

# The use of data mining techniques in location-based recommender system

Nafiseh Shabib

Dept. of Computer and Information  
Science Norwegian University of  
Science and Technology, 7491  
TRONDHEIM, Norway  
+47 451 27432

shabib@idi.ntnu.no

Prof. John Krogstie

Dept. of Computer and Information  
Science Norwegian University of  
Science and Technology, 7491  
TRONDHEIM, Norway  
+47 735 93677

Krogstie@idi.ntnu.no

## ABSTRACT

Due to exponential growth of available information about products, users face problems when trying to identify products of interest. Because of this there is a need to design systems to help users select products more tailored to their interests through recognition of users' behavior. In the present paper, by using data mining techniques and making clustering model, the users with similar preferences were recognized and provided with appropriate recommendations. These recommendations were not only personalized on the basis of customer's behavior but also by considering user's location. MovieLens, a movie recommendation site, was used for model assessment. The results of the assessment showed that the method presented in this paper has a good accuracy for prediction of users' preferences.

## Categories and Subject Descriptors

H.2.8 [Database Applications]: Data mining. K.8.m [Personal Computing]: miscellaneous.

## General Terms

Algorithms, performance, Design, Experimentation

## Key words:

Recommender system, data mining, context-aware

## 1. Introduction

Advances in technology and development in mobile technologies have led to the advent of a new type of electronic commerce; namely mobile commerce. Mobile commerce is referred to as buying and selling of goods and services using mobile devices e.g. smartphones. On one hand, due to exponential growth of information available on goods and services, users face problems when trying to select goods and services they like. For the same reason the researchers seek a way to reduce users' challenges upon selecting goods and

services of their interest. A recommender system can be defined as a system which has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options [0]. To be able to perform a personalization, a recommender system needs to understand the user's interest or preference [0]. These systems were created to help in searching items or services tailored to users' preferences, because it recommends the most suitable item by taking into account users' preferences upon selection of items. Recommender systems are especially of high importance in electronic commerce and mobile commerce; most recommender systems having appeared in the past years were intended to help users in electronic commerce. But, previously recommendation systems were of limited for use in mobile shopping environments. A unique feature in such mobile shopping environments is the awareness of users' location; through which the recommender systems are not only able to make recommendations based on users' interests and preferences, but also these recommendations may be personalized on the basis of users' location. Taking into account this significant feature of mobile commerce, this study made attempts to design a system to recommend items for users taking into account not only their interests, but also their physical location. The rest of the paper is structured as follows. In Section 2 we present previous research in the area. Section 3 describes the overall architecture of the approach, whereas details are presented in section 4. Section 5 presents an empirical evaluation of the approach, before concluding the paper in section 6.

## 2. Overview of Previous Research

Recommender systems are systems that aim to present personalized recommendations of various items such as movies, websites or research papers to users. The first recommender system [0], called Tapestry, was presented by Goldberg et al. [0], and started off a research field which has been gaining much attention since, both in research communities, and from commercial actors. Nowadays, people surfing the Internet will bump into recommender systems when visiting popular websites like amazon.com or YouTube.

The job of recommender systems is to support the user in the search process by providing personalized recommendations or meta information such as ratings and user reviews to make Internet based shopping easier. The benefits for the user are more efficient shopping as well as increased confidence in that they are making the right decision. For the companies integrating recommender system into their solutions the primary benefit is

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WTMS'11, May 25–27, 2011, Sogndal, Norway.

Copyright 2011 ACM 978-1-4503-0148-0/11/05...\$10.00.

increased revenue, as they can present a small set of items to customers that are likely to find these particular items interesting. Further, companies can benefit from that the users get attached to differentiating attributes recommender system represents. Many recommender systems long-term interaction to improve recommendations, thus making it less favorable for the user to switch to a competitor [0].

## 2.1 Type of recommender system

Recommender system comes in many variants, being different the internal mechanisms for generating the recommendations or the user interface or both [0]. However, the core of any implementation is some form of information filtering based on a model of the user's preferences. The most common filtering techniques used are [0]:

- Content based filtering
- Collaborative filtering

Systems combining these two filtering techniques, so called hybrid systems, have in recent years become popular a way to address the limitation of the two methods [0] as described below.

