Hybrid-ε-greedy for Mobile Context-Aware Recommender System

Djallel Bouneffouf, Amel Bouzeghoub, and Alda Lopes Gançarski

Department of Computer Science, Télécom SudParis,
UMR CNRS Samovar, 91011 Evry Cedex, France
{Djallel.Bouneffouf,Amel.Bouzeghoub,
Alda.Gancarski}@it-sudparis.eu

Abstract. The wide development of mobile applications provides a considerable amount of data of all types. In this sense, Mobile Context-aware Recommender Systems (MCRS) suggest the user suitable information depending on her/his situation and interests. Our work consists in applying machine learning techniques and reasoning process in order to adapt dynamically the MCRS to the evolution of the user's interest. To achieve this goal, we propose to combine bandit algorithm and case-based reasoning in order to define a contextual recommendation process based on different context dimensions (social, temporal and location). This paper describes our ongoing work on the implementation of a MCRS based on a hybrid-ε-greedy algorithm. It also presents preliminary results by comparing the hybrid-ε-greedy and the standard ε-greedy algorithm.

Keywords: Machine learning, contextual bandit, personalization, recommender systems, exploration/exploitation dilemma.

1 Introduction

Mobile technologies have made access to a huge collection of information, anywhere and anytime. Thereby, information is customized according to users' needs and preferences. This brings big challenges for the Recommender System field. Indeed, technical features of mobile devices yield to navigation practices which are more difficult than the traditional navigation task.

A considerable amount of research has been done in recommending relevant information for mobile users. Earlier techniques [8, 10] are based solely on the computational behavior of the user to model his interests regardless of his surrounding environment (location, time, near people). The main limitation of such approaches is that they do not take into account the dynamicity of the user's context. This gives rise to another category of recommendation techniques that tackle this limitation by building situation-aware user profiles. However, these techniques have some problems, namely how to recommend information to the user in order to follow the evolution of his interest.

In order to give Mobile Context-aware Recommender Systems (MCRS) the capability to provide the mobile user information matching his/her situation and adapted to the evolution of his/her interests, our contribution consists of mixing bandit algorithm (BA) and case-based reasoning (CBR) methods in order to tackle these two issues:

P.-N. Tan et al. (Eds.): PAKDD 2012, Part I, LNAI 7301, pp. 468–479, 2012.

- Finding situations that are similar to the current one (CBR);
- Making the deal between exploring the user interests and recommending the most relevant content according to the current situation (BA).

The remainder of the paper is organized as follows. Section 2 reviews some related works. Section 3 presents the proposed recommendation algorithm. The experimental evaluation is described in Section 4. The last Section concludes the paper and points out possible directions for future work.

2 Background

We reference in the following recent relevant recommendation techniques that tackle the both issues namely: following the evolution of user's interests and managing the user's situation.

2.1 Following the Evolution of User's Interests

The trend today on recommender systems is to suggest relevant information to users, using supervised machine learning techniques. In these approaches, the recommender system has to execute two steps: (1) The learning step, where the system learns from samples and gradually adjusts its parameters; (2) The exploitation step, where new samples are presented to the system to perform a generalization [14].

These approaches suffer from difficulty in following the evolution of the user's interests. Some works found in the literature [3, 11] address this problem as a need for balancing exploration and exploitation studied in the "bandit algorithm". A bandit algorithm B exploits its past experience to select documents that appear more frequently. Besides, these seemingly optimal documents may in fact be suboptimal, due to imprecision in B's knowledge. In order to avoid this undesired situation, B has to explore documents by actually choosing seemingly suboptimal documents so as to gather more information about them. Exploitation can decrease short-term user's satisfaction since some suboptimal documents may be chosen. However, obtaining information about the documents' average rewards (i.e., exploration) can refine B's estimate of the documents' rewards and in turn increase long-term user's satisfaction. Clearly, neither a purely exploring nor a purely exploiting algorithm works best in general, and a good tradeoff is needed. The authors on [3, 11] describe a smart way to balance exploration and exploitation in the field of recommender systems. However, none of them consider the user's situation during the recommendation.

