HAL
open science

Ant Clustering to Improve Situation-Aware Recommender Systems

Djallel Bouneffouf

To cite this version:

Djallel Bouneffouf. Ant Clustering to Improve Situation-Aware Recommender Systems. 2014. hal-01056896

HAL Id: hal-01056896
https://hal.science/hal-01056896v1

Preprint submitted on 20 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Ant Clustering to Improve Situation-Aware Recommender Systems

Djallel Bouneffouf
Télécom SudParis
9, rue Charles Fourier
91011 Evry, France
+33 1 60 76 47 82
Djallel.Bouneffouf@it-sudparis.eu

Abstract
We improve in this paper an algorithm that tackles the user's content dynamicity by modeling the CRS as a contextual bandit algorithm. We have including a situation clustering algorithm to improve the precision of the CRS. The algorithm combines k-mean and the ant clustering approaches. The experimental results reveal the performance of our proposed algorithm.

Introdution
H.3.3 [**Information Search and Retrieval**]: *information filtering, Selection process, Relevance feedback.*

General Terms
Algorithms

Keywords
Recommender system; context-aware; machine learning; exploration/exploitation dilemma;k-mean Clustering; Ant Clustering.

1. INTRODUCTION
An amount of research has been done in recommending interesting content for mobile users. Earlier techniques in Context-Aware Recommender Systems (CRS) [6, 13, 5, 12, 24] 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.

Few works found in the literature [1, 2, 3] solve this problem by addressing it as a need for balancing exploration and exploitation studied in the "bandit algorithm" [21].

The authors on [1, 2, 3] describe a smart way to balance exploration and exploitation in the field of CRS. However, only [3] considers the user's situation clustering for recommendation.

In this setting, we propose to improve the clustering algorithm used in [3] by considering ant clustering.

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 Clustering Overview
The clustering is one of the must machine learning approaches used to group information.

Different clustering algorithms exist and two of them are the most used.

The ant based algorithm provides acceptable clusters of data without any knowledge of the initial clusters [14], and the K-means algorithm is more efficient for large datasets but it requires good initialization. These initial values are provided by the ant based algorithm [8] and keep reassigning the data objects in the dataset to cluster centroids based on the similarity between the data object and the cluster centroid.

We use an hybrid ant approach called antkmean-ε-greedy that includes two modules, the ant module and the K-means module. Using this approach, we try to improve the precision of the CRS.

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. Key Notion
The user's model is structured as a case base, which is composed of a set of situations with their corresponding user's preferences, denoted $U = \{(S^i; UP^i)\}$, 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 navi-

gation 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 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*.

3.3 The proposed CRS (antkmean-ε-greedy)

Formally, our contextual-bandit algorithm proceeds in discrete 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* the set of past situations. The system compares S^t with the situations in *PS* in order to choose the most similar S^p using the *Retrieve-Case()* method [1, 2, 3]

Task 2: Let *D* be the document collection and $D_p \in D$ the set of documents recommended in situation S^p. After retrieving S^p, the system observes the user's behavior when reading each document $d_i \in D_p$. Based on observed rewards, the algorithm chooses the document d_p with the greater reward r_p *using* the *RecommendDocuments*() method [1, 2, 3]

Task 3: The algorithm improves its document-selection strategy with the new observation (d_p, r_t). The updating of the case base is done using the *Auto_improvement()* method (Section 4), to accelerate the situation similarity computing this method includes a situation clustering algorithm.

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 rt,d is observed for unchosen documents d ≠ dt previously displayed.

4. SituationClustering

To accelerate the computing similarity between the current situation S* and the situations existing on the case base, we study the existing clustering algorithm. We use an hybrid ant approach that includes two modules, the ant module and the K-means module. The ant based algorithm provides acceptable clusters of data without any knowledge of the initial clusters [18], and the K-means algorithm is more efficient for large datasets but it requires good initialization. These initial values are provided by the ant based algorithm [19] and keeps reassigning the data objects in the dataset to cluster centroids based on the similarity between the data object and the cluster centroid. In the initial stage, the ant module is executed for a short period (50 to 100 iterations) to discover the number of cluster and to avoid consuming high computation. The result from the ant module is used as the initial seed of the K-means module. The K-means algorithm is applied for refining and generating the final result. The whole approach can be summarized as: (1) Start the ant clustering process until the maximum number of iterations is exceeded.(2) Inherit clustering result from ant as the initial centroid vectors of K-means module. Start K-means process until maximum number of iterations is reached.

