Project 3: Collaborative Filtering

Zeyu Bai, 904514671 Yi-Jui Chang, 804587644

Nan Liu, 004434221 Ruizhi Yang, 704514506

Winter 2019

1 Introduction

Recommender systems arise due to the increasing growth of electronic business, where customer behaviour can be used to predict their interests. This project applies both user-based and model-based methods on the "MovieLens" data set to implement recommender systems. The "MovieLens" data set contains the information of a sparse rating matrix between users and movies, and our task is to predict the missing ratings and finally provide a recommendation for each user.

First a user-based neighbourhood model using k-NN collaborative filter on Pearson-correlation coefficient is designed and tested using 10-fold cross validation to find out the optimal k. Then three differently trimmed test set are tested as well with average RMSE and MAE reported. Similar analysis are conducted with model-based collaborative filtering, concretely speaking latent factor based methods, using different matrix factorization and different cost functions(NNMF and MF with bias). To compare with these relatively complicated methods, a naive collaborative filter is also done. Based on our predictions, recommendations for each user naturally show up.

2 Collaborative filtering models

In this project, we will implement and analyze the performance of three types of collaborative filtering methods:

(i) Neighborhood-based collaborative filtering

(ii) Model-based collaborative filtering

(iii) Naive collaborative filtering

3 MovieLens data set

In this project, we consider the prediction of movie ratings in the MovieLens data set. The rating matrix which is denoted by R is a 610×9724 matrix containing 610 users and 9724 movies.

Question 1: Sparsity

$$\text{Sparsity} = \frac{\text{Available ratings}}{610 \times 9724} = 0.0169997$$

Question 2: Frequency of the rating values

Figure 1 shows the frequency of the rating values. The distribution is skewed left with values concentrated at the right tail. So the users tend to rate higher values ($3 \sim 5$) instead of lower values ($0 \sim 2.5$).

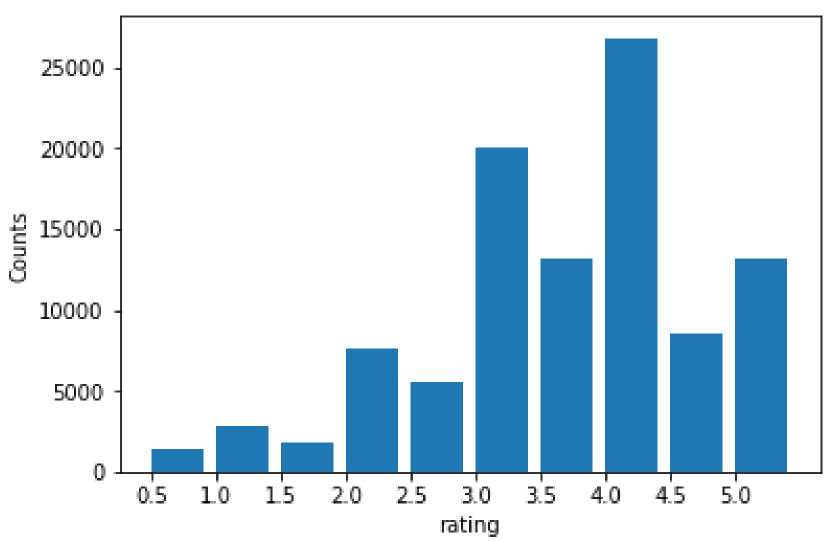


Figure 1: Frequency of the rating values

Question 3: Distribution of ratings among movies

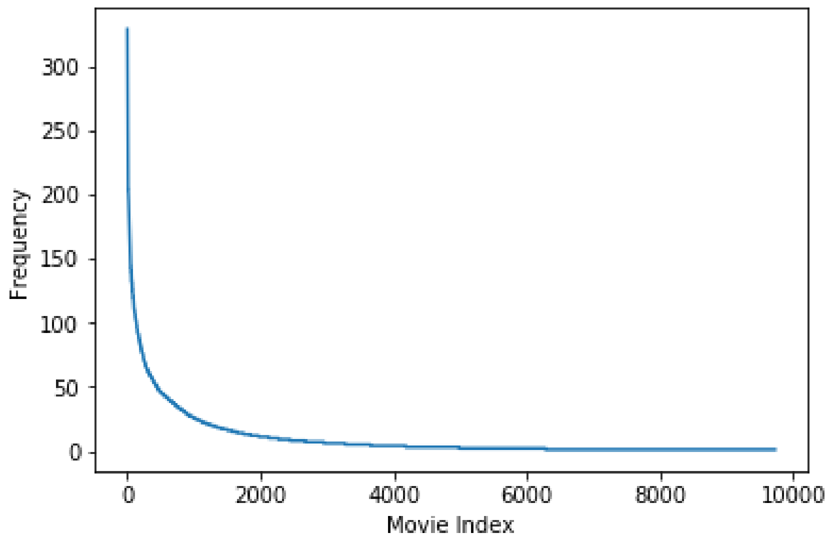


Figure 2: Distribution of ratings among movies

Question 4: Distribution of ratings among users

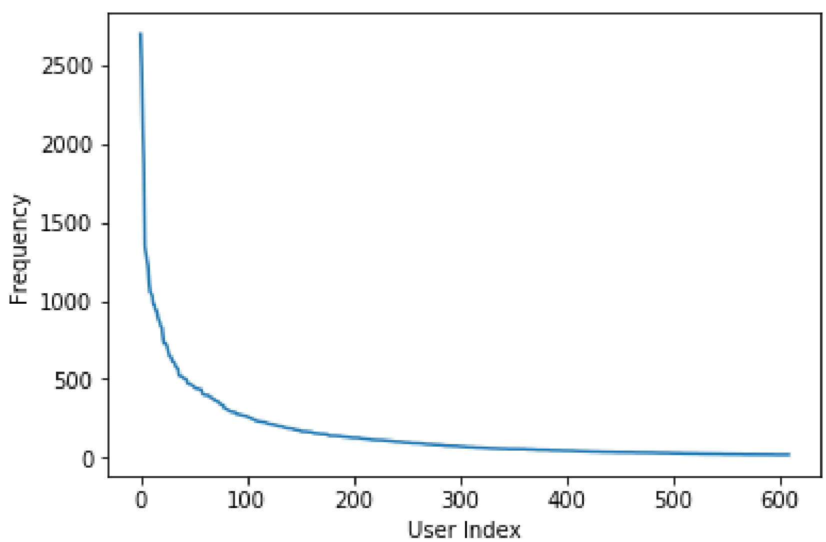


Figure 3: Distribution of ratings among users

Question 5:

The distribution of ratings among movies showed in figure 2 is highly skewed to the right. Only very few movies receive more than 50 ratings. The long right tail of the distribution indicates that most of the movies receive few ratings. The distribution of ratings among users in figure 3 has similar pattern. This sparsity propriety of the ratings matrix is the main challenge of our recommendation process.

