Extending Content-Boosted Collaborative Filtering for Context-aware, Mobile Event Recommendations

Daniel Herzog and Wolfgang Wörndl
Department of Informatics, Technische Universität München, Boltzmannstr. 3, 85748 Garching, Germany

Keywords: Recommender System, Event Recommendations, Content-Boosted Collaborative Filtering, Context-awareness, Mobile Application.

Abstract: Recommender systems support users in filtering large amounts of data to find interesting items like restaurants, movies or events. Recommending events poses a bigger challenge than recommending items of many other domains. Events are often unique and have an expiration date. Ratings are usually not available before the event date and not relevant after the event has taken place. Content-boosted Collaborative Filtering (CBCF) is a hybrid recommendation technique which promises better recommendations than a pure content-based or collaborative filtering approach. In this paper, CBCF is adapted to event recommendations and extended by context-aware recommendations. For evaluation purposes, this algorithm is implemented in a real working Android application we developed. The results of a two-week field study show that the algorithm delivers promising results. The recommendations are sufficiently diversified and users are happy about the fact that the system is context-aware. However, the study exposed that further event attributes should be considered as context factors in order to increase the quality of the recommendations.

1 INTRODUCTION

Recommender systems are information filtering and decision support tools providing personalized recommendations by identifying information or products which best satisfy the user's needs. Recommender systems are increasingly used in a mobile context due to the widespread use of smartphones and tablets. These devices allow more accurate recommendations because they can identify the context of the recommendation in a more detailed manner (Ricci, 2011). In general, context can be described as "any information that can be used to characterize the situation of entities (i.e., whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves" (Dey et al., 2001). Mobile devices are able to collect this type of information as they are often equipped with sensors which allow them to identify context data like the user's current position or speed of travel.

Event recommendations pose a new challenge in the field of recommender systems. Minkov et al. (2010) explain that events, as opposed to movies or restaurants, usually take place only one-time under the same conditions, thus they come with an expiration date. User ratings, which can be considered for recommendations, are usually not available before the event takes place and no longer of importance when the event is over or expired. This lack of ratings makes it impossible to recommend, for example, future events other users with similar preferences have liked. Hence, additional data has to be collected and processed by more sophisticated techniques in order to generate recommendations for events. Event recommendations are moreover a good example showing the importance of context in recommender systems. For instance, in case of a bad weather forecast, recommendations for outdoor events could be inappropriate.

Different techniques are available for recommending items such as events. Hybrid recommenders use the combination of two or more techniques in order to overcome weaknesses of single techniques and to improve the quality of recommendation. Melville et al. (2002) present Content-boosted Collaborative Filtering (CBCF), a hybrid recommendation technique which generates better recommendations than a pure content-based (CB) or a pure collaborative filtering (CF) approach by combining these two techniques.

The main goal of this work is to identify new methods to improve the quality of event recommenda-

Herzog, D. and Wörndl, W.
Extending Content-Boosted Collaborative Filtering for Context-aware, Mobile Event Recommendations.
In *Proceedings of the 12th International Conference on Web Information Systems and Technologies (WEBIST 2016) - Volume 2*, pages 293-303
ISBN: 978-989-758-186-1

tions in a mobile scenario. For this purpose, we introduce a new approach to generate personalized event recommendations. To the best of our knowledge, we are the first to use CBCF within a context-aware recommender to recommend all kinds of cultural events. As a first step, this paper presents important recommendation techniques and CBCF is explained. We further explain the details of our algorithm and show how it can be used for recommending events. Our Android application which implements the context-aware CBCF algorithm is then presented and used for the field study of the system. After discussing our findings, we present related work in the field of recommender systems and event recommendations. The paper finishes with a conclusion and an outlook on future work in this area.

2 RECOMMENDATION TECHNIQUES

Different techniques and algorithms can be implemented in recommender systems in order to generate personalized recommendations. These techniques differ in the information they collect and how they rate items. Two popular techniques are CB predictions and CF.

CB systems try to recommend items which are similar to those the user has liked in the past (Balabanović and Shoham, 1997). A profound knowledge of the item representation is mandatory because CB recommender systems analyze item descriptions and attributes to identify similar items. Pazzani and Billsus (2007) differentiate between structured and unstructured data to represent items: items represented by structured data comprise a set of attributes and there is a known set of values that each of these attributes may have. Unstructured data have no well-defined values, e.g., a text field allowing entry into every possible text. In most cases, semi-structured data, a combination of attributes with a known set of values and free-text fields, is chosen. One advantage of CB recommending is that no critical mass of users is necessary to provide recommendations. Nevertheless, CB recommending comes with some limitations. As the *New User Problem* explains, a new user has to rate a significant number of items before the system can offer accurate recommendations for her or him. Furthermore, CB systems aim to recommend items which tightly fit a user's profile, thus a lack of diversity can be an issue (Adomavicius and Tuzhilin, 2005).

