Proceedings of the Federated Conference on Computer Science and Information Systems pp. 19–28
DOI: 10.15439/2017F258
ISSN 2300-5963 ACSIS, Vol. 11

Top k Recommendations using Contextual Conditional Preferences Model

Aleksandra Karpus*, Tommaso di Noia†, Krzysztof Goczyła*
* Faculty of Electronics Telecommunication and Informatics,
Gdańsk University of Technology
ul. G. Narutowicza 11/12, 80-233 Gdańsk, Poland
Email: {aleksandra.karpus, krzysztof.goczyla}@eti.pg.edu.pl
† Electrical & Information Engineering Department,
Polytechnic University of Bari,
via E. Orabona 4, Bari, Italy
Email: tommaso.dinoia@poliba.it

***Abstract*—Recommender systems are software tools and techniques which aim at suggesting to users items they might be interested in. Context-aware recommender systems are a particular category of recommender systems which exploit contextual information to provide more adequate recommendations. However, recommendation engines still suffer from the cold-start problem, namely where not enough information about users and their ratings is available. In this paper we introduce a method for generating a list of top *k* recommendations in a new user cold-start situations. It is based on a user model called Contextual Conditional Preferences and utilizes a *satisfiability* measure proposed in this paper. We analyze accuracy measures as well as serendipity, novelty and diversity of results obtained using three context-aware publicly available datasets in comparison with several contextual and traditional state-of-the-art baselines. We show that our method is applicable in the new user cold-start situations as well as in typical scenarios.**

I. Introduction

RECOMMENDER systems are software tools and techniques which aim at suggesting to users items they might be interested in. Context-aware recommender systems are a particular category of recommender systems which exploit contextual information to provide more adequate recommendations. For example, a restaurant recommendation for a Saturday evening with your friends should be different from one suggested for a workday lunch with co-workers [1].

We distinguish three forms of context-aware recommendation processes: a *contextual pre-filtering*, a *contextual post-filtering* and a *contextual modeling* [2]. *Pre-filtering* approaches use a current context to select a relevant subset of data on which a recommendation algorithm is applied. *Post-filtering* approaches exploit a contextual information to select only relevant recommendations returned by some algorithm. *Contextual modeling* differs from other techniques as it incorporates a context into a recommendation algorithm.

During last decades many context-aware approaches were proposed. But usually they have considered a situation where a lot of data is available. On the other hand, a recommender systems research still strives for solving the cold-start problem, namely where we have not enough information about users and their ratings. For example, matrix factorization methods do not work well in the cold start scenarios [3].

Different situations described in the literature are called a cold-start problem. Two of them are well-known and have also another names, respectively: a new item and a new user cold-start problem. Both occur when a recommender system is well-established and a lot of ratings are available. When we introduce a new item into such system, in many recommendation algorithms it will not be recommended to users, because of the lack of its history, i.e. user ratings. The same happens when a new user registers into the recommender system. He will not receive interesting recommendations just because the system does not know his preferences yet [4].

In this paper we introduce a method for generating a list of top k recommendations in a new user cold-start situations. It is based on a user model called contextual conditional preferences [5] which represents user interests in items in a compact way. We run our experiments on a context-aware datasets publicly available in the Web, i.e. `LDOS-CoMoDa` dataset [6], `Unibz-STS` [7] and `Restaurant & consumer data` [8]. We confirmed that our method is applicable in the new user cold-start situations as well as in typical scenarios.

The main contributions of this paper are:

- a new measure of *satisfiability* to describe how much an item satisfies a contextual conditional preference,
- an algorithm for context-aware reshuffling of items in the recommendations list using contextual conditional preferences.

The advantages of the method are: (I) a possibility to combine it with existing algorithms for a ranking task, and (II) the ability to work well in a typical scenario and a new user cold-start scenario.

The remainder of the paper is constructed as follows. Related work is presented in Section II. Section III briefly introduces contextual conditional preferences and describes our method for generating a list of top k recommendations. In Section IV the datasets are described. Algorithms and measures used for the evaluation are presented in Section

V. Section VI provides our evaluation approach and obtained results. Conclusions close the paper.

II. Related work

The idea of modeling user interests with a preference relation is not new. In [9] a formalism of CP-nets was proposed. CP-nets are intuitive graphical models for representing conditional preferences under the *ceteris paribus* ("all else being equal") assumption. Preferences presented in this paper always contain "conditional part" which consists of contextual parameters only. Another difference is the lack of the *ceteris paribus* assumption.

In [10] constraint-based recommender systems were described. Since users define their preferences in the form of requirements for a product, they mainly focus on solving the constraints satisfaction problem while recommending new items. Additionally, authors proposed an algorithm that ranks the recommended items according to their degree of a constraint fulfillment. Our approach is slightly similar to this method, because the contextual conditional preferences could be seen as constraints. Furthermore, we also rank and reshuffle items in the primary recommendation list according to a level of a satisfaction of the CCPs. Nevertheless, we focus mostly on the context-awareness and learn the CCPs from a users history.

Contextual preferences were described in [11] as database preferences annotated with a contextual information, where contextual parameters take values from hierarchical domains, allowing different levels of abstraction. While using CCPs, a generalization of contextual variables is not possible.

Context-Aware Recommender Systems is a well-established research area and many recommendation techniques were already proposed. A multi-agent system for making context and intention-aware recommendations of Points of Interest (POI) was presented in [12]. The tasks of collecting an information about POIs and storing a users' profiles data were divided into two kinds of agents. The user's Personal Assistant Agent is responsible for receiving queries, storing user data, computing recommendations and updating user preferences according to his feedback. Authors incorporated not only the context but also a user's goal in visit the POI. Besides a context-awareness, this approach and ours are completely different.

An interesting approach for a context-awareness was proposed in [13]. Authors introduced *micro profiles* which split a user profile into partitions depending on the values of context parameters. They showed that usage of such *micro profiles* gives a significant improvement in the prediction accuracy in the movie domain while considering time as a context variable. CCPs could be seen as a kind of *micro profiling*, because each preference statement consists of user interests and a context in which it is true.

In [14] a new context-aware music recommender system was presented. As a main recommendation technique authors used case-base reasoning (CBR). CBR systems store knowledge in the case base in the form of `cases`. During a recommendation task, the cases are compared to the current case according to some similarity measure. In the paper, 2-step case-based reasoning was used. Firstly, to determine similar context, and then to find similar users to make predictions. Contextual conditional preferences could be seen as cases, but in fact they are something different. We chose active preferences according to a similarity measure so we could position our work in the CBR research area. However, we do not have iterations or a relevance verification in the recommendation process.

One of the possibilities for contextual pre-filtering are Context-Aware Splitting Approaches (CASA). We could distinguish three kinds of them, i.e. item splitting, user splitting and UI splitting which combines the first two [15]. For the item splitting, we split the item into two items depending on the contextual factor and its value assuming that the user's ratings are significantly different. Analogously, we could split user into two users based on the contextual condition. The UI splitting uses both kinds of splits, for items and for users. It should be notice that the best contextual factor for splitting users and items could be, and usually is, different, i.e. we do not use the same contextual condition to split users and items. The only two similarities between our post-filtering method and this approach are incorporating contextual information into a recommendation process and dependance from other existing non-contextual algorithms, i.e. both methods cannot be used alone.

