Recommending Stores for Shopping Mall Customers with RecStore

Diogo V. de S. Silva, Renato de S. Silva, Frederico A. Durão

Federal University of Bahia, Salvador, Brazil
diogovss@ufba.br, renatosilva@dcc.ufba.br, freddurao@dcc.ufba.br

Abstract. Today mobility is a key feature in the new generation of Internet, which provides a set of custom services through numerous terminals. Smartphones, for example, are a tendency and almost mandatory for anyone living in an urban and modern context. Most developed cities have at least one shopping mall full of mobile device users. These shopping malls provide a number of stores, and people tend to have difficult in finding what they really need. This article proposes a solution called RecStore. RecStore is a recommendation model to assist customers in reaching what they consider relevant at malls. The recommendation model comprises user activities, 330 stores, 30 users and 3 baseline models. The precision, recall and f-measure improved at rates of 118%, 76% and 95% respectively in comparison to the second best model of each metric. Additionally, a mobile application — called InMap — was implemented based on our model RecStore.

Categories and Subject Descriptors: H.2 [**Database Management**]: Database Applications; H.3 [**Information Storage and Retrieval**]: Systems and Software; H.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing

Keywords: customization, mall, mobile, recommendation, shopping

1. INTRODUCTION

To find a store in large sales centers is not always an easy task. Customers everywhere are always looking for the best deals and offers. However, it is not always easy to find the best stores, since sometimes customers forget their location or get distracted among a number of options. Visiting shopping malls increases the possibilities to get around this problem. Shopping malls are attractive places, where many people buy products, use public services or just have fun. Spending time at shopping malls is a first leisure option in many countries, not only for purchasing things. One of the reasons is that shopping malls usually offer security and convenience [Martin and Mason 1987; Miller 1998]. Other advantages include bringing together several stores and services like banks, clinics, cinemas and restaurants.

Due to the high number of stores at shopping malls, consumers are not always able to find the best offerings. The consumers also are usually interested in grouping those stores that sell the same products. Most of the stores have limited promotion strategies in the way the products are presented. Usually the ads are made at the shop windows or through leaflet distribution. Even so, most of the time, customers are not always aware and get lost among so many options. Aiming to assist customers, some shopping malls provide digital totems and sometimes mobile applications to guide people. However, the stores layout does not follow a particular criterion of a person, which makes it more difficult to find the wanted stores.

This research was supported by Fraunhofer Project Center/UFBA who provided insight and expertise that greatly assisted the research. The authors would also like to thank to FAPESB for the support to carry out this research.

The task of offering personalized information for any customer is not straightforward. The customers, obviously, have distinct preferences and the recommendations are not always appropriate. There are cases where did ads and audio-visual marketing are done without thinking in customers individualities. A context-sensitive environment is one which can automatically choose from a multiplicity of options based on the current or previous state(s) of the environment. In the context of shopping malls, context-sensitive information (e.g.: location or schedule) are needed to satisfy customers preferences [Kramer and Modsching 2005].

Mobile technologies have had a major impact in the last decades. Smartphones and tablets are more and more important in people's routine. A multitude of applications were developed for assisting activities related to home, work and leisure. The use of the Internet has exponentially increased, thus generating a huge amount of information. However, most of the applications do not address an efficient filtering mechanism.

The access of the Internet throughout mobile devices like smartphones or tablets has become popular in recent decades. According to Holding [2015], in Brazil, in the first three months of 2015, 68.4 million people accessed the Internet through mobile devices. In the previous quarter there were 58.6 million people. According to the survey, most applications are about social networks, videos, e-mails, instant messengers, songs and portals.

Mobile devices store many users' information like photos, videos, browser history, browser cookies, application configuration, etc. Besides, the mobile device's GPS permits the storage of history about the places the user has visited. All this information could be used to understand specific user preferences [Marinho et al. 2012]. In this way, mobile applications may give alerts about one specific shopping mall, close to the user's current location.

Customized information tends to increase the consumer fidelity [Hennig-Thurau and Klee 1997]. Consumer fidelity makes the difference in the current competitive market. Therefore, one question arises: *how to recommend stores that are aligned with distinct customers interests?* This problem can be broken down into smaller problems such as:

—How to trace a consumer's profile only based on information collected by a mobile application?

—How to categorize stores in a way that simplifies the recommendation task?

—Which store should be recommended to the customer?

In order to support customers in shopping centers with a better shopping experience, we propose a recommendation model called RecStore. This research was initially presented in a previous article [Silva et al. 2017]. This article presents in detail a solution called RecStore. RecStore is a recommendation model to assist customers in reaching what they consider relevant at malls. In this way, RecStore aims to provide a personalized list of stores according to user preferences. RecStore uses as input two informations, user location and purchase behavior. This work also presents an android application — called InMap — which implements the RecStore strategy. InMap monitors, in the background, the required RecStore inputs and also offers an optimized search engine. The search engine returns categorized stores and a map to indicate the store locations. The approach was compared with other recommendation baseline models. Experiments indicate that RecStore better attends the customer's needs than well-known baseline models. The precision, recall and f-measure improved at rates of 118%, 76% and 95% respectively in comparison to the second best model of each metric.

The remainder of this article is structured as follows: Section 2 positions our work with respect to related findings in the literature. Section 3 introduces the InMap mobile application. Section 4 depicts the composition of user model, store model and presents the RecStore recommendation model. Section 5 describes the experimental setup and experiment results. Section 6 outlines conclusions and points out some future work.

2. RELATED WORK

Many studies have explored e-commerce recommender applications for purchase purposes, however, few of them have focused on shopping malls. Many of them try to overcome the problem of location. But, most of them study the recommendation of products using context-based information.

Lemos and Goes [2015] evaluate the behavior of consumers who make purchases through e-commerce using web platform and mobile platform. It examines the reason why consumers prefer to complete purchases through a computer and use mobile devices only as a means of tracking the delivery of the product. Similarly, Huang et al. [2018] examine the consumer profile of e-commerce in a dataset that covers 9.8 million users, 1.4 million of whom visited e-commerce platforms in one week during the study. Several observations were made about the cultural differences of users from different regions. The analysis shows that most mobile users are loyal to their favorite websites and that 60 % of customers tend to make quick decisions to buy something online, which usually takes less than half an hour. The author also reports that people in residential areas are much easier to shop than in business districts. Ricci [2010] presents a survey of shopping recommendation mobile systems focusing on mobile tourism recommendations. It highlights the challenges and opportunities in this field.

