Parsimonious and Adaptive Contextual Information Acquisition in Recommender Systems

Matthias Braunhofer
Faculty of Computer Science
Free University of Bozen-Bolzano
Bozen-Bolzano, Italy
mbraunhofer@unibz.it

Ignacio Fernández-Tobías
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Madrid, Spain
ignacio.fernandezt@uam.es

Francesco Ricci
Faculty of Computer Science
Free University of Bozen-Bolzano
Bozen-Bolzano, Italy
fricci@unibz.it

ABSTRACT

Context-Aware Recommender System (CARS) models are trained on datasets of context-dependent user preferences (ratings and context information). Since the number of context-dependent preferences increases exponentially with the number of contextual factors, and certain contextual information is still hard to acquire automatically (e.g., the user's mood or for whom the user is buying the searched item) it is fundamental to identify and acquire those factors that truly influence the user preferences and the ratings. In particular, this ensures that (i) the user effort in specifying contextual information is kept to a minimum, and (ii) the system's performance is not negatively impacted by irrelevant contextual information. In this paper, we propose a novel method which, unlike existing ones, directly estimates the impact of context on rating predictions and adaptively identifies the contextual factors that are deemed to be useful to be elicited from the users. Our experimental evaluation shows that it compares favourably to various state-of-the-art context selection methods.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*information filtering*

Keywords

Context-Aware Recommender Systems; Contextual Information; Feature Selection

1. INTRODUCTION

Context-Aware Recommender Systems (CARSs) generate more relevant recommendations than traditional Recommender Systems (RSs) by adapting them to the specific contextual situation of the user (e.g., time, weather, location) [1]. The development of an effective CARS faces many challenges [2]. First, it is necessary to identify the contextual factors that could potentially influence individual's preferences (ratings) and the decision-making process, and hence are worth to be collected, either automatically (e.g., the time, or the location), or by querying the user. The second challenge is to develop a predictive model that is capable of predicting the users' ratings for items under various contextual situations. Finally, the design of a proper human-computer interaction layer on top of the predictive model is the third and last but not least challenge for building a CARS.

In this paper we are focusing on the first challenge. In this respect, previous approaches have mainly applied feature selection techniques to identify which contextual factors should be used in the rating prediction phase. The downside of this approach is that it may force users to add to ratings contextual information that later on, when the prediction model is built, may be found not to be useful for improving the system performance. Because of that, here we propose a new method for identifying which contextual factors should be acquired from the user upon rating an item, so that the user will not enter the value of many contextual factors (parsimonious), and the accuracy of the subsequent recommendations is improved the most.

As a concrete motivation, consider the places of interest (POIs) CARS that is illustrated in Figure 1 and Figure 2 [5]. That system is called STS (South Tyrol Suggests) and it uses 14 contextual factors (e.g., weather, mood, distance, time available). Users may specify any of them when entering a rating for a POI (and also when the user requests context-aware recommendations). These are, however, not all equally important for different user-item pairs, in the sense that they contribute differently to the improvement of the system's rating prediction and recommendation accuracy. In fact, we must avoid any possible waste of time and effort of the user while entering this information and also keep away from the potential degradation of the system performance that could be caused by the usage of irrelevant information. For example, the user's mood may be extremely important to predict the ratings only of certain users, and weather may be an essential factor for one class of items, while negligible for others.

Unlike current state-of-the-art strategies, which measure the relevance of contextual factors on a global basis, our strategy dynamically and adaptively selects the contextual factors to be elicited from the user when she enters a rat-

.

ing for an item. This is achieved by using the CARS rating prediction model itself, and asking the user to specify, when she is rating an item, those contextual factors that if considered in the model would produce a rating prediction for that item that is most different from the prediction computed by a context-free model. We consider this as a heuristics: if this contextual information has an impact on rating prediction it should be acquired and used in the model.

Several CARS algorithms can be used to implement the above mentioned solution; here we employ a new variant of Context-Aware Matrix Factorization (CAMF) [3] that leverages latent correlations and patterns between users, items as well as contextual conditions, thus making it well-suited for selective context acquisition, but also for prediction and recommendation as well. We have compared our proposed method with several state-of-the-art context selection strategies in an offline experiment on two contextually-tagged rating datasets. The results show that the proposed parsimonious and personalized acquisition of relevant contextual factors is efficient and effective, and allows to elicit ratings augmented with contextual factor values that best improve the recommendation performance in terms of accuracy, precision and recall.

