When Mobile Commerce Embraces User Contexts

1 Cenzech Zhu, 2 Teng Tiow Tay
1 Research Student, Department of Electrical and Computer Engineering, NUS, Singapore
2 Assoc. Prof., Department of Electrical and Computer Engineering, NUS, Singapore
1 cenzhe_zhu@nus.edu.sg, 2 etietyt@nus.edu.sg

ABSTRACT

Mobile commerce has emerged as the ubiquity of smart phones and ultra fast mobile data network. While much of the work done in the field of mobile commerce has focused on the customer behavior, business model, and wireless infrastructures, this paper proposes a context-aware solution to mobile commerce. Context-awareness can improve customer’s shopping experience, and thus is crucial in advocating mobile commerce. In this paper, an entertainment recommendation system is described. Special treatment of context information storage and its usage in recommendation systems are analyzed. The context-aware collaborative filtering algorithm (CCF) proposed in the paper is tested to produce better performance as compared to a traditional context-enabled CF method. A prototype of the system is built on the domain of music recommendation and it’s well received by selected users.

Keywords: Mobile commerce, context-aware, collaborative filtering

1. INTRODUCTION

According to IDC Financial Insights’ 2012 Consumer Payments Survey, 34 percent of survey respondents have made a purchase using their mobile phone compared to 19 percent a year ago. This clearly shows the emergence of mobile commerce. The report also found that physical goods were the most common mobile purchase, with more than 70 percent having purchased a physical good. 60 percent have purchased online services and digital goods instead.

Many technical aspects of mobile commerce research have reached commercializing quality. The works that remain are largely to steer the shift of user purchase habit from stationary computers to mobile terminals, and to improve users’ experience when doing such purchases to accelerate this shift. Context-aware applications, being first introduced to improve user experience, are the natural choice for further advocating mobile commerce.

However, existing research on mobile commerce mostly solves problems like behavior, business model, wireless infrastructure, etc. How to take context information into account is seldom considered. This paper mainly looks into two aspects of incorporating context into mobile commerce: How to capture and represent context information, and how the context information is used to provide better services. The “services” mentioned here specifically mean recommending users to the most relevant product in the current situation.

The scenario in this paper is set as follows: Alice is on her way back home after one day’s work. The subway to home is taking approximately 1 hour so she decided to have some fun on the phone to kill time. When she refers to our application, the system detects Alice has following contexts: She’s leading a frugal life style, though purchasing is OK, it’s limited to 1 dollar (from previous purchase history or inputted personal profile); She’s on her way back home and the time is estimated to be 1 hour (from current GPS position and pre-defined home/work location); She’s got no company so the entertainment should be for a single person (from the observation that no friend’s device is nearby); Though the earplug is on, ambient noise level is high (from earplug detection and speaker sampling); Alice is fond of reading verse, listening to symphonies and playing social network games (from personal profile).

An ideal system should make an overall evaluation of the user’s current situation, and finally comes to the decision to recommend a little social network game Alice’s friends are playing online. But building such a system from the expert system approach is difficult. The sheer number of recommendation rules can overburden the developers. In this paper, we propose to build such a system using Collaborative Filtering (CF) techniques that are specially tuned to consider context information. Using CF techniques, we assume Alice choice will be similar to like-minded users’ choices, so no explicit rules are required.

The contribution of this paper includes:
- We proposed a novel system that captures and manages context information to be used by recommendation algorithms.
- We have formulated a context-aware collaborative filtering algorithm (CCF). This algorithm managed to solve the recommendation problem in 2-D space instead of higher. Context information is considered quantitatively rather than qualitatively.

The following texts are organized as such: Section 2 briefly reads the background of this area of research. Section 3 focuses on the context information gathering and distribution. The context-aware collaborative filtering algorithm is explained in detail in Section 4. Section 5 evaluates both the algorithm and the prototype system and we conclude in Section 6.
2. RELATED WORK

Mobile commerce emerges around 2000[1]. It can be viewed as a subset of E-commerce and is usually defined as “any transaction with monetary value that is conducted via a mobile network” [2]. Reference [3] surveys the features and characteristics of current smart phones, putting an emphasis on the required and desirable features for mobile commerce. Sumita [4] provides a mathematical model for comparing e-commerce via the traditional PC access with m-commerce which accommodates both the traditional and mobile access. However, there is no specific work on utilizing context-awareness in m-commerce settings.