Question 6: Distribution of variance of ratings by movies

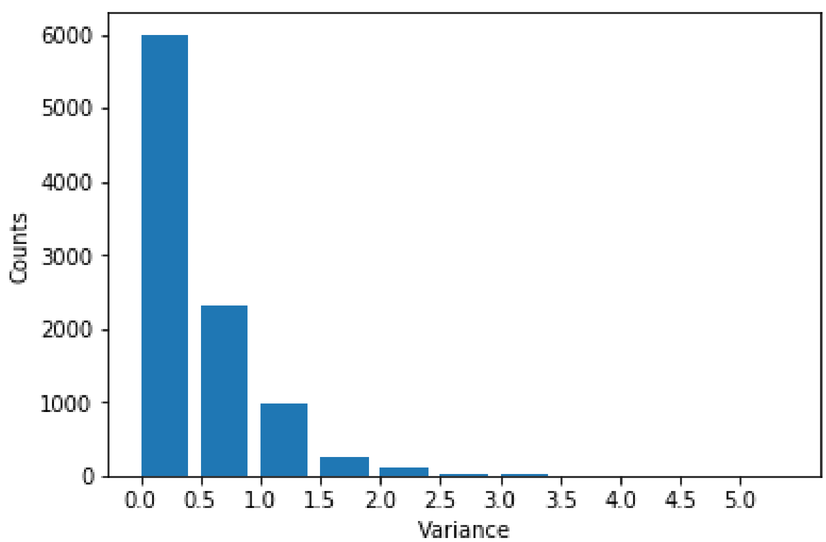


Figure 4: Distribution of variance of ratings by movies

Figure 4 shows that the distribution of the variance of ratings by movies is also skewed to the right. For most of the movies, the variation of ratings among users is very small. And only very few movies have rating variances larger than 2.5. This indicates that most of the users who rated the movies have similar preferences. Thus we can use this similarity property to predict the missing rating values .

4 Neighborhood-based collaborative filtering

Question 7: Mean rating for user u: μ_u

$$\mu_u = \frac{1}{|I_u|} \sum_{k \in I_u} r_{uk}$$

Question 8: The meaning of $I_u \cap I_v$

$I_u \cap I_v$ is the set of item indices for which ratings have been specified by both users u and v. Since the rating matrix R is sparse, $I_u \cap I_v = \emptyset$ is possible. Then there is no common item which is rated by both users.

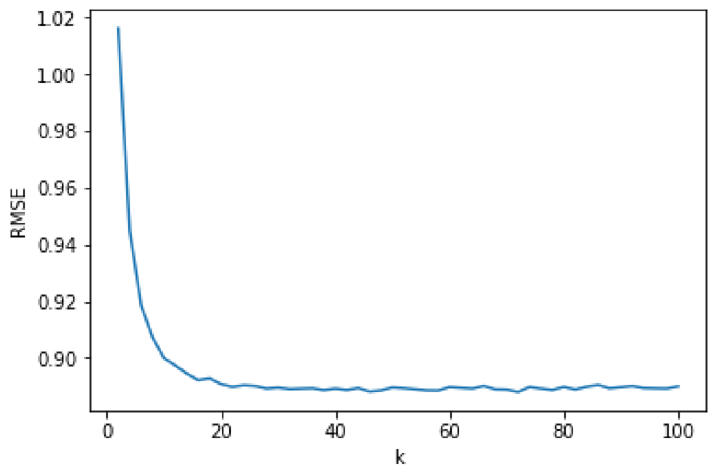
Question 9: Prediction function

$$\widehat{r}_{uj} = \mu_u + \frac{\sum_{v \in P_u} Pearson(u,v)(r_{vj} - \mu_v)}{\sum_{v \in P_u |Pearson(u,v)|}} \tag{1}$$

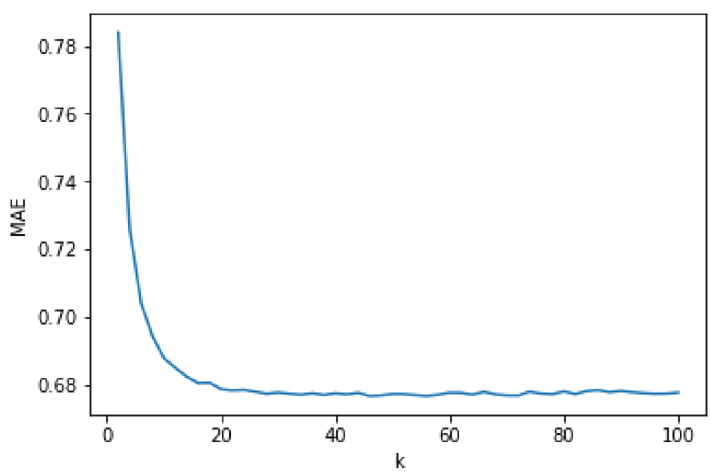
Mean-centering the raw rating of user v, ($r_{vj} - \mu_v$) will make the prediction function of user u ($\widehat{r}_{uj}$) be unaffected by the user v's rating preference. No matter user v rate all items highly or poorly, the demeaned ratings will be centered around zero.

Question 10: k-NN collaborative filter

In this question, we consider choosing the best number of neighbors, k, in a k-NN collaborative filter. We use 10-folded cross validation and sweep k from 2 to 100 with step 2. The performance of the prediction is evaluated by using the average of RMSE and MAE across all 10 folders.



(a) RMSE against k



(b) MAE against k

Figure 5: RMSE and MAE against number of neighbors k for k-NN

We plotted the RMSE and MAE against k in figure 5(a) and 5(b) respectively. The two plots have the same pattern which drop quickly when k increases at first but then become flat. The reason is that for small values of k, we only use a very small numbers of neighbors to do prediction and the RMSE or MAE should be large. As k increases, the RMSE and MAE drop quickly at first. But if we further increase the k, the Pearson-correlation coefficients for user u with the newly added neighbors become close to zero. From equation (1) we know the prediction function will not change too much after that. Thus the RMSE and MAE converge to a steady state and the two plots become flat.

Question 11: Best number of neighbor k

From the two plots in Question 10 we know, the 'minimum k' that would not result in a significant decrease in average RMSE or average MAE should both be 22. So we choose

$$k_{best} = 22$$

The steady state values of average RMSE and average MAE are 0.8899 and 0.6783 respectively.

Question 12: Popular movie trimming

In Questions 12-14, we perform k-NN collaborative filter in predicting the ratings of the movies in the trimmed test set. For Question 12, we trimmed the test set to include only the popular movies (movies that have received more than 2 ratings).

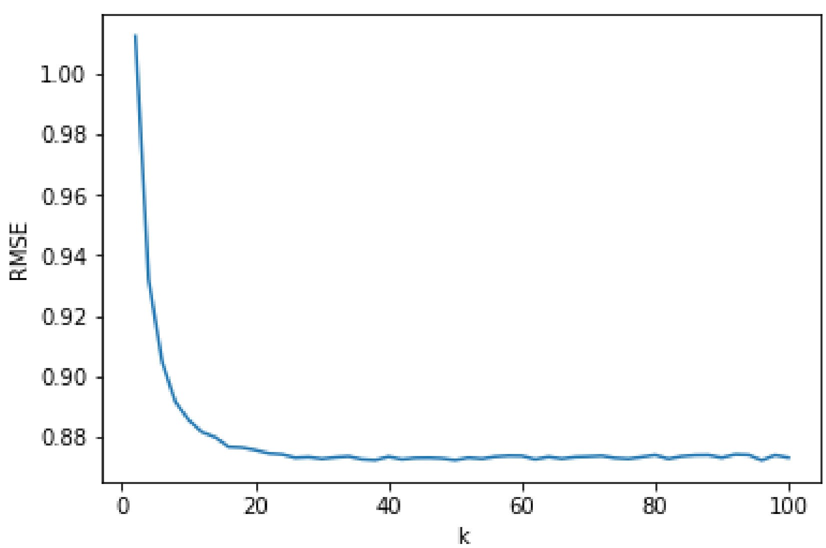


Figure 6: RMSE against number of neighbors k for popular movies

Figure 6 plotted RMSE against number of neighbors k for popular movies which has the same pattern as for all movies in figure 5(a). The 'minimum k' or k_{best} is 22 and the minimum average RMSE is 0.8746. We see that the popular trimmed data set performs better compared with the original data set.