We note that parsimoniously acquiring from the user relevant contextual information can be considered as an Active Learning problem [8]. But, while in previous work [6, 4] we focused on the active identification of the *items* to present to the user to rate, in this article we focus on the subsequent decision of identifying which *contextual factors* the user should enter, i.e., under which conditions the user experienced the item.

The rest of the paper is structured as follows. In Section 2, we review the related work. Section 3 introduces our main application scenario. Section 4 presents in detail the proposed context acquisition method. Then, we describe the experimental evaluation in Section 5, and detail the obtained results in Section 6. Finally, conclusions are drawn and future work directions are described in Section 7.

2. RELATED WORK

Finding the most relevant features for building a prediction model has been extensively studied in machine learning. Feature selection is aimed at improving the performance of learning algorithms and gaining insight into the unknown generative process of the data [9]. There are three main approaches to feature selection: wrappers, filters and embedded methods. While wrapper methods optimize the selection within the prediction model, filter methods employ statistical characteristics of the training data to select features independently of any prediction model, and thus are substantially faster to compute. Popular examples of filter methods used in machine learning include mutual information, t statistic in Student test, χ^2 test for independence, F statistic in ANOVA and minimum Redundancy Maximum Relevance (mRMR) [14], which uses the mutual information of a feature and a class as well as the mutual information of features to infer features' relevance and redundancy, respectively. Differently from the two previous methods, embedded methods use internal parameters of some prediction model to perform feature selection (e.g., the weight vector in support vector machines).

Focussing now on CARSs, previous research has explored methods: a) for identifying a priori the factors that should be considered by the system, or b) for selecting, a posteriori, after the ratings and context data was acquired, those factors that are most influential for computing rating predictions. The first task was tackled by exploiting domain knowledge of the RS's designer or market expert [2], whereas the second one was addressed by using feature selection algorithms.

In order to tackle the second task, Odić et al. [13] provide several statistical measures for relevant-context detection (i.e., unalikeability, entropy, variance, χ^2 test and Freeman-Halton test), and show that there exists a significant difference in the prediction of ratings when using relevant and irrelevant context. Another example can be found in [16], where a Las Vegas Filter (LVF) algorithm [12] is employed: it repeatedly generates random subsets of contextual factors, evaluates them based on an inconsistency criterion and finally returns the subset with the best evaluation measure. Finally, Zheng et al. [17] presented a set of approaches based on multi-label classification for the task of recommending the most suitable contexts in which a user should consume a specific item.

Rather than post filtering (after the rating data was acquired) the contextual factors in the rating prediction phase, we are interested in detecting which contextual factors should be acquired upfront from the user in the first place. Hence, when a specific user rates a particular item, our goal is to parsimoniously request and possibly elicit only the contextual factors that improve the most the system performance. These factors can differ for each user-item pair. Moreover, instead of relying on statistical measures, which has been the major trend so far, our work uses a CARS rating prediction model itself to estimate the usefulness of contextual factors. Our approach is similar to some Active Learning [8] solutions of the cold-start problem that also use the rating prediction model to identify which items are better to propose to the users to rate. An example of such an Active Learning method can be found in [10]; it asks users to rate the items whose ratings, if known, contribute most to reduce the system prediction error on a set of held-out test ratings. Another similar approach is the influence-based method presented in [15], which selects those items whose ratings are estimated to have the highest influence on the rating predictions of other items.

3. APPLICATION SCENARIO

Our application scenario is a mobile CARS called STS (South Tyrol Suggests) [5] that is available on Google Play Store and recommends POIs to visit in the South Tyrol region of Italy. STS can generate POI recommendations (Figure 1, left) adapted to the user's and items' current contextual situation by exploiting 14 contextual factors whose conditions (values) are partially acquired automatically by the system (e.g., weather at the POI, season, daytime) and partially entered manually by the user through an appropriate screen (e.g., user's budget, companion, feeling), as shown in Figure 2 (right). More information about the used contextual factors and their possible values, which are called contextual conditions, can be obtained from Table 1. The user's preference model is learned using a set of in-context ratings that the system actively collects from the users and that describe the users' evaluations for the POIs together with the contextual situations in which the users visited the POIs (see Figure 2). However, in our application scenario, given the relatively large number of contextual factors we faced the problem of choosing the contextual factors to ask to the end user upon rating a POI. This is an important and practical problem: asking the value of all the contextual factors is not effective, as it would take too much time and effort for the user to specify them. Moreover, asking the wrong subset of contextual factors may result in the degradation of the prediction model performance and in poor recommendations.

