Kevin Wang (604036846)

EE 239AS Project 3

Part 1

The wnmfrule function from the Matrix Factorization toolbox is used in MATLAB to complete this project. In this part, the weight matrix W is created with 1's where there are data and 0's where there is no data.

The Frobenius Norm is found for the difference between R and R_new=W.*U*V. This value is the same as the error residual parameter returned from the wnmfrule function. Also reported is the standard norm of R_new-R.

k	Error (Frobenius)	2-norm
10	235.7660	26.2358
50	145.7698	17.7104
100	87.5316	11.8411

Part 2

In this part, the data is tested using 10-fold validation. For each of the 10 trials, the average absolute error R_{pred}-R_{actual} is found. Also listed is the highest and lowest values of the error. For this part, the maximum number of iterations was set to 200 such that the computation could finish in a reasonable amount of time.

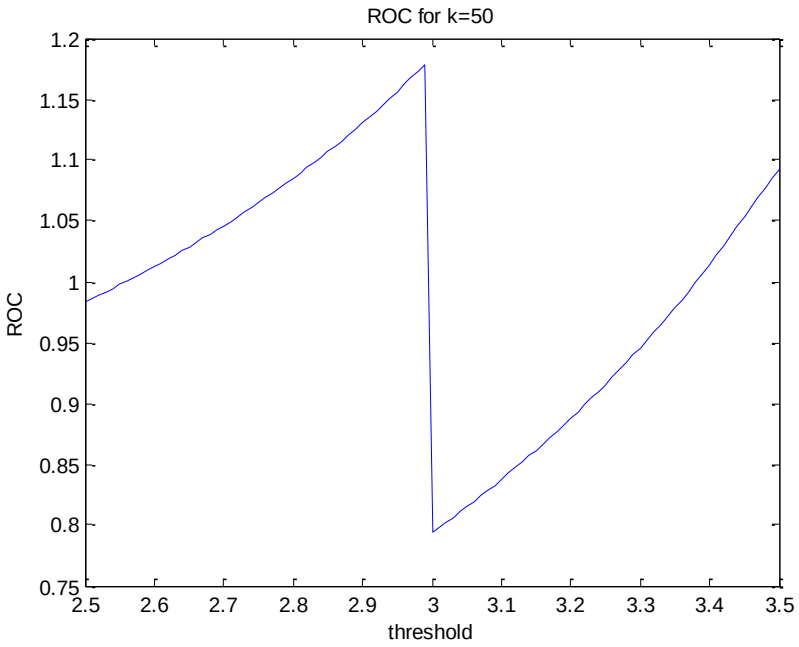
k	Average Absolute Error	Max error	Min Error
10	0.8027	0.8178	0.7936
50	0.9535	0.9798	0.9378
100	0.9774	1.0132	0.9602

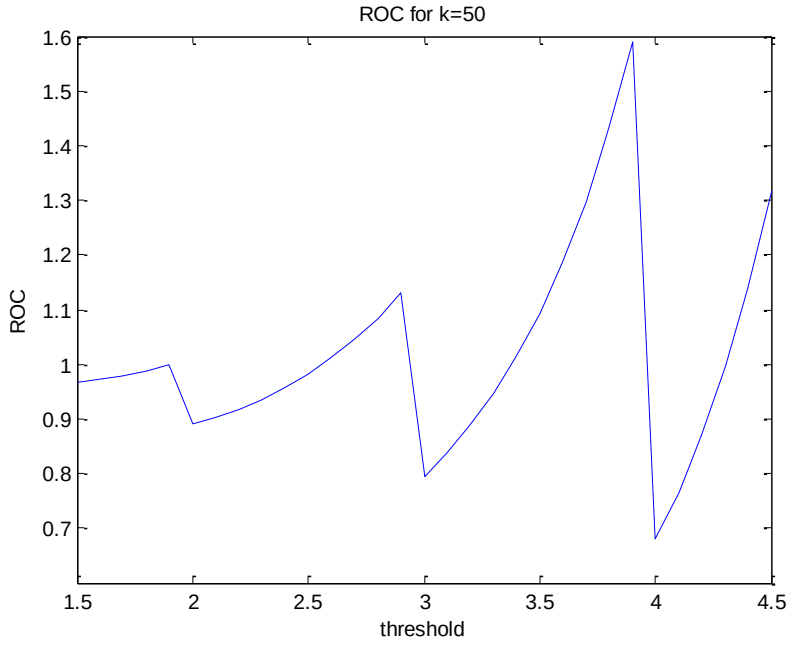
For some reason sometimes one or two of the validations would have an extremely large error; these values were thus dropped as perhaps they were results of wnmfrule placing a very large number into the predicted resulting matrix. The Frobenius norms that the function returns still seem reasonable in these cases, so it is likely that the function just predicted a very large number for one of the test data points, perhaps in a column/row with few actual ratings in which too many from the same row/column were removed for use as the test set.

