Considering the High Level Critical Situations in Context-Aware Recommender Systems

Djallel Bouneffouf
Department of Computer Science, Télécom SudParis, UMR CNRS Samovar
91011 Evry Cedex, France
Djallel.Bouneffouf@it-sudparis.eu

Amel Bouzeghoub
Department of Computer Science, Télécom SudParis, UMR CNRS Samovar
91011 Evry Cedex, France
Amel.Bouzeghoub@it-sudparis.eu

Alda Lopes Gançarski
Department of Computer Science, Télécom SudParis, UMR CNRS Samovar
91011 Evry Cedex, France
Alda.Gancarski@it-sudparis.eu

ABSTRACT

Most existing approaches in Context-Aware Recommender Systems (CRS) focus on recommending relevant items to users taking into account contextual information, such as time, location, or social aspects. However, none of them have considered the problem of user's content dynamicity. This problem has been studied in the reinforcement learning community, but without paying much attention to the contextual aspect of the recommendation. We introduce in this paper an algorithm that tackles the user's content dynamicity by modeling the CRS as a contextual bandit algorithm. It is based on dynamic exploration/exploitation and it includes a metric to decide which user's situation is the most relevant to exploration or exploitation. Within a deliberately designed offline simulation framework, we conduct extensive evaluations with real online event log data. The experimental results and detailed analysis demonstrate that our algorithm outperforms surveyed algorithms.

1. INTRODUCTION

Mobile technologies have made access to a huge collection of information, anywhere and anytime. In particular, most professional mobile users acquire and maintain a large amount of content in their repository. Moreover, the content of such repository changes dynamically, undergoes frequent updates. In this sense, recommender systems must promptly identify the importance of new documents, while adapting to the fading value of old documents. In such a setting, it is crucial to identify and recommend interesting content for users.

A considerable amount of research has been done in recommending interesting content for mobile users. Earlier techniques in Context-Aware Recommender Systems (CRS) [3, 6, 12, 5, 22, 23] are based solely on the computational behavior of the user to model his interests regarding his surrounding environment like location, time and near people (the user's situation). The main limitation of such approaches is that they do not take into account the dynamicity of the user's content.

This gives rise to another category of recommendation techniques that try to tackle this limitation by using collaborative, content-based or hybrid filtering techniques. Collaborative filtering, by finding similarities through the users' history, gives an interesting recommendation only if the overlap between users' history is high and the user's content is static[18]. Content-based filtering, identify new documents which match with an existing user's profile, however, the recommended documents are always similar to the documents previously selected by the user [15]. Hybrid approaches have been developed by combining the two latest techniques; so that, the inability of collaborative filtering to recommend new documents is reduced by combining it with content-based filtering [13].

However, the user's content in mobile undergoes frequent changes. These issues make content-based and collaborative filtering approaches difficult to apply [8].

Few works found in the literature [13, 21] solve this problem by addressing it as a need for balancing exploration and exploitation studied in the "bandit algorithm" [20].

A bandit algorithm B exploits its past experience to select documents (arms) that appear more frequently. Besides, these seemingly optimal documents may in fact be suboptimal, because of the imprecision in B's knowledge. In order to avoid this undesired situation, B has to explore documents by 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 increases long-term user's satisfaction.

Clearly, neither a purely exploring nor a purely exploiting algorithm works well, and a good tradeoff is needed.

The authors on [13, 21] 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.

In order to give CRS the capability to provide the mobile user's information matching his/her situation and adapted to the evolution of his/her content (good exr/exp tradeoff in the bandit algorithm), we propose an algorithm witch takes into account the user's situation for defining the (exr/exp) tradeoff, and then selects suitable situations for either exploration or exploitation.

The rest of the paper is organized as follows. Section 2 reviews some related works. Section 3 presents the user's model of our CRS. Section 4 describes the algorithms involved in the proposed approach. The experimental evaluation is illustrated in Section 5. The last section concludes the paper and points out possible directions for future work.

