A HYBRID GA-BASED COLLABORATIVE FILTERING MODEL FOR ONLINE RECOMMENDERS

Yvonne Ho, Simon Fong and Zhuang Yan
Faculty of Science and Technology, University of Macau

Keywords: Collaborative Filtering, Recommendation Systems, Genetic Algorithm.

Abstract: Online recommenders have been a prevalent tool in e-Business that assists users to find items of their interest. Different algorithms that predict scores based on the heuristic of similarity of other peoples' taste to that of the user have been experimented by many researchers. However most of the current recommendation systems take all users with common items as neighbors in their measurement, which may induce noise and hence inaccurate prediction. In this paper, we attempt to solve this problem by proposing a hybrid model that combines content-based filtering and collaborative filtering for online recommenders. The model exploits merits from these two techniques by selectively encoding both the user profiles and the product information into the same chromosomes in a Genetic Algorithm. Our experiments demonstrated that this new approach gives relatively high accuracy rate in predicting user preferences.

1 INTRODUCTION

Generally, there are two common methods applied in recommendation systems for predicting the users' items of interest, they are *Collaborative Filtering,* and *Content-based Filtering*. *Collaborative Filtering (CF)* method (Breese, Heckerman, and Kadie, 1998) is based on the similarity between currently active user and other users. It can either be measured by the same item which is known as item-based CF or by the same type of user, known as user-based CF. The goal of this method is to suggest new items, to predict the utility of a certain item for a particular user based on his previous likings or the opinions of other like-minded users. In the traditional CF method, the system only searches for users who contain common items with the active user. In this case, only the top popular items will be selected. As a few common items are insufficient to reflect a user's taste, the recommender will not be very accurate.

Content-based Filtering selects items based on the correlation between the content of the items and user preferences. It recommends the items similar to those a given user has liked in the past. There are however some short-comings while applying Content-based Filtering technique on recommenders. Firstly, for text documents, the system can only capture certain aspects of the content, so that only a shallow analysis of certain kinds of content can be supplied. Secondly, the system can only recommend items scoring highly against the user profile, so the user is restricted to see the items similar to those already rated, new items will seldom be recommended. Furthermore, the user's own rating is the only factor influencing the future performance.

A *Hybrid model* combines the *Collaborative Filtering* and *Content Based Method*. However, a mere integration of the two techniques suffers from prohibited oversize of data and a very long processing time. In our proposed model, a genetic algorithm (GA) in such hybrid CF was used in order to give a reasonably quick and optimally accurate prediction, while avoiding the need of large amount of data from both users and items information.

One key feature of this hybrid model powered by GA is feature (and feature weights) selection. This is to eliminate the situation where redundant features were included when making a prediction. They are noises that damper the prediction accuracy of the recommendation during calculation.

In this paper, we described a GA-based (Ujjin and Bentley, 2002) hybrid CF model and experimented this model on an online movie recommendation system for performance evaluation. The model is able to select the appropriate feature weights in the computation.

Ho Y., Fong S. and Yan Z. (2007).
A HYBRID GA-BASED COLLABORATIVE FILTERING MODEL FOR ONLINE RECOMMENDERS.
In *Proceedings of the Second International Conference on e-Business*, pages 200-203
DOI: 10.5220/0002113602000203

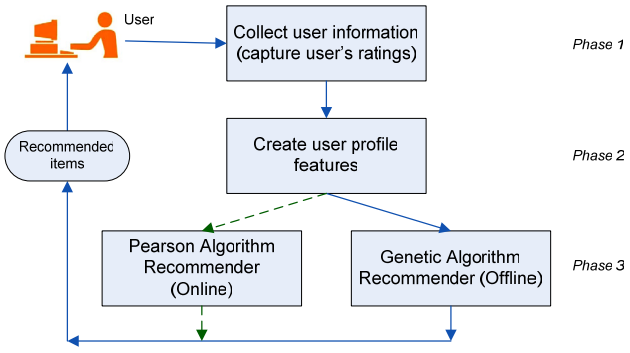

Figure 1: A typical workflow of CF online recommender that includes user's profiles.