Case-based recommendations are a special case of CB recommenders. According to Smyth (2007), cases are solutions to a given recommendation problem. The simplest scenario is the recommendation of the top k most similar cases matching a user query. Case-based recommendations can be used for recommending items with a structured item representation as they allow the use of sophisticated approaches to judge how cases respond to the user query. Formula 1 is one exemplary approach how to calculate the similarity between a case and the query:

$$Similarity(q,c) = \frac{\sum_{i=1}^{n} w_i \cdot sim_i(q_i,c_i)}{\sum_{i=1}^{n} w_i}, \qquad (1)$$

(Smyth, 2007), where n is the number of attributes of the item. sim_i is the similarity between an attribute i of a query q and a case c. The attribute similarity can be calculated with formula 2 and is weighted by w.

$$sim_i(q_i,c_i) = 1 - \frac{|q_i - c_i|}{max(q_i,c_i)} \qquad (2)$$

(Smyth, 2007). The presented formulas depend on the selected similarity metric. Smyth differentiates between symmetric and asymmetric similarity metrics. A symmetric similarity metric reduces the similarity by the same value if the case attribute value is lower or higher than the query attribute value. An asymmetric metric prefers either higher or lower values.

In contrast, CF recommends items other users with similar preferences have liked (Balabanović and Shoham, 1997). In the majority of cases, nearest neighbor algorithms are implemented in CF recommenders. Schafer et al. (2007) differentiate between *User-Based Nearest Neighbor* and *Item-Based Nearest Neighbor* algorithms. While *User-Based Nearest Neighbor* algorithms call users who rate objects similar neighbors, *Item-Based Nearest Neighbor* algorithms rate items based on similarities between items (Sarwar et al., 2001). Like CB systems, CF has some limitations. In addition to the *New User Problem*, the *New Item Problem* is an issue because items not rated by a substantial number of users cannot be recommended (Adomavicius and Tuzhilin, 2005). These problems define the *Cold-Start Problem* - a serious problem for event recommendations. Because users cannot rate unique future events before actually attending them, most of the items in the system remain unrated (Dooms et al., 2011). The limitation of such a sparely filled user-item rating matrix is called sparsity (Melville et al., 2002). Sparsity causes a low probability of finding a set of users with significantly similar ratings, thus leading to fewer or no recommendations.

The presented limitations of CB and CF recommenders can be illustrated using a simple example.

Table 1 shows a user-item rating matrix with two users A and B and five events. Four events are assigned to the genre comedy, thus they are assumed to be similar. The other event is a musical. User A already provided a good rating to two comedy events, indicated by the + symbol. User B gave positive ratings to the two other comedy events and to the musical. A CB recommender system is able to recommend to each user the comedy events not previously rated as these events are similar. As CB recommendation is not taking other users into account, no further recommendations for User A are possible. A CF approach, however, is not able to recommend any event to any user. User A and B had not initially rated any common events, thus they cannot be identified as neighbors.

Table 1: Exemplary user-item rating matrix for events.

	User A	User B
Comedy 1	+	
Comedy 2	+	
Comedy 3		+
Comedy 4		+
Musical 1		+

Hybrid recommender systems combine two or more techniques in order to overcome such limitations. One example of a hybrid recommending method is CBCF. CBCF is a feature augmentation hybrid which uses the output from a CB prediction to generate recommendations in the subsequent CF phase (Burke, 2002). The initial CB prediction is executed on the user-item rating matrix containing all ratings given by the users. The predicted ratings are then stored in the user-item rating matrix, now called pseudo user-item rating matrix, which is characterized by a lower sparsity. Finally, the CF algorithm is executed on the pseudo matrix (Melville et al., 2002). In the presented example (Table 1), the CB approach extends the user-item rating matrix by CB predictions, in this case, two comedy events for each user. Based on the new pseudo matrix, CF is able to identify User A and B as neighbors. This allows the recommendation of the musical to User A since it was highly rated by a user with similar preferences.

3 CONTEXT-AWARE CONTENT-BOOSTED COLLABORATIVE FILTERING FOR MOBILE EVENT RECOMMENDATIONS

The example presented in Table 1 illustrates the strengths of hybrid recommenders. CB recommenders identify events similar to those the user already liked. To arrive at this conclusion, past feedback has to be analyzed but no critical mass of users is necessary. CF promises an increase in the variety of the recommendations but because of the lack of ratings for future events, CF has to be combined with other techniques. This is why we use CBCF for our event recommender as it combines these two techniques and promises better recommendations than pure CB or CF recommenders (Melville et al., 2002). Furthermore, we want to show that CBCF can be used for context-aware recommendations. We briefly introduced the idea of our algorithm and first results of a field study in (Herzog and Woerndl, 2015).