Question 13: Unpopular movie trimming

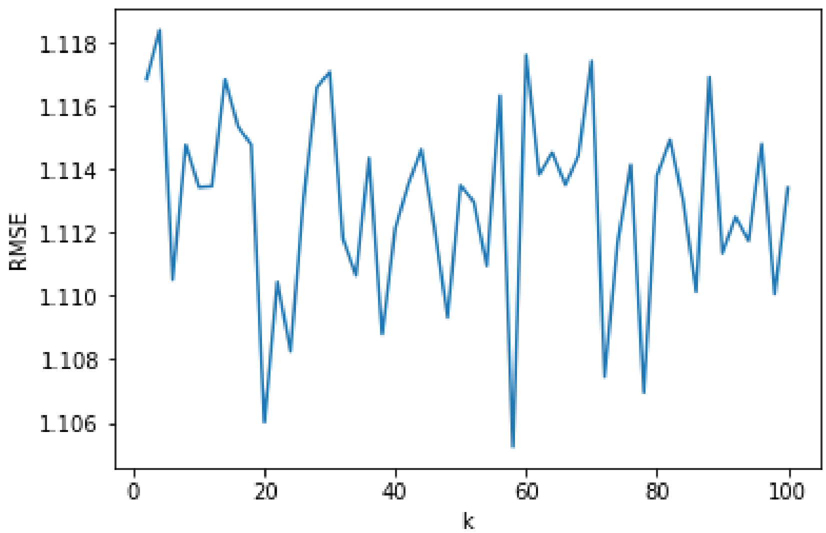


Figure 7: RMSE against number of neighbors k for unpopular movies

For the unpopular movie trimmed test set, we plotted the RMSE against number of neighbors k in figure 7. Under this case, the RMSE does not have a clear trend and fluctuates at around 1.112 when sweeping k from 2 to 100. The reason is that for unpopular movies, the rating matrix is so sparse that any user in the test set with observed ratings has at most 1 available ratings in his/her k nearest neighbors. Thus the prediction function in equation (1) actually mainly based on the user's mean ratings μ_u and the rating of its nearest neighbour no matter k is small or large.

Under this case, the 'minimum k' or k_{best} is 58 and the minimum average RMSE is 1.1052. But the k_{best} might not be useful due to the fluctuation of RMSE against k. And the minimum RMSE for the unpopular movies is much higher compared with the whole sample which is 0.8899. This is not surprise because of the almost naive prediction for the unpopular movie trimmed set.

Question 14: High variance movie trimming

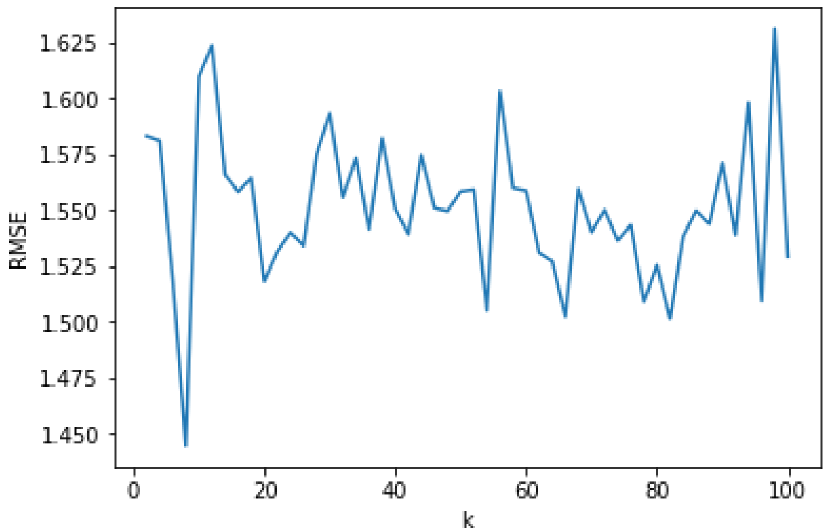
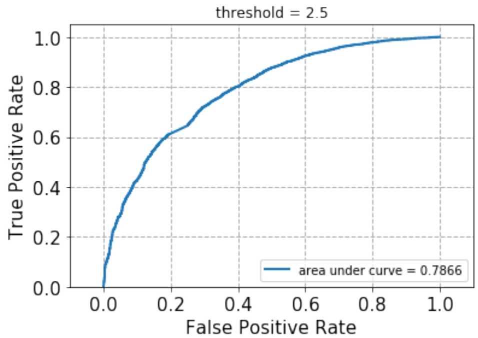


Figure 8: RMSE against number of neighbors k for high variance movies

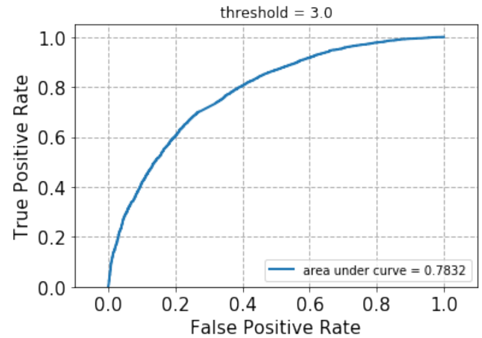
For the high variance movie trimmed test set, we plotted the RMSE against number of neighbors k in figure 7. Under this case, the RMSE also does not have a clear trend and fluctuates at around 1.550 when sweeping k from 2 to 100. The reason is that for high variance movies, the k nearest neighbors' ratings might not be referential for predicting user u's rating, i.e. the adjustment term, $\frac{\sum_{v \in P_u} Pearson(u,v)(r_{vj} - \mu_v)}{\sum_{v \in P_u} Pearson(u,v)}$, in the prediction function (1) might not be useful and even misleading for predicting the missing ratings. This will leads to the fluctuation of RMSE against k.

The 'minimum k' or k_{best} is 8 and the minimum average RMSE is 1.4447. The k_{best} under this case also might not be useful due to the fluctuation of RMSE against k.

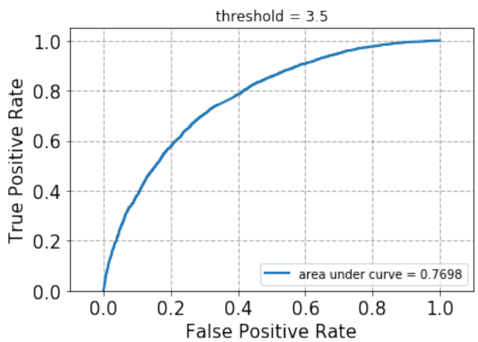
Question 15: ROC curves for the k-NN collaborative filter



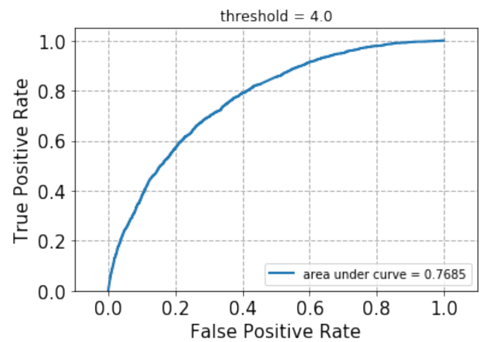
(a)



(b)



(c)



(d)

Figure 9: ROC curves for the k-NN collaborative filter

We use the ROC curve to evaluate the performance of the k-NN collaborative filter with $k_{best} = 22$. Since the observed rating values is continuous (0-5) and ROC curve can not be plotted directly. So we first classify the ratings into binary scale with threshold values [2.5, 3, 3.5, 4]. And then the four ROC curves are plotted in figure 9.

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

Table 1: AUC for the ROC curves

Threshold	2.5	3	3.5	4
AUC values	0.7866	0.7832	0.7698	0.7685

From table 1 we know, the AUC value is highest for threshold= 2.5 and decreases as we increase the threshold from 2.5 to 4. This phenomenon might be explained by the distribution of the rating values in figure 1. Figure 1 shows that the rating values are concentrated at values $3, 3.5$ and 4. If we choose thresholds at these values, true rankings that are very close to each other will be divided into different ground truth label groups. Thus it is more likely to obtain lower true positive rate. Thus the AUC values are all lower compared with threshold= 2.5.