As a starting point for our context-aware recommendation algorithm, the legacy recommendation algorithm for E-commerce should be investigated. Existing recommendation algorithms can be classified in many categories: content-based methods [5] seek to discover co-purchased products and return top-N recommendations; collaborative filtering (CF), being the most successful recommender technology to date, recommends products that are similar to the ones that the target customer has purchased. This similarity can be calculated in many ways, and thus classifies the family of CF algorithms into user-based CF [6] and item-based CF [7]. Some of the new implementations of these recommendations include [8], [9]. It is believed that item-based CF algorithms generally provide better scalability and higher accuracy in giving recommendations. Our algorithm is rooted in item-based collaborative filtering.

3. SYSTEM OVERVIEW

In this section we explain details of the proposed system, but leave the recommendation algorithm in next section.

From the perspective of data structure, this system comprises two databases. The first one is the rating table, storing all user ratings as well as user contexts. This is a 2-D table with rows corresponding to users and columns corresponding to items/contexts. The second database is ontology-based, storing the profiles of users, properties of the items as well as their relationships. To link these two databases, a translation table is constructed to translate ontology objects to row/column number and vice versa. The ontology-based database is constructed because of ontology’s power to do reasoning. When the rating table is sparse, meaning not many people have rated items, we can depend on the ontology reasoning to provide satisfying results.

Context information gathered can be either physical sensor data, or user profile.

Modern smart phones have evolved into something that is much more than a telecommunication tool. Many sensors are embedded in the smart phones, and this number is still increasing. Smart phone sensors can detect ambient noise level, ambient light level, moving speed, turbulence level, GPS location, etc. These sensor data can be used to reproduce the physical environment of the user in the virtual world. The richness of user context enables high quality of artificial intelligence in our system. For example, in a noisy and trembling subway cabinet, recommending a tranquil novel wouldn’t be a good choice even if the user is generally fond of reading that genre.

Besides the physical context of the user, the profile of the user can also be detected. With the user’s permission, our system can gain access to the user’s social network sites and screen her friend list and/or historical posts. Using information retrieval techniques, this inspection can provide the recommender system with more knowledge about the user’s preference. User preferences together with user’s purchase history in E-commerce sites constitute the context of this user in a longer time frame when compared with the context gained by sensor data interpretation. Moreover, statistics show that most people’s purchase decisions are suggested by friends, because friends’ recommendation has the highest trust level. The area of social commerce has been investigating this phenomenon for long. Our system embraces this feature whenever it is possible to retrieve a friend’s profile.

4. RECOMMENDATION ALGORITHM

Our system uses a modified item-based collaborative filtering algorithm for giving predictions and recommendations, called Context-aware Collaborative Filtering (CCF). Unlike MD [10] and RST [11], our approach managed to limit the dimensionality of the algorithm in 2-D space.

4.1 Context-aware Collaborative Filtering

The goal of a CF algorithm is to predict the utility of a certain item for a particular user based on the preference (both explicit and implicit) of the target user and other like-minded users. Usually an m x n rating matrix $R$ is employed to represent all the user-item data. Each entry of the matrix $R_{ij}$ in $R$ represents the rating of the ith user on the jth item. Ratings are in a numerical scale indicating the preference of the user. A typical application uses 1 to 5 to denote lowest preference to highest preference, and 0 is used to represent that item is not yet rated by the user.

A multi-dimensional rating matrix that incorporates context information is denoted by $R_{c}^{r_{context}}$, where $c$ is the number of context information types. CCF unfolds the third dimension and has a rating matrix denoted by $R_{m \times n \times c}^{'},$ where $m'$ is the unfolded user number. For example, a user may have rated on two movies under different context sets. By duplicating the user, these two ratings are projected to the 2-D space as 2 rows. The first/second row represents the rating and contexts for the first/second movie, both given by the same user.

