South China University of Technology

The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Muyi Li, Weizhao Li, Dong Liang

Supervisor:
Mingkui Tan

Student ID:
201530612019
201530611890
201530651292

Grade:
Undergraduate

December 28, 2017

Recommender System Based on Matrix Factorization

Abstract—In this experiment, we constructed a recommender system under small-scale dataset with Matrix Factorization, to predict whether the user likes a movie he has never seen or not.

I. Introduction

Recommender system recommends appropriate objects to users, based on their behavior or item data. It mainly solves two problems, the one is information overload, and the other is the diversity of users' demand. There are tons of algorithms to generate recommendations. Though user-based or item-based collaborative filtering methods are simple and intuitive, matrix factorization techniques are usually more effective. Therefore, we tried to realize the recommender system with matrix factorization.

II. Methods and Theory

Firstly, we have a set U of users, and a set D of items. Matrix R of size |U|×|D| contains all the ratings that the users have assigned to the items.

$$R \approx P \times Q^T = \hat{R}$$

Each row in matrix P in size of |U|×K represents the strength of the associations between a user and features. And each row in matrix Q of size |D|×K represents the strength of the associations between an item and the features. To get the prediction of a rating of an item d_j by u_i, we can calculate the dot product of the two vectors corresponding to u_i and d_j:

$$\hat{r}_{ij} = P_i^T q_j = \sum_{k=1}^{k} p_{ik} q_{kj}$$

While obtaining P and Q, here is an error between the estimated and real value:

$$e_{ij}^2 = \left(r_{ij} - \sum_{k=1}^{K} p_{ik} q_{kj} \right)^2 + \frac{\beta}{2} \sum_{k=1}^{K} (||P||^2 + ||Q||^2)$$

The parameter β set to 0.02 is used to control the magnitudes of the user-feature. And item-feature vectors such that P and Q would give a good approximation of R without having to contain large numbers.

Having obtained the gradient, we can now formulate the update rules for both P_{ik} and Q_{kj}:

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

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

And α determines the rate of approaching the minimum.

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.
3. The files 'u1.base' and 'u1.test' are sets for training and validation respectively, separated 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'.

B. Steps

1. Use dataset u1.base / u1.test to u5.base / u5.test directly. Populate the original scoring matrix R_{n_user, n_item} against the raw data, and fill 0 for null values.
2. Initialize the user factor matrix $P_{n\_user,K}$ 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 α and the penalty factor β.
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 $P_{n\_user,K}$ and $Q_{n\_item,K}$;
   4.4 Calculate the loss on the validation set, comparing with the loss 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 Q. 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. Results

parameters	
α	0.02
β	0.02
features	5

Fig 1

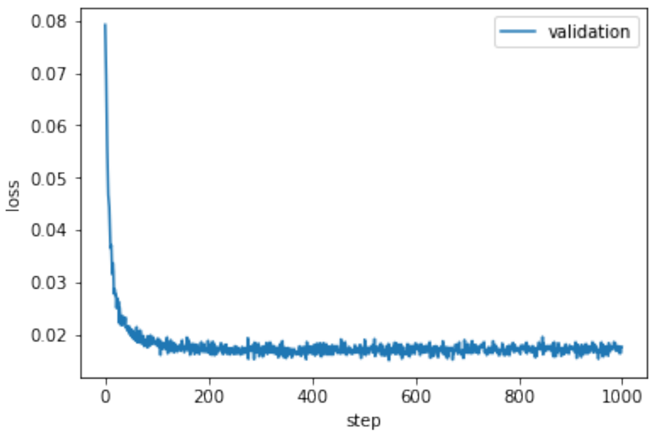

Fig 2

IV. CONCLUSION

In this experiment, we have constructed a simple recommender system under small-scale dataset with Matrix Factorization, and tried to use SGD to optimize, so that we have got a better result. With the exploration of constructing the system, we basically understand the principle of matrix factorization. And we are more familiar with the gradient descent. Therefore, from this experiment, we have cultivated our engineering ability and have strengthened the understanding of knowledge. We believe that the process means a lot.