ARE RECOMMENDER SYSTEMS REAL-TIME IN MOBILE ENVIRONMENT?
Towards Instantaneous Recommenders

Armelle Brun and Anne Boyer
LORIA-Nancy University, 615, rue du jardin botanique, 54506 Vandœuvre les Nancy, France

Keywords: Recommender systems, m-Commerce, Computation time, Real-time, Usage mining, Association rules.

Abstract: Recommendation technologies have traditionally been used in domains such as e-commerce to recommend resources to customers so as to help them to get the right resources at the right moment. The interest of model-based collaborative filtering, as sequential association rules, in recommender systems has highly increased over the last few years. These models are usually presented as real-time recommenders. In the last few years, the m-commerce domain has emerged, that displays recommendations on the mobile device instead of the classical screen of the computer. In this paper user privacy preservation is an important objective and one way to be compliant with this constraint is to store the recommender on the mobile-side. Though model-based recommenders are real-time, many of them require a significant time to generate recommendations to users and may not be real-time anymore when implemented on a mobile device. Although some works focused on the way to decrease the time required to compute recommendations, the computation complexity still remains relatively high. We put forward a new incremental recommender to get instantaneous recommendations when exploiting usage mining recommender systems in the framework of m-commerce.

1 INTRODUCTION

The democratization of the Internet and network technologies has resulted in an increase in the volume of information easily accessible. However, this profusion of information leads to unsatisfied users as they cannot get easily the information they search. A critical issue of Web applications is thus the incorporation of mechanisms for delivering information that fits user attempts to increase their satisfaction. In e-commerce or m-commerce environments, it is a way to increase both customer fidelity and the selling rate of the (web) stores.

Recommender systems (RS) are such a mechanism, they provide users with personalized recommendations on products based on either the knowledge about users or their past behavior.

To build a RS based on users' past behavior, one can either analyze the content of the previously accessed resources by the active user (the *content-based approach*) or use the information about which resources have been accessed by which users, called *collaborative filtering* (CF).

CF algorithms can be classified into memory-based and model-based algorithms (Adomavicius and Tuzhilin, 2005). Memory-based RS face a scalability problem. Model-based RS cope with this scalability problem due to their off-line computation part. In recent years, the interest of data mining models in the framework of RS (Bozdogan, 2004) has highly increased. In this paper we are interested in data mining models and specifically the well-known association rules.

In model-based approaches the computation time is however still a question: can recommendations be really provided within a small time?

Model-based and particularly data mining-based RS are often presented as real-time RS (Huang et al., 2006). What is a real-time system ? A real-time system "updates information at the same rate as it receives data" [1]. In RS, the data received is the set of user consultations. New data is thus available each time the active user consults a resource. Thus, a real-time RS is able to compute recommendations within two resource consultations of a given user.

RS are usually implemented on the server-side, they may however be implemented on the client-side. In mobile-commerce (m-commerce) for example, the

[1] http://www.thefreedictionary.com

Brun A. and Boyer A.
ARE RECOMMENDER SYSTEMS REAL-TIME IN MOBILE ENVIRONMENT? - Towards Instantaneous Recommenders.
DOI: 10.5220/0002807801010106
In *Proceedings of the 6th International Conference on Web Information Systems and Technology (WEBIST 2010)*, page
ISBN: 978-989-674-025-2

RS can be implemented on the mobile. We consider in thus work that privacy preservation is an important feature to make users use a RS. Implementing RS on the client-side has the advantage of preserving user privacy, we thus choose to implement the RS on the client-side.
M-commerce can be defined as follows “m-commerce applications not only cover the electronic commerce applications, but also include applications that can be performed at any time and from anywhere by using mobile computing technology” (Hu et al., 2006). M-commerce is a recent application domain that has emerged in the last decade, due to improvement of mobile terminal capability, increase of bandwidth, wireless and mobile networks. The research in recommendation algorithms for mobile commerce has thus also been accelerated (Tarasewich, 2003).