Adomavicius and Tuzhilin (2011) present three different paradigms for incorporating contextual information into a recommendation process. Our suggested context-aware CBCF approach implements *contextual pre-filtering* to diminish the amount of events available for recommendations before the actual recommendation process takes place. This approach is advantageous because events which are impossible to recommend are excluded immediately: for example, an event taking place too far away from the user's location. In future work, we plan to try the other paradigms for incorporating context as well and compare those results to the solution presented in this paper.

After the pre-filtering phase, our algorithm analyzes user feedback on past recommendations in order to predict the ratings of the remaining events in the CB recommendation phase. These predictions are entered in the pseudo rating matrix. Finally, the CF phase is executed in order to consider all predicted ratings of all users and identify additional recommendations. This step is necessary to increase the variety of our event recommendations. To sum up, the proposed recommendation algorithm comprises three phases:

1. Contextual pre-filtering
2. Content-based prediction
3. Collaborative filtering

In the following, we describe these phases in detail.

3.1 Contextual Pre-filtering

Before predicting ratings for the events available in the database, the number of possible recommendations is decreased by taking context into account. Relevant context factors in this work were identified in expert interviews with selected representatives from the German event and ticketing industry. These context factors are:

- Current Position: It is likely that the user prefers selected venues but the travel distance to these venues has to be appropriate. The system should be able to identify the user's current location and the user should be able to set a radius around it. Only venues within this radius should be considered for recommendation.
- Temporary Preference of Selected Genres: The algorithm should ignore certain genres during the recommendation process, e.g., when a genre is inappropriate for the user's companions. The user should be able to select or deselect genres in order to tell the system the appropriate genres.
- Budget: The algorithm has to respect the user's available budget. For this purpose, the user has to set an upper limit for event prices or the sum she or he can spend per week or month.
- Weekday: Recommendations have to respect the identified days the user is available for events (e.g., only on weekends). The user should be able to tell the system the weekdays for the pre-filtering of events.
- Time of Day: Recommendations are only useful when the user is available at the suggested time of day (e.g., not during the morning). The user should be able to tell the system the times of day for the pre-filtering of events.
- Scheduling Conflict: If the user already purchased tickets for a recommended event, no further events which take place at the same time should be recommended. The recommender system has to identify such conflicts automatically and exclude events if necessary.

As described, these factors are used as criteria for exclusion. If the context of a potential recommendation exceeds a defined threshold, e.g., the distance to the venue, the corresponding event will not be considered for recommendation. The developed prototype in section 4 allows the user to set and modify these thresholds as explained above.

3.2 Content-Based Prediction

After excluding events which do not satisfy the context constraints, the classical CBCF approach can be executed. At first, the CB prediction phase of CBCF has to be adapted to the special case of event recommendations. In this section, we describe how we analyze event attributes rated by the user in the past in order to estimate ratings for events comprising these attributes. At this point, context is no longer the focus. Nevertheless, the presented context factors are reflected in the item representation during the CB prediction, e.g., when determining how much the user likes a certain venue.

As explained, items can be represented by structured data, unstructured data and semi-structured data. Based on the expert interviews we conducted, events are mainly characterized by structured data. A Munich-based event and ticketing company provided a dataset with approximately 3700 real events which were used for the survey in section 5. The dataset includes the following, relevant event attributes:

- The event name
- The name and address of the venue
- The genre
- The exact date when the event starts
- The vendor

The dataset did not provide information about the ticket price which therefore was not considered in this work. The structured characteristic of events is the reason why we propose a case-based approach for the CB prediction of ratings.

In order to adapt CBCF to event recommendations, we propose a slightly different similarity metric than the symmetric and asymmetric similarity metrics presented in section 2. Nominal case attributes such as the genre or the venue are treated as binary values. If an event takes place at a certain venue, its value is 1 for this venue and 0 for all other venues. The query attributes depend on the user's history. If a user liked 90% of all recommendations of a venue, the query value q_i for this venue is 0.9. The attribute similarity is calculated with Formula 2 using q_i equals 0.9 and the case value c_i equals 0 or 1 as input. Other attributes, such as the event's price, could apply an asymmetric similarity metric. If the price is lower than the average as identified by the user's history, the similarity will be less reduced than for a higher price.

