Movie Recommendation System

April 7, 2020

1 Abstract

On the Internet, where the number of choices is overwhelming, there is need to filter, prioritize and efficiently deliver relevant information in order to alleviate the problem of information overload, which has created a potential problem to many Internet users. Recommender systems solve this problem by searching through large volume of dynamically generated information to provide users with personalized content and services. This paper explores the different characteristics and potentials of different prediction techniques in recommendation systems in order to serve as a compass for research and practice in the field of recommendation systems. Recommender systems are often based on Collaborative Filtering (CF) , which relies only on past user behavior—e.g., their previous transactions or product ratings—and does not require the creation of explicit profiles. Notably, CF techniques require no do- main knowledge and avoid the need for extensive data collection. In addition, relying directly on user behavior allows uncovering complex and unexpected patterns that would be difficult or impossible to profile using known data attributes. As a consequence, CF attracted much of attention in the past decade, resulting in significant progress and being adopted by some successful commercial systems,including Amazon, TiVo and Netflix.

2 Solution Overview

We plan to use the techniques of collaborative filtering to to recommend movies to users based on their interests. Collaborative Filtering is the techniques where we use the past transactions to predict the relations between future users and products.

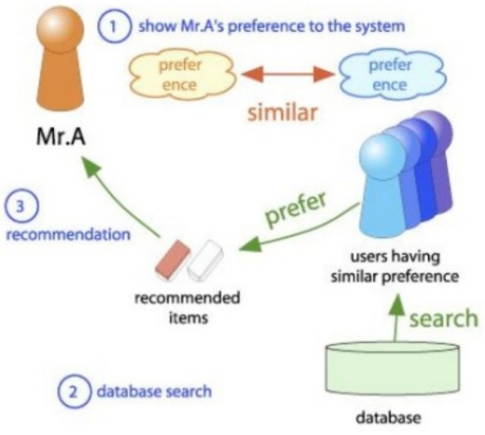

Figure 1: Basic Idea behind Collaborative Filtering

The two major techniques in collaborative filtering is the Neighborhood model and the latent factor model. Neighborhood methods are centered on computing the relationships between items or, alternatively, between users. An item oriented approach evaluates the preference of a user to an item based on ratings of similar items by the same user. In a sense, these methods transform users to the item space by viewing them as baskets of rated items.This way, we no longer need to compare users to items, but rather directly relate items to items. Latent factor models, such as Singular Value Decomposition (SVD), comprise an alternative approach by transforming both items and users to the same latent factor space, thus making them directly comparable. In our project, we plan to combine both the above models forming a hybrid model which would provide benefits of both the neighborhood and latent factor models. Recommender systems rely on different types of input. Most convenient is the high quality explicit feedback, which includes explicit input by users regarding their interest in products. But there is also another form of of feedback (Implicit Feedback) which we plan to include in our project.The dataset does not only tell us the rating values, but also which movies users rate, regardless of how they rated these movies. In other words, a user implicitly tells us about her preferences by choosing to voice her opinion and vote a (high or low) rating. This would be the implicit rating.

3 Dataset

For this project,we have used MovieLens dataset, which is one of the standard datasets used for implementing and testing recommender engines.

MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota.

This data set consists of:

- 100,000 ratings (1-5) from 943 users on 1682 movies.
- Each user has rated at least 20 movies.

Features of the dataset -

- The ratings are at intervals of 0.5 on a 5-point scale,starting from 0.5 and going to 5.
- The data is randomly ordered.
- The time stamps are unix seconds since 1/1/1970 UTC
- Each user is represented by an id, and no other information is provided.

The dataset is split into 2 partitions, train and test sets.(80-20) split.

4 Exploratory Data Analysis

In statistics, exploratory data analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task.So let's get right to it.