As m-commerce and e-commerce have a lot in common, we are interested in studying the exploitation of the well-known RS from e-commerce in the m-commerce domain. Nevertheless, mobiles have several limitations as a low computation capacity, input/ouput facilities and memory. Due to the computation limitation, one can ask if RS are real-time when implemented at client-side.

To answer this question, we specifically focus on the exact way data mining models compute recommendations and put forward a new incremental RS to get instantaneous recommendations.

Section 2 presents RS in m-commerce and describes our scenario of application. Section 3 is an overview of the SAR-based RS. Section 4 is interested in works about the improvement of the computation time in data mining RS. The next section presents the way recommendations are computed in SAR-based RS. Then, we suggest a new incremental RS that reduces the computation time. The last section concludes and presents perspectives.

2 RS IN M-COMMERCE AND SCENARIO

2.1 RS in m-Commerce

Related to the success of RS in e-commerce, personalized RS are now emerging in m-commerce. The application domains are wide and varied, as tourism or restaurant recommendations (Wan, 2009). M-commerce has however some differences and specificities compared to e-commerce, such as mobility, limited processing, transmission capability, size of display, battery power, etc. (Zenebe et al., 2005).

In m-commerce, the RS can be implemented in several ways. In (Wan, 2009), it is implemented on the server side. The mobile registers information about the users and transmits it to the server that computes recommendations. These recommendations are then sent to the mobile that displays them. This solution faces security (connection stability) and privacy problems.
(Tveit, 2001) proposes a P2P RS to cope with the scalability problem of RS. In that case, the number of communications between mobiles and the communication capability is a problem and the author stresses the importance of the computation power of mobiles. This approach is also sensitive to malicious behaviors.
RS can also be implemented on the client side (Lee, 2004). Client-side RS face several drawbacks such as computational capacity and memory size. However, communication bandwidth, security and privacy are no more a problem as there is no communication between the server and the mobile.

2.2 Scenario

This paper focuses on RS in m-commerce in the context of supermarkets. The RS observes the users behavior (the products the users have been interested in) and suggests them relevant products (linked to previous interests, special offers, etc.).

Concretely, the products the users are interested in are captured by the mobile and recommendations are displayed on the screen of the mobile.

In our scenario, we propose to use the RFID technology. The mobile is the RFID reader. A RFID tag is put on items and is detected when it is nearby the reader. We use this technology to identify the items a user puts in his caddy and those he takes in his hand.

2.2.1 Recommend to Users

Customers use a caddy that they fill with the products they want to buy. A mobile device provided by the store is placed on each caddy. Customers thus do not need to use their own mobile (they can even not have any mobile).

In that context, our priorities are user privacy and computation time of recommendations. If the RS is implemented on the server side, the number of communications between the mobiles and the server is huge. Moreover, quality and security of communications between the server and the mobile may be a drawback, as presented in the previous section, leading to a non preservation of the privacy. To avoid these problems, we choose to implement the RS on the mobile-side. The remaining problem is thus the computation time.

Each time a user is interested in a product *i.e.* when he puts it in his caddy or takes it in his hand, this information is captured by the mobile and is exploited by the RS on the mobile to propose to the user the products related to those he has been interested in. The customers do not need to be identified and no information is explicitly asked to them, users' privacy is thus preserved. Moreover, the information automatically collected about the user is only known by the mobile, this information is not transmitted to any other device or server when the user shops.

2.2.2 Train the RS

The mobile device registers all the items the users are interested in. When the customer arrives at the cash desk, the information in the mobile device is transmitted to the server. If the user has no discount card, the information transmitted is anonymous. The RS model is then trained and built on the server side. The order of purchase is important here to recommend items to users that they will see next and not recommend them items they have already seen.

The only information transmitted to the server is the sequence of products the user has actually purchased. Those he has only been interested in are not sent to the server. Thus, the information stored on the server is similar to those stored by classical systems with discount cards.

