Inf Technol Tourism (2017) 17:5–29
DOI 10.1007/s40558-017-0075-6

ORIGINAL RESEARCH

Selective contextual information acquisition in travel recommender systems

Matthias Braunhofer[1] · Francesco Ricci[1]

Received: 30 June 2016 / Revised: 25 December 2016 / Accepted: 2 February 2017 /
Published online: 14 February 2017

Abstract Context-aware recommender systems are information filtering and decision support applications that generate recommendations by exploiting context-dependent user preference data, such as ratings augmented with the description of the contextual situation detected when the user experienced the item. In fact, many contextual factors (e.g., weather, season, mood or companion) may potentially affect the user's experience of an item, but not all of them are equally important for the recommender system performance, or easy to be automatically acquired. Hence, it is important to identify and collect only those factors that truly affect the user preferences (ratings) and can improve the effectiveness of the recommendations computed by the recommender system. Extending our previous work, in this paper, we propose a novel method which adaptively elicits the most useful factors from the user upon rating an item. The proposed method deems a contextual factor as useful to be elicited when a user is rating an item, if it has an impact on the user's predicted rating for that item. The results of our offline experiments, which we executed on travel-related rating datasets, show that the proposed method performs better than other state-of-the-art context selection methods. This paper is an extended and updated version of a conference paper titled 'Contextual Information Elicitation in Travel Recommender Systems' previously published in the proceedings of Information and Communication Technologies in Tourism 2016 Conference (ENTER 2016) held in Bilbao, Spain, February 2–5, 2016.

Keywords Context-aware recommender systems · Travel recommender systems · Context acquisition

✉ Matthias Braunhofer
mbraunhofer@unibz.it

Francesco Ricci
fricci@unibz.it

[1] Faculty of Computer Science, Free University of Bozen-Bolzano, Bozen-Bolzano, Italy

1 Introduction

In our everyday lives, we regularly face choices problems, such as what restaurant to have dinner at, where to spend holidays or which hotel to stay at. Some of these decisions are relatively straightforward and easy to make. However, sometimes decision-making is hard because of too many choices in front of us, or because we do not have sufficient knowledge about the alternative options or simply because of pressing time constraints. This is also true in the context of web search and Internet usage where users are faced with a huge amount of information, e.g., millions of hits for a search, that renders the decision-making slow and complicated.

Recommender systems (RSs) attempt to alleviate this problem. They are personalized information filtering and decision support tools suggesting interesting information items to the user and therefore helping the user to avoid irrelevant ones (Ricci et al. 2015). A wide variety of techniques for generating the recommendations have been proposed. Depending on the exact type of knowledge and data needed to compute recommendations, these techniques can be roughly classified into four major classes (Burke 2007): (1) collaborative filtering, where recommendations are generated by using the preferences (ratings) of other users whose past preferences are similar to those of the target one; (2) content-based, which uses descriptions (features) of items to identify items with features similar to those possessed by the items that the target user has shown a preference for in the past; (3) knowledge-based, where recommendations are based on specific domain knowledge about how certain item features match user needs and interests; and finally (4) hybrid, which are based on the combination of the aforementioned techniques.

A recent trend for RSs is to develop context-aware applications. These systems make use of context-dependent rating datasets, i.e., containing ratings for items tagged with the contextual situations of the user while experiencing the rated item (e.g., weather, temperature, mood or companion). Then, analysing this data, context-aware recommendation techniques can extract hidden dependencies between context and user preferences and adapt the recommendations to the contextual situation of the user when requesting a recommendation (Adomavicius et al. 2011). For instance, in a tourist attraction RS, it is important to determine the weather conditions at the recommended places. In fact, on bad weather conditions the system may have inferred that tourists prefer indoor attractions (e.g., museums, churches, or castles), while on good weather conditions they prefer outdoor attractions (e.g., lakes, mountain lodges, or scenic walks). In this case, the reasoning process is apparently simple, however, in order to be precise and deal with a large space of contextual situations, users, and items, a recommender requires highly sophisticated inferences, which are based on the available rating data. Moreover, the system predictions may depend also on the user's individual sensibility to specific contextual conditions, for instance, the current weather conditions, as shown in Braunhofer et al. (2013).

Travel and tourism is a major application area of Context-Aware Recommender Systems (CARSs). It is due to the fact that changing contexts can significantly affect the tourists' satisfaction and their travel related decisions. Numerous commercial and research context-aware travel RSs, such as Foursquare, Yelp, ReRex (Baltrunas et al. 2012) and South Tyrol Suggests (STS) (Braunhofer et al. 2014), have been already implemented. They exploit the current user's and item's context when recommending Points Of Interest (POIs). However, developing and designing a successful system is not an easy task, and the system designer must face many challenges (Baltrunas et al. 2012). First and foremost, it is required to overcome the major issue of cold-starting the system. This means to design a solution that can compute effective recommendations even when the system has not acquired enough preference data (ratings) (Braunhofer et al. 2014). This issue also involves identifying the contextual factors that do influence the users' individual preferences (ratings) and the decision-making process, and hence are worth to be acquired from the users along with the ratings, either automatically (e.g., the time, season or location), or by querying the user (e.g., the mood, budget or companion). The second challenging task is to develop an effective predictive model that, using possibly a small number of ratings for items in certain contexts, can predict how the ratings change as a function of the different contextual situations. Finally, the design of a proper human-computer interaction layer on top of the predictive model is the last but not least important challenge that must be faced when building a CARS.

In this paper, we address the issue of identifying the contextual factors that influence the user's preferences, and hence should be elicited from the users upon rating a POI. In order to tackle this task, we apply a novel context relevance identification method, which is called Largest Deviation. Largest Deviation estimates the usefulness of a specific contextual factor by measuring the deviation of the user's predicted rating for an item if the system considers or not that factor. The system then dynamically and adaptively selects the contextual factors to be elicited from a specific user when she enters a rating for a particular item as those that produce the largest deviation of the predicted rating for that user-item combination. This approach is very different from current state-of-the-art context selection strategies, which measure the usefulness of a contextual factor: on a global basis, i.e., without considering specific relations between a particular user, item, and context combination; and a posteriori, which means, selecting contextual information after all the information is acquired (Odić et al. 2013; Vargas-Govea et al. 2011). More specifically, current state-of-the-art context selection strategies detect the relevant contextual factors for the entire population of users and items, while Largest Deviation detects the relevant context at the user-item level, i.e., for each user-item pair separately. Furthermore, while current state-of-the-art context selection strategies request all available contextual factors from the users upon rating items and then post-process the obtained rating and context data by filtering out irrelevant contextual information, Largest Deviation a priori identifies the irrelevant contextual information and then only acquires the relevant one from the users upon rating items. We believe that our method suits very well tourism applications since they are characterized by a large number of potentially relevant contextual factors and by very sparse ratings datasets.

The proposed solution relies on an extended matrix factorization-based prediction model to generate rating predictions for users and items under various contextual conditions. Then, it uses the generated predictions rather than the sparse observed ratings to derive the influence of a contextual factor on a particular user-item pair. We have compared the proposed method with several state-of-the-art context selection strategies in a series of offline experiments on three context-dependent leisure and tourism rating datasets. The results show that the proposed parsimonious and personalized acquisition of relevant contextual factors is efficient, effective, and allows to elicit information that best improve the recommendation performance in terms of accuracy and precision.

The proposed context selection technique was first introduced in two previous papers (Braunhofer et al. 2015; Braunhofer and Ricci 2016). In this paper we: provide a more detailed discussion of related work; describe in more detail our main application scenario, i.e., our developed South Tyrol Suggests (STS) app (Braunhofer et al. 2014); present a new and more realistic evaluation procedure that better simulates the influence of the proposed context acquisition method on the evolution of a recommender's performance; and finally we illustrate the results obtained from this new evaluation setting.

The rest of the paper is structured as follows. In Sect. 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 Sect. 5, and detail the obtained results in Sect. 6. Finally, conclusions are drawn and future work directions are described in Sect. 7.