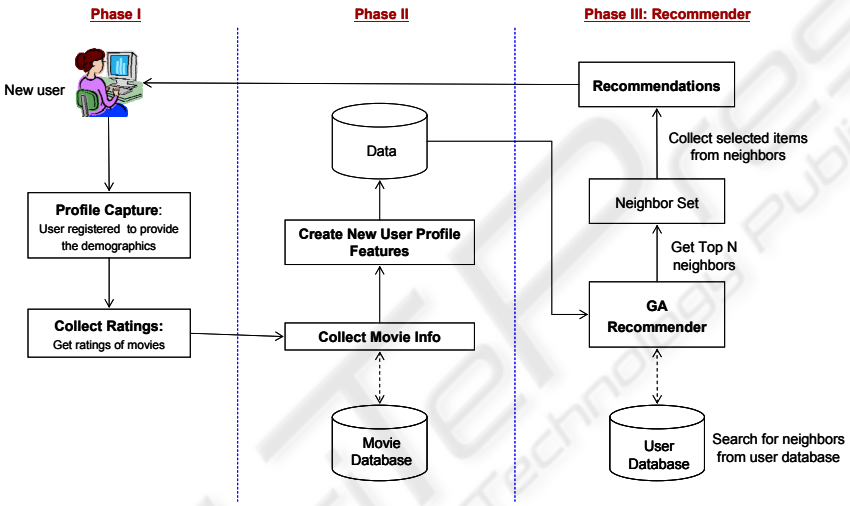

Figure 2: GA-based Recommendation System.

In this example case, we showed that user profile features are most important to the target user in relative to other movie attributes which may turn into noises. Moreover, the similarity measure can collect the neighbor sets with a similar taste; in this way the recommendations will be more adaptable. From the results of our experiment, we show that GA combined with some fine-tuned User Profiles features is a good candidate for recommendation systems.

2 GA RECOMMENDER MODEL

Based on the techniques mentioned in the previous section, we constructed a GA-based Recommender as a prototype model for conducting experiments.

The experimental protocol is capable of gathering, disseminating, and using ratings from some users to predict other users' interest in movies.

A general workflow of an online recommender model is shown in Figure 1. Comparing to GA, Pearson Algorithm is fast though the prediction is not as accurate. An e-Business website may consider using Pearson recommender for instant notification of recommended results. If the user however desires recommended results of higher accuracy, GA which usually takes a few minutes for processing can send the results by email. This process can be done offline, perhaps in batches. Figure 2 shows the architecture of our system that can be deployed as an online application. The process flow is divided into three phases:

- *Phase I: Collect User Information*
- *Phase II: Create User Profile features*
- *Phase III: GA Recommender*

This phase has the following GA related functions that search for the appropriate

recommendation by selecting and weighing the features.

2.1 Feature Selection

The first step in the GA Recommender phase is to prepare the features that are needed. From the experiments we observed that some of the user profile attributes are more effective than the others. So we only select 12 features out of 37: *Rating, Age, Gender, Occupation, Prefer FilmType, Prefer Director, Prefer Actress, Prefer Actor, Prefer Producer, Prefer Writer, Prefer Editor and Prefer Language.*

2.2 Feature Weighing

The structure of the chromosomes is similar to Figure 1. There are totally 12 genes. Each gene represents by a feature weight *w* in real value. The heavier the weight the more important the feature is, so that the value can represent the feature importance to the user. For example, the weight of the feature *Prefer Director* is the highest, that indicates the user favors over his choice movies by certain movie directors.

We programmed in GALib (GALib) to find out the feature weight *w,* distance measure *d* and obtain a group of neighbor set for the active user by choosing half of the top scores from the users of similar taste.

2.3 Recommendations

After we obtained the neighbor set, we can provide the active user a list of recommendations by summarizing the neighbor's rated movies which have not been rated by the active user. Also we can predict the votes of those movies, that helps guiding the user to choose his favorites by ranking the votes. The vote for movie *i* for active user *a* can be predicted by:

$$predict\_vote(a,i) = \bar{v}_a + k\sum_{j=1}^{n} euclidean(a,j)(v_{j,i} - \bar{v}_j)$$

As we can also measure the predict vote for those movies that the active user rated before, we can compare it with the actual vote to cross-check the fitness rate for user *a* on movie *i*:

$$fitness(a,i) = \left| \frac{predict\_vote(a,i) \times 100\%}{actual\_vote(a,i)} \right|$$

3 EXPERIMENTS

In order to evaluate the effectiveness of our work, we would compare the traditional Collaborative Filtering method using Pearson Coefficient and our proposed schemes by using Genetic Algorithm. We repeated the experiments by using different features as to show how the feature weights affect the fitness accuracy and performance.