The challenge is to find a way to calculate the query value q_i for each attribute. As described, user feedback for an upcoming event is usually not available up-front, hence the user's history has to be used as a basis for the calculation. For event recommendations, two scenarios are possible: the user either likes or rejects a recommendation. Additionally, if the user actually purchases tickets for a recommended event, this would be considered a more positive feedback than just liking the recommendation. In this work, we count all liked, rejected and purchased recommendations and calculate the share of the positive feedback. Thereby, liked events are increased by factor 3 and events with purchased tickets by factor 5 as we assume that users often have to reject recommendations only because of time constraint issues. If a user liked one theater recommendation, rejected another theater recommendation but purchased tickets for a third theater performance, q_i for $i = theater$ will be calculated as $\frac{8}{9} \approx 0.8889$. This value means that the user likes the attribute theater in a recommendation with a probability of 88.89%.

If q_i can be calculated for a sufficient number of attributes, i.e., past feedback is available for these attributes, a prediction is called accurate. Formula 1 can be used to calculate the similarity between the item and the query corresponding to the value of the recommended item for the user. This value is stored in the pseudo user-item rating matrix and used for the upcoming CF phase.

3.3 Collaborative Filtering

Recommendations which were not rated during the CB phase are candidates for the upcoming CF phase. Rating recommendations in the CF phase is of prime importance because focusing solely on the user history could lead to a poor diversity of recommendations. As explained, *User-Based Nearest Neighbor* and *Item-Based Nearest Neighbor* algorithms are available for CF. In this work, we implement a *User-Based Nearest Neighbor* algorithm as it is already used for the CBCF approach in (Melville et al., 2002).

Some assumptions have to be made to reach valuable predictions. Only users with a similarity of at least 50% are considered as neighbors. The CF fills the pseudo matrix with additional ratings. In the end, the events with the highest value for the user are recommended. Every recommendation has to have a value of at least 0.5 on a scale from 0 to 1. In order to achieve a high serendipity, we divide the events in three groups. Around one third of the recommendations are events which take place within 7 days, one third within one month and one third not within one month. If not enough events are available for a group, the list of recommendations is filled with the best available recommendations. We limit the maximum number of recommendations at one request to 10 to avoid overwhelming the user with events. As a result, our context-aware CBCF method is able to recommend a set of events which respect the user's context and expectations and promise a satisfying diversity.

4 DEVELOPED MOBILE PROTOTYPE

We developed an Android application to implement our algorithm and to evaluate the recommender system. The application can be installed on devices running Android 4 or newer.

For our prototype, the dataset is stored in a MySQL database connected to a RESTful web service we developed. When a client demands recommendations from the web service, it transmits the user id and the current context information to the server. The web service then is able to provide personalized recommendations based on the user-item rating matrix which is stored centrally in the MySQL database. These recommendations are received by the application and displayed to the user.

In this section, we briefly present the application. The application can be used without a login. A unique, device-dependent id allows personalized recommendations without creating a user account. After starting the app, a request is sent to the server automatically and the user immediately receives a list of new recommendations (Figure 1). Recommendations are presented as cards, a design principal made popular through applications such as Tinder and Google Now. The main advantages of this representation are the low cognitive load because of the small number of items visible and the fact that the user can navigate through the app and provide feedback with only one hand (Torkington, 2014). In the developed app, the cards provide the most interesting information about the recommended event such as its name, the genre, the location and the date. Furthermore, the calculated rating for the user is expressed as a percentage and presented together with a short explanation. For example, the explanation ”dein Feedback” (German for ”your feedback”) indicates that the event is recommended because of a rating from the CB recommendation phase. The user can swipe recommendations to the right to give a positive rating to a recommendation and to the left in order to reject the recommendation and to provide a negative rating.

Clicking on a recommendation displays a detailed view of the selected event. At the selected event, the user can find additional information concerning the venue or find Facebook friends who are attending the event. A button linking to the ticket vendor is also available. For testing purposes, the prototype assumes

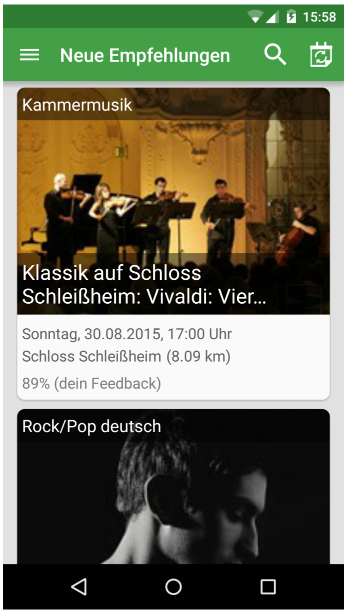

Figure 1: New recommendations provided by the recommender system.

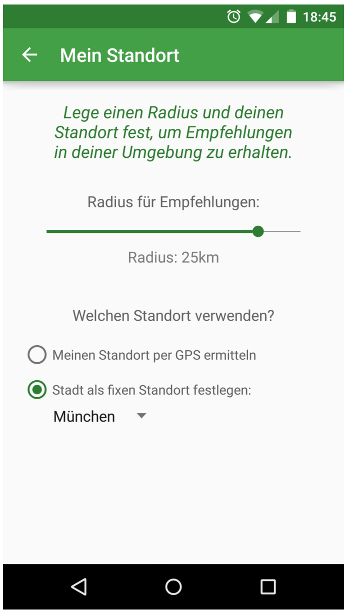

Figure 2: Geographical context settings.

that a user clicking this button eventually purchases a ticket.

