Factorization-Based Data Modeling

Practical Work 3

Students : ZHAO Mengzi, ZHOU Juncheng

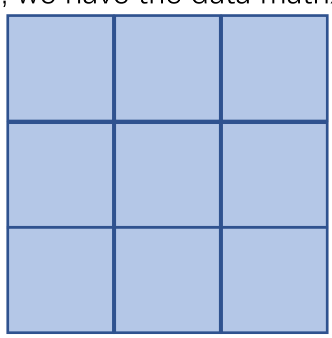
I – Introduction :

The problem that we aim to solve :

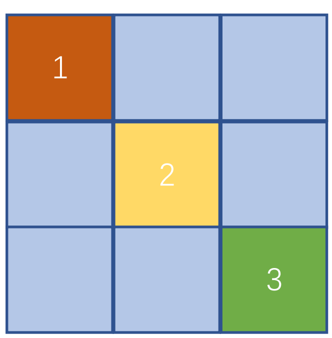
$$({Z_1}^*, {Z_2}^*) = arg \min_{Z_1, Z_2} \frac{1}{2} \|M \odot (X - Z_1 Z_2)\|^2$$

In general, we use two factor matrices to generate values of prediction which are approximate to the true value and we use the stochastic gradient descent to update these two matrices. But SGD can be processed by different threads in parallel, we can cut data matrix in several blocks and these blocks are independent, because these blocks share neither the same column nor the same row of the data matrix. According to the position of the data in the blocks of the data matrix, we can find the corresponding values in the 2 factor matrices and they can be updated in parallel, this is the algorithm of Distributed SGD.

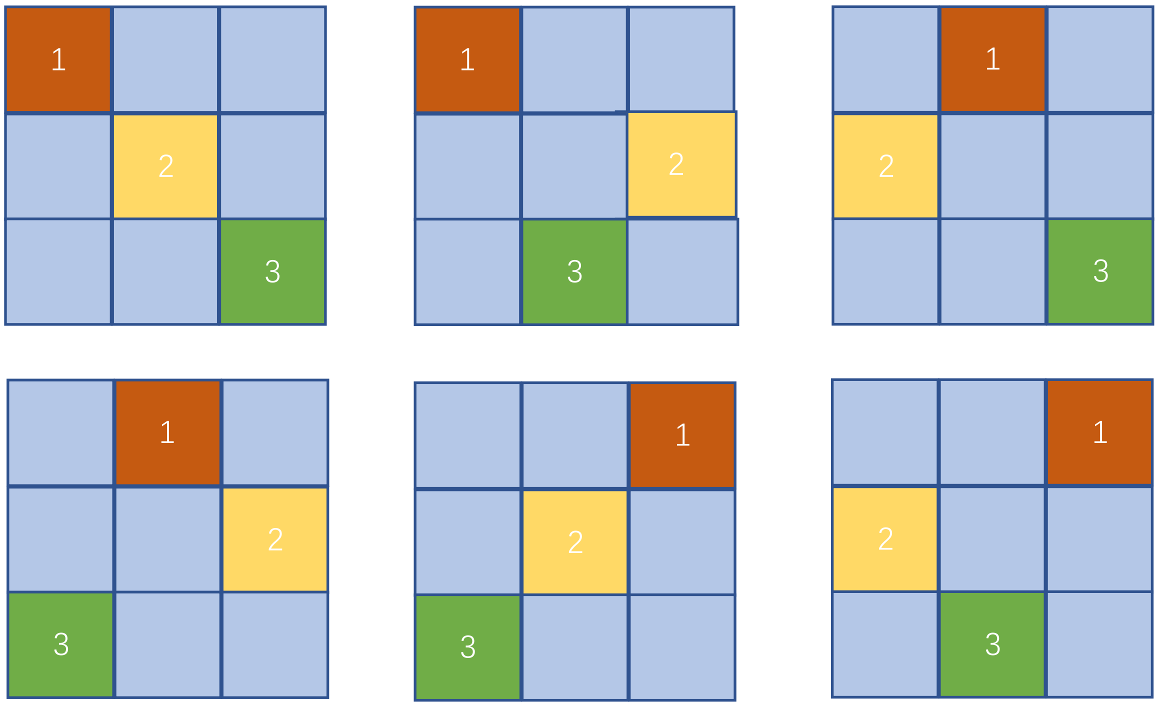
For example, at the beginning, we have the data matrix A, we cut it in 3 * 3 blocks :