2 Related work

CARSs have been a topic of growing research interest in the recent years. In a CARS the system adapts the recommendations to the specific contextual situations of the user (e.g., her mood or location) and the recommended items (e.g., the weather at the recommended POI) (Adomavicius et al. 2011). To adapt the recommendations to a contextual situation, it is necessary to understand the relationship between user preferences (ratings) and contextual situations. This is operationally implemented by capturing user ratings for items that are augmented with a description of the contextual situation observed when the user experienced the item. For instance, when a user rates a POI, such as a museum, she must also specify the weather, the season, and her mood, when the visit to that museum occurred. Acquiring such data is expensive (in terms of user effort), hence it is important for CARSs to ignore irrelevant and unuseful contextual factors.

Finding the most useful information for building a prediction model has been extensively studied in machine learning and is known as feature selection. Feature selection is aimed at improving the performance of learning algorithms and gaining insight into the unknown generative process of the data (Guyon and Elisseeff 2003). 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) (Peng et al. 2005), 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 previously mentioned methods, embedded methods use internal parameters of the prediction model to perform feature selection (e.g., the weight vector in support vector machines), hence feature selection is an integral component of the model itself.

Focusing on CARSs, previous research has explored two types of methods: (1) for identifying a priori the factors that should be modelled by the system, and (2) for selecting a posteriori, after the ratings and context data was acquired, those factors that are most useful for correctly computing rating predictions.

The first method is exemplified in Baltrunas et al. (2012), where the authors present a survey-based approach to identify the contextual information that is relevant for a mobile tourism RS. They first estimated the dependency of the user preferences from an initial candidate set of contextual factors. This was achieved through a web tool, in which users were requested to evaluate if a particular contextual condition (e.g., "you are on a wellness trip", "it is a cold day", "it is raining") has a positive, negative or no influence on the user's rating of a particular type of POI (e.g., spa, cycling, museum). Using the obtained data, they were able to select the most important contextual factors for different types of POIs. Then, ratings and contextual information for the selected factors that were obtained in the second step were used to train a context-aware matrix factorization model and to provide users with context-dependent recommendations in a mobile application for iPhone.

Odić et al. (2012) identify two approaches to deal with both of these tasks: the first one is defined as "assessment" and it is based on surveying the users, while the second is denoted as "detection" of the context relevance and is performed by mining the rating data. In order to determine which of these two approaches is better, they used real movie rating data, and survey data in which users were asked to rate the influence of each contextual condition on their rating behaviour. Based on the obtained results, they concluded that the detection method performs better than the assessment one for identifying the contextual factors to be exploited in the rating prediction model.

In a related paper (Odić et al. 2013) the same authors investigated in more detail the "detection" approach and provided several statistical measures for relevant-context detection, i.e., unalikeability, entropy, variance, χ^2 test and Freeman–Halton test. Among these measures, they found the Freeman–Halton test as the most useful and flexible measure to identify the relevant and irrelevant contextual factors in the LDOS-CoMoDa rating database. Moreover, the authors showed that the ratings could be predicted more accurately when the system was only using the relevant contextual factors.

Another example of a posteriori selection of the most relevant contextual factors can be found in Vargas-Govea et al. (2011). In this paper, the authors focus on a CARS for restaurants, and show that its efficiency and predictive accuracy can be improved by using a reduced subset of contextual factors. To select contextual factors, the Las Vegas Filter (LVF) algorithm was chosen. LVF repeatedly generates random subsets of factors, computes their evaluation measure based on an inconsistency criterion, which tests the extent to which a reduced subset can still predict the rating values, and finally returns the subset yielding the best result.

Instead of selecting the most relevant contextual factors before or after acquiring ratings from the users, here we focus on parsimoniously and adaptively selecting the most useful contextual factors from the users at the time when they enter a rating for an item. Here, “useful” means that the prediction model is improved by the knowledge of this information. This paper extends our earlier contributions (Braunhofer et al. 2015; Braunhofer and Ricci 2016) through a more realistic evaluation procedure that better simulates the influence of the proposed context acquisition method (Largest Deviation), as well as other baseline methods, on the evolution of the recommender’s performance.

3 Application scenario

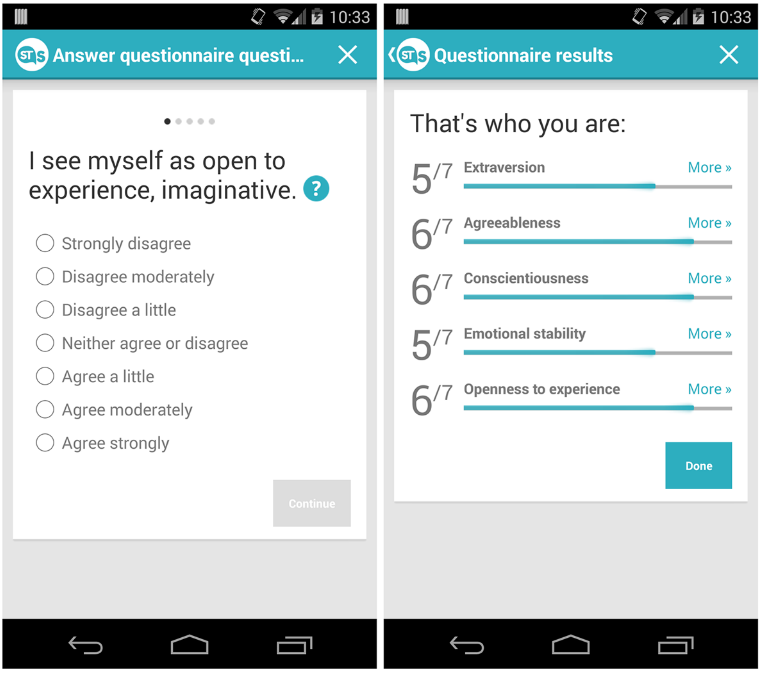
The target application for the proposed context selection techniques is South Tyrol Suggests (STS) (Braunhofer et al. 2014). STS is an Android-based CARS that provides users with context-aware recommendations from a repository of approximately 27,000 POIs, including accommodations, restaurants, attractions, events, public services and parking stations, that are located in the South Tyrol province of Italy. It is available on Google Play Store[1] and as of October 15, 2016, it was downloaded and installed by 997 users. In this section, we describe a typical system-user interaction and show some of the system functions.

3.1 Bootstrapping a user profile

After the user has registered to STS by entering her username, password, birthdate and gender, she is asked to complete the Five-Item Personality Inventory (FIPI) (Gosling et al. 2003), so that the system can measure her Big Five personality traits. These are: openness to experience, conscientiousness, agreeableness, extraversion and neuroticism. The Big Five personality traits are important user’s features since previous research has shown that personality influences human behaviours and that there exist direct relations between personality and tastes and interests (Rentfrow and Gosling 2003). Consequently, the incorporation of human personality allows us to build a better model of user ratings, even when no ratings for the target user are available (see Sect. 3.2). Personality information is also used in STS for more accurately identifying the items that may be known to the user and can be asked to rate (see Sect. 3.3).

Figure 1 (left) shows a screenshot of our application where one of the questionnaire statements is illustrated. The full FIPI consists of the following five

[1] https://play.google.com/store/apps/details?id=it.unibz.sts.android.



10:33

Answer questionnaire questi...

I see myself as open to experience, imaginative. ?