Algorithm 1 Situation clustering()

1: **Input:** (ε_1, ..., ε_T): candidate values for ε
2: *ck=antclustering(LS) ;*
3: C_k*=K-means(ck) ;*
4: **Output:** (C_1, ..., C_T)

Algorithm 1 Situation clustering()

1: **Input:** Situations = {S_1,…, S_n}, Agents = {a_1,…, aA}, grille G
2: Scatter the situation and ants randomly on the board
3: for (t = 1; t <= t_{max} ; t++)
4: for (ant = 1; ant <= number of ants; ant++)
5: if (ant is not carrying an situation and find situation Si)
6: if (probPick(ant, Si)>= threshold)
7: pickItem(ag, Si);
8: else if (ant transport situation Si and la cellule is empty)
9: if (probDrop(ag, Si) >= threshold)
10: dropItem(ag, xi);
11: the ant moves to an empty cell
12: **Output:** (location of situations {S1,…, Sn} on the board G)

Algorithm 8 Situation clustering()

Input: (C_1, ..., C_k): candidate k-clustering

1: **For** t=1 to t_{max} **do**
//Assign each situation vector to the closest cluster centroids.
//Recalculate the cluster centroid vector $\mathbf{c}_j$ *using*
2: $$c_j = \frac{1}{n} \sum_{\forall d_j \in S_j} d_j$$
3: **Endfor**
4: **Output:** ((ε_1, p_1),..., (ε_T, p_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 is to evaluate the performance of the proposed algorithm w. r. t. the ε variation and the dataset size. In the following, we describe our experimental datasets and then present and discuss the obtained results.

We have conducted a diary study with the collaboration of the French software company Nomalys[1]. This company provides a history application, which records the time, the current location, the 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.

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 by giving to these last as an entry a situation S_j^t selected randomly from the sampling *HS*, where *t* is the identifier of the situation, *j* is the number of time that the situation S^t has selected and j_{max} is the number occurrence of S^t in *HS*. The evaluation algorithm calculate and display the average CTR every 1000 iterations, The average CTR for a particular iteration is computed by the method *getAVCTR* and it is the ratio between the total number of clicks and the total number of displays. The number of documents (*N*) returned by the recommender system for each situation is 10 and we have run the simulation until the number of iterations *(i)* reaches 10000..

5.1 Results for ε Variation

In order to evaluate only the impact of considering the user's situation clustering in our bandit algorithm, we have compared the presented algorithm to a variant were we use only keam algorithm we called the algorithm kmaen-ε-greedy .We omitted also the clustering part of the antkmean-ε-greedy that we called ε-greedy. Each of the competing algorithms requires a single parameter ε. Figure 4 shows how the average CTR varies for each algorithm with the respective ε.

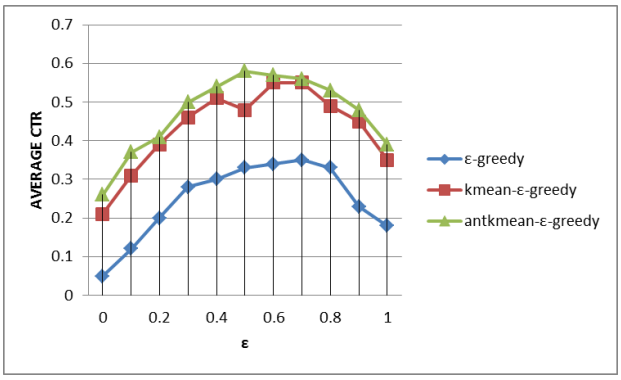

Figure 1. Variation ε tradeoff

Figure 4 shows that, when the parameter 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).

We can conclude from the evaluation that using antkmean-ε-greedy perform the other algorithms.

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

6. CONCLUSION

In this paper, we have proposed an apoche that combine k-mean and ant clusted to improve the situation's clusting in tcontext awer recommender system.

In order to evaluate the performance of the proposed algorithm, we compare it with other standard exr/exp strategies. The experimental results demonstrate the positif impact of the situation clustering in the contextual bandit algorithm. In the future, we plan to extend our situation with more context dimension, and we plan to evaluate our approach using an online framework.

7. References

[1] D. Bouneffouf, A. Bouzeghoub, A. L. Gançarski. A Contextual-Bandit Algorithm for Mobile Context-Aware Recommender System. ICONIP (3): 324-331, 2012

[2] D. Bouneffouf, A. Bouzeghoub, A. L. Gançarski. Hybrid-ε-greedy for Mobile Context-Aware Recommender System. PAKDD (1): 468-479, 2012

[3] D Bouneffouf. The Impact of Situation Clustering in Contextual-Bandit Algorithm for Context-Aware Recommender Systems. arXiv preprint arXiv:1304.3845, 2013

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

[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] G. Hussein. Enhanced K-means-Based Mobile Recommender System. International journal of information studies, vol 2, issue 2, 2010.

[9] 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.

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

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

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

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

[13] 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

[14] Urszula Boryczka, Ant Clustering Algorithm, Intelligent Information Systems, pages 377–386, 2008