In order to cope with this problem we propose here a novel method that is able to dynamically and adaptively identify the most important contextual factors to be elicited from a specific user upon rating a particular POI. This method serves the purpose of minimizing the amount of information that the users have to input manually, while at the same time allowing the system to still obtain all the relevant information needed to maintain a high level of rating prediction performance. Referring to Figure 2, by means of our proposed method, we can identify for instance the three most relevant contextual factors for "Restaurant Pizzeria Amadè" and then present the user with three screens that step-by-step elicit the contextual conditions for these factors. Otherwise, the user would be required to go through 14 screens, one for each available contextual factor.

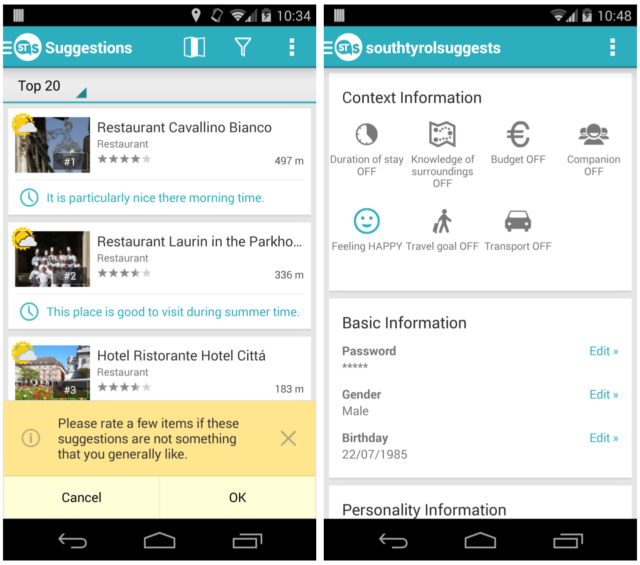

Figure 1: Context-aware suggestions in STS

4. SELECTIVE CONTEXT ACQUISITION

Before presenting the proposed selective context acquisition method, we introduce the CARS predictive model that we have adopted in this study. It is a new variant of the context-aware predictive model CAMF [3] that treats contextual conditions similarly to either item or user attributes and uses a distinct latent factor vector corresponding to each user- and item-associated attribute. More specifically, a contextual condition is treated as a user attribute if it corresponds to a dynamic characteristic of a user, e.g., the mood, budget or companion of the user, whereas it is considered as an item attribute if it describes a dynamic characteristic of the item, e.g., the weather and temperature at the POI. The model is scalable and flexible, and is able to capture latent correlations and patterns between a potentially wide range of knowledge sources (e.g., users, items, contextual conditions, demographics, item categories), making it ideal to derive the usefulness of contextual factors in rating prediction. Given a user u with user attributes $A(u)$, an item i with item attributes $A(i)$ and a contextual situation consisting of the conjunction of individual contextual conditions $c_1, ..., c_k$ that can be decomposed into the subset of user-related contextual conditions $C(u)$ and the subset of item-related contextual conditions $C(i)$, it predicts a rating using the following rule:

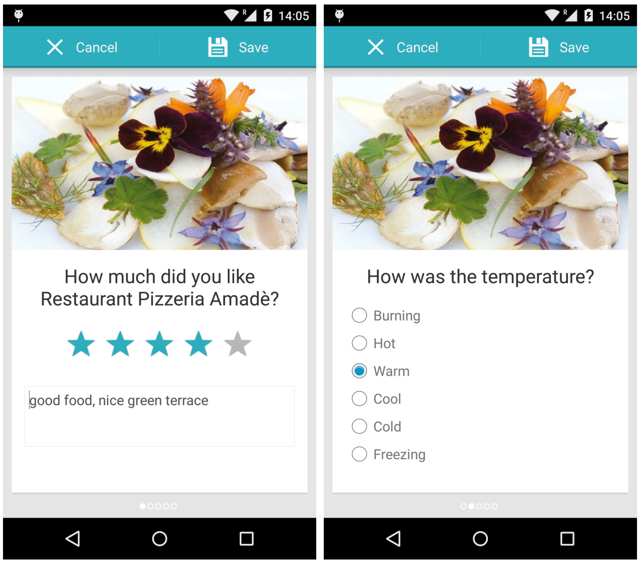

Figure 2: Rating interface of STS

Table 1: Contextual factors used in STS

Contextual factors	Associated contextual conditions
Weather	Clear sky, sunny, cloudy, rainy, thunderstorm, snowing
Season	Spring, summer, autumn, winter
Budget	Budget traveler, high spender, none of them
Daytime	Morning, noon, afternoon, evening, night
Companion	Alone, with friends/colleagues, with family, with girlfriend/boyfriend, with children
Feeling	Happy, sad, excited, bored, relaxed, tired, hungry, in love, loved, free
Weekday	Working day, weekend
Travel goal	Visiting friends, business, religion, health care, social event, education, scenic/landscape, hedonistic/fun, activity/sport
Transport	No transportation means, a bicycle, a motorcycle, a car, public transport
Knowledge of the travel area	New to area, returning visitor, citizen of the area
Crowdedness	Crowded, some people, almost empty
Duration of stay	Some hours, one day, more than one day
Temperature	Burning, hot, warm, cool, cold, freezing
Distance	Far away (over 3 km), nearby (within 3 km)

$$\hat{r}_{uic_1,\ldots,c_k} = (q_i + \sum_{a \in A(i) \cup C(i)} x_a)^\top (p_u + \sum_{b \in A(u) \cup C(u)} y_b) + \bar{r}_i + b_u \quad (1)$$

where q_i is the latent factor vector associated to item i, p_u is the latent factor vector associated to user u, x_a is the latent factor vector associated to an attribute of the item i, that may either describe a conventional attribute (e.g., genre, item category) or a contextual attribute (e.g., weather, temperature), y_b is the latent factor vector associated to an (contextual or not) attribute of the user u. Finally, $\bar{r}_i$ is the average rating for item i, and b_u is the bias associated to the user u, which indicates the observed deviation of user u's ratings from the global average.

CARSs can generate recommendations only after having gathered ratings from the users that are augmented with information about the contextual conditions (values of the contextual factors) observed at the time the item was experienced and rated. It is, however, not always easy to identify which contextual information should be requested and acquired from the users upon rating an item, given the numerous conditions that might or might not be relevant to predict new ratings (in various contextual situations). This is where parsimonious and adaptive context acquisition comes in. Parsimonious and adaptive context acquisition aims at predicting, for a given user-item pair, the most useful contextual factors, i.e., those that when elicited together with the rating from the user improve more the quality of future recommendations, both for that user and for other users of the system.

As we mentioned in the related work section, there exist many algorithms that even though principally designed for context / feature selection (i.e., selection of the most useful contextual factors / features to be used for prediction) can be used also for the purpose of parsimonious context acquisition (i.e., selection of the contextual factors to be elicited from the user upon rating an item). In this paper, we propose a new strategy, which we call *Largest Deviation*. Differently from several state-of-the-art context / feature selection strategies, it personalizes the selection of the contextual factors to ask to the user when rating an item by computing a personalized relevance score for a contextual factor C_j and user-item pair (u, i). To achieve this, for each user u and item i pair (whose rating is acquired) we first measure the "impact" of each contextual condition $c_j \in C_j$, denoted as $\hat{w}_{uic_j}$, by calculating the absolute deviation between the rating prediction when the condition holds (i.e., $\hat{r}_{uic_j}$) and the predicted context-free rating (i.e., $\hat{r}_{ui}$):

$$\hat{w}_{uic_j} = f_{c_j} |\hat{r}_{uic_j} - \hat{r}_{ui}|, \quad (2)$$

where f_{c_j} denotes the normalized frequency of the contextual condition c_j, and is calculated as the fraction of ratings in the entire dataset that are tagged with contextual condition c_j (i.e., $\frac{|R_{c_j}|}{|R|}$). The normalized frequency adjusts the raw absolute deviation by taking into account that the contextual conditions with largest frequency are more reliable. For example, suppose that you want to estimate the impact of *Sunny* weather on the user-item pair (*Alice*, *Skiing*). Let us assume that the rating prediction for *Alice* of *Skiing* is 5 under *Sunny* weather (i.e., $\hat{r}_{Alice\,Skiing\,Sunny} = 5$), and that the corresponding context-free rating prediction is 3.5 (i.e., $\hat{r}_{Alice\,Skiing} = 3.5$). Furthermore, assume that 20% of the ratings in the rating dataset are tagged with *Sunny* weather. Then, the impact of *Sunny* weather on the user-item pair (*Alice*, *Skiing*), i.e., $\hat{w}_{Alice\,Skiing\,Sunny}$, is 0.3 ($0.2 \cdot |5 - 3.5|$).

