The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Hong Zijie, LiKaizhe and ZhaoRong

Supervisor:
Qingyao Wu

Student ID：
201530611593
201530613740
201530611982

Grade:
Undergraduate

December 20, 2017

Recommender System Based on Matrix Decomposition

Abstract—Trying to explore the recommended system's matrix decomposition using the method ,including alternate least squares optimization and gradient descent

I. Introduction

we do a lab about matrix factorization in handling recommendation system problem. It's useful when the user-item matrix miss some data, which is common because no user can have review on every item, or else the recommendation system is not needed. By using the ALS and gradient descent , we optimize the decomposed matrix.

II. Methods and Theory

THE THEORY BEHIND OUR LAB IS THAT SINCE THE MATRIX OF USER-ITEM IS SO BIG, WE CAN SIMPLY MAKE THEM INTO TWO MATRIX, AFTER WE MULTIPLY THE TWO MATRIX TOGETHER, THE ORIGINAL MATRIX COMES, WITH THE MISSING DATA FULFILLED. IF WE COULD MAKE THE TWO SPLIT MATRIX OF SHAPE M*K, K*N, WHILE M*N IS THE SHAPE OF THE ORIGINAL MATRIX, THE TWO MATRIX WILL HAVE THEIR PHYSIC MEANINGS, ONE IS THE USER'S FEATURE, THE OTHER IS THE ITEM'S FEATURE. THE FEATURE DESCRIBE THEM IN A SAME WAY SO THAT WE COULD FIND THE CONNECTION BETWEEN USER AND ITEM, NEVERTHELESS, WE COULD REALIZE THE RECOMMENDATION SYSTEM IN AN EASY WAY.

To realize the theory by code, there are two ways in general. One is the ALS, since there are two matrix to update, the first idea comes up is to fix one matrix while updating another. Then use gradient descent to update every line in the updating matrix. The other way is just to use the gradient descent to update the two matrix together, it's also useful in most of time.

III. Experiment

A.data:

Utilizing MovieLens-100k dataset., using the u1.base as the training set and u1.test as the test set

B.Implementation

1.GD:

```
for i in range(iter):
    #print (i)
    for i in range(m):
        for j in range(n):
            if R[i, j] > 0:
                error = R[i, j]
                for k in range(K):
                    error = error - P[i, k] * Q[k, j]
                for k in range(K):
                    P[i, k] = P[i, k] + lr * (2 * error * Q[k, j] - lamda * P[i, k])
                    Q[k, j] = Q[k, j] + lr * (2 * error * P[i, k] - lamda * Q[k, j])
```

2.ALS

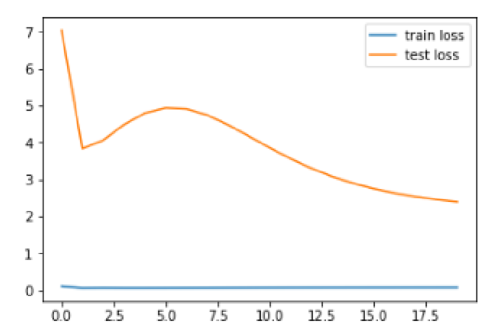
```
    #optimize P
    for i in range(m):
        u = 0
        v = zeros((30,30))
        for j in range(n):
            if R[i, j] > 0:
                #u = u + R[i, j] * Q[:, j].T
                v = v +Q[:, j]* Q[:, j].T + 0.01*I
        #print (shape(v),shape(u))
        u = R[i,:].T * Q.T
        if linalg.det(v) != 0:
            P[i, :] = u * linalg.pinv(v)
    #optimize Q fixing P
    for j in range(n):
        u = 0
        v = zeros((30,30))
        for i in range(m):
            if R[i, j] > 0:
                #u = u + R[i, j] * P[i, :].T
                v = v + P[i, :].T * P[i, :] + 0.01 * I
        u =  R[:,j] * P
        if linalg.det(v) != 0:
            Q[:, j] =  (u * linalg.pinv(v)).T
```

3.parameter tabel

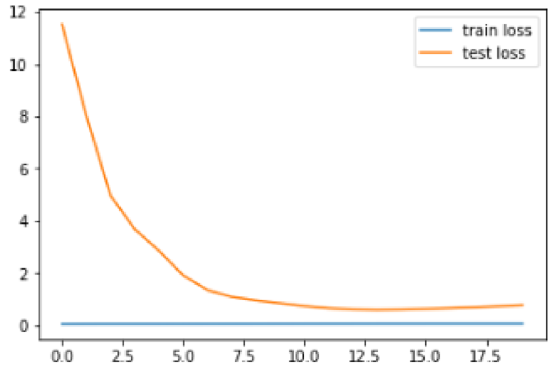
K(dimension of the decomposed matrix)	30
P lambda	0.01
Q lambad	0.01
Learing rate	0.01

4.Result:

(1) GD



(2)ALS:



IV. CONCLUSION

After we finish the code, we get the result that ALS works faster to lower its train loss, but use more time to lower its test loss, and it’s not easy to get to the lowest point but just wondering around that point. The gradient descent method works much slower than ALS, but it’s not that obviously to oscillation from the lowest point. After all, we drew a conclusion that ALS works faster, and gradient descent works slower but have a better result, they will have their own usage on different occasion.

The matrix factorization method is a basic method in nowadays recommendation system. Some other method is just to do something about initializing the two matrix, not change it. To connect with more information, we just need add some more part in the update formula, the basic theory is still constant. All these above shows how important the method is.