2.2 Managing the User's Situation

Few research works are dedicated to manage the user's situation on recommendation. In [1, 4,5] the authors propose a method which consists of building a dynamic situation and user profile based on time and user's experience. The user's preferences and interests in the user profile are weighted according to the situation (time, location) and user behavior. To model the change on user's preferences according to his temporal situation in different periods, like workday or vacations, the weighted association for the concepts in the user profile is established for every new experience of the user. The user activity combined with the user profile are used together to filter and recommend relevant content.

Another work [2] describes a MCRS operating on three dimensions of context that complement each other to get highly targeted. First, the MCRS analyzes information such as clients' address books to estimate the level of social affinity among users. Second, it combines social affinity with the spatiotemporal dimensions and the user's history in order to improve the quality of the recommendations.

Each work cited above tries to recommend interesting information to users on contextual situation; however they do not consider the evolution of the user's interest.

To summarize, none of the mentioned works tackles both problems. This is precisely what we intend to do with our approach, exploiting the following new features:

- Inspired by models of human reasoning developed by [7] in robotic, we propose to consider the user's situation in the bandit algorithm by using the case-based reasoning technique, which is not considered in [3, 4, 14].
- In [3, 14] authors use a smart bandit algorithm to manage the exploration/exploitation strategy, however they do not take into account the content in the strategy. Our intuition is that, considering the content when managing the exploration/exploitation strategy will improve it. This is why we propose to use content-based filtering techniques together with ε-greedy algorithm.

In what follows, we summarize the terminology and notations used in our contribution, and then we detail our methods for inferring the recommendation.

3 The Proposed MCRS Algorithm

3.1 Terminology and Notations

User Profile. The user profile is composed of the user's personal data and other dynamic information, including his preferences, his calendar and the history of his interactions with the system.

User Preferences. Preferences are deduced during user navigation activities. They contain the set of navigated documents during a situation. A navigation activity expresses the following sequence of events: (i) the user logs in the system and navigates across documents to get the desired information; (ii) the user expresses his/her preferences on the visited documents. We assume that a visited document is relevant, and thus belongs to the user's preferences, if there are some observable user's behaviors through 2 types of preference:

- The direct preference: the user expresses his interest in the document by inserting a rate, like for example putting stars ("*") at the top of the document.
- The indirect preference: it is the information that we extract from the user system interaction, for example the number of clicks or the time spent on the visited documents.

Let UP be the preferences submitted by a specific user to the system at a given situation. Each document in UP is represented as a single vector $d=(c_1,...,c_n)$, where c_i (i=1, .., n) is the value of a component characterizing the preferences of d. We consider the following components: the total number of clicks on d, the total time spent reading d, the number of times d was recommended, and the direct preference rate on d.

History. All the interactions between the user and the system are stored together with the corresponding situations in order to exploit this data to improve the recommendation process.

Calendar. The user's calendar has information concerning the user's activities, like meetings. Time and location information is automatically inferred by the system.

User Situation. A situation *S* is represented as a triple whose features *X* are the values assigned to each dimension: $S = (X_l, X_t, X_s)$, where X_l (resp. X_t and X_s) is the value of the location (resp. time and social) dimension.

Suppose the user is associated to: the location "48.8925349, 2.2367939" from his phone's GPS; the time "Mon Oct 3 12:10:00 2011" from his phone's watch; and the meeting with Paul Gerard from his calendar. To build the situation, we associate to this kind of low level data, directly acquired from mobile devices capabilities, more abstracted concepts using ontologies reasoning means.