Finally, these individual scores for the contextual conditions are aggregated into a single relevance score for the contextual factor C_j by simply computing the arithmetic mean of the scores of the various conditions/values for that contextual factor. We conjectured that the contextual factors with largest estimated deviation are more useful to optimize the system performance. Note that this is quite similar to the influence-based Active Learning strategy proposed in [15], which estimates the influence of an item's rating on the rating predictions of other items, and selects the items with the largest influence for rating acquisition.

5. EXPERIMENTAL EVALUATION

5.1 Datasets

In order to evaluate the proposed selective context acquisition method, we have considered two contextually-tagged rating datasets with different characteristics. Table 2 provides some descriptive statistics of both datasets.

- The *CoMoDa* movie-rating dataset was collected by Odić et al. [13]. It consists of ratings acquired in contextual situations that are described by the conjunction of multiple conditions coming from 12 different factors, for instance, time, daytype, season and mood. In addition to the ratings data, this dataset also includes well-defined user attributes (i.e., age, gender, city, country) and movie attributes (i.e., director, country, language, year, budget, genres, actors).
- The *TripAdvisor* dataset is a dataset that we crawled from the TripAdvisor[1] website, which is one of the largest travel sites in the world. It contains ratings for POIs in the South Tyrol region of Italy that are tagged with contextual situations described by the conjunction of contextual conditions coming from three contextual factors, namely, type (e.g., couple, family or business trip), month (e.g., January, February) and year (e.g., 2015, 2014) of the trip. Additionally, also the TripAdvisor dataset has well-defined user (e.g., user location, member type) and POI attributes (e.g., item type, amenities, item locality).

We note that other rating datasets, which are commonly used in CARS research, are not suitable for our analysis since they contain ratings augmented only with the knowledge of a *subset* of all the contextual factors. For instance, in STS, the POIs RS that we mentioned in Section 3, when a user rates a POI she commonly specifies only the value of two or three of the fourteen contextual factors that the system manages (see Table 1). The lack of knowledge of all the contextual factors for each rating is a problem in our case, because, as we will describe in Section 5.2, we wanted to simulate a rating acquisition process where, for a given item, the system requests the user to rate it and to enter the values of the contextual factors identified by the proposed method. Therefore, every contextual factor must be available in the dataset in order to be acquired during the simulated interactions.

[1] http://www.tripadvisor.com/

Table 2: Datasets' characteristics

Dataset	CoMoDa	TripAdvisor
Domain	Movies	POIs
Rating scale	1-5	1-5
Ratings	2,098	4,147
Users	112	3,916
Items	1,189	569
Contextual factors	12	3
Contextual conditions	49	31
User attributes	4	2
Item features	7	12

5.2 Evaluation Procedure

In the evaluation we have simulated system/user interactions where the users rate items specifying only the values of contextual factors (contextual conditions) that have been identified by a context selection strategy. To achieve this, we adapted a procedure which was employed to evaluate Active Learning strategies for RSs [7]. This procedure first randomly partitions all the available ratings into three subsets in the ratio 25:50:25%, respectively: (i) *training set* that contains the ratings that are used to train the context acquisition strategies; (ii) *candidate set* containing the ratings that can be potentially transferred into the training set with the contextual conditions matched by the context acquisition strategies; and finally (iii) *testing set* which contains the part of the ratings that is withheld from the system in order to calculate various performance metrics, i.e., user-averaged MAE (U-MAE), Precision@10 and Recall@10. Then, for each user-item pair (u, i) in the candidate set, the N most relevant contextual factors according to a context usefulness strategy are computed, with N (in different experiments) varying from 1 to the total number of contextual factors in the rating dataset, and the corresponding rating r_{uic} in the candidate set is transferred to the training set as $r_{uic'}$ with $c' \subseteq c$ containing the associated contextual conditions for these contextual factors. For instance, if the top two contextual factors for the user-item pair $(Alice, Skiing)$ are $Season$ and $Weather$, and $Alice$'s rating is $r_{Alice\,Skiing\,Winter,Sunny,Warm,Morning} = 5$, then $r_{Alice\,Skiing\,Winter,Sunny} = 5$ is added to the training set. Since in the considered rating datasets all the contextual factors were specified for each rating, we could always acquire the contextual conditions for the top contextual factors. Finally, the evaluation metrics were measured on the testing set, after training the rating prediction model on the new extended training set.

The above process was repeated 20 times with different random seeds and the results were averaged over the splits to yield more robust estimates (i.e., repeated random subsampling validation [11]).

