Application of Using Simulated Annealing to Combine Clustering with Collaborative Filtering for Item Recommendation

Zhiming Feng
School of Computer and Electronic Information
Guangxi University
Nanning, China
e-mail: aifeike@163.com

Yidan Su
School of Computer and Electronic Information
Guangxi University
Nanning, China
e-mail: 452473354@qq.com

***Abstract*—Item-item collaborative filtering was widely used in item recommender system because of good recommend effects. However when facing a large amount of items, there would be performance reduction, because of building a very large item comparison dataset in order to find the similar item. K-means cluster had a very good effect in classification and a good performance even though the dataset being processed is very large. But the cold start was a problem to k-means and we must do some extra work to use it in item recommendation. By using the simulated annealing theory to combine the two methods to fixed the problems of the two methods mentioned above and take use of their advantages for better recommendation effect and performance. The experimental results show that, using simulated annealing to combine the clustering and collaborative filtering in item recommendation system can get more stable recommendation results of better quality.**

Keywords-item-item collaborative filtering; k-means clustering; simulated annealing; recommender system; recommendation algorithm

I. INTRODUCTION

There have been many extra methods which were introduced to improve collaborative filtering(CF) for item recommendation in recommender system. Some of these like using PCA to the resulting dense subset of the ratings matrix to improve the efficiency of the collaborative filtering[1], using improved regularized singular value decomposition to improve the prediction of collaborative filtering[2], using factorization to find neighborhood to improve collaborative filtering[3], using Restricted Boltzmann Machines for collaborative filtering[4] and so on. K-means clustering were also introduced to help collaborative filtering for item recommendation in recommender system. Such methods like using k-means clustering to group the similar users into same cluster to improve the accuracy of collaborative filtering[5].

K-means was popular classification algorithm and item-item collaborative filtering is widely used in recommendations. They had a good effort when used to discover the similar items, however, when facing the large scale data, they had some problems like the procedure of calculation was slow and so on. If combining the two methods together to compensate for each other`s shortcomings, we would get a better result when use them for item recommendation. So we introduce the simulated annealing algorithm to take the combination job.

II. INTRODUCTION OF USED THEORY

A. K-means Clustering

K-means clustering is a unsupervised classification algorithm.For a observation set (x_1 , x_2 ,..., x_n),each observation object is a n-dimensional vector.K-means cluster aims at divided the n observations into k(k<n) classifications $S = \{S_1, S_2, ..., S_k\}$ and reduce the sum of the square of the distance between the clusters:

$$\min_S \sum_{i=1}^{k} \sum_{x_j \in S_i} \left\| x_j - \mu_i \right\|^2 . \quad (1)$$

, where μ_i is the mean of points in S_i .

There are k classification set,Given an initial set of k means $m_1^{(1)},...m_k^{(1)}$,the algorithm proceeds by alternating between two steps[1]:(1)Assignment step: Assign each observation to the cluster with the closest mean.

$$S_i^{(t)} = \{x_p : \left\| x_p - m_i^{(t)} \right\| \le \left\| x_p - m_j^{(t)} \right\| \forall 1 \le j \le k\} . \quad (2)$$

,where each x_p goes into exactly one $S_i^{(t)}$, even if it could go in two of them.(2)Update step: Calculate the new means to be the centroid of the observations in the cluster.

$$m_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{x_j \in S_i^{(t)}} x_j . \quad (3)$$

The algorithm is deemed to have converged when the assignments no longer change.

B. Item-Item Collaborative Filtering

Item-item collaborative filtering use the item comparison data set to predict the items which the user preferred potentially.Item-item use the item comparison data set to calculate the similarity,and use the similarity as the weight to calculate the weighed scores of the items.In this article, we

Published by Atlantis Press, Paris, France.

use the Pearson correlation coefficient to calculate the similarity.we calculate the weighted rating in this way :

$$r_{u,i} = k\sum_{i' \in I} sim(i,i')r_{u,i'} \,. \tag{4}$$

, $sim(i,i')$ is the similarity between two item, $r_{u,i}$ is the rating of item i which marked by user u, $k = 1/\sum_{i' \in I}|sim(i,i')|$,I is the set of the items.