**Content-based filtering** has its background in information retrieval [0], and is based on the idea that users will like items that are similar to items that they have previously expressed liking in.

For example, in a recommender system providing recommendations of restaurants the associated features of restaurant can be type of cuisine served, price level or whether or not the particular place have out-door service. To find a restaurant recommendation for the user, the content based filtering algorithm can conceptually be said to work like this: The system has collected a set of features combined with the probability that each user will like the feature. To find a restaurant the user probably will like, the system does a comparison between the set and the various restaurants in the system.

The comparison can be some sort of heuristics or simply a similarity calculation as the cosine-based calculation [0]. The system then returns the restaurants that represent the best fit. Content-based filtering can be very useful when much descriptive data is available on the items. This can be codified and represented internally as an ontology of features, giving detailed description of the domain applicable for similarity calculation against the user's preferences. Nevertheless, to be useful a content based scheme must have this ontology and means to classify items. Because of this, content based filtering can be problematic when the amount of information available on each item is sparse or difficult to interpret with computer algorithms. Further, content based recommendations can only be based on the explicit information available on a certain item.

**Collaborative filtering** is by far the most widespread recommender technique because of its power and simplicity [0]. The greatest strength of collaborative techniques is that they are completely independent of any machine readable representation of the objects being recommended, and work well for complex objects such as music and movies [0] where there can be many reasons for the liking of the product. E.g., for a movie, the user might like a certain genre of movies, a specific director, a specific actor or actress etc. Collaborative filtering is often referred to as

social filtering as the technique recognizes commonalities between users on the basis of their ratings, and generate new recommendations based on inter-user comparisons [0]. This process can be said to mimic the social process of recommending items among friends, as the system gives recommendation to users based on what people who are similar to the user in terms of ratings has liked in the past. An important distinction between movie recommendations around the coffee machine and in commercial collaborative system is that number of "friends" in such a system can be millions. This brings us to one of the disadvantages of the collaborative approach. An obvious challenge is to be able to present recommendations on new items, which not yet have been given many ratings from users. This problem is known as the cold start problem. Further, both collaborative and content based filtering (as well as approaches combining the two) rely on knowledge of the user in order to do deductions in feature space or calculate correlation with other users.

The other types of recommendation techniques identified in [0] are the demographic, utility-based and knowledge-based techniques. These are techniques that are rarely seen as stand-alone techniques in recommender systems, but are often combined with collaborative filtering or content-based systems in hybrid approaches.

Demographic approaches base their recommendations on personal attributes of the target users. On background of these attributes, users are grouped together in groups that receive similar recommendations. An example of a system using demographic filtering for marketing is presented in [0].

*Utility-based* approaches define the user profile as a utility-function. Each user has its own utility function that is used to generate recommendations. The main issue of these systems is how to create the utility function. One example of how this can be done is the utility function in the system Tête-à-Tête presented by Guttman [0]. This approach uses a similar set of utility functions to weight the attribute's utility based on how well the offer value satisfies the attribute's criterion.

The final recommendation technique identified in [0] is the knowledge-based technique. This technique is based on encoding knowledge of how items meet user needs into data structures that recommender systems can use to infer recommendations. The knowledge can be encoded in different ways, for instance as rules in a rule-based system. An example of a recommender system automatically inferring rules from underlying data is proposed in [0].

## 2.2 Context awareness

We follow the widely accepted definition of context by Dey and Abowd. Context is any information that can be used to characterize the situation of an entity. An entity is person, place or object that is considered relevant to the interaction between a user and application, including the user and application themselves [0]. Context awareness relate to how the applications adapt to information such as location, time etc. Context aware systems work to provide the user with relevant information. The adapted strategy can have major implications for the user experience. One of the important parts of context is location. The context is much more than location, but its other elements are still

difficult to identify or measure. Current mobile services can be enhanced with location-aware features, thus providing the user with a smooth transition towards context-aware services [0]. In addition, location- awareness offers great opportunities like instant location and map guidance.

