Panitan Satamalee

CS 235 (MSOL)

Recommendation of Movies (Software 1)

1. Introduction

Recommendation systems are evolving technologies that apply data analysis towards helping consumers get products they need. The current most popular algorithm for recommendation system is the collaborative filtering (CF) algorithm, which finds a target user's item by finding other users with similar interests. Currently, there are three main problem using this algorithm, mentioned in the three literary surveys in the "Related Work" section, which are scalability, sparsity, and cold-starting. With growing amounts of data and consumer base, the amount of users that CF needs to consider grows. The time it takes to compute each recommendation also grows, so this show how scalability problem of the system. The sparsity and cold-start deals with the data the system gets. Users usually don't have to provide that much data to start getting recommendations (sparsity due to few ratings and total number of items available in the database), and, when we want to give new users recommendations there is no data to start the prediction on (cold-start). This project aims to compare a few improvements to the CF algorithm to overcome a few of these obstacles.

2. The Data

We will be using the dataset from GroupLens Research (which collected data from MovieLens). The dataset consists of 5 different files. "ratings.csv" and "tags.csv" contains the ratings and tags of a given user ID for a given movie ID. "movies.csv" and "links.csv" contains

the movie ID, its title, the movie's genre, and the links for the movie's IMDb and TMDb profiles. There are also the tag genome files that contain the relevance scores between the relationships of a specific tag to a movie.

The first step in preprocessing the data is to combine all the files into a single table. The ratings and tags files are combined such that there is a row for each unique user and each column is either a rating or tag for a movie. Then the tags is used to find the corresponding tag ID which would give its relevance. We then generate potentially useful features such as the average rating, or relevance, for certain movie genre for a user, or we can also generate some useful information from the movie titles such as series, location, etc. Stratified sampling, or random sampling, would be implemented to create a training and testing/validation set.

3. General Collaborative Filtering

By using ratings, and possibly tags and relevance, we cluster users together based on the ratings that they give to each movie and find some commonality between user's opinions on a movie or maybe towards a genre. Given a user and his ratings and tags for the movies he has seen, we want to be able to give a movie recommendation that he would enjoy. Another way of saying this would be that we want to recommend a movie where he would give a high rating, somewhere between 3-5 stars. If a list of movies a user likes is given, then we can also assume that the ratings he would have given those movies are probably 3-4, and then we can try to give a recommendation off of that.

4. Related Works

Collaborative Filtering with Clustering

The first recommended change to the CF algorithm is to use clustering on the data to form centers that represent groups of users or items that can be used on the overall CF algorithm (1). This means that, instead of scanning through several users and comparing several of their items for similarity, this algorithm sets the amount of users to a limited number of groups being compared and a limited number of item centers that are compared. The article argues that this technique works towards the challenge of sparsity and scalability. By grouping items and users that are similar, the sparsity isn't much of a problem because items with no ratings may be grouped with ones with ratings so all of the items in a cluster will probably have the same rating. The scalability is also solved due to having a limited number of user and item groups regardless of how many users or items increase over time, since they will just be added to the group. The results of the article shows that this method produces recommendations that are similar, and maybe even better, to the original CF algorithm.

Incremental Singular Value Decomposition for Recommenders

The third type of recommendation system suggests an entirely new approach using matrix decomposition, or more specifically the incremental singular value decomposition (3). Singular value decomposition recommenders currently exist and are known for very fast online performance because it requires a few simple operations for each recommendation. The challenge of this system is that computing the SVD matrix is very expensive and, with new data being added consistently for current recommendation systems, could lead to very expensive and inefficient operations. This article suggests the use of incremental SVD algorithms to consistently update the SVD matrix that already exists. The suggested algorithms include the folding-in algorithm, which trades a significantly sped up process for lower prediction, and the

SVD-update algorithm, which requires more time than folding-in but results in higher recommendation quality. By using SVD, we can overcome the problem of scalability really easily since the computation costs of recommendations will be severely reduced, and with the folding-in or SVD-update techniques the SVD could be comparable to CF algorithms.

Collaborative Filtering with Neural Networks

The second type of system suggests the use of neural networks to improve the ability for recommenders (2). In general, this article suggests the use of general matrix factorization (GMF) and multi-layer perceptron (MLP) for a recommendation system. They also explore the combination of both forming the neural matrix factorization model (NeuMF).

References

1. Gong, S. (2010). A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering. *Journal of Software, Vol. 5, No. 7*.
2. He, X., Lizi, L., Zhang, H., Nei, L., Hu, X., Chua, T. (2017). Neural Collaborative Filtering. *International World Wide Web Conference Committee (IW3C2)*.
3. Sarwar, B., Karypis, G., Riedl, J. (2002). Incremental Singular Value Decomposition Algorithm for Highly Scalable Recommender Systems. *GroupLens Research Group / Army HPC Research Center*.

Current Progress Report

I have obtained the data and have begun working on merging the data into a large array of users and movie items with ratings as the values. I will then build each of the three

mentioned recommendation systems in the “Related Work” section and compare them to each other, with the potential of combining these ideas into one large recommendation system to see if working together they can overcome the weaknesses of the CF recommendation systems.