Recommender System Using Collaborative Filtering

COMP9417, Assignment 2 Report

Jingwen Tian (z5089825) Fangtong Liu (z5089847) Xinhong Wu (z5089853)

1 Introduction

Recommender system is widely applied on variety kinds of websites and software over the past decade. It has become an important part of information filtering system. A good recommendation can attract more and more users and produce huge business values. There are several ways to do recommendation, and collaborative filtering algorithm is one of them to recommend by predict the rating of a user given an item. Netflix has held a competition to improve collaborative filtering algorithm on a dataset of movies.

In this project, we implement a movie recommender system. The benchmark of our system is older MovieLens[1] 100k dataset. First we predict the rating of a user given a movie by using a naïve item-item collaborative filtering algorithm. Then improve this model by adding a baseline predictor. A content-based model by using features to predict the preference of users is also implemented. And we have evaluated relevant methods and done comparisons for them.

2 Implementation

2.1 Baseline predictor

There is a simple algorithm to predict movie rating by computing average rating of each movies and average rating of each user.

According to an essay, it says, the mean value of Netflix Prize's training data set is 3.6. This result can reflect the mean level of all movies ratings but nothing, it is not a good predict value definitely.

Let's say, if we have the mean rating of movie i, we can infer that this is a popular movie because we have a high mean rating of that movie, and it is an unwelcome movie if we get a lower mean rating. It's the same if we have a high mean value of user j, then we can

[1] https://grouplens.org/datasets/movielens/

infer that this user prefer giving a high rating, otherwise this user prefer giving a low rating.

We combine the optimal user bias and the optimal item bias according to these three mean values. The following equation can compute the baseline of the prediction:

$$\textbf{baseline} = \textbf{mean}(\textbf{rating\_of\_cur\_movie}) + \textbf{mean}(\textbf{rating\_of\_cur\_user}) - \textbf{mean}(\textbf{all\_rating})$$

We use this baseline predictor to predict ratings for dataset, and we got 0.96+ of RMSE values for these test datasets.

2.2 Collaborative filtering with baseline: Rating prediction

User-user and Item-item are two models to implement the collaborative filtering algorithm. Item-item model was chosen in our system after comparing with user-user. The main idea of Item-item collaborative filtering is: finding a set N other movies whose ratings are most "similar" to movie x's ratings, estimate x's ratings based on the ratings of movies in N.

There are three methods to get "similarity": Jaccard similarity measure, Cosine similarity measure and Person correlation coefficient.

We choose Cosine similarity measure to solve similarity problem. If we have two rating lists of movie x and movie y, then the similarity of these two lists is calculated by the following equation:

$$\text{sim}(x, y) = \cos(r_x, r_y) = < r_x, r_y > / ||r_x|| \cdot ||r_y||$$

Where r_x is the vector of movie x's ratings, $\boldsymbol{r}_y$ is the vector of movie y's ratings.

The predicted value of user i for movie s is:

$$r_{xi} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{xj}}{\sum_{j \in N(i;x)} s_{ij}}$$

where s_{ij} is the similarity of movie i and j, r_{xj} is the rating of user u on item j, N is the set of top-k most similar movies to x who have been rated by user s, and N(i;x) is the set items rated by x similar to i.

We have a naïve collaborative filtering model. To get better results, we will combine baseline and item-item collaborative filtering.

$$r_{xi} = b_{xi} + \frac{\sum_{j \in N(i;x)} s_{ij} \cdot (r_{xj} - b_{xj})}{\sum_{j \in N(i;x)} s_{ij} + b_i}$$

$$b_{xi} = \mu + b_x + b_i$$

where μ = overall mean movie rating, $b_x = mean\ rating\ of\ user\ x$, b_i = mean rating of movie i

We use the above formula to predict the ratings. And cut the prediction which is larger than 5 into 5, and make the prediction which is less than 1 equals to 1.

2.3 Content-based: Preference prediction

The main idea of Content-based approach is analyze the features of movies to identify which movies that user would prefer. To implement this model, get the features of each movie is the first thing need to be done.