### 2.3 Context aware recommender system

Predicting the preferences of users and providing the personalized services or products based on their preferences are important issues. However, the research considering users' preferences on context-aware computing is a relatively immature research field [0]. Context-aware recommender systems utilize information about the situation of the target user to generate more accurate recommendations. Consider a guy who is usually into action movies, but this time around need a recommendation for a movie which he can watch with his friend who is less enthusiastic about action movies. Or similarly, consider a tourist who has spent the last five hours since breakfast walking around in museums, and is eager to find a place nearby to have lunch. Context-independent recommender systems, only considering the user and item dimensions as described previously in this section, will struggle in these situations to recommend suitable items because they do not take into consideration the circumstances in which and when the user need recommendation. As a solution to this problem, which is identified as one to solve for the upcoming generation of recommender systems [0], context- aware recommender systems have been proposed. How a recommender system should be context-aware depends on the domain. In the movie domain it might be most important to incorporate a dimension that is who the target user is watching a movie with (cf. example above) or what movies are currently running in a nearby theater. In a tourist domain, it might be more important to take time of day, location or previous visiting history into consideration. The main problem with context adaptation is that many aspects of the context cannot be easily identified or measured. The location of the user is an element of the context that currently can be measured more or less accurately depending on the positioning system in use. Location-based services are services that provide information about certain places or locations. Thus location-aware services are a special case of location-based services. So far, context-awareness has mainly been studied from the technical point of view and the studies have concentrated on location.

### 3. System architecture

A simple recommender system has been illustrated in Figure1. As shown, a mobile user enters the environment; the recommender system identifies the relevant user and, then, taking into consideration the user location, his/her profile examination as well as various recommendation personalization techniques available in the system, the system dispatches the recommended content to the user. Thus, a highly flexible and dynamic recommendation is sent to the mobile user and, the customers are more motivated to pay attention to the recommended messages.

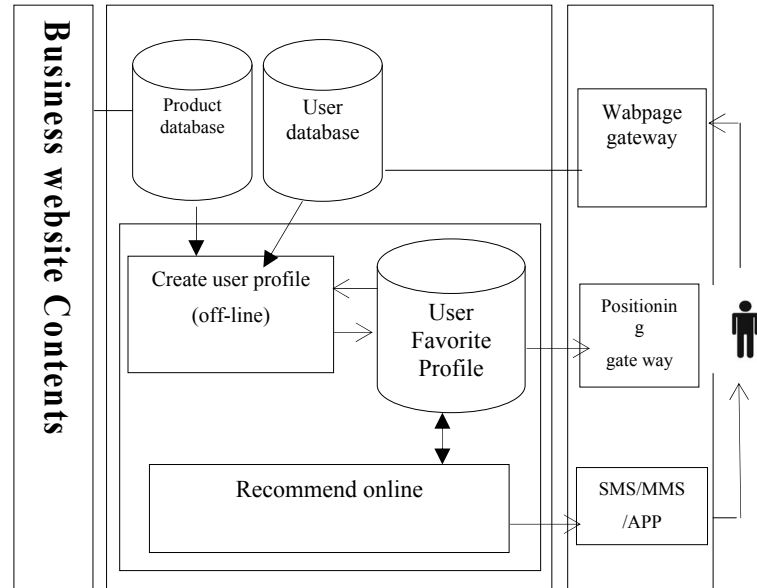


Figure1. Mobile Recommender System Architecture

## 4. Details of approach

### 4.1. Recommendation procedure

This paper suggests a methodology for context-aware recommendations. The methodology consists of four phases as shown in Fig. 2. The recommendation problem should be initiated by classifying the products and customers to improve the analysis process of similar customers and products.

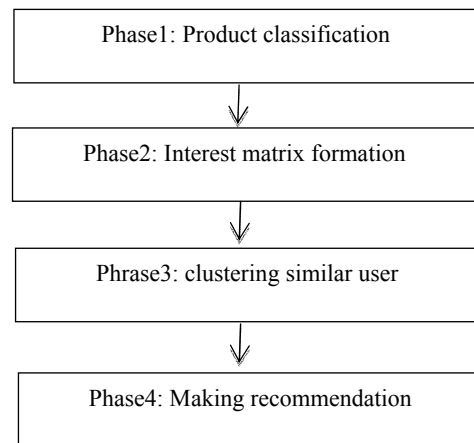


Figure2. Recommendation procedure

#### Phase 1: Products Classification

A formal definition of classification is not attempted; for our purposes it is sufficient to think of classification as describing the process by which a classificatory system is constructed. The

word *classification* is also used to describe the result of such a process. Although indexing is often thought of as classification we specifically exclude this meaning. A further distinction to be made is between classification and *diagnosis*. Everyday language is very ambiguous on this point:

- How would you classify (identify) this?
- How are these best classified (grouped)?