In a traditional context-aware collaborative filtering algorithm, the third dimension (context
information) is simply used as a filter condition. Only if the value of context information exceeds some arbitrarily-set threshold, the rating as a whole can be considered in later predictions. It’s hard to determine the threshold and this scheme restricts the usage of context information. In our approach, context information is treated as a special class of items. The columns of the matrix can be divided into normal items and context items. The meaning of the ratings in this matrix is also augmented, and is thus referred to as extended ratings. With extended ratings, the kernel of CF algorithm can remain largely intact while other modifications of CF algorithm involve a quantification of context information, and an extra weighting scheme.

Quantification of context information converts context information of various formats into the same format as ratings. Context information that is on a continuous scale is quantified to its nearest integer between 1 and 5. Binary or Boolean context information is either grounded to 1 or raised to 5. Some of the context information may have discrete value and the size of the range is greater than 5. They are restructured as a series of ratings, or more mathematically, a vector of the same length as the size of its range. In this vector, the field that corresponds to the currently active value will be set to 5 and all others are set to 1.

The weighting scheme in our approach endows extra significance to context items than normal items. In typical CF algorithms, the last step is usually a weighted sum prediction:

\[
P_{u,i} = \frac{\sum_{N_{i} \in N} (s_{i,N_{j}} * R_{u,N_{j}})}{\sum_{N_{i} \in N} (|s_{i,N_{j}}|)}
\]

where \(P_{u,i}\) is the prediction of user u’s opinion towards target item i, \(N\) is the set of items that are similar to item i, \(s_{i,N_{j}}\) is the similarity between item i and item \(N_{j}\), and \(R_{u,N_{j}}\) is user u’s rating on item \(N_{j}\). The similarity measure used in our system is correlation-based:

\[
s_{i,j} = \frac{\sum_{u \in U} (R_{u,i} - \overline{R})(R_{u,j} - \overline{R})}{\sqrt{\sum_{u \in U} (R_{u,i} - \overline{R})^2} \sqrt{\sum_{u \in U} (R_{u,j} - \overline{R})^2}}
\]

where \(U\) is the set of users who both rated item i and item j.

Equation (1) is a weighted sum of ratings, which is then normalized to scale. In our system, context items are assigned a higher weight as shown below:

\[
P_{u,i} = \frac{\sum_{N_{j} \in N} s_{i,N_{j}} R_{u,N_{j}} + \sum_{C_{j} \in C} w_{u,C_{j}} s_{i,C_{j}} R_{u,C_{j}}}{\sum_{N_{j} \in N} (|s_{i,N_{j}}|) + \sum_{C_{j} \in C} (|w_{u,C_{j}} s_{i,C_{j}}|)}
\]

where \(C\) is the set of context items. Note that \(R_{u,N_{j}}\) can be read from the multiple rows in the rating matrix because there are multiple rows that correspond to the user u. The parameter \(w_{u,C_{j}}\) is determined by a learning process. It represents the level of fastidiouslyness of user u on context \(C_{j}\). When it is set to 0, it means the context item is completely irrelevant to this user. When given a value greater than 1, the context item is treated with escalated importance. Note that this set of parameters are independent of the target item i, it is only dependent to user and context item. The effect of changes of target item is completely embodied in the parameter \(s_{i,C_{j}}\).

### 4.2 Learning Process

In addition to similarities, the extra weight parameters are usually computed offline. Now we describe the supervised learning process that we use to tune the weight parameters.

The inputs to the process include: the m×(n+c) rating matrix \(R\), a pre-computed (n+c)×(n+c) similarity matrix \(S\) based on (2) (there is a c×c blank sub-matrix because similarity between context items are not required). The output will be an m×c weight matrix \(W\).

