Movie Recommendation System

1155084561 DENG Weijun 1155084518 PENG Yibo
1155089134 GUAN Chenye 1155089150 Ou Kaishen

Group Email address: 1155089150@link.cuhk.edu.hk

Motivation

Our motivation is to build a movie recommendation system.

Related topic

The most related parts between this course and our project are recommendation system, item similarity, map-reduce algorithm.

Deliverable

By the end of the project, the deliverables we plan to submit is a recommendation system. The input and the output of the recommendation are following:

Input: A user with ratings for some sample movies.
Output: Top N (e.g. 5) movies that this user may be interested in.

Dataset

The dataset we plan to use is “MovieLens” (20M version, 190MB zip). Stable benchmark dataset. 20 million ratings and 465,000 tag applications applied to 27,000 movies by 138,000 users. Includes tag genome data with 12 million relevance scores across 1,100 tags.

Technique

The technique we are going to apply includes: $\boxed{1}$. Content-based recommendation; $\boxed{2}$. Collaborative filtering recommendation; $\boxed{3}$, LFM (Latent Factor Model, to be confirmed).

Detail of the technique:

First of all, Content-based systems examine properties of the items (e.g. movie in our case) recommended. In the content-base system, we must construct for each item a profile, which is a record of its important characteristics. For example, a movie can be represent by a vector

[Tom cruise, Steven Allan Spielberg, 1995, adventure]. If there are M movies, each have 10 features, then there are at most M*10 features total (if neither is same). After creating the item profile, we have to create the user profile using the same features. To be more specific, we need to use movie features to represent users' preference. For instance, if user A only see two movies, say, star war and mission impossible 4, we can use the vector [Tom cruise, Sky walker] to represent his preference. Finally, we use LSH first to reduce the computation range and then compute the cosine between user profile and item profile.

Collaborative Filtering is different from the content-based recommendation. In this filtering approach, we do not create a vector for items (e.g. movies) but just use the column in the utility matrix. Also, no user profile is used but we use the row of the utility matrix to represent a user. Finally, recommendation for a user U is then made by looking at the users that are most similar to U, and then recommend items that these similar users like. To be more specific, we have to ways to start. A. We start from looking for similar users. For user 'U' and item 'I', first we find n (e.g.10) users that are similar to user U. Then in these n similar users, we count only those who has rated item I (e.g. 4 people give a rating to I) and we average the ratings. This ratings is the possible prediction that user U will give to item I. B. The second choice is that we start form looking for the similar items. Still for user U and item I, firstly find the m (e.g. 100) items similar to I. Here, we use the cosine between two columns and many possible ways. Then, in these m similar items, we count only those the user U has given a ratings and average them. This ratings is also the possible prediction that user U will give to item I. Since users' number is too large (138,000) we would choose the second way.

Criteria to demonstrate the results

Three criteria are shown to demonstrate our analytical results: 1. Precision; 2. Call-back rate; 3. Popularity. More details of these criteria will be explained in the future report.

Existing work

Existing work include the Netflix competition. Also, there are some movie recommendation sites like (IMDB).

Milestones

11.1-11.15: A python prototype code (single machine version) of the content-based recommendation system should be completed.
11.16-11.22: Build the JAVA Hadoop program and make the algorithm distributed to handle the big datasets. A python prototype code of LFM is attempted.
11.23-11.29: Try to implement a JAVA Hadoop Distributed program of LFM. Test the System.
11.30-12.7: Prepare for the presentation and write the final report.
12.8-12.13: Prepare for the final report and the related materials.