A context-aware extension of the SLIM algorithm, contextual SLIM (CSLIM), was introduced in [16]. Authors used a binary vector to denote a contextual situation, i.e. context parameters and their corresponding values. They followed the idea of an aggregation of users' ratings on other items, and add contextual factors into this aggregation. In the case when no other items were ranked in a certain context, the rating is estimating based on user's non-contextual ratings on this item. Authors showed that the method outperforms the basic SLIM algorithm as well as context-aware matrix factorization methods. This algorithm differs from our method as it incorporates context in the recommendation phase. Thus, it could be classify as a context modeling method. In contrast, our method is positioned as a post-filtering technique.

An interesting approach was introduced in [17]. Authors presented a context-aware system for events recommendation that addresses the new item cold-start scenario. They identified many contextual signals and models, and used them as features for learning to rank events.

A hybrid matrix factorization model for the cold start problem was presented in [3]. It was shown to work well with the cold and warm start scenarios. Similarly to our work, author used both, user and item information.

III. Generating Top k recommendations

The proposed method can be classified as a post-filtering technique. We rely on existing non-contextual algorithms to generate a primary recommendations list which we then reshuffle as described in Section III-C. For this purpose we use

MOST WIEDZY Downloaded from mostwiedzy.pl

the Contextual Conditional Preferences Model whose details are presented in Section III-A.

A. Contextual Conditional Preferences Model

Contextual conditional preferences (CCPs) introduced in [5], [18] are a compact representation of user interests in items in different situations. This model describes relations between a context related to a user's ratings and an item content, and consists of a set of conditional preferences.

We define the Contextual Conditional Preference (CCP) as an expression of the form:
$(\gamma_1 = c_1) \wedge \ldots \wedge (\gamma_n = c_n) \mid (\alpha_1 = a_1) \succ (\alpha_1 = a'_1) \wedge \ldots \wedge (\alpha_m = a_m) \succ (\alpha_m = a'_m)$
with γ_i being contextual variables and α_i item attributes, and $c_1, ..., c_n,\ a_1, a'_1, ..., a_m, a'_m$ being concrete values of these parameters.

The above preference is read as *given the context* $(\gamma_1 = c_1) \wedge \ldots \wedge (\gamma_n = c_n)$ *I prefer* a_1 *over* a'_1 *for* α_1 *and* a_m *over* a'_m *for* α_m. An example of the CCP for the `Unibz-STS` dataset is shown below.

$$\begin{aligned} &\text{weather} = sunny \wedge \text{companion} = with\ children \\ &\mid\ \text{category} \in \{walk\ and\ trail, park\} \\ &\succ \text{category} \in \{museum\} \end{aligned}$$

It means that for a given context (i.e. a sunny weather and a companion of the children) a user prefers POIs with categories like "walk and trail" and "park" to those with category "museum".

We distinguish two types of CCPs: individual and general. An individual CCP (ICCP) represents preferences of a single user, while a general CCP (GCCP) catches a general trend of interests for all users in a certain contextual situation, i.e. we treat ratings from all users like they were made by one person. The GCCPs are very important for this work, since we are unable to learn ICCPs for new users (they do not have any rating history yet).

During our experiments we automatically generated CCPs. Details are described in the next section.

B. Contextual Conditional Preferences Extraction

An algorithm of a preferences extraction was originally published in [5].

In order to elicit preference relations we split the dataset into two parts based on the value of the ratings. Depending on a rating scale for a dataset we use a different threshold to divide ratings into positive and negative ones. Then, both datasets are divided into smaller sets containing all of the contextual information and one of the movie features. With such prepared data we computed context-aware individual preferences for each user by running the `Prism` algorithm[19] from the WEKA library[1] (version 3.6.11) to generate rules of the form

$$(\gamma_1 = c_1) \wedge \ldots \wedge (\gamma_n = c_n) \mid (\alpha_1 = a_1) \succ (\alpha_1 = a'_1).$$

[1] `http://www.cs.waikato.ac.nz/ml/weka/`

Then we compacted preferences with the same "conditional part" into one preference of the form shown below.

$$\begin{aligned} &season = 3 \wedge weather = 1 \wedge time = 2 \wedge mood = 1 \\ &\mid\ genre \in \{18\} \succ genre \in \{8, 12, 7\} \\ &\wedge\ director \in \{5, 8\} \succ director \in \{3\}\,. \end{aligned}$$

It means that for a given context (e.g. season is 3 - Autumn) a user prefers a genre with id 18 to those with 8, 12 or 7 and directors from clusters 5 and 8 to those from cluster 3 etc.

If the value of some content parameter was the same on both sides of a preference relation for some certain user's context, then this value was marked as meaningless and not taken into consideration in this context for the user.

The main difference in the computation of general and individual preferences is that in the first case all the ratings from the dataset were treated like they were made by one person. As a consequence, we removed many contradictory values during the merging phase. To better understand the issue, let us consider an example in the movie domain from Tab. I. Besides information about rating for an item, we have also two contextual factors, i.e. *companion* and *day*, and one movie feature, i.e. *genre* in sample user profiles. For all three users we could compute ICCPs. We obtained following individual preferences for Alice:

$$\begin{aligned} &\text{companion} = family \wedge \text{day} = Sunday \\ &\mid\ \text{genre} \in \{animated\} \succ \text{genre} \in \{superhero\}\ , \end{aligned}$$

$$\begin{aligned} &\text{companion} = friend \\ &\mid\ \text{genre} \in \{thriller\} \succ \text{genre} \in \{drama\}\ , \end{aligned}$$

$$\begin{aligned} &\text{day} = Saturday \\ &\mid\ \text{genre} \in \{fantasy\} \\ &\succ \text{genre} \in \{drama, supernatural\}\ . \end{aligned}$$

We could observe that Alice's movie preferences vary depending on the company and day. The same applies for Bob and Carol. General preferences (GCCPs) computed for sample profiles are shown below.

$$\begin{aligned} &\text{companion} = alone \\ &\mid\ \text{genre} \in \{fantasy\} \succ \text{genre} \in \{sciencefiction\}\ , \end{aligned}$$

$$\begin{aligned} &\text{companion} = friend \\ &\mid\ \text{genre} \in \{fantasy\} \succ \text{genre} \in \{drama\}\ , \end{aligned}$$

$$\begin{aligned} &\text{day} = Saturday \\ &\mid\ \text{genre} \in \{fantasy\} \succ \text{genre} \in \{drama\}\ . \end{aligned}$$

C. Reshuffling of recommendations list

An algorithm is presented in Algorithm 1. We describe it and refer to its concrete lines below.

We assume that ICCPs and GCCPs are generated for all non-new users, since new users do not have any rating history.

For a certain user and his current context, first we generate a primary list of top 100 recommendations with some existing non-context-aware algorithm, e.g. *User k Nearest Neighbors* (User kNN) [20] (line 1). Then we have to find the best CCPs that will be further used in the reshuffling process (line 2).

MOST WIEDZY Downloaded from mostwiedzy.pl

TABLE I
SAMPLE USER PROFILES OF ALICE, BOB AND CAROL.

