Project 3: Collaborative Filtering

Prof. Vwani Roychowdhury

ECE 219 Winter 2020

UCLA, Department of ECE

Siyuan Peng 805426447 Haonan Huang 605322629
Yinghan Cui 005430212 Yunzheng Zhu 505231998

1. Introduction

In this project, we are designing the recommender systems, which enables users to provide feedback about their likes or dislikes. It is to utilize the user data to infer customer interests. The entity is referred to as the user, and the product being recommended is referred to as an item. The basic models are working with the two kinds of data here: User-Item interactions such as ratings, and attribute information about the users and items such as textual profiles or relevant keywords. The two corresponding types are collaborative filtering methods and content based methods. In this project, collaborative filtering methods will be designed.

2. Collaborative filtering models

The two types of collaborative filtering methods are implemented:

1. Neighborhood-based collaborative filtering
2. Model-based collaborative filtering

3. MovieLens dataset

In this project, we are building a recommendation system to predict the ratings of the movies in the MovieLens dataset. Since there are 610 users and 9724 rated movies, the rating matrix is 610 x 9724.

Question 1 - Compute the sparsity of the movie rating dataset

$$Sparsity = \frac{\text{Total number of available ratings}}{\text{Total number of possible ratings}} = \frac{\text{Total number of available ratings}}{610 \times 9724} = 0.016999683055613623$$

Question 2 - Plot a histogram showing the frequency of the rating values

Fig. 1 below shows the frequency of the rating values. The figure explains that the distribution is skewed left. Thus, the users are mainly giving rates in the range of 3.0 to 5.0, which is relatively high.

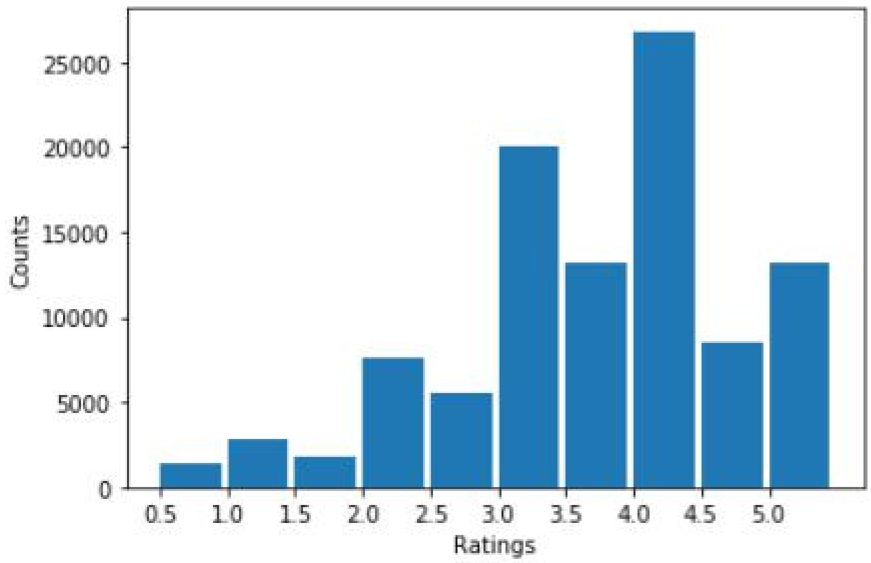


Figure 1. Histogram showing the frequency of the rating values.

Question 3 - Plot the distribution of the number of ratings received among movies

Fig. 2 below shows the distribution of ratings for movies. The figure is a monotonically decreasing curve.

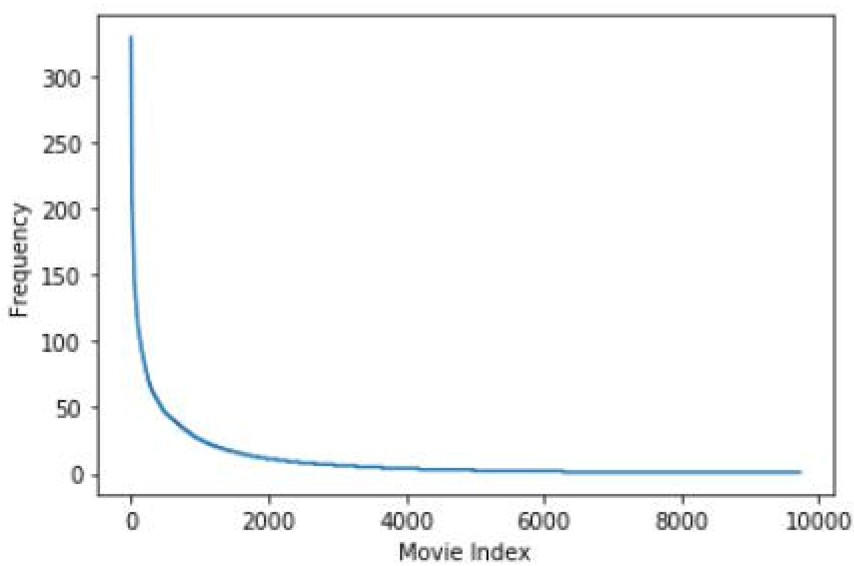


Figure 2. Distribution of the number of ratings received among movies.

Question 4 - Plot the distribution of ratings among users

Fig. 3 below shows the distribution of ratings among users. The figure is a monotonically decreasing curve.

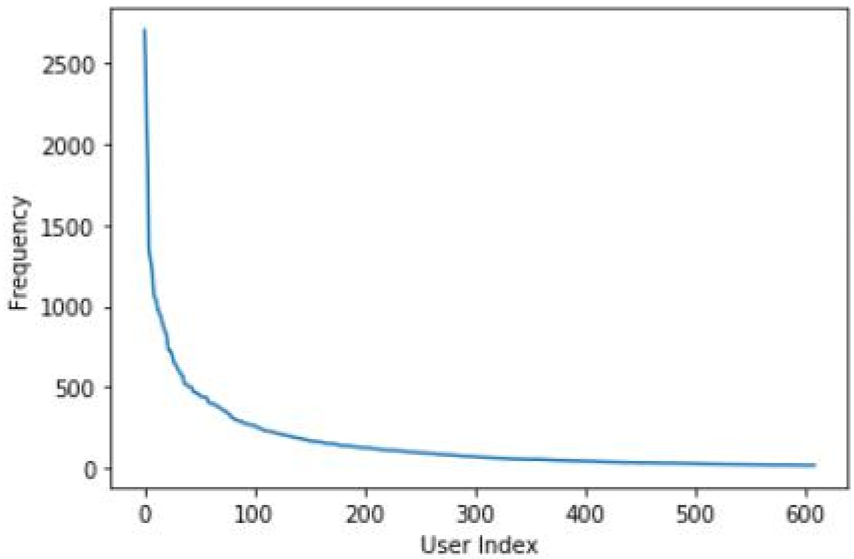


Figure 3. Distribution of ratings among users.