5 Model-based collaborative filtering

Question 16: Convexity of optimization problem of NNMF

$$\begin{array}{ll} \min\limits_{U,V} & \sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} W_{ij}\big(r_{ij} - (UV^T)_{ij}\big)^2 + \lambda\|U\|_F^2 + \lambda\|V\|_F^2 \\ s.t. & U \geq 0, V \geq 0 \end{array}$$

To check the convexity of the optimization problem, we only need to check the convexity of the function

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

For simplicity, consider $m = n = k = 1$, then R, U and V are all 1×1 scalars. Assume $W = 1$. Then

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

Calculate the Gradient and Hessian matrix of L w.r.t. U, V:

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

The determinant of the Hessian matrix is

$$\begin{aligned} |\nabla^2 L(U,V)| &= U^2V^2 - (-R+2UV)^2 \\ &= -(R-UV)(R-3UV) \end{aligned}$$

which is not all positive for values of $R, U, V > 0$. Thus the Hessian matrix is not positive semi-definite and the optimization problem (2) is non-convex. And we can extend this simple example and show that in general the optimization problem for NNMF is non-convex.

Next, we need to show the optimization problem (2) can be solved by alternating least square(ALS) method. First introduce some notations. For each $j = 1, ..., n$, Denote

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

Consider if U is fixed, then the L function in (2) becomes

$$L(V) = \frac{1}{2}\sum_{j=1}^{n}\left(r_j - UV_j\right)^T W_j\left(r_j - UV_j\right) \tag{3}$$

By minimizing the loss function $L(V)$ we have the solutions for the movie factors V:

$$V_j = (U^T W_j U)^{-1} U^T W_j r_j; \quad (j = 1, ..., n) \tag{4}$$

which is the formula for the (Weighted) Least Square estimator.

Similarly, if V is fixed, then we have the solutions for the user factors U:

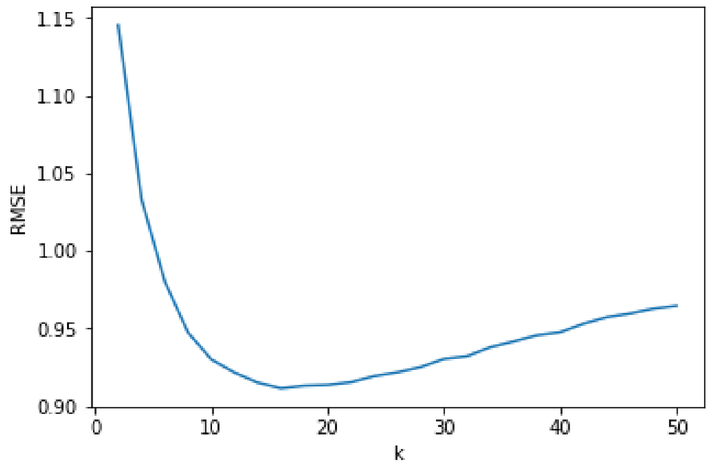
$$U_i = (V^T W_i V)^{-1} V^T W_i r_i; \quad (j = 1, ..., n) \tag{5}$$

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

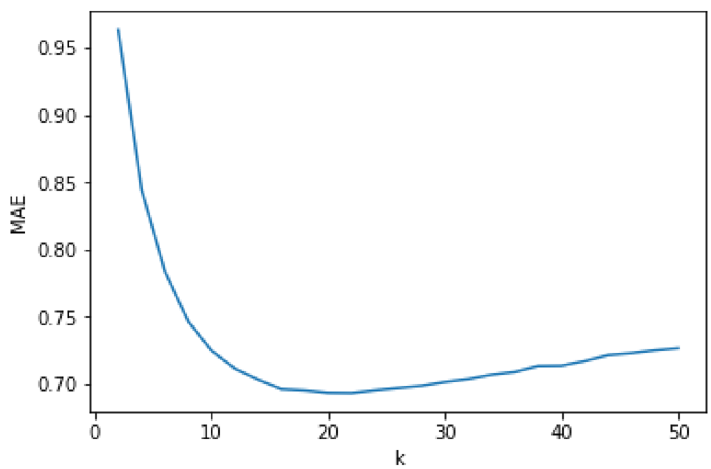
This property of the NNMF problem allow us to use ALS as the optimization algorithms. The main idea is first to keep U fixed and then solve for V. In the next stage, keep V fixed and solve for U. At each stage we are solving a least-squares problem which makes this algorithm to be stable and has a faster convergence rate.

Question 17: NNMF-based collaborative filter

In this question, we consider choosing 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.



(a) RMSE against k



(b) MAE against k

Figure 10: RMSE and MAE against number of latent factors k for NNMF

We plotted the RMSE and MAE against k in figure 10(a) and 10(b) respectively. As k increases, RMSE and MAE decrease at first and then increase after the turning point. Take RMSE as an example, roughly speaking

$$\text{RMSE}=\text{Estimation error}^2 + \text{Prediction variance}$$

For small values of k, the model underfits as it is not complex enough to capture the features in R. Thus 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. So the RMSE will increase after some turning point.

Question 18: 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 RSME, we have $k_{opt} = 16, \quad RMSE_{min} = 0.9115$

If we use minimize MAE, we have $k_{opt} = 22, \quad MAE_{min} = 0.6932$

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.

Question 19: Popular movie trimming

In Questions 19-21, we performance of the NNMF collaborative filter in predicting the ratings of the movies in the trimmed test set. For Question 12, we trimmed the test set to include only the popular movies (movies that have received more than 2 ratings).

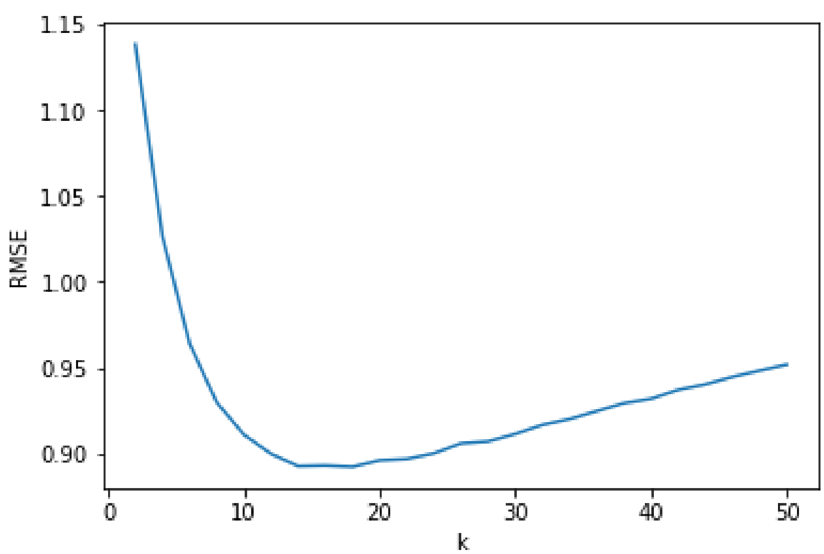


Figure 11: RMSE against number of latent factors k for popular movies

Figure 11 plotted RMSE against number of latent factors k for popular movies which has the same pattern as for all movies in figure 10(a).

The optimal number of latent factors is $k_{opt} = 18$ and correspondingly the minimum average RMSE is $RMSE_{min} = 0.8922$. We find the minimum average RMSE is lower than the original data set and the reason is the same as Question 11.

