 1) and RMSE (Equation 2) and model training time based on 5-fold cross-validation, then we will compare the best performing context-aware approach and context-free approach in terms of MAE and RMSE. The context-free baselines we chose are: Singular Value Decomposition (SVD) [14], Biased Singular Value Decomposition (BSVD) [14], Non-negative Matrix Factorization (NMF) [21], Slope One [17], K-nearest neighbors (KNN) [19] and Co-clustering [12]. We choose these context-free methods as baselines because they are already integrated in the Surprise library [13]. Table 4 presents the results of our experiments, we have the following precisions:

- UB-Agg (The same as CBPF-UB by [10]), UB-Con and UB-Rel refer to user-based CBPF model and Aggregation, Concatenation, Relevance method respectively as described in Section 3.
- CUB-Agg (The same as CBPF-CUB-AG by [10]), CUB-Con (The same as CBPF-CUB-CN by [10]) and CUB-Rel refer to the same model but the PCC was calculated with users clustered.
- IB-Agg (The same as CBPF-IB by [10]), IB-Con and IB-Rel refer to item-based CBPF model and Aggregation, Concatenation, Relevance method describe in Section 3.
- CIB-Agg (The same as CBPF-CIB-AG by [10]), CIB-Con (The same as CBPF-CIB-CN by [10]) and CIB-Rel refer to the same model but the PCC was calculated with items clustered.
- For the **CarMusic** and **TijuanaRestaurant**, we did not have the clustered version because corresponding information related to users and items is not available.

We have the following observations:

- We can find clustered approach outperforms non-clustered approach in terms of MAE and RMSE, which is in line with the results reported by [10]; the improvements are also observed in terms of model training time (This experiment was carried out in Dell Latitude 7310, Intel® Core™ i5-10310U CPU @ 1.70GHz × 8 and the operating system is Ubuntu 18.04.5 LTS). See rows(1,4), rows(2,5) and rows(3,6) in **CoMoDa** dataset for example, the same observations are made for item-based methods. In item based methods, clustered versions outperform non-clustered versions in terms of MAE,

Table 4. Results of experiments

Model\Dataset	Restaurant			CarMusic			CoMoDa		
	MAE	RMSE	Time(s)	MAE	RMSE	Time(s)	MAE	RMSE	Time(s)
1 UB-Agg	1.03	1.20	1704.09	1.26	1.47	1311.46	0.89	1.01	7841.22
2 UB-Con	0.97	1.16	1290.34	1.25	1.45	1210.85	0.84	0.97	13725.96
3 UB-Rel	0.76	1.03	1009.26	1.30	1.48	616.64	0.79	0.87	1930.65
4 CUB-Agg	-	-	-	-	-	-	0.82	1.01	5713.87
5 CUB-Con	-	-	-	-	-	-	0.76	0.82	1998.93
6 CUB-Rel	-	-	-	-	-	-	0.70	0.82	1464.82
7 IB-Agg	0.89	1.12	1918.80	1.30	1.47	1414.14	0.83	0.90	24490.18
8 IB-Con	0.87	1.10	1967.53	1.27	1.45	1389.16	0.77	0.93	31150.52
9 IB-Rel	0.87	1.09	1449.08	1.27	1.42	721.12	0.72	0.85	2689.49
10 CIB-Agg	-	-	-	0.90	1.07	1007.12	0.74	0.90	2885.63
11 CIB-Con	-	-	-	0.79	0.91	918.49	0.73	0.88	2187.28
12 CIB-Rel	-	-	-	0.82	1.16	588.81	0.70	0.80	2164.82

RMSE and training time. When original data is clustered, the calculation of Pearson Correlation Coefficient (PCC) [7] is more precise since more data will be available for each cluster and the size of the vector for representing contextual situations is reduced, resulting in better results and less training time. It should be noted that, the time recorded here is the time for pre-training not the time required for a recommendation request. A recommendation request can be processed within a few seconds.