○ Strongly disagree

○ Disagree moderately

○ Disagree a little

○ Neither agree or disagree

○ Agree a little

○ Agree moderately

○ Agree strongly

Continue

10:33

Questionnaire results

That's who you are:

5/7 Extraversion More »

6/7 Agreeableness More »

6/7 Conscientiousness More »

5/7 Emotional stability More »

6/7 Openness to experience More »

Done

Fig. 1 Personality questionnaire

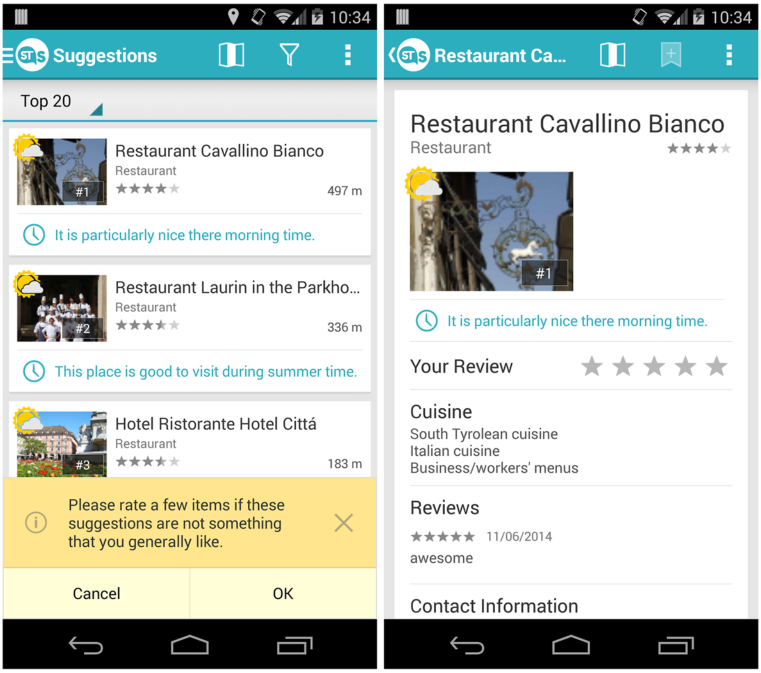
questions which require a 7-point Likert response ranging from “strongly disagree” to “strongly agree”:

1. I see myself as open to experience, imaginative;
2. I see myself as dependable, organized;
3. I see myself as extraverted, enthusiastic;
4. I see myself as agreeable, kind;
5. I see myself as emotionally stable, calm.

Since these questions may be difficult to understand, the application provides users with on-screen help including term definitions that can be accessed by clicking the question mark symbol next to each question, as can be seen in Fig. 1 (left).

3.2 Recommendations

Using the assessed personality (as illustrated in Fig. 1, right), the user’s age and gender (if available), and the current values of 14 contextual factors, which are described in Sect. 3.4, the system identifies and shows a list of 20 highly relevant POIs (see Fig. 2, left). As already noted above, this allows the system to generate personalized recommendations even though at this stage of the interaction no ratings of the user are known by the system. In fact, the system overcomes the new user



Top 20

Restaurant Cavallino Bianco
Restaurant
#1
497 m
It is particularly nice there morning time.

Restaurant Laurin in the Parkho...
Restaurant
#2
336 m
This place is good to visit during summer time.

Hotel Ristorante Hotel Cittá
Restaurant
#3
183 m

Please rate a few items if these suggestions are not something that you generally like.

Restaurant Cavallino Bianco
Restaurant
#1
It is particularly nice there morning time.

Your Review

Cuisine
South Tyrolean cuisine
Italian cuisine
Business/workers' menus

Reviews
11/06/2014
awesome

Contact Information

Fig. 2 Recommendations

problem by learning a rating prediction model that determines how the user preferences depend on her personality.

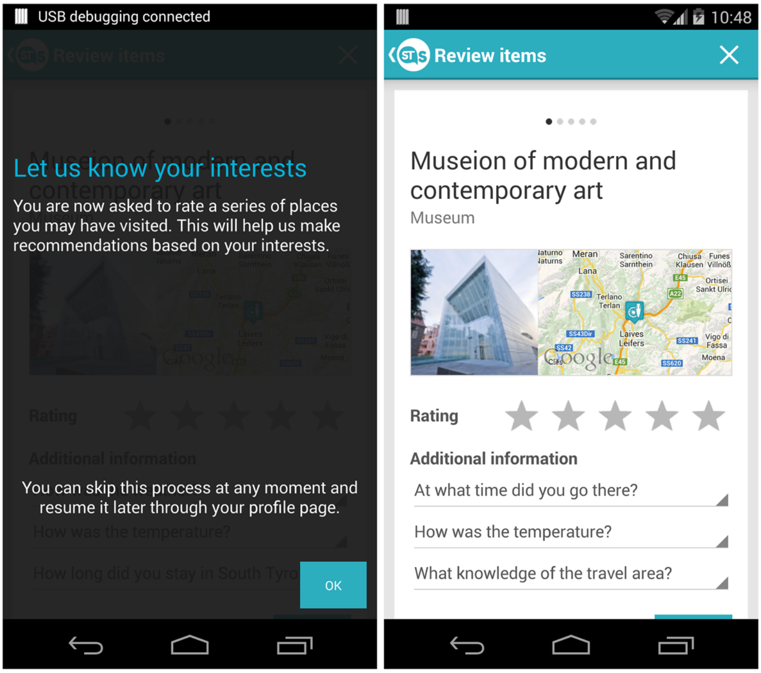
In the event the user is interested in one of the recommended POIs, she can click on it and access the POI details window, as illustrated in Fig. 2 (right). This window shows various information about the selected POI, such as a photo, its name, a description, user reviews, its category as well as an explanation of the recommendation based on the system estimated most influential contextual condition. These are the contextual conditions that according to the rating prediction model have the effect to largely increase the predicted rating for the POI, hence the system argues that because of these conditions the item is particularly suited to be visited.

The system implements other support functions, such as the offered possibility to write a review for the POI, to request a route suggestion for reaching the POI from the current location, to tag the POI and to bookmark the POI, which makes it easy to get back to it later.

Another particularly interesting feature of the POI suggestions screen is that it provides users with two types of pop-up windows with information about how to obtain better recommendations. The first one, as can be seen in Fig. 2 (bottom part of the left screen), requests the user to provide (more) ratings; clicking OK, forwards the user to a screen where she is requested to rate some specific items that are identified automatically by the system (see Sect. 3.3). The second type of pop-up window requests the user to specify contextual factors, such as budget, companion and transport, that the system is not able to acquire without explicitly querying the user. This is managed by an appropriate interface, as illustrated in more detail in Sect. 3.4.

3.3 Items rating

In order to adapt the recommendations to the current contextual situation, it is necessary to understand the relationship between user preferences (ratings) and contextual situations. This requires acquiring user ratings for items together with descriptions of the contextual situations while experiencing the items. In STS, the rating acquisition interaction is started by presenting to the user a semi-transparent screen with a short explanatory text (see Fig. 3, left). After dismissing this screen, the implemented personality-based preference elicitation module identifies five POIs that the system expects the user knows and can rate, and also whose ratings are useful for improving the quality of the subsequent recommendations (see Braunhofer et al. 2014 for the details on the active learning preference elicitation module). Figure 3 (right) shows a screenshot where the user is asked to rate a POI that she has possibly experienced, and to specify the contextual situation of that experience, if she remembers and is eager to provide it. For instance, here, the user is asked to specify the time of the day of her visit, the temperature as well as her knowledge of the travel area. For each of the displayed POIs, the user can specify the value of up to three contextual factors, describing the contextual situation of the user when she experienced the POI. The three displayed factors are the most



Let us know your interests

You are now asked to rate a series of places you may have visited. This will help us make recommendations based on your interests.

You can skip this process at any moment and resume it later through your profile page.

Museion of modern and contemporary art

Museum

At what time did you go there?

How was the temperature?