C. Pearson Correlation Coefficient

In order to make the program for convenience, we use the formula of the sample Pearson correlation coefficient to calculate the similarity:

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}} \,. \tag{5}$$

, x_i and y_i is a dimension of the n-dimension vector x_n and y_n .we will use the formula above to calculate the similarity of the k-means and item-item collaborative filtering in the article.

D. Simulated Annealing

Simulated annealing(SA) is local search method that finds its inspiration in the physical annealing process studied in statistical mechanics[6]. An SA algorithm repeats an iterative neighbour generation procedure and follows search directions that improve the objective function value While exploring solution space, the SA method offers the possibility to accept worse neighbour solutions in a controlled manner in order to escape from local minima. More precisely, in each iteration, for a current solution x characterized by an objective function value $f(x)$, a neighbour x' is selected from the neighborhood of x denoted $N(x)$, and defined as the set of all its immediate neighbors. For each move, the objective difference $\Delta = f(x') - f(x)$ is evaluated. For minimisation problems x' replaces x whenever $\Delta \le 0$. Otherwise, x' could also be accepted with a probability $P = e^{(-\Delta)/T}$. The acceptance probability is compared to a number $y_{random} \in [0,1]$ generated randomly and x' is accepted whenever $P > y_{random}$ [7].

III. Recommendation Algorithm Based on Simulated Annealing of Combining Clustering with Collaborative Filtering for Item Recommendation

A. Generate item-item Collaborative Filtering Result

1) Step 1.

from data set, generate a dictionary with format like {U => { I => R } }, U stands for a user of the user set and I stands for the item from the item set for recommendation to the users. R is the rating rated by user U to the item I.

2) Step 2.

use the dictionary generated in step 1 to form the comparison item set: $\{I => \{I' => sim(i,i')\}\}$, $sim(i,i')$ is the similarity between the two item I and I'

3) Step 3.

from the comparison item set generated in step 2, generate the recommendation items for each user in the user set. Use equation (4) to define the recommendation sequence. The final output item recommendation sequence from the item-item collaborative filtering we marked it as $(x_1, x_2, ..., x_n)$.

B. From Collaborative Filtering Result Generate K-means Clustering Result

1) Step 1.

use the k-means clustering to divide the item set into k clusters $\{S_1, S_2, ..., S_k\}$.

2) Step 2.

for the item recommendation sequence $(x_1, x_2, ..., x_n)$, traverse the each element x_i in the sequence and find a similar element y_j for it randomly from one cluster S_k . The similarity calculation can refer to equation (4) in 2.2.

3) Step 3.

generate the recommendation items for each user in the user set. The final output item recommendation sequence from the k-means clustering we marked it as $(y_1, y_2, ..., y_n)$.

C. Simulated Annealing Combine Results From Clustering And Collaborative Filtering For item Recommendation

Let the cost function of the item recommendation problem to be *f*, *f* is defined as a payoff function in the item recommendation problem. Payoff function *f* is the bigger, the better.

The acceptance function *P* of the simulated annealing algorithm will be

$$P(f(X), f(Y), T) = \exp(-(f(Y) - f(X))/T) \,. \tag{6}$$

if *f*(X)>*f*(Y).

The algorithm steps as shown in table below:

TABLE I. Simulated Annealing to Combine Clustering and Collaborative Filtering

In:	Recommendation sequence $X = (x_1, x_2, ..., x_n)$ from item-item collaborative filtering. Recommendation sequence $Y = (y_1, y_2, ..., y_n)$ from k-meas clustering. Temperature T, cooling_rate.
Out:	Final recommendation sequence $Z = (z_1, z_2, ..., z_n)$.

```
1   s ← X; f ← f(X);
2   s_best ← s; f_best ← f;
3   while T > ending_temperature do
4     T ← T * cooling_rate;
5     s_new ← perform steps in 3.1 to get the sequence
      Y based on X using k-means clustering;
6     f_new ← f(s_new);
7     if  P(f, f_new, T) > random() then
8         s ← s_new; f ← f_new;
9     end if
10    if  f_new < f_best then
11        s_best ← s_new; f_best ← f_new;
12    end if
13  end while
14  Z ← s_best;
15  return Z;
```