Question 5 - Explain the salient features of the distribution found in question 3 and their implications for the recommendation process

The distribution of ratings among movies is skewed right. There are just a few movies receiving more than 50 ratings. Also, most of the movies are receiving not too many ratings. This implicates that rating matrix is highly related to the quality of the recommender system, correspondingly the choice of the recommendation process is important.

Question 6 - Compute the variance of the rating values received by each movie

Fig. 4 below shows that the distribution of the variance of ratings is skewed right. Thus, for most of the movies, the variance of ratings is not too large. Only a few movies have variance of ratings higher than 2.0. And there are no movies having variance of ratings higher than 3.5.

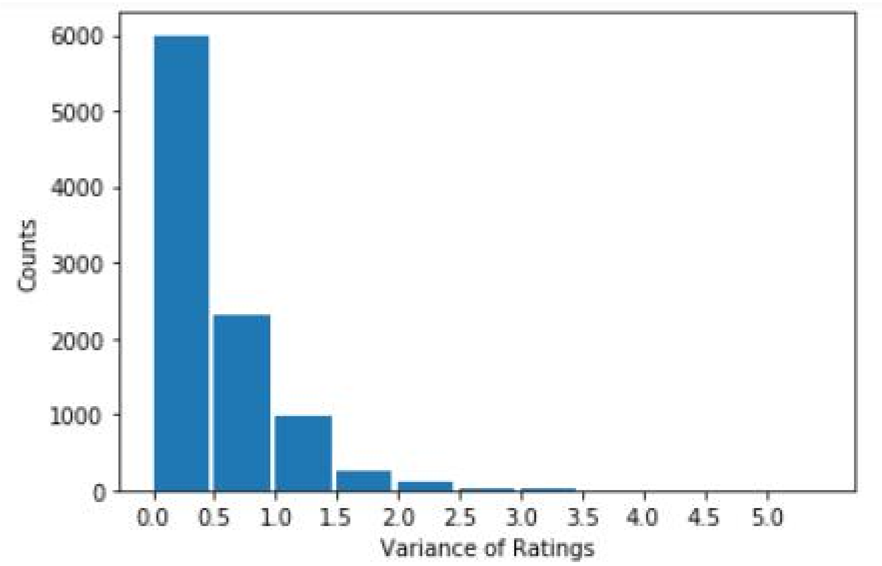


Figure 4. Histogram showing variance of ratings among users.

4. Neighborhood-based collaborative filtering

There are two basic principles used in neighborhood-based models: user-based models and item-based models. In this project, we will only implement user-based collaborative filtering.

We define a few notations for the Pearson-correlation coefficient Pearson(*u*,*v*):

I_u : Set of item indices for which ratings have been specified by user u

I_v : Set of item indices for which ratings have been specified by user *v*

μ_u: Mean rating for user *u* computed using her specified ratings

r_{uk}: Rating of user *u* for item *k*

Question 7 - Mean rating for user *u* computed using the specified ratings

First, the equation for the mean rating for user *u* is:

$$\mu_u = \frac{1}{|I_u|} \sum_{k \in I_u} \mathrm{r}_{uk}$$

Question 8 - Explain the meaning of $I_u \cap I_v$. Can $I_u \cap I_v = \varnothing$?

$I_u \cap I_v$ is the set of item indices for which ratings have been specified by both user u and user v. Since the rating matrix R is sparse, it is possible to have $I_u \cap I_v = \varnothing$. This means there is no item indice rated by both user u and user v.

Question 9 - Prediction function

$$\hat{r}_{uj} = \mu_u + \frac{\sum_{v \in P_u} Pearson(u,v)(r_{vj} - \mu_v)}{\sum_{v \in P_u} |Pearson(u,v)|}$$

The mean-centering the raw ratings $(r_{vj} - \mu_v)$ is to make the prediction function of user u independent from the ratings of user v. The demeaned ratings will be centered nearly at zero.

Question 10 - Design a k-NN collaborative filter

First, we choose the k, the number of neighbors, in the k-NN collaborative filter. Using 10-fold cross validation and sweep k from 2 to 100 in step sizes of 2. The performance is evaluated with the average of RMSE and MAE across all 10 folds. The plots are shown below:

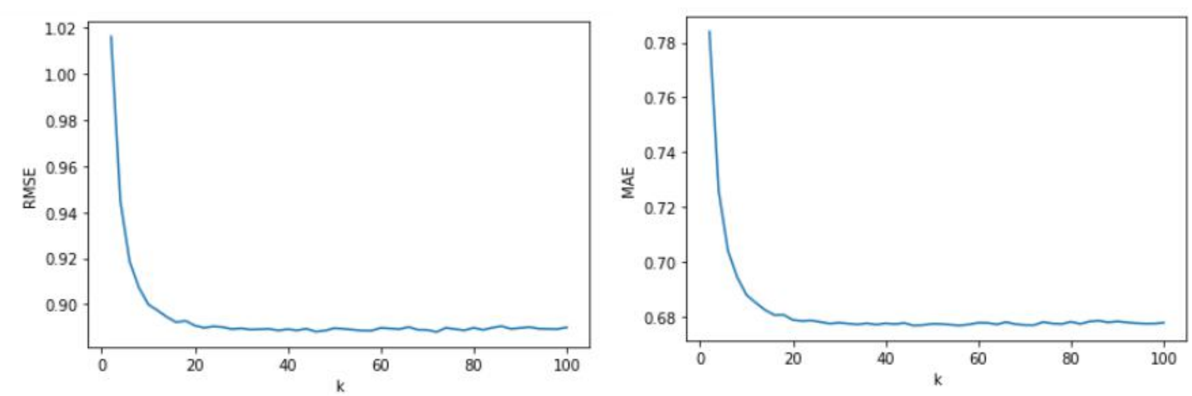


Figure 5. RMSE and MAE against number of neighbors k for k-NN collaborative filter

Both plots (RMSE and MAE) show that the value is dropping sharply initially as k increases (k < 20). After k reaching nearly 20, the variation becomes smooth (almost stable), which is correspondly converge.

Question 11 - Best value of neighbor k

In this question, we consider carefully RMSE and MAE values in Question 10, and design a good method of best k. Specifically, we use for loop for k value, and determine the best k if both RMSE and MAE values change less than 0.1% and they do not change larger than 0.1% for the next three steps.

By such method, we derive that best k value is 24. The steady state values of average RMSE is 0.890, and the steady state values of average MAE is 0.678.

Question 12 - Average RMSE with popular movie trimming

In Question 12-14, we design a k-NN collaborative filter to predict the ratings of the movies in the popular movie trimmed test set, the ratings of the movies in the unpopular movie trimmed test set, and the ratings of the movies in the high variance movie trimmed test set separately.