Question 20: Unpopular movie trimming

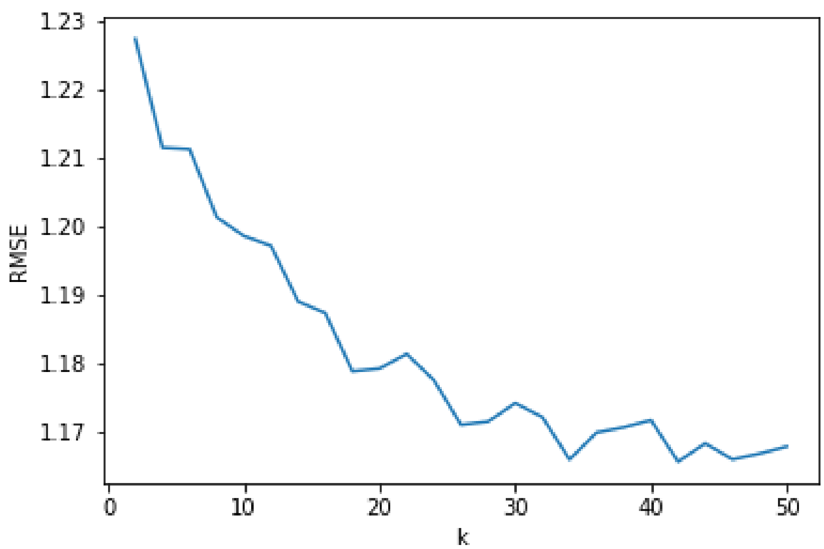


Figure 12: RMSE against number of latent factors k for unpopular movies

For the unpopular movie trimmed test set, we plotted the RMSE against number of latent factors k in figure 12. Under this case, the RMSE drops at first and then becomes flat when sweeping k from 2 to 50. Also there is small fluctuations along the trend.

The optimal number of latent factors is $k_{opt} = 42$ and correspondingly the minimum average RMSE is $RMSE_{min} = 1.1655$.

Question 21: High variance movie trimming

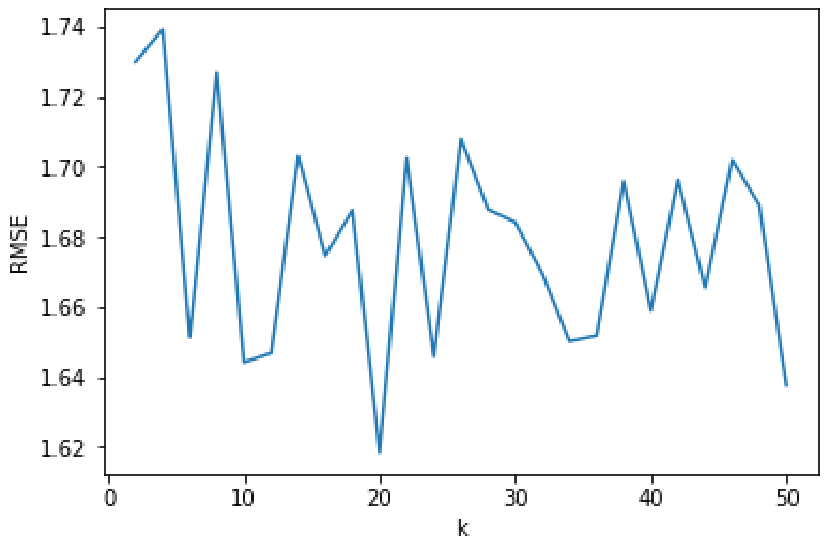
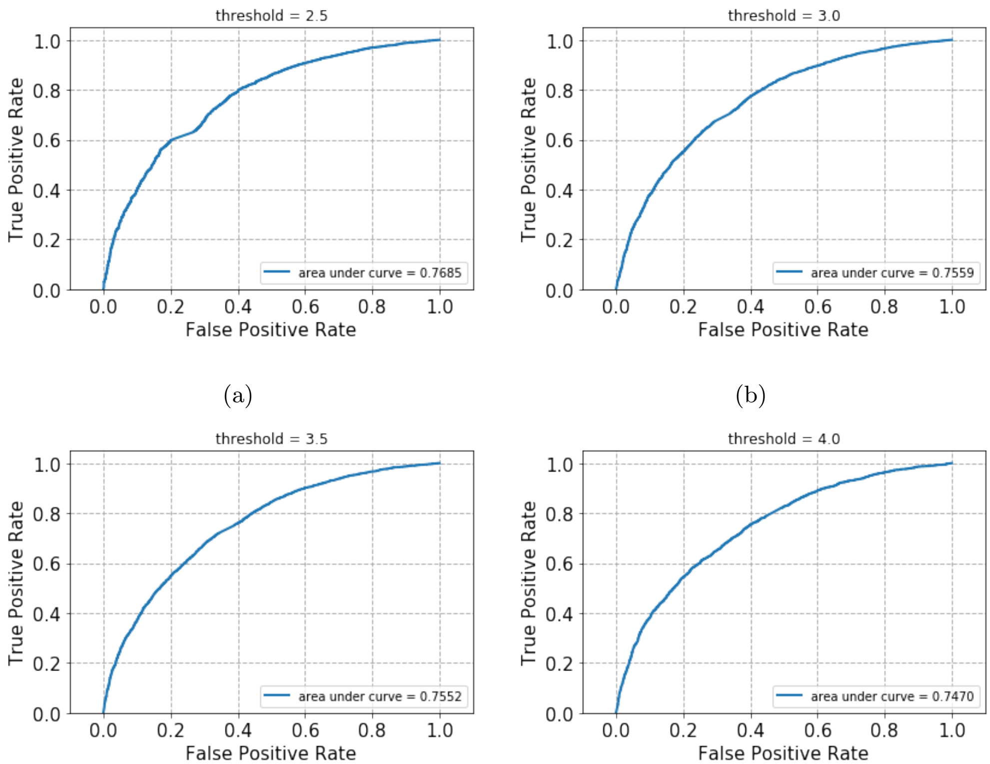


Figure 13: RMSE against number of latent factors k for high variance movies

For the high variance movie trimmed test set, we plotted the RMSE against number of latent factors k in figure 13. Under this case, the RMSE does not have a clear trend and fluctuates at a high value of 1.68 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 poor no matter what the k is chosen.

The optimal number of latent factors is $k_{opt} = 20$ and correspondingly the minimum average RMSE is $RMSE_{min} = 1.6185$.

Question 22: ROC curves for the NNMF collaborative filter



(a) (b) (c) (d)

Figure 14: ROC curves for the NNMF collaborative filter

We use the ROC curve to evaluate the performance of the NNMF collaborative filter with $k_{opt} = 16$. Similar as 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 figure 14.

For each of the ROC curve, we summarized the area under the curve (AUC) values in the following table 2:

Table 2: AUC for the ROC curves

Threshold	2.5	3	3.5	4
AUC values	0.7685	0.7559	0.7552	0.7470

Same as the k-NN case in Question 15, the AUC value is highest for threshold= 2.5 and decreases as we increase the threshold from 2.5 to 4. The reason is the same with what we explained in Question 15.

Question 23: Interpretability of NNMF

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 table 3.