Using the *Navigation Drawer*, the app's menu bar, the app user can call up the settings view allowing her or him to modify the thresholds for the contextual pre-filtering (cf. section 3.1). Figure 2 shows a screenshot of the geographical context settings. The user can choose the radius threshold for the contextual pre-filtering of events by moving the slider thumb. The radius is drawn around a certain position which can be either determined by activating the device's GPS sensor or chosen from a predefined list with cities. In this example, the current user position is set to the city center of Munich and only events within a radius of 25 kilometers are considered for recommendation. In the second settings view, the user can determine her or his desired weekdays and times of day for the pre-filtering phase by selecting the corresponding time slots (Figure 3). For this purpose, we offer a calendar view which splits every day of the week into five times of day from morning (6 am until 11 am) until night (10 pm until 6 am). This design allows the user to select time slots for each day individually. This selection option is important for event recommendations because we believe a potential visitor cannot attend events every day at the same time. The additional check box allows users to select and deselect all slots at once. The third setting view displays a list of genres which can be selected or deselected by the user in order to determine appropriate genres for the pre-filtering phase.

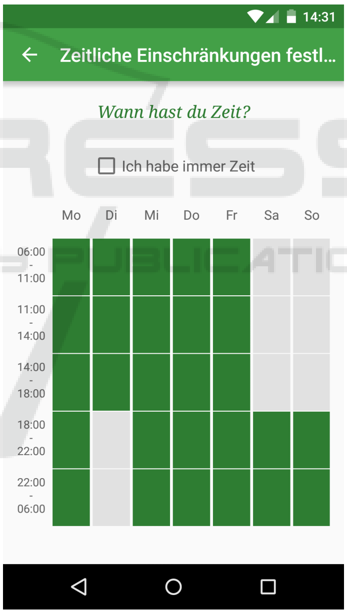

Figure 3: Temporal context settings.

5 FIELD STUDY

In this section, we describe a field study conducted to evaluate the developed recommender system and the context-aware CBCF algorithm. The study was conducted after the implementation of the presented system. The main goal of the study was to test the recommender system in a realistic scenario. The study was intended to deliver insights into the quality of the recommender system. The method of influencing the context-awareness in the app settings also had to be evaluated. The study results are presented and interpreted at the end of this section.

5.1 Setting and Procedure

The field study was conducted as a two-week beta test, meaning, the users installed the application on their own smartphones. Users were advised to use the application as if they were using an application they installed voluntarily in everyday life. This means that during the test, users were allowed to use the system whenever and wherever they wanted to, in the desired intensity. Nevertheless, they were recommended to use the app at least for a certain amount of time to get a first impression of the system and the delivered recommendations.

After two weeks, the beta test terminated and a survey was sent to the participants. The main objective of the survey was to evaluate: the algorithm, the offered method to influence the context-awareness, the user interfaces and the recommender system as a whole. Table 2 lists all evaluation statements included in the survey. The participants had to rate these statements using a 5-point Likert scale with 1 representing no agreement at all and 5 representing complete agreement.

Table 2: Survey statements.

#	Statement
1	Overall, I like the recommender system
2	The recommendations meet my expectations
3	The recommendations are sufficiently diversified
4	My feedback leads to better recommendations
5	The system provides sufficient means to express my expectations
6	I like the way I can set local constrains
7	I like the way I can set temporal constrains
8	I like the way I can choose certain genres
9	The user interfaces are intuitive
10	I like the presentation of new recommendations

Further personal questions were added to the survey in order to obtain an overview of the participant's background, personal experience with events and personal experience with similar recommender systems.

5.2 Participants

The participants were selected to achieve a cross-sectional survey of event visitors. Different age groups were considered as were casual visitors and so-called expert visitors who attend events on a regular basis. Even though the participants were obligated to own a smartphone, less technophilic participants were as important as more experienced users.

Twenty-one participants started the field study of whom 16 terminated successfully after two weeks by completing the survey. Of the 16 participants who completed the study, one was younger than 18 years old, three were between the ages of 18 and 25, nine were between 26 and 35, two were between 36 and 45 and one was older than 45 years old. The average technical affinity of the participants is 3.73 (σ: 0.88) on a 5-point Likert scale according to their own estimation. Mobile event applications were used by 62.5% of the participants once a month or less often and by 18.8% at least once a week. The participants reported different experiences with recommender systems such as the Amazon website. Recommender systems were used several times per week by 25% and 43.8% used them once per month or less often. About a third of the participants, 37.5%, can be called expert visitors as they attend events requiring a ticket purchase once per month or more often whereas 62.5% purchase tickets not more often than a few times per year.