The first example refers to diagnosis whereas the second talks about proper classification [0]. In this section we study how to diagnose the class of one object. A (*product*) class is a set of products. For products classification we put products that have the same attributes in the same class. For instance, we can put movies with the same genre together in the one class.

## Phase2: interest Matrix Formation

One of the most important steps in recommendation system is the formation of an interest matrix. A customer-product rating matrix is used to determine the preferences of each customer to each product. In the previous step the products were classified; but, no categorization was made of customers. Our proposed solution in this step is making customer-product rating matrix. Because of this matrix grows linearly with number of customers and products, our customer-product matrix involves the preferences of each customer to class of each product will be shown as *favorite* [i, j] as you can see in Figure3. In the matrix of *favorite* [i, j] with a size of  $x*y$  ( $x$  customers and  $y$  product class) indicates the proportion of customer interest  $i$  to the products available in class  $j$ . The value of *favorite* [i,j] can be computed as follows where any purchases that is made the customer is updated in class  $j$ .

$favorite[i,j] = (\text{Sum of the rating of products belonging to a certain class}) / (\text{Number of products rated in certain class} * 5)$

For instance, each customer rated some products in a certain class between 1 and 5. Suppose to, he rates five movies in the genre of action in this way (1, 2, 1, 3 and 5). The *favorite* [i,j] is calculated through  $(1+2+1+3+5)/(5*5)=0.480$ . By using this calculation, we find his rate to the certain class.

|                | pc <sub>1</sub> | pc <sub>2</sub> | ... | pc <sub>x</sub> |
|----------------|-----------------|-----------------|-----|-----------------|
| C <sub>1</sub> | 0.8266          | 0.362           | ... | 0.628           |
|                |                 |                 |     |                 |
| C <sub>y</sub> |                 |                 |     |                 |

Figure3. Customer -Product Rating Matrix

## Phase3: Similar User Clustering

In this phase the demographic characteristics of users such as age, job and gender accessible in any customer's profile along

with the above matrix, which determines the interests of any user to special class of products, were employed for the purpose of similar users clustering.

K-means [0] is one of the simplest unsupervised learning algorithms that solve the clustering problem. Among clustering formulations that are based on minimizing a formal objective function, perhaps the most widely used and studied is k-means clustering. . Given a set of data points and the required number of  $k$  clusters ( $k$  is specified by the user), this algorithm iteratively partitions the data into  $k$  clusters based on a distance function. This measure is often called the squared-error distortion [0], [0] and this type of clustering falls into the general category of variance based clustering [0], [0]. Clustering based on k-means is closely related to a number of other clustering and location problems. These include the Euclidean k-medians in which the objective is to minimize the sum of distances to the nearest center and the geometric k-center problem [0] in which the objective is to minimize the maximum distance from every point to its closest center.

The algorithm is composed of the following steps:

1. Place  $K$  points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the  $K$  centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

In this system the users can be clustered through two models. The best one, the model described above, categorizes users based on their behavior patterns similarity. But, where a new user enters the system (cold start), since there is no information on user's behavior patterns for selecting the products class of interest, the system would temporarily employ demographic characteristics. Then, with considering the classes of products rated by user, the system would examine the user's behavior pattern and relocate the user according to this.

## Phase4: Making Recommendation to the Users of Special Cluster

This phase is involved in product recommendation. This algorithm recommends some of the products in each class of product classes based on the rating values in the matrix and the location of the user. As explained before, the recommender system recognizes the users' profile based on their interests and, then, estimates the latter. Thereafter, a list of products (top-N) tailored to the recommendation is selected.

The recommendation system presented in this study, in addition to user's profile, takes into consideration the physical distance between users and the vendor. In recommender system it has been supposed that the probability of observing a

vendors' web page is related to exponential distance between the user and the vendor [0].

$$DistanceDecay(c,w) = 1/(e^{\lambda \times Distance(c,w)})$$

In this formula we have  $\lambda \in [0, \infty]$  which is a parameter indicating the user's sensitivity to location.

Distance(c,w) shows the Euclidean distance of user  $c$  to the product of vendors' web page  $w$ .