Part 3

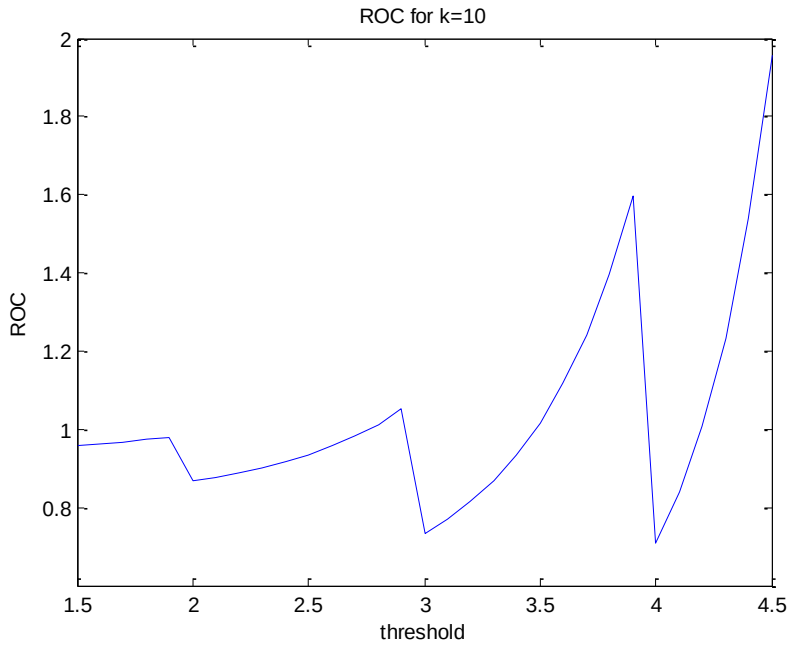
For this part, a recommendation system must be designed and tested using a method similar to that used in part 2.

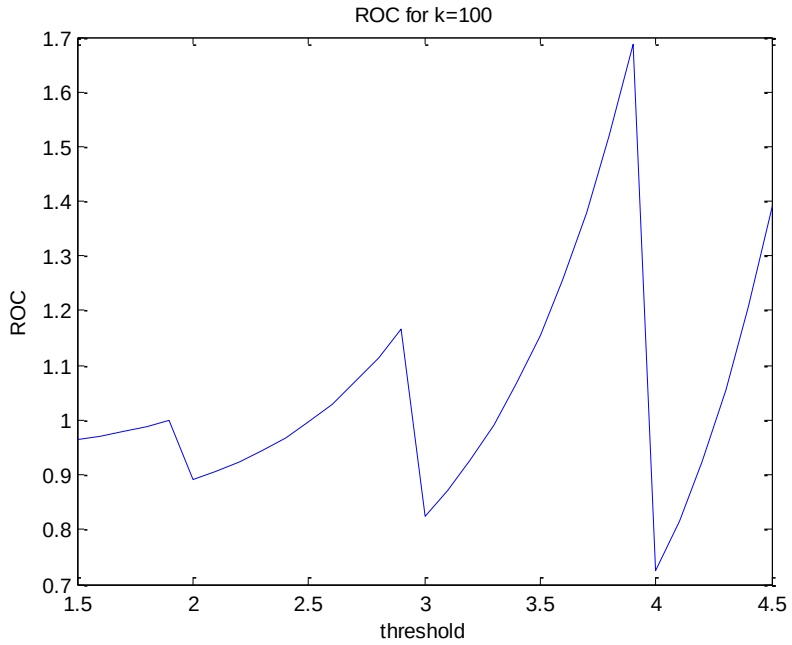
First, k=50 was tested. The threshold was swept from 2.5 to 3.5, then expanded from 1.5 to 4.5 to check for patterns. It seems that ROC=precision/recall rises until it hits an integer, where it drops and then starts rising again.





