Open Source Recommendation Systems for Mobile Application

Renata Ghisloti De Souza
LISITE-ISEP
28 rue Notre Dame Des Champs
75006 Paris
renata.ghisloti@isep.fr

Raja Chiky
LISITE-ISEP
28 rue Notre Dame Des Champs
75006 Paris
raja.chiky@isep.fr

Zakia Kazi Aoul
LISITE-ISEP
28 rue Notre Dame Des Champs
75006 Paris
zakia.kazi@isep.fr

ABSTRACT

The aim of Recommender Systems is to suggest useful items to users. Three major techniques can be highlighted in these systems: Collaborative Filtering, Content-Based Filtering and Hybrid Filtering. The collaborative method proposes recommendations based on what a group of users have enjoyed and it is widely used in Open Source Recommender Systems. The work presented in this paper takes place in the context of SoliMobile Project that aims to design, build and implement a package of innovative services focused on the individual in unstable situation (unemployment, homeless, etc.). In this paper, we present a study of open source recommender systems and their usefulness for SoliMobile. The paper also presents how our recommender system is fed by extracting implicit ratings using the techniques of Web Usage Mining.

Categories and Subject Descriptors

H.3.4 [**Systems and Software**]: Performance evaluation (efficiency and effectiveness); H.2.8 [**Database applications**]: Data mining—*Web usage mining*

General Terms

Algorithms, Experimentation, Theory

Keywords

Open source recommender systems, collaborative filtering, Mahout, Web usage mining

1. INTRODUCTION

The amount of information in the web has greatly increased in the past decade. This phenomenon has promoted the advance of the recommender systems research area. These systems intend to help users by providing useful suggestions to them. They may suggest items in different manners, such as comparing the user taste with other users tastes or comparing the users preferences with other items definitions. These two methods are the so called collaborative filtering [1] and content-based filtering [10]. The collaborative method presents advantages over the content-based one. It is more efficient in practice and simpler to implement. Due to this fact, the majority of open source projects choose it. Current open source recommender system projects are usually built on the item-based approach, a type of collaborative filtering. Their features vary on the programming language, extent of documentation and magnitude of the project.

We give in this paper an overview on known recommendation techniques and we analyze open source projects in this field of research. Our interest of recommender systems is justified by the fact that we have to choose one of the studied systems and to integrate it in a complex platform that includes a Web platform, a personalization system and a mobile interface. This platform is developed through the SoliMobile project, funded by ProximaMobile [12].

The SoliMobile project in which we are involved, aims at providing a portal services helping and assisting people who are in different unstable situations. This project provides end users with information to facilitate the process to access to charities services from anywhere. The portal has to offer services adapted to each user profile, taking into account their preferences and navigation traces. Our work aims to provide the user with a recommendation of items (services) based on the profile. The recommendation's main function is to aggregate content from different sources and mobile Web portal and to customize the presentation of services for each user according to his profile. It allows classification or restriction of services into a selection that fits the user profile.

The remainder of this paper is organized as follows. Section 2 describes the context of the work presented in this paper. In Section 3 , we present the global architecture of the SoliMobile Project. Section 4 details the analysis of existing Open Source Recommender Systems. The recommender system used in the project is explained in section 5. Section 6 presents the utility of web usage mining in the recommendation. Finally, Section 7 concludes this paper and gives an outlook upon our ongoing and future research in this area.

2. WORK CONTEXT

The work presented in this paper fits in a collaborative project that aims to design, develop and implement a set of innovative services focused on persons in situations of instability or emerging from instability, in order to help them to find useful information such as jobs, offers of housing, welfare, or medical assistance. The charity association partner in the project observed that a large majority of people in unstable situation own a mobile phone that is considered as a link with family, friends or society. The project aims to facilitate, for the vulnerable people, processes to access charities services using their mobile phone from anywhere. However, different services are not suitable for all persons in a precarious situation. For example, a single mother needs child services such as pediatrics or nursery while an unemployed needs services to find a job or professional training.

Our role in this project is to enhance and customize services to users. In this context, we deal with the implementation of algorithms to adapt to user profile the platform services, to filter them and to show only items that may be of interest. Personnalization according to user profile is based both on data available on the platform (eg. databases), the features and traces of user navigation, and also the social environment of the user (collaborative filtering approach).