Therefore, it is said that the interest of user  $c$  to the product of vendors' web page  $w$  at user's current place can be calculated by the following formula:

$$Interest(C,W) = Favorite(C,W) * DistanceDecay(C,W)$$

For those customers that do not care for the distance to the vendors, parameter  $\lambda$  takes the value of zero and for those customers very sensitive to the place of vendors' distance it takes value of close to  $\infty$ . It is obviously noted that when the value of  $\lambda$  is small, the exponential distance function is increased slowly indicating that the customer is interested in selecting those products of vendors which are located at a farther distance than the customer.

When the value of  $\lambda$  is large, the exponential distance function is decreased rapidly indicating that the customer has no interest to select those products of web page located at a farther distance than the customer.

The customer sequence in the proposed recommendation system is shown in Fig.4. In the next section, our proposed method is investigated by doing some evaluation tests.

The geographical location of users can be found for instance by GPS or in specific city wlan-environments such as Wireless Trondheim [0] by wlan-triangulation. The system need to keep the server updated on the user current context. When a user's location is defined, the system sends a personalized advertisement. Also, regarding to the  $\lambda$  parameter, we can estimate the sensitivity of users with location. When the process finds a new recommendation to add to the plan, the  $\lambda$  parameter is updated in the user profile.

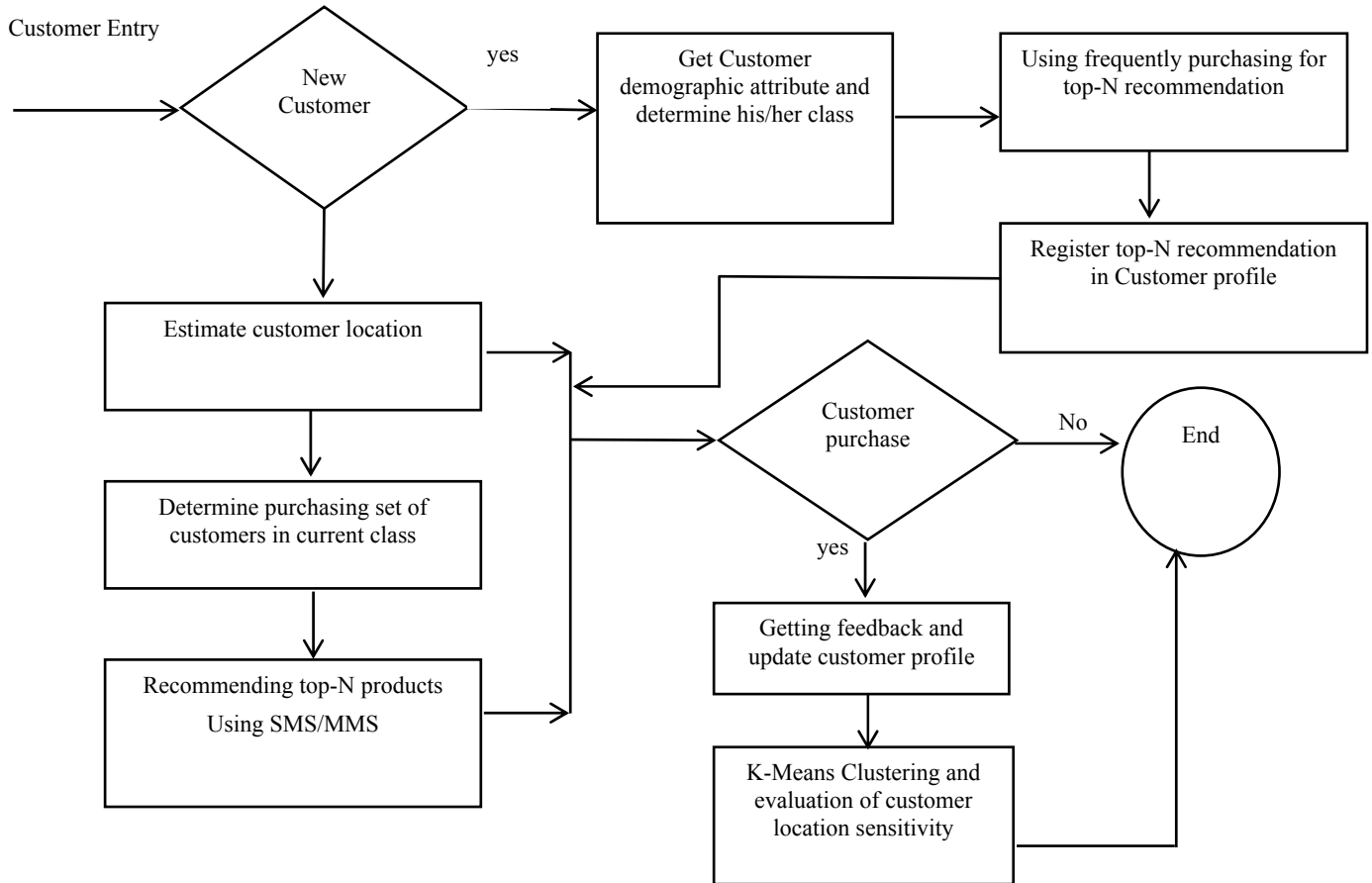


Figure4. Customer sequence in Recommending

## 4. Assessments

Our proposed model is based on Collaborative filtering. As a first evaluation of the approach we used a public dataset for collaborative filtering, MovieLens, for part of the offline recommendation. MovieLens is a web-based recommender system established in 2003. There are over 6000 users enrolled in this site who show their interest in 3800 products (movies). But, since we need additional information of users that is related to their place we developed a scenario. The user-movie set indicates how the user has rated for which movies, the type of movies rated by any user was determined and a vector of 18 movies of various types was specified for each user. Then, by using K-Means the similar users were located into 20 clusters. Afterwards, 80% were positioned in a learning set and 20% in a testing set, respectively. The recommendation algorithm was performed