1. Movie recommendations

1.1. Memory

We have 69878 users which about 2^16, if it is stored in 69878X69878 matrix using double format. The memory should be at least 2^16*2^16*2^3 byte---2^35 byte = 32 GB, which is beyond the memory limitation 2GB.

Actually we can find that the Persons correlations matrix is a sparse matrix. Because when we calculate the persons correlation between two users, we need find the movies that both users rated first. As a matter of fact the numbers N the movies rated in common mostly is small and even is 0. When N is 0, the Persons Correlation will be set to NaN.

After we know it is a sparse matrix, we will have several methods to store matrix efficiently

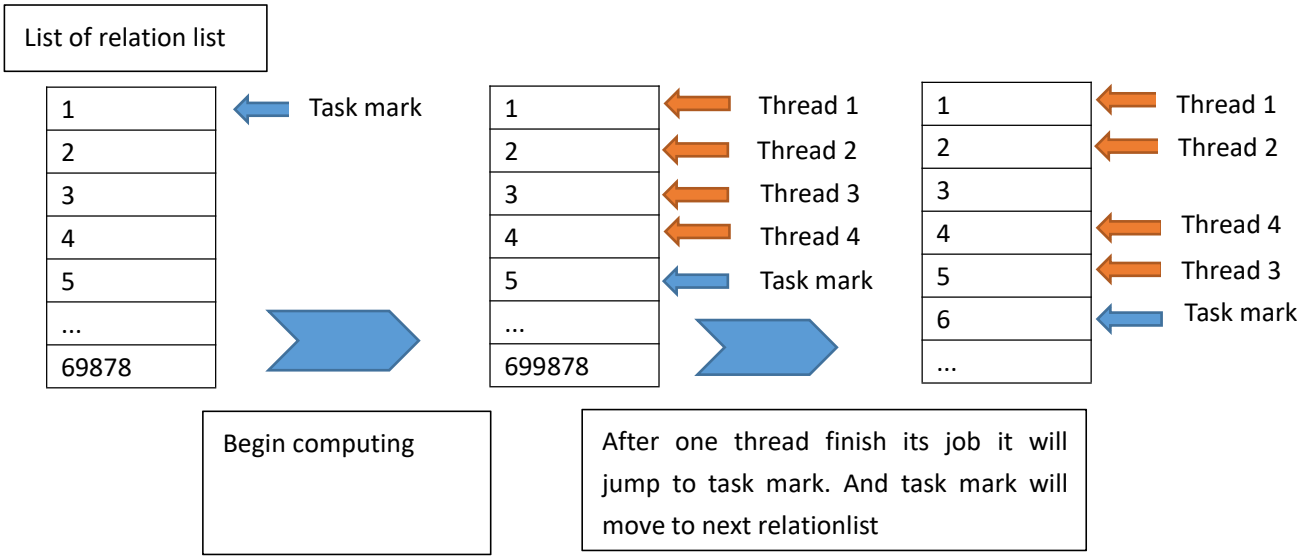
1) DOK: Use list of {row, column, value } to store pairs
2) LIL: Use list of {column index,value} to store a row,and we will have a list of lists

Here we can assume every user have K valuable persons correlations. We have M users in total. If use 1) the memory for the matrix will be about 3*4*K*M byte, if we use 2) the memory will be 2*4*K*M byte. It is obvious that in this problem we should choose 2).

However after using LIL, we still have memory issues. So I choose to sort every list, then choose the first 200 neighbors to solve this problem. When we sort the list we compare the abstract value of Persons correlation.

1.2. Speed: parallel computing

In order to make the whole process of getting matrix faster, I use thread method to do parallel computing. The structure of computing is shown below.



A lock is used to ensure when a thread get task mark and task mark move to next row, the task mark won't be reached by other threads. Details can be seen from the code. After applied parallel computing, the time spent on department machine decreased from **400** minutes to **100** minutes.