Typically, adaptation to the user profile will consolidate the resources (services) to target only the relevant users. Conversely, user profiles will also be ordered to form homogeneous groups in order to assign them to a given resource.

3. GLOBAL ARCHITECTURE

We present in this section the overall architecture of the application, illustrated in Figure 1, in order to show the role of the recommendation in the SoliMobile platform. In fact, end users create an account via the Web platform where many services are provided. The traces of Web browsing (also called logs) are collected from servers to feed the recommender system. These navigation traces will be used to create the user item ratings matrix. Services play the role of items. Information regarding the user profile such as age, address, occupationo or preferences as well as information concerning the description of services such as the category of services (health, employment, child care, etc..) and their addresses will be provided as input of recommender system. These inputs will be sent in XML format through Web services. Once the ratings matrix constructed, the recommendation is made to categorize and customize the layout of proposed services on the mobile phone. The recommendation system will provide as output an XML file that contains a subset of sorted services to be transmitted to the mobile. Traces of mobile browsing will also be used as input to the recommender system to improve results, they can also serve as a feedback to our system.

Our goal is structured along the following lines:

- Construct a generic model for the user profile and also for structural and semantic information of the application in order to integrate new data when needed;

- Select, automatically and dynamically, variables describing the user, the services and the log navigation that improve the quality of the recommendation;

- Ensure the proper functioning of the recommender system in case of registration of a new user whose profile is poor (or nonexistent) or in case of creating new services (items) that no one (in our data set) has yet rated or visited. This problem is well known in the field of information filtering and is referred as ”Cold Start” problem. Almost solutions for the cold-start problem [Lam et al. 2008] are not suitable as they involve users to rate items.

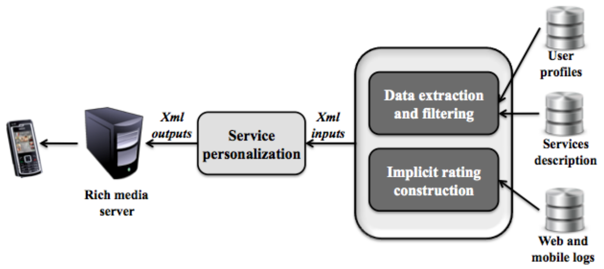

Figure 1: Global architecture of SoliMobile Project

- Develop a generic recommender system, i.e. that adapts to any application. The challenge is to design a real-time recommender system that filters resources dynamically depending on variation in user interests but also on variation in the environment. The idea is to associate with each resource a ranking based on the user profile and its context. We use for that incremental learning techniques [3] and mining data streams [4] that requires a limited number of passes on data and needs to process data on the fly. Using these methods improves computation time and memory space so we can ensure robustness and scalability of the system;

- Define satisfactory indicators in order to assess the quality of the recommendation;

- Conduct a software platform integrating all the tools developed during the project.

  Given the short duration of the project (18 months), we decided to study open source recommender systems. Thus, we present in the following section the related state of the art.

4. OPEN-SOURCE RECOMMENDER SYSTEMS

The growth of Web content and the expansion of e-commerce has deeply increased the interest on recommender systems. This fact has led to the development of some open source projects in the area. Among the recommender systems algorithms available in the Web, we can distinguish the following: Duine [5], Apache Mahout [9], OpenSlopeOne [11], Cofi [2], SUGGEST [13] and Vogoo [14]. All of these projects offer collaborative-filtering implementations, in different programming languages.

The Duine Framework supplies also a hybrid implementation. It is a Java software that presents the content-based and collaborative filtering in a switching engine: it dynamically switches between each prediction given the current state of the data. For example if there aren't many ratings available, it uses the content-based approach, and switches to the collaborative when the scenario changes. It also presents an Explanation API, which can be used to create user-friendly recommendations and a demo application, with a Java Client example.

Mahout constitutes a Java framework in the data mining area. It has incorporated the Taste recommender system, a collaborative engine for personalized recommendations. Vogoo is a PHP framework that implements a collaborative filtering recommender system. It also presents a Slope-One code.