As the user does not need to be identified, no private information is stored, and users are more enthusiastic to use the system. Moreover, the model is stored only on mobile devices that are owned by the store, it cannot be picked up by other persons or stores and users do not have to store a software on their own mobile. Additionally, the transactions with the server are limited, the system is thus more secured.

3 DATA MINING MODELS IN RS

Many models have been studied in RS (Adomavicius and Tuzhilin, 2005), among which we can find data mining-based RS. Due to the success of these models, we are interested in their ability to be used in m-commerce. We present here data mining RS.

Data mining refers to "discovering, extracting and analyzing knowledge from a large amount of data" (Han and Kamber, 2001). In the framework of e-commerce or m-commerce, the data is made up of the traces of usage (traces of navigation) and the data mining algorithms are used to predict user's preference and future behavior.

Data mining RS proceed in two phases to perform recommendations. In the off-line phase, regular usage patterns of users are extracted. The on-line phase is devoted to the generation of recommendations. The RS exploits both the active user's session and the patterns/relationship obtained from the off-line phase. The active user's session is matched against the patterns of the model to compute the scores of the resources. The recommendation process is thus made on-line. The resources with the highest scores are recommended to the user.

Usage mining approaches have the advantage to be used to compute recommendations without requiring that the active user is identified, or to compute recommendations to an unknown user, which is a very common situation in practice. In e-commerce and m-commerce, it is of high importance to provide users with recommendations when they are not identified, not only to recommend them resources as yet as they arrive on the web site (to improve their satisfaction) but also to attract potential customers.

Sequential Association Rules (SAR) (Srikant and Agrawal, 1996) are are used to capture ordered relationships between resources within sequences of navigation. A SAR is an expression of the form $X \Rightarrow Y$, where X (called the antecedent) and Y (the consequent) are sequences of resources. $X \Rightarrow Y$ is considered as a SAR if both its support and confidence are above two thresholds that have to be fixed.

A SAR means that, when users consult the sequence of resources in X, they usually consult Y. SAR are widely used techniques in RS to discover useful links within or between user sessions of navigation.

The off-line part of SAR-based RS searches all the SAR according to their support and confidence. Most of the algorithms used to mine SAR are incremental as GSP (Srikant and Agrawal, 1996), an APriori-like algorithm, where SAR made up of $k+1$ resources are deduced from the SAR of k resources.

SAR-based RS exploit the sequence of resources in the active session of the user, and compare it to the antecedents of the SAR discovered off-line. A rule that matches the active session (called a matching rule) is a rule with an antecedent that is a subsequence of the active session (as_a) of the active user a. The resources (r_m) that are consequence of the matching rules are then candidates to be recommended.

To compute the score of each resource $S(r_m|as_a)$, the confidence of each matching rule (confidence($X \Rightarrow r_m$)) is used. As several matching rules may have the same consequence, a resource may have several scores. Several policies are used in the literature to deal with this problem as max_policy or sum_policy (Brun and Boyer, 2009).

4 COMPUTATION TIME IN RS

At the opposite of memory-based RS, model-based RS provide recommendations within a small time. Although this feature is an advantage, many works have been dedicated to the reduction of the computation time required to suggest recommendations to users (Yan et al., 1996; Nakagawa and Mobasher, 2003; Brun and Boyer, 2009).

To reduce the computation time, one can focus either on the off-line phase, or on the on-line phase.

4.1 Off-line Focus

Clustering is one way to reduce computation time. In the framework of WUM, (Yan et al., 1996) performs a clustering of sessions and a model is constructed on each cluster. In the on-line phase, the active user session is classified to the previously constructed clusters, and resources are then recommended. Clustering of SAR is also performed. This model is not only lighter than a general model, but also more relevant for the active user, compared to the generation of a single set of SAR.