2. RELATED WORKS

We review in the following recent relevant recommendation techniques that tackle the two issues mentioned above, namely: following the evolution of the user's contents using bandit algorithm and considering the user's situation on recommender system.

2.1 Bandit Algorithms Overview

The (exr/exp) tradeoff was firstly studied in reinforcement learning in 1980's, and later flourished in other fields of machine learning [16, 19]. Very frequently used in reinforcement learning to study the (exr/exp) tradeoff, the multi-armed bandit problem was originally described by Robbins [20].

The ε-greedy is the one of the most used strategy to solve the bandit problem and was first described in [14]. The ε-greedy strategy choose a random document with epsilon-frequency (ε), and choose otherwise the document with the highest estimated mean, the estimation is based on the rewards observed thus far. ε must be in the open interval [0, 1] and its choice is left to the user.

The first variant of the ε-greedy strategy is what [9, 14] refer to as the ε-beginning strategy. This strategy makes exploration all at once at the beginning. For a given number $I \in N$ of iterations, the documents are randomly pulled during the εI first iterations. During the remaining (1−ε)I iterations, the document of highest estimated mean is pulled.

Another variant of the ε-greedy strategy is what Cesa-Bianchi and Fisher [14] call the ε-decreasing strategy. In this strategy, the document with the highest estimated mean is always pulled except when a random document is pulled instead with an ε_i frequency, where n is the index of the current round. The value of the decreasing ε_i is given by $\varepsilon_i = \{\varepsilon_0 / i\}$ where $\varepsilon_0 \in]0,1]$. Besides ε-decreasing, four other strategies are presented in [4]. Those strategies are not described here because the experiments done by [4] seem to show that, with carefully chosen parameters, ε-decreasing is always as good as the other strategies.

Compared to the standard multi-armed bandit problem with a fixed set of possible actions, in CRS, old documents may expire and new documents may frequently emerge. Therefore it may not be desirable to perform the exploration all at once at the beginning as in [9] or to decrease monotonically the effort on exploration as the decreasing strategy in [14].

Few research works are dedicated to study the contextual bandit problem on Recommender System, where they consider user's behavior as the context of the bandit problem.

In [10], authors extend the ε-greedy strategy by updating the exploration value ε dynamically. At each iteration, they run a sampling procedure to select a new ε from a finite set of candidates. The probabilities associated to the candidates are uniformly initialized and updated with the Exponentiated Gradient (EG) [10]. This updating rule increases the probability of a candidate ε if it leads to a user's click. Compared to both ε-beginning and decreasing strategy, this technique improves the results.

In [13], authors model the recommendation as a contextual bandit problem. They propose an approach in which a learning algorithm selects sequentially documents to serve users based on contextual information about the users and the documents. To maximize the total number of user's clicks, this work proposes the LINUCB algorithm that is computationally efficient.

The authors in [4, 9, 13, 14, 21] describe a smart way to balance exploration and exploitation. 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 [7, 17] the authors propose a method which consists of building a dynamic user's profile based on time and user's experience. The user's preferences in the user's profile are weighted according to the situation (time, location) and the user's behavior. To model the evolution on the user's preferences according to his temporal situation in different periods, (like workday or vacations), the weighted association for the concepts in the user's profile is established for every new experience of the user. The user's activity combined with the user's profile are used together to filter and recommend relevant content.

Another work [12] describes a CRS operating on three dimensions of context that complement each other to get highly targeted. First, the CRS analyzes information such as clients' address books to estimate the level of social affinity among the users. Second, it combines social affinity with the spatiotemporal dimensions and the user's history in order to improve the quality of the recommendations.

In [3], the authors present a technique to perform user-based collaborative filtering. Each user's mobile device stores all explicit ratings made by its owner as well as ratings received from other users. Only users in spatiotemporal proximity are able to exchange ratings and they show how this provides a natural filtering based on social contexts.

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 content.

