South China University of Technology

# The Experiment Report of Machine Learning

**SCHOOL:** SCHOOL OF SOFTWARE ENGINEERING

**SUBJECT:** SOFTWARE ENGINEERIN
G


Author:
Feiqin Huang and Yuebo Guo
and Wanting Su

Supervisor:
Qingyao Wu

Student ID：
201530611678 and 201530611517
and 201530612767

Grade:
Undergraduate


December 9, 2017

# Recommender System Based on Matrix Decomposition

**Abstract—In this experiment we explore the construction of recommender system and understand the principle of matrix decomposition. Through the experiment we are more familiar to the use of gradient descent. In the end we realize a recommender system based on matrix decomposition under small-scale dataset, cultivating engineering ability.**

## I. Introduction

Recommender systems is a family of methods that enable filtering through large observation and information space in order to provide recommendations in the information space that user does not have any observation, where the information space is all of the available items that user could choose or select and observation space is what user experienced or observed so far.

The concept of matrix decomposition refers that in the mathematical discipline of linear algebra, a matrix decomposition or matrix factorization is a factorization of a matrix into a product of matrices. There are many different matrix decomposition; each finds use among a particular class of problems.

Alternate least squares optimization(often shortened to ALS) is a kind of collaborative filtering algorithm based on matrix factorization.Collaborative Filtering(CF) is a subset of algorithms that exploit other users and items along with their ratings(selection, purchase information could be also used) and target user history to recommend an item that target user does not have ratings for. Fundamental assumption behind this approach is that other users preference over the items could be used recommending an item to the user who did not see the item or purchase before.

Stochastic gradient descent (often shortened to SGD), also known as incremental gradient descent, is a stochastic approximation of the gradientdescentoptimization and iterative method for minimizing an objective function that is written as a sum of differentiable functions. In other words, SGD tries to find minima or maxima by iteration. In this experiment, we Use the stochastic gradient descent method to decompose the sparse user score matrix, get the user factor matrix and item factor matrix.

In this experiment, we implement recommendation system with alternate least squares optimization(ALS) and stochastic gradient descent method(SGD) mentioned above.

## II. Methods and Theory

Using alternate least squares optimization(ALS):

Read the data set and divide it (or use u1.base / u1.test to u5.base / u5.test directly). Populate the original scoring matrix $R_{nu,ni}$ against the raw data, and fill 0 for null values.

Initialize the user factor matrix $P_{nu,K}$ and the item (movie) factor matrix $Q_{ni,K}$ where K is the number of potential features.

Determine the loss function and the hyperparameter learning rate $\eta$ and the penalty factor $\lambda$.

Use alternate least squares optimization method to decompose the sparse user score matrix, get the user factor matrix and item (movie) factor matrix:

- With fixd item factor matrix, find the loss partial derivative of each row (column) of the user factor matrices, ask the partial derivative to be zero and update the user factor matrices.
- With fixd user factor matrix, find the loss partial derivative of each row (column) of the item factor matrices, ask the partial derivative to be zero and update the item
- Calculate the $L_{validation}$ on the validation set, comparing with the $L_{validation}$ of the previous iteration to determine if it has converged.

Repeat step 4. several times, get a satisfactory user factor matrix P and an item factor matrix Q. Draw a $L_{validation}$ curve with varying iterations.

The final score prediction matrix $R'_{nu,ni}$ is obtained by multiplying the user factor matrix $P_{nu,k}$ and the transpose of the item factor matrix $Q_{ui,K}$.

Using stochastic gradient descent method(SGD):

Read the data set and divide it (or use u1.base / u1.test to u5.base / u5.test directly). Populate the original scoring matrix $R_{nu,ni}$ against the raw data, and fill 0 for null values.

Initialize the user factor matrix $P_{nu,K}$ and the item (movie) factor matrix $Q_{ni,K}$ where K is the number of potential features.

Determine the loss function and the hyperparameter learning rate $\eta$ and the penalty factor $\lambda$.

Use the stochastic gradient descent method to decompose the sparse user score matrix, get the user factor matrix and item (movie) factor matrix:

- Select a sample from scoring matrix randomly;
- Calculate this sample's loss gradient of specific row(column) of user factor matrix and item factor matrix;
- Use SGD to update the specific row(column) of $P_{nu,K}$ and $Q_{ni,K}$.
- Calculate the $L_{validation}$ on the validation set, comparing with the $L_{validation}$ of the previous iteration to determine if it has converged.

Repeat step 4. several times, get a satisfactory user factor matrix P and an item factor matrix Q. Draw a $L_{validation}$ curve with varying iterations.

The final score prediction matrix $R'_{nu,ni}$ is obtained by multiplying the user factor matrix $P_{nu,k}$ and the transpose of the item factor matrix $Q_{ui,K}$.

## III. Experiment

A. Dataset

1. Utilizing MovieLens-100k dataset.
2. u.data -- Consisting 10,000 comments from 943 users out of 1682 movies. At least, each user comment 20 videos. Users and movies are numbered consecutively from number 1 respectively. The data is sorted randomly

| user id | item id | rating | timestamp |
|---|---|---|---|
| 196 | 242 | 3 | 881250949 |
| 186 | 302 | 3 | 891717742 |
| 22 | 377 | 1 | 878887116 |
| 244 | 51 | 2 | 880606923 |
| 166 | 346 | 1 | 886397596 |

3. u1.base / u1.test are train set and validation set respectively, seperated from dataset u.data with proportion of 80% and 20%. It also make sense to train set and validation set from u1.base / u1.test to u5.base / u5.test.
4. You can also construct train set and validation set according to your own evaluation method.