We can also focus on the way to store the model so as to the information of the model is accessible in a short time. Many works (Mobasher et al., 2002) propose to store navigation patterns in a suffix tree.

(Mobasher et al., 2001) presents a frequent itemsets graph for all kth order markov models that is also used in the framework of SAR.

4.2 On-line Focus

Some other works specifically focus on the on-line phase to decrease the computation time.

A first way to decrease the computation time lies in the reduction of the size of the active session considered, by using a sliding window (Mobasher et al., 2001). This configuration assumes that only the last n visited resources influence the choice of the next resource the user will consult (and thus influences the recommendation). All the resources before the window are not considered, thus reducing the number of resources to consider and the computation time. Obviously, when mining SAR, a sliding window is also used. (Nakagawa and Mobasher, 2003) stores sequential patterns in a frequent sequence trie. As a sliding window is used, only frequent patterns of size equal to the size of the sliding window are considered.

In (Brun and Boyer, 2009) we have proposed anytime WUM RS to provide users with recommendations whenever they want, recommendations can thus be generated even though small time is available.

Most of these works focus on the way to store the model and access it to compute recommendations. To our mind, the most critical part for computing recommendations in these RS is the effective matching of the active session against the patterns discovered offline, that can take time.

5 REDUCING COMPUTATION TIME IN RS

We now focus on the exact way recommendations are computed in data mining RS, and show that this computation is complex. Data mining RS may thus take time to generate recommendations under some conditions. We then focus on a new way to compute predictions, by incrementally constructing the recommendation list.

To our best knowledge, no work has presented the exact way the active user session is matched against the model. We present here how this matching can be performed and show that this task is complex.

Let a be the active user and $as_a = (r_{a1}, r_{a2}, \ldots, r_{an})$ his active session. The RS has to compute the score of each resource $r_m \in R$. To perform these computations, the RS matches the active session against the model.

5.1 Classical Approach

In the framework of SAR, the model is made up of a set of SAR. As shown in section 4, SAR are generally stored in a tree to decrease both the space complexity of the model and the access time to the rules.

Most of the works deal with non contiguous SAR both off-line and on-line, *i.e.* there can be additional resources between the resources of the SAR.

To compute the score of each resource in $r_m \in R$, the system has to find all the rules in the model that match the active session as_a.

To perform this matching, the set of subsequences of the active session as_a is generated. Let *SubSeq* be the function mapping as_a to the list of its sub-sequences. Let $ss = SubSeq(as_a)$ be the resulting set of sub-sequences. Then, given this set, each sub-sequence is searched in the SAR tree. Assuming that reaching a son node in a tree, when placed in a node has a cost of 1, (Nakagawa and Mobasher, 2003) shows that the search of a sequence ss_i in a tree is performed in $O(|ss_i|)$ where $|ss_i|$ is the length of

ss_i. As there are 2^n possible sub-sequences of size between 1 and n in the active session of size n, matching the active session against the set of SAR is made in 2^n searches. As the cost of each search is dependent on its size, the overall cost is equal to $n \cdot 2^{n-1}$. This value is under-evaluated as updating the score of r_m (as several SAR can lead to the same resource) is costly.

In e-commerce or m-commerce, the active session may be large. For example, in supermarkets, a session of 30 products leads to more than 1 *billion* searches in the tree and thus more than 16 *billions* operations in the tree. The number of operations exponentially increases with the size of the active session.

Obviously, when using a sliding window, this number decreases, as the size of the window is lower. However, we have shown that, in the frame of web navigation (Bonnin et al., 2009), the most accurate recommendations are obtained with a sliding window of 10 elements (5k operations).

In conclusion, using SAR in the frame of m-commerce when implementing the RS on the mobile requires a large computation time, and will probably not compute recommendations in real-time.

5.2 Towards Incremental RS

Let a customer be searching in the supermarket the products he is interested in. The recommendations made at a given time are computed on products that the user has accessed previously (the active session of the user). The user then chooses a product that can be in the recommendation list, or not. The RS has then to present to the user a new list of recommended products, etc.

