The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
YangZhuodong、LiJiahao、ChenYukun

Supervisor:
Qingyao Wu

Student ID：201530613412、
201530611951、201530611326

Grade:
Undergraduate

December 22, 2017

Recommender System Based on Matrix Decomposition

Abstract—

A. Introduction

1. Explore the construction of recommended system.
2. Understand the principle of matrix decomposition.
3. Be familiar to the use of gradient descent.
4. Construct a recommendation system under small-scale dataset, cultivate

B. Experiment

1. 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
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.

2. Experiment Step

The experiment code and drawing are both completed on jupyter.:

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 against the raw data, and fill 0 for null values.
2. Initialize the user factor matrix and the item (movie) factor matrix , where is the number of potential features.
3. Determine the loss function and the hyperparameter learning rate and the penalty factor .
4. Use alternate least squares optimization method to decompose the sparse user score matrix, get the user factor matrix and item (movie) factor matrix:

4.1 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

4.2 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

4.3 Calculate the on the validation set, comparing with the of the previous iteration to determine if it has converged.

5. Repeat step 4. several times, get a satisfactory user factor matrix and an item factor matrix , Draw a curve with varying iterations.
6. The final score prediction matrix is obtained by multiplying the user factor matrix and the transpose of the item factor matrix .

C. Methods and Theory

Recommendations can be generated by a wide range of algorithms. While user-based or item-based collaborative filtering methods are simple and intuitive, matrix factorization techniques are usually more effective because they allow us to discover the latent features underlying the interactions between users and items. Of course, matrix factorization is simply a mathematical tool for playing around with matrices, and is therefore applicable in many scenarios where one would like to find out something hidden under the data. Alternating Least Squares Method is a way for Collaborative Filtering and we use it to construct a recommendation system in this experiment. It aims to use two low-dimension matrix X(m*k) and Y(n*k) to approximate the original scoring matrix, that is:

$$R_{m\times n} \approx X_{m\times k} Y_{n\times k}^T$$

And the loss function is:

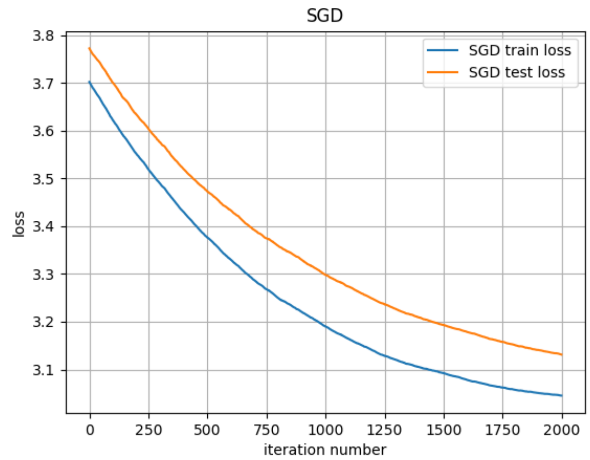
$$L(X,Y) = \sum_{u,i} (r_{ui} - x_u^T y_i)^2 + \lambda(|x_u|^2 + |y_i|^2)$$

II. EXPERIMENT

The experiment is implemented using MovieLen-100k dataset which including 10k comments of movies from 943 users. In this experiment, u1.data and u1.test splitting is used as training and test set.

The algorithm is implemented by Python3/Anaconda toolkit. The K is set to 10 and the learning rate is 0.005. 20000 iterations are implemented during the training process.

The experiment result are shown below:



III. CONCLUSION

In this experiment we implement the recommendation system via matrix decomposition algorithm. By implementing this, we have a deeper understanding to the building of a recommendation system.