Introducing Context into Recommender Systems

Wolfgang Woerndl, Johann Schlichter

Technische Universitaet Muenchen
Boltzmannstr. 3
85748 Garching, Germany
{woerndl,schlicht}@in.tum.de

Abstract

In this paper, we give an overview of our work to investigate the integration of context into different kind of recommender systems. Context adds an additional another dimension to the user-item data model of recommender system and can be utilized in different ways during content-based or collaborative recommendation processes. We give several application examples we are working on to apply contextual recommenders in real world scenarios.

1 Motivation

Personalization and recommender systems can potentially reduce the omnipresent information overload in our networked world. Another promising and possibly complementary approach is to utilize context, but this has been rarely applied in personalization systems so far, according to a recent survey by Anand and Mobasher (2005). (Adomavicius et.al. 2005) argue for considering contextual information using multidimensionality of recommendations as a possible extension of current recommender systems. However, contextualized recommender systems are a rather novel and unexplored research and application area.

Thus, the goal of this work is to investigate how to integrate context into different kind of recommender systems in various application domains. Therefore, the focus is on the applicability of contextual recommenders. The rest of this paper is organised as follows. Section 2 discusses the introduction of context into recommender systems in theory; Section 3 outlines several application domains where we are applying contextualized recommenders at the moment. The paper finally concludes with a brief summary.

2 Context in Data Model and Recommendation Process

Recommender systems traditionally operate on a user-item matrix. Each user may be described by a vector of attributes, which is also called user profile. It is necessary to distinguish between profile and context information, some differences are shown in Table 1. Items can have associated meta data, for example the location, price range and opening hours of a restaurant in a tourist guide. Collaborative filtering (CF) algorithms relies on ratings of items by (other) users, for example a grade on how much a particular user liked a given point-of-interest, while content-based approaches matches the meta data of items with the user profile of the active user without taking other users into account.

User profile	Context
rather static and somewhat longer lasting	highly dynamic and transient
stored in user profile	not stored permanently
implicitly observed or explicitly provided by user	observed only, never manually entered by user
Example: user preferences or interests	Example: current location and time

Table 1. User profile vs. context

The integration of context into recommender systems adds an additional dimension of complexity to the recommender data model because ratings may be valid in one particular context only. Context can be described by a vector of context attributes, e.g. time, location or currently available network bandwidth in a mobile scenario. The actually used context attributes are largely dependent on the application domain.

Several principal approaches can be identified to introduce context in different types of recommenders. In a content (or knowledge-based) approach, rules could be used to capture preferences or restrictions arising from context constraints. The problem is to identify relevant rules that can be applied to filter items.

In collaborative filtering, the idea is to associate ratings with a snapshot of the context when the rating is made. (Adomavicius et.al. 2005) applies a reduction-based approach which eliminates “out of context” ratings. (Chen 2005) uses context to weigh rating according to context similarity. Thereby, a major problem is sparseness of available ratings in the same (or comparable) contexts.

Another approach is to combine different kind of recommenders to reduce the complexity of the item-user-context matrix by applying a cascading hybrid recommender. This means, first only two dimensions of the user-item-context matrix are analyzed, and in a second step the third dimension is considered in addition:

1. Use content- or knowledge based filtering to find relevant items based on context, for example taking the current end user device and location into account
2. Apply collaborative filtering to rank and additionally filter the result set from step 1

This can be done vice versa, i.e. first using collaborative filtering to generate initial items – without considering context –, and then using the knowledge base of applications to figure out which items are actually relevant in the current context. We have experimented with this cascading hybrid approach in recommending mobile applications and will present this example in the next chapter, along with other application areas we are currently working on.

3 Application Areas

3.1 Recommending Mobile Applications

Project Background. This recommender is integrated in a framework supporting the development of mobile applications. Part of the framework is a deployment server where developers of mobile applications can register their services and end users can browse and search for relevant and interesting gadgets. Users can access the deployment server and download client modules on their mobile devices. One problem for users is to find interesting and – with regard to their current context – relevant applications.

Recommender System. The hybrid recommender system developed in this scenario recommends mobile applications to users derived from what other users have installed in a similar context (location, currently used type of device, etc.) (Woerndl, Schueller, and Wojtech 2007). Users can choose between several content-based and collaborative filtering components:

- LocationAppRecommender: recommends applications that were used in a similar location by other users
- CFAppRecommender: apply existing collaborative filtering algorithms to generate results using the *Taste* library (Lemire and Maclachlan, 2005)
- PoiAppRecommender

The PoiAppRecommender does not recommend point-of-interests (POIs) but recommends mobile applications based on POIs in the vicinity of the user using triggers. An administrator can select among types of point-of-interests (such as restaurant, museum or train station) and specify within which radius of an actual POI an application is recommended. This is done when registering the application with the mentioned deployment server.

When making a recommendation, the system then retrieves the current user position (using a GPS-enabled mobile device), determines POIs in the vicinity and generates a recommendation based on this context information. For example, an administrator can specify that her mobile train table application shall be recommended when the user is near a train station. After applying the trigger rules, our approach uses collaborative filtering to rank found items according to user ratings of applications in a second step. User ratings are collected implicitly by automatically recording when a user installs an application within our framework. It is also optionally possible for users to explicitly rate applications after usage. The ratings are stored together with context information (time, location, used device, …) to capture the situation when a rating was made.

We have designed and implemented the recommender in the explained framework for the development of mobile applications. We are currently improving the components of our framework, developing more real world applications with student teams and then testing the system in practice.