Asthana et al. [1994] focus on retail stores like supermarkets. A hardware device is attached to the cart and it accesses user history, providing product recommendations. The equipment also compares different brands of the same product, indicating the product location. Unfortunately, this model presents high infrastructure costs. Similarly, Stahl et al. [2005] present a pervasive solution involving navigational and shopping assistance by means of a personal device and an intelligent environment. Stah claims that portable devices are not appropriate to provide navigational assistance, suggesting in-place displays to guide the user, instead. The downside is scalability. It may work nicely with few users, but with dozens it decreases in performance. Both studies [Asthana et al. 1994; Stahl et al. 2005] need the acquisition of additional hardware to implement their solutions. This article proposes to save costs by only using devices already owned by mainstream users.

Kim and Park [2000] propose a custom recommendation system for a virtual shopping center. The system detects purchase intention patterns by grouping the purchase history of target customers. The system can also understand the buying characteristics of each customer. Once you understand some buying patterns, it determines a recommendation according to the customer's location within the virtual mall. According to the path of each client, items are recommended. In addition, the system stores huge amounts of information that can be used for personalized marketing campaigns. In turn, Anacleto et al. [2011] have studied recommendation systems in shopping centers. An overview of current support systems for shopping center environments is showed and then challenges and possible features using Semantic Web, mobile devices and sensor technologies are described. Unlike, Chang Lee and Chung [2005] present an online shopping center using a virtual avatar and a decision support system on the web. The virtual reality technique ensures a sense of reality for customers and facilitates the complex decision-making process in shopping.

Sae-Ueng et al. [2008] propose a consumer-friendly assistance service based on personal behavior data in ubiquitous commercial spaces. The system attempts to study individual behavior using RFID sensors and cameras in the environment to create a user profile. Subsequently, the services present — through monitors — new products. Similarly, Von Reischach et al. [2009] propose a product recommendation In their idea, shopping products have bar codes or RFIDs, enabling the user to review products and receive recommendations based on other users' reviews. Bajo et al. [2009] propose a software named SHOMAS which also uses RFID technology. SHOMAS is a multi-agent system that provides navigational assistance and suggestions in a shopping mall. Both of them [Von Reischach et al. 2009; Bajo et al. 2009] observe products instead of stores. RecStore is specialized in recommending stores, not products. Another problem is that these two studies need RFID tags, an additional cost.

Church and Smyth [2008] propose a multidimensional and context sensitive interface. Church combines context features (location, time, and community preferences) to deliver a search experience tailored to the needs of mobile users. The study adopts the user context in the form of location and temporal data. The preference information from other users with similar interests is also used. Later, it presents a research activities vision carried out by the user community. The user navigates the community search experiences and manipulates the searches of other users. Users also learn from searches by starting their own search. Thus, the system becomes proactive. Instead of explicitly recommending information requested by the user it recommends information about other users in similar contexts. Similarly, Yang et al. [2008] present a location-aware recommendation system. Yang analyzes accesses to web pages history to compose user profiles, and then recommends nearby stores. Bohnenberger et al. [2002] capture the user's location in a different way. In strategic places, beacons affixed to walls get information and send it to the system. Fang et al. [2012], in turn, apply RSS (received signal strength) to ascertain user position. Zhu and Tian [2014] explore information detection by mining users context. Likewise, Pessemier et al. [2010] introduce a context-sensitive recommendation system for mobile devices with Bayesian classifier. The system adds a new dimension in the user profile to record date and time stamp, apart from other contextual information such as mood and location. When analyzing user preferences, the system considers users' evaluations in different contexts, segregated by the Bayesian classifier.

Some related work have shown convergences with our approach. Kim and Kim [2005] propose a recommendation system based on data mining techniques. The focus is on shopping malls, however, contrary to our approach it is focused only on online shopping malls. Hussein [2009] presents a recommendation system implemented in mobile environments related to shopping malls. In this approach the clustering technique is used to generate recommendations through the K-means algorithm. However, the proposed approach does not compare with other baselines and experimental evaluations. The work is limited to proposing and presenting the architecture.

The proposal of this article is to monitors user activities and delivers personalized recommendations. In order to evaluate the performance and to raise comparatives with other baselines an experimental evaluation is performed.

3. INMAP - THE MOBILE APPLICATION

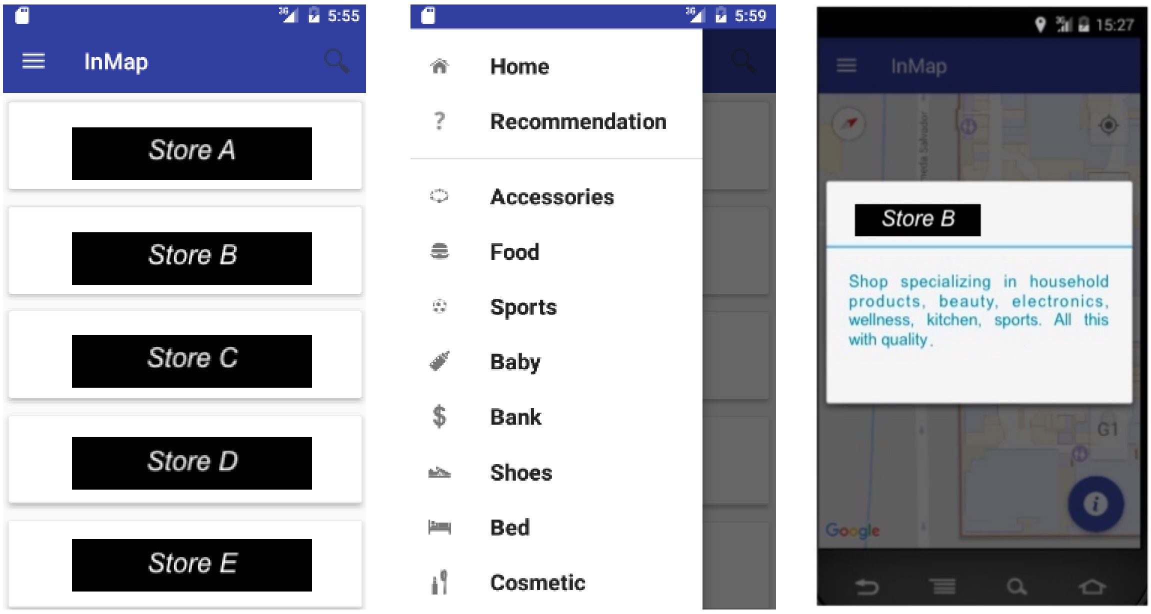
As aforementioned, a mobile application was implemented using the RecStore model. This section introduces such an application, the InMap.