We explain the learning process for a specific user u. This will generate one row of matrix \(W\). The complete matrix is obtained after applying the learning process for all users. We define:

\[
w_u = \begin{bmatrix} w_{u,C_{1}} & w_{u,C_{2}} & \cdots & w_{u,C_{|C|}} \end{bmatrix}^T
\]

\[
s_i = \begin{bmatrix} s_{i,C_{1}} & s_{i,C_{2}} & \cdots & s_{i,C_{|C|}} \end{bmatrix}^T
\]

\[
R_u = \begin{bmatrix} R_{u,C_{1}} & R_{u,C_{2}} & \cdots & R_{u,C_{|C|}} \end{bmatrix}^T
\]

Substituting known values with constants:

\[
P_{u,i} = \frac{C_{1,u,i} + w_u \times s_i \times R_u}{C_{2,u,i} + \text{tr}(|W_u S_i|)}
\]

where \(s_i \times R_u\) is the entry wise product of \(s_i\) and \(R_u\), \(\text{tr}(A)\) is the trace of a matrix \(A\).
Suppose the number of items rated by user \( u \) is \( n_u \). Their indices are from 1 to \( n_u \). Then we can list an array of equations based on (7) with \( i \) values from 1 to \( n_u \). Our objective is to minimize the prediction error denoted by \( \Sigma_{i=1}^{n_u} (P_{ui} - R_{ui})^2 \). We adopt an iterative method to approximate the best \( w_u \) values.

Our approach avoids solving a set of non-linear equations by treating the denominator in (7) as a constant in each iteration. Letting \( P_{ui} = R_{ui} \) for \( i \in \{1, 2, \ldots, n_u\} \), the set of equations will have the form:

\[
Aw_u = b
\]

where \( A \) is a \( n_u \times ||C|| \) coefficient matrix, and \( b \) is a coefficient vector of length \( ||C|| \). Then, we can use Moore-Penrose pseudo-inverse to get \( w_u \) for next iteration:

\[
w_u = A^+b
\]

### 4.3 Sparsity Problem

Sparsity problem is well-known in the community of recommendation systems. In our system, we tackle the sparsity problem by leveraging the power of semantic web.

If the rating matrix is not sparse (specifically, the number of ratings given by user \( u \) is greater than \( \delta \)), our modified CF algorithm is applied to give recommendations. Otherwise, our system will work in the expert system mode. The system will first ask the user to input several selecting criteria. The input UI is demonstrated as in Fig. 1. The UI prompts users to input a searching criteria, represented by a subject, a property, and an object/value. The top spinner (drop-down list) contains all ontology classes. They are sorted according to the class hierarchy to ensure easy access. The second spinner specifies a restriction on ontology properties. Initially this spinner contains all possible properties, but as users select some specific class in the first step, some properties that cannot be applied to the specified class are filtered out. Depending on whether the property selected is an Object Property or Data type Property, the user will be prompted to input either the third spinner or the text field. Inside the text field, users can input either a number, or an expression (for example “\( >1 \)”). After the selection criteria is determined, a semantic language query is fired to retrieve all relevant items matching those ontology classes, and the answers are formatted and displayed to users. The number of ratings received for each item in the answer set is used to determine the order by which these answers are displayed to users.

5. **EVALUATION**

This section evaluates our system in two perspectives. Firstly, we examine the effectiveness of the recommendation algorithm as well as the learning process through quantitative metrics. Secondly, we survey users of the prototype system to receive qualitative responses.

#### 5.1 Effectiveness of the Algorithm

The differences between CCF and other multi-dimensional (MD) context-aware CF algorithms are: 1. CCF incorporates context-aware CF algorithms. 2. A weighting scheme is employed to translate the impact of context information into prediction values. 3. An iterative learning process is devised to produce the weights.

In order to compare the performance of CCF with MD, a dataset with context information appended to each rating is required. However, traditional recommendation system benchmarks do not consider the context of the rating. We have to collect the information by ourselves.

We built a spreadsheet to collect user ratings on movies. By referring to the data collection procedure as described in [10] and [11], each rating is appended with 4 context attributes: Gender, Time (weekday or weekend), Location (at home or at cinema), and Companion (alone, with friends, with lover, or with family). All of the context information can be input with the possible choice of “don’t remember”. Finally, a dataset with 52 users, 38 movies and 945 ratings is constructed.

The evaluation of our system is carried out as follows: The ratings given by 52 users are split into two groups. The one with 45 users is used as training set and the other with 7 users is test set. This split is done 15
times. The metrics chosen are Mean Absolute Error (MAE), Precision and Recall. MAE is defined as:

$$\text{MAE} = \frac{\sum_{i \in T} |R_i - P_i|}{\|T\|}$$

(10)

where $T$ is the test set, $R_i$ is the actual rating, and $P_i$ is the predicted rating.