As shown in above, none of the mentioned works tackles both problems of the evolution user's content and user's situation consideration in the recommendation. This is precisely what we intend to do with our approach, by modeling the CRS as a contextual bandit algorithm, and considering the user's situation when managing the (exr/exp)-tradeoff on recommendation.

The two features cited above are not considered in the surveyed approaches as far as we know.

In what follows, we define briefly the structure of the user's model and the methods for inferring the recommendation situations. Then, we explain how to manage the exploration/exploitation strategy, according to the current situation.

3. USER AND CONTEXT MODELS

The user's model is structured as a case base, which is composed of a set of past situations with their corresponding user's preferences, denoted $PS = \{(S^i; UP^i)\}$, where S^i is a user's situation (Section 3.2.1) and UP^i its corresponding user's preferences (Section 3.1).

3.1 The User's Preferences

The user's preferences are contextual and might depend on many factors, like the location or the current task within an activity. Thus, they are associated to the user's situation and the user's activity. Preferences are deduced during the user's navigation activities. A navigation activity expresses the following sequence of events:

(i) the user's logs in the system and navigates across documents to get the desired information;

(ii) the user expresses his/her preferences about 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 two types of preference:

- The direct preference: the user expresses his/her interest in the document by inserting a rate, like for example putting starts ("*") at the top of the document.

- The indirect preference: it is the information that we extract from the user's system interaction, for example the number of clicks on the visited documents or the time spent on a document.

Let *UP* be the preferences submitted by a specific user in 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 document's identifier, 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.

3.2 Context Model

A user's context C is a multi-ontology representation where each ontology corresponds to a context dimension $C=(\boldsymbol{O}_{Location}, \boldsymbol{O}_{Time}, \boldsymbol{O}_{Social})$. Each dimension models and manages a context information type. We focus on these three dimensions since they cover all needed information. These ontologies are described in [1] and are not developed in this paper.

3.2.1 Situation Model

A situation is a projection on one or several user's context dimensions. In other words, we consider a situation as a triple $s = (\boldsymbol{O}_{Location}.x_i, \boldsymbol{O}_{Time}.x_j, \boldsymbol{O}_{Social}.x_k)$ where x_i, x_j and x_k are ontology concepts or instances. Suppose the following data are sensed from the user's mobile phone: the GPS shows the latitude and longitude of a point "48.8925349, 2.2367939"; the local time is "Mon *May* 3 12:10:00 2012" and the calendar states "meeting with Paul Gerard". The corresponding situation is:

$S=(\boldsymbol{O}_{Location}$,*"48.89,2.23"*,

$\boldsymbol{O}_{Time}$.*"Mon_May_3_12:10:00_2012"*, $\boldsymbol{O}_{Social}$. *"Paul_Gerard")*.

To build a more abstracted situation, we interpret the user's behavior from this low-level multimodal sensor data using ontologies reasoning means. For example, from *S*, we obtain the following situation:

MeetingAtRestaurant=

($\boldsymbol{O}_{Location}$.*Restaurant*, $\boldsymbol{O}_{Time}$.*Work_day*, $\boldsymbol{O}_{Social}$.*Financial_client)*.

For simplification reasons, we adopt in the rest of the paper the following notation:

$S = (x_i, x_j, x_k)$. The previous example situation became thus:

MeetingAtRestarant=(Restaurant, Work_day, Financial_client).

Among the set of captured situations, some of them are characterized as *high-level critical situations*.

3.2.2 High Level Critical Situations (HLCS)

A HLCS is a class of situations where the user needs the best information that can be recommended by the system, for instance, when the user is in a professional meeting. In such a situation, the system must exclusively perform exploitation rather than exploration-oriented learning. In the other case, for instance where the user is using his/her information system at home, on vacation with friends S = (*home, vacation, friends*). The system can make some exploration by recommending the user some information ignoring their interest. The HLCS situations are for the moment predefined by the domain expert. In our case we conduct the study with professional mobile users, which is described in detail in (section 5). As examples of HLCS, we can find *S1* = (*company, Monday morning, colleague*), *S2* = (*restaurant, midday, client*) or *S3*= (*company, morning, manager*).