InMap provides customers a set of features related to shopping recommendations (specially shopping malls). To avoid mentioning the brands in Figures 1 and 2 the real name of the stores were covered with a black stripe and replaced with letters from A to E. Figure 1a depicts a screen-shot with a list of recommended stores. The user may choose a specific store and click or she may type some query. For example, typing "shoes" in the text field, it shows only stores that sell that product. Figure 1b shows a second option. The user may search a shop by category. All recommendations are made based on user preference. Figure 1c shows the store details page, containing its name and a short description.

When a user executes the InMap application, their current location is automatically acquired. As Figure 2 illustrates, the current location is monitored at runtime every 3 seconds and shown on the map. The goal is to assists the user by defining the distance between the current point and the destination. The user may browse and select the store that is displayed on an interactive map. There are a few ways to get information about stores. The most intuitive is to navigate the list of stores and then select it (see Figure 1a).

The user may also browse through categories (as shown in Figure 1b). After selecting a category, all stores in this category are shown with their names and descriptions. Viewing all stores in a specific category is useful if the user knows what she wants but does not know exactly how to find it. In addition, it is an indication of interest in the subject.

Another way to get information about stores is by conducting a search. The search engine uses the query provided by the user and a comparison is made with the store name, tags, and description, to provide stores associated with the query. When the search is performed, a list of all matching stores is displayed using the same interface as the list of categorized stores. The app can use the search keywords to know a little more about the user. Once it is performed there is an implicit return of customer interest.



(a) Customized Stores. (b) Store Categories. (c) Store Detail.

Fig. 1: InMap Screenshot.

Fig. 2: Map with User Location Screenshot.

All of these workflows can lead to the detail page (see Figure 1c). The detail page shows all information about the store. So if the user accesses the detail page this means an interest in that store or type of store.

If the customer does not use the app, we take into account the user's location history within the mall. For example, if the customer stays more than 60 seconds at a relevant distance from a particular store, that information can be used implicitly to suggest stores when the application is used. All of this provided information is sufficient to create a refined user model, which will be used to generate recommendations, that can effectively determine user needs.

The RecStore model in practice is a software component responsible for recommendations. This component is self-contained and therefore it can be integrated with other platforms. The idea behind RecStore is presented in the next section.

4. RECSTORE - THE RECOMMENDER MODEL

RecStore suggests stores to a customer based on the location history and performed searches. The RecStore comprises 3 subcomponents, which are: the user model, store model and the recommendation model. These subcomponents are detailed below.

4.1 The User Model

Our user model follows the methodology proposed by Brusilovsky [2001], but incorporating the user's activities. To analyze this tracked information and make it productive by using, we may divide and formally represent the user model into distinguished sets according to the work of Senot et al. [2010]. For this we define as a tuple $\langle SP, SD, CV, LC \rangle$, where:

—SP represents a set of weighted terms that describes a **search performed** by a user. Once a search indicates an interest in a product or store, it provides strong evidence of which store a user might be interested in.

—SD represents a set of weighted terms from the **visited store's detail page**. This set is a strong indicator of user interests, once it clearly exposes the customer's will in that kind of store.

—CV represents the set of weighted terms of **store categories visited**. This set is a strong indicator of user interests once it clearly exposes the customer's will in that category of store.

—LC represents the set of store weighted terms identified by user **location monitoring**. It is also a strong indicator of user interest, since the user's frequency in certain types of store demonstrate an interest of this user.

4.1.1 *Weighing the User Model.* The number of times a term is searched indicates how much that term is important to the user. In other words, some terms can be more representative than others, meaning that the *frequency* of a term may denote its importance. For instance, suppose a user has visited the categories: "Clothing" and "Cosmetics", and the first category has been visited three times while the second has been visited only once. This means that the user has shown more interest in "Clothing" than "Cosmetics". Then the term frequency of the visited categories *CV={Clothing, Clothing, Cosmetics, Clothing}* will be represented as *{("Clothing",0.75), ("Cosmetic",0.25)}*. The term frequency is defined as:

$$termFreq(t, s) = \frac{n_t}{|T_s|}, \tag{1}$$

where n_t means the number of occurrences of the term $t \in T_s$, whereas T_s represents the terms in set $s \in S$ and $|T_s|$ is the amount of terms in a given set $s \in S$. The set T_s is normalized such that

$\sum_{i=1}^{|T_s|} termFreq(i) = 1.$

4.1.2 *Weighing the Sets.* Similar to terms, each set has its own importance to the user model. For example, the set SP (search performed) better exposes the user's need than the set SD (store details), shown on the details page. This is explained by the fact that SD only shows the user's curiosity in a given store, not necessarily a real need. A user can view the store detail page just to know more about what type of product the store sells.

Unlike terms, the importance of a set is not computed by a mathematical equation. Instead, weights are empirically predefined based on the common knowledge of the system administrator. We suggest the most appropriate weights respecting the following importance order:

$$\rho = SP > CV > SD > LC. \tag{2}$$

4.2 The Store Model

Similar to the user model, the store model SM_s comprises terms that describe the candidate store $s \in S$. In our context, the relevant information that is considered in SM_s includes its category and corresponding *tags*. The category refers to what the store is, and summarizes the main products or services available in the store. The *tags* provide specific information about a particular store and are pre-defined by the system administrator without user interaction. Formally, the store model SM_s is a pair $\langle C, T \rangle$, where:

—C represents the store category. The categories are pre-defined and remain static. The categories represent the type of store, for example: $C = 17$ means "Food" is the category of such stores.

—T represents a set of terms related to the store's products and services. The tags are pre-defined and static. Unlike the category, users do not know the tags, but tags are useful for the recommendation system and search engine. It is important to note that a set of *tags* is not a set of weighted terms. Both terms and tags are needed by the store model. Consider the following example: $T = \{$"*salad*", "*natural food*", "*juice*", "*orange*"$\}$. T represents the *tags* of the store, summarizing its main products.