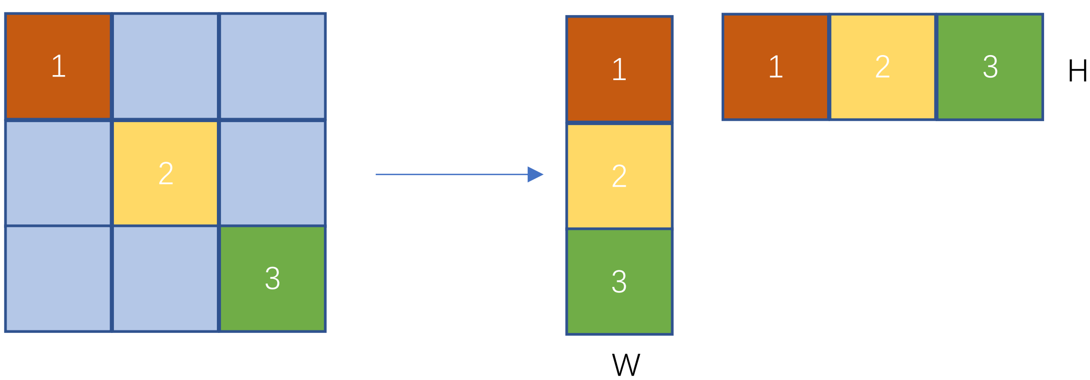
So we should have 3 threads who can take blocks randomly, for example, we have :

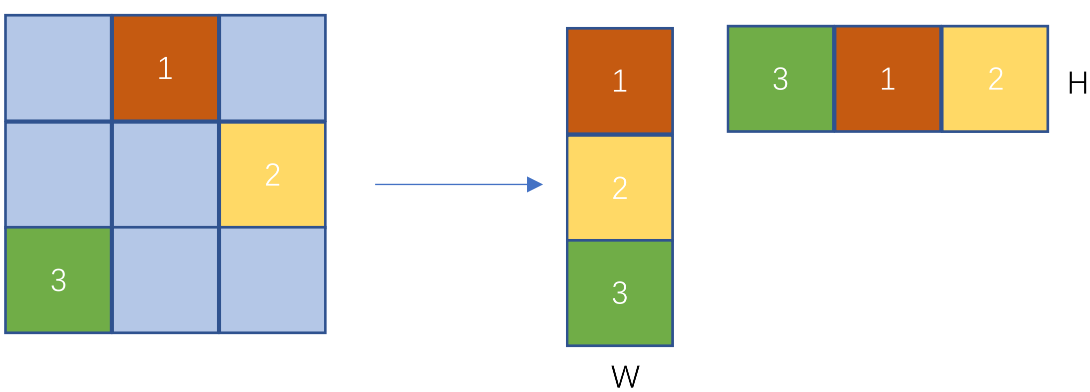


There are 6 possibilities :



We find the position of thread in the 2 factor matrices corresponding to the position of threads in data matrix, for example, we have :





We can notice that the position of threads in the matrix W never changes.

II – Exercices

2 – Set the rank to 10 and the step size to 0.00001, run the code for MovieLens 1 Million Dataset.

We run the command to compile :

mpicxx dsgd_mf_template.cpp -Wall -I/usr/local/include -L/usr/local/lib -lgsl -lgslcblas -lm -o dsgd_mf_template

We run the command to execute :

