Project No. 03:
Collaborative Filtering

By: Gu Dazhong, Zhu Yannan, DUCK-HA HWANG

Instructor: Roychowdhury Vwani

EE 239AS Winter 2016

March 3, 2016

I. INTRODUCTION

A. Purpose

In this project, we work with dataset. It is a collection of 943 users, each of whom rates at least 20 movies among total 1682 movies. To predict missing ratings, we use Matrix Factorization Toolbox in MATLAB.

B. Equipment

There is a minimal amount of equipment to be used in this project. The few requirements are listed below:

- Matlab R2015a(8.5.0.197613)
- Computer capable of running the software mentioned

C. Structure

This report will be seperated into 4 parts:

1. Introduction
2. Normal matrix factorization
3. 0-1 matrix factorization
4. Discussion & Conclusion

II. NORMAL MARTIX FACTORIZATION

In this part, we will first create a matrix named R containing user ratings with users on rows and movies on columns. Then we want to run a matrix factorization job to get matrices U and V such that $R_{m*n} \approx U_{m*n}V_{m*n}$ in known data points. UV is our prediction for unknown data. Also, we will create a weight matrix W, which contains 1 in entries where we have known data points and 0 in entries where the data is missing.

A. Problem 1

First, we download the dataset of 100k ratings and create a matrix named R. Then we do a matrix factorization to R and calculate the total least squared error to have a first look about the matrix factorization method. This is how we solve problem 1 step by step:

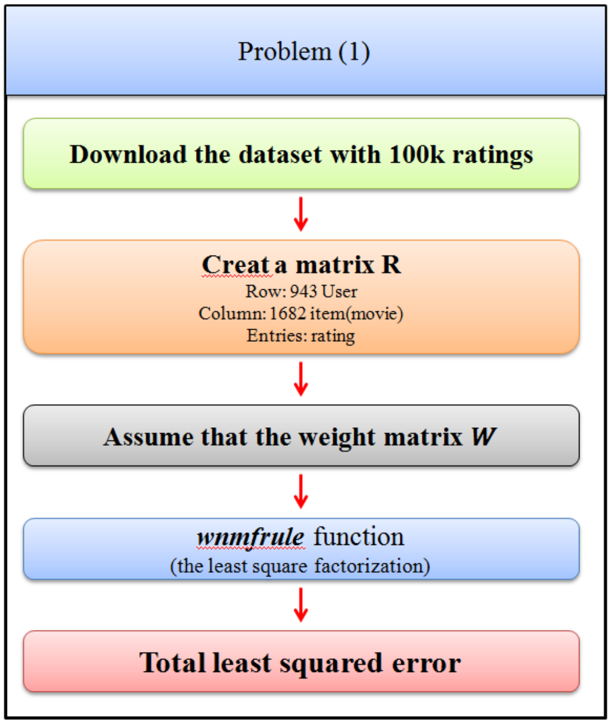
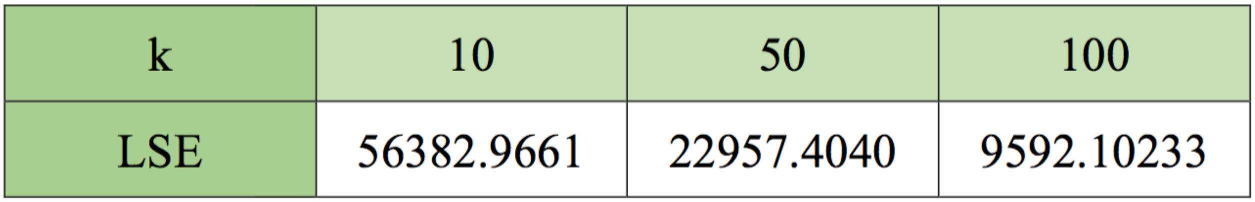

Problem (1)
Download the dataset with 100k ratings
Creat a matrix R
Row: 943 User
Column: 1682 item(movie)
Entries: rating
Assume that the weight matrix W
wnmfrule function
(the least square factorization)
Total least squared error

FIG. 1: Summary of Problem 1

In matlab, the function wnmfrule() can help us factorize matrix. This fuction has a important parameter k. We will set k to 10,50,100 to see the influence of k. The total least square error in each condition is as below:



k	10	50	100
LSE	56382.9661	22957.4040	9592.10233

FIG. 2: Total Leasted Error

B. Problem 2

In this problem, we will use a 10-fold Cross-validation to further examine our suggestion model. We use absolute error as the evaluation standard. This is how we solve problem 2 step by step:

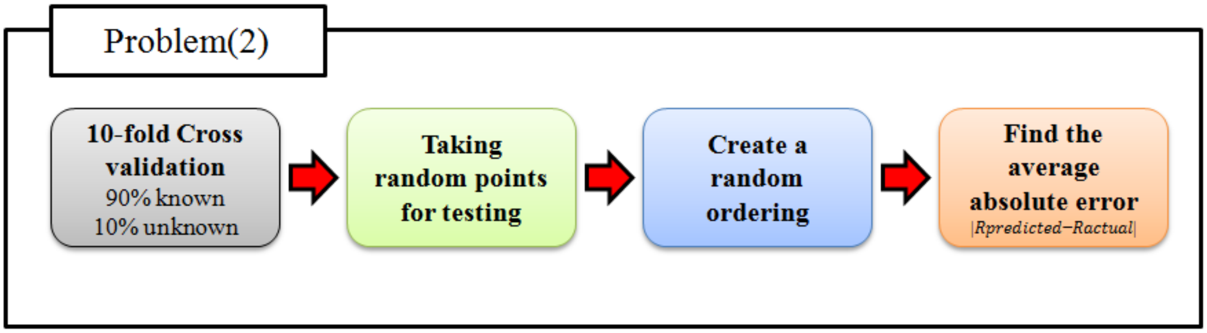
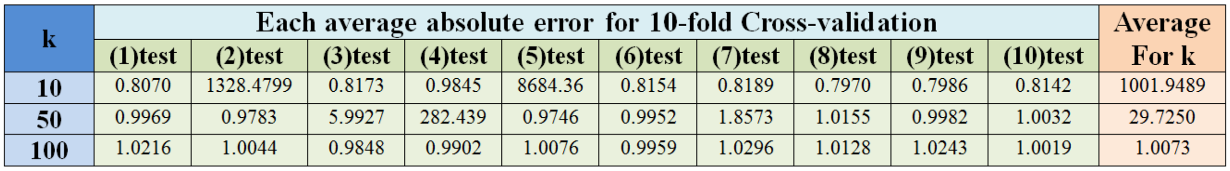


FIG. 3: Summary of Problem 2

The average error, highest error value and lowest error value is shown as below:



k	Each average absolute error for 10-fold Cross-validation										Average For k
	(1)test	(2)test	(3)test	(4)test	(5)test	(6)test	(7)test	(8)test	(9)test	(10)test	
10	0.8070	1328.4799	0.8173	0.9845	8684.36	0.8154	0.8189	0.7970	0.7986	0.8142	1001.9489
50	0.9969	0.9783	5.9927	282.439	0.9746	0.9952	1.8573	1.0155	0.9982	1.0032	29.7250
100	1.0216	1.0044	0.9848	0.9902	1.0076	0.9959	1.0296	1.0128	1.0243	1.0019	1.0073

FIG. 4: Average Absolute Error

1. k = 10
   (a) highest value: 8684.36
   (b) lowest value: 0.7970

    (c) average value: 1001.9489

2. k = 50
    (a) highest value: 282.439
    (b) lowest value: 0.9746
    (c) average value: 29.7250
3. k = 100
    (a) highest value: 1.0296
    (b) lowest value: 0.9848
    (c) average value: 1.0073

C. Problem 3

In this problem, we will divide the movies into two parts: liked by user and not liked by user. For example, if the rating is higher than 3 then it is liked by the user. We will change the threshold rating to examine the trade off between precision and recall. This is how we solve problem 3 step by step:

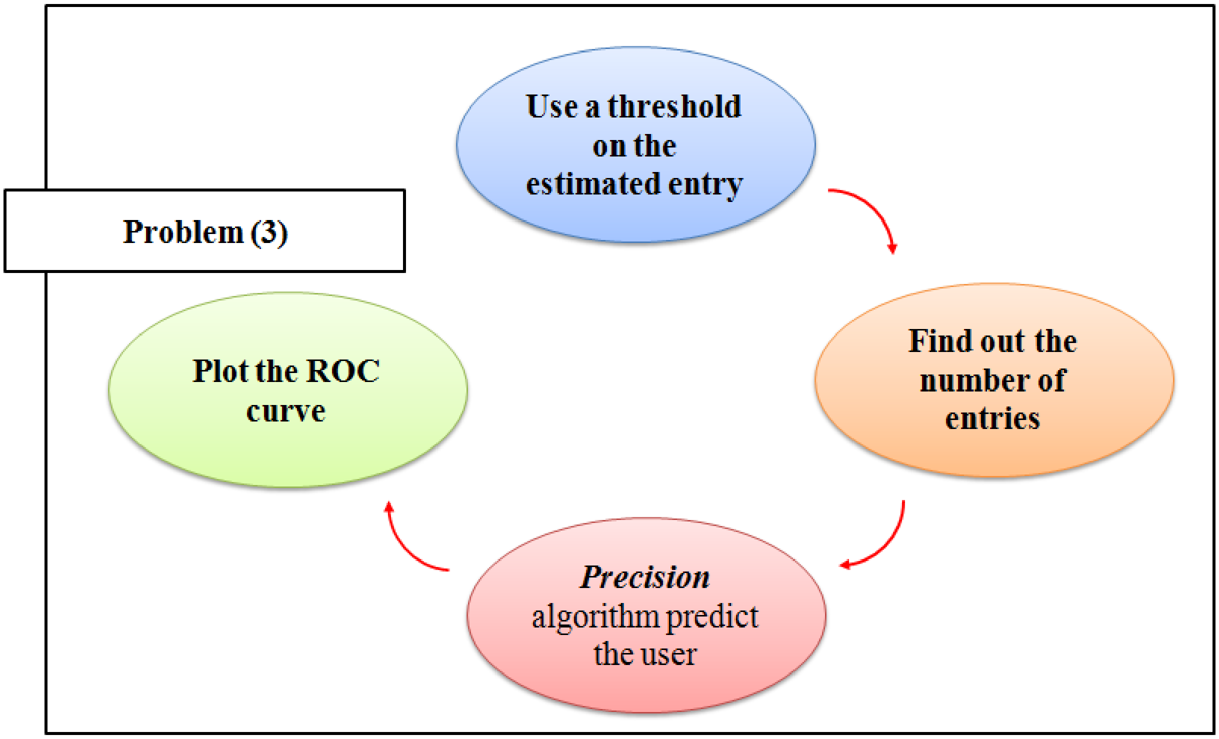
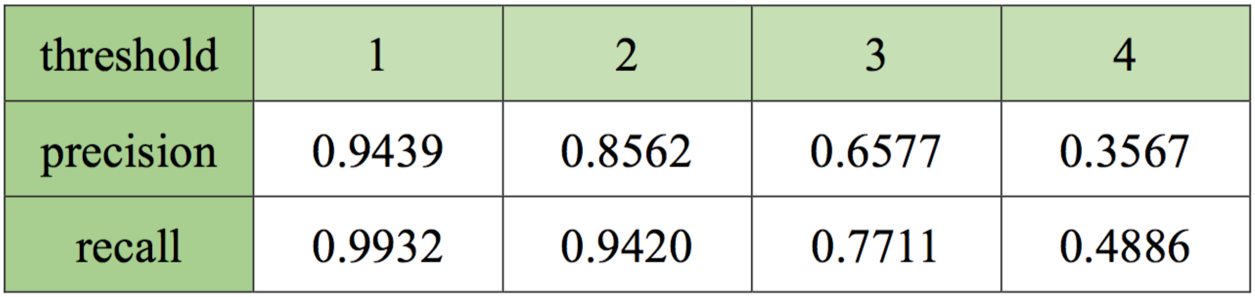