4. THE PROPOSED RECOMMENDATION ALGORITHM

The problem of recommending documents can be naturally modeled as a multi-armed bandit problem with context information. In our case we consider the user's situation as the context information of the multi-armed bandit. Following previous work [11], we call it a contextual bandit. Formally, our contextual-bandit algorithm proceeds in trials $t = 1 \ldots T$. For each trial t, the algorithm performs the following tasks:

Task 1: Let S^t be the current user's situation, and *PS* be the case base containing the set of past situations and corresponding user's preferences. The system compares S^t with the situations in *PS* in order to choose the most similar S^p using the *RetrieveCase()* method (Section 4.2.1).

Task 2: Let D be the document collection and $D_p \in D$ the set of documents that were recommended in situation S^p. When the user read each document $d_i \in D_p$, the system observed his behavior and interpreted it as a reward. Based on the observed documents' rewards, the algorithm chooses the document $d_p \in D_p$ with the greater reward r_p; this is done using the *RecommendDocuments*() method (Section 4.2.2).

Task 3: The algorithm improves its document-selection strategy with the new current observation (d_p, r_t). The updating of the case base is done using the *Auto_improvement()* method (Section 4.2.3).

In tasks 1 to 3, the total T-trial reward for each document d_i in D is defined as $\sum_{t=1}^{T} r_{t,d_i}$ while the optimal expected T-trial reward is defined as $\mathrm{E}\left[\sum_{t\in T_i} r_{t,d_i^*}\right]$ where d_i^* is the document with maximum expected total reward, where T_i is the set of trials from T where d_i^* was recommended to the user. Our goal is to design the bandit algorithm so that the expected total reward is maximized.

In the field of document recommendation, 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 dividing the total number of clicks on it by the number of times it was recommended. It is important to know here that no reward is observed for non-recommended documents.

4.1 The ε-greedy() Algorithm

The *ε-greedy* algorithm recommends a predefined number of documents N, each one computed using the following equation:

$$d_i = \begin{cases} \arg\max_{UC}\left(getCTR(d)\right) & \text{if } q > \varepsilon \\ Random(UC) & otherwise \end{cases} \tag{1}$$

In Eq. 1, $i\in\{1,\ldots N\}$, $UC=\{d_1,\ldots,d_P\}$ is the set of documents corresponding to the user's preferences; *getCTR* is the function which estimates the CTR of a given document; *Random* is the function returning a random element from a given set, allowing to perform exploration; q is a random value uniformly distributed over [0, 1] which defines the exploration/exploitation tradeoff; ε is the probability of recommending a random exploratory document.

4.2 Contextual-ε-greedy()

To adapt the ε-greedy algorithm to a context aware environment, we propose to compute the similarity between the current situation and each one in the situation base; if there is a situation that can be reused, the algorithm retrieves it, and then applies the ε-greedy algorithm to the corresponding user preferences. Alg. 1 describes the proposed *Contextual-ε-greedy*() algorithm which involves the following three methods.

Algorithm 1 Context-ε-greedy()
Input: ε, N, PS, S^t, B
Output: D^t
// Retrieve the most similar case
(S^P, UP^p) = ***RetrieveCase***(S^t, PS);
// Recommend documents
D^t=***RecommendDocuments***$(\varepsilon, UP^p, S^t, S^p, N, B)$;
Receive a feedback UP^t from the user;
// update user's profile
Auto_improvement(PS, UP^t, S^t, S^p);
Endfor

4.2.1 *RetrieveCase()*

Given the current situation S^t, the *RetrieveCase()* method determines the expected user's preferences by comparing S^t with the situations in past cases PS in order to choose the most similar one S^p. The method returns, then, the corresponding case (S^p, UP^p). S^p is selected from PS by computing the following expression:

$$S^p = \arg\max_{S^i\in PS}\left(\sum_j \alpha_j \cdot sim_j\left(X_j^t, X_j^i\right)\right) \tag{2}$$