mpirun -hostfile host -n 4 ./dsgd_mf_template 3883 6040 10 20 0.00001 1

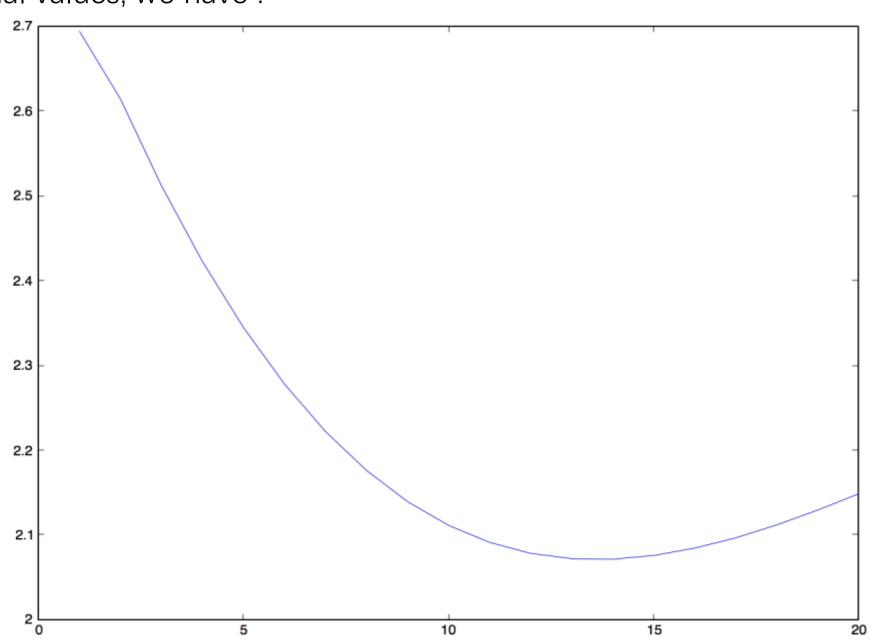
3 – Compute the RMSE by using the code compute_rmse.cpp and plot the RMSE in Matlab by using plot_rmse.m :

$$RMSE = \sqrt{\frac{\| M \odot (X - WH) \|_F^2}{N}}$$

N is the number of the observed entries.

The RMSE is to check if the function is converged or not, if the variation of value of RMSE is small, we can know that the function is converged.

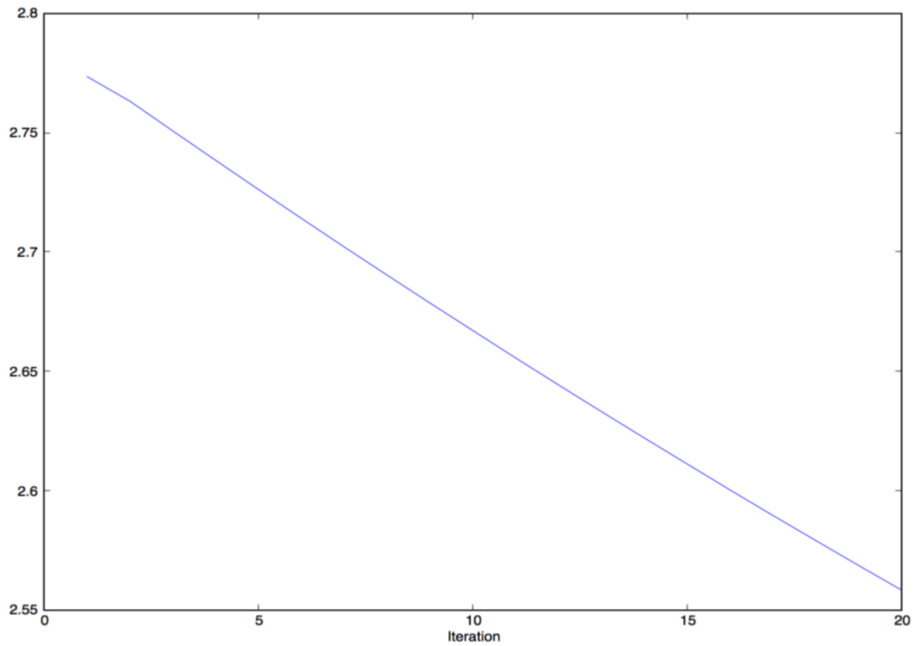
For orignal values, we have :



(original)