User	Item (Movie)	Rating	Companion	Day	Genre
Alice	Donnie Darko	1	friend	Saturday	drama, supernatural
Alice	Girl Interrupted	2	friend	Friday	drama
Alice	How To Hook Up Your Home Theater	4	family	Sunday	animated
Alice	Inception	5	friend	Friday	heist, thriller, science fiction
Alice	The Imaginarium of Doctor Parnassus	5	friend	Saturday	fantasy
Alice	Shrek	5	family	Saturday	animated, fantasy
Alice	Spiderman	1	family	Sunday	superhero
Alice	The Counselor	4	friend	Friday	thriller
Alice	The Lion King	4	family	Sunday	animated, adventure
Bob	An Unexpected Journey	5	alone	Saturday	fantasy, epic, adventure
Bob	City Of Angels	2	girlfriend	Saturday	fantasy, romantic, drama
Bob	Armageddon	2	alone	Friday	thriller, disaster, science fiction
Bob	Inception	1	alone	Tuesday	heist, thriller, science fiction
Bob	Green Mile	5	alone	Saturday	drama, fantasy
Bob	Hunger Games	2	alone	Saturday	science fiction, adventure
Bob	Tourist	4	girlfriend	Friday	thriller, comedy, romantic
Bob	Sleepless In Seattle	4	girlfriend	Friday	drama, comedy, romantic
Bob	The Desolation Of Smaug	5	alone	Tuesday	adventure, epic, fantasy
Carol	At Worlds End	5	friend	Friday	fantasy, swashbuckler
Carol	Dead Mans Chest	5	friend	Friday	fantasy, swashbuckler
Carol	Gangs Of New York	2	friend	Saturday	historical, drama, epic
Carol	The Imaginarium of Doctor Parnassus	5	friend	Saturday	fantasy
Carol	Return Of The King	5	alone	Saturday	epic, fantasy
Carol	The Curse Of The Black Pearl	5	friend	Friday	swashbuckler, fantasy
Carol	The Fellowship Of The Ring	5	alone	Saturday	epic, fantasy
Carol	Two Towers Film	5	alone	Tuesday	epic, fantasy
Carol	Cast Away	2	alone	Saturday	drama, adventure

In this case, the best preferences are those which are the most similar to the considered context. In order to count a contextual similarity between a CCP p and a current user context $ctx(u)$ we used the following metric:

$$sim\,(p, ctx(u)) = \sum_{(\gamma_i, c_i) \in p} overlap(ctx(u), (\gamma_i, c_i)),$$

$$overlap(ctx(u), (\gamma_i, c_i)) = \begin{cases} 1 & (\gamma_i, c_i) \in ctx(u); \\ 0.5 & c_i = -1; \\ 0 & otherwise. \end{cases}$$

The overlap function returns 1 when the pair (γ_i, c_i) is contained in both: the contextual part of p and in the current user context $ctx(u)$. When the pair (γ_i, c_i) is not contained in neither or only in one set of pairs, 0 is returned. When it is uncertain, i.e. when the value c_i for the dimension γ_i is equal to -1 (the unknown value), 0.5 is returned. Please note that the current user context $ctx(u)$ is also a set of pairs (γ'_i, c'_i), i.e. the name of the contextual variable and its value.

For each item in the primary recommendations list and each best CCP we compute *satisfiability* (line 7), namely how much an item i satisfies a CCP p:

$$sat(i, p) = \frac{\sum_{\alpha \in a(p)} \left(sim(v^m_\alpha(p), v_\alpha(i)) - sim(v^l_\alpha(p), v_\alpha(i))\right)}{|a(p)|},$$

where sim denotes Jaccard similarity, α is the name of an item feature, $a(p)$ is the set of item attributes considered in the CCP p, $v_\alpha(i)$ is the set of values of an attribute α for an item i. Similarly $v^m_\alpha(p)$ and $v^l_\alpha(p)$ denotes the sets of values of an attribute α for a CCP p on both sides of the preference relation - m stands for *more preferred* and l for *less preferred*.

The *satisfiability* measure represents the difference between item similarities to the both sides of the CCP's preference relation, i.e. the similarity to most preferred part minus the similarity of the less preferred part. In this way we reward items that fit the best to user preferences and penalize items that have features that user does not like, e.g. horror movies. The size of a set of item attributes serves as a normalization factor. Thus, disregarding to the number of item features, the value of *satisfiability* is always between 0 and 1.

The next step is to order the primary recommendations list according to the value of average *satisfiability* of the best CCPs (line 13). The last part is to cut off unneeded items from resulting recommendations list to receive top 5, top 10 or other top *k* ranking (line 14).

Let us consider again an example from Tab. I. We assume that some traditional recommendation algorithm returned a following top 10 list for Alice:

Gangs Of New York, The Curse Of The Black Pearl, Cast Away, An Unexpected Journey, City Of Angels, Armageddon, Green Mile, Hunger Games, Tourist, Sleepless in Seattle.

We consider a situation when Alice wants to watch a movie with a friend. With our reshuffling method, using two rules (ICCP for Alice profile and GCCP) for this contexts, we obtained the final top 5 recommendations list:

An Unexpected Journey, Armageddon, Tourist, The Curse Of The Black Pearl, City Of Angels.

Fantasy and *thriller* movies are higher in the final list, while *drama* movies were mostly cut off the list as expected from the user preferences. At this point, we will not evaluate results of this example. A comprehensive evaluation of the algorithm

MOST WIEDZY Downloaded from mostwiedzy.pl

Algorithm 1 Generating the list of top k recommendations with CCPs

Require: alg - a name of a baseline algorithm,
k - a number of recommendations in the final list,
u - a user,
ctx - a user context,
$ccps$ - a list of all CCPs for user u
Ensure: $topK$ - an ordered list of top k recommendations
$list \leftarrow$ generateTop100Recommendations(alg, u);
$best \leftarrow$ findBestCCPs$(ccps, u, ctx)$;
$map \leftarrow$ empty HashMap;
for all $item$ in $list$ **do**
  $sum \leftarrow 0$;
  for all ccp in $best$ **do**
    $sat \leftarrow$ satisfiability$(item, ccp)$;
    $sum \leftarrow sum + sat$;
  end for
  $avg \leftarrow sum/$sizeOf$(best)$;
  $map[item] \leftarrow avg$;
end for
$rec \leftarrow$ order(map);
$topK \leftarrow$ cutOff(rec, k);

TABLE II
BASIC STATISTICS OF THREE DATASETS: LDOS-COMODA (COMODA), UNIBZ-STS (STS) AND RESTAURANT & CONSUMER (R&C).

	CoMoDa	STS	R&C
Number of users	121	325	138
Number of items	1232	249	130
Number of ratings	2296	2534	1161
Max number of ratings per user	275	175	18
Min number of ratings per user	1	1	3
Avg number of ratings per user	18.98	7.80	8.41
Max number of ratings per item	26	282	36
Min number of ratings per item	1	1	3
Avg number of ratings per item	1.86	10.18	8.93

is presented in Section VI.

