

App Recommendation System Based on Collaborative Filtering

Aziz Nasridinov, Young-Ho Park
Dept. of Multimedia Science, Sookmyung Women's University, Seoul, South Korea
{aziz, yhpark}@sm.ac.kr

Abstract

It gives to users a difficulty for searching between this huge numbers of programs. Searching the best applications for our needs is a big challenge today. In this paper, we propose a study on collaborative filtering based app recommendation system. The proposed method is composed of three steps. In the first step, we extract the data set from the target website. In the second step, we parse the extracted raw data according to the types, and store in a database. In the third, we perform recommendations based on the stored data in database

1. Introduction

With thousands of different mobile applications available and more than 2.5 billion of total downloads the android market is one of the most used applications container of the world [1]. Unfortunately it gives to users many difficulties for searching between this huge numbers of mobile applications. In other words, searching the best applications for our needs is a big challenge today. Such growth in volumes of mobile applications has created a need to develop a recommender system that assists the users to take the right choice, when searching for a mobile application.

A lot of mobile applications have emerged that provide app recommendation and aggregation services which filter, rank and recommend the best apps to end users. For example, Karatzoglou et al. [2] proposed a mobile app discovery method through context-aware recommendations. The method proposes a Djinn model, a novel context-aware collaborative filtering algorithm for implicit feedback data that is based on tensor factorization. Yan et al. [3] presented the AppJoy system that makes personalized application recommendations by analyzing how the user actually uses her installed applications. Based on all participants' application usage records, AppJoy employs an item-based collaborative filtering algorithm for individualized recommendations.

In this paper, we propose a study on collaborative filtering based app recommendation system. The proposed method is composed of three steps. In the first step, we extract the data set from the target website. In the second step, we parse the extracted raw data according to the types, and store in a database. In the third, we perform recommendations based on the stored data in database. The rest of this paper proceeds in as follows. Section 2 describes the related work. Section 3 explains the proposed method. Section 4 presents the implementation results of proposed method. Section 5 highlights conclusions.

2. Related Work

The Android Market presents a list of "Featured" and "Top" applications. "Featured" applications are characterized by a recent release and a rapidly increasing download rate. "Top" applications have a high number of downloads and are separated between free and paid version. Moreover the market shows the stars score and the comments. This allows a user to see what other users think about the app.

There have been many efforts on making recommendation systems for mobile applications. For example, Karatzoglou et al. [2] proposed a mobile app discovery method through context-aware recommendations. The method proposes a Djinn model, a novel context-aware collaborative filtering algorithm for implicit feedback data that is based on tensor factorization. The method deals with the dual challenge of building a preference model in the absence of explicit feedback information from the user and incorporating real contextual information into a single recommendation model. The proposed method can be used in any context-aware recommendation setting with implicit feedback data.

Yan et al. [3] presented the AppJoy system that makes personalized application recommendations by analyzing how the user actually uses her installed applications. Based on all participants' application usage records, AppJoy employs an item-based collaborative filtering algorithm for individualized recommendations. This is analogous to the "vote by your feet" approach, in which what the user does matters more when providing her application needs. Compared with other solutions, AppJoy is completely automatic without requiring manual input and AppJoy is adaptive to the changes of the user's application taste.

3. Collaborative Filtering-Based App Recommendation System

The proposed method is composed of three steps. The rest of this section explains the steps of the proposed method in detail.

Step 1 (Data Extraction): We developed a web crawling method to collect the raw data from the target website. In our method, we collect the raw data from Calcutta Communication [5] website that provides all the latest app rankings from the App Store and Google Play. Calcutta Communication shows the weekly trend data for the past three years, and offers the most reliable ranking data in the market. It also has detailed information about apps according to their types, such as genre, app developing company. Thus, the developed crawler first collects raw data from the website according to categories. Then it examines each category page to find the list of links of all the apps in each category, using this to extract and prepare raw data on each app within each category for a data parsing process.

Step 2 (Data Parsing): The raw data is parsed according to a set of pattern templates. We implemented patterns to

capture information about Category, Description, Price, Customers' Rating, and the Rank of Downloads of each app. Note that we can use our method to different websites that stores the app ranking data by simply modifying only the data extractor and the parsing step.