The following page contains the 1.5-4.5 threshold plots for k=10 and k=100.


ROC for k=10
ROC
threshold
2
1.8
1.6
1.4
1.2
1
0.8
1.5
2
2.5
3
3.5
4
4.5


ROC for k=100
ROC
threshold
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
1.5
2
2.5
3
3.5
4
4.5

Part 4

First, a small edit is made to the wnmfrule function from the toolbox to swap R and W such that W is factored instead. This function is saved as "wnmfrule_4," and as expected predicts a resulting U*V matrix that is all approximately 1. The error is small, but making recommendations will be difficult using this system.

k	Error (Frobenius)	2-norm
10	6.0432	1.3597
50	15.6563	2.4486
100	15.5266	2.0776

The "wnmfrule_reg" function changes the algorithm to include the regularization terms to the one in the reference slides, X=AY:

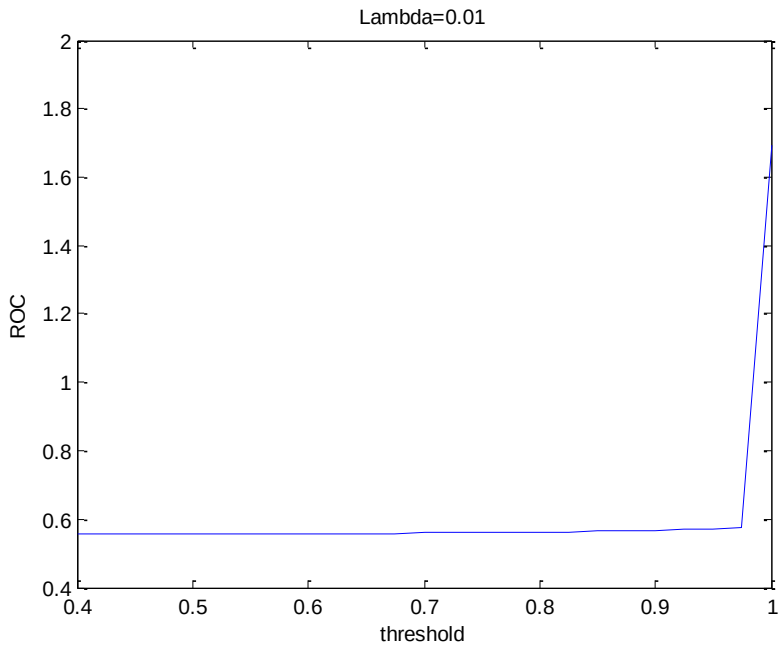
```
A(u,:) = linsolve(((Y * diag(W(u,:))* Y') + lambda*eye(k)),Y*diag(W(u,:))*X(u,:)')';
Y(:,v) = linsolve(((A' * diag(Wt(v,:))* A) + lambda*eye(k)),A'*diag(Wt(v,:))*X(:,v));
```