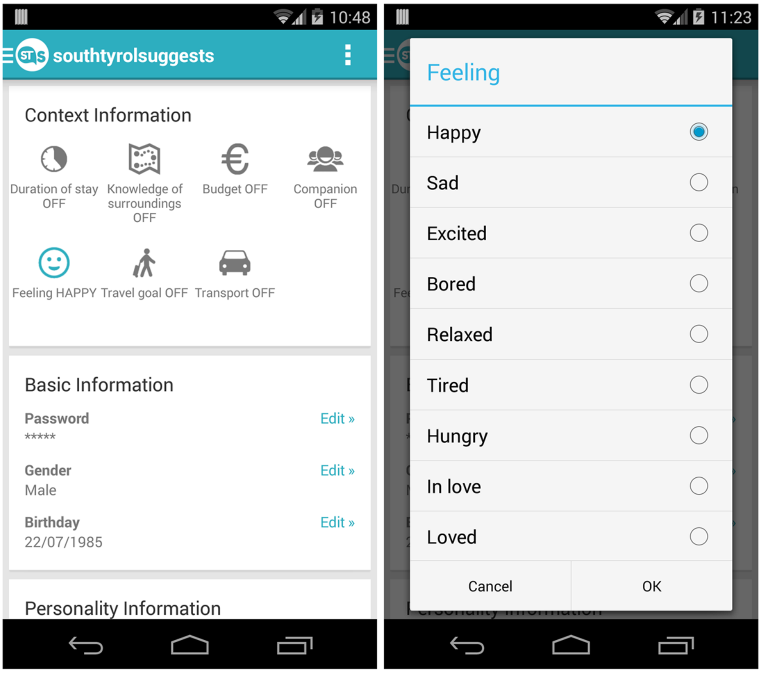
What knowledge of the travel area?

Fig. 3 Active learning

important contextual factors for this particular user-item pair, and are selected from the full set of 14 factors (see next section) according to our proposed parsimonious and adaptive context acquisition strategy as described in Sect. 4. In such a way, the system can minimize the amount of information that the end user has to input manually, while at the same time the system still obtains information useful to maintain a high level of rating prediction performance. Without selective context acquisition, in principle, the user would be required to navigate through 14 drop-down boxes, that is, one for each available contextual factor.

3.4 Context settings

The context settings are used by the system when it generates its recommendations and must be able to adequately describe the current situation of the user. They are accessible from the user profile page, as illustrated in Fig. 4 (left), and allow the user to fine-tune the system's knowledge of the current contextual situation by setting the values of those factors that can not be automatically acquired, such as the duration of the current stay, the user knowledge of the travel area, the current budget, the actual companion and feelings. An overview of the contextual factors and contextual conditions used in the system can be found in Table 1. The contextual factors daytime, weekday, distance (to POI), weather, season, temperature and crowdedness are not entered by the user, but are automatically obtained by the system through GPS/cellular network/WiFi, the internal clock as well as weather



Context Information

Duration of stay OFF
Knowledge of surroundings OFF
Budget OFF
Companion OFF
Feeling HAPPY
Travel goal OFF
Transport OFF

Basic Information

Password
*****
Edit »

Gender
Male
Edit »

Birthday
22/07/1985
Edit »

Personality Information

Feeling

Happy ●
Sad ○
Excited ○
Bored ○
Relaxed ○
Tired ○
Hungry ○
In love ○
Loved ○

Fig. 4 User profile

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)

and traffic web services. The remaining contextual factors, if the user has enabled them, must be entered manually by the user, as displayed in Fig. 4 (right). We note that the full set of contextual factors and their conditions has been derived from Baltrunas et al. (2012) and were selected as found in the literature on consumer behaviour in tourism (Swarbrooke and Horner 2007). That set is sufficiently large to adequately represent the target user context when generating recommendations, however, it is way too large to be specified by the user for each rated item. Thus, it is an ideal set of contextual factors for testing our research hypotheses.

4 Parsimonious and adaptive context acquisition

In a RS, the problem of context acquisition comes in conjunction with the problem of rating acquisition. This is due to the fact that the system 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. For instance, the system must have collected a sufficient number of POI ratings tagged with sunny weather conditions before it can provide good POI recommendations under sunny weather conditions.

Identifying which items should be asked the users to rate is already a non-trivial task. In fact, is common for a user to have experienced, or to have knowledge about, only a small fraction of the items in the catalogue (e.g., POIs). These are the “ratable” items and guessing what they are is the task of a system implemented preference acquisition (active learning) strategy. In our system, we use a personality-based active learning strategy as described in Braunhofer et al. (2014).

Once these items have been selected, the next difficult question that arises is which contextual information should be requested and acquired from the users upon rating an item, given the numerous situation parameters that might or might not be important to know in order 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 fed together with the rating into the predictive model improve more the quality of future recommendations, both for that user and for other users of the system. “Parsimonious” means that the system selectively requests and possibly elicits only the most relevant contextual factors, whereas “adaptive” means that it personalizes the selection of the most relevant contextual factors to each individual user and item. For instance, referring to our STS app, by means of parsimonious and adaptive context acquisition, we can narrow down the set of all possible contextual factors to a small subset of the most useful contextual factors and then present the user with an appropriate GUI that elicits these contextual factors. Otherwise, the user would be required to interact with an inefficient and annoying user interface, and to navigate to and specify the values of the factors, among the full set of 14 contextual factors, that she is able and willing to provide.

Before presenting the proposed selective context acquisition method we introduce the CARS predictive model it relies on. It is a variant of Context-Aware Matrix Factorization (CAMF) (Baltrunas et al. 2012). It 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, the budget or the companion of the user, whereas it is considered as an item attribute if it describes a dynamic characteristic of an item, e.g., the weather or the temperature at a 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).

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 $c_1, \ldots, c_k$ that can be decomposed into the subset of user-related contextual conditions $C(u)$ and the subset of item-related contextual conditions $C(i)$, the CARS model predicts ratings using the following rule:

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

where q_i is the latent vector associated to item i, p_u is the latent vector associated to user u, x_a is the latent factor vector associated to an attribute of 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 attribute of user u. Finally, $\bar{r}_i$ is the average rating for item i, and b_u is the bias associated to user u, which indicates the observed deviation of user u's ratings from the global average.

We recall—from Sect. 2—that generally, there exist many algorithms that even though principally designed for context and feature selection (i.e., the selection of the most useful contextual factors or features to be used for rating prediction) can also be used for selective context acquisition (i.e., the selection of the contextual factors to be elicited from the user upon rating an item). For instance, one could employ: mutual information, t statistics in Student test, χ^2 test for independence, F statistics in ANOVA and minimum Redundancy Maximum Relevance (mRMR) (Peng et al. 2005).

Here, we propose a new strategy, which we call *Largest Deviation*. Differently from state-of-the-art context 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., $\hat{r}_{ui}$):

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

where f_{c_j} denotes the frequency of the contextual condition c_j, i.e., the number of ratings in the entire dataset that are tagged with contextual condition c_j. The 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*). Now, say that system predicts that *Alice* will rate *Skiing* as 5 under *Sunny* weather (i.e., $\hat{r}_{AliceSkiingSunny} = 5$), and that the corresponding context-free rating prediction is 3.5 (i.e., $\hat{r}_{AliceSkiing} = 3.5$). Furthermore, assume that 100 ratings in the rating dataset are tagged with *Sunny* weather. Then, the impact of *Sunny* weather on the user-item pair (*Alice*, *Skiing*) is 150 ($100 \cdot |5 - 3.5|$). We note that, to adjust the absolute deviations of contextual conditions, we also tried other weighting schemes (e.g., log normalization), but we found that the raw frequency gives the best results.

Finally, these individual scores for the contextual conditions are then 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 Rubens and Sugiyama (2007), 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 three context-dependent rating datasets with different characteristics (see Table 2).

- The STS dataset[2] contains the ratings entered by the users of the STS app that we mentioned in Sect. 3. The ratings are for a subset of the items managed by the system and were acquired in contextual situations described by the conjunction of up to 14 different factors. In addition to the ratings data, this dataset includes general user information (i.e., age, gender and the Big-5 personality trait scores) as well as content (POI) metadata in the form of category information.
- The TripAdvisor dataset[3] was crawled from the TripAdvisor website. It contains ratings for POIs in the South Tyrol region of Italy and are tagged with 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, the TripAdvisor dataset has well-defined user (e.g., user location, member type) and POI attributes (e.g., item type, amenities, item locality). We stress that due to the small number of available contextual factors the TripAdvisor dataset provides only little potential for personalization of the contextual factors selection, and thus is far from being the ideal dataset for the purpose of this paper. However, since publicly-available context-dependent rating datasets are scarce, we nevertheless decided to use it for our evaluation. Moreover, the usage of this dataset can help to understand the impact of the proposed technique even in those cases where only few contextual factors are available.
- The CoMoDa dataset is a movie-rating dataset that was collected by Odić et al. (2013). It contains 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. Also the CoMoDa dataset has well-defined user attributes (i.e., age, gender, city, country) and movie attributes (i.e., director, country, language, year, budget, genres, actors).