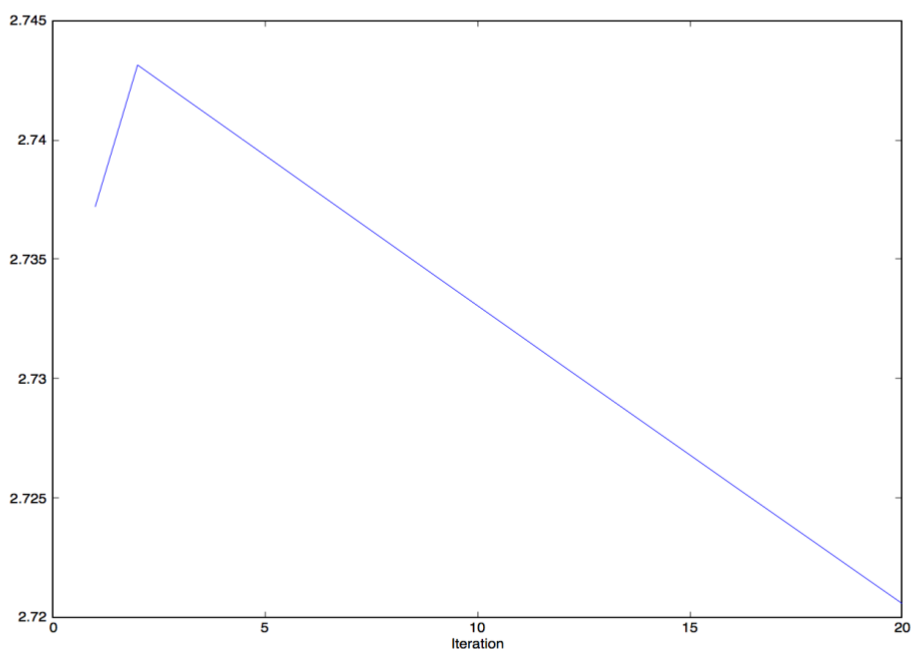
4 – Play with the rank and the step-size, what do you observe ?

k = 10, step-size = 0.000001



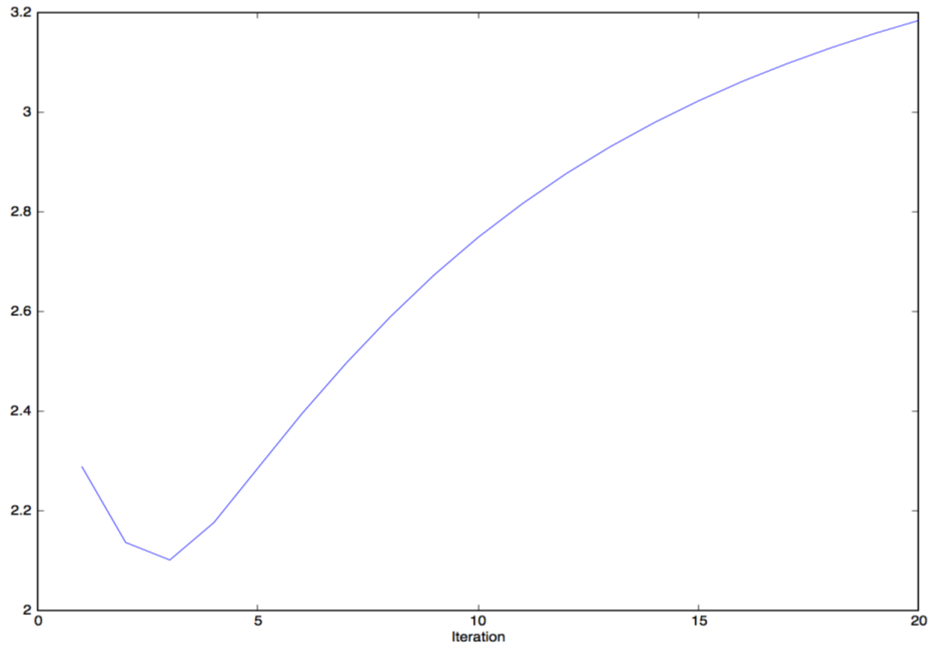
(1)

k = 10, step-size = 0.0000001



(2)