5.3 Results

The responses (Figure 4) show that the participants are satisfied with our solution ($\varnothing$: 3.75, σ: 0.83). Only 25% of the participants rated the system with a 3 or less. We also wanted to know if the received recommendations met their expectations. The responses are slightly above average ($\varnothing$: 3.38, σ: 0.60). According to the survey results, the recommendations can be called sufficiently diversified ($\varnothing$: 3.63, σ: 1.05). The participants believe that their feedback improves the personalized recommendations ($\varnothing$: 3.69,

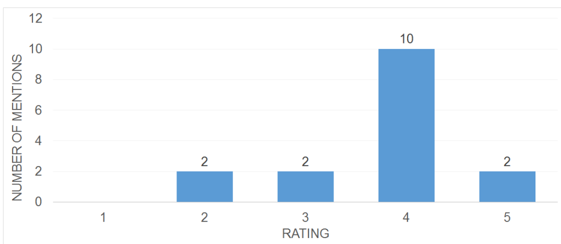

Figure 4: Overall satisfaction with the recommender system.

σ: 0.77). The frequency of usage differed between participants. The 16 participants would have bought a total of 16 tickets based on the recommendations which means one sold ticket per person during the two week study period. A majority of the participants, 87.5%, mentioned that they would like to continue using the system to find interesting events in the future.

In general, the users were satisfied with the choice of settings which allowed them to modify the context-awareness (∅: 3.5, σ: 0.79). They particularly liked how they were able to influence the geographical context (∅: 3.94, σ: 0.75). A higher variance of ratings can be observed when asking the participants about the calendar view (∅: 3.56, σ: 1.17) and the list of genres (∅: 3.44, σ: 1.12). A free text field at the end of the survey allowed the participants to add further thoughts. Some participants requested further options to modify the context-awareness, for example, the user's current budget for events.

The developed Android application is only one example of how our algorithm can be implemented in a mobile event recommender system. According to the participants, the apps' user interfaces are intuitive (∅: 4.06, σ: 0.97) and they were pleased with the chosen card layout for the new recommendations (∅: 4.07, σ: 0.77).

5.4 Discussion

Our field study allowed us to evaluate the recommendations generated by our context-aware CBCF algorithm in a realistic scenario. The results show that our approach to adapt CBCF is a promising way to generate personalized event recommendations. The user's feedback on past recommendations is a valuable basis to fill the user-item rating matrix which can be extended by a CB prediction. The CF algorithm based on this matrix is able to provide accurate recommendations and ensures a sufficient diversity of events. Furthermore, our system is able to provide recommendations even if only one user is using the system.

CBCF can be extended by context-aware recommendations, another important finding of the study. The recommendations do not only satisfy the search activities of the users, but the participants in our survey were also pleased they were able to influence the context-awareness. Nevertheless, our study reveals that further context factors should be identified and considered during the recommendation process. Our first prototype focuses on the most obvious context factors such as the user's current location or temporal context factors. In addition to the price of the event, further context factors which were completely ignored during this work should be incorporated in order to improve the quality of the recommendations in different situations. Examples of such factors are the current weather or if the user is alone or accompanied by someone. In this case, a group recommender could help to find suitable recommendations for all group members. If, for example, a fan of music events is accompanied by her or his children, recommending a child-friendly event will be more appropriate than a visit to a concert. While our first prototype mainly supports users in receiving recommendations at a certain time before the actual event takes place, we believe that additional, contextual information would also allow to find suitable recommendations even more spontaneously (Herzog and Woerndl, 2015).

According to the feedback we got from our participants, we learned that the users of mobile recommender systems value appealing user interfaces. The chosen card layout for new recommendations received positive feedback, confirming our belief that this layout is an appropriate way to present recommendations and to offer a quick and easy solution for providing feedback at the same time.