For Question 12, we plot Average RMSE with popular movie trimming graph, as shown below:

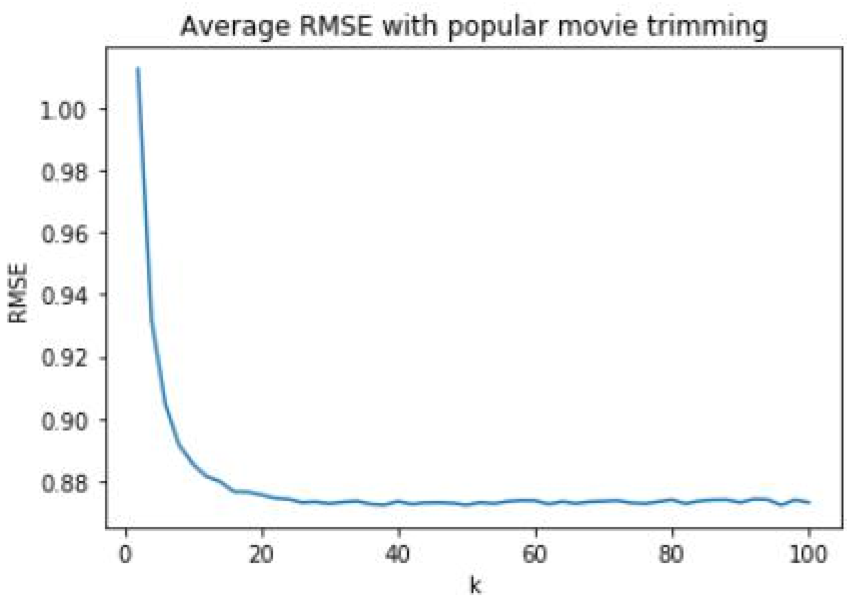


Figure 6. Average RMSE with popular movie trimming

We use similar criteria as Question 11 to derive best k value, which is 28 in our case. And the steady state values of average RMSE is 0.873.

Question 13 - Average RMSE with unpopular movie trimming

In this Question, we plot Average RMSE with unpopular movie trimming graph, as shown below:

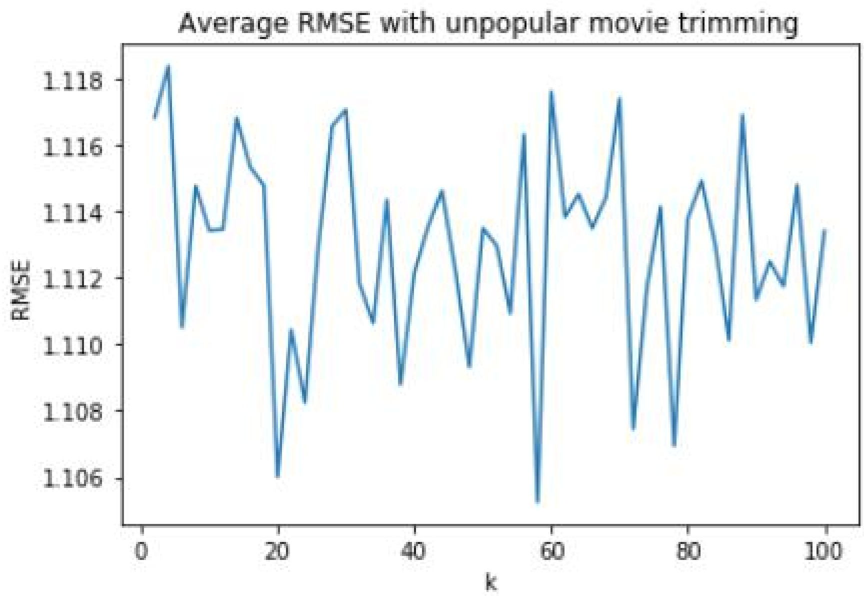


Figure 7. Average RMSE with unpopular movie trimming

This time, by using the same criteria, however, we cannot find a best k value. That is because RMSE does not converge(nor does MAE) to certain value. So we say instead the average RMSE is the average of the whole graph, which is 1.112.

Question 14 - Average RMSE with high variance movie trimming

In Question 14, we plot Average RMSE with high variance movie trimming graph, as shown below:

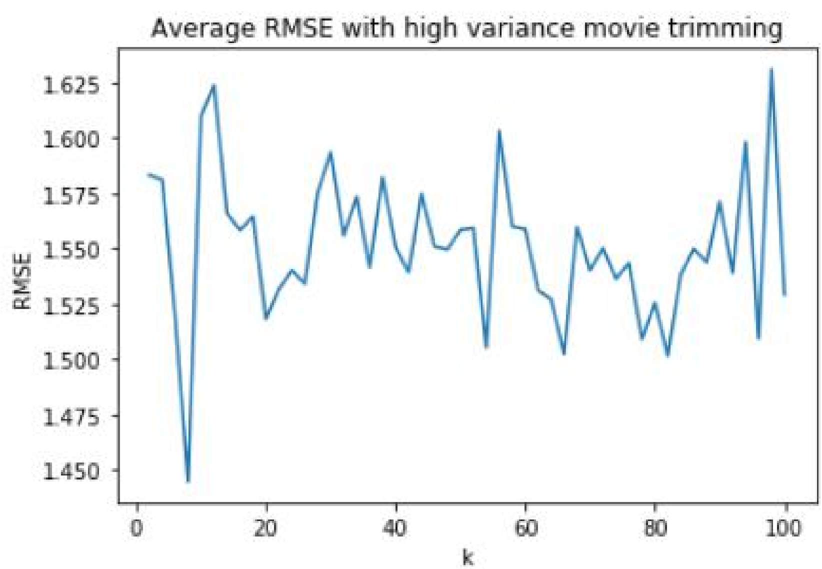


Figure 8. Average RMSE with high variance movie trimming

For the same reason as Question 13, we cannot find a best k value. That is because RMSE does not converge(nor does MAE) to certain value. So we say instead the average RMSE is the average of the whole graph, which is 1.551.