Table 3: Genres of Top 10 movies in column 1 and 3

Top 10 movies in column 1	Top 10 movies in column 3
Crime—Thriller	Adventure—Animation—Children—Comedy—Fant
Drama—Horror—Mystery—Thriller	Action—Comedy
Comedy—Drama—Romance	Horror
Drama	Comedy—Drama—Romance
Drama—Horror—Thriller	Comedy—Drama
Drama	Horror
Action—Crime—Thriller	Drama—Sci-Fi
Drama	Comedy—Drama—Romance—War
Crime—Drama—Thriller	Drama
Drama—Romance	Children—Comedy—Fantasy—Horror

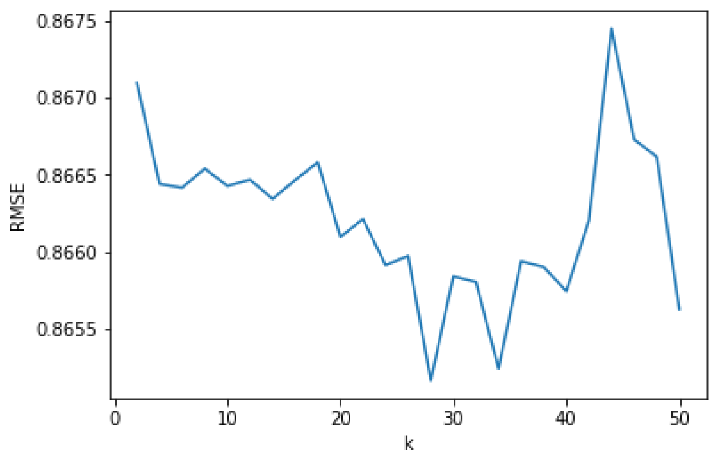
From table 3 we know the top 10 movies in each column do not belong to a particular genre but a small collection of genres. For example, we can conclude the column 1 represents the Drama or Thriller movies. And column 3 should be Comedy or Horror movies.

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 few collection of movie genres.

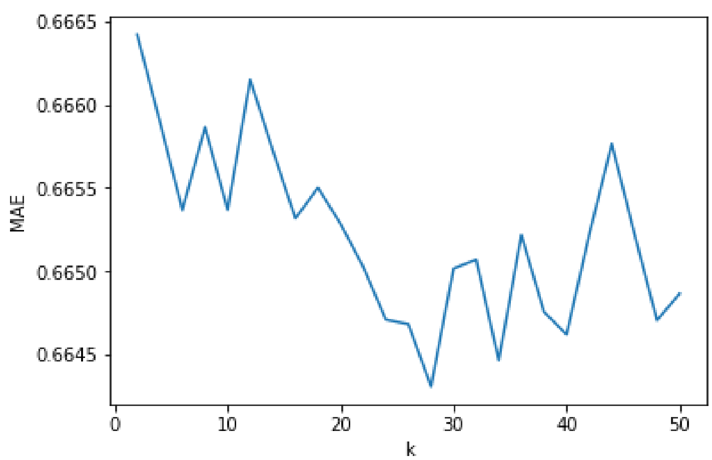
Question 24: MF-based collaborative filter with bias

In this question, we modify the MF by adding biased terms for both user (b_u), and movie (b_i). 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 term of both lower RMSE and lower MAE.



(a) RMSE against k



(b) MAE against k

Figure 15: RMSE and MAE against number of latent factors k for MF

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 $k_{opt} = 28, \quad RMSE_{min} = 0.8652$

If we use minimize MAE, we have $k_{opt} = 28, \quad MAE_{min} = 0.6643$.

Question 26: Popular movie trimming

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. For Question 26, we trimmed the test set to include only the popular movies (movies that have received more than 2 ratings).

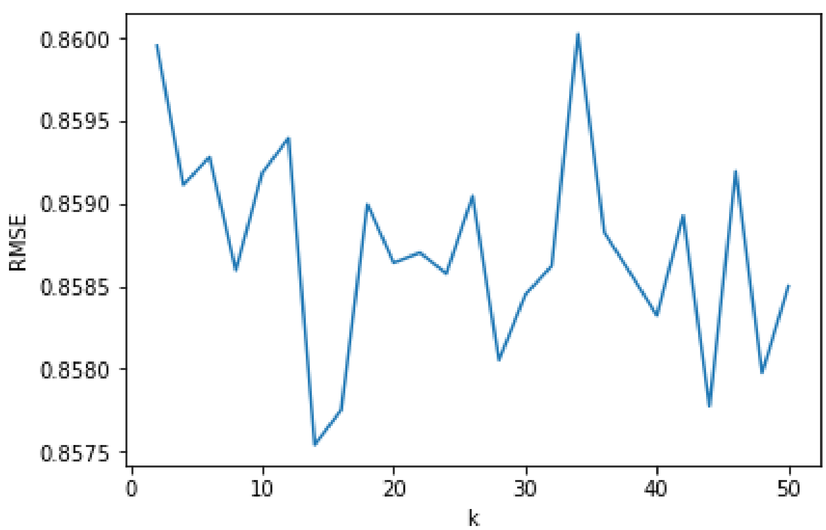


Figure 16: RMSE against number of latent factors k for popular movies

Figure 16 plotted RMSE against number of latent factors k for popular movies.

The optimal number of latent factors is $k_{opt} = 14$ and correspondingly the minimum average RMSE is $RMSE_{min} = 0.8575$.

Question 27:Unpopular movie trimming

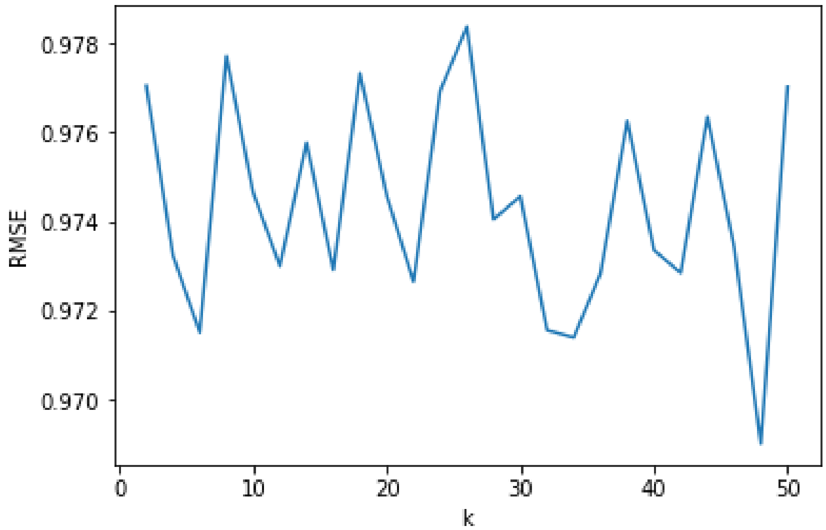


Figure 17: RMSE against number of latent factors k for unpopular movies

Figure 17 plotted RMSE against number of latent factors k for popular movies.

The optimal number of latent factors is $k_{opt} = 48$ and correspondingly the minimum average RMSE is $RMSE_{min} = 0.9690$.

