



## The Experiment Report of Machine Learning

---

**SCHOOL: SCHOOL OF SOFTWARE ENGINEERING**

**SUBJECT: SOFTWARE ENGINEERING**

Author:  
Hainan Huang, Ruolin Liang and  
Zequan Zeng

Supervisor:  
Qingyao Wu

Student ID:  
201720145150, 201721046029  
and 201720144917

Grade:  
Graduate

December 24, 2017

# Recommender System Based on Matrix Decomposition

**Abstract**—The emergence and popularization of the Internet have brought a great deal of information to the users and have satisfied users' demands for information in the information age. However, with the rapid development of the Internet, the substantial increase of the amount of information on the Internet has made users face a great deal of information. When you can not get from that part of the information that is really useful to yourself, but the efficiency of the use of information but reduced, and this is the so-called information overload. A very potential solution to the information overload problem is the recommendation system, which is based on the user's information needs, interests, etc. In this experiment, the matrix decomposition is used to recommend the system. The matrix decomposition is to predict the missing values in the scoring matrix and then recommend to the users according to the predicted values.

## I. INTRODUCTION

The emergence and popularization of the internet bring a great deal of information to the users and satisfy the user's demand for information in the information age. However, with the rapid development of the network, the substantial increase of the amount of information on the Internet makes the users face a great deal of information. When you can not get from that part of the information that is really useful to yourself, but the efficiency of the use of information but reduced, and this is the so-called information overload problem.

A very potential solution to the information overload problem is the recommendation system, which is based on the user's information needs, interests, etc., will be interested in the user information, products, etc. to recommend to the user's personalized information recommendation system. Compared with the search engine recommendation system through the study of the user's interest preferences, personalized calculations, the system found that the user's point of interest, so as to guide the user to find their own information needs.

Currently recommended system is the most used matrix decomposition method. Matrix factorization predicts missing values in the scoring matrix and then recommends the user somehow based on the predicted value. Common matrix decomposition methods include basic MF, Regularized MF, probability-based matrix factorization (PMF), etc

Basic MF is the most basic decomposition method. The score matrix  $R$  is decomposed into the user matrix  $U$  and the item matrix  $S$ , and the product of  $U$  and  $S$  gets closer and closer to the real matrix through continuous iterative training. Projections close to the true value is to minimize the difference, which is our objective function, and then iteratively calculate  $U$  and  $S$  using a gradient descent, which is the matrix that is decomposed when they converge.

## II. METHODS AND THEORY

Matrix decomposition is the decomposition of a matrix into two or more matrices. For the above user-movie matrix (score matrix), denoted by  $R_{m \times n}$ . It can be broken down into the product of two or more matrices, presumably decomposed into two matrices  $P_{m \times k}$  and  $Q_{k \times n}$ , So that their product can restore the original matrix.

### A. Experiment Step

Our experiment is based on the stochastic gradient descent method (SGD). The Experiment Step are as follows.

1. Read the data set and divide it ( use  $u1.base / u1.test$  to directly). Populate the original scoring matrix  $R$  against the raw data, and fill 0 for null values.
2. Initialize the user factor matrix  $p$  and the item (movie) factor matrix  $Q$ .
3. Determine the loss function and hyperparameter learning rate  $\eta$  and the penalty factor  $\lambda$ .
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  $P$  and  $Q$ .
  - 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  $R$  is obtained by multiplying the user factor matrix  $P$  and the transpose of the item factor matrix  $Q$ .

Our recommendation system is mainly accomplished through the above steps. Next, we describe each process by some mathematical formulas.

B. The derivation of the formula

$$R_{m \times n} \approx P_{m \times k} \times Q_{k \times n} = \hat{R}_{m \times n} \quad (2-1)$$

The problem is how to solve each of the elements  
Loss fuction:

$$e_{ij}^2 = (r_{ij} - \sum_{k=1}^K p_{ik} q_{kj})^2 + \frac{\beta}{2} \sum_{k=1}^K (\|P\|^2 + \|Q\|^2) \\ E = \sum_{(u_i, d_j, r_{ij}) \in T} e_{ij} \quad (2-2)$$

Negative Gradient of Loss Function:

$$\frac{\partial}{\partial p_{ik}} e_{ij}^2 = -2(r_{ij} - \hat{r}_{ij})(q_{kj}) = -2e_{ij} q_{kj} \\ \frac{\partial}{\partial q_{kj}} e_{ij}^2 = -2(r_{ij} - \hat{r}_{ij})(p_{ik}) = -2e_{ij} p_{ik} \\ \hat{r}_{ij} = p_i^T q_j = \sum_{k=1}^k p_{ik} q_{kj} \quad (2-3)$$

Negative gradient direction update variable:

$$p'_{ik} = p_{ik} + \alpha \frac{\partial}{\partial p_{ik}} e_{ij}^2 = p_{ik} + \alpha(2e_{ij} q_{kj} - \beta p_{ik}) \\ q'_{kj} = q_{kj} + \alpha \frac{\partial}{\partial q_{kj}} e_{ij}^2 = q_{kj} + \alpha(2e_{ij} p_{ik} - \beta q_{kj}) \quad (2-4)$$

### 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.(showed on the TABLEI).

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.( In our experiment, the training set we used was u1.base, and the validation set was u1.test).

