Online Movie Recommendation System (OMRES)

Yusuf Aytaş
Mustafa Gündoğan

Kemal Eroğlu
Fethi Burak Sazoğlu

Content

- Introduction
- Problem Statement
- Background
- Proposed Solution
- Future Work
- Experiment Results
- Conclusion

Introduction

- People seek information via words, recommendation letters, news reports etc.
- Recommendation systems imitate this social process to enable quick filtering of the information on the web
- Lots of companies try to offer services that involve recommendations to address the right user groups.

Problem Statement

- Providing related content out of relevant and irrelevant collection of items to users of online service providers [1]

- OMRES (Online Movie Recommendation System) aims to recommend movies to users based on user-movie (item) ratings.

Problem Statement

- Given a set of users with their previous ratings for a set of movies, can we predict the rating they will assign to a movie they have not previously rated?

- **Ex.** “Which movie will you like” given that you have seen X-Men, X-Men II, X-Men : The Last Stand and users who saw these movies also liked “X-Men Origins : Wolverine”?

Background (Dataset)

- MovieLens is a data set that provides 10000054 user ratings on movies.
- 95580 tags applied to 10681 movies by 71567 users.
- Users of MovieLens were selected randomly.
- All users rated at least 20 movies.
- Each user represented by a unique id.

Background (Dataset)

- u.data : The full u data set, 100000 ratings by 943 users on 1682 items.
- u.info : The number of users, items, and ratings in the u data set.
- u.item : Information about the items
  - 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 |
  - 1 indicates the movie is of that genre, a 0 indicates it is not
- u.genre : A list of the genres.
- u.user : Information about the users
  - user id | age | gender | occupation | zip code

Background (Dataset Stats)

- Total ratings possible = 71567 (user) * 10681 (movies) = 764.407.127 ~ **750 million**
- Total available = 100.000.054
- The User x Movies matrix has 650 million entries **missing**
- Sparse Data

Problem Background

- Earlier systems implemented in 1990s.
  - GroupLens (Usenet articles) [Resnick, 1997]
  - Siteseer (Cross linking technical papers)[Resnick, 1997]
  - Tapestry (email filtering) [Goldberg, 1992]

- Earlier solutions provided for users to rate the item.

Problem Background

- Item-to-item collaborative filtering (people who buy x also buy y), an algorithm popularized by Amazon.com's recommender system
- Last.fm recommends music based on a comparison of the listening habits of similar users
- Facebook, MySpace, LinkedIn, and other social networks use collaborative filtering to recommend new friends

Background For the Solution

- Content Based Filtering
  - Selects items based on the correlation between the content of the items and the user's preferences
  - Compare user profile to content of each item

- Collaborative Filtering
  - chooses items based on the correlation between people with similar preferences.
  - Rate items based on ratings of the users that rated the same items

Background For the Solution

- There are 3 types of CF
- Memory-based
- Model-based
- Hybrid
  - Memory based and Model-based
  - Content Based and Collaborative Filtering

Background For the Solution

- Memory based Collaborative Filtering
  - Neighbour-based filtering
  - Item-based and user-based correlation
- Pearson Correlation for similarity
- Recommendation based on weighted sum of others' ratings
- Prediction for user a on item i [1]

$$\text{Prediction}(a, i) = \text{Avg. Rating for a} + \frac{\sum_{u \in U} (\textit{Rating of item i by u}) * (\textit{weigth between a and u})}{\textit{Sum of the total weights}}$$

Background For the Solution

- K-NN
- Given a query instance q(movieId, UserId).
- Find the distance of this instance with all the users who rated this movie.
- Of the these users select the K users that are nearest to the query instance as its neighborhood.
- Average the rating of the users from this neighborhood for this particular movie.

Proposed Solution

- C-KNN by AtulS. Kulkarni[3]
- Computation of similarities between the user and the remaining group
- Construction of neighborhood information
- Computation of recommendation based on weighted average of the neighbors ratings on movies.

Experiment Results

- Root Mean Square Error (RMSE) as evaluation measure

$$\text{RMSE} = \sqrt{\frac{1}{n} * \sum_{\{i,j\}} (p_{i,j} - r_{i,j})^2}$$

where n is the number of total ratings

$p_{i,j}$ is the prediction for user i on item j

$r_{i,j}$ is the the actual rating

Experiment Results

- Current Netflix system had RMSE 0.95 before they conducted a challenge to improve this RMSE.
- Winner of the challenge had RMSE 0.85, which is 10% improvement.
- We reached RMSE of 1.24 which is still high considering 0.85, because they used hybrid of different Collaborative Filtering techniques.

Future Work

- K-NN method
  - Different values of K can be tried
  - Distributed processing of this problem
  - Distance weighting the contributions from neighbor

- C-K-NN
  - K-Means clustering then applying K-NN
  - Trying different # of clusters

Conclusion

- Recommendation systems provide content for us by taking what other people recommend as well as our selections into account

- Collaborative Filtering is a widely used solution for this problem which we make use of in our project

References

[1] Xiaoyuan Su and Taghi M. Khoshgoftaar, “A Survey of Collaborative Filtering Techniques,” Advances in Artificial Intelligence, vol. 2009, Article ID 421425, 19 pages, 2009

[2] R. M. Bell, Y. Koren and C. Volinsky, "The BellKor solution to the Netflix grand prize," March 2012. [Online]. Available at:

http://www2.research.att.com/~volinsky/netflix/ProgressPrize2007BellKorSolution.pdf

[3] A Nearest Neighbor Approach using Clustering on the Netflix Prize Data

Questions