Problem 1

A matrix $R_{u\times v}$ where rows indicates users, columns indicates movies and value indicates rating, can be decomposed into the product of two low-dimensional matrices $R_{u\times v} = U^T_{k\times u}V_{k\times v}$. Here U indicates the users matrix and V indicates the movies matrix, u and v are the number of users and movies, k is a given number. We can assume that the probability of the rating follow a normal distribution $p(R|U,V) = \mathcal{N}(\hat{R},\sigma) = \mathcal{N}(U^TV,\sigma)$. Also users' behaviour and movies' quality can be assumed by following a normal distribution $p(U) = \mathcal{N}(0,\theta_u^2)$, $p(V) = \mathcal{N}(0,\theta_v^2)$. Thus we can derive that:

$$p(U,V|R) = p(U,V,R)/p(R) \propto p(U,V,R) = p(R|U,V)P(U)p(V) \tag{1}$$

Take the logarithm of the posterior probability, we can get:

$$\begin{aligned} f(U,V) &\triangleq -\log(p(U,V|R)) \\ &\propto -\log(p(R|U,V)) - \log(P(U)) - \log(P(V)) \\ &\propto \frac{||(R-U^TV)||_2}{2\theta^2} + \frac{U^TU}{2\theta_u^2} + \frac{V^TV}{2\theta_v^2} \\ &\propto \sum_{i=1}^{u}\sum_{j=1}^{v}(I_{ij}(r_{ij}-u_i^Tv_i)^2 + \lambda_u||u_i||^2 + \lambda_v||v_j||^2) \end{aligned} \tag{2}$$

where I_{ij} indicate that there is a rating from user i to movie j. We want to maximize $\log(p(U,V|R))$ in this problem which is equivalent to minimize $f(U,V)$. Let the gradient of $f(U,V)$ be 0, we have,

$$\begin{aligned} \frac{\partial f(U,V)}{\partial u_i} &= 2\sum_{i=1}^{u}\sum_{j=1}^{v}(I_{ij}(r_{ij}-u_i^Tv_i)v_{kj} + 2\lambda_u u_{ki} \qquad \forall i,k \\ &= 2(V_{I_i}V_{I_i}^T + \lambda_u E)u_i - 2V_{I_i}R^T(i,I_i) \qquad \forall i \\ &\triangleq 0 \end{aligned} \tag{3}$$

where I_i indicate that the movies that user i gave the rating. By solving the function, we can derive,

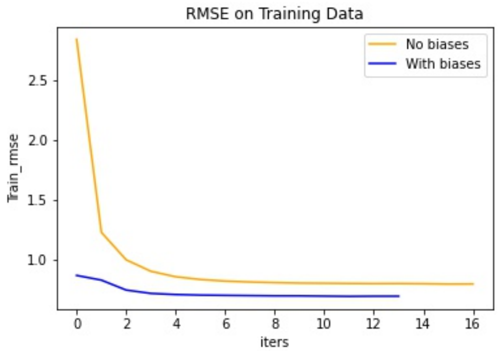
$$u_i = (V_{I_i}V_{I_i}^T + \lambda_u E)^{-1}V_{I_i}R^T(i,I_i) \tag{4}$$

Similarly, Let $\frac{\partial f(U,V)}{\partial v_j} = 0$, we can get,

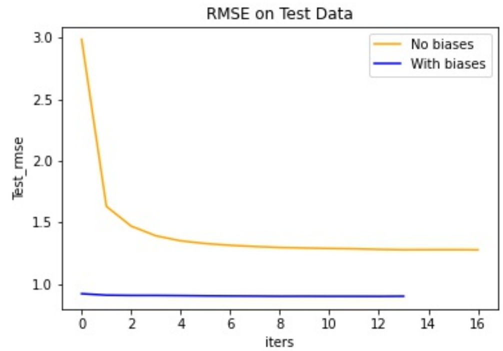
$$v_i = (U_{I_j}U_{I_j}^T + \lambda_v E)^{-1}U_{I_j}R(I_j,j) \tag{5}$$

By the analysis showed in (Y Zhou et al., 2008)[1], we know that the algorithm below can converge. Step 1: Initialize U and V. Step 2: Fix V, solve U by (4). Step 3: Fix U, solve V by (5). Step 4: Repeat step 2 and 3 until convergence.