Step 3 (Recommendation): We analyze the raw data obtained in the first and second phases, and generate various recommendations using Collaborative Filtering technique. Similarity measure is the most important part of collaborative filtering algorithms. Choosing a proper similarity function can obviously improve the performance of such algorithms. In the literature, there are several functions to calculate the similarity between the users. Cosine similarity is the most widely used in collaborative filtering. The two users are thought as two vectors, while the similarity between two vectors is calculated by their cosine angle [2]:

$$\cos(u_i, u_j) = \frac{\sum_{k=1}^m v_{ik} v_{jk}}{\sqrt{\sum_{k=1}^m u_{ik}^2 \sum_{k=1}^m u_{jk}^2}} \quad (1)$$

where v_{ik} is the score of the element k given by the user i .

With this formula we compare two different users looking to all the rates without any distinction.

4. Implementation of App Recommendation System

In this section, we describe the implementation of app recommendation system. We implement the proposed system in Java language. We used the computer with an Intel i5-760 quad core processor running at 2.80GHz in Windows Operating System with 16GB of main memory. The java implementation of collaborative filtering is Apache Mahout. Apache Mahout [4] is a new open source project by the Apache Software Foundation (ASF) with the primary goal of creating scalable machine-learning algorithms that are free to use under the Apache license. Mahout contains implementations for clustering, categorization, collaborative filtering, and evolutionary programming.

An example [5] of the process of creating recommendations by user-based collaborative filtering is shown in Figure 1. To the left is the rating matrix R with 6 users and 8 items and ratings in the range 1 to 5 (stars). We want to create recommendations for the active user u_a shown at the bottom of the matrix. To find the k -neighborhood (i.e., the k nearest neighbors) we calculate the similarity between the active user and all other users based on their ratings in the database and then select the k users with the highest similarity. We used equation 1 in order to calculate the similarities between users. The $k = 3$ nearest neighbors (u_4 , u_1 and u_3) are selected and marked in the database. To generate an aggregated estimated rating, we compute the average ratings in the neighborhood for each item not rated by the active user. To create a top- N recommendation list, the items are ordered by predicted rating. In Figure 1, the order in the top- N list is i_2 , i_1 , i_7 and i_5 .

We used the user-based collaborative filtering in order to implement the proposed method. However, there are two main problems of user-based collaborative filtering. The whole user database has to be kept in memory and that expensive similarity computation between the active user and

all other users in the database has to be performed.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
u_1	?	4.0	4.0	2.0	1.0	2.0	?	?
u_2	3.0	?	?	?	5.0	1.0	?	?
u_3	3.0	?	?	3.0	2.0	2.0	?	3.0
u_4	4.0	?	?	2.0	1.0	1.0	2.0	4.0
u_5	1.0	1.0	?	?	?	?	?	1.0
u_6	?	1.0	?	?	1.0	1.0	?	1.0
u_a	?	?	4.0	3.0	?	1.0	?	5.0
r_a	3.5	4.0			1.3		2.0	

Figure 1. User-based collaborative filtering [5]

5. Conclusion

In this paper, we propose a study on collaborative filtering based app recommendation system. The result of our methods can provide useful information for organizations in competitive advantages over rival organizations and result in business benefits, such as more effective marketing and increased revenue. It has benefits to the users too. For example, finding new application will be faster. Users always want to know if there are valuable applications that is good to have in their smart-phones. Moreover, a user is able to find the application that really fits with his needs. Today without a recommender system a user maybe installs an application just because he knows only that one.

Acknowledgement

This work was supported by the IT R&D program of MKE/KEIT. [10041854, Development of a smart home service platform with real-time danger prediction and prevention for safety residential environments].

Reference

- [1] S. Perez. Mobile app martetplace. Available at: <http://www.readwriteweb.com>
- [2] A. Karatzoglou, L. Baltrunas, K. Church, M. Bohmer, "Climbing the app wall: enabling mobile app discovery through context-aware recommendations," The Proceedings of the 21st ACM international conference on Information and knowledge management (CIKM'12), pp. 2527-2530, 2012.
- [3] B. Yan, G. Chen, "AppJoy: personalized mobile application discovery," The Proceeding MobiSys '11 Proceedings of the 9th international conference on Mobile systems, applications, and services, pp. 113-126, 2011.
- [4] Apache Mahout. Available at: <http://mahout.apache.org/>
- [5] Hahsler M. Lab for Developing and Testing Recommender Algorithms. Available at: <http://r-forge.r-project.org/projects/recommenderlab/>