Question 15 - ROC curves for k-NN collaborative filtering

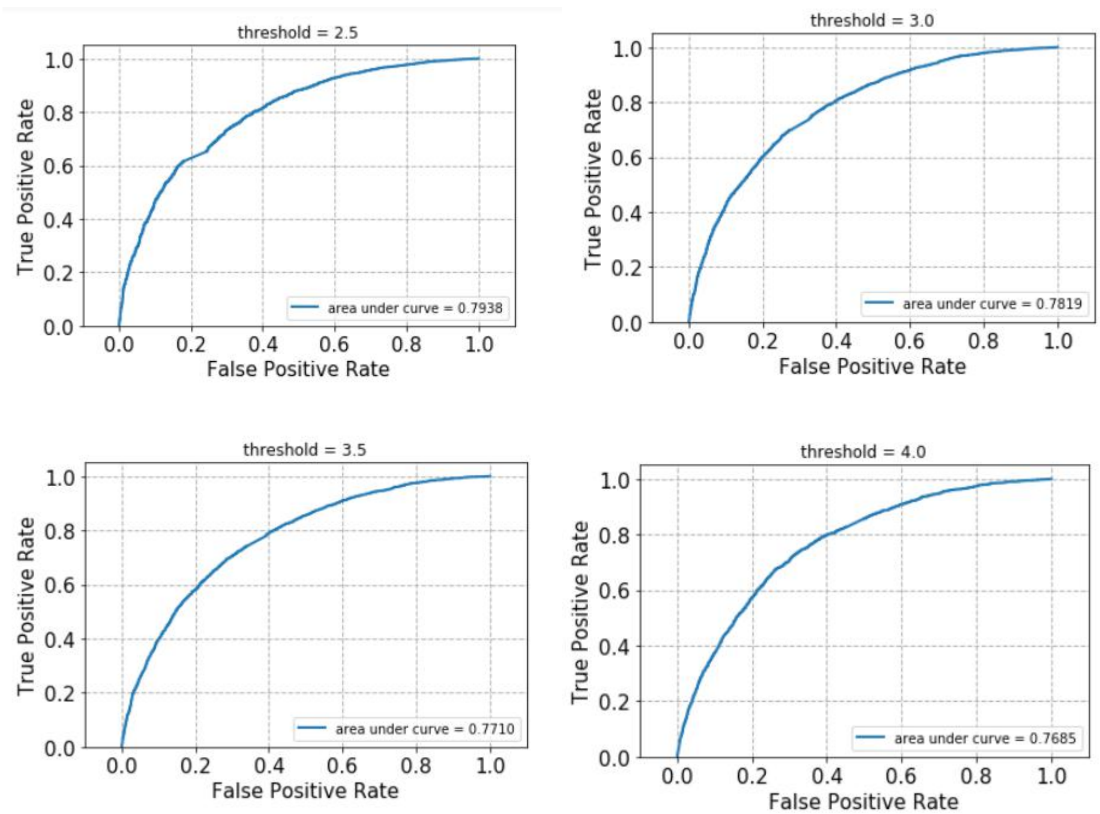


Figure 9. ROC curves for k-NN collaborative filtering

As shown above, 4 ROC curves for k-NN collaborative filtering are designed for threshold values [2.5,3,3.5,4].

We also calculate the AUC for each ROC curve, as shown below:

Table 1. AUC for the 4 ROC curves

Threshold 2.5 3 3.5 4

AUC values 0.7938 0.7819 0.7710 0.7685

5. **Model-based collaborative filtering**

Question 16 - Convexity of optimization problem of NNMF

The optimization problem is given as

$$\begin{aligned} \min_{\mathrm{U,V}} \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} W_{ij}(r_{ij} - (UV^T)_{ij})^2 + \lambda\|\mathrm{U}\|_F^2 + \lambda\|V\|_F^2 \\ s.t. \quad & U \geq 0, V \geq 0 \end{aligned}$$

To check its convexity is equivalent to checking the convexity of the following term:

$$L(U,V) = \frac{1}{2}\sum_{i=1}^{m} \sum_{j=1}^{n} W_{ij}(r_{ij} - (UV^T)_{ij})^2$$

To make things simpler, we take m=n=k=1, then R,U,V are all 1*1 scalars. Assume W = 1, then we have

$$L(U,V) = \frac{1}{2}(R - UV)^2$$

Then calculate the gradient and Hessian matrix.

$$\nabla \mathrm{L}(U,V) = \begin{bmatrix} \dfrac{\partial L}{\partial U} \\ \dfrac{\partial L}{\partial V} \end{bmatrix} = \begin{bmatrix} -(R-UV)V \\ -(R-UV)U \end{bmatrix}$$

$$\nabla^2 \mathrm{L}(U,V) = \begin{bmatrix} \dfrac{\partial^2 L}{\partial U^2} & \dfrac{\partial^2 L}{\partial U \partial V} \\ \dfrac{\partial^2 L}{\partial V \partial U} & \dfrac{\partial^2 L}{\partial V^2} \end{bmatrix} = \begin{bmatrix} V^2 & -R+2UV \\ -R+2UV & U^2 \end{bmatrix}$$

The determinant of Hessian matrix is

$$\left|\nabla^2 L(U,V)\right| = U^2V^2 - (-R+2UV)^2 = -(r-UV)(R-3UV)$$

which is not all positive with values R,U,V>0. Hence, Hessian matrix is not positive semi-definite, and the problem is non-convex.

We can extend this conclusion in general for NNMF.

We need to show that the optimization problem can be solved by alternating least square(ALS) method.

First, we denote some terms. For each j=1,...,n, we denote

$$r_j = \left[r_{1j}, \ldots\ldots r_{mj}\right]^T; V_j = \left[V_{1j}, \ldots\ldots V_{kj}\right]^T; W_j = diag\{W_{1j}, \ldots\ldots W_{mj}\}$$

Consider that if U is fixed, L function in the problem becomes

$$L(V) = \frac{1}{2}\sum_{j=1}^{n}(r_j - UV_j)^T W_j (r_j - UV_j)$$

By minimizing the loss function L(V), we get the solutions for the movie factors V:

$$V_j = (U^T W_j U)^{-1} U^T W_j r_j; \qquad (j = 1,....n)$$

which is the formula for the Least Square estimate.

Similarly, if V is fixed, we get the solutions for the user factor U:

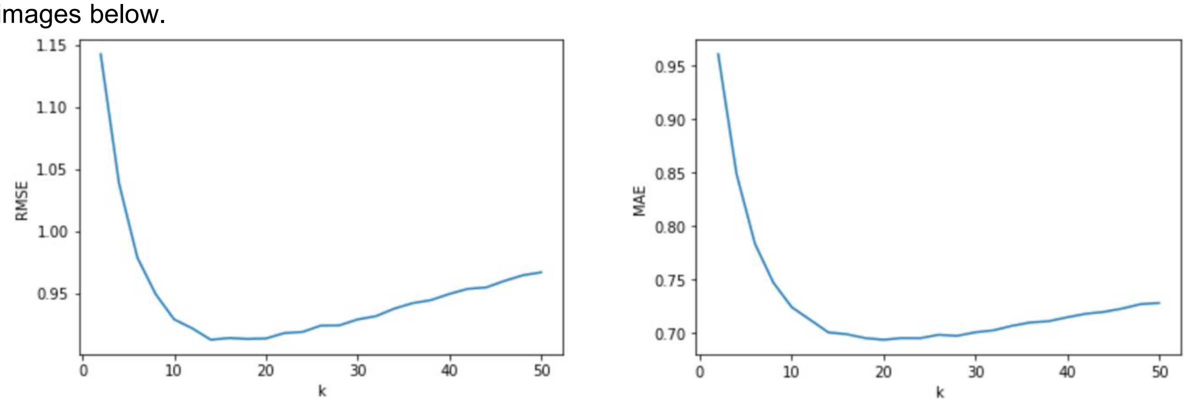
$$U_i = (V^T W_i V)^{-1} V^T W_i r_i; \qquad (j = 1,....n)$$

where $r_i = [r_{i1},.....r_{in}]^T; U_i = [U_{i1},.....U_{ik}]^T; W_i = diag\{W_{i1},....W_{in}\}$

The property of NNMF problem allow us to use ALS as the optimization algorithms. The main idea is first to keep U fixed and then solve for V. Next, keeping V fixed and solve for U. At each stage we are solving a Least Square problem that enables the algorithm to be stable and has a convergence factor.