It is important to note that in the STS dataset, differently from the TripAdvisor and CoMoDa datasets, ratings are augmented with the knowledge of a subset of all the potentially available contextual factors. Indeed, the STS rating dataset originates from the first version of the STS Android app, where users when rating a POI specified the values of, at most, four randomly selected contextual factors (out of a total of fourteen contextual factors). As already mentioned in Sect. 3.3, the current version of STS selects the contextual factors using the proposed Largest Deviation algorithm.

[2] https://www.researchgate.net/publication/305682479_Context-Aware_Dataset_STS_-_South_Tyrol_Suggests_Mobile_App_Data.

[3] https://www.researchgate.net/publication/308968574_TripAdvisor_Dataset.

Table 2 Datasets' characteristics

Dataset	STS	TripAdvisor	CoMoDa
Domain	POIs	POIs	Movies
Rating scale	1–5	1–5	1–5
Ratings	2534	4147	2098
Users	325	3916	112
Items	249	569	1189
Contextual factors	14	3	12
Contextual conditions	57	31	49
Avg. no. of conditions/rating	1.49	3	12
User attributes	7	2	4
Item features	1	12	7

As we will describe in Sect. 5.2, the lack of knowledge of all the contextual factors for each rating implied that during the simulated interactions the value of a contextual factor identified by the proposed method could not be always acquired.

5.2 Evaluation procedure

Before conducting an expensive user study—which is left for future work—we have performed offline experiments aimed at simulating as closely as possible the user interaction with a context selection strategy deployed in our STS system. In these offline experiments we have simulated system/user interactions where the users rate items specifying only the values (conditions) of the contextual factors that have been assessed as relevant by one of the compared context acquisition strategies. To achieve this, we adapted an evaluation procedure that was employed to evaluate active learning strategies for RSs (Elahi et al. 2013).

First, all the available ratings are randomly partitioned into three subsets, in the ratio 25% : 50% : 25%: (1) *training set*, containing the ratings used to initially train the context acquisition strategies; (2) *candidate set*, containing the ratings that could be potentially transferred into the training set with the contextual conditions matched by the context acquisition strategies; and (3) *testing set*, containing the share of the ratings (not considered in system training) that was used for calculating various performance metrics, i.e., user-averaged MAE (U-MAE), which measures the capability of the system to accurately estimate the ratings users would give to items, and Precision@10, which measures the capability of the system to accurately select ten items that the user will like (Herlocker et al. 2004). In particular, given the set of test users U, the set of user u's test ratings $R(u)$, the known rating for the user-item-context tuple (u, i, c), r_{uic}, the predicted rating for the user-item-context tuple (u, i, c), $\hat{r}_{uic}$, the set of test items that u rated positively $T(u)$ (i.e., having $r_{uic} \geq 4$), and the top-10 recommendation list for u as $L(u)$, U-MAE and Precision@10 are defined as follows:

$$\mathrm{U-MAE} = \frac{1}{|\mathrm{U}|}\sum_{\mathrm{u}\in\mathrm{U}}\left(\frac{1}{|\mathrm{R(u)}|}\sum_{\mathrm{r_{uic}}\in\mathrm{R(u)}}|\hat{\mathrm{r}}_{\mathrm{uic}} - \mathrm{r_{uic}}|\right) \tag{3}$$

$$\mathrm{Precision@10} = \frac{1}{|\mathrm{U}|}\sum_{\mathrm{u}\in\mathrm{U}}\frac{|\mathrm{T(u)}\cap\mathrm{L(u)}|}{|\mathrm{L(u)}|} \tag{4}$$

In a first experiment, for each user-item pair (*u*, *i*) in the candidate set, the *N* most useful contextual factors according to a context acquisition strategy are computed, with *N* varying from 1 to the total number of contextual factors in the rating dataset. Then, the corresponding rating r_{uic} in the candidate set was transferred to the training set as $r_{uic'}$ with $c' \subseteq c$ containing the associated contextual conditions for these selected contextual factors, only if these conditions are known for that specific rating. For instance, consider the case that the 2 most useful contextual factors for the user-item pair (*Alice*, *Skiing*) are *Season* and *Weather*, and *Alice*'s rating was $r_{Alice\,Skiing\,Winter\,Sunny\,Warm\,Morning} = 5$, then $r_{Alice\ Skiing\ Winter\ Sunny} = 5$ are added to the training set. In other words, we ignore some available contextual information (i.e., *Warm* and *Morning*) associated to the rating, and we train the predictive model by adding the rating without this information. Finally, the evaluation metrics are measured on the test set, after training the rating prediction model on the new extended training set.

In a second experiment, to better simulate the influence of a context acquisition strategy on the evolution of the RS's performance, we divide the ratings in the candidate set into ten (roughly) equally sized subsets, and we repeatedly apply the procedure used in the first experiment for each of these subsets. In other words, we first transfer back into the training set the ratings in the first subset of the candidate set, and then we train and test the prediction model using the new extended training set. Afterwards, we transfer back into the training set also the ratings contained in the second subset of the candidate set, etc. This is done until the full set of ratings in the candidate set is transferred back into the training set.

Both experiments were repeated 20 times with different random seeds and the results were averaged over the splits to yield more robust estimates (i.e., repeated random sub-sampling validation Kohavi 1995).

We note that in the TripAdvisor and CoMoDa datasets we could always acquire the contextual conditions for the top contextual factors since in these datasets all the considered contextual factors are specified for each rating. Conversely, in the STS dataset each rating is augmented with the knowledge of only a (rating dependent) subset of the contextual factors that the system manages. Hence, in the experiments it often occurred that only a subset of the top contextual factors identified by the method could be really acquired and transferred to the training set along with the rating. This is, however, a more realistic scenario since in actual system/user interactions one cannot assume that the user will always enter all the requested contextual factors.

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 (see Table 3 for a summary of all the tested methods):

- *Mutual Information*: given a user-item pair (u, i), it computes the relevance score for the 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 to the user, i.e., the same contextual factors are asked to be specified by any user upon rating an item belonging to a particular category. We have chosen this strategy since it was reported to be well-suited for context relevance assessment by Baltrunas et al. (2012).
- *Freeman–Halton Test*: 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 states that the contextual factor C_j and the ratings are independent, whereas the alternative hypothesis states that they are dependent. If the null hypothesis can successfully be rejected, it can be concluded that the contextual factor C_j and the ratings are dependent and the contextual factor C_j is relevant. This strategy is calculated on the whole population, i.e., it does not depend on the user or item. According to Odić et al. (2013) the *Freeman–Halton Test* can find the relevant contextual factors to improve the prediction performance of context-aware recommenders.
- *Minimum Redundancy Maximum Relevance (mRMR)*: mRMR (Peng et al. 2005) 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. Similarly to the Freeman–Halton test, it is calculated on a global basis without considering rating differences between users and items. In other words, context selection is not personalized to the user and the item. We have tested our proposed *Largest Deviation* strategy against *mRMR*, since it is one of the most frequently used feature selection algorithms, which, however, to the best of our knowledge, has not yet been used for the specific purpose of context selection.

Table 3 Overview of tested strategies for selective context acquisition

Strategy	User personalization	Item dependence
Largest deviation	✓	✓
Mutual information	×	✓
Freeman–Halton test	×	×
mRMR	×	×

6 Evaluation results

6.1 System performance with context selection

Figures 5, 6 and 7 show the U-MAE and Precision@10 results of the CARS algorithm obtained by applying the various context acquisition strategies on the STS, TripAdvisor and CoMoDa dataset, respectively. As already explained in Sect. 5.2, U-MAE and Precision@10 are two evaluation metrics for RSs that measure important properties affecting the user experience, i.e., accuracy of rating predictions and recommendation precision, respectively. In the figures, the *x*-axis represents the number of acquired contextual 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. We note that the number of selected contextual factors goes only up to a maximum of 3 for TripAdvisor (out of 3), and 4 for STS (out of 14) as well as CoMoDa (out of 12) in order to focus the analysis on a small number of factors. The performance differences between the strategies, in fact, vanish when more than 3/4 contextual factors are acquired.