In Eq.2, sim_j is the similarity metric related to dimension j between two concepts X^t and X^i. This similarity depends on how closely X^t and X^i are related in the corresponding ontology (location, time or social). α_j is the weight associated to dimension j, and it is set out by using an arithmetic mean as follows:

$$\alpha_j = \frac{1}{T}\sum_{k=1}^{T}\gamma_j^k \tag{3}$$

In Eq. 3, $\gamma_j^i = sim_j\left(x_j^t, x_j^p\right)$ at trial $k\in\{1,\ldots,T\}$ from the T previous recommendations, where $x_j^p \in S^p$. The idea here is to augment the importance of a dimension with the corresponding previously computed similarity values, reflecting the impact of the dimension when computing the most similar situation in Eq. 2.

The similarity between two concepts of a dimension j in an ontological semantics depends on how closely they are related in the corresponding ontology (location, time or social). We use the same similarity measure as [24] defined by Eq. 4:

$$sim_j\left(x_j^t, x_j^c\right) = 2 * \frac{deph(LCS)}{(deph(x_j^t) + deph(x_j^c))} \tag{4}$$

In Eq. 4, LCS is the Least Common Subsumer of x_j^t and x_j^c, and deph is the number of nodes in the path from the node to the ontology root.

4.2.2 *RecommendDocuments()*

In order to insure a better precision of the recommender results, the recommendation takes place only if the following condition is verified: $sim(S^t, S^p)\geq B$, where B the similarity threshold value and $sim(S^t, S^p) = \sum_j \alpha_j sim_j\left(x_j^t, x_j^p\right)$.

To improve the adaptation of the ε-greedy algorithm to HLCS situations, if $S^p \in HLCS$, we propose the system to not make exploration when choosing the document to recommend, as indicated in the following equation:

$$d_i = \begin{cases} \arg\max_{UC}(\, getCTR(d)) & if\ S^p \in HLCS \\ \varepsilon\text{-}greedy(\,) & otherwise \end{cases} \quad (5)$$

In Eq. 5, if S^p is not *HLCS*, the system recommends documents using *ε-greedy* with an *ε* computed at an initialization step by testing different *ε* and selects the optimal one, this step is described below (Section 5.4).

4.2.3 *Auto_improvement ()*

This method is used to update the user's preferences w. r. t. the number of clicks and number of recommendations for each recommended document on which the user clicked at least one time. Depending on the similarity between the current situation S^t and its most similar situation S^p (computed with *RetrieveCase()*, Section 4.2.1), being 3 the number of dimensions in the context, two scenarios are possible:

- $sim(S^t, S^p) \neq 3$: the current situation does not exist in the case base; the system adds to the case base the new case composed of the current situation S^t and the current user preferences UP^t.

- $sim(S^t, S^p) = 3$: the situation exists in the case base; the system updates the case having premise situation S^p with the current user preferences UP^t.

5. EXPERIMENTAL EVALUATION

In order to evaluate empirically the performance of our approach, and in the absence of a standard evaluation framework, we propose an evaluation framework based on a diary set of study entries. The main objectives of the experimental evaluation are:

(1) to find the optimal parameters of our algorithm.

(2) to evaluate the performance of the proposed algorithm w. r. t. the ε variation. In the following, we describe our experimental datasets and then present and discuss the obtained results.

5.1 Evaluation Framework

We have conducted a diary study with the collaboration of the French software company Nomalys[1]. This company provides a history application, which records time, current location, social and navigation information of its users during their application use. The diary study has taken 18 months and has generated 178369 diary situation entries.

Each diary situation entry represents the capture of contextual time, location and social information. For each entry, the captured data are replaced with more abstracted information using time, spatial and social ontologies. Table 1 illustrates three examples of such transformations.