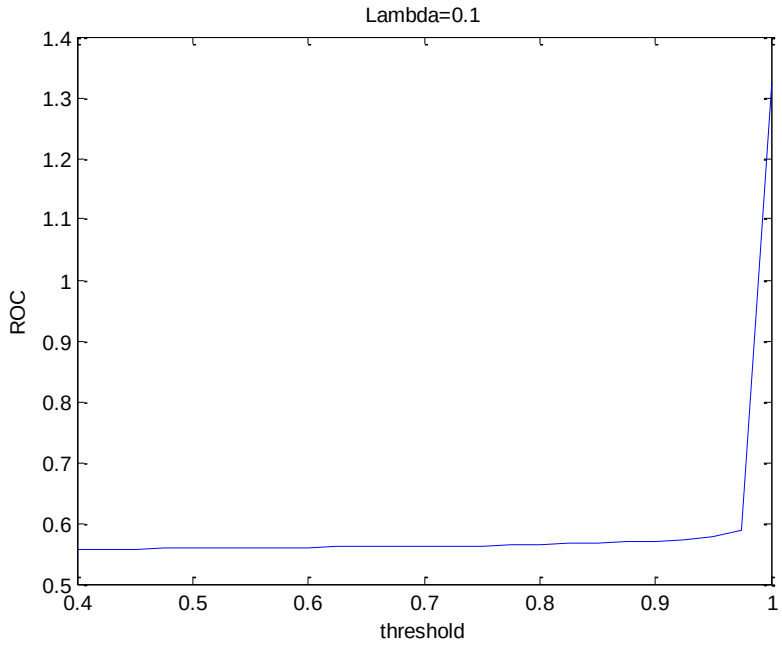
The matrices are solved alternately row-by-row or column-by-column.

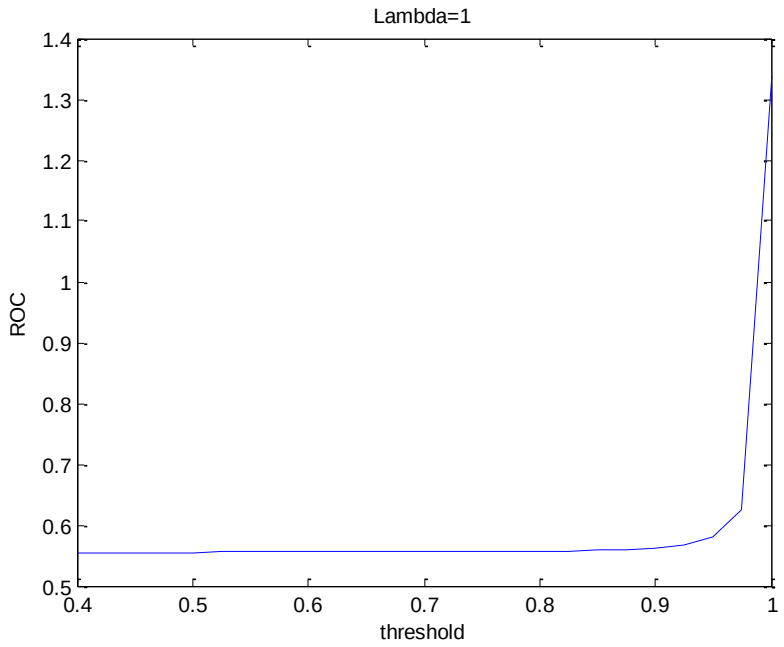
$$A(u) = (YW(u)Y^T + \lambda I)^{-1}(YW(u)X^T)$$
$$Y(v) = (AW(v)A^T + \lambda I)^{-1}(AW(v)X^T)$$

In the algorithm above, u and v are row indices, and W(u) or W(v) are diagonal matrices of the corresponding Weight Matrix row/column.