- **Location:** We use a local spatial ontology to represent and reason on geographic information. Using this ontology, for the above example, we get, from location "48.8925349, 2.2367939", the value "Paris" to insert in the location dimension of the situation.
- **Time:** To allow a good representation of the temporal information and its manipulation, we propose to use OWL-Time ontology [6] which is today a reference for representing and reasoning about time. We propose to base our work on this ontology and extend it if necessary. Taking the example above, for the time value "Mon Oct 3 12:10:00 2011", we get, using the OWL-Time ontology, the value "workday".
- **Social connection:** The social connection refers to the information of the user's interlocutors (e.g. a friend, an important customer, a colleague or his manager). We use the FOAF Ontology [9] to describe the social network by a set of concepts and properties. For example, the information about "the meeting with Paul Gerard" can yield the value "wine client" for the social dimension.

3.2 The Bandit Algorithm

In our MCRS, documents' recommendation is modeled as a multi-armed bandit problem. Formally, a bandit algorithm proceeds in discrete trials t = 1,…T. For each trial t, the algorithm performs the following tasks:

- Task 1. It observes the current user u_t and a set A_t of arms together with their feature vectors $x_{t,a}$ for $a \in A_t$. The vector $x_{t,a}$ summarizes information of both user u_t and arm a, and is referred to as the context.

- Task 2. Based on observed rewards in previous trials, it chooses an arm $a_t \in A_t$, and receives reward r_{t,a_t} whose expectation depends on both the user u_t and the arm a_t.
- Task 3. It improves its arm-selection strategy with the new observation, $(x_{t,a_t}, a_t, r_{t,a_t})$. It is important to emphasize here that no feedback (namely the reward $r_{t,a}$) is observed for unchosen arms $a \neq a_t$.

In tasks 1 to 3, the total T-trial reward of A is defined as $\sum_{t=1}^{T} r_{t,a_t}$ while the optimal expected T-trial reward is defined as $\mathrm{E}\left[\sum_{t=1}^{T} r_{t,a_t^*}\right]$ where a_t^* is the arm with maximum expected reward at trial t. Our goal is to design the bandit algorithm so that the expected total reward is maximized.

In the field of document recommendation, we may view documents as arms. When a document is presented to the user and this one selects it by a click, a reward of 1 is incurred; otherwise, the reward is 0. With this definition of reward, the expected reward of a document is precisely its Click Through Rate (CTR). The CTR is the average number of clicks on a recommended document, computed diving the total number of clicks on it by the number of times it was recommended. Consequently, choosing a document with maximum CTR is equivalent, in our bandit algorithm, to maximizing the total expected rewards.

3.3 The Proposed Hybrid-ε-greedy Algorithm

There are several strategies which provide an approximate solution to the bandit problem. Here, we focus on two of them: the greedy strategy, which always chooses the best arms, thus uses only exploitation; the ε-greedy strategy, which adds some greedy exploration policy, choosing the best arms at each step if the policy returns the greedy arms (probability = ε) or a random arms otherwise (probability = 1 – ε).

We propose a two-fold improvement on the performance of the ε-greedy algorithm: integrating case base reasoning (CBR) and content based filtering (CBF). This new proposed algorithm is called hybrid-ε-greedy and is described in (Alg. 3).

To improve exploitation of the ε-greedy algorithm, we propose to integrate CBR into each iteration: before choosing the document, the algorithm computes the similarity between the present situation and each one in the situation base; if there is a situation that can be re-used, the algorithm retrieves it, and then applies an exploration/exploitation strategy.

In this situation-aware computing approach, the premise part of a case is a specific situation S of a mobile user when he navigates on his mobile device, while the value part of a case is the user's preferences UP to be used for the recommendation. Each case from the case base is denoted as C= (S, UP).

Let $S^c=(X_l^c, X_t^c, X_s^c)$ be the current situation of the user, UP^c the current user's preferences and $PS=\{S^1,....,S^n\}$ the set of past situations. The proposed hybrid-ε-greedy algorithm involves the following four methods.

RetrieveCase() (Alg. 3)

Given the current situation S^c, the RetrieveCase method determines the expected user preferences by comparing S^c with the situations in past cases in order to choose the most similar one S^s. The method returns, then, the corresponding case (S^s, UP^s).