on the learning set and a series of recommendations were made for each user. After that the algorithm test had been made, and it was demonstrated that the algorithm made correct recommendation to the users of any cluster with an error mean between +0.77 and -0.77 that is shown in Figure 5. Moreover, for new user the average mean is about +,-0.84. The experimental software used to implement our system was developed by visual studio.net.

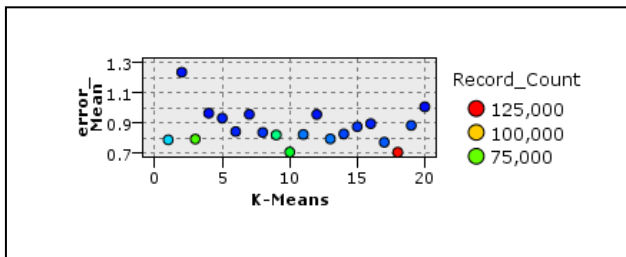


Figure5. Error mean in each cluster

In this study, we proposed a novel recommender system for recommending advertisements of the location based service. Our proposed system is designed to apply the personalization of mobile users- demographic attributes, behaviors and location. It sends the personalized advertisements to the users. As we know, most advertisements that are forwarded on the streets are wasted, because they are not sent to specific user. In addition the location of advertising market might be important for users. As an informal statement "All things are related, but near things are more related than far things."

Our study also has some limitation. First, there is not public dataset that has parameters related to context situation, especially location. Designing the systems like this is made complicated by a number of factors. Presenting product information, designing product classification, solving cold start and sparsity problems and customization the system for mobile users are the challenging for recommendation systems in general. Finally, in order to recommend more precisely, recommender system needs more information. However, people may not want their habits or views widely known. So, customizing privacy for each user may be a way to improve the efficiency of this system.

## 5. Conclusion

In this research, the recommendation mechanism was investigated in movielens and it was shown that this will lead to good results. Because the recommendations, in addition to being personalized on the basis of users' profile, will benefit from the unique feature of mobile environments, i.e. awareness of user's location. The recommended model comprises of two main parts:

- 1) The first part is offline which, due to users' profile, would provide users with a list of personalized and private recommendations.
- 2) The second part is online which, based on user's location, would sort the list of previous recommendation and send it to user.

Meanwhile, it has been shown that the system had numerous advantages and, in general, it can be said that the recommended model has a good efficiency.