Problem (3)
Use a threshold on the estimated entry
Find out the number of entries
Precision algorithm predict the user
Plot the ROC curve

FIG. 5: Summary of Problem 3

This is the average Precision after 10-fold Cross-validation with different threshold 1,2,3,4. In original data, if a movie is higher than the threshold it is liked by the user. When k = 100:



threshold	1	2	3	4
precision	0.9439	0.8562	0.6577	0.3567
recall	0.9932	0.9420	0.7711	0.4886

FIG. 6: Average Precision after 10-fold Cross-validation with different threshold when k = 100

Now, we set fix the threshold for original data as 3, which means if a movie rating is equal or lower than 3, then it is not liked by the user. If a movie is rated equal or higher to 4, then it is liked by the user. Then we change the threshold for our predicted data to draw the ROC curve. This is ROC curve with different k:

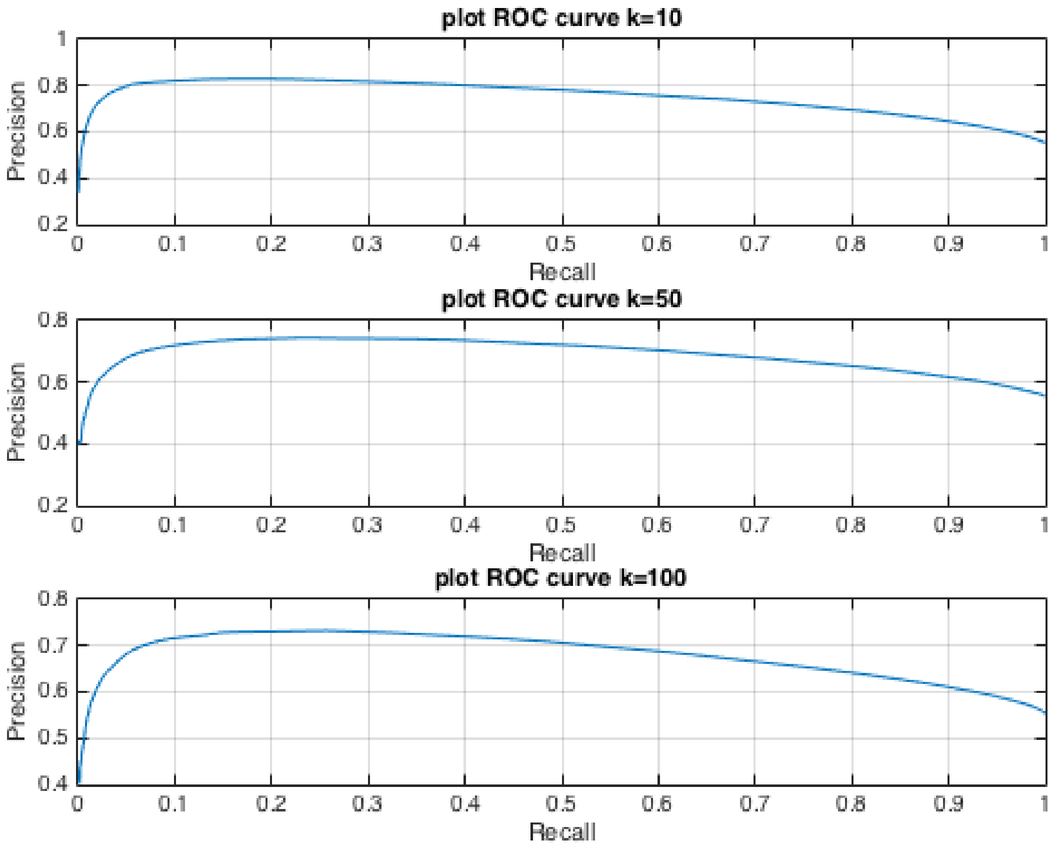
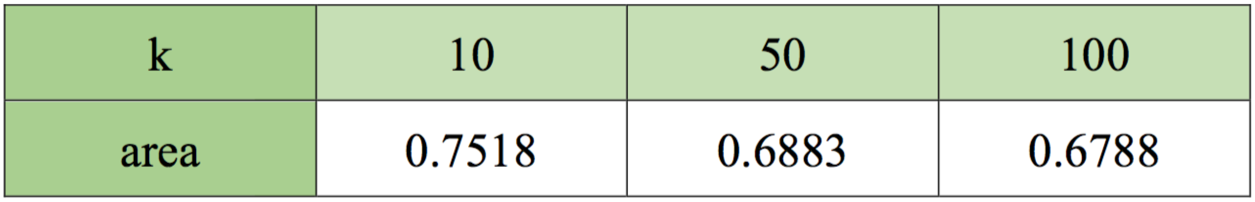