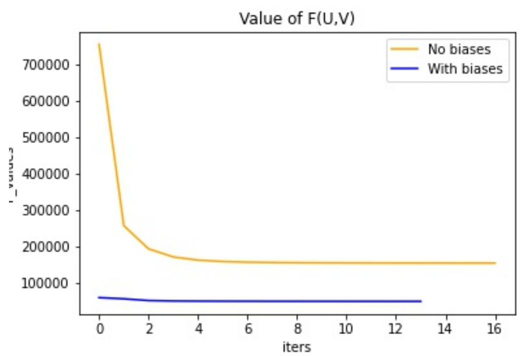
There are some biases in the model defined in (Koren et al., 2009)[2]. I also implemented the algorithm above on biases model and compared the results. I assumed $\lambda_u = \lambda_v = 10$ here, but I also tried different values and compared them later.



(a) Training RMSE



(b) Test RMSE



(c) F(U,V)

Figure 1: Results

The graphs aboves is the average result on each model for 10 different initialization, the resultes shows that both models converge very well. I let the algorithm run until the training RMSE no longer decrease. From the graph we can see that the model with biases performs better than the original model. It has a lower root of mean square error on both training data and test data. It also has a lower value on $f(U,V)$, which means this model's negative logarithm of the posterior probability is higher.Also, the biases model can converge quickly than original model. The final result is:

Metrics	F(U,V)	Train RMSE	Test RMSE
Original Model	153480	0.8011	1.2769
Bias Model	47990	0.6988	0.8987

Next, by calculating the distance between two movies on V, we can define closest movies. For example, I got the result below. The original movie is Avengers (2012). And the closest movies are mostly close to this movie because I have seen most of them. It can be seen that this model performs well in some cases.

```
Original movie: Avengers, The (2012), Genres: Action|Adventure|Sci-Fi|IMAX
1st close movie: Captain America: The Winter Soldier (2014), Genres: Action|Adventure|Sci-Fi|IMAX
2nd close movie: How to Train Your Dragon (2010), Genres: Adventure|Animation|Children|Fantasy|IMAX
3rd close movie: Rambo: First Blood Part II (1985), Genres: Action|Adventure|Thriller
4th close movie: Aladdin (1992), Genres: Adventure|Animation|Children|Comedy|Musical
5th close movie: Harry Potter and the Goblet of Fire (2005), Genres: Adventure|Fantasy|Thriller|IMAX
6th close movie: Guardians of the Galaxy (2014), Genres: Action|Adventure|Sci-Fi
7th close movie: X-Men: First Class (2011), Genres: Action|Adventure|Sci-Fi|Thriller|War
8th close movie: Mummy, The (1999), Genres: Action|Adventure|Comedy|Fantasy|Horror|Thriller
9th close movie: Wayne's World (1992), Genres: Comedy
10th close movie: Harry Potter and the Half-Blood Prince (2009), Genres: Adventure|Fantasy|Mystery|Romance|IMAX
11st close movie: Spider-Man (2002), Genres: Action|Adventure|Sci-Fi|Thriller
12nd close movie: Harry Potter and the Deathly Hallows: Part 1 (2010), Genres: Action|Adventure|Fantasy|IMAX
```

The Test RMSE on model with biases under different λ_u and λ_v is :

$\lambda_u \vert \lambda_v$	0.01	0.1	1	10	100	1000
0.01	1.0778	0.9878	0.9328	0.9255	0.9561	0.9191
0.1	1.0682	0.9913	0.9334	0.9251	0.9463	0.9002
1	1.0579	0.9761	0.9302	0.9260	0.9011	0.9352
10	1.0476	0.9603	0.9209	0.8987	0.9453	0.9707
100	0.9967	0.9439	0.9142	0.9218	0.9568	0.9814
1000	0.9759	0.9489	0.9589	0.9628	0.9942	1.0249

From the table, we can see that a higher λ or a lower λ will led to a high rmse on test data. Choose $\lambda_u = \lambda_v = 10$ is better.

References

[1] Large-scale Parallel Collaborative Filtering for the Netflix Prize (Y Zhou et al., 2008)
[2] Matrix factorization techniques for recommender systems (Koren et al., 2009)