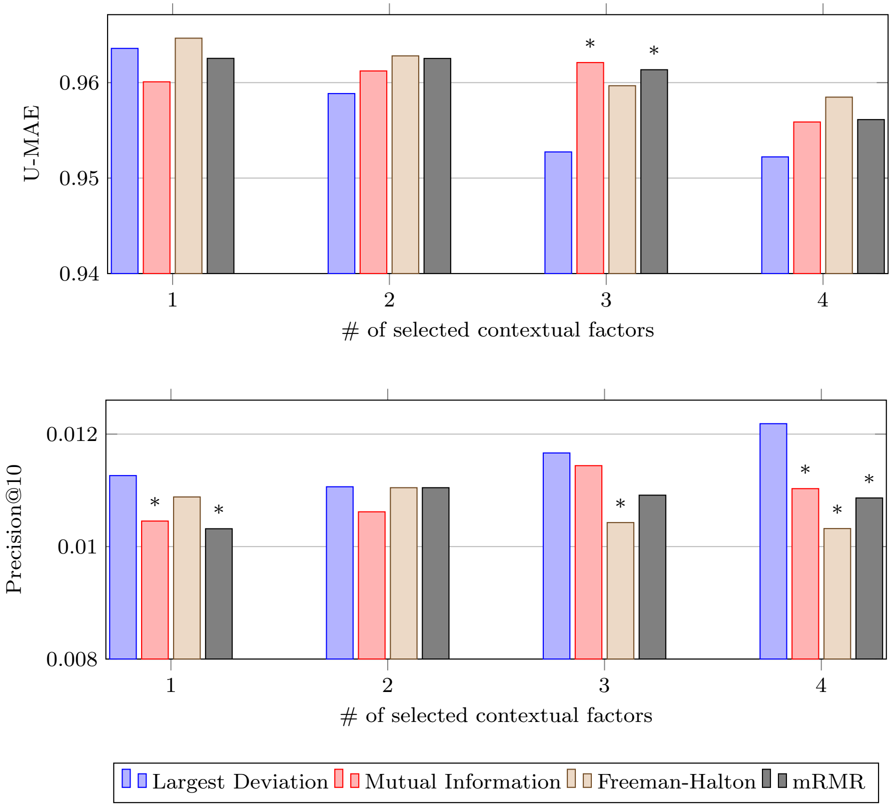


Fig. 5 Experiment 1: accuracy and precision results for the STS dataset

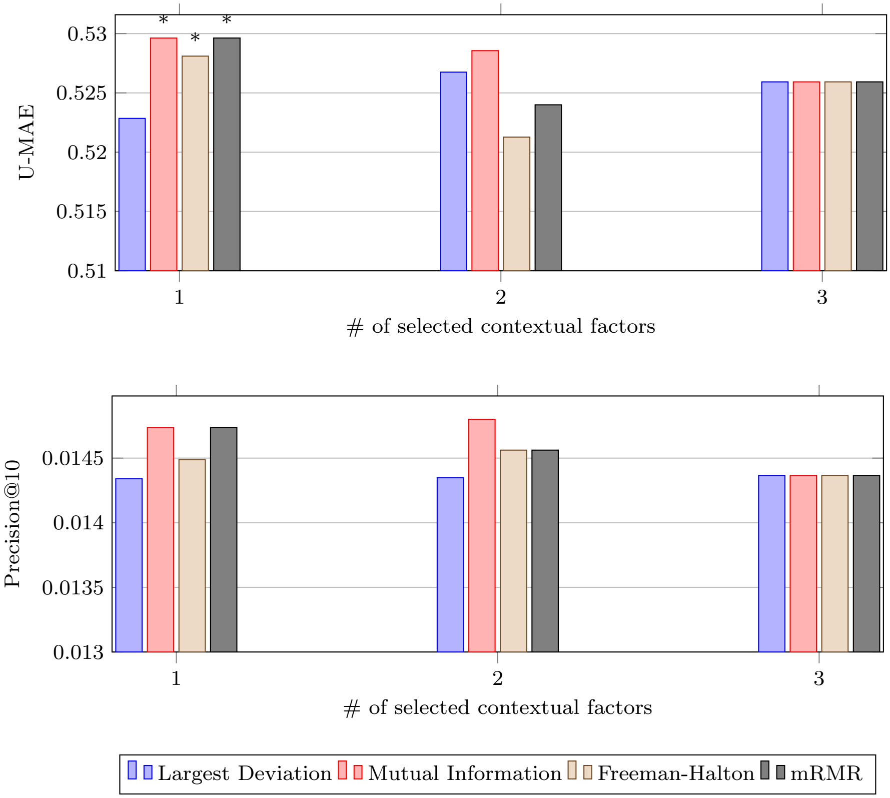


Fig. 6 Experiment 1: accuracy and precision results for the TripAdvisor dataset

In the STS dataset, the best U-MAE (lower is better) and Precision@10 (higher is better) results are achieved by *Largest Deviation*.

An interesting observation in the STS dataset can be made by looking at the average number of contextual conditions acquired by the considered context acquisition strategies. This is shown in Fig. 8. We can observe that the best context acquisition strategy is *Largest Deviation*, which is able to acquire 0.16, 0.30, 0.42 and 0.54 contextual conditions, on average for each rating, when the top 1, 2, 3 and 4 contextual factors are asked from the user to specify, respectively. Hence, it clearly outperforms all the other state-of-the-art context selection strategies, which acquire significantly less contextual conditions. Thus, there is some evidence that our proposed *Largest Deviation* strategy can also better estimate which contextual factors are perceived by the users as relevant since more often it requests to the users contextual information that they have actually entered when rating items.

Looking at the results for the TripAdvisor dataset, one can find that here only minor differences (especially in Precision@10) between the considered context acquisition strategies are present, and especially when only one contextual factor is acquired. This is due to the fact that in this dataset only three contextual factors are available, thus providing only little potential for personalization in contextual factor