IV. DATASETS

We performed our experiments with three datasets, i.e. the LDOS-CoMoDa[2] dataset (LDOS), the Unibz-STS dataset (STS) and the Restaurant & consumer dataset[3] (RC). Basic statistics of the datasets are presented in Tab. II.

The LDOS-CoMoDa [6] contains user interaction with the system, i.e. the rating on a 5-star scale, the basic users' information, the content information about multiple item dimensions and twelve additional contextual information about the situation when the user consumed the item. According to [21] the choice of contextual variables to be used is crucial because of a different amount of information they gain. To eliminate irrelevant variables we computed correlation coefficients between context related attributes. We found only two of them to be strongly correlated, i.e. *city* and *country*, which was known before the computation. Thus, we could conclude that none of the other contextual factors are correlated.

In [21] six variables in the LDOS-CoMoDa were identified as informative. Since we focus on the cold start problem in this paper, we want to limit the sparsity of the data as much as possible. Therefore, we chose two of six most informative contextual variables, i.e. *dominant emotion* and *end emotion*, to use in our further work presented in this paper. Since we also focus on general trends, we will use *age* parameter which we categorized into 5 groups.

The Unibz-STS [7] dataset was collected by a mobile application that recommends places of interests (POIs) in South Tyrol in Italy. The recommender is called South Tyrol Suggests (STS). The dataset contains ratings on a 5-star scale, an information about a users' personality (e.g. *extraversion*, *emotional stability*), a context of visiting a POI (e.g. *weather*, *season*, *companion*) and a POI's category.

The Restaurant & consumer data [8] consists of three types of information: a restaurant data (e.g. *cuisine*, *smoking*, *dress*), a user information (e.g. *smoker*, *dress preference*, *transport*) and a rating that a user gave to a restaurant. In this dataset ratings are expressed on a 0-2 scale. Contextual parameters such as an information about a user's mood or companion are not available.

V. ALGORITHMS AND MEASURES

We had to choose some existing recommendation techniques to evaluate our approach since it is designed to work with any of baseline algorithms that generate a list of top k recommendations. We used six algorithms from the LibRec[4] library [22] that are appropriate for the ranking task, i.e. User kNN, BPR[23], FISM[24], Latent Dirichlet Allocation (LDA)[25], SLIM [26] and WRMF [27], [28] to be used in both scenarios.

To compare our work with other context-aware state-of-the-art algorithms, we chose two methods, i.e. Contextual SLIM (CSLIM) [16] and UI Splitting [15], and used their implementations from the CARSKit[5] library [29]. Since the UI Splitting approach is a pre-filtering technique, it needs to be combined with other existing algorithms. From the methods proposed in the CARSKit library, we chose those that overlap with the algorithms that we already used with our method, i.e. BPR, SLIM and User kNN.

To evaluate our method we use several measures for the ranking task available in the LibRec library, i.e. *mean average precision* (MAP), *mean reciprocal rank* (MRR), *normalized discounted cumulative gain* (nDCG) and the classical information retrieval measures: *precision* and *recall*. The latter two were computed on the top 10 recommendations list. We have also implemented four additional measures. The first one is a *diversity* measure proposed by [30], i.e. Intra-List Diversity

[2]The data is available at http://212.235.187.145/spletnastran/raziskave/um/comoda/comoda.php.

[3]The data sets are available at https://github.com/irecsys/CARSKit/tree/master/context-aware_data_sets.

[4]http://www.librec.net/

[5]https://github.com/irecsys/CARSKit/

MOST WIEDZY Downloaded from mostwiedzy.pl

(ILD) that computes the average distance between each couple of items in the list R:

$$ILD(R) = \frac{1}{|R|\,(|R|-1)} \sum_{i,j \in R, i \neq j} (1 - sim(i,j)), \quad (1)$$

where i, j are items. The sim function is configurable and application dependent. In our work, we used Jaccard similarity as a similarity measure for all item attributes.

We wanted to compute the serendipity value for obtained recommendations lists. But the problem is that there is no one common serendipity measure. Thus, we decided to implement two measures, i.e. a simple metric presented in [31] and given by a formula (2) that we called *expectedness* and *unserendipity* proposed by Zhang et al. [32] and given by a formula (3).

$$expectedness = \frac{1}{k} \sum_{i=1}^{k} pop(i), \quad (2)$$

where k is the size of the recommendations list, i denotes an item and $pop(i)$ is a popularity of an item i.

$$unserendipity = \frac{1}{|H_u|} \sum_{h \in H_u} \frac{1}{k} \sum_{i \in R_{u,k}} sim(i,h), \quad (3)$$

where u denotes a user, h is an item from a user history H_u, k is the size of a user u recommendation list $R_{u,k}$ and i denotes an item from a recommendations list $R_{u,k}$. The sim function used by Zhang et al. [32] was a cosine similarity. However, in our work we used the Jaccard similarity as with the previous measures.

Expectedness is a simple measure which sums up the popularity of all items in the recommendations list. The unserendipity measure is more complicated and checks how much items from a recommendations list are similar to those from a user history. Both measures are in opposite to the definition of serendipity. Thus, the lower values of those measures are, the better the serendipity of a recommendations list is.

The last measure is *novelty* [33] which expresses how much items from the list are unknown for a user. It is given by a formula:

$$novelty = \frac{1}{k} \sum_{i \in R_{u,k}} log_2(pop(i)). \quad (4)$$

Similarly to the formulas presented above, u denotes a user, k is the size of a recommendations list $R_{u,k}$, i denotes an item and $pop(i)$ is its popularity.

All of the four measures above were computed on the top 10 recommendations list.

In recommender systems, we provide a list of top k recommendations for each user. However, in the context-aware recommender systems we need to incorporate a context also into an evaluation. Thus, we generate the top k list for each pairs of a user and his context. The resulting measures values are usually much smaller than the ones in traditional recommender systems, because it is not very common for users to rate multiple items within a same context. This type of evaluation has been used in prior research [15], [16].

VI. Experiments and results

We performed two experiments on three datasets described in Section IV. The first, to simulate the new user cold-start situation. The second, to check if our method works also in a typical scenario.

To simulate two different scenarios we prepared two separate splits of each dataset into training and test sets for hold out validation. The following procedures were applied on each datasets.

To be able to check if the method is applicable for a new user cold-start scenario, we randomly chose 20% of users and put all of their ratings in the test set. Remaining ratings were used as a training set. With this construction of the training and test sets, we were unable to generate ICCPs for the test users (we do not have any rating of those users in the training set). Thus, we used GCCPs only. The results obtained with these splits are presented in the Tables III, IV and V, for `LDOS-CoMoDa`, `Unibiz-STS` and `Restaurant & Customer` datasets respectively. Because the *unserendipity* measures a similarity with a user profile and we have only new users in these splits, we omitted it in the tables. In all of the following tables a prefix *ctx-* denotes that the list obtained by the algorithm was reshuffled with our method.

The second splits were to test a typical situation. Thus we randomly chose 20% of each user's ratings and put them in the second test sets, while remaining users ratings were placed in the second training sets. The results obtained with these splits are presented in the Tables VI, VII and VIII, for `LDOS-CoMoDa`, `Unibiz-STS` and `Restaurant & Customer` datasets respectively. A prefix *ctx-* denotes that the list obtained by the algorithm was reshuffled with our method.

