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:

Peiwei Chang, ZhiweiWu and Jing jing Hu

Supervisor:

Mingkui Tan

Student ID：

201530611128, 201530613092 and 201530611609

Grade:

Undergraduate

December 24, 2017

Recommender System Based on Matrix Decomposition

Abstract—

I. INTRODUCTION

II. METHODS AND THEORY

III. EXPERIMENT

Dataset:

The dataset of this experiment utilizing is MovieLens-100k dataset, which consist 10,000 ratings from 943 users out of 1,682 movies. And at least each user give 20 movies ratings.

The dataset has its own structure like the table below.

User id	Item id	rating	timestamp
196	242	3	881250949
186	302	3	891717742
22	377	1	878887116

First, I read the dataset 'u1. base' as my training set and 'u1. test' as validation set. According to the training data, I can initial the scoring matrix $R_{nusers,nitems}$. Every row of R refers to a user and column refer to a movie. If we the find the user gives its rating to the movie, the value will be the rating. If not, the initial value will be 0.

After initializing the scoring matrix $R_{nusers,nitems}$, I initialize the user factor matrix $P_{nusers,K}$ and the item(movie) factor matrix $Q_{nitem,K}$. , given K=5.

Determine that the Squared loss as loss function, and learning rate 0.002, and the regularization parameter 0.02. The next step is doing the decomposition. I use alternate least squares optimization method to do that:

```
for step in range(steps):
    for i in range(len(R)):
        for j in range(len(R[i])):
            if R[i][j] > 0:
                eij = R[i][j] - numpy.dot(P[i, :], Q[:, j])
```

1. With fixed 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 matric.

```
for k in range(K):
P[i][k] = P[i][k] + alpha * (2 * eij * Q[k][j] - beta * P[i][k])
```

2. With fixed 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 factor matric.

```
for k in range(K):
Q[k][j] = Q[k][j] + alpha * (2 * eij * P[i][k] - beta * Q[k][j])
```

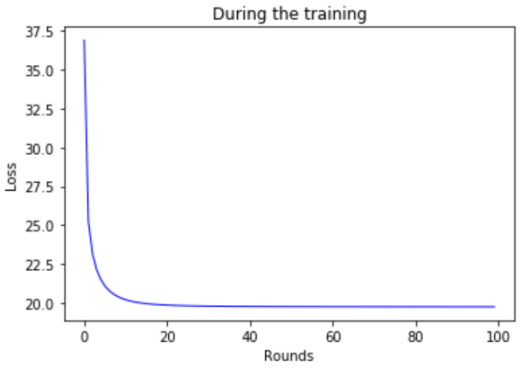
3. Calculate the loss on the validation set.

```
for each in TestData:
    userid,movieid,rating=each[0],each[1],each[2]
    predict_rating=R[int(userid)-1][int(movieid)-1]
    loss+=pow(predict_rating-rating,2)
```

After multi times of the 3 steps, we can get the final user factor matrix P and the item factor matrix Q. And after the dot of P and Q.T, we can get the final score prediction matrix R.

Result:

K=6, learning rate=0.002, regularization parameter=0.02



IV. CONCLUSION

In this experiment, we implement the matrix decomposition based on the alternate least squares optimization method. After the implementation, I have a outline of recommender system in mind. By converging the loss value, I really get sense of achievement. And during the experiment, I found that if the learning rate is a bit higher, the loss value may not converge. And if K is a bit lower, it will lead to the similar result. So it's really hard to find a balance between speed and quality of model. But finally, this experiment reinforce my understanding of recommender system indeed.