REALTIME RECOMMENDATION SYSTEM(Explicit Feedback)

This document contains information about implementaion, working and using this system.

Dataset: I am using MovieLens 1M dataset, which contains data of movies rated by user.

http://grouplens.org/datasets/movielens/1m/

I am going to use item based collaborative filtering, in this similarities between items(here movies) is determined and then users with similar items are recommended to items found most similar to others.

Frameworks: python-recsys(for SVD algorithm), numpy, pandas

Working:

Three datasets are imported movies, ratings and users. Since we are using item-item based collboratove filtering, I am only interested in ratings dataset.

Ratings(user_id, movie_id, ratings, timestamp)

I am only interested in the first 3 columns as others are irrelevant.

Dataset: Ratings(user_id, movie_id, ratings)

As this is only item-item based filtering item, user_id and rating is required to run matrix factorisation(SVD).

Create matrix where rows are user_id and cols are movie_id and values are ratings in which we can predict missing values of unrated movies by users.

Factorisation is done on the matrix and missing ratings are computed and user gets suggestions on the movies.

User supplies explicit feedback on some of the unrated movies previously not rated by the user, recomputation is called and new suggestions are shown and feedback to show more suggestions is appreciated for more recommendations.

DEEP LEARNING BASED RECOMMENDATION SYSTEM

Although previous systems are widely used in recommendation systems but it is very much data dependent and the most complex and advanced system are based on deep learning.

I wanted to show you this system but it takes a lot of time(2-3 weeks) to implement and more time to test and train this system.

I just want to give you an overview about real time implementation of these systems.

In case of Movie Recommendation, we will take two types of dataset, movies and users.
Movies(Name, Genre, Directors, Actors, Date, IMDB, …... more than 20 vars)
Users(Name, Age, Gender, Movies, Address, Language … many vars)

Now with a large dataset of more than 1M with more than 20 or 30 variables, we can design a recurrent neural network to learn individually about movies and users.

Example Layers or Links of Neural Networks:

Movies:

Movie->Genre->Director->Actor->IMDB->Date->.....->Least Dependent Var

Users:

User->Name->Age->Address->Language->Movies->....->Least Dependent Var

Now that we have learned about the movies, we can use:

Supervised Learning: Recommend New Movies, Learn more about trends in movies based on users.

Unsupervised Learning: Similarity Between Users and Movies or Movies-Movies and Users-Users.

Implementation:

Framework: TensorFlow, PyCaffe, Torch
Computations: CPU, GPU(CUDA) for Parallel Computations
Algorithms: RNN(Supervised), LSTM(Supervised), Autoencoders(unsupervised), MLR(Hybrid), Similarity(Pearson, Cosine)