IV. EXPERIMENT

A. Dataset And Experimental Purpose

The dataset we use for the experiment is the MovieLens 100K dataset which is consist of 100000 movie ratings rated by 943 people for 1682 movies.And we split the ratings of each people into 2 sets randomly with the proportion of 8:2.The experiment will process the 80% of the data to produce the results and the 20% data for validation.(we will call the two dataset short for '80%' and '20%' for convenience later)

The experiment environment we run in is on a computer with a AMD CPU Athlon II X2 250 and a 2GB RAM. All the programs were wrote by Python.

This experiment we perform in this paper is to show that using simulated annealing to combine k-means clustering with collaborative filtering for item recommendation can get more stable recommendation results with better quality than using collaborative filtering purely in item recommendation system.

The measurement factors used in this experiment are the average payoff value, the standard deviation, the precision rate, the recall rate and the running time of the algorithm. The average payoff here means the expectation of the sum of the rating and the popularity of the items(movies here).

B. Experiment Procedure

1) select 20 users(marked from 0) from the users of the 80% data set randomly.

2) run the item-item collaborative filtering alone, record the average payoff, run time, standard deviation of the average payoff, recall, precision.

3) run the recommendation algorithm that use simulated annealing(cooling rate 0.94, initial temperature 8000) to combine the item-item collaborative filtering and k-means. Record the average payoff, run time, standard deviation of the average payoff, recall, precision.

4) Use the result from 2) to 3) to draw the result graphics.

C. Experiment Result And Analysis

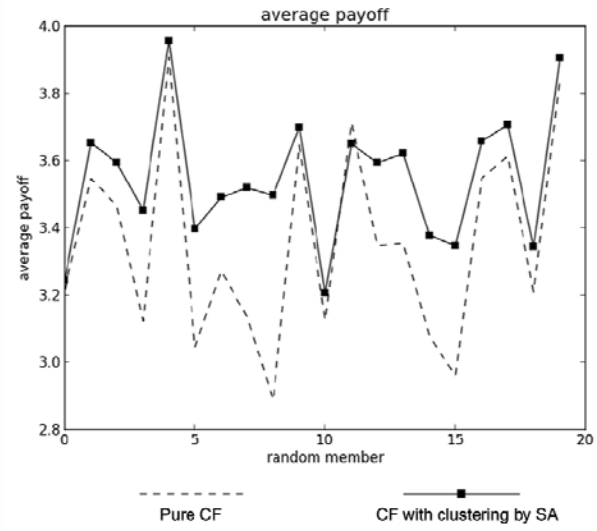

Figure 1. Comparison of the average payoff between pure CF and CF combined with clustering by simulated annealing

From figure 1, we can see that average payoff of the result from using simulated annealing algorithm to combine k-means clustering with collaborative filtering is almost greater than the result from pure collaborative filtering. That means the recommended movies from the improved collaborative filtering method have a higher average rating and higher popularity than the movies recommended by the original pure collaborative filtering. In fact, this is reasonable, because movies with high rating and high popularity will be accept by users more easier. Thus, the movies recommended by collaborative filtering improved by simulated annealing was better than the original collaborative filtering.

Published by Atlantis Press, Paris, France.

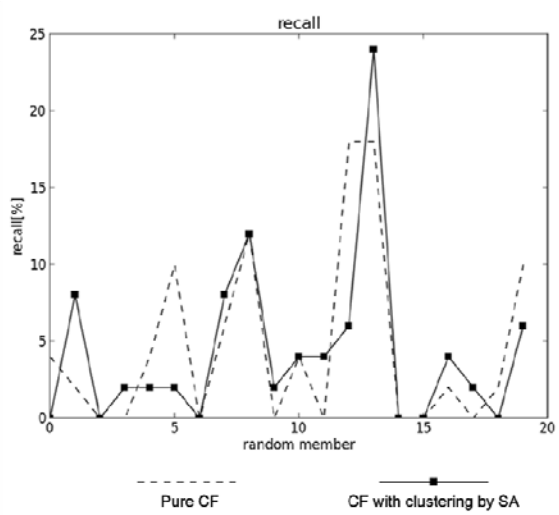

Figure 2. Comparison of the recall rate between pure CF and CF combined with clustering by simulated annealing