Fig 2 shows the comparative MAEs of CCF and MD under all 15 experiments. The average MAE is 0.55 for CCF and it’s 0.63 for MD.

Fig 3 shows the precision/recall measure of both approaches. The precision/recall for CCF is averaged to 0.625/0.401, while for MD it is 0.575/0.415. The F-measures are 0.53 and 0.48 for CCF and MD respectively. Though there are exceptions when CCF performs worse than MD, the overall performance of CCF is better, with smaller MAE and higher F-measure.

To sum up, by utilizing our approach, context information can be better utilized and thus better recommendation performance is observed.

5.2 User Survey

So far we have justified the effectiveness of our recommendation algorithm, now we want to examine how the system as a whole can help users. A prototype system is built on Android and a server is set up to respond to queries from the clients. The knowledge base of the system is based on the test bench proposed in [12]. This is a test bench specially designed for the domain of mobile context-awareness. Currently we have implemented only one feature of the whole system, i.e. context-aware music album recommendation. For our system, we added a thorough hierarchy of modern music genres and albums into the ontology. Specifically, DBPedia1 is linked to the ontology to provide the professional music taxonomy. The information of a total of 1545 albums released in the years 2010, 2011 and 2012 are extracted from Wikipedia lists.

The system works as follows: At the initial setup, the application will require users to input several ratings for the albums she had listened to. This can help the system to recommend with better precision. However, this step can be skipped when the user prefers to try out the application first. Then the system will recommend several music albums to the user following the algorithm we have proposed. This can be done either through a collaborative filtering process or through the expert system approach. The expert system approach requires the user to input several choosing criteria. If no criterion is input, the most popularly recommended albums are recommended to the user. When the recommended item is decided, a Google link and a YouTube link are given to redirect users to listening pages. After the user has finished listening to the music, the users can input her rating for the album in the application. This rating is taken down together with the current context of the user, to improve future recommendation precision.

10 users are invited to try out this application after a prototype has been developed. Some of the user feedbacks are quoted below:

1 http://wiki.dbpedia.org/About
“Interesting app. Hope to see it in one piece and will definitely try again then.”

“It gets me introduced to some rock music that I have never tried. But it turns out to be quite good.”

“The idea is good. But the application seems to be too dull. More background information could be given when recommending so that we get more than just an album name.”

Overall, the feedback is positive. Users are delighted to use the application and many suggestions on improving it are given.

6. CONCLUSION

In this paper, we have proposed a novel system to realize context-aware mobile commerce. Two important aspects of the system are analyzed. The first is how to capture and represent context information. The second is how to make use of the context information in a recommendation algorithm.

Context information, after being captured by a sensor or a crawler, is represented as a triple in the knowledge base. This triple is then quantified into a scale from 1 to 5, and it is plugged in the rating matrix. Context information is then used in the context-aware collaborative filtering algorithm to tune recommendations. Unlike the traditional solution for context-aware collaborative filtering algorithms, our approach managed to represent context information within 2-D space. With a specially designed weighting scheme, the context information can be utilized in the calculation of similarities between items. This change rid us from setting an arbitrary numerical threshold or a cut-off qualitative context, and we can benefit from the quantitative effect of context information. Together with the weighting scheme, an iterative learning procedure is proposed to learn the weights from training sets.

The algorithm is tested in a movie recommendation scenario. Experiment results show our approach can decrease MAE and produce higher precision and recall. A prototype system on the domain of music recommendation is constructed and multiple users are invited to try and comment on it. The feedback from users shows the system is promising and it gives them positive mobile commerce experiences.

REFERENCES


AUTHOR PROFILES

Zhu Cenzhe received the B.Sc. degree in Computer Science in 2009 in Shanghai Jiao Tong University, China. From 2009-present, he’s pursuing the Ph.D. degree in National University of Singapore. His research interest includes context-aware systems, semantic web, and networks.

Tay Teng Tiow received the BEng in Electrical Engineering with first class honours from the National University of Singapore in 1985 and the PhD in System Engineering from the Australian National University in 1989. Since 1989, he is with the National University of Singapore (NUS).