Context-Aware Collaborative Filtering System: Predicting the User's Preferences in Ubiquitous Computing

Annie Chen
IBM Zurich Research Laboratory
Säumerstrasse 4, CH-8803 Rüschlikon, Switzerland
ach@zurich.ibm.com

ABSTRACT
In this paper I propose a context-aware collaborative filtering system that can predict a user's preference in different context situations based on past user-experiences. The system uses what other like-minded users have done in similar context to predict a user's preference towards an item in the current context.

Author Keywords
Context-aware; collaborative filtering; recommendation system; ubiquitous computing

ACM Classification Keywords
H5.3. Group and Organization Interfaces: Collaborative computing.

INTRODUCTION
In the ubiquitous computing world, computing devices are part of the bigger environment, known as the *pervasive context*. These devices could be aware of various aspects of contexts in the environment, such as the location, surroundings, or even the weather forecast. Computers in this environment have become more like that of a personal assistant than, say, of a help-desk. This shift in interaction prompted me to look at how ubiquitous devices could assist users better by anticipating their preferences in a dynamic environment.

Currently most applications for ubiquitous computing rely on manually defined rules to determine application behavior for different context. These rules can either be predefined by application developers, or alternatively user-configured either by static preferences or formed over time from user feedback [3, 4]. Static rules are inflexible and difficult to customize for individuals, whereas the underlying learning process in the latter case has a long learning curve. More importantly, these systems are unable to accurately predict a user's preference for an unseen situation.

Collaborative Filtering (CF) is a technology that has emerged in e-Commerce applications to produce personalized recommendations for users. CF works by combining the opinions of people who have expressed inclinations similar to yours in the past to make a prediction on what may be of interest to you now [2]. One well-known example of a CF system is Amazon.com. An implementation of such CF system could build user profiles from feedback on items made over time. It then finds like-minded users by "weighing" the active user against every other user with respect to the similarity in their ratings given to the same items. All the neighboring like-minded users' ratings are then combined into a prediction by computing a weighted average of the ratings.

Thus far, CF has mostly been applied to applications for which the context is static, hence the recommendations do not change. In the dynamic environment of ubiquitous computing, users' decisions can be influenced by many things in their surrounding context. For instance, when people travel, their preferred activities may largely depend on the weather. Existing CF systems could not model this complexity of context. They are as likely to recommend mountain routes for a person who likes hiking whether it rains or shines.

I propose a context-aware CF system that can predict a user's preference in any context environment by leveraging past experiences of like-minded users in similar context. In the sections to follow, I will propose algorithmic extensions to the CF process that incorporate pervasive context in making predictions.

PROPOSED SOLUTION
Introducing Context
Context is a description of the situation and the environment a device or a user is in. Schmidt et al. [1] categorized context into six high-level subspaces. Three relate to human factors: information about the user, social environment, and user's tasks. The other three concern the physical environment: Location, infrastructure, and physical conditions.

To model context in a CF system, a user's choice or preference needs to be associated with the context in which the user made that choice. This means that the current context needs to be captured each time the user makes a choice. The same applies for the reciprocal: when a user asks for recommendations, we need to capture the current context and evaluate what others have chosen in a similar context in the past. This poses two main problems: how to manage context in the user profile in terms of data modeling and storage, and how to measure similarities between contexts.

Context Modeling in CF
In a standard CF system, a user's profile is made up of a set of items with at most one rating assigned to each item. In

CHI 2005, April 2–7, 2005, Portland, Oregon, USA.
ACM 1-59593-002-7/05/0004.

a dynamic environment, a user's preference towards an item may change with the context. To capture this, a snapshot of the context need to be stored along with each user-rating.

A snapshot of context is a composite of different types of context data from various sources. This can be acquired from either the embedded sensors in the mobile device itself, or a smart environment. Consequently, various context data can be available or not, depending on what is accessible in the current environment. This yields the requirement that different context types should be managed independently, and their combined impact be calculated algorithmically.

Context Similarity

The goal of calculating context similarity is to determine which ratings are more relevant for the current context. For instance, when Bob wants to go fishing in spring, ratings of fishing locations in spring would be more relevant than ratings of fishing locations in autumn. The similarity of the context in which an item is rated with the current context of the active user determines the relevance of this rating. Consequently, for each context type there needs to be a *quantifiable measure* of the similarity between two context values.

