

DEVELOPMENT OF A RECOMMENDER SYSTEM FOR MOBILE APPLICATIONS

¹ALI Martha Ngozi

alimarthangozi@gmail.com

²Mgbeafulike Ike

ike.mgbeafulike@gmail.com

³Okide S.O

samuelokide@yahoo.com

^{1,2} Computer Science Department, Chukwuemeka Odumegwu University, Uli.

³Computer Science Department, Nnamdi Azikiwe University, Awka, Anambra State

ABSTRACT

With the amount of mobile applications available increasing rapidly, users have to put a lot of effort into finding applications of interest. The purpose of this thesis is to investigate how to aid users in the process of discovering new mobile applications by providing them with recommendations. A prototype system is then built as a proof-of concept. The work of the thesis is divided into three phases where the aim of the first phase is to study related work and related systems to identify promising concepts and features. During the second phase, a prototype system is designed and implemented. The outcome and result of the first two phases is then evaluated and analyzed in the third and final phase. The prototype system integrates and extends an existing recommender engine previously used to recommend media items. As a part of the system, an Android application is developed, which observes user actions and presents recommended applications to the user. In parallel to the development, the system was tested by a small group of users recruited among colleagues at Ericsson. The data generated during this test period is analyzed to show the usefulness of observed user actions over explicit ratings and the dependency on context for application usage.

Keywords: Applications, rapidly, investigate, recommendations, prototype, implemented,

1.0 INTRODUCTION

With a rapid increase of mobile applications available for download it has become a very difficult task for the user to find exactly what he wants. In order to choose one among all of the offered applications he has to put in a lot of effort and waste a lot of time. Most famous Apple's slogan: "There's an app for that!" has proven itself to be correct long time ago with the help of both iStore and Google Play which, today, offer over 1 million applications for iOS and Android platforms. To make things easier for the users app markets are using recommender systems, which provide users with app suggestions. Unfortunately these kinds of recommendations neglect the very mobility of the users and their constantly changing context although every application naturally serves a special purpose not every application can be addressed to a special context of use.

1.1 Statement of the Problems

Sometimes application users find it difficult to identify the application suitable for them to be used in their mobile devices in a way that minimizes the information overload, i.e.

- i. The difficulty in making decision caused by the presence of too much application.
- ii. Time wasting in finding suitable applications.
- iii. Being unable to find suitable application

1.2 Aim and Objectives of the Study

The aim of this project work is to design and implement a recommender system for mobile application. The objectives of the study are as follows;

- i. To recommend suitable applications for the users.
- ii. To save the user's time of searching for application.
- iii. To assist the user in getting their application easily.

2.0 Literature Review

Application Recommender system has been so extensively used these days that it has become a preferable choice for researchers. First paper on recommender system was published in year 1998. Since then a significant number of papers had been published. Different factors have been explained to increase the reliability of recommender system. In year 2005 John O'Donovan , Barry Smyth have taken trust as the percentage of correct predictions that a profile has made in general (profile-level trust) or with respect to a particular item (item-level trust). Authors have described a number of ways in which these different types of trust values might be incorporated into a standard collaborative filtering algorithm and evaluated each against a tried-and-tens benchmark approach and on a standard data-set. This decreases the prediction error by 22%.

In year 2007 Paul Resnick proposed an idea of "influence limiter algorithm" in recommender system. This algorithm prevents any attack which shows the irrelevant result for our search. This algorithm limits the number of content that an attacker can modify. In the same year Michael P. O'Mahony proposed on development of a course recommender system for University College Dublin's on-line enrolment application. In his paper in support of this application he also gave a empirically evaluation of then ongoing approach using historical student enrolment data was given and it shows that promising performance was achieved with proposed design. Again in year 2007 a trust based recommender system was proposed by Punam Bedi, Harmeet Kaur, Sudeep Marwaha for semantic net. Description of the design of a recommender system that uses knowledge stored in the form of ontologies is given. The interactions amongst the peer agents for generating recommendations are based on the trust network that exists between them.

3.0 Proposed system and Implementation

Design and implementation of a recommender system for mobile applications to developed proof-of-concept recommendation system is an Android application and since the development for Android platform is carried out in Java the Eclipse IDE was a tool that was chosen for the implementation. The Android platform provides APIs to access the required context information as well as to track application usage. AppDetective recommends based

on two different datasets. One set is the context information and the other set are the information collected by Tracker which tracks application usage on the device. The Tracker is running in the background as a service and records the start time of the applications and their runtime. To preserve power it only keeps track of applications while the device is not in standby mode which is logical because the user can use apps only when the device is in the active mode. Also, the user can control by himself when does he want the tracker to collect information and when does he want the tracker to stop tracking by clicking the corresponding buttons in the main menu. All Tracker's data are stored into a local SQLite database. The context data is collected every time the app is started or when the location changes. The mobile application also communicates with web services via HTTP requests in order to get information about the weather prognosis for the current location. Also it uses HTTP requests to execute queries for the Android Market. As there is no official Google Play API, an open source Android API is used for searching for apps, for getting app screenshots and for downloading apps. Both responses from the weather service and Google Play are received in the JSON format for easier parsing.