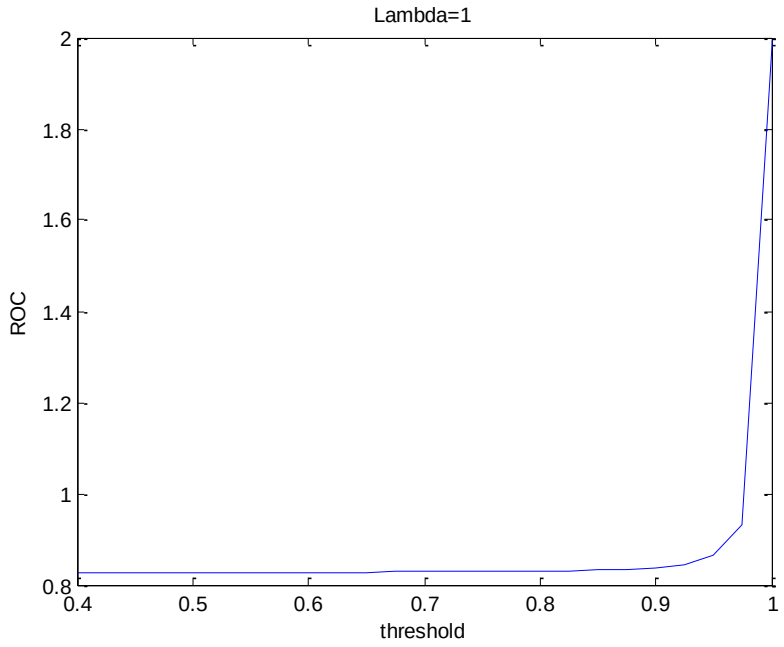
Unfortunately, this function was very slow, so the max iterations was dropped to a smaller number. Since the resulting U*V matrix instead represented the weights, the threshold to determine if a user liked a movie originally was fixed to be scores greater than 3. Then, various thresholds for the weight matrix were tested to create the ROC plots; here, the plots for 3 different lambdas are displayed for k=10:


Lambda=0.01
ROC
threshold
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.4
0.5
0.6
0.7
0.8
0.9
1


Lambda=0.1
ROC
threshold
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.5
0.6
0.7
0.8
0.9
1



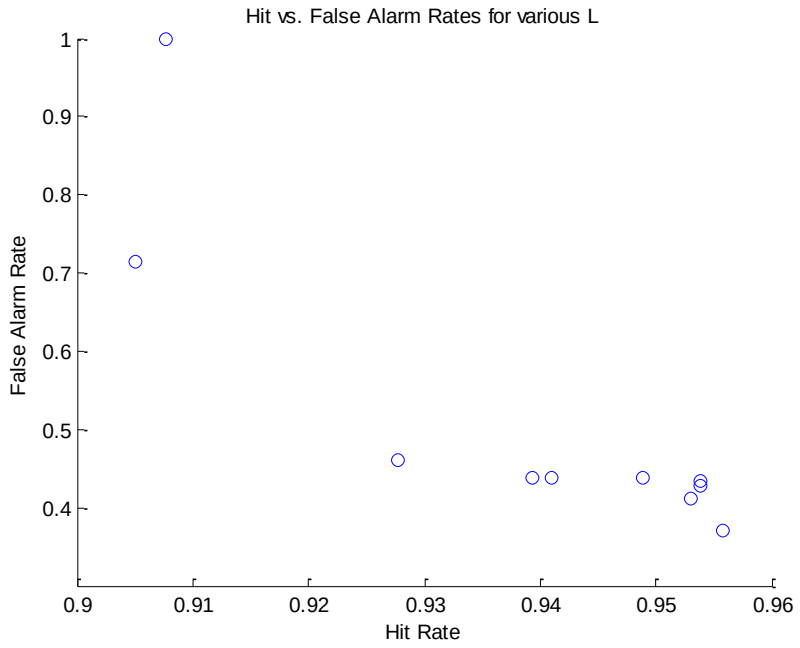
We can also test the system if we fix the threshold to include 3 as an acceptable liked rating as well; the next plot shows the result with lambda=1. The shape of the plot remains similar.



Part 5

We run the recommendation system with L=5, to find the top 5 movies that are recommended to each user. Only movies in which there were actual ratings are considered; if a user has rated a number X less than 5 movies, then only the top X movies are considered instead.

The precision for this system is found to be 0.9686.



L is then swept from 1 to 10, and the total hit/false alarm rates are recorded in the plot on the previous page. Since some users had very few ratings, perhaps it was very hard to come up with proper recommendations since they could have either all or none of their data as part of the testing sets.