Other information about the store itself, such as name, description and location may be useful. The description can be used to know more about the store, although the tags are much more useful. In our database, all stores are georeferenced, which will clearly improve the results of our recommendation model.

4.3 The RecStore Model Recommendation

A recommendation system is considered effective only if it presents unknown items. At the same time, those items must be something the user would like to see. Obviously, if the user already knows the item he or she does not need a recommendation system. Aiming to recommend unknown stores to users respecting their interests, we compare the similarity of one user model against each unvisited store model. A store is considered visited if the user opens the page of the respective store, or if he or she has spent a certain time on it. This distinction is adopted because our recommendation system should not recommend stores that the user already knows about. The result of this comparison is a ranking of candidate stores, similar to the user model.

The recommendation model is described as follows: For each user $u \in U$, we want to recommend unknown stores ($s^{max,u} \in S$) which maximizes the customization function *storeRec*, described as:

$$\forall u \in U, s^{max,u} = \arg\max_{s \in S} storeRec(u, s) \tag{3}$$

where U is the dataset of the entire user model, and S is the dataset of the unknown stores of u. The *storeRec* function is our custom similarity function between the user model and store model, and is described as:

$$storeRec(u, s) = sim(SD_u, T_s) \cdot W_{SD} + sim(SP_u, T_s) \cdot W_{SP} + termFreq(C_s, CV_u) \cdot W_{CV} + termFreq(T_s, LC_u) \cdot W_{LC} \quad (4)$$

where W_{SD}, W_{SP}, W_{CV} e W_{LC} are the normalized weights of each set in our recommendation model (see 4.1.2). *sim* is a similarity function used to calculate the similarity between the set of user model SD_u or SP_u of a user $u \in U$ and the store tags T_s of a store $s \in S$. The $termFreq(C_s, CV_u)$ is the function that calculates the similarity between the categories, visited by user CV_u, and store category [C_s (see 4.1.1). The $termFreq(T_s, LC_u)$] computes the similarity between store terms identified by customer location tracking and store tags.

These user set templates SD_u e SP_u and store assembly models T_s are represented as $\overrightarrow{SD_u}, \overrightarrow{SP_u}$ and $\overrightarrow{T_s}$ respectively. Technically, the cosine similarity [Baeza-Yates and Ribeiro-Neto 1999] between the vectors is calculated as:

$$sim(\overrightarrow{SP_u}, \overrightarrow{T_s}) = \frac{\overrightarrow{SP_u} \cdot \overrightarrow{T_s}}{|\overrightarrow{SP_u}||\overrightarrow{T_s}|} \quad (5)$$

It is worth mentioning that the aforementioned vectors comprise real numbers (weights) in which each value (normalized [0.1]) measures the importance of the corresponding term to the user or store.

The equation 4 always generates a real number in the range of [0.1] The number 0 represents no resemblance between the user and the store (it should not recommend). The number 1 represents absolute resemblance (it should recommend).

4.3.1 *Decay Function.* As time goes by, user preferences can change. Dealing with constant modifications is a typical issue in recommendation systems [Picault et al. 2011]. For example, a person who seeks — over a period of time — to buy a car of a certain brand and model, and some time later decides to buy a motorcycle. Recommending cars may not be interesting to meet the needs of such a user. The fact that this user wanted to buy a car for a while does not mean he or she will continue to want it in the future. Therefore, user preferences can change rapidly over time, and the performance of recommendation systems depends on frequent information updates. This problem has been solved by adding a *time decay function* (*decayRec*), defined in Equation 6.

$$decayRec(s) = SC_s.\alpha^{(\theta - t_s)} \quad (6)$$

The definition of *decayRec* involves the tuple :$\langle ST, SC, DT \rangle$. ST represents a store that was recommended with a certain score (SC) at a certain moment or date (DT). Thus, the list of recommended stores takes into account two dates: the current date and the past recommendation date. The older the suggestion, the less relevant it is.

The top-N recommendation bring a list of stores. For each store we need to decrease its relevance according to the difference of the current date θ and the date t_s that the store s was recommended (both *Unix TimesTamp*). The *store score* SC_s is multiplied by the decay function $\alpha^{(\theta - t_s)}$, where $\alpha \in [0, 1[$. The variable α represents the decay coefficient. The higher the number of days, the lower the *store score* will be on the top-N list, and consequently it will lose its relevance. The α defines how quick the decay will be, the closer it is to 1, the slower it will be.

4.3.2 *User Location Monitoring.* It is not always the user is going to have the InMap application open to check the stores. For this reason, the user's location is captured every three seconds. Once the user stays more than 60 seconds in a location, we check the closest stores. Next, we calculate their frequency by the equation $termFreq$. There are two approaches commonly used to calculate distance between coordinates, these are highlighted below:

—**Haversine Formula:** Generates the distance between two points on a sphere, according to the latitude and longitude of the points. The Haversine is widely used in navigation. It is a specific case of the Haversine law of spherical trigonometry, where the sides are related to angles of a "triangular" sphere. This approach is not so accurate for short distances [Robusto 1957].

—**Vincenty's inverse formula:** Developed by Thaddeus Vincenty, this formula is used in geodesy to calculate the distance between two points on the surface of a spheroid. They are based on the assumption that the Earth is an oblate spheroid and therefore are more accurate than the methods that assume a spherical Earth. There are two ways of calculating the distance of two points using Vincenty's formula. The first method (direct) calculates the location of a point that is a given distance and an azimuth (direction) of another point. The second method (inverse) calculates the geographic distance and the azimuth between two points. The two methods have been widely used in geodesy because they are accurate at small distances in the terrestrial ellipsoid. The inverse strategy of Vincenty's formula is more precise than the Haversine formula for calculating the geodetic distance between a pair of points with latitude and longitude. It adopts an ellipsoidal model of the Earth [Vincenty 1975]. However, it has a higher computational cost.

Vincenty's inverse formula was chosen due to its high accuracy. Besides, the Android localization API supports Vincenty's inverse formula. The use of Vincenty's inverse formula was adapted as follows: for each user ($u \in U$), we want to recommend stores close to $s^{min,u} \in S$. It minimizes results that have a distance d between them. The variable u_l means the user location. The variable s_l means the store location and d represents the maximum distance (in meters) between the store and the user.

$$\forall u \in U, s^{min,u} = \arg\min_{s \in S} vincenty(u_l, s_l, d) \tag{7}$$

4.3.3 *Updating Recommendations in InMap.* It is mandatory to provide a responsive application when dealing with recommendation systems. For this reason the recommendations are pre-calculated in background. To calculate a recommendation the application must be used at least once. Such a decision preserve battery consumption and memory usage.