When a user is interested in a product, at the exact moment he takes it in his hand, the RS has to suggest him products related to his interests (and also related to this last product). If the computation time is high, no recommendation may be available at that moment and the user may take another product in his hand before the recommendation is computed. Thus, the system is one phase late. If the user wants a recommendation, he will have to wait and may be disappointed. However, as previously noticed, in the frame of SAR-based RS, when implemented on mobiles, the time required to compute recommendations may be high.

To maximize the user satisfaction, the RS should not be real time but instantaneous. Indeed, in the case of real-time RS, recommendations can be made just at the moment before the user takes in his hand another product. This recommendation is made too late, it should be made instantaneously just after the user takes a product in his hand.

Let us recall that the RS exploits the sequence of the products the user has been interested in (by using SAR here). At a given time t, the recommendations proposed are not independent of the recommended items from the preceding time $t-1$. The history exploited by the RS at the time t is partially identical to the one used at time $t-1$. The only difference is the last item the user took in his hand, all the preceding items considered are similar. We propose to take advantage of the previously computed recommendations to compute new recommendations. We thus propose an **incremental recommender**. We choose to update the score of each product from the previous time $t-1$ by exploiting the new product of time t. The complexity of the recommendation will thus be highly decreased (only the last resource consulted will be studied). The recommendations will be instantaneous.

Let us recall that at each step, only the resources with the highest scores are recommended, but the score of all the resources is computed.

In the case of max_policy, if the confidence of a matching rule is above the score of the resource of the consequence part, then its score is updated and replaced by the new confidence value. In the case of the sum_policy, whatever is the confidence value of a matching rule, the score of the consequence resource is updated: the confidence is added to the score.

The most commonly used mining algorithm (as APriori algorithm), constructs the SAR set step by step, by increasing the size of the antecedent: SAR made up of $k+1$ resources are deduced from the SAR of k resources. This feature allows us to store SAR in a tree, as (Mobasher et al., 2002).

Based on this observation, at time t the set of SAR that match the active session and contain the resource r_{at} are dependent of the set of matching rules from the step $t-1$. The search of the set of matching rules at step t can then be oriented as the candidate matching rules can be partially deduced from step $t-1$.

Let two rules m_{t-1} and m_t with the antecedent of m_{t-1} included in the antecedent of m_t, and the first elements of the antecedent of m_t are similar to the antecedent of m_{t-1}. The last element of m_t is the only difference between the two antecedent parts. If m_{t-1} matches at time $t-1$, then m_t will possibly match at time t if the last resource the user has consulted is the last element of the antecedent of m_t.

In the SAR tree, the two SAR m_{t-1} and m_t are stored in the same branch and m_t is a son node of m_{t-1}. We thus propose to memorize, at each time $t-1$, the set of SAR in the tree that matches the active session. Concretely, a pointer on the node of each matching SAR is stored. At the following time t, these pointers are used to search if the product actually seen by the user is among the son nodes of each pointer.

In cases some matching rules are found, the score of each product is updated and pointers are updated too. The computation complexity of each time will be highly reduced in that case. Classically, with a session of size n, there were 2^n possible sub-sequences. Here, the number of sub-sequences studied is thus equal to the number of sub-sequences from the preceding session $n-1$. This value is thus at most 2^{n-1}, the number of sub-sequences searched is thus divided by at least 2. In terms of overall cost, the cost of each sub-sequence is no more dependent on the length of the sub-sequence but has a fixed cost: 1. The resulting cost is thus 2^{n-1}, compared to $n \cdot 2^{n-1}$ in classical approaches. The cost is thus divided by n.
For example, with a session of size 30, the computation time is thus divided by 30. The recommendations will thus be more likely made instantaneously.