- Improvements (in terms of MAE and RMSE) of Relevance method over the Aggregation and Concatenation method in **CoMoDa** and **TijuanaRestaurant**. See rows(1,2,3) and rows(4,5,6); rows(7,8,9) and rows(10,11,12). When adopting Aggregation method, the influence of each contextual condition could be neutralized, for example, when aggregating (0.3, 0.5, 0.4) and (-0.33, -0.41, -0.28), the result will be (-0.01, 0.03, 0.04); when adopting Concatenation method, since the vector contains information of other user clusters, some noise information is also integrated; in Relevance method, we select only the contextual conditions that appear in the target contextual situation to represent it, therefore, the representation of target contextual situation is more relevant and can better model the target contextual situation, as a result, the local dataset selected can better model the target user’s preference in that contextual situation, resulting in better results.
- Improvements (in terms of MAE and RMSE) of Relevance method are not observed in **CarMusic** dataset. The potential reason could be that the contextual information in this dataset is too little. For the **CarMusic** dataset, only one contextual condition is known for every observation in the dataset, which can not well represent the target contextual situation. As a result, the predictions are not accurate, resulting in higher MAE and RMSE. Some of the pre-filtering methods that consider contextual information even perform worse than the context-free methods. The results of context-free methods are presented in Table 5

Table 5. Comparison with context-free approaches

Model\Dataset	Restaurant		CarMusic		CoMoDa	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
CBPF	0.76	1.03	0.79	0.91	0.70	0.80
SVD	0.95	1.25	0.88	1.18	2.05	2.49
BSVD	0.80	1.12	0.85	1.08	0.83	1.03
NMF	0.82	1.16	0.81	1.12	0.92	1.16
KNN	0.90	1.19	1.04	1.34	0.86	1.14
SlopeOne	0.79	1.13	1.04	1.35	0.87	1.13
CoClustering	0.77	1.11	1.07	1.37	0.86	1.12

- The time consumed for model training can be reduced by adopting the Relevance method, see the column Times. Suppose there are x user clusters, each contextual situation is composed of q contextual factors (q contextual conditions). When adopting Aggregation method, the size of vector required for representing target contextual situation will be x , but it requires calculating the mean of q vectors of size x ; when adopting Concatenation method the size required for representing target contextual situation will be $q \times x$; when adopting Relevance method the size required for representing target contextual situation will be q . This shows that Relevance method requires less size for representing the target contextual situation, there is no need for calculating the mean of vectors, therefore the time required for training is reduced. We note that, when adopting the Concatenation and Relevance method, the number of contextual factors in each situation should be equal. If not, the size of vectors representing contextual situations is the same, making it impossible to calculate the similarities between them. However, Aggregation does not have this limit.

We then select the best performing context-aware approaches listed in Table 4 to compare with the baselines, the results are presented in Table 5. Note that CBPF refers to the best performing approaches in Table 4. For **Restaurant** dataset, CBPF refers to UB-Rel; for **CarMusic** dataset, CBPF refers to CIB-Con; for **CoMoDa** dataset, CBPF refers to CIB-Rel. It is clear that CBPF outperforms context-free approaches, indicating that integrating contextual information can improve the performance of recommender systems and generate more accurate recommendations, this is in line with the experiment results reported by [10].

6 Conclusions and perspective

In this paper we propose a new method for representing contextual situations in Context-aware recommender systems (CARSs). The contributions are:

- We revisit the correlation-based pre-filtering methods proposed by [10] and prove its effectiveness compared with context-free methods.

- We propose a new method for representing target contextual situations and showed that it is more effective and more efficient than the approach proposed by [10] when more contextual information is given. The selection of representation method of contextual situations depends on the application domain, the availability of contextual information.

We point out avenues for future work:

- In this paper we cluster the users/items based on their characteristics, we plan to cluster them according to ratings, which could be another way to discover similar users/items in case that the characteristics of users/items are not available.
- Contextual conditions in this work are defined as binary variables, they could probably be extended to fuzzy number to handle the situations of partial appearance. For example, “mid-day” is partially “morning” and “afternoon”.
- Considering the expressiveness of ontology, we plan to explore the possibility to combine ontology-based method and our data-driven method.
- We also plan to explore the possibility profiling approaches to get more personalised representation of contextual situations for each user.
- State-of-art datasets for CARSs research are not perfect, few of them contain explicit and complete contextual information, which has limited the research in CARSs community, we plan to carry out user studies to collect usable dataset that contain more complete contextual information for CARSs research.
- CARSs can better model users’ preferences in different contextual situations, we plan to explore the possibilities of generating context-aware explanations. For example, an explanation “We recommend you this romantic movie because you are with your lover” would be more convincing than an explanation such as “We recommend this movie to you because it is similar to movies you watched before”.