FIG. 7: ROC Curve with Different K

The area under the curve is as below:



k	10	50	100
area	0.7518	0.6883	0.6788

FIG. 8: ROC Curve area

The area is a measure of the performance of our suggestion model. A high area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall relates to a low false positive rate. High score for both show that the classifier (our suggestion model classify the movies into liked by user and not liked by user, so it is a classifier) is returning accurate results (high precision), as well as returning a majority of all positive results (high recall).

III. 0-1 MATRIX FACTORIZATION

In this part, we will still use matrix factorization method to create suggestions. However, in this part, we turn R into a 0-1 matrix, where r_{ij} = 1 for known ratings and r_{ij} = 0 for unknown entries. And change the weight matrix and use rating values as weights, instead of 1, for the known data points.

A. Problem 4

In this part, we change the weight matrix and use rating values as weights, instead of 1, for the known data points. Turn R into a 0-1 matrix where $r_{ij} = 1$ for known ratings and $r_{ij} = 0$ for unknown entries. Then run matrix factorization again with different k. This is how we solve problem 4 step by step:

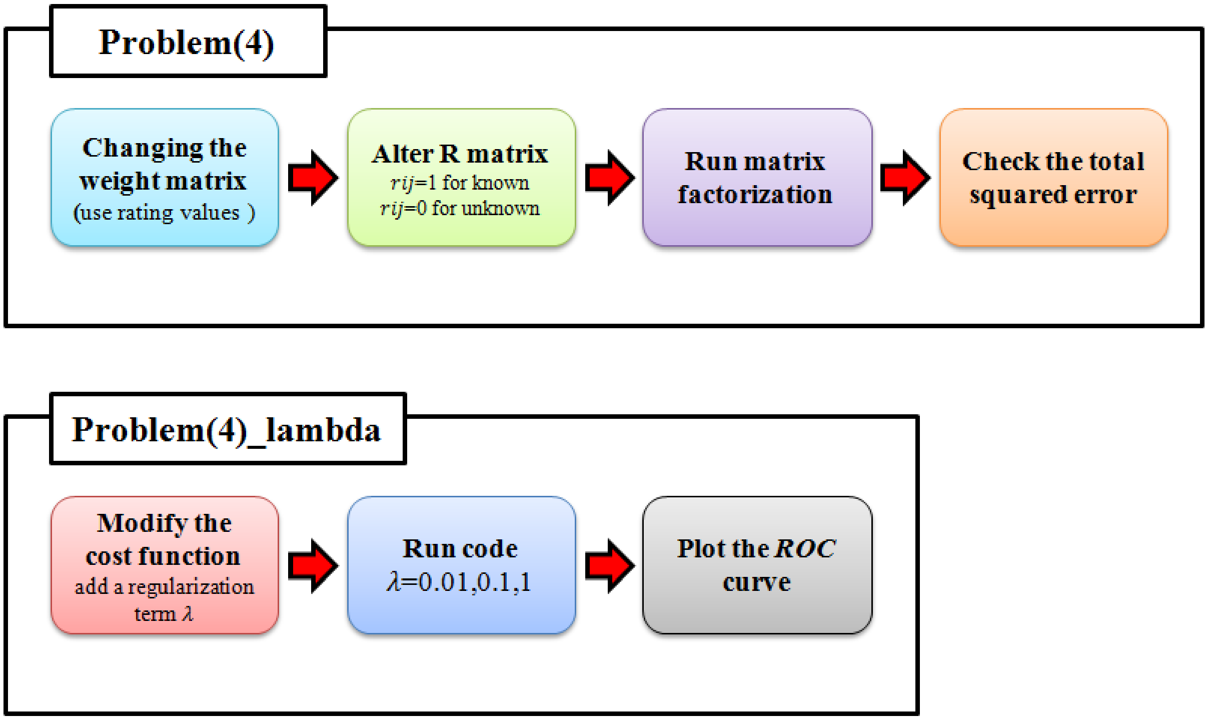
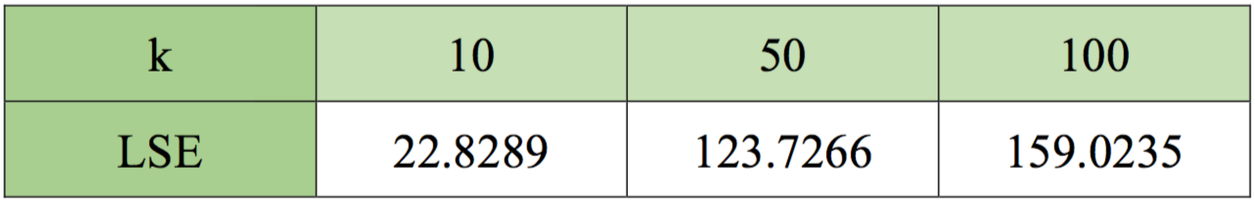