It should be notice, that we did not consider the new item problem during the splits. Therefore, all test sets contain some number of items which do not appear in the corresponding training sets.

It has been shown that the most informative contextual variables in the `LDOS-CoMoDa` dataset are those related to emotions, i.e. *dominant emotion* and *end emotion* [21]. Thus, we decided to use them in all the situations when we could compute both, ICCPs and GCCPs. For the new user scenario, when we are able to generate GCCPs only, we found also *user age* informative. It was not considered in the work [21], since it is fixed for a user for a long time (we have an age categorization), and could not be seen as a user context. Because of the same reasons, it is a bad contextual candidate to compute ICCPs.

The most informative contextual variables in the `Unibiz-STS` dataset are *weather* and *companion*. In the `Restaurant & customer` dataset, there are no truly contextual variables. However, we found *smoker*, *drink level*, *dress preference*, *ambience*, *transport*, *personality* and *color* the most useful for the further work.

We tested our method also with other contextual parameters, but the results were similar to those obtained by the traditional

MOST WIEDZY Downloaded from mostwiedzy.pl

TABLE III
MEASURES FOR THE NEW USER COLD-START SCENARIO FOR LDOS–CoMoDa DATASET.

algorithm	precision	recall	MAP	nDCG	MRR	expectedness	novelty	diversity
CSLIM	0.0032	0.0117	0.0045	0.0075	0.0077	0.00	Infinity	0.4257
ctx-BPR	0.0026	0.0134	0.0019	0.0049	0.0037	0.0013	9.8509	0.3117
UISplitting-BPR	0.0008	0.0025	0.0022	0.0029	0.0050	0.00	Infinity	0.3979
BPR	0.0013	0.0006	0.0001	0.0006	0.0018	0.0013	9.7731	0.3377
ctx-FISM	0.0218	0.0992	0.0382	0.0615	0.0765	0.0037	8.2339	0.2843
FISM	0.0179	0.0605	0.0298	0.0455	0.0702	0.0051	7.7079	0.2996
ctx-LDA	0.0218	0.1114	0.0383	0.0637	0.0765	0.0039	8.1771	0.2874
LDA	0.0154	0.0471	0.0280	0.0409	0.0682	0.0054	7.6109	0.2997
ctx-SLIM	0.0077	0.0330	0.0086	0.0176	0.0238	0.0016	9.8338	0.3180
UISplitting-SLIM	0.0032	0.0117	0.0045	0.0074	0.0077	0.00	Infinity	0.4257
SLIM	0.0064	0.0202	0.0085	0.0140	0.0173	0.0019	9.4216	0.3661
ctx-UserKNN	0.0077	0.0330	0.0086	0.0176	0.0238	0.0016	9.8338	0.3180
UISplitting-UserKNN	0.0032	0.0117	0.0045	0.0074	0.0077	0.00	Infinity	0.4257
UserKNN	0.0064	0.0202	0.0085	0.0140	0.0173	0.0019	9.4216	0.3661
ctx-WRMF	0.0077	0.0330	0.0086	0.0176	0.0238	0.0016	9.8338	0.3180
WRMF	0.0064	0.0202	0.0085	0.0140	0.0173	0.0019	9.4216	0.3661

TABLE IV
MEASURES FOR THE NEW USER COLD-START SCENARIO FOR UNIBIZ–STS DATASET.

algorithm	precision	recall	MAP	nDCG	MRR	expectedness	novelty	diversity
CSLIM	0.1121	0.2868	0.0706	0.1571	0.1454	0.00	Infinity	0.1889
ctx-BPR	0.1165	0.3057	0.1836	0.2623	0.3573	0.0417	3.9549	0.2931
UISplitting-BPR	0.2664	0.6596	0.4328	0.5186	0.5217	0.00	Infinity	0.3748
BPR	0.2055	0.5761	0.3583	0.4397	0.4454	0.0634	4.2905	0.3707
ctx-FISM	0.1174	0.3103	0.1883	0.2673	0.3629	0.0417	3.9723	0.2894
FISM	0.2055	0.5728	0.3557	0.4362	0.4358	0.0674	4.1225	0.3667
ctx-LDA	0.1174	0.3103	0.1903	0.2688	0.3637	0.0417	3.9723	0.2894
LDA	0.2055	0.5728	0.3630	0.4427	0.4498	0.0674	4.1225	0.3667
ctx-SLIM	0.1083	0.2610	0.0790	0.1578	0.1656	0.0465	3.7607	0.1864
UISplitting-SLIM	0.1121	0.2868	0.0706	0.1571	0.1454	0.00	Infinity	0.1889
SLIM	0.0899	0.2685	0.0582	0.1354	0.1212	0.0551	5.4390	0.1889
ctx-UserKNN	0.1083	0.2610	0.0790	0.1578	0.1656	0.0465	3.7607	0.1864
UISplitting-UserKNN	0.1121	0.2868	0.0706	0.1571	0.1454	0.00	Infinity	0.1889
UserKNN	0.0899	0.2685	0.0582	0.1354	0.1212	0.0551	5.4390	0.1889
ctx-WRMF	0.1083	0.2610	0.0790	0.1578	0.1656	0.0465	3.7607	0.1864
WRMF	0.0899	0.2685	0.0582	0.1354	0.1212	0.0551	5.4390	0.1889

TABLE V
MEASURES FOR THE NEW USER COLD-START SCENARIO FOR RESTAURANT & CUSTOMER DATASET.

algorithm	precision	recall	MAP	nDCG	MRR	expectedness	novelty	diversity
CSLIM	0.0958	0.1233	0.0671	0.1282	0.2472	0.0131	6.7086	0.3339
ctx-BPR	0.2000	0.1962	0.1518	0.2192	0.3444	0.1588	2.7032	0.1325
UISplitting-BPR	0.1167	0.1437	0.0858	0.1529	0.2903	0.0178	5.9118	0.3214
BPR	0.1750	0.1703	0.1120	0.1952	0.3869	0.1520	2.7644	0.1753
ctx-FISM	0.2000	0.1897	0.1578	0.2185	0.3304	0.1717	2.5809	0.1684
FISM	0.1714	0.1680	0.1074	0.1859	0.3533	0.1562	2.7093	0.1918
ctx-LDA	0.2071	0.1965	0.1535	0.2209	0.3299	0.1724	2.5735	0.1637
LDA	0.1571	0.1502	0.0965	0.1732	0.3474	0.1569	2.7000	0.1933
ctx-SLIM	0.1571	0.1591	0.1271	0.1833	0.3191	0.1352	3.2177	0.1358
UISplitting-SLIM	0.0958	0.1233	0.0671	0.1282	0.2472	0.0131	6.7086	0.3339
SLIM	0.1179	0.1290	0.0790	0.1558	0.3726	0.0938	3.8699	0.1844
ctx-UserKNN	0.1571	0.1591	0.1271	0.1833	0.3191	0.1352	3.2177	0.1358
UISplitting-UserKNN	0.0958	0.1233	0.0671	0.1282	0.2472	0.0131	6.7086	0.3339
UserKNN	0.1179	0.1290	0.0790	0.1558	0.3726	0.0938	3.8699	0.1844
ctx-WRMF	0.1571	0.1591	0.1271	0.1833	0.3191	0.1352	3.2177	0.1358
WRMF	0.1179	0.1290	0.0790	0.1558	0.3726	0.0938	3.8699	0.1844