5. EVALUATION

The focus is on the RecStore model. The results of the recommendations generated by the RecStore were compared with the results of other recommendation models.

5.1 Dataset

During the experiments, the evaluation was performed using an offline dataset. Aiming at simulating the customer behavior, the use of application is simulated. The simulation was performed assuming different user profiles. For example, some people preferring electronic products, unlike others, who preferring natural foods.

The possible recommended stores were restricted only to the first floor of a *shopping center*. There is a relationship between the store and customer location to generate future recommendations. As aforementioned, this work does not apply hardware solutions for *indoor* localization systems. Therefore, using information from other floors of the *shopping center* resulted in a low recommendation accuracy.

Table I: Dataset of the store models and their pre-defined tags.

Name	Category(C)	Tag(SD)
Casas Bahia	12	Electronics, home appliances, Refrigerator, lcd, tv, microwave ...
Polishop	5	Electronics, furniture, appliances, tvs, lcds, dvds ...
Mega Pixel	10	Photo, machine, photographic, photos, machines ..

Table II: Example of a user model with its respective terms.

User	SP	CV	DV	LC
135	sushi, temaki ...	17, 13 ...	healthy, body ...	books, dvds ..

Table III: Dataset of recommendations suggested to the customer by InMap.

Store	Score	Registration Date
Riachuelo	0.036	1475453400
Centauro	0.020	1475280600
Saraiva	0.017	1475280000
Cacau Show	0.016	1475367000
Polishop	0.014	1475799000

The customer's route is simulated in the *shopping center*, by using a markup language called KML [1]. KLM was developed by Google, which is derived from XML (Extensible Markup Language). KML has permitted to simulate the customer's journey within the *shopping center*.

The dataset consists of 182 candidate stores with an average of 7 *tags*, 22 categories. About 15 users were considered, with an average of 7 views on the store details page, 6 searches performed, 5 selected categories and 3 visited stores. This dataset was generated manually based a real shopping center. The users were simulated aiming to create different profiles for each one.

Table I provides an example of the store model dataset layout. The first column refers to the store name, the second is the category identifier of the store. And finally, the tags define the characteristics of each store that later will be used in the RecStore recommendation model.

Table II shows a user model in the first column of which we have the user identifier followed by: the results of the searched terms (SP), identifiers of the visited categories (CV), detailed store tags (DV) and finally store tags (LC).

Table III presents the result of the RecStore recommendation model. The first column describes which store is being recommended. Each recommended store has a score that was obtained through the similarity of the user model with the model of the respective store. The Registration Date is in the Unix Timestamp format and it is used in the decay function.

5.2 Evaluation Protocol and Setting

In order to analyze the number of recommendations represented in Equation 4, 5 (five) variations were created, applying some metrics — as presented in Section 5.3. The variations are the following:

—**RecStore 1:** We disregard the frequency of terms generated by user location activity:

[1]Keyhole Markup Language

$$storeRec1(u,s) = sim(SD_u, T_s) \cdot W_{SD} + sim(SP_u, T_s) \cdot W_{SP} + termFreq(C_s, CV_u) \cdot W_{CV} \quad (8)$$

—**RecStore 2:** We remove the weights W_{SD}, W_{SP}, W_{CV} e W_{LC} :

$$storeRec2(u,s) = sim(SD_u, T_s) + sim(SP_u, T_s) + termFreq(C_s, CV_u) + termFreq(T_s, LC_u) \quad (9)$$

—**RecStore 3:** We disregard the similarity between the terms generated by the research activities:

$$storeRec3(u,s) = sim(SD_u, T_s) \cdot W_{SD} + termFreq(C_s, CV_u) \cdot W_{CV} + termFreq(T_s, LC_u) \cdot W_{LC} \quad (10)$$

—**RecStore 4:** We take into account only the frequency of the terms of the categories visited from the user's location:

$$storeRec4(u,s) = termFreq(C_s, CV_u) \cdot W_{CV} + termFreq(T_s, LC_u) \cdot W_{LC} \quad (11)$$

—**RecStore 5:** We only consider the similarity of terms generated by the search activities and store detail pages.

$$storeRec5(u,s) = sim(SD_u, T_s) \cdot W_{SD} + sim(SP_u, T_s) \cdot W_{SP} \quad (12)$$

5.2.1 *Other Models for Comparison.* The number of predicted items was configured to be among 4, 8 and 12. Multiple of 4 were used because these numbers can fit in a standard-size mobile screen. Regarding the user model, the values assigned to the composing sets were: $\{(SP : 0.6), (CV : 0.25), (SD : 0.15)\}$. Those values were experimentally configured, keeping in mind the previously defined importance of each set as $SP > CV > SD$. Three baseline models were implemented:

—*Tag-based* model [Martins et al. 2013] uses the cosine similarity between tags of the candidate store and tags of previously visualized stores, defined as $tagBased(u,s) = sim(SD_u, T_s)$.

—*Similarity-based* model that calculates the rate of the candidate store tags in the set of previously visualized stores. Simple model is a simplified version of *Tag-based* model, defined as:

$$simple(u,s) = \frac{|SD_u \cap T_u|}{|SD_u|} \quad (13)$$

—*Random* model arbitrarily recommends stores regardless any reasoning. The random model is defined as $random(u,s) = random(0..1)$.

5.3 Metric

In order to obtain a quantitative evaluation, a set of store data and user models is considered. Next, we use recommendation models collected from literature and define some variations of the RecStore model. The results are aggregated in a comprehensive way.

5.3.1 *Metrics.* As we want to evaluate the top n recommendations for each user $u \in U$, the chosen evaluation metrics are:

—**Precision** - expresses the fraction of recommendations relevant to the user, calculated as:

$$prec(u) = \frac{|R_u \cap R'_u|}{|R_u|} \tag{14}$$

—**Recall** - expresses the fraction of the relevant recommendations retrieved, calculated as:

$$rec(u) = \frac{|R_u \cap R'_u|}{|R'_u|} \tag{15}$$

—**F-measure** - the weighted harmonic mean of precision and recall, calculated as:

$$fm(u) = \frac{2 \cdot prec(u) \cdot rec(u)}{prec(u) + rec(u)} \tag{16}$$

where $|R_u|$ means the amount of retrieved recommendations for an user u. $|R'_u|$ represents the amount of *relevant* recommendations for the user u.