Question 28:High variance movie trimming

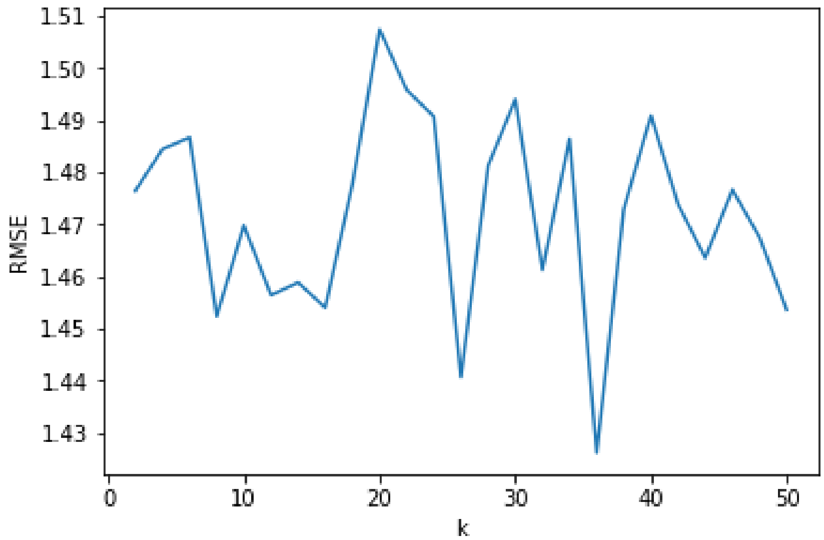


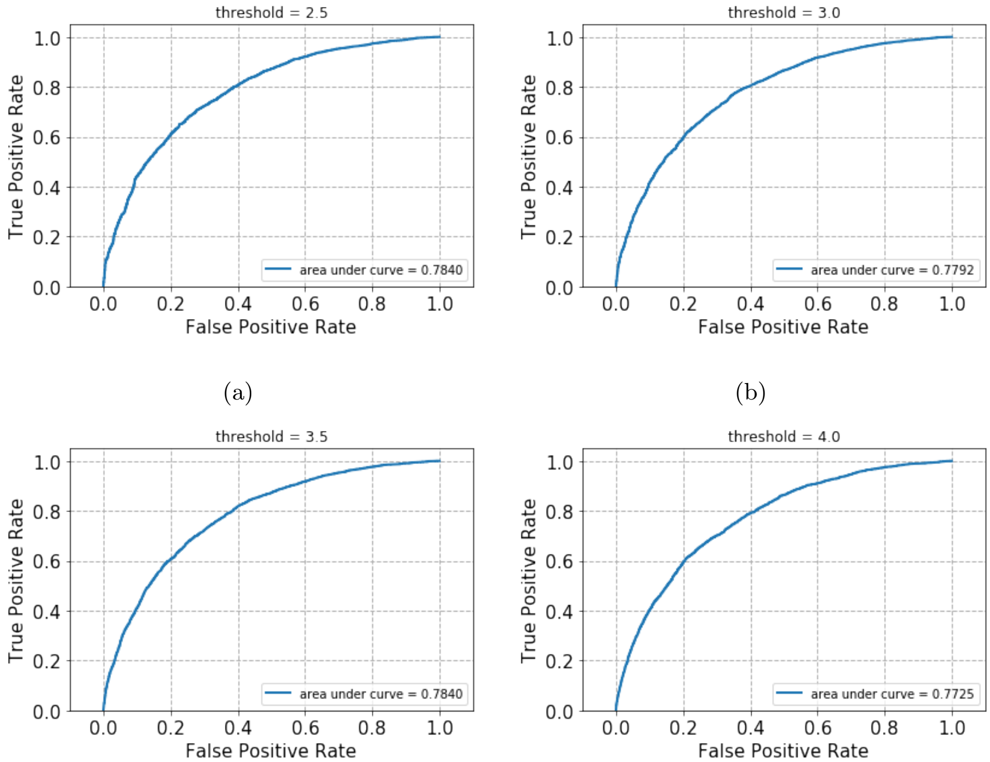
Figure 18: RMSE against number of latent factors k for high variance movies

Figure 18 plotted RMSE against number of latent factors k for popular movies.

The optimal number of latent factors is $k_{opt} = 36$ and correspondingly the minimum average RMSE is $RMSE_{min} = 1.4262$.

Question 29: ROC curves for the MF with bias

For each of the threshold value [2.5; 3; 3.5; 4]. we plotted the ROC curves in figure 19



(a) (b)

(c) (d)

Figure 19: ROC curves for the MF collaborative filter

We summarized the area under the curve (AUC) values in the following table 4.

Table 4: AUC for the ROC curves

Threshold	2.5	3	3.5	4
AUC values	0.7840	0.7792	0.7840	0.7725

6 Naive collaborative filtering

In this section, we implement a naive collaborative filter to predict the ratings which directly use the mean rating of the user.

$$\hat{r}_{uj} = \mu_u \tag{6}$$

Question 30: Naive collaborative filtering

We use 10-fold cross validation to evaluate the performance of the naive collaborative filter. The average RMSE across all 10 folds is

$$RMSE = 0.9347$$

The RMSE for the Naive collaborative filtering is the largest among the four filtering methods which is not surprise.

Question 31: Popular movie trimming

$$RMSE_{pop} = 0.9323$$

The RMSE for the popular movie trimmed set is slightly lower than the original dataset. The reason is that for popular movies each user has more avaiable ratings and the prediction using mean ratings is relatively reliable.

Question 32: Unpopular movie trimming

$$RMSE_{unpop} = 0.9711$$

In contrast, the RMSE for the unpopular movie trimmed set is higher than the original data set. It is due to the very few available ratings that can be used to calculate the mean ratings which make the naive prediction even worse.

Question 33: High variance movie trimming

$$RMSE_{high-var} = 1.4536$$

For high variance movie trimmed set, the correlations across users and movies are very low. And collaborative filtering methods which are based on the correlations should perform very poor.

7 Performance comparison

Question 34: ROC curves for the k-NN, NNMF, and MF with bias based collaborative filters

Here, we plotted the ROC curve of three filters and showed their own AUC. In fact, there are not much difference between them but the AUC of MF filter is a little larger than others.

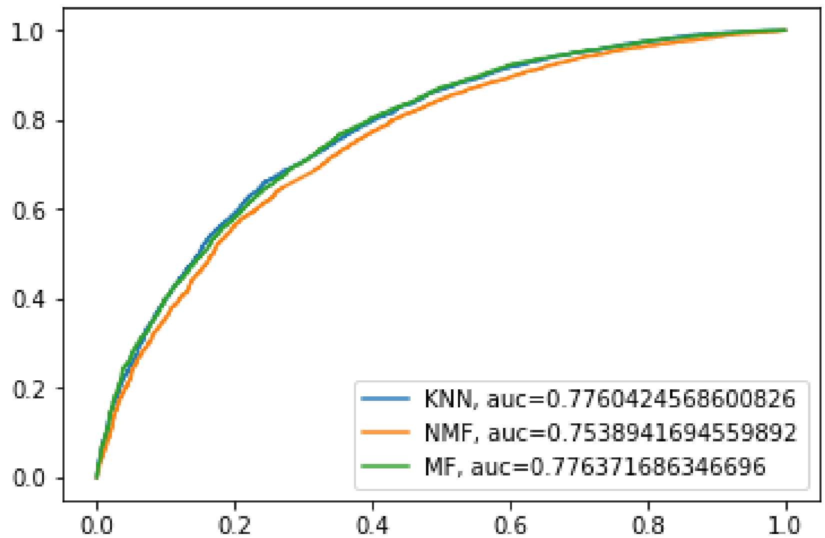


Figure 20

8 Ranking

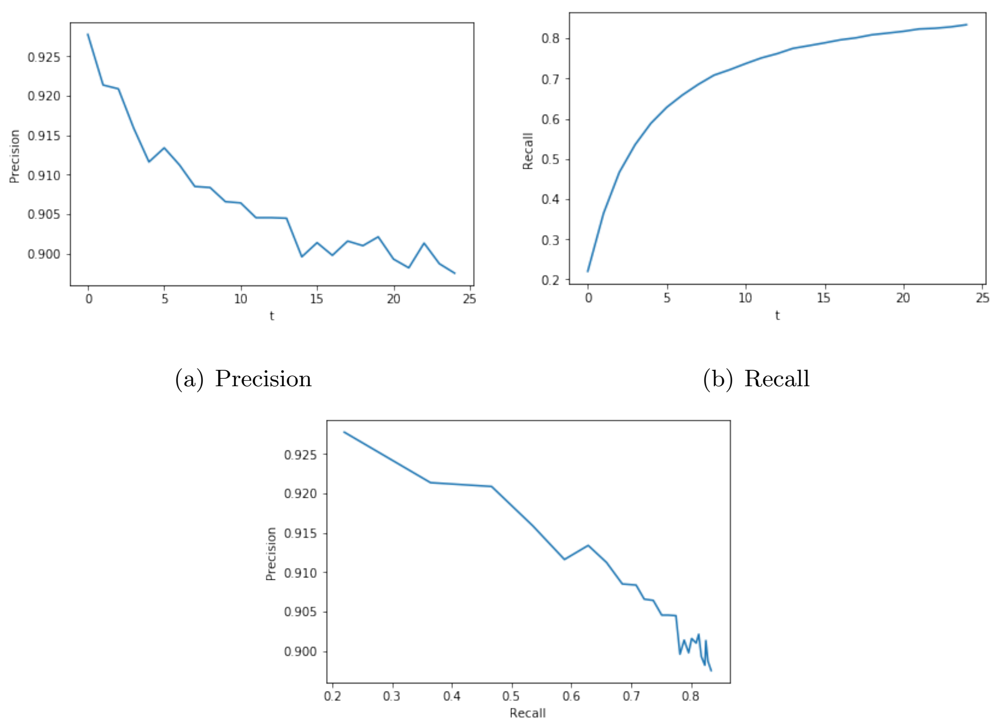
Question 35: Meaning of Precision and Recall

$$\text{Precision}(t) = \frac{|S(t) \cap G|}{|S(t)|}$$
$$\text{Recall}(t) = \frac{|S(t) \cap G|}{G}$$

