Description

We are using the user-based collaborative filtering approach for our recommender system. For each user, the system will recommend movies based on how other similar users like them. We choose this approach instead of content-based approach or item-based collaborative filtering because we believe that the users' tastes are complex and are not simply depending on the movies characters themselves. Thus, we tend to find the similarities between users and recommend movies to them from their "neighbors".

The followings are the detailed steps in our approach.

1.) We read the "user_reviews" file and transform it into a N*M dataframe. The row represents users and the columns represent movies. Each element of the matrix (i,j) represents the rating of user i to the movie j. If the user has not rated the movie, the element is zero.
2.) Then we calculate the correlations between each pair of users and store the data into a N*N dataframe.
3.) For each user i without rating movie j, we pick other users who have rated the movie, compute the weighted ratings by multiplying the users' correlations with their movies ratings, sum them up and divide by the number of users, and we finally get the predicted rating for the movie j by user i.
4.) Finally, we sort the movies by their ratings. Movies with the highest ratings are more likely to be favored by the users, so we will recommend these movies to the users.

After running our recommender system, the suggested movies to the users are,

***Vincent**:*

Blue Like Jazz, The Object of My Affection, The Tempest, Beautiful Creatures, Spotlight

***Edgar**:*

How Do You Know, Spotlight, Mad City, Beautiful Creatures, The Curious Case of Benjamin Button

***Addilyn**:*

Evolution, Beautiful Creatures, How Do You Know, Seeking a Friend for the End of the World, The Curious Case of Benjamin Button

***Marlee**:*

Mad City, Spotlight, Blue Like Jazz, How Do You Know, Evolution

***Javier**:*

Evolution, A Scanner Darkly, Spotlight, The Curious Case of Benjamin Button, The Tempest

The strength of the system is that it acts in a collaborative way. The more user feedback, the better the recommendation could be. Besides, the system can be run in real time and provide instant recommendation based on the current user feedback. The weakness of the system can be its high dependence on the user feedback volume. Moreover, the system can hardly capture latent information which is not easy to be observed.

Discussion

We have concluded several main challenges in evaluating the recommendation system and the underlying problems in practice.

1.) **Hard to evaluate the accuracy**
   Most of the approaches used by the recommendation system request for sufficient user ratings on the items to reflect the real feedback. However, there're always very few user ratings especially for new coming items. Thus, it's hard to evaluate the accuracy of our model based on a sparse data set. In practice for example, this will cause tendency to recommending new movies with a few high scores to the users.
2.) **Hard to evaluate the real user satisfaction**
   Since most of the recommendation systems are based on the historic user ratings, while it's hard to collect the real user satisfaction on the recommendation system. This is why many websites nowadays are asking users to rate how satisfied they are to the websites' recommendations. Without evaluating users' satisfaction, it's hard for the recommendation system to improve their algorithm further.
3.) **Hard to evaluate the system's recommendation variety**
   Most of the recommendation systems are evaluated based on the current user-item ratings. However, for any other items beyond that list, it's hard to evaluate the performance of the recommendation systems. This will cause inaccurate recommendations on those totally unseen items, or invariant recommendations without much variety or novelty.