Problem(4)
Changing the weight matrix (use rating values)
Alter R matrix rij=1 for known rij=0 for unknown
Run matrix factorization
Check the total squared error
Problem(4)_lambda
Modify the cost function add a regularization term λ
Run code λ=0.01,0.1,1
Plot the *ROC* curve

FIG. 9: Summary of Problem 4

Then the total leasted squared error becomes:



k	10	50	100
LSE	22.8289	123.7266	159.0235

FIG. 10: Total Leasted Squared Error with Different k

Now, in order to avoid singular solutions, we modify the cost function to add a regularization term λ:

$$\min \sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}(r_{ij}-(UV)_{ij})^2+\lambda(\sum_{i=1}^{m}\sum_{j=1}^{k}u_{ij}^2+\sum_{i=1}^{k}\sum_{j=1}^{n}v_{ij}^2) \tag{1}$$

Using this formula we do the procedure that we did before again. We set rating 3 as the division of movie liked by user and not liked by user. We change the threshold of the predicted value and draw ROC curves (precision-recall) for different k, to see this model works. The ROC curve with respect to different λ is as below:

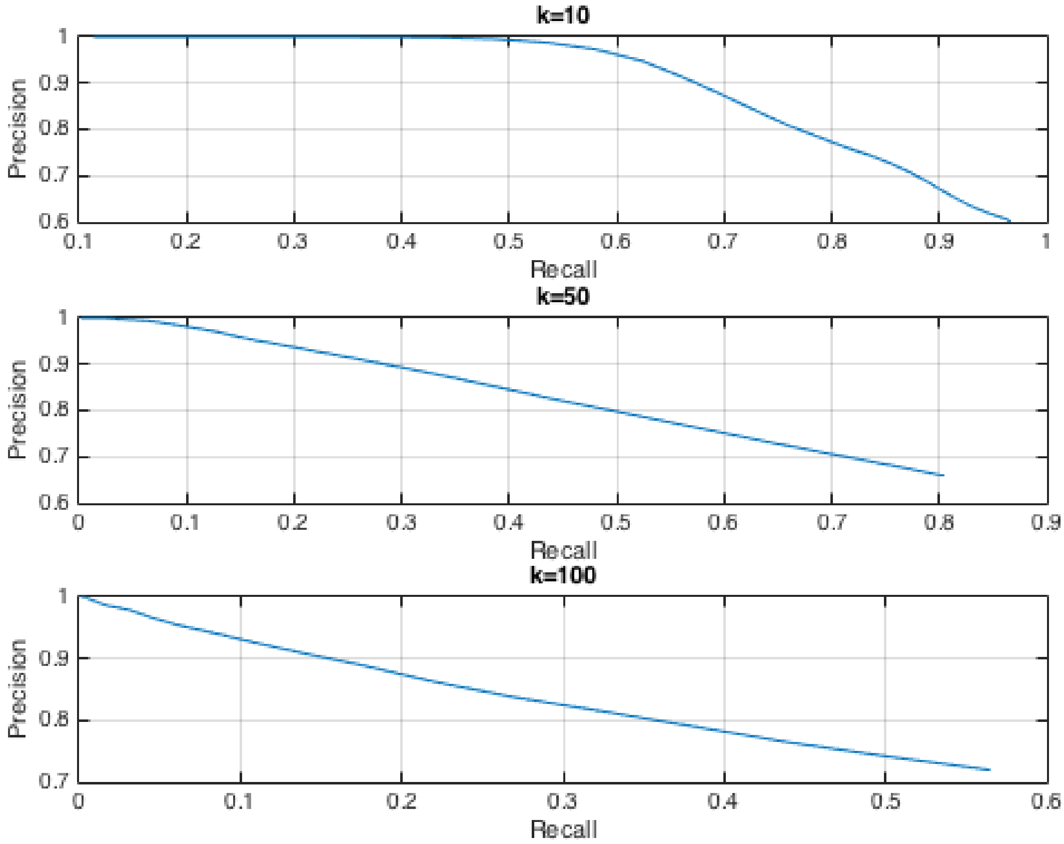


FIG. 11: ROC Curve with $\lambda = 0.01$

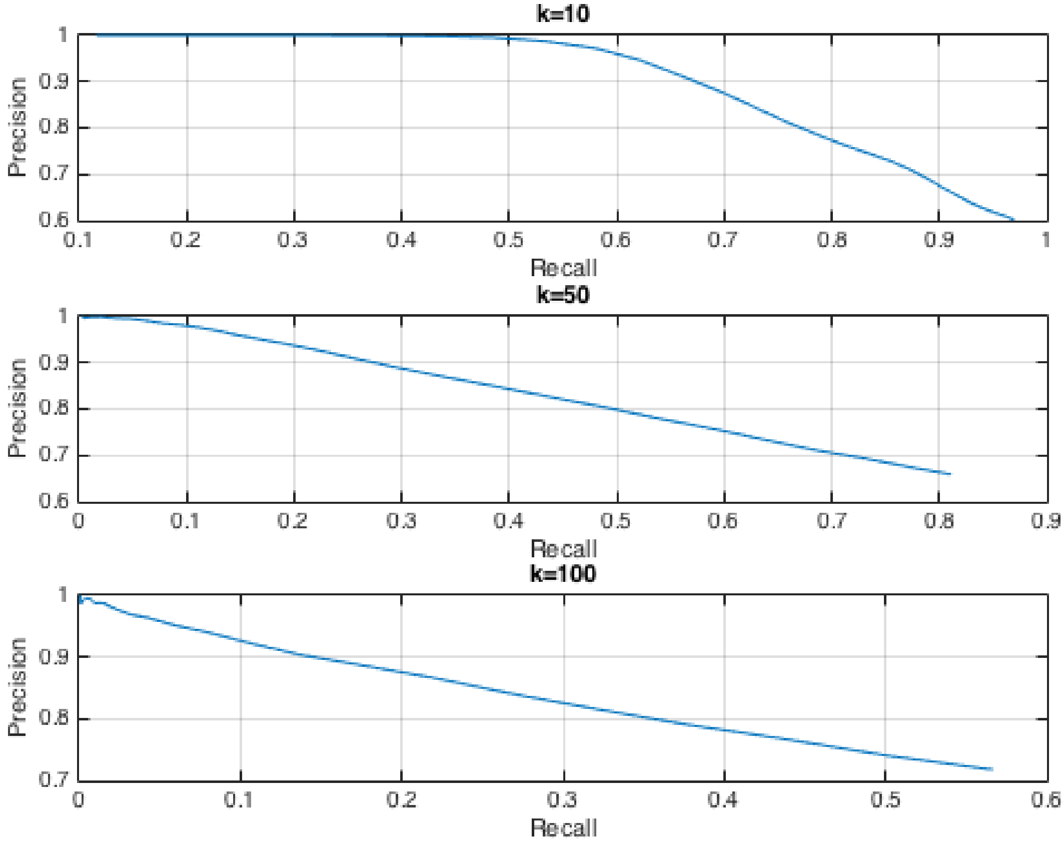


FIG. 12: ROC Curve with $\lambda = 0.1$

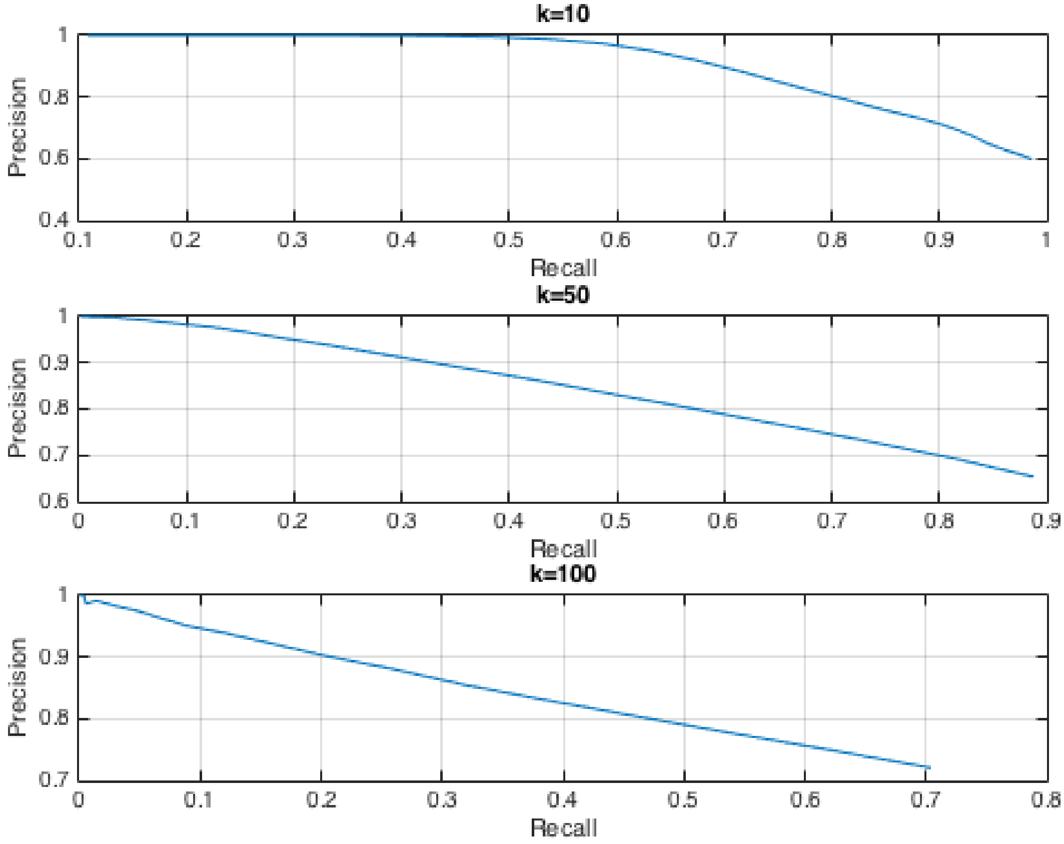


FIG. 13: ROC Curve with $\lambda = 1$

We can see form the pictures above that, when the λ is bigger, the area of the curve becomes larger, which means a better performance. This is because the λ give a regulation to our predict matrix. The regulation can make the predict matrix more stable and closer to the origin matirx.

B. Problem 5

In this part, we will find the top L movies for each users. If there exist the condition that two movies has the same rating, we will let them both be the top L. For example, we want to choose top 5 movies, but there are 6 movies whose rating are all 5. We will see them all as top 5 movies. This will make the precision really high, however make the recall low. This is how we solve problem 5 step by step:

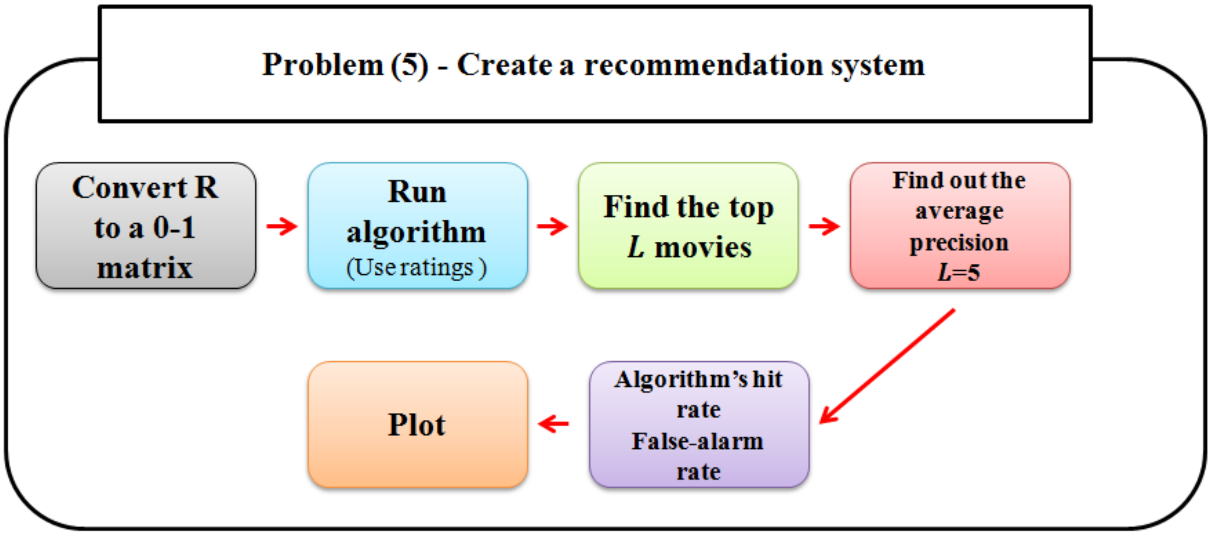
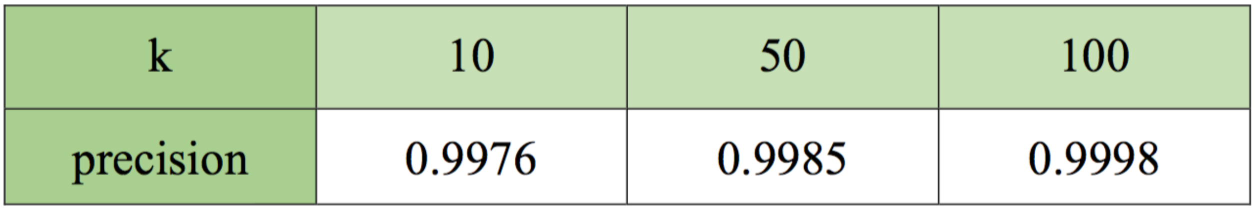

Problem (5) - Create a recommendation system
Convert R to a 0-1 matrix
Run algorithm (Use ratings)
Find the top *L* movies
Find out the average precision *L*=5
Algorithm's hit rate False-alarm rate
Plot

FIG. 14: Summary of Problem 5

First, let us see the average precision when L=5:



k	10	50	100
precision	0.9976	0.9985	0.9998

FIG. 15: Precision when L=5

We can see the precision is very high. This is because of the way we choose top 5 movies. In the dataset, there is only 6 possible rating: 0,1,2,3,4,5. So many movies will have same rating. For example, for a particular user, he may give 10 movies 5 score. So how to decide the top 5 movie is problem. Different people will have different methods. As to us, we just consider all the 10 movies belongs to top 5. However in the predict matrix, the data can be float data so the chance that two movie have same score is very low. So in the predict matrix, for most user we only pick 5 movies for top 5. As result of this picking method, there are more top 5 movies in the origin matrix than the predicted matrix. This causes that the precision is really high.