TABLE I  
MovieLens-100k dataset

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

#### B. Implementation

The environment of experiment is on the basis of python3 including following python package: sklearn, numpy, jupyter,

matplotlib. In order to achieve better experimental results. Different optimization methods may set different parameters. The parameters of different optimization methods are set as follows.

TABLE II  
SIMULATION PARAMETERS

alpha	beta	steps	K
0.02	0.02	1000	2/5/10

Among all the parameters, alpha represents the learning rate, steps represents the number of iterations and K is the number of potential features.

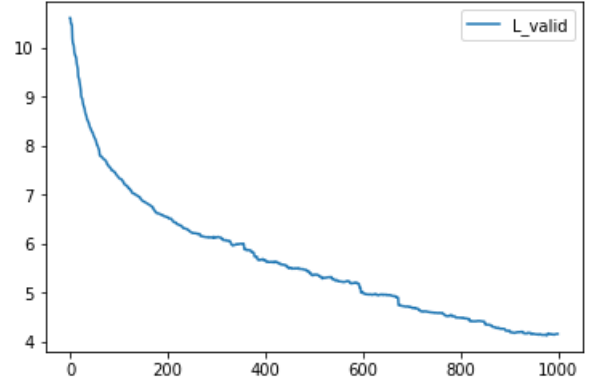


Fig .1 A  $L_{\text{validation}}$  curve with varying iterations.(k=2)

Seemed from the Fig.1 ,we can conclusion that Losses become smaller and smaller as the number of iterations increases. It validates the effectiveness of stochastic gradient descent.

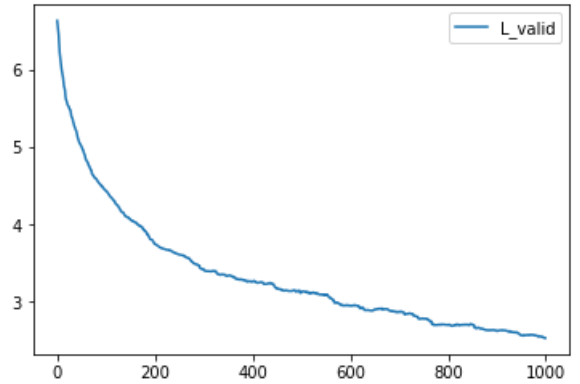


Fig .2 A  $L_{\text{validation}}$  curve with varying iterations.(k=5)

Compare Fig.1 with Fig.2,we can see that different parameters have a certain effect on the effect of the experiment. When k is 5, the algorithm converges faster than where k equal to 2.

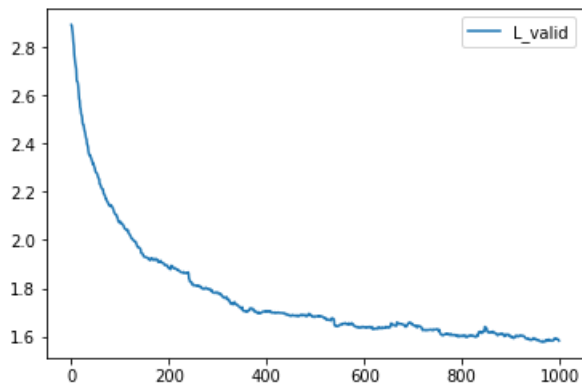


Fig .3 A  $L_{\text{validation}}$  curve with varying iterations.( $k=10$ )

Compare Fig.3 with Fig.1 and Fig.2, we can see that different parameters have a certain effect on the effect of the experiment. With the increase of  $k$ , the convergence rate of the algorithm becomes faster and faster.

#### IV. CONCLUSION

By comparing these kinds of experiments, We can draw the following conclusions.

- 1) The experiments were influenced by parameters
- 2) The algorithm can converge to a better value .
- 3) With the increase of  $k$ , the convergence rate of the algorithm becomes faster and faster.

In general, the stochastic gradient descent algorithm can better solve the experiment of recommendation system. In the study of parameter  $k$ , we found that the greater the  $k$  value, the better the convergence effect.

#### References:

1. Matrix Factorization: A Simple Tutorial and Implementation in Python.
2. Tan, W., Cao, L., & Fong, L. (2016, May). Faster and cheaper: Parallelizing large-scale matrix factorization on gpus. In Proceedings of the 25th ACM International Symposium on High-Performance Parallel and Distributed Computing (pp. 219-230). ACM.
3. Xie, X., Tan, W., Fong, L. L., & Liang, Y. (2017, June). CuMF\_SGD: Parallelized Stochastic Gradient Descent for Matrix Factorization on GPUs. In Proceedings of the 26th International Symposium on High-Performance Parallel and Distributed Computing (pp. 79-92). ACM.