5.4 Results

In this section we will discuss the results of the RecStore evaluation. The previously defined metrics, the variations of the RecStore and other recommendation models were considered in this study.

5.4.1 *Analysis of RecStore Model Variations.* Figure 3 depicts the comparison of the top-10 recommendations of the RecStore model and its variations. Our proposed approach has obtained 71% precision and 69% recall, motivating its adoption. RecStore 1 obtained 65% of precision, without considering the user's location in shopping center. Figure 3 also highlights the impact of removing the weights: W_{SD}, W_{SP}, W_{CV} and W_{LC}. In addition it also shows the disregard of the similarity functions in *RecStore 2* and *RecStore 4*, respectively.

5.4.2 *Comparison Between RecStore and Other Recommendation Models.* Table IV shows the *mean* of precision, recall and f-measure along with their respective standard deviations of each model. The precision, recall and f-measure improved at rates of 118%, 76% and 95% respectively, compared

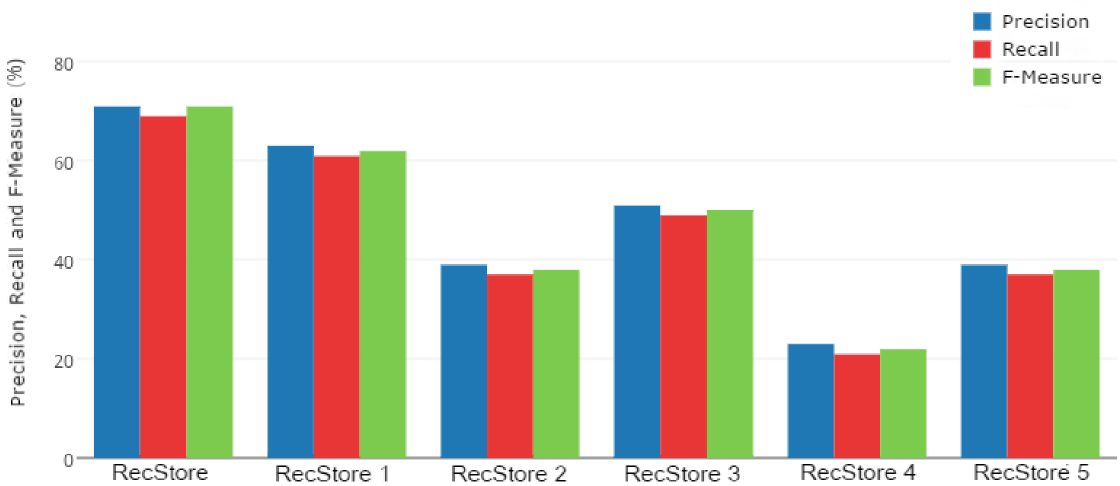


Fig. 3: Precision, Recall and F-measure, extracted from the RecStore's Top-10 Recommendations and Their Variations.

to the second best model of each metric. The tag-based model achieved better precision results than the simple model, since that approach uses cosine similarity. Cosine similarity provides a better similarity between two vectors than a simple count of occurrences. Compared to our approach, the tag-based model neglects the performed search and visited categories. The random model achieved approximately 0 precision, recall and f-measure. In our case, with top n items to be predicted (inside 10), the chances of not finding any relevant store (thus, 0 precision and recall) is around 99%, 98% and 97%, respectively.

As shown in Figure 4 the area below the ROC curve compares the different algorithms. The bigger the area, the higher the model accuracy. The stored-based model presents a bigger area under the curve. Stored-based indicates a few errors while tracing a number of correct recommendations.

The results evidences the potential of our recommendation model compared to baseline models. Also, in this study, even the recommended stores by our model, that are not candidate stores (should not be recommended), probably are not incorrect recommendations, they are just not in the top N recommendations. They could be in position N + 1 or N + 2, for example, but the chosen metrics do not take them into account.

5.4.3 *Impact Decay Factor.* The recommendation decay factor was also evaluated. The variable α (from the Equation 6) was variated by defining a 6-day interval from the store's recommendation date. Figure 5 shows the accuracy of the recommendations according to the coefficient of decay α. The most appropriate value found was $\alpha = 0.7$, and then used in our experiments.

Figure 5 shows the relationship between the customer location and the store location. As explained in Section 4.3.2, we adopted the distance between two points with their respective coordinates. As a result, the quality of the recommendation decreases for longer distances. By default, we defined by a reasonable distance of 2 meters, which obtained accuracy of 90%.

5.5 Discussion and Points for Improvement

From the results obtained (Section 5.4), our model obtained the highest rates for precision and recall in comparison to the baseline test methods. Despite the satisfactory outcome, the results still deserve some caution regarding the quality and availability of data. In a real scenario, the data from stores will not always be available and if so, they may not be public for consumption by third party applications. Even though we are developing an app that will eventually benefit the mall, the shopping administration is very unlikely to deliver its data freely without legal terms applying. The second concern is the quality of data especially when it comes from personal usage. Search terms, for instance, can be misspelled or not present the actual need of the customer. As a point of improvement we have to apply techniques that fix misspelling or simply remove words that impact on decreasing similarity calculus contributing to a bad recommendation.

Another concern is the redundancy of the data from distinct sets. It is very likely that most users will search for stores/products using single keyword queries instead of complex sentences. Having said that one would expect the search set to intersect the category set in more than 80% of cases. As a point of improvement, we have to monitor such behavior and address this issue so that the RecStore model will be enhanced. Despite the redundancy issue, we must have in mind that the search engine

Table IV: Precision, recall and f-measure means with respective standard deviation values.