S^s is selected from PS by computing the following expression as it done in [4]:

$$S^S = \arg\max_{S^i \in PS} \left(\sum_j \alpha_j \cdot sim_j \left(X_j^c, X_j^i \right) \right) \tag{1}$$

In equation 1, sim_j is the similarity metric related to dimension j between two situation vectors and α_j the weight associated to dimension j. α_j is not considered in the scope of this paper, taking a value of 1 for all dimensions.

The similarity between two concepts of a dimension j in an ontological semantic depends on how closely they are related in the corresponding ontology (location, time or social). We use the same similarity measure as [12] defined by equation 2:

$$sim_j \left(X_j^c, X_j^i \right) = 2 * \frac{deph(LCS)}{(deph(X_j^c) + deph(X_j^i))} \tag{2}$$

Here, LCS is the Least Common Subsumer of X_j^c and X_j^i, and *depth* is the number of nodes in the path from the node to the ontology root.

RecommendDocuments() (Alg. 3)
In order to insure a better precision of the recommender results, the recommendation takes place only if the following condition is verified: $sim(S^c, S^s) \geq B$ (Alg. 3), where B is a threshold value and

$$sim(S^c, S^s) = \sum_j sim_j \left(X_j^c, X_j^s \right)$$

In the RecommendDocuments() method, sketched in Algorithm 1, we propose to improve the ε-greedy strategy by applying CBF in order to have the possibility to recommend, not the best document, but the most similar to it (Alg. 1). We believe this may improve the user's satisfaction.

The CBF algorithm (Alg. 2) computes the similarity between each document $d=(c_1,..,c_k)$ from UP (except already recommended documents D) and the best document $d^b=(c_j^b ,.., c_k^b)$ and returns the most similar one. The degree of similarity between d and d^b is determined by using the cosine measure, as indicated in equation 3:

$$\cos sim(d, d^b) = \frac{d \cdot d^b}{\|d\| \cdot \|d^b\|} = \frac{\sum_k c_k \cdot c_k}{\sqrt{\sum_k c_k^{b2} \cdot \sum_k c_k^{b2}}} \tag{3}$$

```
Algorithm 1. The RecommendDocuments() method
Input: ε, UP^c, N
Output:  D
D = Ø
For i=1 to N do
     q = Random({0, 1})
     j = Random({0, 1})
           ⎧argmax_d∈ (UP-D)  (getCTR(d))                  if j<q<ε
     d_i= ⎨CBF(UP^c-D, argmax_d∈ (UP-D) (getCTR(d))      if q≤j≤ε
           ⎩Random(UP^c)                                   otherwise
     D = D ∪ {d_i}
Endfor
Return D
```

```
Algorithm 2. The CBF() method
Input: UP, d^b
Output: d^s
d^s= argmax_{d∈(UP)}(cossim(d^b, d))
Return d^s
```

UpdateCase() & InsertCase().
After recommending documents with the RecommendDocuments method (Alg. 3), the user's preferences are updated w. r. t. number of clicks and number of recommendations for each recommended document on which the user clicked at least one time. This is done by the UpdatePreferences function (Alg. 3).

Depending on the similarity between the current situation S^c and its most similar situation S^s (computed with RetrieveCase()), being 3 the number of dimensions in the context, two scenarios are possible:

- sim(S^c, S^s) ≠ 3: the current situation does not exist in the case base (Alg. 3); the InsertCase() method adds to the case base the new case composed of the current situation Sc and the updated UP.
- sim(S^c, S^s) = 3: the situation exists in the case base (Alg. 3); the UpdateCase() method updates the case having premise situation S^c with the updated UP.

```
Algorithm 3. hybrid-ε-greedy algorithm
Input:  B, ε, N, PS, S^s, UP^s, S^c, UP^c
Output: D
D = Ø
(S^s, UP^s) = RetrieveCase(S^c, PS)
if sim(S^c,S^s) ≥ B then
      D = RecommendDocuments(ε, UP^s, N)
      UP^c = UpdatePreferences(UP^s, D)
      if sim(S^c, S^s) ≠ 3 then
             PS = InsertCase(S^c, UP^c)
      else
             PS = UpdateCase(S^p, UP^c)
      end if
else     PS = InsertCase(S^c, UP^c);
end if
Return D
```