A Java version of the Collaborative Filtering method is implemented in the Cofi library. It was developed by Daniel Lemire [6], the creator of the Slope-One algorithms. There is also a PHP version available in Lemire's webpage. OpenSlopeOne offers a Slope One implementation on PHP that cares about performance.

SUGGEST is a recommendation library made by George Karkys and distributed in a binary format.

Analyzing software in the recommendation area is not a simple task, since it is difficult to define measurement standards. In this work, we propose some criteria of evaluation: types of recommendation implemented by the project, programming language, level of documentation and magnitude of the project.

The documentation was evaluated based on its volume and clarity. It is possible to observe that the volume of documentation presented by Mahout and Duine is remarkably larger than the other systems. Both offer installation and utilization guides and come with a demonstration example. It must be taken into account that OpenSlopeOne and Cofi are smaller projects, and thus, their documentation tend to be smaller. In the Downloads column we have a representation of the magnitude of the project. It is presented the number of times the software, in any version, was downloaded from its source. Although Mahout does not present its number, its very populated mailing lists shows that it is a widely used software.

The two projects that stood out were Apache Mahout and Duine. We installed and tested them in order to verify which one was more applicable to our work. Both of them are based on the Java technology and present a demonstration example with the Movielens data set. The fact that Mahout is a greater project and has multiple machine-learning algorithms made it more interesting to our research. Also, its module structure encouraged us to choose it.

5. APACHE MAHOUT

The Apache Mahout is a solid project in the Data Mining area. It is a framework that features various scalable machine-learning algorithms. It is programmed using the Java language and runs with Maven project manager. In April 2008, it has incorporated the Taste Recommender System, a Java framework for providing personalized recommendations. Besides Taste, it also offers clustering algorithms and a Map Reduce implementation.

Taste is a very consistent and flexible collaborative filtering engine and supports the user-based, item-based and Slope-one recommender systems. It can be easily modified due to its well-structured modules abstractions. The package defines four interfaces: DataModel, UserSimilarity and ItemSimilarity, UserNeighborhood and Recommender.

With these interfaces, it is possible to adapt the framework to read different types of data, personalize the recommendation or even create new recommendation methods.

The User Similarity and Item Similarity abstractions are responsible for calculating the similarity between a pair of users or items. Their function usually returns a value from 0 to 1 indicating the level of resemblance, being 1 the most similar possible.

Trough the DataModel interface is made the access to the data set. It is possible to retrieve and store the data from databases or from filesystems (MySQLJDBCDataModel and FileDataModel respectively). The functions developed in this interface are used by the Similarity abstraction to help computing the similarity.

The main interface in Taste is Recommender. It is responsible for actually making the recommendations to the user by comparing items or by determining users with similar taste (item-based and user-based techniques). The Recommender access the similarity interface and uses its functions to compare a pair of users or items. It then collects the highest similarity values to offer as recommendations.

The UserNeighborhood is an assistant interface that helps to define the neighborhood in the user-based recommendation technique. It is know that, for greater data sets, the item-based technique provides better results. For that, many companies choose to use this approach, such as Amazon [7]. With the Mahout framework, it is not different, the item-based method generally runs faster and provides more accurate recommendation.

In our project, we choose to adapt the Slope One (a type of item-based algorithm) approach to our problem. Here follows a simple Java application example of how to initiate a recommendation with the Slope One technique:

```
1. DataModel model =
          new FileDataModel(new File("data.txt"));
2. Recommender recommender =
          new SlopeOneRecommender(model);
3. Recommender cachingRecommender =
          new CachingRecommender(recommender);
```

The challenge in adapting this approach to our project was the fact that our input data file was available in the XML format, a type not handled by Mahout. It then had to incorporate another file in the DataModel interface. We create a program that deals with the XML input files. To test this new data handler, we used the Movielens data set. A pack with one million ratings was converted to the XML type to be used as example. With this data set and the XMLfile, the running time of the Slope One algorithm takes less than one minute.

6. WEB USAGE MINING FOR RECOMMENDATION

One objective of the SoliMobile project is to develop a recommender system that has to be, as much as possible, the least intrusive. This implies that the system is based only on information that the user can be free to provide (explicit data) and must run properly with alternatives such as implicit data mining. To meet this need, we are studying how to append Web browsing analysis to the recommender system as done in [8]. Web browsing analysis becomes almost necessary for extracting and understanding user behaviors.

