Programming Project 3
Machine Learning, Spring 2019

Introduction

In this assignment you will create a recommender system for movies and use it to make movie recommendations for yourself and your friends.

What to submit

Submit one text document (.doc or .pdf) with your answers to all parts. For full credit this must be well formatted, well organized, and easily understood. I will primarily grade your written report. It should be polished and well written and should thoroughly address all the questions posed in each part.

Also submit .py files that include your code (you may submit one or several). For full credit your code must be well organized and any unclear parts must be commented. I will only be running your code if I have a question about your written report.

Step 1: Obtaining the data

The MovieLens data set is a well know test data set for recommender systems. You can download it here:

https://grouplens.org/datasets/movielens/

Scroll down to the section “recommended for education and development" and get the small data set:

ml-latest-small.zip

Step 2: Loading the Data

You may use my code provided in load_data.py to get started. This code will create

> df_ratings_matrix: This data frame has a row for each user and a column for each movie. If a user did not rate a movie the value will be 0. There are 610 users and 9724 movies, therefore there are 610*9724 ratings, however most are zero. This will be useful for step 4.
>
> df_tag_strings: This data frame has a row for each movie and a single string for each with all the tags and genre names concatenated and separated by a space. This will be useful for step 3. Note this only includes the 1572 movies that have user tags, not the entire set of 9724.

Step 3: Content-based Recommendation

Content based recommender systems make recommendations based on the properties of the items (in this case the tags of the movies). Create a recommender system that works as follows:

- Select a movie that was liked by a user
- Find other similar movies to recommend

To find similar movies you will need a measure of similarity of the tags and genres. A simple method would be to apply a count vectorizer as a first step (see the spam email example code), then map the values to 1 or 0 (either the word appears or it doesn’t), then use the number of words in common as a measure of similarity. More sophisticated measures are also possible. Select a measure that you think makes sense and works well.

Use your recommender to recommend a movie for yourself and/or your friends. (Pick a few movies in the data set that you like and generate a list of similar ones as determined by your measure of similarity.)

Step 4: Collaborative Filtering

Collaborative filtering based recommender systems make recommendations based on the history and properties of the other users. Create a recommender system that works as follows:

- Given a user, find another, similar user based on movies they have rated
- Find other movies liked by the similar user that the first user has not yet rated

To find similar users you will need a measure of similarity of their ratings. Do consider dislikes as well as likes. Examples include correlation and cosine similarity of the vectors, but there are other options as well. Select a measure that you think makes sense and works well.

Use your recommender to recommend a movie for yourself and/or your friends. (Rate some of the movies in the data set, then find a similar user in the dataset and see if there are movies that user rated highly that you have not seen.)

Step 5: Discussion of Results

In your write up:

- Describe the recommendations that your system made (what movies did it recommend for you/your friends and what ratings of yours were the recommendations based on).
- Describe the similarity measures that you used and why you decided to use them. Note: you do not need to consider a huge number of options, but do give the question of how to measure similarity some thought and describe the methods that you considered or tried. In particular, if you tried more than one method, did one measure seem to work better (give better recommendations) than another?
- How well did the recommender systems work? Do the recommendations seem reasonable? Are they movies you would consider watching? Was one system better than the other?
- How could you determine how well the recommenders perform in a more rigorous way than your anecdotal experience described in the previous bullet point? You do not need to implement this (unless you want to), but describe how you could compare your two systems' performance on the MovieLens dataset.

Extension to Final Project

If you wish, you may extend this assignment and turn it into a final project by improving your recommender system. Some ways to improve it include:

- Using the links.csv file to get more information about the movies from their IMDB webpages. This will require some web scrapping code to access information on the web. With this information you can make better content based recommendations.
- Use matrix factorization collaborative filtering methods
- Implement the final bullet point of step 5 and more rigorously evaluate your system(s)
- Use deep learning
- Any other improvements you can think of