4.1 Dataset Overview

```
Final_Data.head()
```

	0	1	2	3
0	196	242	3	881250949
1	186	302	3	891717742
2	22	377	1	878887116
3	244	51	2	880606923
4	166	346	1	886397596

Figure 2: Overview of Dataset

The Timestamp is in UTC format.
Total Data:
Total number of movie ratings = 100000
Number of unique users = 943
Number of unique movies = 1682

4.2 Distribution of Ratings

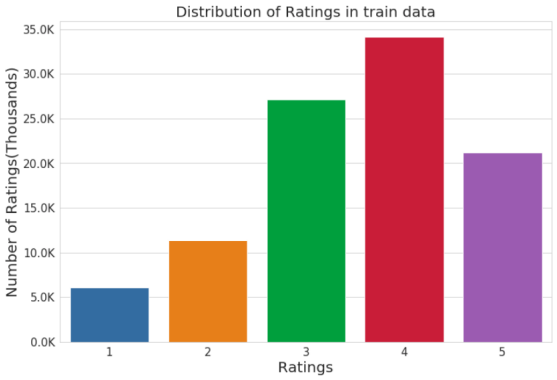

Figure 3: Distribution of Ratings data vs Number of Ratings

The distribution of Ratings in Training Data shows the maximum ratings of around 35K rated around 4.

4.3 Ratings per month

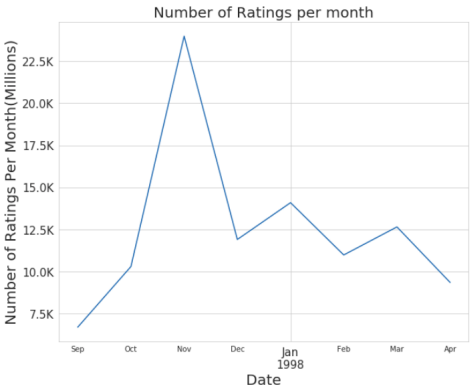

Figure 4: Number of Ratings per month

4.4 PDF and CDF analysis of Data

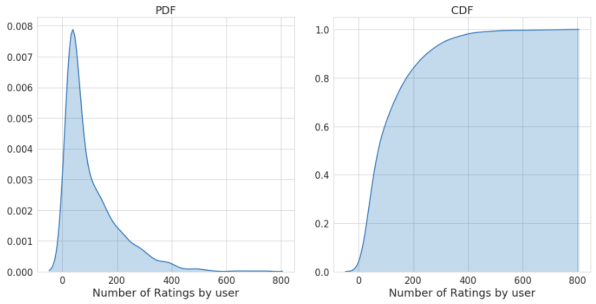

Figure 5: PDF and CDF analysis of Data

The PDF graph shows that almost all of the users give very few ratings. There are very few users who's ratings count is high.Similarly, above CDF graph shows that almost 99% of users give very few ratings.

4.5 Ratings per Movie

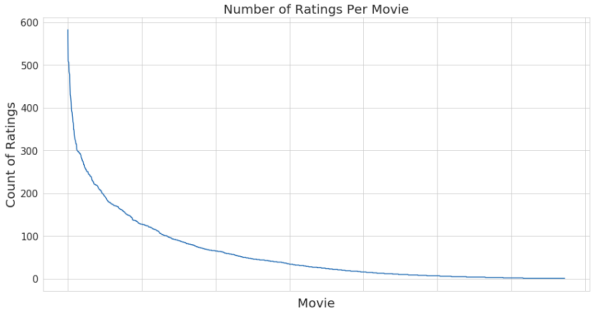

Figure 6: Number of Ratings per Movie

5 Data Flow Diagram

Based on the ratings provided by the user, movie recommendations are made by fitting the user profile against the trained collaborative model. The flow is described by the following diagram:

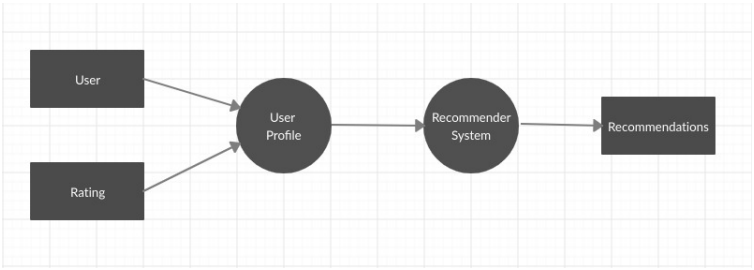

Figure 7: Data Flow Diagram

References

[1] Yehuda Koren *Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model.*

[2] Ml-100k Dataset : *https://grouplens.org/datasets/movielens/100k/*

[3] Winning the Netflix Prize: A Summary *http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summary/*