1.3. Accuracy

The equation used for prediction:

$$P_{ik} = \overline{R}_i + a \sum_j W_{ij} (R_{jk} - \overline{R}_j)$$

$$a = \frac{1}{\sum |W_{ij}|}$$

W_{ij} = Pearson correlation

$$W_{ij} = \frac{\sum_k (R_{ik} - \overline{R}_i)(R_{jk} - \overline{R}_j)}{[\sum_k (R_{ik} - \overline{R}_i)^2 \sum_k (R_{jk} - \overline{R}_j)^2]^{0.5}}$$

Where P_{ik} is the prediction of user i for movie K, $\overline{R}_i$ is the average rating of user i, R_{jk} is the rating of user j for movie k, $\overline{R}_j$ is the average rating of user j.

We use Root Mean Square Error on test data to evaluate these parameters.

$$RMSE = \sqrt{\frac{1}{n} \sum_k (P_k - R_k)^2}$$

1.3.1. Parameter tunning

First, we need to adjust Pearson correlation based on number of co-rated items.

$$W_{ij} = \begin{cases} W_{ij} & \text{If k > =MinCom} \\ W_{ij} * (k / MinCom) & \text{If k < MinCom} \end{cases}$$

MinCom----Minimal number of commonly rated movies, k is the number of co-rated items. In the whole process, MinCom ranges from 25 to 150.

Second, Maximum number of neighbors to predict ratings----MaxNei. If the number of neighbors can be used to predict ratings is larger than MaxNei, we will only choose top MaxNei neighbors to make prediction. The neighbor is sorted by the absolute value of Persons correlation. MaxNei Ranges from 40 to 200.

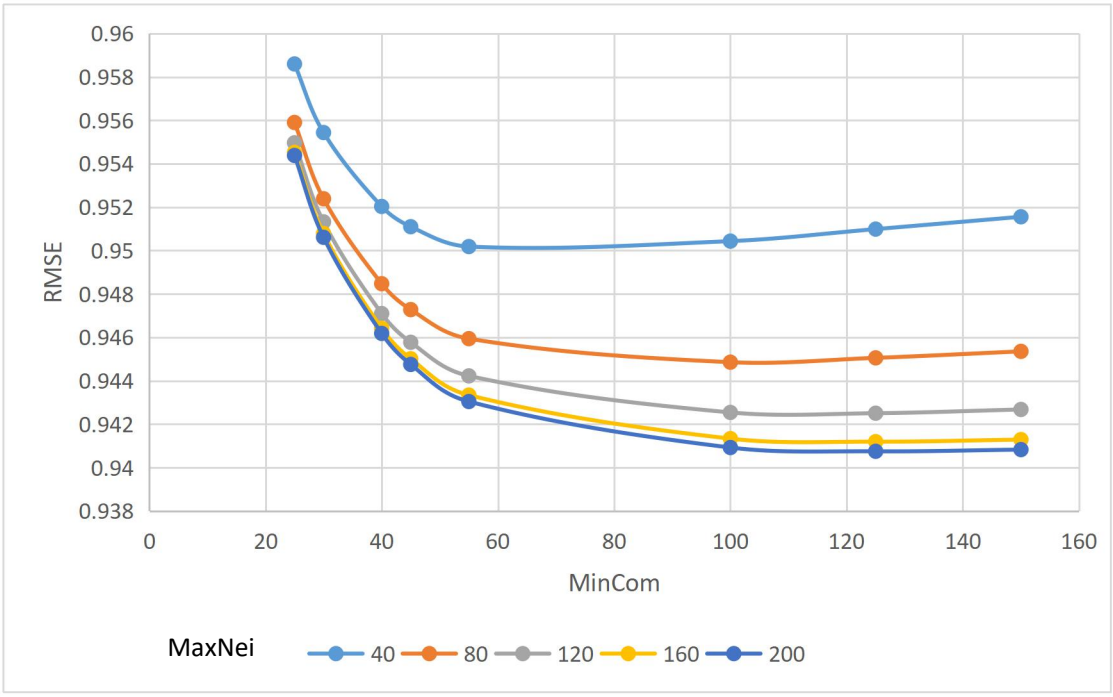


figure 1

From figure 1, we can clearly see the trend of RMSE change,when we change MinCom and MaxNei. And when **MinCom** is **125** and **MaxNei** is **200**, the **RMSE** is smallest which equals **0.94075**.