MOST WIEDZY Downloaded from mostwiedzy.pl

TABLE VI
MEASURES FOR THE TYPICAL SCENARIO FOR LDOS-COMODA DATASET.

algorithm	precision	recall	MAP	nDCG	MRR	expectedness	unserendipity	novelty	diversity
CSLIM	0.00	0.00	0.00	0.00	0.00	0.0013	0.1980	9.9382	0.4099
ctx-BPR	0.0075	0.0259	0.0075	0.0151	0.0209	0.0015	0.3206	9.7732	0.3103
UISplitting-BPR	0.0014	0.0071	0.0021	0.0041	0.0042	0.0018	0.1980	9.5279	0.4065
BPR	0.0075	0.0235	0.0070	0.0144	0.0213	0.0016	0.3063	9.7312	0.3427
ctx-FISM	0.0123	0.0823	0.0473	0.0601	0.0659	0.0034	0.3304	8.3279	0.2866
FISM	0.0130	0.0897	0.0462	0.0615	0.0675	0.0046	0.3148	7.8182	0.3174
ctx-LDA	0.0130	0.0891	0.0504	0.0641	0.0691	0.0034	0.3301	8.3196	0.2864
LDA	0.0137	0.0965	0.0481	0.0645	0.0686	0.0046	0.3165	7.7995	0.3189
ctx-SLIM	0.0062	0.0377	0.0153	0.0226	0.0204	0.0016	0.3186	9.8028	0.3128
UISplitting-SLIM	0.00	0.00	0.00	0.00	0.00	0.00	0.1945	Infinity	0.4098
SLIM	0.0055	0.0360	0.0147	0.0217	0.0216	0.0016	0.2968	9.8361	0.3527
ctx-UserKNN	0.0068	0.0438	0.0249	0.0314	0.0322	0.0020	0.3248	9.2998	0.2995
UISplitting-UserKNN	0.0012	0.0122	0.0030	0.0052	0.0030	0.00	0.1776	Infinity	0.4136
UserKNN	0.0062	0.0386	0.0136	0.0215	0.0206	0.0026	0.3128	8.8376	0.3247
ctx-WRMF	0.0075	0.0512	0.0281	0.0348	0.0307	0.0020	0.3280	9.4117	0.3020
WRMF	0.0048	0.0324	0.0070	0.0140	0.0101	0.0026	0.3190	8.8903	0.3260

TABLE VII
MEASURES FOR THE TYPICAL SCENARIO FOR UNIBIZ-STS DATASET.

algorithm	precision	recall	MAP	nDCG	MRR	expectedness	unserendipity	novelty	diversity
CSLIM	0.0615	0.5426	0.1927	0.2773	0.2007	0.0237	0.2353	8.1371	0.4321
ctx-BPR	0.1129	0.7473	0.4723	0.4527	0.2938	0.0064	0.6137	7.4848	0.1715
UISplitting-BPR	0.0844	0.7393	0.2714	0.3859	0.2789	0.0553	0.3122	4.6419	0.3727
BPR	0.0817	0.5448	0.2636	0.3519	0.3306	0.0062	0.6063	7.6027	0.1976
ctx-FISM	0.0538	0.3082	0.1960	0.1951	0.1400	0.0107	0.6065	6.4944	0.1687
FISM	0.0409	0.2312	0.1220	0.1630	0.1715	0.0101	0.5906	6.6668	0.2060
ctx-LDA	0.0516	0.2975	0.1982	0.1927	0.1398	0.0107	0.6070	6.4910	0.1676
LDA	0.0387	0.2222	0.1222	0.1601	0.1698	0.0101	0.6072	6.6580	0.1854
ctx-SLIM	0.1000	0.6703	0.3490	0.3732	0.2280	0.0060	0.6140	7.5653	0.1722
UISplitting-SLIM	0.0728	0.6452	0.2787	0.3701	0.2900	0.0365	0.3211	6.1613	0.3744
SLIM	0.0860	0.5824	0.2469	0.3452	0.2987	0.0058	0.6039	7.7061	0.1985
ctx-UserKNN	0.0452	0.2885	0.1824	0.1795	0.1258	0.0052	0.6063	7.7853	0.1736
UISplitting-UserKNN	0.0095	0.0906	0.0230	0.0385	0.0234	0.0097	0.1678	9.0467	0.4304
UserKNN	0.0366	0.2240	0.0935	0.1397	0.1489	0.0051	0.5872	7.8932	0.2077
ctx-WRMF	0.0892	0.5502	0.2931	0.3418	0.2720	0.0061	0.6107	7.4968	0.1706
WRMF	0.0828	0.5287	0.2376	0.3298	0.3108	0.0057	0.6009	7.6870	0.2004

TABLE VIII
MEASURES FOR THE TYPICAL SCENARIO FOR RESTAURANT & CUSTOMER DATASET.

algorithm	precision	recall	MAP	nDCG	MRR	expectedness	unserendipity	novelty	diversity
CSLIM	0.0581	0.4068	0.1413	0.2173	0.1806	0.0091	0.3225	7.0568	0.3393
ctx-BPR	0.1129	0.7473	0.4723	0.4527	0.2938	0.1001	0.6137	3.5661	0.1715
UISplitting-BPR	0.0720	0.5000	0.1869	0.2781	0.2411	0.0110	0.3281	6.7637	0.3352
BPR	0.0817	0.5448	0.2636	0.3519	0.3306	0.0971	0.6063	3.6413	0.1976
ctx-FISM	0.0538	0.3082	0.1960	0.1951	0.1400	0.1671	0.6065	2.5757	0.1687
FISM	0.0409	0.2312	0.1220	0.1630	0.1715	0.1568	0.5906	2.7055	0.2060
ctx-LDA	0.0516	0.2975	0.1982	0.1927	0.1398	0.1673	0.6070	2.5723	0.1676
LDA	0.0387	0.2222	0.1222	0.1601	0.1698	0.1575	0.6072	2.6967	0.1854
ctx-SLIM	0.1000	0.6703	0.3490	0.3732	0.2280	0.0939	0.6140	3.6465	0.1722
UISplitting-SLIM	0.0194	0.1165	0.0524	0.0758	0.0824	0.0081	0.2930	7.2665	0.3593
SLIM	0.0860	0.5824	0.2469	0.3452	0.2987	0.0907	0.6039	3.7447	0.1985
ctx-UserKNN	0.0452	0.2885	0.1824	0.1795	0.1258	0.0806	0.6063	3.8666	0.1736
UISplitting-UserKNN	0.0183	0.1022	0.0302	0.0543	0.0505	0.0086	0.3027	7.1533	0.3520
UserKNN	0.0366	0.2240	0.0935	0.1397	0.1489	0.0789	0.5872	3.9318	0.2077
ctx-WRMF	0.0892	0.5502	0.2931	0.3418	0.2720	0.0943	0.6107	3.5781	0.1706
WRMF	0.0828	0.5287	0.2376	0.3298	0.3108	0.0884	0.6009	3.7256	0.2004

MOST WIEDZY Downloaded from mostwiedzy.pl

baseline algorithms. It could be seen as a constrain for the proposed method - it is strongly context dependent.