5.3 Baseline Methods for Evaluation

We have compared the performance of our proposed *Largest Deviation* method with the following three state-of-the-art context / feature selection strategies, in addition to *Random* which we used as a baseline (see Table 3 for a summary of all the tested methods):

- *Mutual Information*: the usage of mutual information for context selection was proposed in [2]. Given a user-item pair (u, i), it computes the relevance score for contextual factor C_j as the normalized mutual information between the ratings for items belonging to i's category and C_j; the higher the mutual information, the better the contextual factor can explain the user ratings for items of a particular category. We note that this strategy depends on the item category but is not personalized, i.e., the same contextual factors are requested to any user upon rating an item belonging to a particular category.
- *Freeman-Halton Test*: proposed as context selection strategy in [13], it calculates the relevance of a contextual factor C_j using the Freeman-Halton test. The Freeman-Halton test is the Fisher's exact test extended to contingency tables larger than 2×2, which is a common alternative to the χ^2 test in case the Cochran's rule about small expected frequencies is not satisfied. The null hypothesis of the test is that the contextual factor C_j and the ratings are independent. If the null hypothesis can be rejected, one can conclude that the contextual factor C_j and the ratings are dependent and thus that the contextual factor C_j is relevant. This test is performed on the full dataset and therefore the selected factors do not depend on the user or the item to be rated.
- *Minimum Redundancy Maximum Relevance (mRMR)*: mRMR [14] is a widely used feature selection algorithm, which, to the best of our knowledge, has not yet been used for the purpose of context selection. It ranks each contextual factor C_j according to its relevance to the rating variable and redundancy to other contextual factors, where both relevance and redundancy are measured based on mutual information. Analogous to the Freeman-Halton test, it is calculated on the full dataset and the selected factors are used for all user-item rating combinations.
- *Random*: the score for a contextual factor C_j is simply a random float in the interval [0, 1). Hence, the top N contextual factors for a user-item pair are simply randomly chosen. This is a baseline strategy used for comparison.

Table 3: Overview of tested strategies for selective context acquisition

Strategy	User Personalization	Item Dependence
Largest Deviation	✓	✓
Mutual Information	✗	✓
Freeman-Halton Test	✗	✗
mRMR	✗	✗
Random	✗	✗

6. EVALUATION RESULTS

Figure 3 and Figure 4 show the U-MAE, Precision@10 and Recall@10 results of the CARS algorithm obtained by applying the various context acquisition strategies on the CoMoDa and TripAdvisor dataset, respectively. In the figures, the x-axis represents the number of acquired contextual