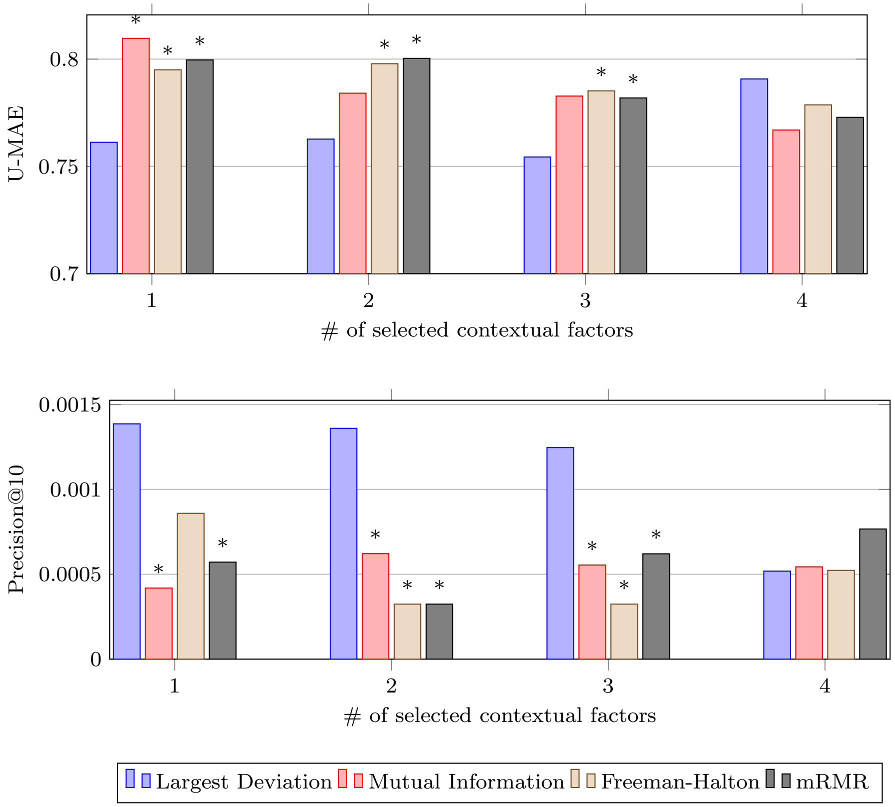


Fig. 7 Experiment 1: accuracy and precision results for the CoMoDa dataset

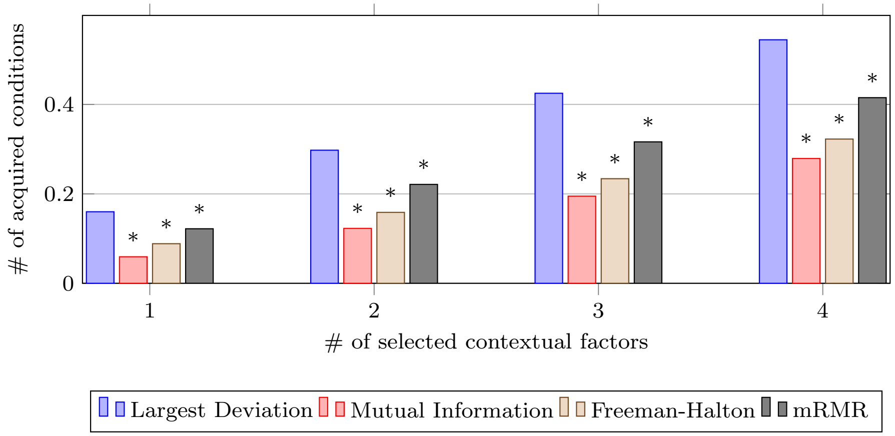


Fig. 8 Experiment 1: no. of acquired contextual conditions for the STS dataset

selection. Nevertheless, it can be seen that *Largest Deviation* achieves a very good accuracy for the tested number of selected contextual factors (1–3). This proves the efficiency and effectiveness of adapting the selection of the relevant contextual factors to the target user-item pair. Similarly, good accuracy results can also be observed for the *Freeman–Halton Test* when two factors are selected.

Finally, on the CoMoDa dataset, we observe that *Largest Deviation* can achieve a significantly better performance in terms of U-MAE and Precision@10 when compared with the other strategies, i.e., *Mutual Information*, *Freeman–Halton Test* and *mRMR*. When four contextual factors selected, then there is a notable increase in U-MAE of *Largest Deviation*, which also causes Precision@10 to drop. We note, however, that these performance differences are neither large nor statistically significant.

6.2 System performance with increasing number of ratings

Figures 9, 10 and 11 show the results of the second experiment that we performed. In this case we studied the system performance with an increasing proportion of ratings collected from the simulated users. We show here U-MAE and Precision@l0 of the CARS algorithm obtained by repeatedly selecting—in 10% steps—the top-N contextual factors according to the various context acquisition strategies, with $N=3$ (for STS), $N=1$ (for TripAdvisor) and $N=3$ (for CoMoDa), which corresponds approximately to 25% of the contextual factors available for each dataset.

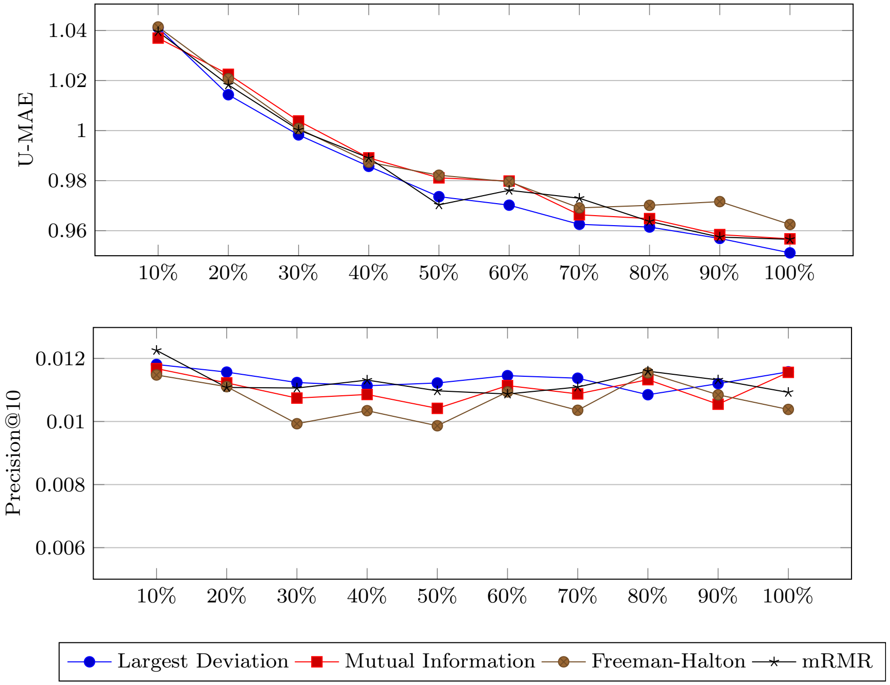


Fig. 9 Experiment 2: accuracy and precision results for the STS dataset

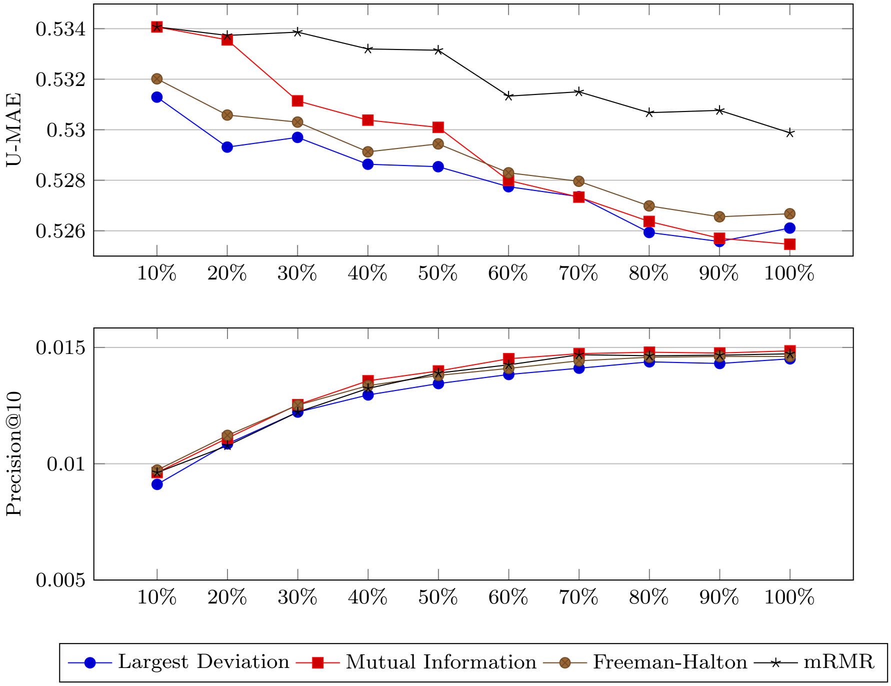


Fig. 10 Experiment 2: accuracy and precision results for the TripAdvisor dataset

Largest Deviation achieves a better performance in terms of MAE when compared with the other strategies, i.e., *Mutual Information*, *Freeman–Halton Test* and *mRMR*. Precision@10 results are quite similar, even though the differences between the different considered strategies are smaller. It is also noteworthy that while in the STS and CoMoDa datasets precision of *Largest Deviation* is always the largest, this does not occur in the TripAdvisor dataset.

However, overall, from these graphs one can clearly see that the benefit of using the proposed parsimonious and adaptive context acquisition strategy is effective in all the states of the preference acquisition procedure; both at the beginning, when few ratings are known and successively, when more preference data are known to the system.