Question 17 - NNMF-based collaborative filter

In this question, we chose the best number of latent factors, k, in a NNMF collaborative filter. We use 10-fold cross validation and sweep k from 2 to 50 with step 2. The performance of the prediction is evaluated by using the average of RMSE and MAE across all 10 folders. And the results are analyzed as the images below.



a) RMSE against k

b) MAE against k

Figure 10. The plot for RMSE and MAE against k

As k increases, RMSE and MAE decrease at first and then increase after the turning point. For small values of k, the model underfits as it is not complex enough to capture the features in R. The estimation error starts with high values and so does the RMSE. As k increases, the bias drops quickly which drives the RMSE to decrease at first. But increasing in model complexity also leads to the overfitting problem and the prediction variance will increase. The RMSE will increase after some turning point.

Question 18 - Optimal number of latent factor

Choosing the optimal number of latent factors k is a trade-off between estimation bias and prediction variance.
If we use minimize RMSE, we have kopt = 14, RMSEmin = 0.9127
If we use minimize MAE, we have kopt = 20, MAEmin = 0.6938

The number of movie genres in our data set is 20. And we can see the optimal number of latent factors is close to that number but not exactly the same. And when using MAE, the latent factor is equal to the number of our data set

Question 19,20,21

In Questions 19-21, we performed the NNMF collaborative filter in predicting the ratings of the movies in the trimmed test set.

Question 19 - Movie trimming-popular

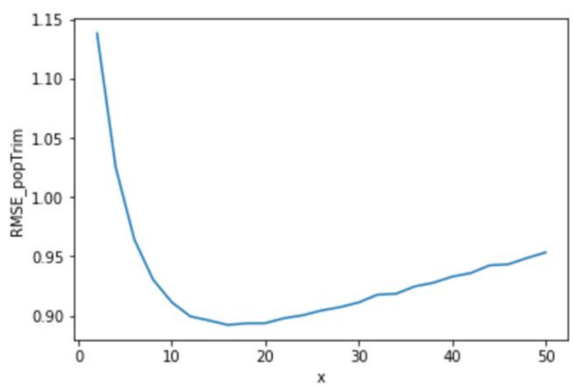


Figure 11. RMSE against k for popular-trimmed movies

The optimal number of latent factors is kopt = 16 and correspondingly the minimum average RMSE is RMSEmin = 0.8921. We find the minimum average RMSE is lower than the original data set and the reason is the same as Question 11.

Question 20 - Movie trimming-unpopular

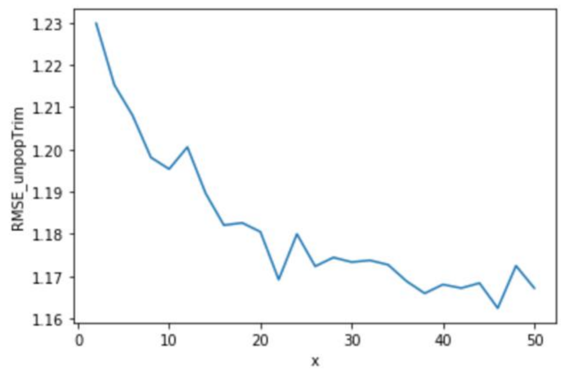


Figure 12. RMSE against k for unpopular-trimmed movies

For the unpopular movie trimmed set, we plotted the RMSE against the number of latent factors k in figure above. Under this case, the RMSE drops at first and then becomes flat when sweeping k from 2 to 50.

Also, there are small fluctuations along the trend. The optimal number of latent factors is kopt = 46 and correspondingly the minimum average RMSE is RMSEmin = 1.1624.

Question 21 - Movie trimming-high variance

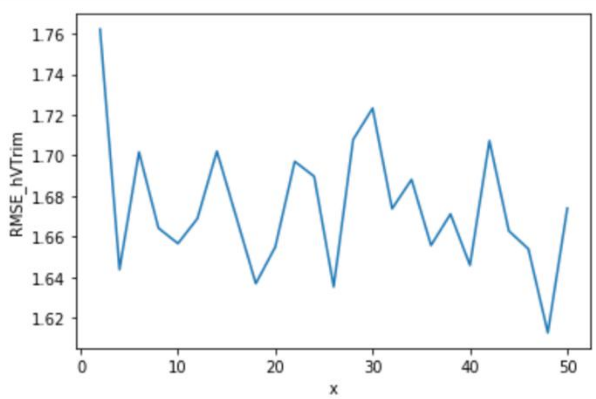


Figure 13. RMSE against k for high variance movies

For the high variance movie trimmed test set, we plotted the RMSE against the number of latent factors k in figure above. Under this case, the RMSE does not have a clear trend when sweeping k from 2 to 50. The reason is that for high variance movies, the correlations across users and movies are very low. And NNMF prediction which is based on the correlations should perform very poorly no matter what the k is chosen.

The optimal number of latent factors is kopt = 48 and correspondingly the minimum average RMSE is RMSEmin = 1.6127.

Question 22 - ROC curves for NNMF collaborate filter

We use the ROC curve to evaluate the performance of the NNMF collaborative filter with kopt = 16. Similar to Question 15, we also classify the ratings into binary scale with threshold values [2.5, 3, 3.5, 4]. And then the four ROC curves are plotted in figures below.

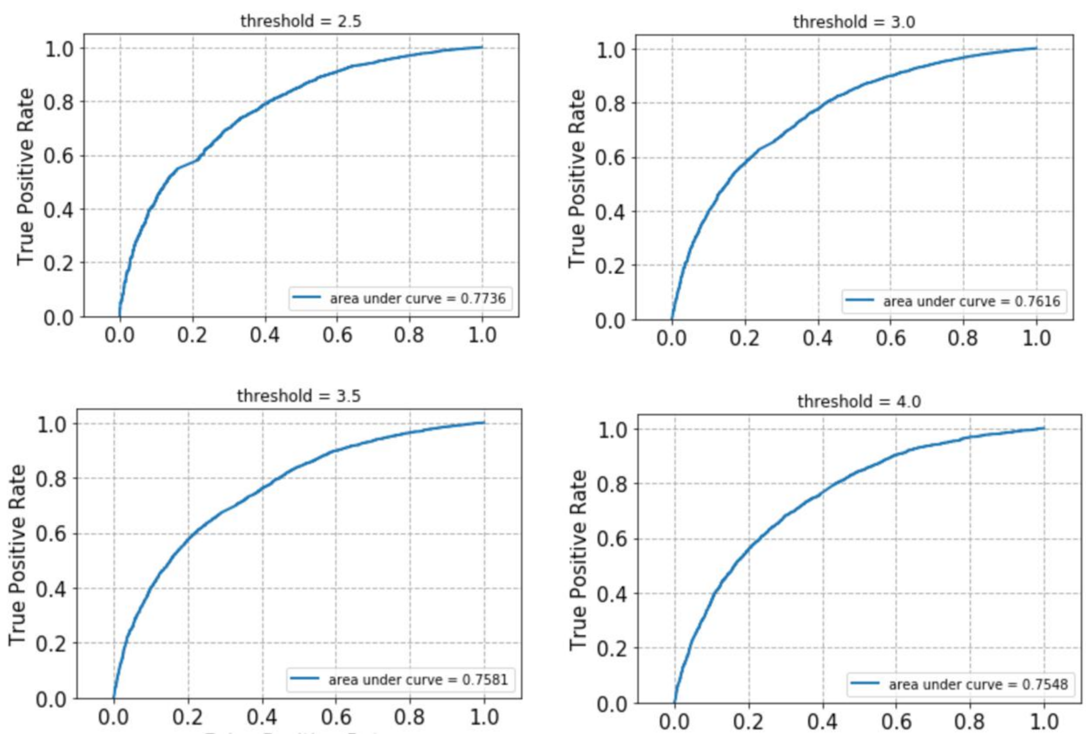


Figure 14.ROC curves for NNMF collaborate filter

For each of the ROC curve, we summarized the area under the curve (AUC) values in the following table:

Threshold	2.5	3	3.5	4
AUC value	0.7736	0.7616	0.7581	0.7548

Table 1. AUC for the ROC curves

Question 23 - Explore the NNMF method

In this question, we explore the interpretation of the NNMF method. Ideally, each latent factor should represent one movie genre. We choose the number of latent factors k = 20 and obtain the movie latent factors in *V* matrix. For each column of *V*, we pick the top 10 movies and for column 1 and column 3 their genres are reported in the table below.

Top 10 movies in column 1	Top 10 movies in column 3
Comedy	Fantasy|Horror|Mystery|Thriller
Comedy|Romance	Drama
Fantasy|Horror|Mystery|Thriller	Children|Drama
Drama|Thriller	Adventure|Comedy|Fantasy|Sci-Fi
Horror|Sci-Fi|Thriller	Children|Comedy|Romance

Musical	Drama
Sci-Fi|Thriller	Drama
Action|Adventure|Sci-Fi	Comedy|Drama|Romance
Mystery|Thriller	Drama|Horror|Thriller
Comedy	Comedy|Romance

Table 2. Genres of Top 10 movies in column 1 and column 3

We know the top 10 movies in each column do not belong to a particular genre but a small collection of genres. For example, we can conclude that column 1 represents the Comedy or Thriller movies. And column 3 should be Comedy or Drama movies.

Although the latent factors from NNMF is not one-to-one mapped to the 20 movie genres in our data set because we only have a very sparse rating data. They are closely connected in that each latent factor represents only a few collections of movie genres.

Question 24 - MF-based collaborative filter with bias

In this question, we modify the MF by adding biased terms for both user (bu), and movie (bi). These two components represent the deviation of each user u and each movie i from the overall mean rating μ. UV T now only captures the interaction between movie and user.

Compared with the NNMF method, the prediction performance of MF with bias is better in terms of both lower RMSE and lower MAE.

The RMSE and MAE against k are shown as figures below:

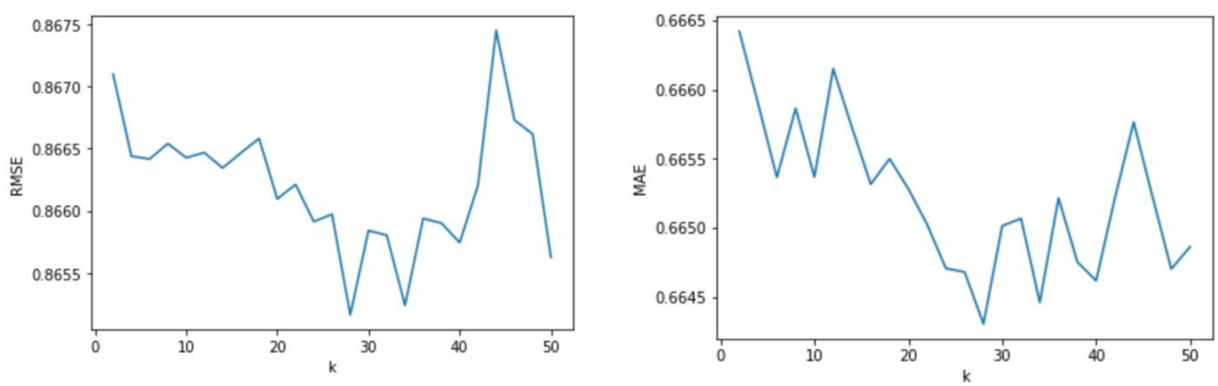


Figure 15. RMSE and MAE against k

Question 25 - Optimal number of latent factors

Choosing the optimal number of latent factors k is a trade-off between estimation bias and prediction variance.

If we use minimize RMSE, we have kopt = 28, RMSEmin = 0.8651
If we use minimize MAE, we have kopt = 28, MAEmin = 0.6643

In Questions 26-28, we performance of the MF with bias collaborative filter in predicting the ratings of the movies in the trimmed test set.

Question 26: Movie trimming-popular

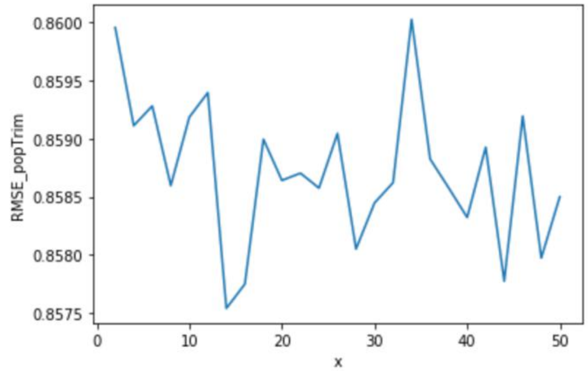


Figure 16. RMSE against k for popular-trimmed movies

The optimal number of latent factors is kopt = 14 and correspondingly the minimum average RMSE is RMSEmin = 0.8575.

Question 27: Movie trimming-unpopular

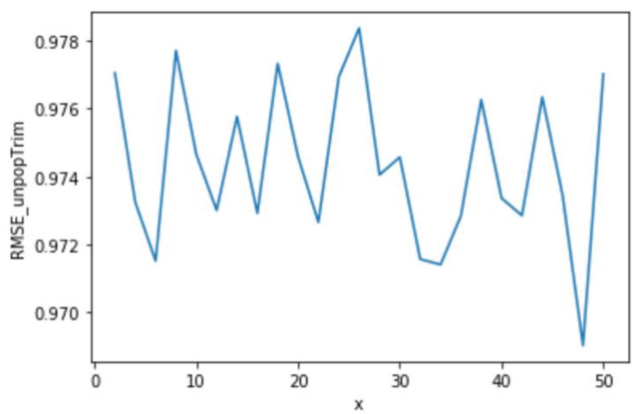


Figure 17. RMSE against k for unpopular-trimmed movies

The optimal number of latent factors is kopt = 48 and correspondingly the minimum average RMSE is RMSEmin = 0.9690.

