Mobile Recommendations for Shopping Mall Customers

Gustavo D. Carvalho
Department of Computer Science
University Federal of Bahia
Salvador,Bahia, Brazil
gdacarv@gmail.com

Renato S. Silva
Department of Computer Science
University Federal of Bahia
Salvador,Bahia, Brazil
renato.silva@dcc.ufba.br

Frederico A. Durão
Department of Computer Science
University Federal of Bahia
Salvador,Bahia, Brazil
freddurao@dcc.ufba.br

Diogo Vinícius de S. Silva
Department of Computer Science
University Federal of Bahia
Salvador,Bahia, Brazil
diogovss@ufba.br

ABSTRACT

Advances in technology allows us to carry a computer in our pockets. Smartphones are a tendency and almost mandatory to anyone living in an urban and modern context. Considering this, we realize malls are indispensable to our society. Nevertheless, with hundred of stores and products, people tend to loose themselves or to waste a lot of time finding something in the midst of its hugeness. This paper proposes a model to assist and to recommend customers to find what they consider relevant at malls. Using a mobile application, InMap, the model does recommendations based on user activities and they rely on content-based techniques that provide the most relevant results.

Keywords

recommendation, shopping, mall, mobile, customization

1. INTRODUCTION

Mobile applications can use more contextual information, such as time and geographic location, than desktop applications. In this sense, mobile applications can provide users with just-in-time information about several places. For instance: airports, universities, amusement parks and shopping malls.

Shopping malls are attractive places, where many people buy products, use public services or just have fun. Depending on the culture, spending time at malls is placed among the top choices for spending free time, and it is justified because of security and convenience. Even the act of shopping is increasingly seen as a leisure activity [8] rather than just provisioning of required goods.

With overcrowd of products and stores, it is more and more difficult to find relevant items, and this is a major issue in a shopping mall. With hundreds of stores and thousands of products, customers may have problems finding what they want or need. Therefore, the problem we try to solve is: How to recommend the best stores that are aligned to different customers interests and needs? This problem can be divided in smalls problems as: How classify a customer using only informations accessible by the application? How to classify and categorize stores? Which store should be recommended to client? This work is structured as follows: Section 2 positions our work with respect to related findings in the literature. Section 3 introduces the InMap application and its various features. Section 4 depicts the composition of user model and presents the store recommendation model and how the recommendations are generated. Section 5 describes the experimental setup and the results achieved from the experiments. Section 6 we discuss the final remarks, future work and acknowledgments.

2. RELATED WORK

There is a lot of studies about the web-based recommendation for purchase, However, little about shopping malls. Many of them try to overcome the problem of location. But, most are recommendation of products using information gathered from context, others users and their preferences. [10] and [1] presents a summary of mobile and shopping recommendations systems, the challenges and opportunities, the former focused on mobile tourism recommendations and the latter on shopping centers.

[2] approach is focused on retail stores like supermarkets, where a specific hardware is attached to the cart and it has access to your history, provides product recommendations, compares different brands of the same product and indicates location of each product. Unfortunately, this model presents high infrastructure costs. [12] presents a pervasive solution of navigational and shopping assistance using user personal device and an intelligent environment. It claims that portable devices are not good to provide navigational assistance, so the suggestion is to use in-place displays to guide the user. The downside is scalability. It can work nicely with few users, but if dozens of users need to use at the same time, it's not possible. Both studies suffer for hardware cost because the need of sensors, displays and specific devices attached on each cart, as is the case of [2]. So

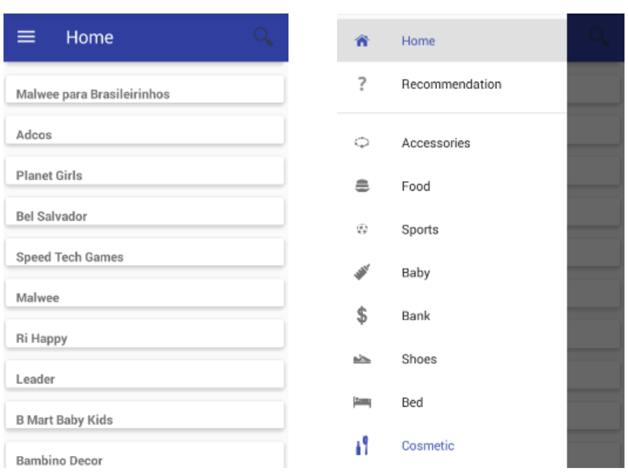
In: Sessão de Posteres., 2016, Teresina. Anais do XXII Simpósio Brasileiro de Sistemas Multimídia e Web. Porto Alegre: Sociedade Brasileira de Computação, 2016. v. 2.
ISBN: 978-85-7669-332-1

they differ from our approach in the sense that we only use hardware devices already owned by mainstream users.