	Implementation	Language	Documentation	Downloads
Mahout	Item-based, User-based, Slope One	Java	High	Not available
Duine	User-based, Content Filtering	Java	High	1,113
Cofi	Item-based	Java	Low	Not available
OpenSlopeOne	Slope One	PHP	Low	653
Vogoo	Slope One, Item-based	PHP	Medium	2,128
SUGGEST	Item-based, user-based	C	Medium	Not available

Table 1: Utilisation ratio of each method.

In recent years, Web usage mining has become an important issue in the field of data mining. The term, Web usage mining focuses on predicting and learning the users preferences on the Internet. Generally, the data for Web usage mining are the user interactions on the web, usually residing on Web clients, Web servers, and proxy servers. The aim of Web usage mining is to analyze user behavior through analysis of its interaction with the Web platform. This analysis is particularly focused on all the users clicks where visiting the web application (also known as clickstream analysis). The interest of Web usage mining in our framework is to enrich the input of recommender system with user data extracted from the raw clickstream data, in order to refine the user profiles and behavioral patterns. The analysis of Web logs can also be used as implicit feedback of the user which will allow to assess the performance of models involved in the recommender system.

It is obvious that Web logs change over time for several reasons: an update of the Web application content or structure, a change in the user preferences, a change in the execution context, etc. This is why it is important to take into account the temporal dimension in the analysis of Web usage. To consider the temporal data in a dynamic way, we plan to use the techniques of data streams mining. By definition, data stream is a real-time, continuous, ordered (implicitly by arrival time or explicitly by timestamp) sequence of items. It is impossible to control the order in which items arrive, nor is it feasible to locally store a stream in its entirety. Therefore, all the treatment have to be applied in one pass. Several techniques for mining data streams have emerged as CluStream for clustering, StreamSamp for sampling, VFDT for incremental decision trees, etc. The reader may refer to [4] for more explanations on these different techniques.

7. CONCLUSIONS

In this paper, we presented the problem that we deal with in the SoliMobile project. Then, we presented the global architecture that is under development in this project. This architecture includes a recommender system to customize the services offered to users based on their profile and their browsing history. Given the limited duration of the project, we opted for an open source recommender system that is modular in order to easily integrate future developments, in particular the use of Web usage mining to address the problem of cold start.

In this paper, we also discussed several points concerning the issue of treatment of the temporal dimension in data analysis. The raised issues demonstrate the need for defining new methods or adapting existing methods for extracting knowledge and monitoring changing and evolutive data. Although there are many efficient methods for extracting knowledge, few studies have been devoted to the issue of temporary evolutive data.

8. REFERENCES

[1] John S. Breese, John S. Breese, David Heckerman, and Carl" Kadie. Empirical analysis of predictive algorithms for collaborative filtering. pages 43–52, 1998.

[2] Cofi. http://www.nongnu.org/cofi/.

[3] Antoine Cornuéjols. Getting order independence in incremental learning. In *ECML '93: Proceedings of the European Conference on Machine Learning*, pages 196–212, London, UK, 1993. Springer-Verlag.

[4] Baptiste Csernel, Fabrice Clerot, and Georges Hébrail. Streamsamp: Datastream clustering over tilted windows through sampling. *ECML PKDD 2006: the International Workshop on Knowledge Discovery from Data Streams (IWKDDS-2006)*, 2006.

[5] Duine. http://www.duineframework.org/.

[6] Daniel Lemire and Anna Maclachlan ". Slope one predictors for online rating-based collaborative filtering. 2005.

[7] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

[8] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. Personalized news recommendation based on click behavior. In *IUI '10: Proceeding of the 14th international conference on Intelligent user interfaces*, pages 31–40, New York, NY, USA, 2010. ACM.

[9] Mahout. http://mahout.apache.org/.

[10] Raymond J. Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. In *DL '00: Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204, New York, NY, USA, 2000. ACM.

[11] OpenSlopeOne. http://code.google.com/p/openslopeone/.

[12] ProximaMobile. http://www.proximamobile.fr/.

[13] Suggest. http://glaros.dtc.umn.edu/gkhome/suggest/overview.

[14] Vogoo. http://www.vogoo-api.com/.