3.2 A Gas Station Recommender for Internet-Worked Cars

Project Background. One current research topic is ad-hoc networking of cars, for example carried out in the Network-on-Wheels project (NoW, http://www.network-on-wheels.de). The ultimate goal is to widen drivers' horizons and improving driving safety. An example is to automatically detect icy road conditions and relay this information to other cars in the vicinity. In addition, another aim is to utilize this car-to-car infrastructure to provide useful and entertaining information to passengers. This is also important with regard to achieving a higher level of market penetration that is necessary for active safety car-to-car applications.

Recommender System. We are currently designing and implementing a context-aware recommender for internet-worked cars in the NoW project (Woerndl and Eigner, 2007). Thereby, we are using the NoW communication infrastructure to exchange information between cars and also between a car and a hotspot. In addition, NoW provides context information about the car, that is normally not available.

The specific application scenario is a gas station recommender that selects possible gas stations within the current fuel range along a route. The context information such as current location of the car or fuel level can be retrieved via NoW interfaces. Prices of gas stations – or, more generally, meta data about items – can be provided by existing databases or entered by drivers and propagated from car to car. In the user profile, user preferences such as preferred gas station chains and also (static) information about the car are managed, e.g. the required fuel type of the car.

The proposed recommender process works in 2 steps:

1. Determine all gas stations that have to be considered, by analyzing the route, heading direction, required fuel type and remaining range.
2. Rank the available gas stations from step 1 according to price, user preferences for gas station chains and whether a detour is necessary or not (dependent on this user profile attribute)

We are currently designing the details and implementing the gas station recommender. This application is rather simple, but we are also investigating additional potential uses for contextual recommenders in this area of internet-worked cars. One other area is location-aware community support among cars resp. drivers. For example, information about parking or traffic jams can be exchanged.

3.3 Social Context in Collaborative Filtering

Project Background. Exploitation and visualization of social networks, and platforms to detect and manage relationships among users are becoming more and more popular. Examples include myspace.com, XING (formerly Open BC) or the German Web site http://www.lokalisten.de. (Ehmig 2007) has extracted social network data from this platform and asked users to submit ratings for Munich clubs as an example (about 1000 users and 100 clubs). In this scenario, we are investigating how to improve collaborative filtering algorithms by taking the social context of users into account.

Recommender System. Ehmig (2007) has realized a "social recommender system" to predict how much a user possibly likes a given club. Thereby, the similarity of users is not computed by analyzing user ratings, but on the basis of social networks of users. Early evaluation results based on the mentioned data set indicate that the social recommender outperforms standard collaborative filtering algorithm in this scenario.

We are currently working on combining the social recommender with standard algorithms for user similarity computation. The basic idea is to apply a higher weighing to (positive or negative) ratings of items when a high context similarity can be computed. In our case, the context is the overlap in the social networks of users. In the simplest case, the social network is a list of buddies that are managed in social network platforms. The social context is not only used for forming a neighbourhood of similar users in collaborative filtering, but also to predict ratings for an active user. To take changing social contexts into considerations, we determine the current social network of a user and store this information when a rating is made. Hence, the proposed social recommender is also an exploitation of a user-item-context matrix.

4 Conclusion

In this paper, we have outlined our work to integrate context into recommender systems. We have explained to consider context as separate entity in the recommender data model, not as part of user or item descriptions. We are currently refining the used recommender systems and evaluating our ideas in the explained application areas. One possible benefit of this approach is transferability of ideas to additional domains. Contextual recommenders can possibly be applied in dynamic scenarios where the relevance of items to users is dependent on rapidly changing environments.

One problem for evaluating contextual recommender systems is the availability of relevant data sets. Existing data sets (e.g. from the GroupLens project, http://www.grouplens.org/) are not associated with contextual information. Therefore, we use a rather practical approach to gather data and gain experiences in our real world scenarios.

References

Adomavicius, G., Sankaranarayanan, R., Sen, S., and Tuzhilin, A., 2005. "Incoporating contextual information in recommender system using a multidimensional approach". *ACM Transactions on Information Systems*, Vol. 23, No. 1, pp. 103-145

Anand, S.S., and Mobasher, B., 2005. "Intelligent Techniques for Web Personalization". In *ITWP 2003*, Springer LNAI 3169, Berlin/Heidelberg, pp. 1-36

Chen, A., 2005. "Context-Aware Collaborative Filtering System", *Proc. of the International Workshop on Location and Context-Awareness 2005*, Springer LNCS 3479, Berlin/Heidelberg, pp. 244-253

Ehmig, C., 2007. *Improving Recommender Systems via Social Networks*. Diploma Thesis, Department of Computer Science, TU Muenchen, Munich, Germany

Lemire, D., and Maclachlan, A., 2005. "Slope One Predictors for Online Rating-Based Collaborative Filtering". In Proc. *SIAM Data Mining (SDM '05)*

Woerndl, W., Schueller, C., and Wojtech, R., 2007. "A Hybrid Recommender System for Context-aware Recommendations of Mobile Applications". In Proc. *IEEE 3rd International Workshop on Web Personalisation, Recommender Systems and Intelligent User Interfaces (WPRSIUI'07)*, Istanbul, Turkey

Woerndl, W., and Eigner, R., 2007. "Collaborative, Context-Aware Applications for Inter-Networked Cars". In *Proc. 5th International Workshop on Distributed and Mobile Collaboration (DMC 2007)*, Paris, France