References

1. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, P.: Towards a better understanding of context and context-awareness. In: International symposium on handheld and ubiquitous computing. pp. 304–307. Springer (1999)
2. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)* **23**(1), 103–145 (2005)
3. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering* **17**(6), 734–749 (2005)
4. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: *Recommender systems handbook*, pp. 217–253. Springer (2011)
5. Baltrunas, L., Kaminskas, M., Ludwig, B., Moling, O., Ricci, F., Aydin, A., Lüke, K.H., Schwaiger, R.: Incarmusic: Context-aware music recommendations in a car. In: *International conference on electronic commerce and web technologies*. pp. 89–100. Springer (2011)

6. Baltrunas, L., Ludwig, B., Ricci, F.: Matrix factorization techniques for context aware recommendation. In: Proceedings of the fifth ACM conference on Recommender systems. pp. 301–304 (2011)
7. Benesty, J., Chen, J., Huang, Y., Cohen, I.: Pearson correlation coefficient. In: Noise reduction in speech processing, pp. 1–4. Springer (2009)
8. Braunhofer, M., Elahi, M., Ricci, F., Schievenin, T.: Context-aware points of interest suggestion with dynamic weather data management. In: Information and communication technologies in tourism 2014, pp. 87–100. Springer (2013)
9. Codina, V., Ricci, F., Ceccaroni, L.: Distributional semantic pre-filtering in context-aware recommender systems. *User Modeling and User-Adapted Interaction* **26**(1), 1–32 (2016)
10. Ferdousi, Z.V., Colazzo, D., Negre, E.: Cbpf: leveraging context and content information for better recommendations. In: International Conference on Advanced Data Mining and Applications. pp. 381–391. Springer (2018)
11. Ferdousi, Z.V., Negre, E., Colazzo, D.: Context factors in context-aware recommender systems. In: AISR 2017: Atelier interdisciplinaire sur les systèmes de recommandation (2017)
12. George, T., Merugu, S.: A scalable collaborative filtering framework based on co-clustering. In: Fifth IEEE International Conference on Data Mining (ICDM'05). pp. 4–pp. IEEE (2005)
13. Hug, N.: Surprise: A python library for recommender systems. *Journal of Open Source Software* **5**(52), 2174 (2020)
14. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* **42**(8), 30–37 (2009)
15. Korpipää, P., Mäntyjärvi, J.: An ontology for mobile device sensor-based context awareness. In: International and Interdisciplinary Conference on Modeling and Using Context. pp. 451–458. Springer (2003)
16. Košir, A., Odic, A., Kunaver, M., Tkalcic, M., Tasic, J.F.: Database for contextual personalization. *Elektrotehniški vestnik* **78**(5), 270–274 (2011)
17. Lemire, D., Maclachlan, A.: Slope one predictors for online rating-based collaborative filtering. In: Proceedings of the 2005 SIAM International Conference on Data Mining. pp. 471–475. SIAM (2005)
18. Ramirez-Garcia, X., Garcia-Valdez, M.: Post-filtering for a restaurant context-aware recommender system. In: Recent Advances on Hybrid Approaches for Designing Intelligent Systems. *Studies in Computational Intelligence*, vol. 547, pp. 695–707. Springer International Publishing (2014)
19. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: The adaptive web, pp. 291–324. Springer (2007)
20. Strang, T., Linnhoff-Popien, C.: A context modeling survey. In: Workshop Proceedings (2004)
21. Zhang, S., Wang, W., Ford, J., Makedon, F.: Learning from incomplete ratings using non-negative matrix factorization. In: Proceedings of the 2006 SIAM international conference on data mining. pp. 549–553. SIAM (2006)
22. Zheng, Y., Burke, R., Mobasher, B.: Optimal feature selection for context-aware recommendation using differential relaxation. *Acm Recsys* **12** (2012)