Rec. Model	Precision(SD)	Recall(SD)	F-Measure(SD)
RecStore	0.59 (0.31)	0.46 (0.15)	0.51 (0.20)
Tag-based	0.27 (0.24)	0.26 (0.23)	0.26 (0.23)
Simple	0.11 (0.18)	0.11 (0.19)	0.11 (0.18)
Random	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)

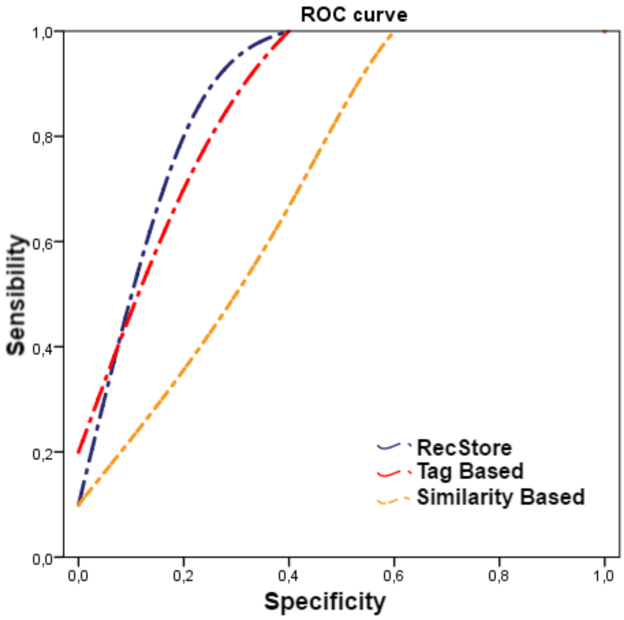


Fig. 4: ROC curve comparing RecStore, Tag-Based and Simple.

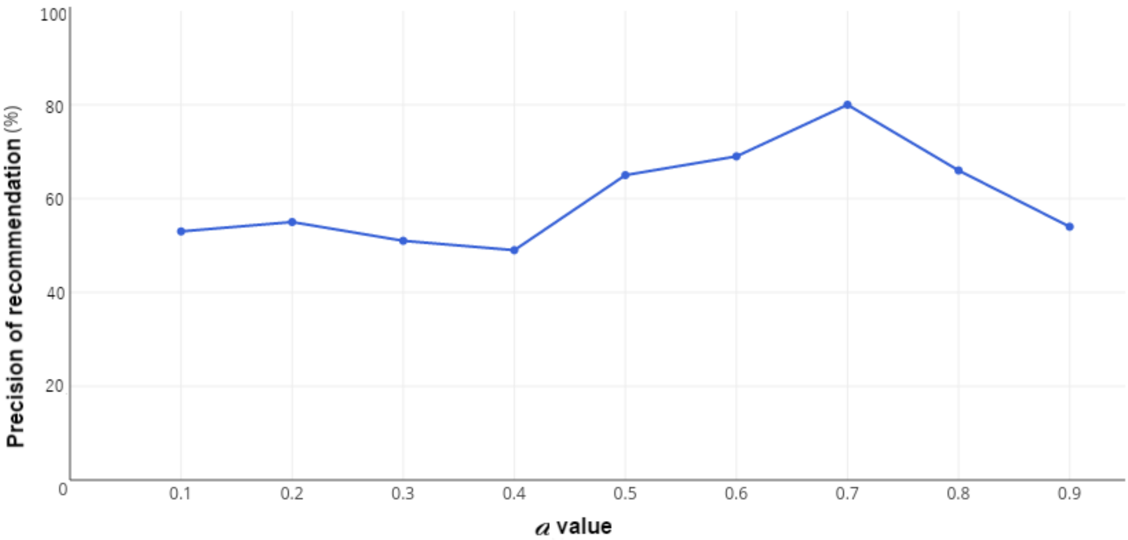


Fig. 5: Impact decay factor on the precision curve of the recommendations.

is one of the most important inputs for our system once the users are typing explicitly their needs.

Assigning weights to datasets deserves equal attention. Currently the weights are manually assigned by a system admin considering the importance of each set. The real meaning of importance is massively subjective and may vary from admin to admin. An easy solution is simply to rely on the popularity of the set usage, i.e. sets more used will receive higher rates. While it seems reasonable, it also can disadvantage users with particular ways of searching for store/products of interest. For instance, if finding products by browsing a list of stores/products is very popular, users who prefer using the search engine are very unlikely to get very precise recommendations because their queries will be weighted low in the model. This is a point of improvement that deserves lengthy monitoring before it comes to final implementation, but will definitely be developed by the simple fact that it needs to be automatized.

6. CONCLUSION

This study tackled problems that some customers handle in their daily activities to find stores in retail environments, specifically in shopping malls. We have described the motivation for creating the environmental recommendation model and reported some problems that some customers are dealing with in their daily activities to find stores in these environments. We have proposed a solution to assist users to find stores by using a recommendation system with a mobile app. Finally, we have presented how our recommendation model, RecStore, works and we have created user models based on the InMap application.

We explained the process of modeling the recommendation system and how we classified useful information for recommendations so that stores could be recommended based on customer interests. In addition, it was explained how user location tracking can help store recommendations. The RecStore model was presented in detail and submitted to an experimental evaluation. The evaluation protocol and the results were presented, showing the performance of our proposal by efficiently conducting shop recommendations according to the customers' interests. In this way, the use of mobile devices to assist in shopping activities, allows a better experience and customer satisfaction in environments such as shopping malls.

To ensure up-to-date recommendations correspond to current user interests, a RecStore recommendation template has been added to a decay function that decreases the relevance of older recommendations.

In order to evaluate the recommended model, an experimental evaluation using approximately 330 stores, 30 users and 3 baseline models was done. This evaluation achieved nearly 118% of precision improvement and 70% of advantage over the best model compared. With these results it is possible to recommend stores that are aligned with distinct customer interests, since the precision, recall and f-measure metrics obtained better results with the RecStore.

In this article the analysis was performed based only the baselines defined here. As a future work we intend to deepen our experiments to generate new analysis based also on other baselines found in the literature. Besides that, we plan to carry out more experiments to compare the RecStore model against other baseline methods and different datasets, varying data in size and content. For the selection of new datasets the number of users will be taken into account. A user study is also planned to rise the first impression of using the system in a real case scenario. We also hope to deepen our studies in future work so that this approach can be used in a real situation.

REFERENCES

Anacleto, R., Luz, N., Almeida, A., Figueiredo, L., and Novais, P. Shopping center tracking and recommendation systems. In *Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011*. Springer, pp. 299–308, 2011.

Asthana, A., Cravatts, M., and Krzyzanowski, P. An indoor wireless system for personalized shopping assistance. In *Mobile Computing Systems and Applications, 1994. Proceedings., Workshop on*. IEEE, pp. 69–74, 1994.

Baeza-Yates, R. and Ribeiro-Neto, B. *Modern Information Retrieval*. Addison Wesley, 1999.