6 CONCLUSIONS

Recommender systems have been a great success in e-commerce applications. M-commerce is a recent application domain that has emerged in the last decade. M-commerce and e-commerce share many properties and we proposed to study the exploitation of well-known RS in the framework of e-commerce in the m-commerce domain. We specifically focused on data mining RS: sequential association rules. This recommendation model has a high time complexity, but due to its implementation on a server, it runs fast enough to be a real-time RS. After having presented the context of application of our research, and our focus on privacy preservation we chose to implement the RS on the mobile side. We have then detailed the SAR model and showed that the computation of recommendations is time consuming. We have then proposed an incremental RS. This RS leads to a smaller complexity and allows to implement SAR-based RS on mobiles.
This work will be pursued by a study of other recommendation models to propose incremental version.

REFERENCES

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art. *IEEE transactions on knowledge and data engineering*, 17(6):734–749.

Bonnin, G., Brun, A., and Boyer, A. (2009). A low-order markov model integrating long-distance histories for collaborative recommender systems. In *Proc. of the ACM Int. Conf. on Intelligent User Interfaces (IUI'09)*, pages 57–66, Sanibel Islands, USA.

Bozdogan, H. (2004). *Statistical Data Mining and Knowledge Discovery*. Chapman & Hall/CRC.

Brun, A. and Boyer, A. (2009). Towards privacy compliant and anytime recommender systems. In *In Proceedings of the E-Commerce and Web Technologies Conference (EC-Web09)*, pages 276–287.

Han, J. and Kamber, M. (2001). *Data Mining: Concepts and Techniques*. The M. Kaufmann Series in DMS.

Hu, W., Yeh, J., and Lee, S. (2006). Adaptive web browsing using web mining technologies for internet-enabled mobile handheld devices. In *Proceedings of the 16th Information Resources Management Association (IRMA 2006) International Conference*.

Huang, Y., Kuo, Y., Chen, J., and Jeng, Y. (2006). Np-miner: A real time recommendation algorithm by using web usage mining. *Knowledge-Based Systems*, 19:272–286.

Lee, S. (2004). A mobile application of client side personalization based on wipi platform. In *Proceedings of the Computational and Information Science Conference (CIS04)*, pages 903–909.

Mobasher, B., Dai, H., Luo, T., and Nakagawa, M. (2001). Effective personalization based on association rule discovery from web usage data. In *in proceedings of the 3rd International Workshop on Web Information and Data Management*, pages 9–15.

Mobasher, B., Dai, H., Luo, T., and Nakagawa, M. (2002). Using sequential and non-sequential patterns for predictive web usage mining tasks. In *Proc. of the IEEE Int. Conf. on Data Mining (ICDM'2002)*.

Nakagawa, M. and Mobasher, B. (2003). Impact of site characteristics on recommendation models based on association rules and sequential patterns. In *Proceedings of the IJCAI'03 Workshop on Intelligent Techniques for Web Personalization*.

Srikant, R. and Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. In *Proc. of the 5th Int. Conf. on Extending Database Technology*, pages 3–17.

Tarasewich, P. (2003). Designing mobile commerce applications. *Communications of the ACM*, 46(12):57–60.

Tveit, A. (2001). Peer-to-peer based recommendations for mobile commerce. In *International Workshop on Mobile Commerce, Proceedings of the 1st international workshop on Mobile commerce*, pages 26–29.

Wan, Z. (2009). Personalized tourism information system in mobile commerce. In *IEEE Int. Conf. on Management of e-Commerce and e-Government*, pages 387–391.

Yan, T., Jacobsen, M., Garcia-Molina, H., and Umeschwar, D. (1996). From user access patterns to dynamic hypertext linking. In *Fifth Int. World Wide Web Conf.*

Zenebe, A., Ozok, A., and Norcio, A. (2005). Personalized recommender systems in e-commerce and m-commerce: A comparative study. In *Proc. of the 11th Int. Conf. on Human-Computer Interaction*.