A Study and Implementation of a Movie Recommendation System in a Cloud-based Environment

Jaime Raigoza, California State University, Chico, CA, USA
Vikrantsinh Karande, California State University, Chico, CA, USA

ABSTRACT

The availability of huge amounts of data in recent years have led users to being faced with an overload of choices. The outcome is a growth on the importance of recommendation systems due to their ability to solve this choice overload problem, by providing users with the most relevant products from many possible choices. For producing recommendations, things like a user’s psychological profile, their browsing history and movie ratings from other users can be considered. To determine how strongly two user’s behavior are related to each other, a Pearson correlation coefficient value is often calculated. In this paper, we study the recommendation system on a proposed cloud based environment to produce a list of recommended movies based on a user’s profile information. Based on the Software-as-a-Service (SaaS) model implemented, we discuss the concepts such as collaborative filtering and content-based filtering. Given a MovieLens data-set, our results indicate that the proposed approach can provide a high performance in terms of precision, and generate more reliable and personalized movie recommendations, when given a greater number of movies rated by a user. An evaluation was done under minimal known data, which commonly leads to the cold-start problem.

KEYWORDS

Collaborative Filtering, Content Filtering, Pearson Correlation Formula, Recommendation Systems

INTRODUCTION

With today’s enormous usage of the Internet, the result has led to vast amounts of data being easily available. This availability of excess data forces consumers to do an excessive amount of decision making, whether it is searching for a good movie or looking for a good restaurant. For guidance purposes, organizations have implemented recommendation frameworks to offer the consumers some assistance with coping with this data explosion. According to Zhang et al. (2014), 75% of the viewing activity on Netflix is based on recommendations.

Recommendation systems are a subclass of data filtering applications which have become an efficient way to solve the data overload problem (Ng, 2013). The aim of recommendation systems is to automatically generate suggested items (i.e. books, music, news articles, movies, shop items, etc.) for users based on their historical preferences and thus saving their online search time by providing an accurate recommendation list of items. Amazon.com is one of the most popular recommendation system being used (Linden et al., 2003). Movie recommendation systems are also widely-used applications which aim to help customers to choose preferred movies intelligently from a vast amount of movie choices, typically from a library. The recommendation framework for books at Amazon.

DOI: 10.4018/IJGHPC.2017010103

Copyright © 2017, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
com and for movies at imdb.com are good examples of online recommendation frameworks which have been developed and are widely used.

The goal is to merge the knowledge of the expert with the user’s psychological profile to filter the accessible data and provide the user with the most relevant data that he/she would like. Two main ideal models for the filtering purpose that we discuss in this paper are the collaborative approach and the content-based approach.

A collaborative approach uses the preferences of other people with similar tastes for recommending items to the user whereas a content-based recommendation system uses the user’s browsing history to recommend new items that the user enjoyed previously (Melville & Sindhwan, 2010). Chen and McLeod (2006) discuss an extensive study on different types of collaborative-based filtering methods being used within recommendation systems. We implemented and extended a movie recommendation system freely available on GitHub (Movie Recommendation, 2015). We then discuss an example that provides a recommendation list of movies based on the similarity of user’s rating history. This example is implemented on the concepts of genre based filtering. The user’s ratings of a movie are compared with the user’s profile to determine a recommendation.

A common problem with the collaborative filtering approach is having minimal information that can be compared with to determine a recommendation. An example is a new user who has not built an abundant profile. This is referred to as the cold-start problem. Methods to tackle this problem have been proposed by researchers, such as by Nadimi-Shahraki and Bahadorpour (2014) who have developed a method to quickly gather profile information using an Ask-to-rate technique. A quick interview is performed to quickly gather pertinent information and integrate it into a recommendation system.

**RELATED WORK**

Recommender frameworks, being a subclass of data filtering framework tries to predict the choices that a client would make for an item (Ng, 2013). Recommendation systems typically produce a list of recommended items in one of two ways - through collaborative filtering or content-based filtering. The collaborative filtering (CF) approach works by collecting and maintaining user ratings for an item and comparing them to rating behavior of several other users for finding an appropriate match to make recommendations for the user. The content-based filtering approach utilizes a series of discrete characteristics of an item to recommend additional items with similar characteristics (Melville & Sindhwan, 2010).

Last.fm and Pandora Radio are the two recommendation systems which can explain the difference between content-based filtering and collaborative filtering. Last.fm observes the user’s behavior as in what group of bands or songs the client has listened to and compares them with those from other clients. It then makes a station of recommended songs. This allows the last.fm to play songs which the client never heard of before and yet are frequently listened to by other clients who have similar preferences. Pandora is a service which plays the songs of our selection of a certain genre. User’s positive and negative feedback ratings are considered in making recommendations of songs more accurately. Pandora utilizes the properties of a song or an artist keeping in mind the end goal to seed a “station” that plays music with similar properties. This is an example of a content-based approach. Whereas Last.fm requires a lot of data about a client to be able to make an accurate recommendation, Pandora requires minimum data to begin, it is much more constrained in degree (Bao & Xia, 2012; Melville & Sindhwan, 2010).

Amazon.com uses a form of collaborative filtering based on an item-to-item similarity approach. Relationships are formed from items which are purchased as a group and then combines those items to similar items to determine a recommendation (Linden et al. 2003). The key to their improved performance is that work can be done offline and by computing accurate similarities, the system is better able to make recommendations (Sarwar et al., 2001). Implementation of collaborative-based filtering is a common approach to determine similarities decisions. The Amazon.com recommendation
Fault Tolerance  
[www.igi-global.com/chapter/fault-tolerance/43107?camid=4v1a](www.igi-global.com/chapter/fault-tolerance/43107?camid=4v1a)

Speculative Scheduling of Parameter Sweep Applications using Job Behaviour Descriptions  
[www.igi-global.com/chapter/speculative-scheduling-parameter-sweep-applications/54922?camid=4v1a](www.igi-global.com/chapter/speculative-scheduling-parameter-sweep-applications/54922?camid=4v1a)