Next, we will calculate the hit rate (what fraction of the test movies liked by the users are suggested by your system) and false-alarm rate (what fraction of the test movies not actually liked by the user, are suggested by your algorithm) and use these two rate to draw ROC curve. The ROC curve for different L is as below:

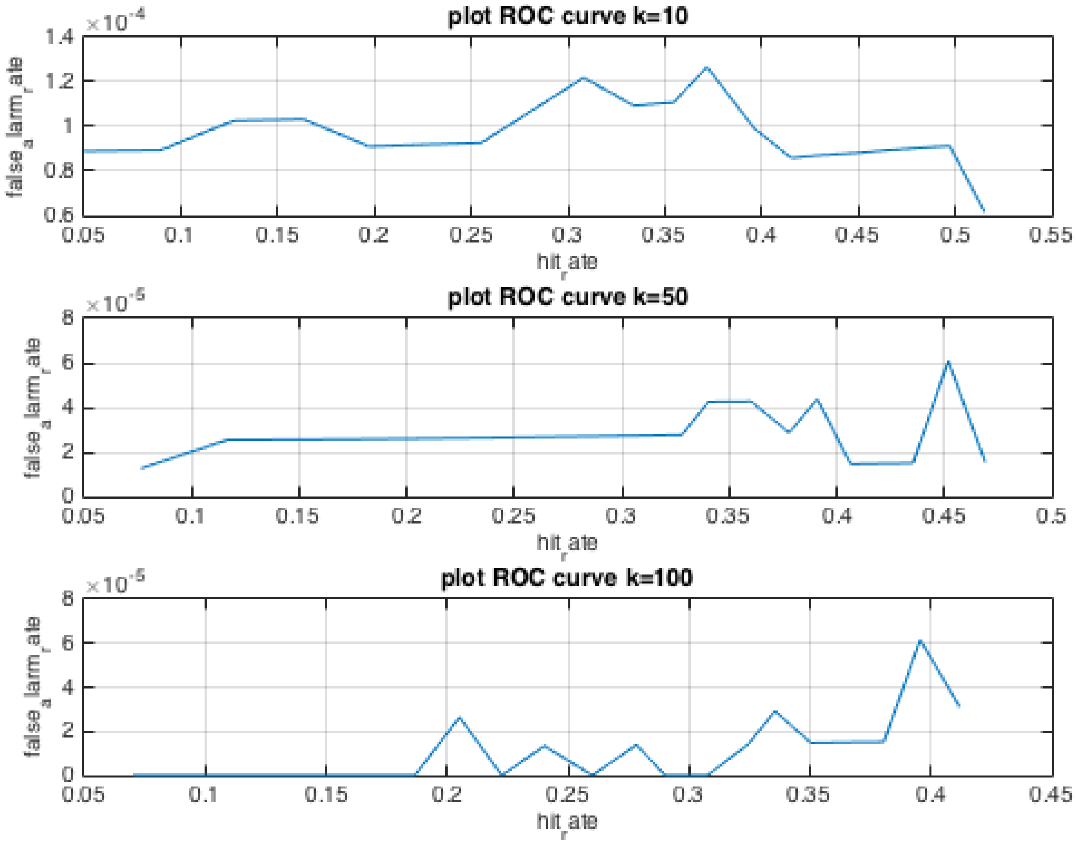


FIG. 16: ROC curve of Problem 5

We can see that k has a great influence on the roc curve. We choose L form 1 to 20. Since the total number of movies that are not suggested to the user is really large, the false alarm rate is really low.

IV. DISCUSSION & CONCLUSION

In this project, we have developed a simple Recommendation systems which help us to suggest movies. The method we use to do recommendation is Collaborative Filtering. Collaborative Filtering refers to methods of predicting a user's opinion on an entity using other users' opinion

We can view this as a matrix problem by putting all users on rows and all items on columns of the matrix. Then the entry (i,j) of the matrix will be the rating user i has given to item j. The problem will now become estimating the empty entries of the matrix to predict what items a user will most probably like other than the ones they have rated.

To do this we factorize the rating matrix R to U,V such that R is close to UV where data exist. And other values in UV, where no data exist in our origin rating matrix, are the prediction we make.

For the cost function, we can just measure the difference between R and UV in where data exist. But we can also add a regularization term λ, which will make our matrix calculations more stable.

We also set a threshold to the movies we suggest. For example, if a movie's rating is higher than 3, we will suggest it. Or, if a movie is in the top 5 list of a user, we will suggest. The threshold will influence our final precision and recall, and will let us to see how our model works in different situation. However the threshold itself won't do anything to how we create the model and train the model. The prediction result has already been done before we set any threshold. The only difference is how we deal with the result.