COMP 4601 Assignment 2

Rui Li & Hector Zhang

Design Decisions:

Persistent Storage:

The raw data from archive directory will be migrated to the flat files (.csv) when user initializes the system. Movies, users, and reviews files will be created respectively in order to store the data sets from archive. After k-means is run, the advertisement file will be created to store advertisements. In our case, using flat files is more efficient than using MongoDB. The system initialization running time is significantly improved by this design decision.

User Preference Model:

In our implementation, five user preference variables are added to each user profile. Those five features are based on movie genres, which are Comedy, Drama, Family, Horror, and Others. The reviews will be classified into those five categories.

The average rating score of each genre of a user is used to compute the value of each feature (i.e. Sum of a User's Rating for the Movies of a Genre / Total Number of That Kind of Movies the user rated). To improve the categorizing accuracy, we use several keywords to determine one genre. For instance, the following keywords is used to determine comedy: laugh, funny, comical, and comedy. If a movie can not be categorized to the four genres, it will be treated as others.

Communities Determination Algorithm:

We consider each user profile as a vector that has five dimensions. Each dimension reflects a users' preference on a genre. There are several algorithms can be used for this purpose including Genetic Algorithm and K-Means. In our implementation, we decided to use k-means algorithm for community classification.

Advertisement Categories:

Since there are five genres of movies and ten levels of ranking for each genres. The Advertisement Factory is consisting of the same five genres and the ten relative levels to the genre. The total number of the Advertisements Factory will be the number of genres multiply by the number of relative level. For each contents of every advertisement, we assume it will be particular advertising in real world. In our case, we are using simply information which is representing the relative level to the genre of the movie which is belongs to corresponding cluster. Therefore, it will have fifty advertisements in the Advertisement Factory.

Advertising Mechanism:

Generally, it depends on the result of the K-Mean Algorithm. The idea is based on the given user and page. The Advertisement Factory will be generated in the beginning. We could find the cluster advertisements by cluster which is determined by the given user. It contains five genres of advertisements corresponding to the rating of the user cluster. The given page provides the numbers of the reviews which comes from the users. According to the reviews, we could determine what the genre of the movie is. Therefore, we could combine the clusters' advertisement and the genre of the movie to determine what the final advertisement is.

Results:

- The /reset/{dir} request clears all existing data and models and import data from {dir} to the system.
- The /context request triggers user profile generation process. A result form contains user profile will be return. Meanwhile, k-mean algorithm is triggered so that the the users are classified into several communities based on their feature.
- The /community returns a form that represent this communities.
- The /fetch/{user}/{page} request gets the advertisement recommendation based on the user's community and the page content.
- The /advertising/{category} returns all the advertisement in the given category.

Potential Future Improvement:

In order to get more precise results, the following options can be experimented:

- Use more keywords to categorize movie reviews.
- Use more features when creating user profile.
- Implement advanced semantic analysis algorithm for the advertisement recommendation. TF-IDF is a potential choice.

Suggest Algorithm

Assumptions:

- The new user has rated and written some reviews for some movies.
- The new user doesn't have connections with existing users
- The communities for the existing users have been determined.
- A page contains all the reviews for a specific movie.

Algorithm:

1. Compute the preference profile for the new user. (use the model designed for requirement 3)
2. Compute the community for the new user:
3. Calculate the euclidean distance between the new user and the existing cluster centers.

4. Choose the nearest cluster as the community for the new user.
5. Inside the community, find 5 users that have nearest euclidean distance with the new user. Find all the other users that are connected to the 5 users. Add all those users to a set, called Similar User Set.
6. Add all the movies that are rated by the users in Similar User Set to a set called Candidate Movie Set. Also, add the movies that are rated by the new user to the Candidate Movie Set.
7. To suggest a set of pages (User item based CF):
    a. Create a matrix called UsersRatings. The rows of the matrix are the ratings of the users in the Similar User Set and the new user. The columns of the matrix are the ratings for the movies in the candidate Movie Set. If a user haven't rated a movie before, set that entry to -1.
    b. Apply the item based CF to the matrix to compute the ratings of the movies(The new user haven't rated before) for the new user.
    c. Choose top 3 movies that have higher newly computed rating values.
    d. recommend the three pages of the top 3 movies to the new user
8. The advertising system would work in the same way as before.(i.e. Advertise the content based on which the community the new user belongs to and the page the new user is viewing)

Discussion:

In our suggest algorithm, both feature clustering and social network are considered. The Similar User Set is determined by social network structure and the user’s feature vector. So in theory this model has less bias than only use one of the factors. Item based CF is used on the Similar User Set to recommend the set of advertisements to the new user.