Context types can vary widely and it would be difficult to manually define a similarity function for each context type. Therefore, I devised an automated method to compare the relevance of one context value to another for the same context type. I make the assumption that if user preferences towards an item (e.g., fishing) do not differ much in different contexts, then the ratings given in one context would also apply for the other. So if the ratings for an item are similar for two different context values, then these two values are very relevant to each other.

Similarity can then be calculated by measuring the correlation between two different context variables with respect to their ratings for an item. For this I used *Pearson's correlation coefficient*, which measures the degree of linear components of relationship between two variables, also commonly used in CF to measure user similarity. I denote the rating given by the user u on item i in context x as $r_{u,i,\mathsf{x}}$, and formulate the similarity weight for two different context variables, x and y for item i as follows:

$$sim_t(\mathsf{x},\mathsf{y},i) = \frac{\sum_{u=1}^{n} (r_{u,i,\mathsf{x}_t} - \bar{r}_i) \cdot (r_{u,i,\mathsf{y}_t} - \bar{r}_i)}{\sigma_{\mathsf{x}_t} \cdot \sigma_{\mathsf{y}_t}}, \quad (1)$$

this returns the *relevance* of two context values in a context dimension C_t over all the ratings users gave in these context.

Incorporating Context into Prediction

In the normal CF process, a prediction is calculated by combining the neighbors' ratings into a weighted average of the ratings, using the neighbors' similarities as weights. In the context-aware CF system, each rating has an associated context. The similarity between the rating's context with the active user's context determines how relevant this rating is, so it must be incorporated into the weight.

I define $R_{u,i,\mathsf{c}}$ as the weighted rating for the user u on an item i in context c, where c is the current context of the active user. This rating is weighted as shown in Eq. 2 with respect to the similarity between context x in which the rating r was given and the context c of the active user. The context is multi-dimensional so I assume linear independence and calculate the similarity for each dimension separately, i.e.,

$$R_{u,i,\mathsf{c}} = k \sum_{\mathsf{x} \in \mathsf{C}} \sum_{t=1}^{z} r_{u,i,\mathsf{x}} \cdot sim_t(\mathsf{c},\mathsf{x},i), \quad (2)$$

where k is a normalizing factor such that the absolute values of the weights sum to unity. It has nested sums: the inner loops over each dimension in context, e.g., Location, Weather; the outer loops over all the values in that dimension, e.g., "Zurich", "Prague" for the Location context.

Now Eq. 2 can be used as the context sensitive version of user's rating to generate prediction, i.e.,

$$p_{a,i,\mathsf{c}} = \bar{r}_a + k \sum_{u=1}^{n} (R_{u,i,\mathsf{c}} - \bar{r}_u) \cdot w_{a,u}, \quad (3)$$

where $p_{a,i,\mathsf{c}}$ is the predicted rating of active user a in context c for item i. This calculation combines all the weighted ratings, with respect to similarity in context, of all the neighbors, which is then again weighted with respect to the similarity of user, to give an overall prediction for the active user on an item in the current context.

FUTURE WORK

I am currently developing a tourist application for mobile phones to exercise and collect data for the context-aware CF engine. I plan to collaborate with IBM's Context Weaver group to use their context middle-ware for data acquisition. I plan to trial this application on a group of summer students in the lab, who will use the application on their weekend travels. The data collected from this deployment will allow us to evaluate and calibrate our algorithms.

REFERENCES

1. Albrecht Schmidt, Michael Beigl, and Hans-W. Gellersen. There is more to context than location. *Computers and Graphics*, 23(6):893–901, 1999.

2. John S. Breese, David Heckerman, and Carl Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 43–52, July 1998.

3. G. Petrone L. Ardissono, A. Goy. INTRIGUE: personalized recommendation of tourist attractions for desktop and handset devices. *Applied Artificial Intelligence*, 17(8-9):687–714, 2003.

4. Mark van Setten, Stanislav Pokraev, and Johan Koolwaaij. Context-Aware Recommendations in the Mobile Tourist Application COMPASS. In *Adaptive Hypermedia 2004*, volume 3137 of *LNCS*, pages 235–244, August 2004.