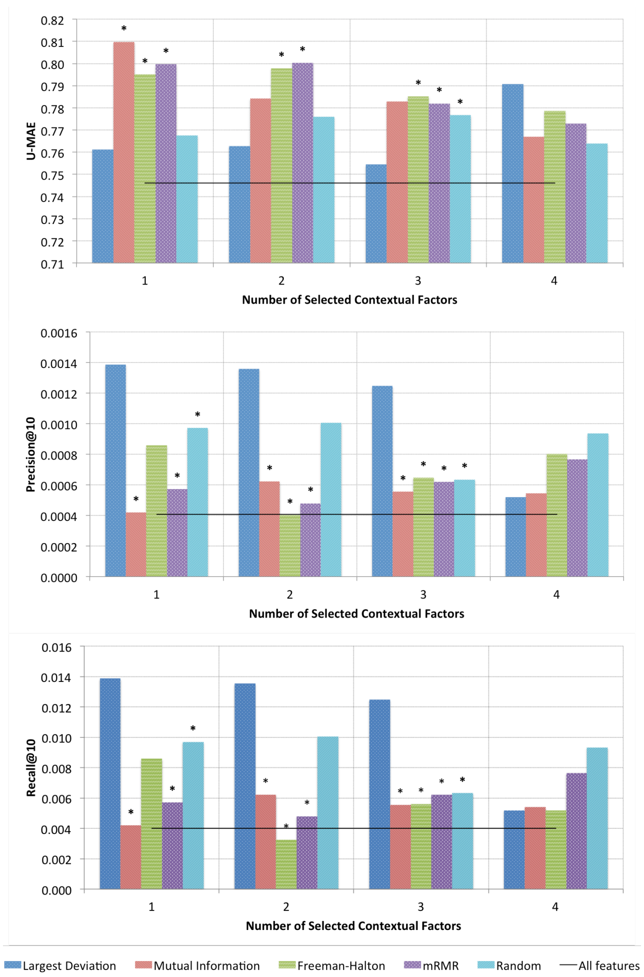

Figure 3: Accuracy, precision and recall results for the CoMoDa dataset

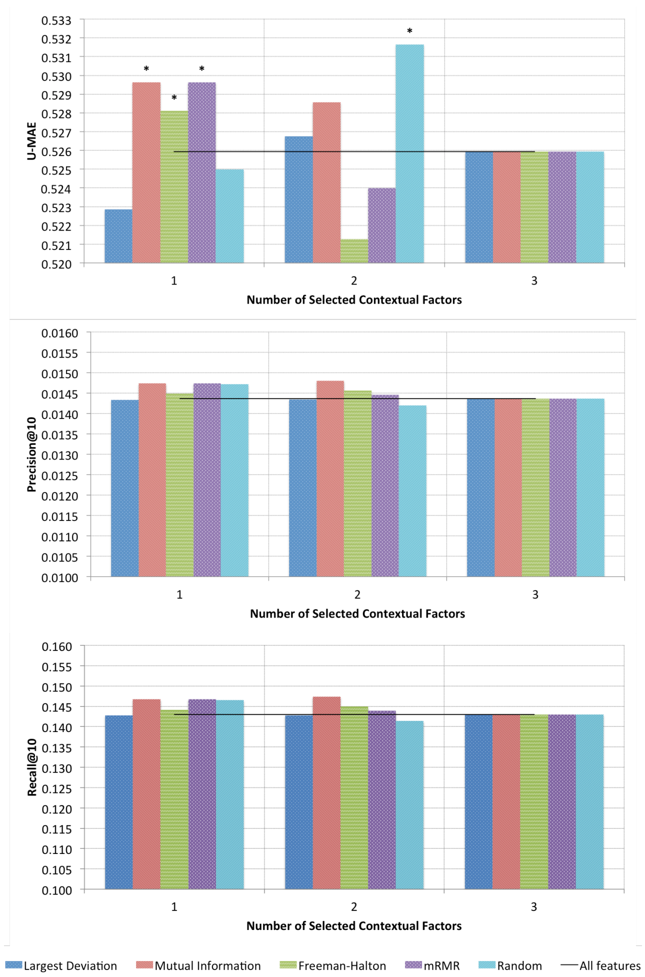

Figure 4: Accuracy, precision and recall results for the TripAdvisor dataset

factors, and statistically significant improvements (paired t-test, $p < 0.05$) of the proposed *Largest Deviation* strategy over the other considered strategies are indicated by asterisks on top of the bars. On the CoMoDa dataset, by using up to three contextual factors, *Largest Deviation* strategy can achieve a significantly better performance in terms of U-MAE, Precision@10 and Recall@10 when compared with the other strategies, i.e., *Mutual Information*, *Freeman-Halton Test* and *mRMR*. With four contextual factors selected, however, there is a notable increase in the U-MAE of *Largest Deviation*, which also causes Precision@10 and Recall@10 to drop. We note that in the graph the number of selected contextual factors goes only up to 4 (out of 12) in order to focus the presentation on the selection of a small subset of factors. In fact, the performance differences between the strategies vanish when more than 4 contextual factors are acquired. We also note that all these 12 contextual factors were supposed to be relevant in the movie recommendation domain [13]. Hence our results clearly indicate that a parsimonious context acquisition strategy is highly beneficial.

Experimental results also indicate that the *Random* strategy has a relatively good performance. Our explanation is that in this strategy, every contextual factor has the same chance of being selected. As a side effect, this allows to better explore the effect of individual contextual conditions on users and/or items. However, the *Random* strategy cannot be practically used since it can often request meaningless contextual factors to the user, e.g., the budget for a POI that can be visited for free. Hence, the random strategy is not directly applicable in a realistic scenario and can only be used in combination with other strategies. This is in line with the findings of Elahi et al. [7], who suggested to consider "partially randomized" strategies that add a small portion of randomly selected items to those identified by another baseline strategy.

Looking at the results for the TripAdvisor dataset, one can note that minor differences (especially in Precision@10 and Recall@10) between the considered context acquisition strategies are present. This is due to the fact that in this dataset in total only three contextual factors are available, thus providing only little potential for parsimonious and adaptive contextual factor selection. Nevertheless, it can be seen that *Largest Deviation* achieves even here a very good accuracy for the tested number of selected contextual factors (1 - 3).