6 RELATED WORK

In comparison to movie recommendations, research has paid little attention to recommender systems for events. Minkov et al. (2010) present an approach which considers the individuals' preferences for past events and combines these preferences with other peoples' likes and dislikes. Dooms et al. (2011) conducted a user-centric evaluation of different event recommender algorithms. Results show that a hybrid of content-based and collaborative filtering performs better than other algorithms in most quality factors like accuracy, satisfaction and usefulness. The hybrid approach of Khrouf and Troncy (2013) combines content-based and collaborative filtering and is enriched by Linked Data to overcome data sparsity. Cornelis et al. (2005) model user and item similarities as fuzzy relations in their hybrid approach. Outlife is an event recommender which groups friends in social groups (De Pessemier et al., 2013). It allows the creation of recommendations for specific groups of friends and can recommend friends who should be invited to an event. Zhang et al. (2013) present a group recommender which recommends event-based groups using matrix factorization. A study using *Meetup.com* data shows the effectiveness of their approach. Quercia et al. (2010) analyzed mobile phone location data to understand how the user's current location influences the acceptances of social event recommendations. They found out that recommending events that are popular among residents of an area is more beneficial than recommending nearby events. This finding can be seen as a promising solution for the *Cold-Start Problem* in event recommender system. If the event is location independent, Daly and Geyer (2011) suggest using the popularity as a metric for overcoming the *Cold-Start Problem*. A special type of events that can be recommended are distributed events. These events are collections of smaller, single but very similar events that occur at the same day (Schaller et al., 2013).

A few mobile applications offering personalized event recommendations are already on the market. One example is Bandsintown, an application available for Android and iOS devices which focuses on music events (Figure 5). Recommendations made for events take into consideration music selections locally stored on the user's device. Furthermore, external sources like Facebook or Twitter can be connected to the application in order to collect additional information about user preferences in regard to music. Other examples of mobile applications which offer some kinds of event recommendations are the EVENTIM DE application, offered by the German ticketing and event company CTS Eventim and XING EVENTS, an application of the German based social network for professionals XING. The fact that the ticketing and event industry has already started to implement personalized recommendations in their offers underlines the significance of the topic for the different players involved in this industry. Visitors, on the one hand, can find interesting events nearby which they might overlook otherwise. On the other hand, the recommendations promise additional sales and the users' feedback allows organizers to predict the number of future visitors for similar events (Minkov et al., 2010).

Setten et al. (2004) focus on context-aware systems. The mobile application COMPASS demonstrates how user interests and context can be respected in a recommender systems. Oku et al. (2006) show that recommendations which take the user's context into consideration lead to a higher user satisfaction. A general model for context-aware and proactive recommending is delivered by Woerndl et al. (2011). Adomavicius and Tuzhilin (2011) emphasize the importance of context in recommender systems and introduce three different paradigms for incorporating contextual information into the recommendation process.

The related work shows the potential of event recommender systems and the advantages of context-awareness. But to the best of our knowledge, there is no work examining the use of CBCF within a

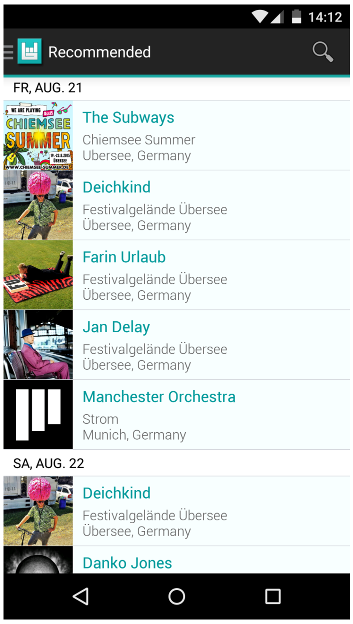

Figure 5: Bandsintown Android application.

context-aware recommender to recommend all kinds of events. We closed this gap by showing that CBCF can be extended by context-aware recommendations and adapted to the scenario of event recommendations. Our results show that the chosen approach is able to deliver promising recommendations.

7 CONCLUSION AND FUTURE WORK

In this paper, we introduced a new method for personalized event recommendations. We developed an algorithm based on CBCF and extended it by context-aware recommendations. This algorithm was implemented in an Android application and evaluated in a two-week field study. According to the results, our approach delivers promising recommendations. By combining different recommendation techniques, certain limitations, such as the *New Item Problem* and a lack of diversity, can be overcome. We believe that the results of the field study emphasize the economic potential of the developed system and underline the importance of mobile recommender systems in the ticketing and event industry. The first applications of recommender systems that provide some kind of personalized recommendations are already available and are steadily gaining popularity.

Future work should focus on identifying additional context factors to be considered during the algorithm execution. These factors should be capable of finding suitable recommendations in different situations, e.g., for tourists who spend only a very short time visiting a city. Improvements are also possible in regards to other solutions of context incorporation. Currently, we use pre-filtering only, but post-filtering and contextual modeling should be considered as well in future work. The contextual pre-filtering immediately excludes inappropriate events but we should examine if users are willing to accept little violations of the thresholds if an event really satisfies their needs. Furthermore, social recommendations promise an improvement of the recommendations. Events of particular importance for the user, e.g., events the user liked on social networks like Facebook, could receive a bonus on the calculated rating to increase the probability of recommendation. Another aspect is implicit user feedback which was not collected by this first prototype but could be used to improve the recommendations. Our first prototype or pure CB or pure CF approaches could be used as a baseline application in order to evaluate the suggested extensions and their influence on the quality of the recommendations. For this purpose, we want to conduct another study with a larger number of users which allows us to obtain more in-depth user feedback and statistics.