Product recommendation is the focus of [13]. Using their software and products with bar code or RFID the users can review products and receive recommendations based on others users reviews. Also using RFID there is SHOMAS software [4], a multi-agent system that provides navigational assistance and suggestions in a shopping using RFID to get user location. Both of them are focused on products instead of stores, in our case. Also, they suffer a lack of hardware support and installation such as RFID tags. [14] presents a location-aware recommendation system that analyzes past customer accesses to web pages to compose the user profile, and then recommends nearby stores web site. Others ways to get user's location are presented by [5] which uses beacons affixed to the walls in strategic places to provide information to the system and [7] which uses RSS (received signal strength) to assert user position.[15] works with detection of information through logs on the device, making the mining of preference-sensitive mobile users context for personalized recommendation context aware and other related services. Although the related works show convergences with our approach, not found out in the literature any recommendations on mobile environments that particularly shopping malls, using the user activities on application to build the user model and provide users with personalized recommendations.

3. INMAP - THE MOBILE APPLICATION

The mobile application InMap provide customers a set of features that help them shop in a shopping mall. Figure 1(a) shows a stores list recommended. Can make keyword queries. For example, typing "shoes" in the text field is shown a stores list where can find the product. Figure 2(b) shows that is also possible to search a shop by category, for each option selected, the corresponding stores list categorized is shown. Thus, all user activities are used in background to create recommendations customizing the stores list Figure 1(a).



(a) customized stores (b) Store Categories

Figure 1: InMap Screenshot

4. THE INMAP RECOMMENDER

4.1 The User Model

Our user model follows [6]'s methodology by 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 [11] as a tuple $\langle SP, SD, CV \rangle$, where:

- SP represents the set of weighted terms that describe a *search performed* by an user. Once a search indicate a declared interest in a product or store, it provides strong evidence of which store an user might be interested in.
- SD represents the set of weighted terms from *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's categories visited*. This set is a strong indicator of user interests once it clearly exposes the customer's will in that category of store.

Our study focuses for implicit feedback using interface navigation history and search patterns.

4.1.1 Weighing the User Model

The terms and the sets serve as our learning units of users preferences, thereby properly accounting for the importance of each set, and each term becomes crucial for recommendations success. Within each set, we understand that some terms can be more representative than others, meaning that the *frequency* of a term can denote its importance to the set it belongs. For instance, suppose an user has visited the categories: "Clothing" and "Cosmetic", 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 "Cosmetic". Then, the term frequency of the categories visited CV={*Clothing, Clothing, Cosmetic, Clothing*} will be represented as {*("Clothing",0.75),("Cosmetic",0.25)*}. The term frequency is defined as:

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

where n_t is the number of occurrences of the term $t \in T_s$, 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 The Set Weighing Model

Similar to terms, each set has its own user model's importance. For instance, the set SP, corresponding to the search performed, better exposes the user's need rather than the store details page view set SD, because the latter basically shows the user's curiosity on a given store, not necessarily a real need. A store detail page, an user can view the page just to know more about what kind of product the store sells.

Unlike terms, the importance of a set is not calculated by a mathematical equation; instead, it is empirically predefined based on the system administrator's common knowledge. Our experiments (see Section 5), we suggest the most appropriate weights respecting the following order of importance:

$$\rho = SP > CV > SD. \quad (2)$$

4.2 Store Model

The store model SM_s comprises terms describing the store $s \in S$. Our context, the relevant information that are considered in SM_s includes its category and tags. The category tells what the store is about and summarizes the key products or services available in the store. Tags assigned to the stores by us, usually tries to categorize or conceptualize the store though keywords that best describe its content.

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

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

The function *storeRec* is described as:

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

where W_{SD}, W_{SP} and W_{CV} are the normalized weights (importance) of each set on our recommendation model (see Section 4.1.2), *sim* is a similarity function utilized to compute the similarity between the user model set SD_u or SP_u of an 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 user's categories visited CV_u and store category C_s (see Section 4.1.1). As seen, by setting up proper weights, the *storeRec* equation tries to privilege the similarity between store and user model by user's search performed, which is the better expression of interest. These user models sets SD_u and SP_u and the store models set T_s are represented as $\overrightarrow{SD_u}, \overrightarrow{SP_u}$ and $\overrightarrow{T_s}$ respectively. Technically, we calculate the cosine similarity [3] between the vectors 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 above described vectors comprise real numbers (weights) in which each value (normalized [0,1]) measures the importance of the corresponding term to the user or store. Also, as said in Section 4.1, before any calculation of (3) a decay factor applies, deleting any information of SD_u, SP_u and CV_u older than two months.

5. EVALUATION

We calculate the recommended stores provided by our store-based model, and others algorithms such as baseline, compare with the expected output and calculate appropriate metrics of evaluation.

5.1 Dataset

We were unable to use real user models while doing experiments. To solve this problem, it was simulated some real usage of the application in order to generate user models to be used in the evaluation process. The simulation was performed assuming different profiles. we use the recommendation model (4) to assign which stores should be recommended to each user model and the result was used as dataset in our evaluation.