3.1 Experiment I - Features Weights

We calculated the weights for each feature, and took ten users to test the importance of features to each user. We show the average weights for each feature on different users. The average feature weights is about 0.2649.

The separation of two groups of line is evident that the weights of user profile features are much higher than other movie attributes. Especially the features *prefer director, prefer actress, prefer actor, prefer producer, prefer writer, prefer editor,* all of their feature weights are over 0.3.

By this observation, we assume that *rating, age, gender, occupation, prefer film type, prefer director, prefer actress, prefer actor, prefer producer, prefer writer, prefer editor* are more relevant to the user preference.

3.2 Experiment 2 - Fitness Accuracy

For testing the fitness accuracy of Pearson Algorithm and Genetic Algorithm, and also the effectiveness of different features on GA, we configured a variety of six components in this experiment and applied them on 50 fixed users

Table 1: Compositions of chromosome types.

Chromosome Type	Chromosome Features			
	1 movie rating and 3 user's particulars	8 user profile features	18 movie genres	7 movie characters
Pearson Algorithm (Pearson)	✓		✓	✓
Genetic Algorithm with 22 features (GA)	✓		✓	
Genetic Algorithm with 29 features (GA Merge)	✓		✓	✓
Genetic Algorithm with 37 features (GA User Profile 37)	✓	✓	✓	✓
Genetic Algorithm with 12 features (GA User Profile 12)	✓	✓		

respectively. The chromosome compositions of the five components in terms of features are shown in Table 1.

The last component is constructed by preprocessing the user profile data pertaining to the movie attributes, that has a set of 7 unique features.

We can observe apparently that prediction of GA is much higher than that of Pearson. The average fitness of Pearson is about 67.9%, whereas the average fitness for 'GA' with various combinations of features range from 83.51% to 83.93%. In particular, the fitness of 'GA UserPro12' is the highest, about 23.61% better than that of Pearson.

The experiment shows the process time for running GA on different features. 'GA UserPro7' that is GA with 7 features out performs the other 4 in terms of speed. Its average process time is 19 seconds. This is a 67.53% reduction over 'GA'.

From the experiment, the longest time taken is by 'GA UserPro37' and the shortest time is 'GA UserPro7'. This reinforces the belief that when more features are into the recommender a longer processing time it takes.

3.3 Experiment 3 - Neighbor Set

For a particular user, we tested the performance on different group sizes of neighbor set, from 10 to 100 respectively.

3.3.1 Process Time vs. Neighbor Set

This experiment shows the performance of different features running on GA with different neighbor sets. At the beginning, their performances are close. As the neighbor set size increases, the process time for 'GA', 'GA Merge', 'GA UserPro37' increase quite sharply. The additional features they have in common are the 18 movie genres.

As we can see, 'GA UserPro37' for 37 features, the process time increases gradually as the neighbor set expands; where for 'GA UserPro7' with 7 features, the process time increases slowly.

3.3.2 Fitness vs. Neighbor Set

In the experiment, the fitness of different features rise up gradually as the neighbor set enlarges. Interestingly when the neighbor set reaches over the size of 35, the fitness continues to stay constant. As indicated by the dotted line in chart, the fitness approaches 95% at the turning point. Further increase on the neighbor set size has no effect on it.

4 CONCLUSION

Our proposed GA-based hybrid CF model combines both correlation analysis of active user to users, and contents of the items rated by peer users. The information of user profiles and item attributes are encoded accordingly into GA chromosomes. As our experiments show, this GA model offers more accurate recommendation than that of Pearson Algorithm. By applying user profile features that are more valuable than other features such as movie genres on GA, the similarity measure finds the neighbors with similar taste to the user; as a result, the user preference can be better predicted. And when user profile features are used alone, the process speeds up. In essence, this hybrid approach exploits merits from CF techniques by selectively encoding both the user profiles and the product information into the same chromosomes in a Genetic Algorithm. Our experiment also shows that the GA fitness keeps constant when neighbor set size increases. This implies some positive elements in the scalability and speed issues of GA online recommendation systems.

REFERENCES

Breese, J., Heckerman, D., and Kadie, C., 1998.Empirical Analysis of Predictive Algorithms for Collaborative Filtering. Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence.

Ujjin, S., Bentley, P., 2002. Evolving Good Recommendations. Genetic and Evolutionary Computation Conference (GECCO).

GALib, http://lancet.mit.edu/ga/