2.3.1 Feature capture

The following features of movies are selected by analyzing daily behaviors of peoples.

Decade *and* ***genre***: different people with different age like different decades of movies. This feature can be obtained from "u.item" database.

Directors, writers, actors: these three features obtained from given URL of each movie. A web crawler has been achieved to capture these features.

2.3.2 Algorithm design

Given a user, recommend the user a list of movies he/she by analyzing features.

Step 1: Choose the movies which have 5 ratings for a specific user.

Step 2: Get the features of these high rating movies, and then find the high similarity movies with it. If a movie has the same years with one movie which with 5 rating, a similarity count for that movie will add 1, the other features are the same.

Step 3: Then we choose movies that similarity counts larger than 4, which means the movies we choose have at least 4 same features with movie j, put them into a high similarity movie list.

Step 4: Intersect these movie lists, choose movies that has high similarity with 2 more 5 rating movies. Then we get the results.

3 Experiments

3.1 Benchmark and Evaluation criterion

Benchmark

The benchmark in this project is movieLens 100k dataset [1], it is an older dataset contains 100,000 ratings from 1000 users on 1700 movies released on 4/1998.

Two groups of training and test data was provided to do testing in this dataset. The first group contains 5 sets of data (u1, u2, u3, u4, u5) are 80%/20% splits of the u data. And the second group contains 2 sets of data (ua, ub) with 10 ratings per user in the test set.

Evaluation criterion

For most of the rating prediction based recommender system, root mean square error(RMSE) is selected as the evaluation criterion.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

During the experiments, we calculate RMSE for each set of data, and get the mean of the RMSE as the accuracy for each group.

3.2 Parameters Selection

To get the best performance of our recommender system, the number of movies which are most similar to the current predict movie needs to be determined. We repeat evaluating all the two groups of test datasets by changing the value k from 10 to 30.

The program and running results has been attached. The results plotted as follows:

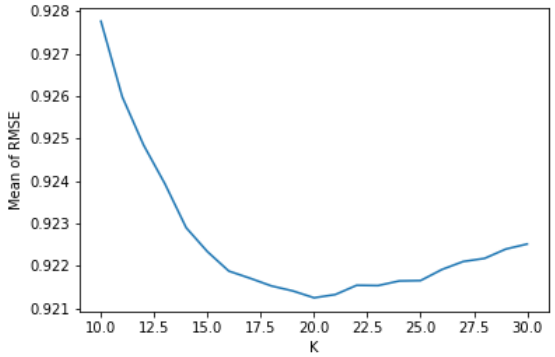

Figure 3.2.1 The mean of RMSE of test data u1 to u5 with different k value

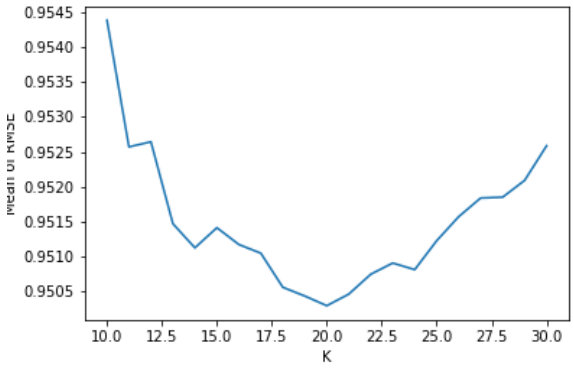

Figure 3.2.2 The mean of RMSE of test data ua and ub with different k value

From the output results and above figures, we can find that the value of k has a great impact on the evaluation results. k = 20 performs best for all the dataset.