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

Table 1: Semantic diary situation

From the diary study, we have obtained a total of 2759283 entries concerning the user's navigation, expressed with an average of 15.47 entries per situation. Table 2 illustrates examples of such diary navigation entries, where **Click** is the number of clicks on a document; **Time** is the time spent on reading a document, and **Interest** is the direct interest expressed by stars (the maximum number of stars is five).

IdDoc	IDS	Click	Time	Interest
1	1	2	2'	***
2	1	4	3'	*
3	2	1	5'	*

Table 2: Diary navigation entries

5.2 Finding the Optimal Parameters

In our experiments, we have firstly collected the 3000 situations (*HS*) with an occurrence greater than 100 to be statistically meaningful, and the 10000 documents (*HD*) that have been shown on any of these situations.

The testing step consists of evaluating the existing algorithms for a situation randomly selected from the sampling *HS*, taking into account the number of times that the situation was selected and the number occurrences of the situation^t in *HS*. The evaluation algorithm computes and displays the average CTR every 1000 iterations.

The average CTR for a particular iteration is the ratio between the total number of clicks and the total number of displays. The number of documents returned by the recommender system for each situation is 10 and we have run the simulation until the number of iterations reaches 10000.

5.2.1 *The threshold similarity value*

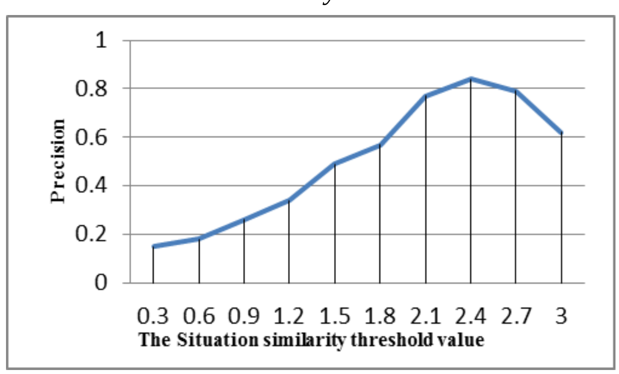

Figure 1. Effect of B threshold value on the similarity precision

[1] Nomalys is a company that provides a graphical application on Smartphones allowing users to access their company's data.

Figure 1 shows the effect of varying the threshold situation similarity parameter B (Section 2.2) in the interval [0, 3] on the overall precision. The results show that the best performance is obtained when B has the value 2.4 achieving a precision of 0.849.

So, we use the identified optimal threshold value (B = 2.4) of the situation similarity measure for testing our CRS.

5.3 Experimental Results

In our experiments, we have firstly collected the 3000 situations (HS) with an occurrence greater than 100 to be statistically meaningful, and the 10000 documents (HD) that have been shown on any of these situations.

The testing step consists of evaluating the existing algorithms for a situation randomly selected from the sampling HS, taking into account the number of times that the situation was selected and the number occurrences of the situation[t] in HS. The evaluation algorithm computes and displays the average CTR every 1000 iterations.

The average CTR for a particular iteration is the ratio between the total number of clicks and the total number of displays. The number of documents returned by the recommender system for each situation is 10 and we have run the simulation until the number of iterations reaches 10000.

5.4 Results for ε Variation

In order to evaluate only the impact of considering the user's situation in our bandit algorithm, we have replaced in *RecommendDocuments()*, the equation 5 by the equation 1, we call the new algorithm *Contextual-ε-greedy without HLCS*. Then we have compared this algorithm to the *ε-greedy* (Section 4.1).

Each of the competing algorithms requires a single parameter ε. Figure 2 shows how the average CTR varies for each algorithm with the respective ε.

