SDL Documentation  
Movie Recommender Systems

Sanjana (2013103602 ) & Senthil R ( 2013103602 )

Dataset Used : http://grouplens.org/datasets/movielens/

horizontal line

Please find the code we did till now, in our github repo,

**Code Repo Link :**

<https://github.com/senthil-ramasamy/MovieLensDataset-SDL-R-Project>

**Analysis Doc Link ( already sent ) :** <https://drive.google.com/file/d/0B_tALASpOlEIVjNNV3lsdXdkYndaaFJBaGFRMk1FNTY2cmtB/view?usp=sharing>

**Libraries Used**

* **Data.table**
* **Proxy**
* **Vegan**
* **Matrix**
* **Reshape2**
* **Recommender Lab**
* **Shiny**

**Movie Recommender Engine**

**Two approaches Used,**

1. **Content based**

**Short Abstract :**

Till now, we have built a Movie recommender Engine with R based on MovieLens Data Set. We built the recommender system using two approaches-Content Based Filtering and Collaborative Filtering. The Content based Filtering approach analyses an item user interacted with and provides recommendations of other item with similar features. In our case,the item is movie and features would be genres,actors,directors etc. We have build a content based recommender system based on genres. UBCF based method is explained in next part.

**In Detail :**

The code does the following,

* Data preprocessing, pipe separated genres had to split
* Created matrix with columns representing every unique genre
* Convert into Data Frame for processing user profile matrix in the next step
* ( Temporary Conversion for faster processing , I first convert the ratings into a binary format to keep things simple. ratings of 4 and 5 are mapped to 1, representing likes, and ratings of 3 and below are mapped to -1, representing dislikes. )
* Dcast function is used to build user profile matrix ( change format to wide from long )
* To generate simple user profile matrix, calculated the dot product of the movie genre matrix and the binaryratings matrix. **This user profiles shows the aggregated inclination of each user towards movie genres.**
* Generate two separate matrix, where in the rows,i.e, movies which has no ratings are removed and another vice- versa, not necessary just for easier processing. (Also since there was observed differences in numbers between both, coz of null sets)
* In order to generate user profile similarity, we are used **Jaccard distance** method of finding from vegan library
* **Recommend movies - Simple content based method**

**2. User Based Collaborative Filtering**

**Short Abstract :**

Now that we have user profiles, we are assuming that users like similar items, and retrieve movies that are closest in similarity to a user’s profile, which represents a user’s preference for an item’s feature. Collaborative Filtering System works like how word of mouth works in the real world. For example, if Alice liked movies A,B and C and Bob liked movies A and B, then Bob is more likely to lik C too. It groups users with similar likings and provides recommendations based on that.To implement Collaborative Filtering,we are using the recommenderlab package available in R.

**In Detail :**

* We first create the rating matrix
* Notes :
  + Cosine Similarity Method in UBCF is used. Nearest Neighbours considered 30.
  + The predicted movie ratings of the user will be derived from the 5 nearest neighbors in its neighborhood.
  + When the predicted item ratings are obtained, the top 10 most highly predicted ratings will be returned as the recommendations.
* First converted rating matrix into a recommenderlab sparse matrix.
* Normalize the data and obtain recommendations

**Next To Do**

* We have discarded timestamp of ratings during the data cleaning process, since lots of null values where there, so that should be imputed and some analysis based on that could be done.
* The idea during the first pitch was that user will choose 3 movies, based on that and his already existing user profile, recommendations will be shown. Now based on this, that extra wrapper should be built
* Evaluation Scheme is provided by the recommenderlab package, so we have to evaluate our top-n recommender.