As could be seen in Tables III, IV, V, VI, VII and VIII, the method is also algorithm dependent. It is impossible to identify one algorithm that is better than others in all of the cases for all of the datasets.

For the new user scenario with the `LDOS-CoMoDa` dataset, our post-filtering method works the best with FISM and LDA algorithms. They improves all of the measures besides *diversity*. The improvements vary for different measures but they are greater than 35 % in comparison with traditional baselines for the first six measures. The reshuffling with other algorithms also gives slightly better results than the traditional baselines in the new user scenario. Surprisingly, baseline context-aware algorithms perform pretty weak according to the accuracy measures. However, they obtained the best values for *expectedness*, *novelty* and *diversity* measures, which is shown in Tab. III.

Interesting is the fact that different algorithms which we combined our method with, are good for a typical scenario in the `LDOS-CoMoDa` dataset. In this case, the best algorithm to work with our approach is WRMF, which improves all metrics besides *unserendipity* and *diversity*. As seen in Tab. VI, all other algorithms combined with our reshuffling method, improve at least some measures - mostly nDCG and MRR, which means that good recommendations are usually higher in the ranking than without reshuffling, even if the number of good recommendations in the top 10 list is the same or smaller.

For the new user scenario with the `Unibiz-STS` dataset, the UI Splitting method with BPR algorithm outperforms all other methods according to all of the measures. For the reshuffling method, the best algorithms are SLIM, User kNN and WRMF, which improve all of the accuracy measures and *expectedness* and only slightly decrease *diversity*, which is presented in Tab. IV.

As could be seen in Tables VII and VIII, our reshuffling method performs the best in the typical scenario when combined with BPR and SLIM algorithms for the `Unibiz-STS` and `Restaurant & customer` datasets. For the `Unibiz-STS` dataset, our method with BPR algorithm gives better results for the *novelty* measure than UI Splitting with BPR, which is surprising, since UI Splitting improves *novelty* for almost all of the cases for all of the datasets.

For the new user scenario with the `Restaurant & customer` dataset, our reshuffling method outperforms all other algorithms according to the accuracy measures when combined with BPR, FISM and LDA algorithms, as shown in Tab. V. Thus, we could conclude that there is no one algorithm which always performs the best with our reshuffling method. It depends on the scenario and the dataset that the experiments are performed on.

From Tables III, IV and V, we could observe that CSLIM and UI Splitting with SLIM and User KNN give exactly the same results for all of the datasets in the new user cold-start scenario. However, this never occurs for the typical scenario.

CSLIM and UI Splitting almost always give better values of the *expectedness*, *unserendipity*, *novelty* and *diversity* measures. Nevertheless, they received the worst *precision* and *recall* values for all of the cases beside the new user cold-start scenario for the `Unibiz-STS` dataset, when UI Splitting with BPR performed the best.

The value of *diversity* measure always decreases after reshuffling the primary recommendations list with proposed method. It seems to be the price for improving the accuracy of the recommendation process.

VII. Conclusions

In this paper we introduce a method for generating a list of top k recommendations, which works well also in the new user cold-start situations. The method is based on user interests model called Contextual Conditional Preferences and it also relies on existing non-contextual algorithms for a ranking task, since it could be classified as a post-filtering technique. We performed experiments on three publicly available datasets, i.e. `LDOS-CoMoDa`, `Unibiz-STS` and `Restaurant & customer`, which contain user ratings, contextual information and item features. The experiments confirmed that our method is applicable in the new user cold-start situations as well as in typical scenarios, which is the main advantage of proposed technique. In the first case, when we do not have any test user's rating in the training set, we use only General Contextual Conditional Preferences, while in the second, we use both types: individual and general ones. We identified different algorithms that work the best with the proposed method for different usage scenarios, e.g. BPR and LDA for the new user situation, and WRMF for a typical scenario in the `LDOS-CoMoDa` dataset. The main constraints of the proposed reshuffling method are the context and the algorithm dependence.

We also compared our reshuffling technique with other context-aware methods, i.e. contextual SLIM and UI Splitting combined with BPR, SLIM and User kNN algorithms. We showed that our method outperforms them according to accuracy measures like *precision* or *recall*, but obtains worse results when considering measures like *novelty* or *diversity*. However, it seems to be the price for improving the accuracy of the recommendation process.

The next step that needs to be taken is a comparison with other cold-start methods. We also plan to automatize the process of a selection of appropriate contextual features, which is crucial to improve our method.

References

[1] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer US, 2011, pp. 1–35. ISBN 978-0-387-85819-7. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-85820-3_1

[2] G. Adomavicius and A. Tuzhilin, *Recommender Systems Handbook*. Boston, MA: Springer US, 2011, ch. Context-Aware Recommender Systems, pp. 217–253. ISBN 978-0-387-85820-3. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-85820-3_7

MOST WIEDZY Downloaded from mostwiedzy.pl

[3] M. Kula, "Metadata embeddings for user and item cold-start recommendations." in *CBRecSys@RecSys*, ser. CEUR Workshop Proceedings, T. Bogers and M. Koolen, Eds., vol. 1448. CEUR-WS.org, 2015, pp. 14–21.

[4] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*, 1st ed. New York, NY, USA: Cambridge University Press, 2010. ISBN 0521493366, 9780521493369

[5] A. Karpus, T. di Noia, P. Tomeo, and K. Goczyła, "Using contextual conditional preferences for recommendation tasks: a case study in the movie domain," *Studia Informatica*, vol. 37, no. 1, pp. 7–18, 2016. [Online]. Available: http://studiainformatica.polsl.pl/index.php/SI/article/view/743/705

[6] A. Kosir, A. Odic, M. Kunaver, M. Tkalcic, and J. F. Tasic, "Database for contextual personalization," *Elektrotehniski vestnik [English print ed.]*, vol. 78, no. 5, pp. 270–274, 2011.

[7] M. Braunhofer, M. Elahi, F. Ricci, and T. Schievenin, "Context-aware points of interest suggestion with dynamic weather data management," in *Information and Communication Technologies in Tourism 2014*, Z. Xiang and I. Tussyadiah, Eds. Springer International Publishing, 2013, pp. 87–100. ISBN 978-3-319-03972-5. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-03973-2_7

[8] B. Vargas-Govea, G. Gonzalez-Serna, and R. Ponce-Medellin, "Effects of relevant contextual features in the performance of a restaurant recommender system," in *Proceedings of 3rd Workshop on Context-Aware Recommender Systems*, 2011.

[9] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole, "Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements," *Journal of Artificial Intelligence Research*, vol. 21, pp. 135–191, 2004.

[10] M. Zanker, M. Jessenitschnig, and W. Schmid, "Preference reasoning with soft constraints in constraint-based recommender systems," *Constraints*, vol. 15, no. 4, pp. 574–595, 2010. doi: 10.1007/s10601-010-9098-8. [Online]. Available: http://dx.doi.org/10.1007/s10601-010-9098-8

[11] K. Stefanidis, E. Pitoura, and P. Vassiliadis, "Managing contextual preferences," *in Info. Sys*, pp. 1158–1180, 2011.