4 Experimental Evaluation

In order to empirically evaluate the performance of our algorithm, and in the absence of a standard evaluation framework, we propose an evaluation framework based on a diary study entries. The main objectives of the experimental evaluation are: (1) to find the optimal threshold B value of step 2 (Section 3.3) and (2) to evaluate the performance of the proposed hybrid ε-greedy algorithm (Alg. 3) w. r. t. the optimal ε value and the dataset size. In the following, we describe our experimental datasets and then present and discuss the obtained results.

4.1 Experimental datasets

We conducted a diary study with the collaboration of the French software company Nomalys. To allow us conducting our diary study, Nomalys decides to provide the “Ns” application of their marketers a history system, which records the time, current location, social information and the navigation of users when they use the application during their meetings (social information is extracted from the users’ calendar).

The diary study took 8 months and generated 16 286 diary situation entries. Table 1 illustrates three examples of such entries where each situation is identified by IDS.

Table 1. Diary situation entries

IDS	Users	Time	Place	Client
1	Paul	11/05/2011	75060 Paris	NATIXIS
2	Fabrice	15/05/2011	59100 Roubaix	MGET
3	Jhon	19/05/2011	75015 Paris	AUNDI

Each diary situation entry represents the capture, for a certain user, of contextual information: time, location and social information. For each entry, the captured data are replaced with more abstracted information using the ontologies. For example the situation 1 becomes as shown in Table 2.

Table 2. Semantic diary situation

IDS	Users	Time	Place	Client
1	Paul	Workday	Paris	Finance client
2	Fabrice	Workday	Roubaix	Social client
3	Jhon	Holiday	Paris	Telecom client

From the diary study, we obtained a total of 342 725 entries concerning user navigation, expressed with an average of 20.04 entries per situation. Table 3 illustrates an example of such diary navigation entries. For example, the number of clicks on a document (Click), the time spent reading a document (Time) or his direct interest expressed by stars (Interest), where the maximum stars is five.

Table 3. Diary navigation entries

IdDoc	IDS	Click	Time	Interest
1	1	2	2’	**
2	1	4	3’	***
3	1	8	5’	*****

4.2 Finding the Optimal B Threshold Value

In order to evaluate the precision of our technique to identify similar situations and particularly to set out the threshold similarity value, we propose to use a manual classification as a baseline and compare it with the results obtained by our technique. So, we manually group similar situations, and we compare the manual constructed groups with the results obtained by our similarity algorithm, with different threshold values.

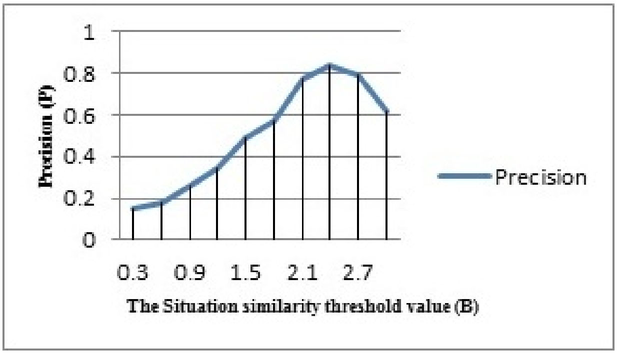

Fig. 1. Effect of B threshold value on the similarity accuracy

Figure 1 shows the effect of varying the threshold situation similarity parameter B in the interval [0, 3] on the overall precision P. Results show that the best performance is obtained when B has the value 2.4 achieving a precision of 0.849. Consequently, we use the identified optimal threshold value (B = 2.4) of the situation similarity measure for testing effectiveness of our MCRS presented below.

4.3 Experimental Datasets

In this Section, we evaluate the following algorithms: ε-greedy and hybrid-ε-greedy, described in Section 3.3; CBR-ε-greedy, a version of the hybrid-ε-greedy algorithm without executing the CBF.