5.2 Evaluation Protocol and Setting

As we want to evaluate the top n recommendations for each user $u \in U$, the appropriate evaluation metrics chosen are *precision*, *recall* and *f-measure*. Precision expresses the fraction of recommendations relevant to the user whereas recall expresses the fraction of the relevant recommendations retrieved. We calculated the precision and recall respectively as: $prec(u) = \frac{|R_u \cap R'_u|}{|R_u|}$ and $rec(u) = \frac{|R_u \cap R'_u|}{|R'_u|}$, where $|R_u|$ is the amount of retrieved recommendations for an user u while $|R'_u|$ is the amount of *relevant* recommendations for the user u. Additionally, we calculated the f-measure, the weighted harmonic mean of precision and recall as $fm(u) = \frac{2 \cdot prec(u) \cdot rec(u)}{prec(u) + rec(u)}$.

We have experimented with the number of top n items to be predicted as 4, 8 and 12. We have chosen values multiple of 4 because that is the average recommendations which can fit in a standard-size mobile screen. As to the user model, the values assigned to the composing sets were: $\{(SP : 0.6), (CV : 0.25), (SD : 0.15)\}$. Those values was set experimentally, adjusting towards the best result, keeping in mind the previously defined importance of each set as $SP > CV > SD$. Three baseline models were implemented:

- *Tag-based* model(see [9]) that uses the cosine similarity between tags of the candidate store and tags of previously stores visualized by the user, defined as $tagBased(u, s) = sim(SD_u, T_s)$.
- *Simple* model that calculate the rate of the candidate store tags in the set of previously stores visualized by the user. This is a simplified version of *Tag-based* model, defined as $simple(u, s) = \frac{|SD_u \cap T_u|}{|SD_u|}$.
- *Random* model that arbitrarily recommends stores regardless any reasoning is $random(u, s) = random(0..1)$.

5.3 Results

Table 1: Precision, recall and f-measure means with respective standard deviation values

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

Table 1 shows the *mean* of precision, recall and f-measure along with their respective standard deviations of each model. As shown, the precision, recall and f-measure improved at rates of 118%, 70% and 88% respectively in comparison to the second best model of each metric. The tag-based model achieved better precision results than the simple model because that approach uses cosine similarity, which provides a better similarity between two vectors than a simple count of occurrences. Compared to our approach, the tag-based model neglects the search performed and categories visited by the user, which provides valuable information about user's needs. Our results, the random model achieved approximately 0 precision, recall and, therefore, f-measure, because, in our case with top n items to be predicted as 4, 8 and 12, the chances of not find any relevant store (thus, 0 precision and recall) is 99%, 98% and 97%, respectively.

Figure 2, we use the area under the ROC curve to compare the different algorithms. Which the larger area model accuracy will be better. It is noted that the Store-based has a greater area under the curve. We can say that the Store-based has few errors while making many correct recommendations of the models Tag-Based and Simple.

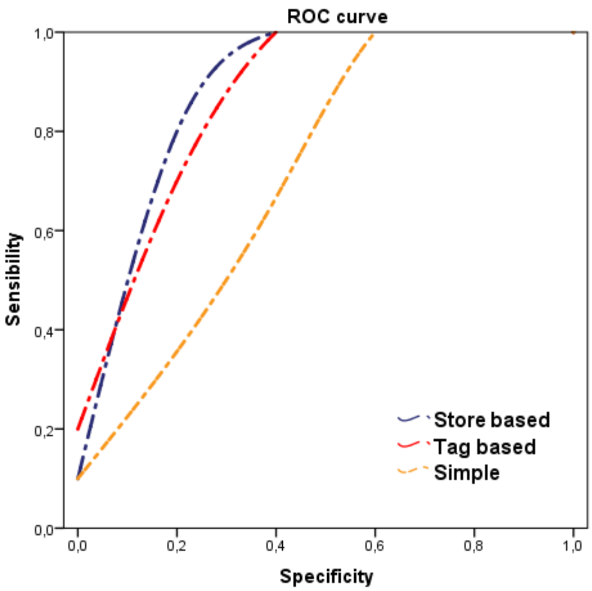


Figure 2: ROC curve comparing Store-Based, Tag-Based and Simple.

The results show us the potential of our recommendation model with significantly improvement 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, it could be in position N + 1 or N + 2, for example, but the chosen metrics do not take them into account.

6. CONCLUSION

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 thereabout 118% of precision improvement and 70% of advantage over the best model compared.

As future work, we intend to evaluate qualitatively the recommendations using real users moreover, to extend the user model by making use of user's account in social networks. Since knowing the user location could improve the model, one of the substantial improvements is being aware of their physical location.

7. REFERENCES

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

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

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

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

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

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

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

[8] B. Martin and S. Mason. Current trends in leisure. *Leisure Studies*, 6:93–97, 1987.

[9] E. F. Martins, F. M. Belém, J. M. Almeida, and M. Gonçalves. 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, pages 325–332, New York, NY, USA, 2013. ACM.

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

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

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

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

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

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