[12] H. Costa, B. Furtado, D. Pires, L. Macedo, and A. Cardoso, "Context and intention-awareness in pois recommender systems," in *Proceedings of 4th Workshop on Context-Aware Recommender Systems*, 2012.

[13] L. Baltrunas and X. Amatriain, "Towards time-dependant recommendation based on implicit feedback," in *Proceedings of 1st Workshop on Context-Aware Recommender Systems*, 2009.

[14] J. S. Lee and J. C. Lee, "Context awareness by case-based reasoning in a music recommendation system," in *Proceedings of the 4th International Conference on Ubiquitous Computing Systems*, ser. UCS'07. Berlin, Heidelberg: Springer-Verlag, 2007. ISBN 3-540-76771-1, 978-3-540-76771-8 pp. 45–58. [Online]. Available: http://dl.acm.org/citation.cfm?id=1775574.1775580

[15] Y. Zheng, R. Burke, and B. Mobasher, "Splitting approaches for context-aware recommendation: An empirical study," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, ser. SAC '14. New York, NY, USA: ACM, 2014. doi: 10.1145/2554850.2554989. ISBN 978-1-4503-2469-4 pp. 274–279. [Online]. Available: http://doi.acm.org/10.1145/2554850.2554989

[16] Y. Zheng, B. Mobasher, and R. Burke, "Cslim: Contextual slim recommendation algorithms," in *Proceedings of the 8th ACM Conference on Recommender Systems*, ser. RecSys '14. New York, NY, USA: ACM, 2014. doi: 10.1145/2645710.2645756. ISBN 978-1-4503-2668-1 pp. 301–304. [Online]. Available: http://doi.acm.org/10.1145/2645710.2645756

[17] A. Q. de Macedo, L. B. Marinho, and R. L. T. Santos, "Context-aware event recommendation in event-based social networks." in *RecSys*, H. Werthner, M. Zanker, J. Golbeck, and G. Semeraro, Eds. ACM, 2015. ISBN 978-1-4503-3692-5 pp. 123–130. [Online]. Available: http://dl.acm.org/citation.cfm?id=2792838

[18] A. Karpus, T. D. Noia, P. Tomeo, and K. Goczyla, "Rating prediction with contextual conditional preferences," in *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2016)*, 2016. doi: 10.5220/0006083904190424 pp. 419–424. [Online]. Available: http://dx.doi.org/10.5220/0006083904190424

[19] J. Cendrowska, "PRISM: an algorithm for inducing modular rules," *International Journal of Man-Machine Studies*, vol. 27, no. 4, pp. 349–370, 1987. doi: 10.1016/S0020-7373(87)80003-2

[20] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "Grouplens: Applying collaborative filtering to usenet news," *Commun. ACM*, vol. 40, no. 3, pp. 77–87, Mar. 1997. doi: 10.1145/245108.245126. [Online]. Available: http://doi.acm.org/10.1145/245108.245126

[21] A. Odic, M. Tkalcic, J. F. Tasic, and A. Kosir, "Predicting and detecting the relevant contextual information in a movie-recommender system," *Interacting with Computers*, vol. 25, no. 1, pp. 74–90, 2013. doi: 10.1093/iwc/iws003. [Online]. Available: http://dx.doi.org/10.1093/iwc/iws003

[22] G. Guo, J. Zhang, Z. Sun, and N. Yorke-Smith, "Librec: A java library for recommender systems," in *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 23rd Conference on User Modeling, Adaptation, and Personalization (UMAP 2015)*, 2015. [Online]. Available: http://ceur-ws.org/Vol-1388/demo_paper1.pdf

[23] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, ser. UAI '09. Arlington, Virginia, United States: AUAI Press, 2009. ISBN 978-0-9749039-5-8 pp. 452–461. [Online]. Available: http://dl.acm.org/citation.cfm?id=1795114.1795167

[24] S. Kabbur, X. Ning, and G. Karypis, "Fism: Factored item similarity models for top-n recommender systems," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '13. New York, NY, USA: ACM, 2013. doi: 10.1145/2487575.2487589. ISBN 978-1-4503-2174-7 pp. 659–667. [Online]. Available: http://doi.acm.org/10.1145/2487575.2487589

[25] T. Griffiths, "Gibbs sampling in the generative model of Latent Dirichlet Allocation," Stanford University, Tech. Rep., 2002. [Online]. Available: www-psych.stanford.edu/~gruffydd/cogsci02/lda.ps

[26] X. Ning and G. Karypis, "SLIM: sparse linear methods for top-n recommender systems," in *11th IEEE International Conference on Data Mining, ICDM 2011*, 2011. doi: 10.1109/ICDM.2011.134 pp. 497–506. [Online]. Available: http://dx.doi.org/10.1109/ICDM.2011.134

[27] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ser. ICDM '08. Washington, DC, USA: IEEE Computer Society, 2008. doi: 10.1109/ICDM.2008.22. ISBN 978-0-7695-3502-9 pp. 263–272. [Online]. Available: http://dx.doi.org/10.1109/ICDM.2008.22

[28] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ser. ICDM '08. Washington, DC, USA: IEEE Computer Society, 2008. doi: 10.1109/ICDM.2008.16. ISBN 978-0-7695-3502-9 pp. 502–511. [Online]. Available: http://dx.doi.org/10.1109/ICDM.2008.16

[29] Y. Zheng, B. Mobasher, and R. D. Burke, "Carskit: A java-based context-aware recommendation engine," in *IEEE International Conference on Data Mining Workshop, ICDMW 2015, Atlantic City, NJ, USA, November 14-17, 2015*. IEEE Computer Society, 2015. doi: 10.1109/ICDMW.2015.222. ISBN 978-1-4673-8493-3 pp. 1668–1671. [Online]. Available: http://dx.doi.org/10.1109/ICDMW.2015.222

[30] B. Smyth and P. McClave, *Similarity vs. Diversity*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 347–361. ISBN 978-3-540-44593-7. [Online]. Available: http://dx.doi.org/10.1007/3-540-44593-5_25

[31] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proceedings of the 14th International Conference on World Wide Web*, ser. WWW '05. New York, NY, USA: ACM, 2005. doi: 10.1145/1060745.1060754. ISBN 1-59593-046-9 pp. 22–32. [Online]. Available: http://doi.acm.org/10.1145/1060745.1060754

[32] Y. C. Zhang, D. O. Séaghdha, D. Quercia, and T. Jambor, "Auralist: Introducing serendipity into music recommendation," in *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, ser. WSDM '12. New York, NY, USA: ACM, 2012. doi: 10.1145/2124295.2124300. ISBN 978-1-4503-0747-5 pp. 13–22. [Online]. Available: http://doi.acm.org/10.1145/2124295.2124300

[33] P. Castells and S. Vargas, "Novelty and diversity metrics for recommender systems: Choice, discovery and relevance," in *In Proceedings of International Workshop on Diversity in Document Retrieval (DDR)*, 2011, pp. 29–37.

MOST WIEDZY Downloaded from mostwiedzy.pl