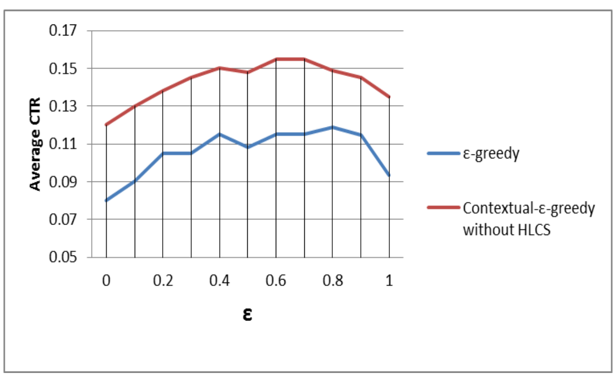

Figure 2. Variation ε tradeoff

Figure 2 shows that, when the ε is too small, there is an insufficient exploration; consequently the algorithms have failed to identify interesting documents, and have got a smaller number of clicks (average CTR).

Moreover, when the parameter is too large, the algorithms seem to over-explore and thus lose a lot of opportunities to increase the number of clicks.

We can conclude from the evaluation that considering the user's situation is indeed helpful for *Context-ε-greedy* to find a better match between the user's interest and the evolution of his content (documents).

5.5 Evaluation The Impact of The HLCS

In order to evaluate the impact of the HLCS situations in the recommender system, we have compared *Contextual-ε-greedy without HLCS* and the original version of *Contextual-ε-greedy*. Figure 3 shows how the average CTR varies for each algorithm with the respective ε.

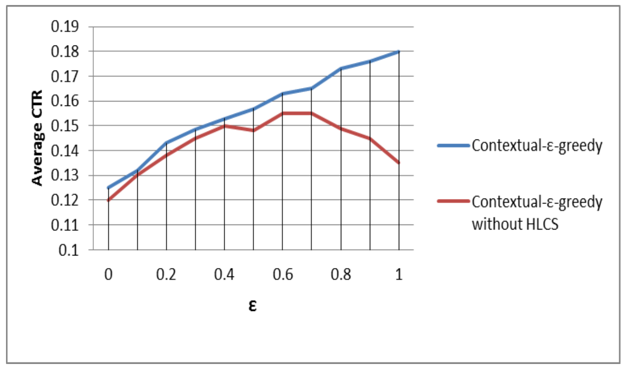

Figure 3. Variation ε tradeoff

As seen in the Figure 3, on one hand, when the ε is too small, there is an insufficient exploration; consequently the impact of the HLCS is low; on the other hand, when the parameter is too large, the *Contextual-ε-greedy* takes full advantage of exploration without wasting opportunities to establish good CTR (the impact of the HLCS is more important).

We can conclude from the evaluation that considering HLCS situations in recommender system allows a better precision on recommendation.

6. CONCLUSION

In this paper, we have studied the problem of exploitation and exploration in context-aware recommender systems and propose a new approach that balances adaptively exr/exp regarding the user's situation.

We have presented an evaluation protocol based on real mobile navigation contexts obtained from a diary study conducted with collaboration with the Nomalys French company. We have evaluated our approach according to the proposed evaluation protocol and show that it is effective.

In order to evaluate the performance of the proposed algorithm, we compare it with standard exr/exp strategy. The experimental results demonstrate that our algorithm performs better on average CTR in various configurations. Moreover, this study yields to the conclusion that considering the situation on the exploration/exploitation strategy significantly increases the performance of the system on following the user's contents evolution.

In the future, we plan to extend our situation with more context dimension, and we plan to evaluate our approach using an online framework.

References

[1] D. Bouneffouf, A. Bouzeghoub & A. L. Gançarski, Following the User's Interests in Mobile Context-Aware recommender systems. AINA Workshops, 657-662, 2012.

[2] G. Adomavicius, B. Mobasher, F. Ricci, Alexander Tuzhilin. Context-Aware Recommender Systems. AI Magazine. 32(3): 67-80, 2011.

