ML Minor Project Final Report

by

Abhishek Mittal 2013005
Kartik Maji 2013051

Introduction

We want to suggest a person some movies to watch if he/she has some 3-5 movies in his mind similar to which he/she will like to watch one . We will output a rank to recommend the user to watch a movie. We will be using MovieLens database for this purpose.
Dataset : http://grouplens.org/datasets/movielens/
Input : user_id, 3-5 movie names along with ratings that user would like to give to these movies (which are in the dataset)
Output : All unwatched movies ordered with their watching preference score.

Dataset and Evaluation

The dataset being used here is spread across multiple files (tables) which are related by some common attributes (Primary key, foreign key concept from Relational DB).

There are 6 files:-
u.data u.info u.item u.genre u.user u.occupation

U.data : This is the file having main data which contains 100,000 ratings by 943 users on 1682 movies where tab separted values are for following attributes:-
user id | item id | rating | timestamp
Users and movies are being represented by their IDs (Primary Key), rating is the score given by the user to the movie on a scale of 1-5, and timestamp depicts the time at which the rating was given (irrelevant to us).
All other required information related to users and movies are in u.user and u.item respectively.

U.item : It contains demographic information about the user and has tab separated lists of:-
user id | age | gender | occupation | zip code

The attribute names are self explanatory.

U.items : It contains information about the movies with 24 tab separated attributes in following order:-
movie id | movie title | release date | video release date | IMDb URL |

unknown | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |

The first 5 attribute names are self explanatory, while last 19 attributes represents genres of the movies. A genre attribute can only 1 or 0 value where 1 depicts that the movie belongs to this genre while 0 represents otherwise.

Redundant files:-
U.info: It just contains the number of distinct users, movies, ratings:-
U.genre: It contains list of all genres
U.occupation: It contains list of all the occupations

We are not extracting any features.

We are using precision and recall for evaluation as these are the ones used for recommendation engines generally.

Methodology

We have chosen Linear Regression and Collaborative Filtering. We are using cosine similarity to find similar movies. Since we are still not feeling much confident in applying these independently so we followed some tutorials.

Linear Regression

A simple linear regression can be done by taking user input rating as labels, and other attributes of movies as feature set. Then we can predict a linear hyperplane that for given movies will predict ratings.

Collaborative Filtering

Created a basic collaborative filtering model using cosine similarity (https://brenocon.com/blog/2012/03/cosine-similarity-pearson-correlation-and-ols-coefficients/)
This works on past user patterns. This approach utilizes user rating information to register the comparability between users and movies. This is utilized for making suggestions. It's effective and simple to implement. Similarity computation between movies or users is an essential piece of this approach. We have used cosine similarity(Also known as vector-based similarity) as a part of our project.

Results

A basic movie recommendation system(using collaborative filtering) based on users past ratings on another movies.

user_id	movie_id	score	rank
1	423	0.980611449435	1
1	202	0.949577563819	2
1	655	0.830657517637	3
1	403	0.776640447724	4
1	568	0.753207756818	5
1	385	0.73268492099	6
2	50	1.12562584877	1
2	181	1.06517731685	2
2	7	1.05039239159	3
2	121	0.941627963231	4
2	117	0.792605129572	5
2	9	0.764417666655	6
3	313	0.635376662016	1
3	328	0.603288030083	2
3	315	0.542258712378	3
3	331	0.535507185893	4
3	332	0.531669611281	5
3	100	0.512969842011	6
4	50	1.13114770821	1
4	288	1.04871511459	2
4	181	0.950599938631	3
4	7	0.941777880703	4
4	302	0.913902146476	5
4	121	0.899338160242	6

Precision and recall summary statistics of our generated model.
Recall is TP/(TP + FN) whereas precision is TP/(TP+FP) where TP stands for True Positive, FN stands for False Negative and FP stands for False Positive.
https://turi.com/products/create/docs/generated/graphlab.recommender.util.precision_recall_by_user.html, http://stats.stackexchange.com/a/62626.

Precision and recall summary statistics by cutoff

cutoff	mean_precision	mean_recall
1	0.383881230117	0.0383881230117
2	0.325556733828	0.0651113467656
3	0.28773418169	0.0863202545069
4	0.266436903499	0.1065747614
5	0.246871686108	0.123435843054
6	0.229586426299	0.137751855779
7	0.215270413574	0.150689289502
8	0.202942735949	0.162354188759
9	0.192294096854	0.173064687169
10	0.184199363733	0.184199363733

Movie Recommendation for user_id 946(My user id).

user_id	movie_id	score	rank
946	241	0.889785885811	1
946	739	0.827600434422	2
946	161	0.804931521416	3
946	226	0.774169787765	4
946	550	0.761774480343	5
946	195	0.757970735431	6

Reference

http://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/itembased.html