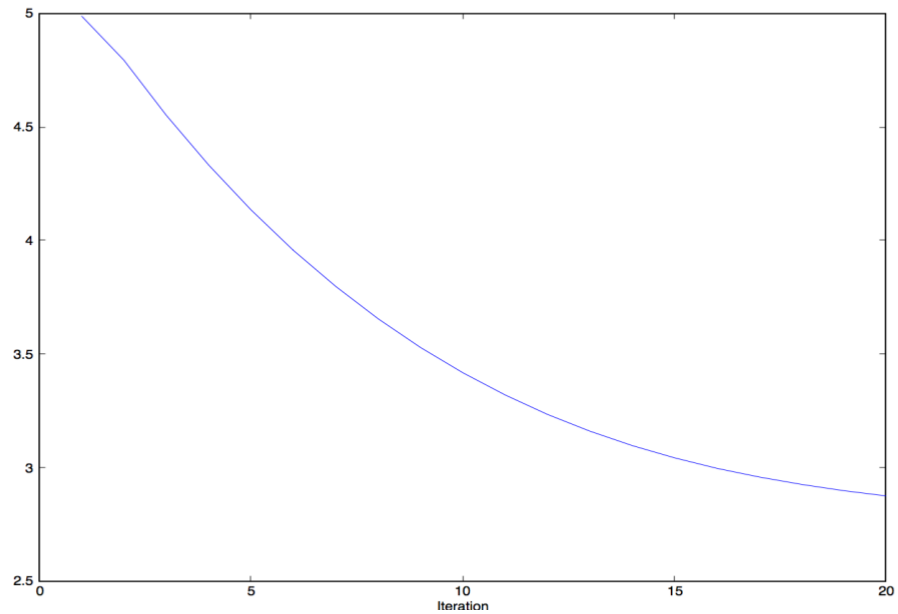
k = 10, step-size = 0.00005



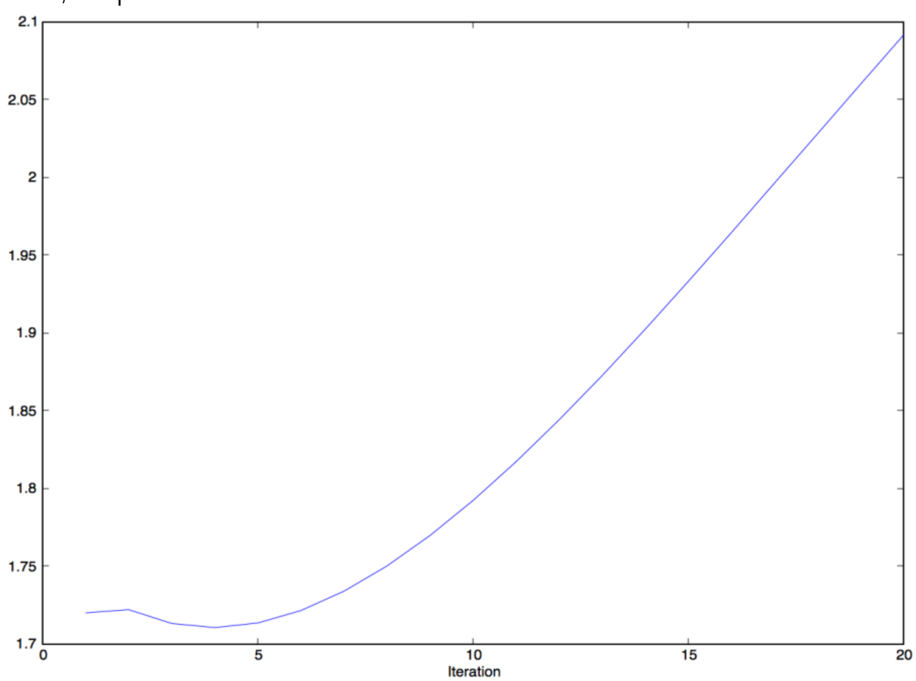
(3)

We can notice that when we have the original step-size = 0.00001, when the number of iteration is 14, the value of RMSE is the best, and when the number of iteration is 20, the value of RMSE is about 2.15. When we set the step size = 0.000001 and the step size = 0.0000001, we can notice that when the number of iteration arrives to 20, the results of these 2 graphs (1) (2) are worse than the original graph, but we can find these 2 graphs continue descending because the learning rate is much smaller than the original value of learning rate, so the speed of descending is slower than before. When we set the value of step size equal to 0. 00005 which is larger than the original value, we can notice that the lowest value of the graph (3) is corresponding to the iteration = 3, the result is about 2.15, so when the step-size becomes larger, the curve descends quickly, but when the iteration is 20, the value of RMSE is pretty high, this is overshooting the minimum, in this case, the value of step-size is too large.
So we can not be sure that we can obtain a better value when the step-size is larger or smaller, because when the step-size is larger, the curve can converge quickly, but may be it can miss the best value, when the step-size is too small, the curve converges too slowly, so when all of the iteration finish, the curve doesn't converge to the lowest value.

When k = 15, step-size = 0.00001 :



When k = 7, step-size = 0.00001 :



K is to do the reduction of dimension, k can represent the features of the matrix, when k is larger, it means we have more features, when we have a larger k, maybe we can improve the result, but when k is too large, it may cause overfitting, when k is too small, we have less features, so it may cause the RMSE becomes high.