## 6. References

- [1]A. Gersho and R.M. Gray, Vector Quantization and Signal Compression. Boston: Kluwer Academic, 1992.
- [2] A.K. Jain and R.C. Dubes, Algorithms for Clustering Data. Englewood Cliffs, N.J.: Prentice Hall, 1988.
- [3] Andresen, S. H., Krogstie, J., & Jelle, T. *Lab and Research Activities at Wireless Trondheim*. Paper presented at the 4.th IEEE International Symposium on Wireless Communication Systems (ISWCS'07) Trondheim, Norway, 2007
- [4] Bamshad Mobasher, Honghua Dai, Tao Luo, and Miki Nakagawa. Effective personalization based on association rule discovery from web usage data. In *WIDM '01: Proceedings of the 3rd international workshop on Web information and data management*, pages 9–15, New York, NY, USA, 2001. ACM. ISBN 1-58113-444-4. doi: <http://doi.acm.org/10.1145/502932.502935>.
- [5]Bruce Krulwich. Lifestyle finder: Intelligent user profiling using large-scale demographic data. *AI Magazine*, 18(2):37–45, 1997.
- [6]B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-Based Collaborative Filtering Recommendation Algorithms," Proc. 10<sup>th</sup> Int'l WWW Conf., 2001.
- [7]David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992. ISSN0001-0782. doi: <http://doi.acm.org/10.1145/138859.138867>.
- [8]Eija Kaasinen, "User needs for location-aware mobile services", Pers Ubiquit Comput (2003) 7: 70–79 DOI 10.1007/s00779-002-0214-7"
- [9] Gediminas Adomavicius and Alexander Tuzhilin. Towards the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

- [10] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, pages 304\_307, London, UK, 1999. Springer-Verlag.
- [11] J. Ben Schafer, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. pages 158\_166, 1999.
- [12] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5\_53, 2004.
- [13] J. B. MacQueen (1967): "Some Methods for classification and Analysis of Multivariate Observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*", Berkeley, University of California Press, 1:281-297
- [14] Jongyi Hong, Eui-Ho Suh, Junyoung Kim, SuYeon Kim, Context-aware system for proactive personalized service based on context history, *Expert Systems with Applications* 36 (2009) 7448–7457
- [15] J. Kwon<sup>1</sup>, S. Kim<sup>2</sup> "Context-Aware Recommendation Service Using Multi-leveled Information in Mobile Commerce" Springer-Verlag Berlin Heidelberg 2006
- [16] M. Inaba, N. Katoh, and H. Imai, "Applications of Weighted Voronoi Diagrams and Randomization to Variance-Based k-clustering," *Proc. 10th Ann. ACM Symp. Computational Geometry*, pp. 332-339, June 1994.
- [17] M. Inaba, H. Imai, and N. Katoh, "Experimental Results of a Randomized Clustering Algorithm," *Proc. 12th Ann. ACM Symp. Computational Geometry*, pp. C1-C2, May 1996.
- [18] Marko Balabanovic and Yoav Shoham. Fab: content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, 1997. ISSN 0001-0782. doi: <http://doi.acm.org/10.1145/245108.245124>.
- [19] N. M. Sadeh, Ting-Chak Chan, Linh Van, OhByung Kwon and K. Takizawa. "Creating an Open Agent Environment for Context-aware M-Commerce", *Lecture Notes in Artificial Intelligence*, p.152-158, 2003.
- [20] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, New York, NY, USA, 1994. ACM. ISBN 0-89791-689-1. doi: <http://doi.acm.org/10.1145/192844.192905>.
- [21] P.K. Agarwal and C.M. Procopiuc, "Exact and Approximation Algorithms for Clustering," *Proc. Ninth Ann. ACM-SIAM Symp. Discrete Algorithms*, pp. 658-667, Jan. 1998.
- [22] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331\_370, 2002.
- [23] Robert H. Guttman. Merchant differentiation through integrative negotiation in agentmediated electronic commerce. Master's thesis, School of Architecture and Planning, Program in Media Arts and Sciences, Massachusetts Institute of Technology, 1998.
- [24] *Three Automatic Classification*, [www.dcs.gla.ac.uk/Keith/Chapter.3/Ch.3.html](http://www.dcs.gla.ac.uk/Keith/Chapter.3/Ch.3.html)
- [25] Wan-Shiou Yang, Hung-Chi Cheng, Jia-Ben Dia, "A location-aware recommender system for mobile shopping environments", (2008)