Question 28: Movie trimming-high variance

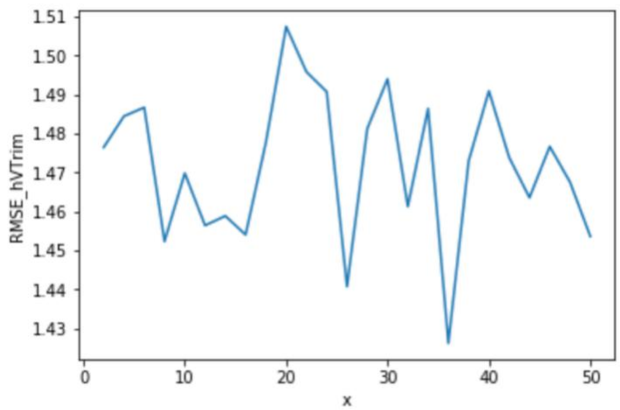


Figure 18. RMSE against k for high variance movies

The optimal number of latent factors is kopt = 36 and correspondingly the minimum average RMSE is RMSEmin = 1.4262.

Question 29: ROC curves for MF with bias

For each of the threshold values [2.5; 3; 3.5; 4]. we plotted the ROC curves in figures below

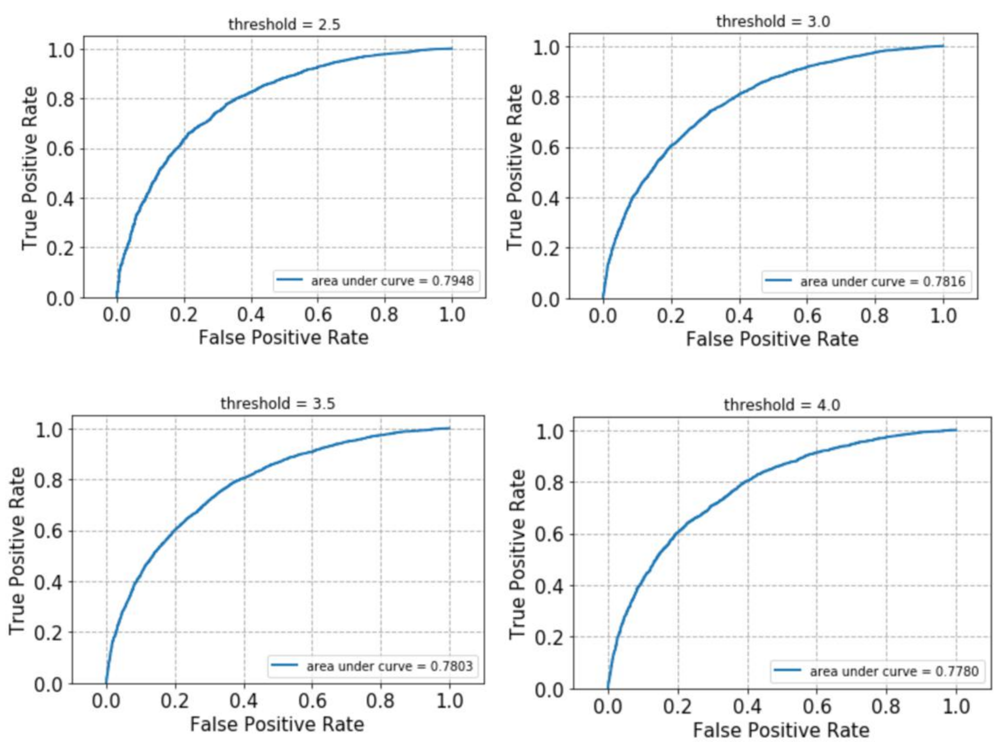


Figure 19. ROC curves for the MF collaborative filter

For each of the ROC curve, we summarized the area under the curve (AUC) values in the following table:

Threshold	2.5	3	3.5	4
AUC value	0.7948	0.7816	0.7803	0.7780

Table 3. AUC for the ROC curves

6. Naive collaborative filtering

Question 30 - Design and test via cross-validation

We design the naive collaborative filter to predict the ratings of the movies in the MovieLens dataset and evaluate it's performance using 10-fold cross validation. The average RMSE across all 10 folds is 0.9346786847813551 and we can compare this with above filtering solutions. The RMSE of naive collaborative filtering is the highest.

RMSE=0.9346786847813551

Question 31 - Predict the ratings of the popular movie trimmed test set

The average RMSE of popular movie trimmed test set across all 10 folds is 0.9322922328020201

The RMSE of the popular movie trimming set is slightly lower than the original data set. The reason may be that each user has more available ratings for popular movies, and the predictions made using average ratings are relatively reliable.

RMSE_popular=0.9322922328020201

Question 32 - Predict the ratings of the unpopular movie trimmed test set

The average RMSE of unpopular movie trimmed test set across all 10 folds is 0.9711408873803059

The RMSE for the unpopular movie trimmed set is higher than the original data set. It is due to the lack of available ratings which can be used to computer the average ratings. So the naive prediction is even worse in this situation.

RMSE_unpopular=0.9711408873803059

Question 33 - Predict the ratings of the high variance movie trimmed test set

The average RMSE of high variance movie trimmed test set across all 10 folds is 1.4535715353696466

The RMSE of the high variance movie trimming set is even worse because the correlations across users and movies are very low. And collaborative filtering method which is based on the correlations performs very poor.

RMSE_highvariance=1.4535715353696466

7. Performance comparison

Question 34 - Compare the performance of the filters

We plot the ROC curves (threshold = 3) for the k-NN, NNMF, and MF with bias based collaborative filters in the same figure. We can see from the figure that the AUC of MF with bias performance a litter better while all three filters performance very close.

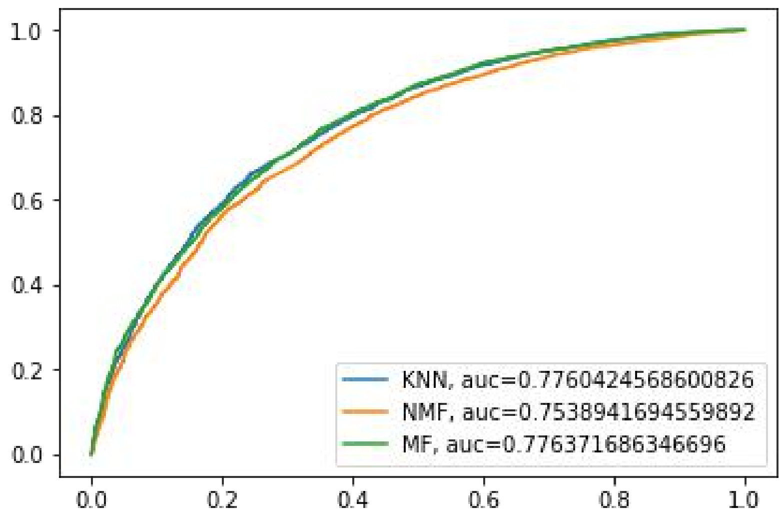


Figure 20. ROC curve for k-NN, NNMF, and MF with bias

8. Ranking

Question 35 - The meaning of precision and recall

Accuracy is the proportion of related instances in the retrieved instances. It evaluates the correctness of percentages in recommendation system that is liked by the users. Recall is the proportion of the total number of related instances actually retrieved. It evaluates the ability that whether the recommendation system is able to capture all items liked by the users.

