CMSC 12300 Project Proposal

Group Member: Boyang Qu, Ruixi Li, Tianxin Zheng, Haowen Shang

User-based and Item-based Movie Recommendation using MapReduce and Hadoop

Introduction

Recommendation system is broadly used to make personalized recommendations. In this project, we plan to use Hadoop MapReduce to build a movie recommendation system based on user's historical preference. Map-Reduce is a programming framework used for processing and generating large datasets. We plan to utilize both user-based and item-based collaborative filtering algorithms to complete the recommendation table. The dataset we will work on is MovieLens dataset, which contains 27,753,444 ratings.

Dataset

The datasets we will employ is the large latest movie-rating datasets from MovieLens. It contains 27,753,444 ratings from 283,228 users on 58,098 movies. The dataset was collected by GroupLens Research between January 09, 1995 and September 26, 2018, and users were selected randomly. For privacy reason, the users and movies are represented by unique Id numbers.

Four primary datasets will be used for our mapreduce algorithm. First, the movie-ratings dataset contains all rating data, with each line representing one rating by one user on one movie. Information on each line is in the format of userId, movieId, rating, timestamp. Ratings are provided on a 5-star scale with half-start increment, and timestamp indicates number of seconds after January 1, 1970 midnight UTC. The datasets is ordered by User Id and then movie Id. Similarly, the movie tags dataset represents the tags by each user on each movie. Each line is in format of userId, movieId, tag, timestamp. The tags are words or phrases that user think is representative of a movie, such as epic, sci-fi, history, and psychothriller. Third, the movie dataset provides information of movie genres, and each line is in the format of movieId, title, genres. Genres are pipe-separated lists including action, adventure, animation, and other common genres. Lastly, the genome-tags datasets labels each tag with a tagId, and genomes-score dataset contains a relevance score for the tag place on each movie. Each line of the first dataset is in format of tagId, tag, and that of the second dataset is in format of movieId, tagId, relevance.

The dataset can be downloaded from http://grouplens.org/datasets/movielens/. Our dataset is 1.22GB in total. We believe the dataset has a scale large enough for MapReduce exploration while not too large for efficient manipulation.

Hypotheses

Our project goal is to construct a recommendation system which predicts the rating or preference that the user would give to a movie. We use two algorithms in our recommendation, the item-based algorithm and the user-based algorithm.

For the item-based algorithm, the way to do that is to calculate the similarity score of two pair of movies. The higher the score, the more similar the two movies are, and the higher the likelihood the user is likely to appreciate the recommendation. There are in total of four steps in our project. Firstly, for each pair of movies X and Y in our dataset, we will find out all the people who rated both X and Y. Secondly, we will use these ratings to form a Movie X vector and a Movie Y vector. Thirdly, we will calculate the correlation between those two vectors. Lastly, when someone watches a movie, we will recommend the movies most correlated with it based on the correlation score.

For the user-based algorithm, the way to do that is to calculate the similarity score of two pair of users. The higher the score, the more similar the two users are, and the higher the likelihood the user is likely to appreciate the movies liked by another user who is similar to he/she. There are in total of four steps in our project. Firstly, for each pair of users A and B in our dataset, we will find out all the same movies they rated. Secondly, we will use these movies to form a Rating X vector and a Rating Y vector. Thirdly, we will calculate the correlation between those two vectors and get the similarity matrix. Lastly, when user A watches a movie, we will recommend the movies highly-rated by the user who has the highest similarity score.

To put it shortly, we will be using a user's ratings and tags (with high relevance), together with the category, to evaluate the item differences between movies. Then we will also employ a user-based approach to predict a user's rating for the movie using the rating by a particular user whose ratings for other movies are similar to the evaluated user.

Algorithms

-MapReduce:
Our first algorithm is item-based. We will use MapReduce to find similar movies based on movie rating.
The first task is to find all movies and their ratings watched by each person. We will use a mapper to extract user and (movie, rating) pair. Then use a reducer to groups all (movie, rating) pairs by user.
The second task is to find every pair of movies that were watched by same person. and measure the similarity of their rating across all users who watched both movies. We will use a mapper to

get every pair of movies viewed by the same user and get their ratings. The key-value pair looks like:(movie1, movie2) - (rating1, rating2). Then use a reducer to compute rating-based similarity between each movie pair (movie1, movie2) - (similarity, number of users who saw both).

The third task is to sort results by movie, then by similarity strength. We will use a mapper to make movie name, similarity score the sorting key and use a reducer to get the final sorted output.

The final output will be movies sorted by name with a list of movies ordered by similarity.

Our second algorithm is user-based. We will use MapReduce to calculate the similarity score of two pair of users. Please see details in hypothesis part.