We evaluated these algorithms over a set of similar user situations using the optimal threshold value identified above (B = 2.4).

The testing step consists of evaluating the algorithms for each testing situation using the traditional precision measure. As usually done for evaluating systems based on machine learning techniques, we randomly divided the entries set into two subsets. The first one, called "learning subset", consists of a small fraction of interaction on which the bandit algorithm is run to learn/estimate the CTR associated to each document. The other one, called "deployment subset", is the one used by the system to greedily recommend documents using CTR estimates obtained from the learning subset.

4.4 Results for ε Variation

Each of the competing algorithms requires a single parameter ε. Figures 2 and 3 show how the precision varies for each algorithm with the respective parameters. All the results are obtained by a single run.

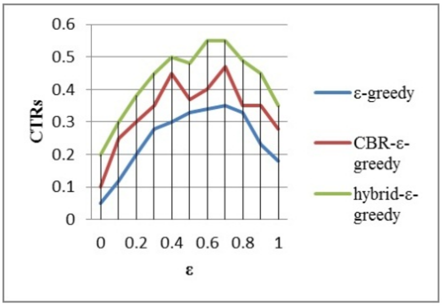

Fig. 2. ε Variation on learning subset

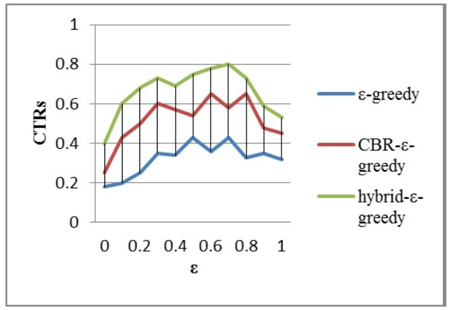

Fig. 3. ε variation on deployment subset

As seen from these figures, when the parameter ε is too small, there is insufficient exploration; consequently the algorithms failed to identify relevant documents, and had a smaller number of clicks. Moreover, when the parameter is too large, the algorithms seemed to over-explore and thus wasted some of the opportunities to increase the number of clicks. Based on these results, we choose appropriate parameters for each algorithm and run them once on the evaluation data.

We can conclude from the plots that CBR information is indeed helpful for finding a better match between user interest and document content. The CBF also helps hybrid-ε-greedy in the learning subset by selecting more attractive documents to recommend.

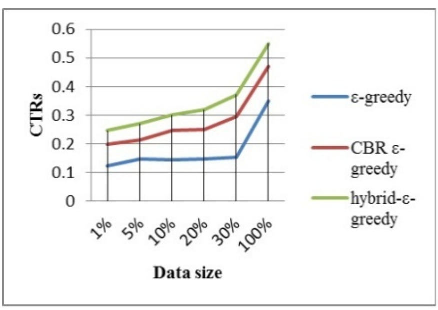

Fig. 4. Learning data size

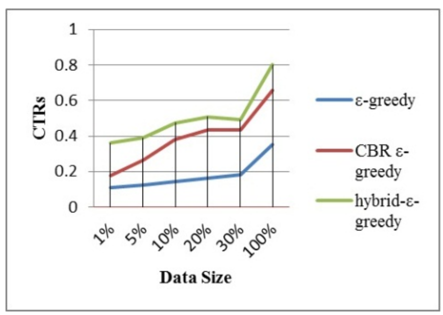

Fig. 5. Deployment data size

4.5 Valuate Sparse Data

To compare the algorithms when data is sparse in our experiments, we reduced data sizes of 30%, 20%, 10%, 5%, and 1%, respectively.

To better visualize the comparison results, figures 4 and 5 show algorithms' precision graphs with the previous referred data sparseness levels. Our first conclusion is that, at all data sparseness levels, the three algorithms are useful. A second interesting conclusion is that hybrid-ε-greedy's methods outperform the ε-greedy's one in learning and deployment subsets. The advantage of hybrid-ε-greedy over ε-greedy is even more apparent when data size is smaller. At the level of 1% for instance, we observe an improvement of 0.189 in hybrid-ε-greedy's precision using the deployment subset (0.363) over the ε-greedy's one (0.174).