[3] S. Alexandre, C. Moira Norrie, M. Grossniklaus and B.Signer, "Spatio-Temporal Proximity as a Basis for Collaborative Filtering in Mobile Environments". Workshop on Ubiquitous Mobile Information and Collaboration Systems, CAiSE, 2006.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite Time Analysis of the Multiarmed Bandit Problem. Machine Learning, 2, 235–256, 2002.

[5] R. Bader , E. Neufeld , W. Woerndl , V. Prinz, Context-aware POI recommendations in an automotive scenario using multi-criteria decision making methods, Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation, p.23-30, 2011.

[6] L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci. Context relevance assessment and exploitation in mobile recommender systems. Personal and Ubiquitous Computing, 2011.

[7] V. Bellotti, B. Begole, E.H. Chi, N. Ducheneaut, J. Fang, E. Isaacs, Activity-Based Serendipitous Recommendations with the Magitti Mobile Leisure Guide. Proceedings On the Move, 2008.

[8] W. Chu and S. Park. Personalized recommendation on dynamic content using predictive bilinear models. In Proc. of the 18th International Conf. on World Wide Web, pages 691–700, 2009.

[9] E. Even-Dar, S. Mannor, and Y. Mansour. PAC Bounds for Multi-Armed Bandit and Markov Decision Processes. In Fifteenth Annual Conference on Computational Learning Theory, 255–270, 2002.

[10] J. Kivinen and K. Manfred, Warmuth. Exponentiated gradient versus gradient descent for linear predictors. Information and Computation, 132-163, 1997.

[11] J. Langford and T. Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In Advances in Neural Information Processing Systems 20, 2008.

[12] R. Lakshmish, P. Deepak, P. Ramana, G. Kutila, D. Garg, V. Karthik, K. Shivkumar , Tenth International Conference on Mobile Data Management: Systems, Services and Middleware CAESAR: A Mobile context-aware, Social Recommender System for Low-End Mobile Devices, 2009.

[13] Li. Lihong, C. Wei, J. Langford, E. Schapire. A Contextual-Bandit Approach to Personalized News Document Recommendation. CoRR, Presented at the Nineteenth International Conference on World Wide Web, Raleigh, Vol. abs/1002.4058, 2010.

[14] S. Mannor and J. N. Tsitsiklis. The Sample Complexity of Exploration in the Multi-Armed Bandit Problem. In Sixteenth Annual Conference on Computational Learning Theory, 2003.

[15] D. Mladenic. Text-learning and related intelligent agents: A survey. IEEE Intelligent Agents, pages 44–54, 1999.

[16] H. Robbins. Some aspects of the sequential design of experiments. Bulletin of the American Mathematical Society, 1952.

[17] G. Samaras, C. Panayiotou. Personalized portals for the wireless user based on mobile agents. Proc. 2nd Int'l workshop on Mobile Commerce, pp. 70-74, 2002.

[18] J. B. Schafer, J. Konstan, and J. Riedi. Recommender systems in e-commerce. In Proc. of the 1st ACM Conf. on Electronic Commerce, 1999.

[19] R. Sutton and A. Barto. Reinforcement Learning: An Introduction. MIT Press, 1998.

[20] C. J. C. H. Watkins, Learning from Delayed Rewards. Ph.D. thesis. Cambridge University, 1989.

[21] Li. Wei, X. Wang, R. Zhang, Y. Cui, J. Mao, R. Jin. Exploitation and Exploration in a Performance based Contextual Advertising System: KDD'10, pp. 133-138. Proceedings of the 16th SIGKDD International Conference on Knowledge discovery and data mining, 2010.

[22] W. Woerndl, Florian Schulze: Capturing, Analyzing and Utilizing Context-Based Information About User Activities on Smartphones. Activity Context Representation, 2011.

[23] W. Woerndl, J. Huebner, R. Bader, D. Gallego-Vico: A model for proactivity in mobile, context-aware recommender systems. RecSys , 273-276, 2011

[24] Z. Wu and M. Palmer. Verb Semantics and Lexical Selection. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, 133-138, 1994.