3.1 Implementation

The developed proof-of-concept recommendation system is an Android application and since the development for Android platform is carried out in Java the Eclipse IDE was a tool that was chosen for the implementation. The Android platform provides APIs to access the required context information as well as to track application usage. AppDetective recommends based on two different datasets. One set is the context information and the other set are the information collected by Tracker which tracks application usage on the device. The Tracker is running in the background as a service and records the start time of the applications and their runtime. To preserve power it only keeps track of applications while the device is not in standby mode which is logical because the user can use apps only when the device is in the active mode. Also, the user can control by himself when does he want the tracker to collect information and when does he want the tracker to stop tracking by clicking the corresponding buttons in the main menu. All Tracker's data are stored into a local SQLite database. The context data is collected every time the app is started or when the location changes. The mobile application also communicates with web services via HTTP requests in order to get information about the weather prognosis for the current location. Also it uses HTTP requests to execute queries for the Android Market. As there is no official Google Play API, an open source Android API is used for searching for apps, for getting app screenshots and for downloading apps. Both responses from the weather service and Google Play are received in the JSON format for easier parsing.

4.0 The Result

By manually searching through numerous mobile applications in the application stores like Google store, Samsung app, I-croud etc. But this can be as tedious as ever. It not an easy task to carry out as such will take a whole lots of time. And will not be an easy task for a mobile user.

1. Recommender systems collect feedback information from the users that rate items. They connect the users, which manifest similarities in ratings and use that information for recommendations. In short, collaborative filtering algorithms recommend those items, which were approved by similar users. This kind of recommending is also called personalized recommending.
2. Content-based recommendations use methods which focus on item descriptions and characteristics. These methods create a profile for every user (content-based profile) that memorizes characteristics of previously viewed items. Comparing the information from the profile and item descriptions recommender system tries to find items suitable for the user. In other words content-based algorithms recommend those items that are similar to the items, which have been previously viewed by the user.

3.0 CONCLUSION

In order to help the users to choose an application they want to download a recommender system was developed. User's choice was guided by recommendations based on user's current context and their preferences. It demonstrates how the constantly changing context and the constantly changing preferences influence recommendations can. It tries to exploit all of the advantages a mobile phone offers, and context and give to the user the most useful recommendations.

As it was said, very little of the literature has focused on context-based recommendations of mobile apps, not to mention global similarity measuring among apps. The current understanding of both user preferences and market features is far from sufficient. In this work these problems were therefore studied and an experimental solution was built. Overall, the goals set out for the project were met and the basic problems and challenges were overcome.

6.0 REFERENCES

Adam Mustansar Ali Ghazanfar and adam prugel-bennett," A Scable Accurate Hybrid recommender system," in thurd international conference on knowledge discovery and data mining,2010,app. 94-98.

Gediminas adomavicius and Alexander tuzhilin," towards the next generartion of recommender systems: a survey of the state-of-the-art and possible extension,"IEEE transaction on knowledge and data engineering. Vol.17, no.6 2005,pp.734-749.

Andrew I. schein, alexandrinpopescul,lyle H. Ungar and david M.

Pennok,"methods and matrices for cold-start recommendations<"ACM, 2002.

Francesco ricci,"mobile recommender systems,"2010.

Wolfgang woerndi, Christian schueller and rolf wojtech,” A Hybrid Recommender system for context-aware recommendations of mobile Applications,”IEEE, 2007

Wolfgang woerndl and georg groh,” utilizing physical and social Context to improve recommender system”, IEEE, 2007

Koljahagelich DietmanJannach,” effectiveness of different recommender algorithms in the mobile internet: A case study of,”AAAI, 2009.

Smark Dunlop & Stephen Brewster,” the challenges of mobile devices for HCT”. Personal and ubiquitous computing, 2002.

Maryam hosseini-pozveh, mohamadaliNematbaksh and naser movahhedina,” a multidimensional approach for context-aware recommendation in mobile commerce,” in international journal of computer science and information security, 2009.

Adomavicious, G., Sankaranarayanan, R., Sen., and Tuxhilim, A.,” Incorporating contextual information in recommender system using a multidimensional approach,” ACM Trans. Inf. Syst., 23(1):103-145, 2005.