3.3 Recommender system implement

3.3.1 Item-item collaborative filtering with baseline

Example: the following is the top 10 recommend movies for user 5

Movie ID	Movie Rate (predict)	Movie Name
1598	5	City of Industry (1997)
1200	5	Kim (1950)
1499	4.993	Grosse Fatigue (1994)
1121	4.993	Umbrellas of Cherbourg, The (Parapluies de Cherbourg, Les) (1964)
1448	4.866	My Favorite Season (1993)
1466	4.865	Margaret's Museum (1995)
1652	4.83	Temptress Moon (Feng Yue) (1996)
849	4.776	Days of Thunder (1990)
1188	4.618	Young Guns II (1990)
1366	4.585	JLG/JLG - autoportrait de décembre (1994)

Figure 3.3.1 top 10 CF recommend movies for user 5

3.3.2 Content based recommender system

Example: the following is the top 10 recommend movies for user 5

```
Movies recommend to user (content-based) :
****************************
Get Shorty (1995)
Mrs. Doubtfire (1993)
Jack (1996)
Aladdin and the King of Thieves (1996)
Fathers' Day (1997)
Fugitive, The (1993)
Batman Returns (1992)
Under Siege (1992)
Men in Black (1997)
Blown Away (1994)
****************************
```

Figure 3.3.2 top 10 content-based recommend movies for user 5

4 Evaluation

4.1 Accuracy

With selecting the parameter k = 20, the RMSE of each group of datasets are:

```
************************************************************
Test for k = 20:
    use u1.base and u1.test :   0.930900753459
    use u2.base and u2.test :   0.920188979195
    use u3.base and u3.test :   0.918054056464
    use u4.base and u4.test :   0.919733438571
    use u5.base and u5.test :   0.91740733912
mean of root mean square error: 0.921256913362
************************************************************
```

Figure 4.1.1 the RMSE of dataset u1 to u5 with optimal k =20

```
************************************************************
Test for k = 20:
    use ua.base and ua.test :   0.941722476278
    use ub.base and ub.test :   0.958878490701
mean of root mean square error: 0.950300483489
************************************************************
```

Figure 4.1.2 the RMSE of dataset ua and ub with optimal k =20

From the figures, we can know that the RMSE of our system is around 0.921 while a user has rating enough movies. But if a user haven't rate much movies in our system, the accuracy cannot be that good.

4.2 Effect of baseline

To find how much affect baseline has to collaborative filtering, we evaluate both of the naïve collaborative filtering and the one combine with baseline and get following results.

For naïve collaborative filtering algorithm, the optimal value of k is 14, and the RMSE with optimal k is 0.97492614532.

From section 4.1, we know that with using baseline predictor, the optimal value of k is 20, and the RMSE with optimal k is 0.921253005292.

This means the baseline predictor makes a great improve for collaborative filtering.

4.3 User-user vs. Item-item Collaborative Filtering

We have known that collaborative filtering has two models to implement a recommend system. When doing evaluations, user-user collaborative filtering is also evaluated to compare with item-item when selecting the method of collaborative filtering.

For user-user method, the optimal value of k is 29, and the RMSE with optimal k is 0.944534214632, which is larger than item-item model's.

Thus, item-item is better than user-user collaborative filtering.

5 Conclusions

In this project, we implement two recommend systems: one is implemented by content-based view; the other is implemented by collaborative filtering.

Content-based system recommends movies that satisfied user's interest, it recommend movies to user according to the features of movies with high ratings.

For collaborative filtering system, with lots of experiments and analysis before, it is obvious that Item-Item collaborative filtering is better than User-User one in most test datasets, Item-Item produce a smaller RMSE value, and when we add baseline into the calculation of rating prediction and choose top k similar items, it gets a lower RMSE value. So we use Item-Item collaborative filtering, baseline, and top-k methods to implement collaborative filtering recommend system.

6 Related work

To improve the performance of collaborative filtering recommender system, there are many other algorithms has been proposed. Arkadiusz Paterek has proposed RSVD in 2007, and Yehuda Koren has proposed TimeSVD++ in 2009, etc.

References

Ekstrand, Michael D., John T. Riedl, and Joseph A. Konstan. "Collaborative filtering recommender systems." Foundations and Trends® in Human–Computer Interaction 4.2 (2011): 81-173.