Question 36 - k-NN collaborative filter predictions

We can see from the figure that movies recommended to users with the highest t predictions will make the set larger when we sweep t from 1 to 25. Our model can roughly predict user interest correctly, in the precision curve, the accuracy is high at first and then, as the t increases, the precision decreases. The recall curve is shown in the middle figure. As t increases, the recall will increase as expected. In the trade-off between precision and recall in the right figure, we see that the recall increases rapidly with a small drop in accuracy, which is an ideal behavior.

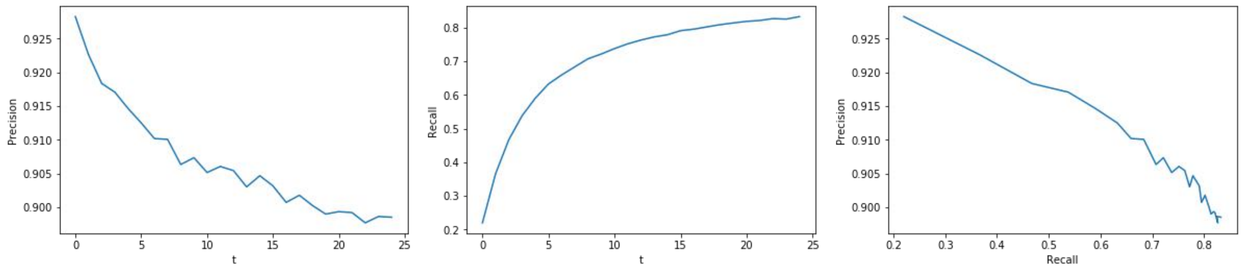


Figure 21. k-NN collaborative filter predictions

Question 37 - NNMF-based collaborative filter predictions

The curve figure in question 37 is similar to that in question 36. We have plotted the precision curve in the left figure. As t increases, the accuracy decreases, but the total value is still greater than ninety percent, which means that the predictions made by the filter are indeed what users like. The recall curve is shown in the middle figure. Its value increases as t increases, which means that the filter's predictions well cover most items that the user likes with a large number of t. The combination between the precision and the recall curve is shown in right figure. The precision decreases as the recall increases.

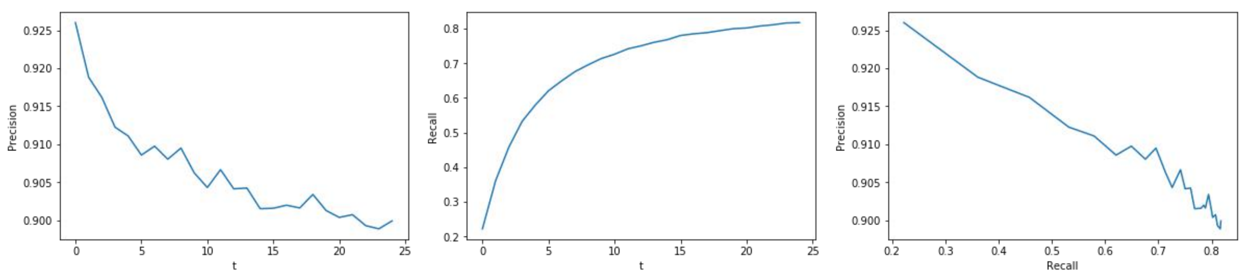


Figure 22. NNMF-based collaborative filter predictions

Question 38 - MF with bias-based collaborative filter predictions

We plotted the precision curve of MF with bias. We can see from the figure that as t increases, the precision decreases, but the total value is still greater than ninety percent, which means that the predictions made by the filter are indeed what users like. The recall curve is shown in the middle. Its value increases as t increases, which means that the filter's predictions well cover most items that the user likes with a large number of t. The combination between the accuracy and the recall curve is shown in the right figure. The precision decreases as the recall increases.

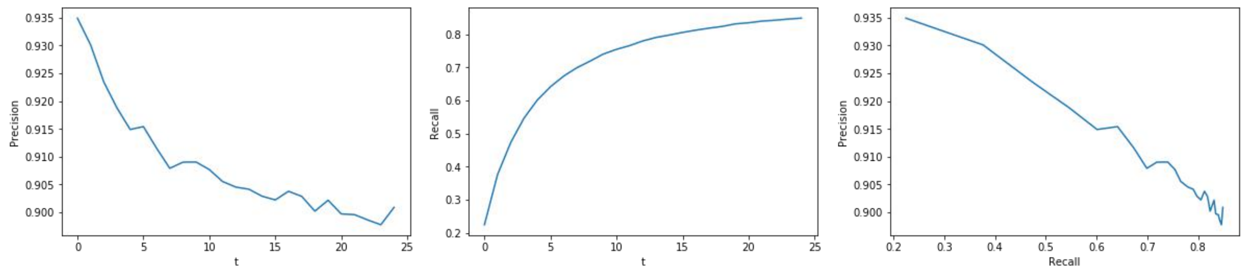


Figure 23. MF with bias-based collaborative filter predictions

Question 39 - Precision-recall curve of k-NN, NNMF, and MF with bias

We plotted the precision-recall curve with k-NN, NNMF, MF with bias. Basically, they all have higher precision but low recall in the small set of t. As the size of t becomes larger, the precision decreases a little but recall increases a lot. We observed that NNMF achieved higher precision but lower recall, with contrast to MF with bias, whose precision is higher but recall is lower. Overall there are not significant difference among all of them.

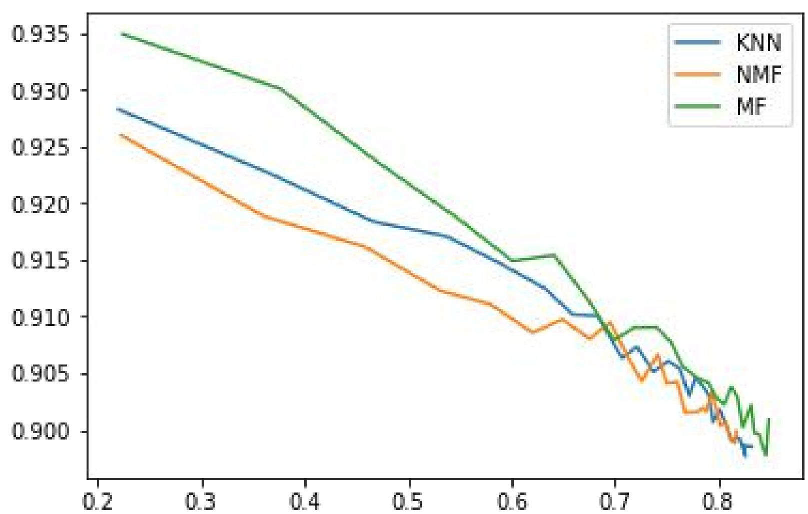


Figure 24. Precision-recall curve of k-NN, NNMF, and MF with bias