Bajo, J., Corchado, J. M., De Paz, Y., De Paz, J. F., Rodríguez, S., Martín, Q., and Abraham, A. Shomas: Intelligent guidance and suggestions in shopping centres. *Applied Soft Computing* 9 (2): 851–862, 2009.

Bohnenberger, T., Jameson, A., Krüger, A., and Butz, A. Location-aware shopping assistance: Evaluation of a decision-theoretic approach. In *Human Computer Interaction with Mobile Devices*. Springer, pp. 155–169, 2002.

Brusilovsky, P. Adaptive hypermedia. *User Modeling and User-Adapted Interaction* 11 (1-2): 87–110, Mar., 2001.

Chang Lee, K. and Chung, N. A web dss approach to building an intelligent internet shopping mall by integrating virtual reality and avatar. *Expert Syst. Appl.* 28 (2): 333–346, Feb., 2005.

Church, K. and Smyth, B. Who, what, where & when: A new approach to mobile search. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*. IUI '08. ACM, New York, NY, USA, pp. 309–312, 2008.

Fang, B., Liao, S., Xu, K., Cheng, H., Zhu, C., and Chen, H. A novel mobile recommender system for indoor shopping. *Expert Systems with Applications* 39 (15): 11992–12000, 2012.

Hennig-Thurau, T. and Klee, A. The impact of customer satisfaction and relationship quality on customer retention: A critical reassessment and model development. *Psychology & Marketing* 14 (8): 737–764, 1997.

Holding, N. Nielsen ibope, 2015.

Huang, H., Zhao, B., Zhao, H., Zhuang, Z., Wang, Z., Yao, X., Wang, X., Jin, H., and Fu, X. A cross-platform consumer behavior analysis of large-scale mobile shopping data. In *Proceedings of the 2018 World Wide Web Conference*. WWW '18. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, pp. 1785–1794, 2018.

Hussein, G. Mobile recommender system analysis and design. *2009 1st International Conference on Networked Digital Technologies, NDT 2009*, 08, 2009.

Kim, K.-J. and Kim, B.-G. Product recommender system for online shopping malls using data mining techniques. *Journal of Intelligence and Information Systems* vol. 11, 01, 2005.

Kim, T. and Park, S. C. One-to-one customized brand recommendation in virtual shopping mall. In *Management of Innovation and Technology, 2000. ICMIT 2000. Proceedings of the 2000 IEEE International Conference on*. Vol. 2. pp. 659–663 vol.2, 2000.

Kramer and Modsching, R. Context-aware adaptation in a mobile tour guide. *Proc. of the 5th International and Interdisciplinary Conference on Modeling and Using Context*, 2005.

Lemos, F. and Goes, L. F. Evaluation of consumer shopping behavior in m-commerce and e-commerce. In *Proceedings of the Annual Conference on Brazilian Symposium on Information Systems: Information Systems: A Computer Socio-Technical Perspective - Volume 1*. SBSI 2015. Brazilian Computer Society, Porto Alegre, Brazil, Brazil, pp. 18:127–18:134, 2015.

Marinho, L. B., Nunes, I., Sandholm, T., Nóbrega, C., Araújo, J. a., and Pires, C. E. S. Improving location recommendations with temporal pattern extraction. In *Proceedings of the 18th Brazilian Symposium on Multimedia and the Web*. WebMedia '12. ACM, New York, NY, USA, pp. 293–296, 2012.

Martin, B. and Mason, S. Current trends in leisure. *Leisure Studies* vol. 6, pp. 93–97, 1987.

Martins, E. F., Belém, F. M., Almeida, J. M., and Gonçalves, M. Measuring and addressing the impact of cold start on associative tag recommenders. In *Proceedings of the 19th Brazilian Symposium on Multimedia and the Web*. WebMedia '13. ACM, New York, NY, USA, pp. 325–332, 2013.

Miller, D. *A theory of shopping*. Cornell University Press, 1998.

Pessemier, T. D., Deryckere, T., and Martens, L. Extending the bayesian classifier to a context-aware recommender system for mobile devices. In *2010 Fifth International Conference on Internet and Web Applications and Services*. pp. 242–247, 2010.

Picault, J., Ribière, M., Bonnefoy, D., and Mercer, K. How to get the recommender out of the lab? In *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor (Eds.). Springer US, pp. 333–365, 2011.

Ricci, F. Mobile recommender systems. *Information Technology & Tourism* 12 (3): 205–231, 2010.

Robusto, C. C. The cosine-haversine formula. *The American Mathematical Monthly* 64 (1): 38–40, 1957.

Sae-Ueng, S., Pinyapong, S., Ogino, A., and Kato, T. Consumer-friendly shopping assistance by personal behavior log analysis on ubiquitous shop space, 2008.

Senot, C., Kostadinov, D., Bouzid, M., Picault, J., Aghasaryan, A., and Bernier, C. Analysis of strategies for building group profiles. In *User Modeling, Adaptation, and Personalization*, P. De Bra, A. Kobsa, and D. Chin (Eds.). Lecture Notes in Computer Science, vol. 6075. Springer Berlin / Heidelberg, pp. 40–51, 2010.

Silva, D. V. d. S., Silva, R. d. S., and Durão, F. A. Recstore: Recommending stores for shopping mall customers. In *Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web. https://doi.org/10.1145/3126858.3126888*. WebMedia '17. ACM, New York, NY, USA, pp. 117–124, 2017.

Stahl, C., Baus, J., Brandherm, B., Schmitz, M., and Schwartz, T. Navigational-and shopping assistance on the basis of user interactions in intelligent environments, 2005.

Vincenty, T. Direct and inverse soluctions of geodesics on the ellipsoid with application of nested equations. *Directorate of Overseas Surveys* 23 (176): 89–90, 1975.

Von Reischach, F., Guinard, D., Michahelles, F., and Fleisch, E. A mobile product recommendation system interacting with tagged products. In *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*. IEEE, pp. 1–6, 2009.

Yang, W.-S., Cheng, H.-C., and Dia, J.-B. A location-aware recommender system for mobile shopping environments. *Expert Systems with Applications* 34 (1): 437–445, 2008.

Zhu, H. and Tian, E. C. H. X. K. Y. H. C. J. Mining mobile user preferences for personalized context-aware recommendation. *ACM Trans. Program. Lang. Syst.*, December, 2014.