Recall rate are defined as

$$recall = tp / (tp + fn) . \quad (7)$$

tp stands for correct result, fn stands for missing result. The recall rate of the result from a recommendation algorithm means the percentage of the correctly recommended results of all correctly recommended results in fact. We can know from figure 2 that almost all of the recommended results from the improved collaborative filtering have a higher precision rate than the original collaborative filtering. That means after comparing the recommended movies in fact with the 20% validation movies, we know that the improved collaborative filtering method can offer a more accurate prediction of the recommended movies, because it guessed more movies which should be recommended to the user.

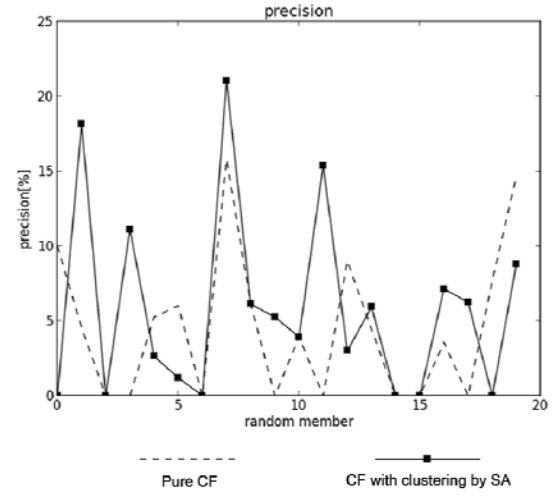

Figure 3. Comparison of the precision rate between pure CF and CF combined with clustering by simulated annealing

Precision rate are defined as

$$precision = tp / (tp + fp) . \quad (8)$$

tp stands for correct result, fp stands for unexpected result. The precision of the result from a recommendation algorithm means the percentage of the correctly recommended results of all recommended results. We can know from figure 3 that almost all of the recommended results from the improved collaborative filtering have a higher precision rate than the original collaborative filtering. That means after comparing the recommended movies in fact with the 20% validation movies, we know that the improved collaborative filtering method can offer a more reasonable prediction of the recommended movies, because it guessed more movies the user have seen in the validation set, so it will guessed more movies the user should see in reality.

V. CONCLUSION

In this article, we use the simulated annealing algorithm to combine the item-item collaborative filtering and k-means cluster for item recommendation. In this way, we can see the recommend quality and performance have a improvement by compare with the original collaborative filtering. In fact, when use the k-means to do the classification job in stead of the collaborative filtering, we can save a lot of time, because use collaborative filtering for recommendation will cost a lot of time to calculate the item-item comparison dataset. So use k-means working with collaborative filtering will have a performance improvement than just use the collaborative filtering for item recommendation.

REFERENCES

[1] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, “Eigentaste: A constant time collaborative filtering algorithm”, Information Retrieval, vol 4, no 2, pp 133–151, 2001.

[2] A. Paterek, “Improving regularized singular value decomposition for collaborative filtering”, in Proceedings of KDD Cup and Workshop, 2007, vol 2007, pp 5–8.

[3] Y. Koren, “Factorization meets the neighborhood: a multifaceted collaborative filtering model”, in Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008, pp 426–434.

[4] R. Salakhutdinov, A. Mnih, and G. Hinton, “Restricted Boltzmann machines for collaborative filtering”, in ACM international conference proceeding series, 2007, vol 227, pp 791–798.

[5] G. R. Xue, C. Lin, Q. Yang, W. S. Xi, H. J. Zeng, Y. Yu, and Z. Chen, “Scalable collaborative filtering using cluster-based smoothing”, in Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, 2005, pp 114–121.

[6] AARTS/KORST, Simulated annealing and boltzmann machines. A stochastic approach to combinatorial optimization and neural computing. John Wiley., 1990.

[7] K. Bouleimen and H. Lecocq, “A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version”, European Journal of Operational Research, vol 149, no 2, pp 268–281, 2003.