Precision is defined as the intersection between condition positive and predicted positive divided by the set of predicted positive. It evaluates the correctness of percentages in recommendation system that is liked by the users. On the other hand, Recall is defined as the intersection between condition positive and predicted positive divided by the set of condition positive. 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

As t sweep from 1 to 25, the movies recommended to the users with top t predictions make the set $S(t)$ larger. The ground truth labels are defined by the movies that a user rated by 3 points or higher. Based on the assumption that our model would roughly correctly predict user's interest, precision should be high at first. Then with the increasing size of denominator $S(t)$, precision should go down, like figure 21(a). The recall curve is shown in figure 21(b). With the numerator $S(t)$ increase, recall increases as expected. In the trade off-between precision and recall in figure 21(c), we see that recall increases quickly with small drop of precision, which is the desired behaviour.



(a) Precision

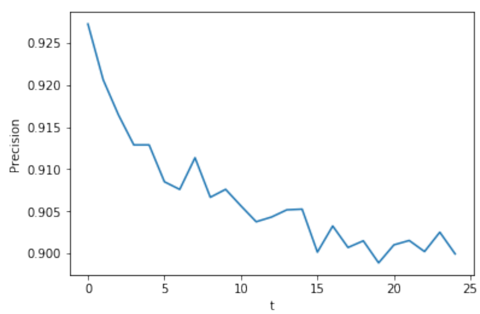
(b) Recall

(c) Precision-Recall curve

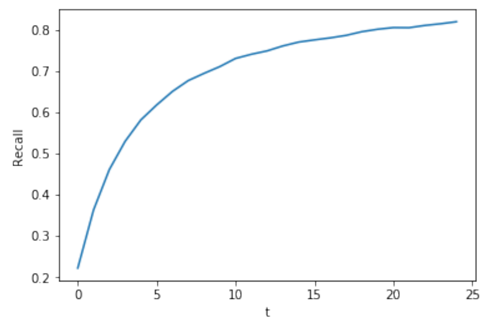
Figure 21: k-NN collaborative filter predictions

Question 37: NNMF-based collaborative filter predictions

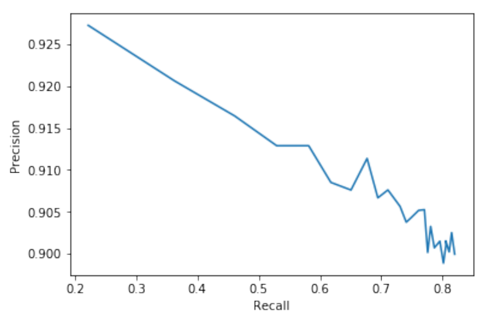
The plots in Question 37 are similar to that in Question 36. We plotted the precision curve in figure 22(a). The precision decreased a little bit when t increased, but the overall values are still greater than ninety percentages, which means the predictions made by the filter is indeed what users like. The recall curve is shown in figure 22(b). Its value increased as the t increased, which means the prediction of the filter is good to cover most of the item liked by the users in large number of t. The combination between precision and recall curve is shown in figure 22(c).The precision decreased as the recall increased.



(a) Precision



(b) Recall

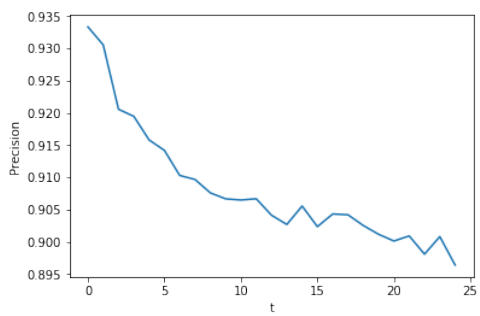


(c) Precision-Recall curve

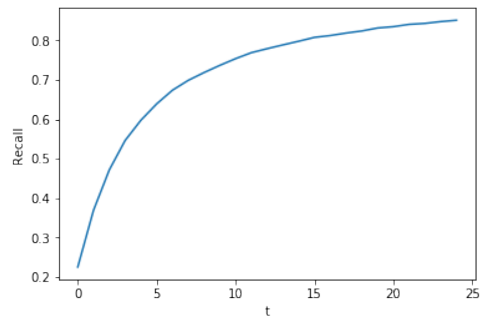
Figure 22: NNMF-based collaborative filter predictions

Question 38: MF with bias-based collaborative filter predictions

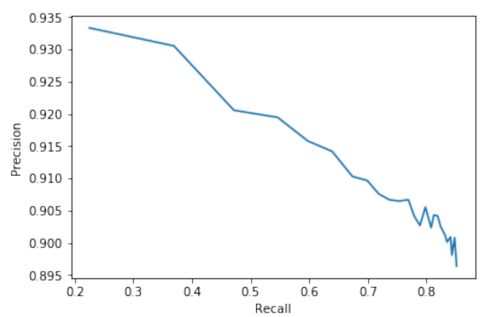
We plotted the precision curve in figure 23(a). The precision decreased a little bit when t increased, but the overall values are still greater than ninety percentages, which means the predictions made by the filter is indeed what users like. The recall curve is shown in figure 23(b). Its value increased as the t increased, which means the prediction of the filter is good to cover most of the item liked by the users in large number of t. The combination between precision and recall curve is shown in figure 23(c).The precision decreased as the recall increased.



(a) Precision



(b) Recall



(c) Precision-Recall curve

Figure 23: MF with bias-based collaborative filter predictions

Question 39: Precision recall curve

In the figure 24, 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.

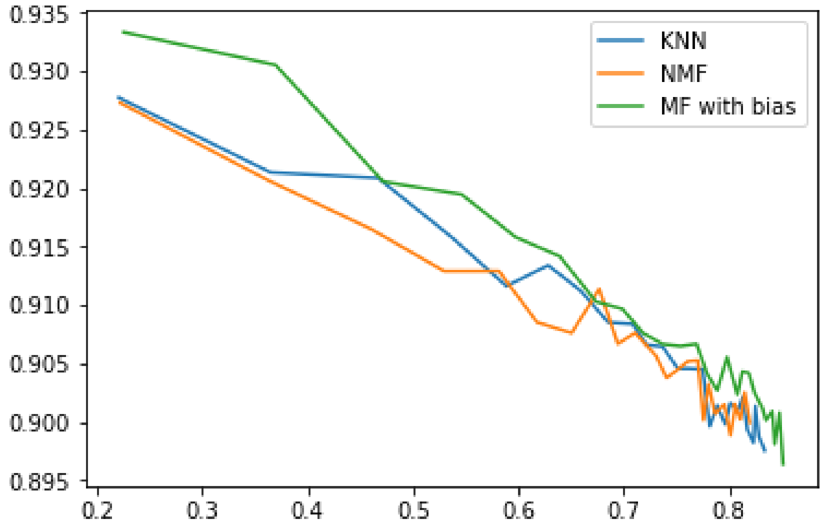


Figure 24: Precision-Recall curve