华南理工大学

South China University of Technology

The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Zilong Li, Di Liu, HaijunYou

Supervisor:
Qingyao Wu

Student ID：
201721045398, 201721045374,
201721045312

Grade:
Graduate

December 14, 2017

Recommender System Based on Matrix Decomposition

Abstract—In this paper, recommender system based on matrix decomposition were realized using stochastic gradient descent(SGD) method. The details and experiments are shown in the following content.

I. INTRODUCTION

This report is about the fourth experiment of machine learning course. This experiment gives two approaches to realize the recommeder system. And our group choose the second method named SGD to finish the task. The main steps of SGD method is shown in Figure 1.

Using stochastic gradient descent method(SGD):

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 $Q_{n\_item,K}$, 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 factc 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 $Q_{n\_item,K}$;
   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 $\hat{R}_{n\_users,n\_items}$ is obtained by multiplying the user factor matrix $P_{n\_users,K}$ and the transpose of the item factor matrix $Q_{n\_item,K}$.

Figure 1 The steps of SGD

II. METHODS AND THEORY

1) Initialization

In this paper, we select u1.base as the training set, and u1.test as the testing set. $\mathbf{P}$ is initialed with random value between $0 \sim 1$ with size is $943 * 2$, and $\mathbf{Q}$ is initialed with random value between $0 \sim 1$ with size is $2 * 1682$. Other parameters are set to $iteration = 500$, $learningRa\,te = 0.01$,and $\lambda_p = 0.001, \lambda_q = 0.001$. 500 points were randomly selected in each iteration.

2) Loss function and gradient

$$Loss = (\sum_{u=0,i=0}(\mathbf{R}_{u,i} - \mathbf{p}_u^T\mathbf{q}_i)^2 + \lambda_p\|\mathbf{p}_u\|^2 + \lambda_q\|\mathbf{q}_i\|^2)/n$$

$$E_{u,i} = r_{u,i} - \mathbf{p}_u^\top \mathbf{q}_i$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{p}_u} = E_{u,i}(-\mathbf{q}_i) + \lambda_p \mathbf{p}_u$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{q}_i} = E_{u,i}(-\mathbf{p}_u) + \lambda_q \mathbf{q}_i$$

$$\mathbf{p}_u = \mathbf{p}_u + \alpha(E_{u,i}\mathbf{q}_i - \lambda_p \mathbf{p}_u)$$

$$\mathbf{q}_i = \mathbf{q}_i + \alpha(E_{u,i}\mathbf{p}_u - \lambda_q \mathbf{q}_i)$$

III. EXPERIMENT

The total loss curve of training set is shown in figure 2. And the total loss curve of testing set is shown in figure 3.

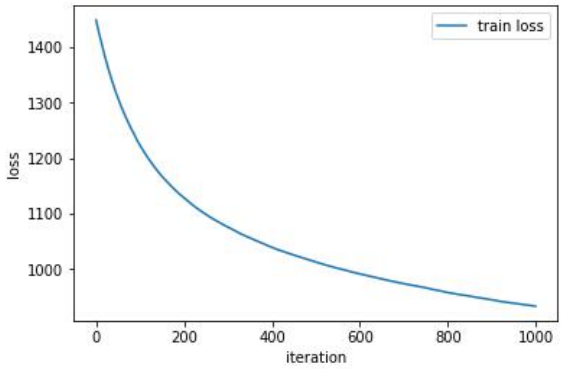

Figure 2 Curve of training set

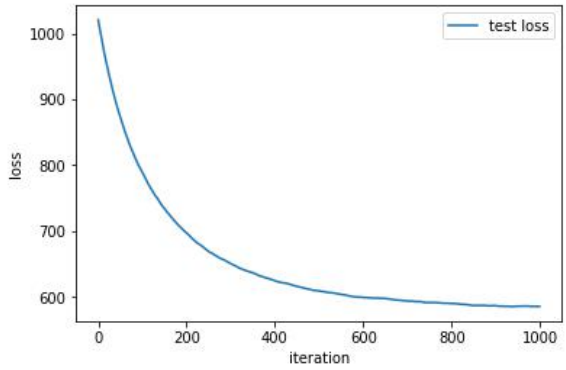

Figure 3 Curve of testing set

IV. CONCLUSION

This paper realized the recommender system with SGD. The running time is long with SGD method, for the calculation of loss is time-consuming. So 500 updates of $\mathbf{P}$ and $\mathbf{Q}$ in each iteration to decrease the running time.

The performance of SGD is good and the convergence is obtained in the end. But there is a puzzled problem about this experiment. The decomposed matrix $\mathbf{P}$ and $\mathbf{Q}$ were calculated based on the training set to make $\mathbf{P} * \mathbf{Q}$ approximately equal the training matrix $\mathbf{R}$, but in this experiment $\mathbf{P} * \mathbf{Q}$ also was used to approximately equal the testing matrix.