5 Conclusion

This paper describes our approach for implementing a MCRS. Our contribution is to make a deal between exploration and exploitation for learning and maintaining user's interests based on his/her navigation history.

We have presented an evaluation protocol based on real mobile navigation. We evaluated our approach according to the proposed evaluation protocol. This study yields to the conclusion that considering the situation in the exploration/exploitation strategy significantly increases the performance of the recommender system following the user interests.

In the future, we plan to compute the weights of each context dimension and consider them on the detection of user's situation, and then we plan to extend our situation with more context dimension. Regarding the bandit algorithms we plan to investigate methods that automatically learn the optimal exploitation and exploration tradeoff.

References

1. Bellotti, V., Begole, B., Chi, E.H., Ducheneaut, N., Fang, J., Isaacs, E., King, T., Newman, M.W., Walendowski, A.: Scalable architecture for context-aware activity-detecting mobile recommendation system. In: 9th IEEE International Symposium on a World of Wireless, WOWMOM 2008 (2008)
2. Lakshmish, R., Deepak, P., Ramana, P., Kutila, G., Dinesh, G., Karthik, V., Shivkumar, K.: A Mobile context-aware, Social Recommender System for Low-End Mobile Devices. In: Tenth International Conference on Mobile Data Management: CAESAR 2009, pp. 338–347 (2009)
3. Lihong, L., Wei, C., Langford, J., Schapire, R.E.: A Contextual-Bandit Approach to Personalized News Article Recommendation. Presented at the Nineteenth International Conference on World Wide Web, CoRR 2010, Raleigh, vol. abs/1002.4058 (2010)
4. Bouidghaghen, O., Tamine-Lechani, L., Boughanem, M.: Dynamically Personalizing Search Results for Mobile Users. In: Andreasen, T., Yager, R.R., Bulskov, H., Christiansen, H., Larsen, H.L. (eds.) FQAS 2009. LNCS, vol. 5822, pp. 99–110. Springer, Heidelberg (2009)
5. Panayiotou, C., Maria, I., Samaras, G.: Using time and activity in personalization for the mobile user. In: On Fifth ACM International Workshop on Data Engineering for Wireless and Mobile Access, MobiDE 2006, pp. 87–90 (2006)
6. Peter, S., Linton, F., Joy, D.: OWL: A Recommender System for Organization-Wide Learning. In: Educational Technology, ET 2000, vol. 3, pp. 313–334 (2000)
7. Bianchi, R.A.C., Ros, R., Lopez de Mantaras, R.: Improving Reinforcement Learning by Using Case Based Heuristics. In: McGinty, L., Wilson, D.C. (eds.) ICCBR 2009. LNCS, vol. 5650, pp. 75–89. Springer, Heidelberg (2009)

8. Samaras, G., Panayiotou, C.: Personalized Portals for the Wireless and Mobile User. In: Proc. 2nd Int'l Workshop on Mobile Commerce, ICDE 2003, p. 792 (2003)
9. Shijun, L., Zhang, Y., Xie, Sun, H.: Belief Reasoning Recommendation Mashing up Web Information Fusion and FOAF:JC 2010. Journal of Computers, JC 02(12), 1885–1892 (2010)
10. Varma, V., Kushal, S.: Pattern based keyword extraction for contextual advertising. In: Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM (2010)
11. Wei, L., Wang, X., Zhang, R., Cui, Y., Mao, J., Jin, R.: Exploitation and Exploration in a Performance based Contextual Advertising System. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2010. ACM (2010)
12. Wu, Z., Palmer, M.: Verb Semantics and Lexical Selection. In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, ACL 1994, pp. 133-138 (1994)
13. Zhang, T., Iyengar, V.: Recommender systems using linear classifiers. The Journal of Machine Learning Research, JMLR 2, 313–334 (2002)