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a new method for parsimonious context acquisition, i.e., for identifying, for a given user-item pair the contextual factors that when acquired together with the rating from the user let the system to generate better predictions. This is an important and challenging problem for CARSs, since usually many contextual factors (e.g., location, weather, time of day, mood) may be available, but only a small subset may be useful and should be asked to the user to avoid an unnecessary waste of time and effort as well as to avoid any degradation of the recommendation model performance.

We have formulated the experimental hypothesis that the proposed parsimonious and personalized selective context acquisition strategy is able to elicit ratings with contextual information that improve more the recommendation performance in terms of accuracy, precision and recall, and also compares favourably with state-of-the-art (context selection) alternatives. In an offline experiment on two rating datasets we were able to confirm these hypotheses.

Selective context acquisition is still a new and under-researched topic, and there are some research questions that deserve future work. Firstly, what is the effect on system performance of employing an Active Learning method for adaptively selecting both the item to rate and the contextual information to add. In this paper we have addressed only partially the problem, by identifying the contextual factors that should be acquired, when a user is rating an item. Secondly, it is interesting to understand how the proposed selective context acquisition method can be extended to generate requests for contextual data that takes into account the possible correlation between contextual factors. Thirdly, it would be interesting to update the evaluation procedure so that it can be used also on datasets of contextually-tagged ratings for which only a subset of the contextual factors is known; as it occurs in the rating dataset collected by our STS app. Finally, we plan to integrate the developed context acquisition method into our STS app so that we can perform a live user study and assess the impact and the benefit of the proposed dynamic and personalized parsimonious acquisition of contextual factors.

8. REFERENCES

[1] G. Adomavicius, B. Mobasher, F. Ricci, and A. Tuzhilin. Context-aware recommender systems. *AI Magazine*, 32(3):67–80, 2011.

[2] L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci. Context relevance assessment and exploitation in mobile recommender systems. *Personal and Ubiquitous Computing*, 16(5):507–526, 2012.

[3] L. Baltrunas, B. Ludwig, and F. Ricci. Matrix factorization techniques for context aware recommendation. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, pages 301–304. ACM, 2011.

[4] M. Braunhofer, M. Elahi, M. Ge, and F. Ricci. Context dependent preference acquisition with personality-based active learning in mobile recommender systems. In *Learning and Collaboration Technologies. Technology-Rich Environments for Learning and Collaboration*, pages 105–116. Springer, 2014.

[5] M. Braunhofer, M. Elahi, and F. Ricci. Usability assessment of a context-aware and personality-based mobile recommender system. In *E-Commerce and Web Technologies*, pages 77–88. Springer, 2014.

[6] M. Elahi, M. Braunhofer, F. Ricci, and M. Tkalcic. Personality-based active learning for collaborative filtering recommender systems. In *AI* IA 2013: Advances in Artificial Intelligence*, pages 360–371. Springer, 2013.

[7] M. Elahi, F. Ricci, and N. Rubens. Active learning strategies for rating elicitation in collaborative filtering: a system-wide perspective. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(1):13, 2013.

[8] M. Elahi, F. Ricci, and N. Rubens. Active learning in collaborative filtering recommender systems. In *E-Commerce and Web Technologies*, pages 113–124. Springer, 2014.

[9] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[10] R. Karimi, A. Nanopoulos, and L. Schmidt-Thieme. A supervised active learning framework for recommender systems based on decision trees. *User Modeling and User-Adapted Interaction*, 25(1):39–64, 2015.

[11] R. Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995.

[12] H. Liu, R. Setiono, et al. A probabilistic approach to feature selection-a filter solution. In *ICML*, volume 96, pages 319–327. Citeseer, 1996.

[13] A. Odić, M. Tkalčič, J. F. Tasič, and A. Košir. Predicting and detecting the relevant contextual information in a movie-recommender system. *Interacting with Computers*, 25(1):74–90, 2013.

[14] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8):1226–1238, 2005.

[15] N. Rubens and M. Sugiyama. Influence-based collaborative active learning. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, pages 145–148. ACM, 2007.

[16] B. Vargas-Govea, G. González-Serna, and R. Ponce-Medellın. Effects of relevant contextual features in the performance of a restaurant recommender system. *ACM RecSys*, 11, 2011.

[17] Y. Zheng, B. Mobasher, and R. Burke. Context recommendation using multi-label classification. In *Proceedings of the 13th IEEE/WIC/ACM International Conference on Web Intelligence*, pages 288–295. IEEE/WIC/ACM, 2014.