It is again interesting to look at the average number of contextual conditions truly acquired by the considered context acquisition strategies in the STS dataset. This is shown in Fig. 12. We can observe that, at every stage of the preference elicitation procedure, the best context acquisition strategy is *Largest Deviation*, which is able to acquire significantly more contextual conditions, when the top-3 contextual factors are asked from the user to specify. Hence, this further indicates that the proposed *Largest Deviation* strategy can better estimate which contextual factors are truly relevant and should be acquired from the user upon rating an item.

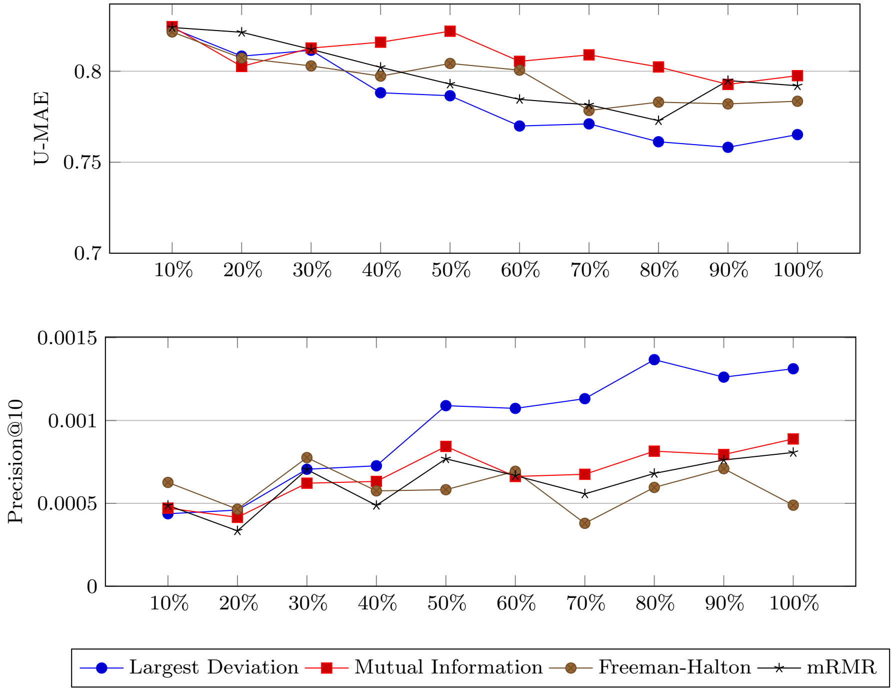


Fig. 11 Experiment 2: accuracy and precision results for the CoMoDa dataset

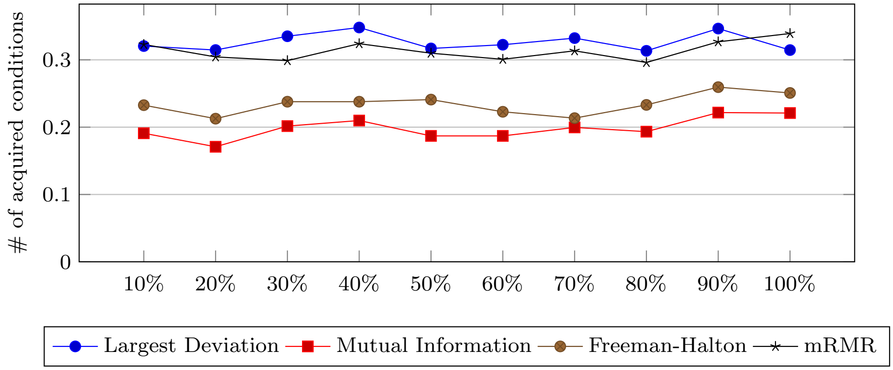


Fig. 12 Experiment 2: no. of acquired contextual conditions for the STS dataset (when 3 are asked)

7 Conclusions and future work

In this paper we have proposed a new method for parsimonious and personalised context acquisition, i.e., a viable technique for identifying the contextual factors that are more useful to request and acquire from users. This contextual information can increase the system rating prediction accuracy and recommendation precision.

This is an important and challenging problem for CARSs, since usually many contextual factors may be requested, but only a small subset is useful and should be asked to the user. The system, in fact, must avoid an unnecessary waste of time and effort which is produced if irrelevant contextual information are requested. Moreover, irrelevant information tends to degrade 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 than state-of-the-art alternatives, in terms of accuracy and precision. The results obtained from offline experiments on three rating datasets confirm these hypotheses.

Parsimonious context acquisition is still a new topic, and there are some research questions that deserve future work. Firstly, one could analyze the effect on system performance of combining the proposed context acquisition technique with an active learning method used for adaptively selecting the items to rate. In other words, the system will intelligently identify on which item the user should reveal her preference and which additional description of the contextual conditions that characterized the item experience should also be given.

Another interesting problem to address in future research is understanding how the proposed method can be improved by considering the correlations between contextual factors. For instance, one may discover that the season of the visit may be largely irrelevant if the system knows the purpose of the visit. Finally, we plan to perform a live user study via our STS app to confirm the results obtained here in the off-line simulations of users' rating behavior.

References

Adomavicius G, Mobasher B, Ricci F, Tuzhilin A (2011) Context-aware recommender systems. AI Mag 32(3):67–80

Baltrunas L, Ludwig B, Peer S, Ricci F (2012) Context relevance assessment and exploitation in mobile recommender systems. Pers Ubiquitous Comput 16(5):507–526

Braunhofer M, Elahi M, Ge M, Ricci F (2014) 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. Springer, Berlin, pp 105–116

Braunhofer M, Elahi M, Ricci F (2014) Techniques for cold-starting context-aware mobile recommender systems for tourism. Intell Artif 8(2):129–143

Braunhofer M, Elahi M, Ricci F (2014) Usability assessment of a context-aware and personality-based mobile recommender system. In: E-commerce and web technologies. Springer, Berlin, pp 77–88

Braunhofer M, Elahi M, Ricci F, Schievenin T (2013) Context-aware points of interest suggestion with dynamic weather data management. In: Information and communication technologies in tourism 2014. Springer, Berlin, pp 87–100

Braunhofer M, Fernández-Tobìas I, Ricci F (2015) Parsimonious and adaptive contextual information acquisition in recommender systems. In: Proceedings of IntRS15

Braunhofer M, Ricci F (2016) Contextual information elicitation in travel recommender systems. In: Information and communication technologies in tourism 2016. Springer, Berlin, pp 579–592

Burke R (2007) Hybrid web recommender systems. In: The adaptive web. Springer, Berlin, pp 377–408

Elahi M, Ricci F, Rubens N (2013) Active learning strategies for rating elicitation in collaborative filtering: a system-wide perspective. ACM Trans Intell Syst Technol (TIST) 5(1):13

Gosling SD, Rentfrow PJ, Swann WB (2003) A very brief measure of the big-five personality domains. J Res Pers 37(6):504–528

Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182

Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst (TOIS) 22(1):5–53

Kohavi R et al (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. Ijcai 14:1137–1145

Odić A, Tkalčič M, Tasič JF, Košir A (2012) Relevant context in a movie recommender system: Users opinion vs. statistical detection. ACM RecSys 12

Odić A, Tkalčič M, Tasič JF, Košir A (2013) Predicting and detecting the relevant contextual information in a movie-recommender system. Interact Comput 25(1):74–90

Peng H, Long F, Ding C (2005) Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 27(8):1226–1238

Rentfrow PJ, Gosling SD (2003) The do re mi's of everyday life: the structure and personality correlates of music preferences. J Pers Soc Psychol 84(6):1236

Ricci F, Rokach L, Shapira B (2015) Recommender systems: introduction and challenges. In: Recommender systems handbook. Springer, Berlin, pp 1–34

Rubens N, Sugiyama M (2007) Influence-based collaborative active learning. In: Proceedings of the 2007 ACM conference on recommender systems. ACM, New York, pp 145–148

Swarbrooke J, Horner S (2007) Consumer behaviour in tourism. Routledge, New York

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