B. Implementation

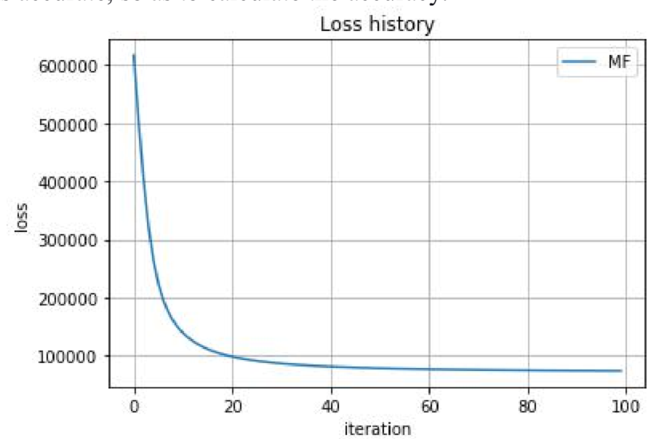
In this experiment, we used a random gradient descent method （SGD） for optimization, reference from the blog [Matrix Factorization: A Simple Tutorial and Implementation in Python]

1. Read the data set and divide it (or use u1.base / u1.test to u5.base / u5.test directly). Populate the original scoring matrix *R_n_users,n_items* against the raw data, and fill 0 for null values.
2. Initialize the user factor matrix *P_n_users,K* and the item (movie) factor matrix *Q_n_items,K* , where K is the number of potential features.
3. Determine the loss function and hyperparameter learning rate η and the penalty factor λ.
4. Use the stochastic gradient descent method to decompose the sparse user score matrix, get the user factor matrix and item (movie) factor matrix:
   4.1 Select a sample from scoring matrix randomly;
   4.2 Calculate this sample's loss gradient of specific row(column) of user factor matrix and item factor matrix;
   4.3 Use SGD to update the specific row(column) of *P_n_users,K* and *Q_n_items,K*;
   4.4 Calculate the *L_validation* on the validation set, comparing with the *L_validation* of the previous iteration to determine if it has converged.
5. Repeat step 4. several times, get a satisfactory user factor matrix P and an item factor matrix Q, Draw a *L_validation* curve with varying iterations.
6. The final score prediction matrix *R_n_users,n_items* is obtained by multiplying the user factor matrix *P_n_users,K* and the transpose of the item factor matrix *Q_n_items,K*.

C. Experiment result

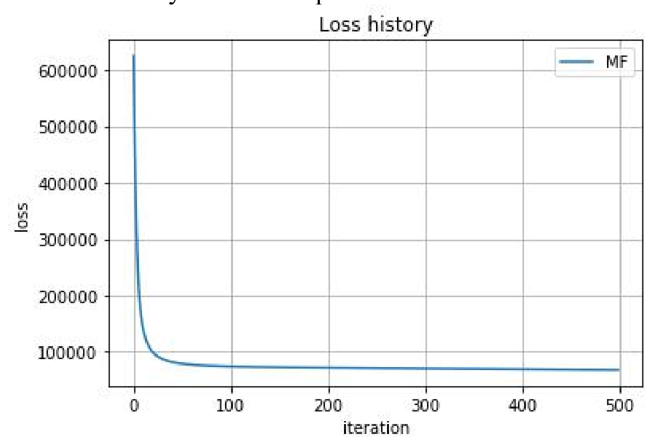
At the beginning of the experiment, we iterated 100 times to get the final Loss value (sum value) of about 73,000.

And we use the final score prediction matrix *R_n_users,n_items*, obtained by multiplying the user factor matrix *P_n_users,K* and the transpose of the item factor matrix *Q_n_items,K* to compare with the original scoring matrix *R_n_users,n_items*. If the difference is less than the threshold value (the experiment to take 0.5), then consider the evaluation is accurate, so as to calculate the accuracy.



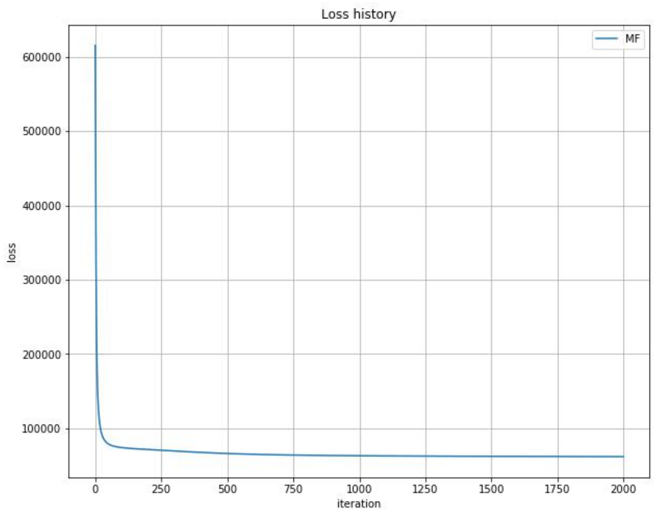

迭代次数为 100 准确率为 0.67175
最后的Loss值为 73665.624019

It then iterated 500 times, resulting in a sharp fall in Loss before 20 iterations, after which it slowed down more slowly. And the accuracy has been improved.




迭代次数为 500 准确率为 0.6888
最后的Loss值为 67385.3722542

After 2000 iterations, not much changed.




迭代次数为 2000 准确率为 0.69905
最后的Loss值为 61602.3172392

## IV. CONCLUSION

In this experiment we implement recommendation system with alternate least squares optimization(ALS) and stochastic gradient descent method(SGD). Through the experiment we find that with the increase of times of iteration,the accuracy will be improved apparently. We can have a clear understanding of recommendation system in alternate least squares optimization and stochastic gradient descent method. The detail of face detection is worth further investigating and we will study deeply in the coming research.