

## Project Report

We designed a personalized movie recommendation system based on ASP program. Just as the name implies, this system can intelligently recommend movies to users based on several factors.

First, we need to build an inheritance hierarchy. The movie is a main class, and it has multiple sub-classes such as action, romance, horror and so on. Meanwhile, user is another main class, and it has two sub-classes: male and female.

Then, we need to build a basic database. In order to reduce computational burden, we only create 6 users which are a1, b1, c1, d1, e1 and f1. Besides, we create 11 movies which are a2, b2, c2, d2, e2, f2, g2, h2, i2, j2 and k2. And we also create another list which is used to store all the reviews of users for the movies they have seen. The content of each review is an integer whose range is 1 to 10, and the higher the value, the higher the user's evaluation of the movie. If the user didn't give the review, then the corresponding value is 0.

The strategy of our algorithm is that the system first selects other users with high similarity to this user, then it checks what movies those similar users have seen. Finally, the system will pick up those reviewed movies that are highly rated by those similar users and then recommend them to this user.

There are three main standards to measure the similarity between users. The first one is the gender of user. If two users have same gender, then the score of this part will be a positive number, otherwise, the value is 0. The second part is the average rating of each user's reviews, and the closer those two averages are, the higher the score of this part will be. The third part is based on the categories of movies users have seen. The score of this part tends to be high if lots of the rated movies are of the same categories. For example, if user A and user B both reviewed many movies that belong to fiction, then their score of this part will be relatively high.

The final point is the sum of these three parts, and if the point is higher than the set threshold, then we can think of these two users as highly similar.

The query format is:

user\_recommendation(X, Q).

X is the user id, which means it is from the set {a1, b1, c1, d1, e1, f1}, and Q is the query results, which store the movie list recommended by the system. Here is the example:

```
?- user_recommendation(a1,Q).
% QUERY:?- user_recommendation(a1,Var0).

ANSWER: 1 (in 678819.667 ms)

BINDINGS:
Var0 = [[a2,e2,k2],[g2,i2,j2,k2]] ? ;
```

Besides, there is another query format:

user\_recommendation(X, [Y], Q).

The second parameter is the partial list of users. In this format, the system will only consider the users in the list. The example is:

```
?- user_recommendation(a1,[c1,d1,e1],Q).  
% QUERY:?- user_recommendation(a1,[c1,d1,e1],Var0).
```

```
ANSWER: 1 (in 280465.027 ms)
```

```
BINDINGS:
```

```
Var0 = [[g2,i2,j2,k2]] ? ;
```

Due to time constraints, the final result has not been further processed. So, there may be duplication of some movies in the list.

ps: The query time may be a little long, which is about 10 to 15 minutes.