ACKNOWLEDGEMENTS

This work is part of the TUM Living Lab Connected Mobility (TUM LLCM) project and has been funded by the Bayerisches Staatsministerium für Wirtschaft und Medien, Energie und Technologie (StMWi).

REFERENCES

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749.

Adomavicius, G. and Tuzhilin, A. (2011). Context-aware recommender systems. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 217–253. Springer US.

Balabanović, M. and Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370.

Cornelis, C., Guo, X., Lu, J., and Zhang, G. (2005). A fuzzy relational approach to event recommendation. In Prasad, B., editor, *IICAI*, pages 2231–2242. IICAI.

Daly, E. M. and Geyer, W. (2011). Effective event discovery: Using location and social information for scoping event recommendations. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 277–280, New York, NY, USA. ACM.

De Pessemier, T., Minnaert, J., Vanhecke, K., Dooms, S., and Martens, L.(2013). Social recommendations for events. In *5th ACM RecSys Workshop on Recommender Systems and the Social Web*.

Dey, A. K., Abowd, G. D., and Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Hum.-Comput. Interact.*, 16(2):97–166.

Dooms, S., De Pessemier, T., and Martens, L. (2011). A user-centric evaluation of recommender algorithms for an event recommendation system. In *Proceedings of the RecSys 2011 : Workshop on Human Decision Making in Recommender Systems (Decisions@RecSys'11) and User-Centric Evaluation of Recommender Systems and Their Interfaces - 2 (UCERSTI 2) affiliated with the 5th ACM Conference on Recommender Systems (RecSys 2011)*, pages 67–73.

Herzog, D. and Woerndl, W. (2015). Spontaneous event recommendations on the go. In *Proceedings of the 2nd International Workshop on Decision Making and Recommender Systems (DMRS2015), Bolzano, Italy, October 22-23, 2015.*

Khrouf, H. and Troncy, R. (2013). Hybrid event recommendation using linked data and user diversity. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 185–192, New York, NY, USA. ACM.

Melville, P., Mooney, R. J., and Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In *Eighteenth National Conference on Artificial Intelligence*, pages 187–192, Menlo Park, CA, USA. American Association for Artificial Intelligence.

Minkov, E., Charrow, B., Ledlie, J., Teller, S., and Jaakkola, T. (2010). Collaborative future event recommendation. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 819–828, New York, NY, USA. ACM.

Oku, K., Nakajima, S., Miyazaki, J., and Uemura, S. (2006). Context-aware svm for context-dependent information recommendation. In *Proceedings of the 7th International Conference on Mobile Data Management*, MDM '06, pages 109–, Washington, DC, USA. IEEE Computer Society.

Pazzani, M. J. and Billsus, D. (2007). Content-based recommendation systems. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The Adaptive Web*, pages 325–341. Springer-Verlag, Berlin, Heidelberg.

Quercia, D., Lathia, N., Calabrese, F., Di Lorenzo, G., and Crowcroft, J. (2010). Recommending social events from mobile phone location data. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 971–976, Washington, DC, USA. IEEE Computer Society.

Ricci, F. (2011). Mobile recommender systems. *Information Technology & Tourism*, 12(3):205–231.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA. ACM.

Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The Adaptive Web*, pages 291–324. Springer-Verlag, Berlin, Heidelberg.

Schaller, R., Harvey, M., and Elsweiler, D. (2013). Recsys for distributed events: Investigating the influence of recommendations on visitor plans. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 953–956, New York, USA. ACM.

Setten, M., Pokraev, S., and Koolwaaij, J. (2004). Context-aware recommendations in the mobile tourist application compass. In Bra, P. and Nejdl, W., editors, *Adaptive Hypermedia and Adaptive Web-Based Systems*, volume 3137 of *Lecture Notes in Computer Science*, chapter 27, pages 235–244. Springer Berlin Heidelberg, Berlin, Heidelberg.

Smyth, B. (2007). Case-based recommendation. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The Adaptive Web*, pages 342–376. Springer-Verlag, Berlin, Heidelberg.

Torkington, J. (2014). Small data: Why tinder-like apps are the way of the future. Retrieved August 13, 2015 from https://medium.com/@janel_az/small-data-why-tinder-like-apps-are-the-way-of-the-future-1a4d5703b4b.

Woerndl, W., Huebner, J., Bader, R., and Gallego-Vico, D. (2011). A model for proactivity in mobile, context-aware recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 273–276, New York, USA. ACM.

Zhang, W., Wang, J., and Feng, W. (2013). Combining latent factor model with location features for event-based group recommendation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 910–918, New York, USA. ACM.