South China University of Technology

The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERIN
G

SUBJECT: SOFTWARE ENGINEERIN
G

Author:
Shoukai Xu and Yaofu Chen
Yujia Zhang and Caili Li

Supervisor:
Qingyao Wu

Student ID：
201530613689
201530611920
201530611821
201530613566

Grade:
Undergraduate

December 9, 2017

Face Classification Based on AdaBoost Algorithm

Abstract—

I. Introduction

Explore the construction of recommended system.
Understand the principle of matrix decomposition.
Be familiar to the use of gradient descent.
Construct a recommendation system under small-scale dataset, cultivate engineering ability.
The experiment code and drawing are both completed on jupyter.

Using alternate least squares optimization(ALS):

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.
Initialize the user factor matrix and the item (movie) factor matrix , where is the number of potential features.
Determine the loss function and the hyperparameter learning rate and the penalty factor .
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.
Repeat step 4. several times, get a satisfactory user factor matrix and an item factor matrix , Draw a curve with varying iterations.
The final score prediction matrix is obtained by multiplying the user factor matrix and the transpose of the item factor matrix .

II. Methods and Theory

ALS has be chosen. Least square method (also known as least squares method) is a mathematical optimization technique. It matches by finding the best function of the data by minimizing the square of the error. The least squares method can be used to find the unknown data easily and minimize the sum of squares of the errors between the obtained data and the actual data. Least squares method can also be used for curve fitting. Other optimization problems can also be expressed in terms of least-squares by minimizing energy or maximizing entropy.

$$Q = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 = \sum_{i=1}^{n} (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i)^2$$

```
P[u] = np.linalg.pinv(Q[b].T.dot(Q[b]) + lamb).dot(Q.T.dot(R[u].reshape(1,-1).T)).ravel()
```

III. EXPRIMENTS

A.

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

user id	item id
196	242
186	302
22	377
244	51
166	346

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.

B.

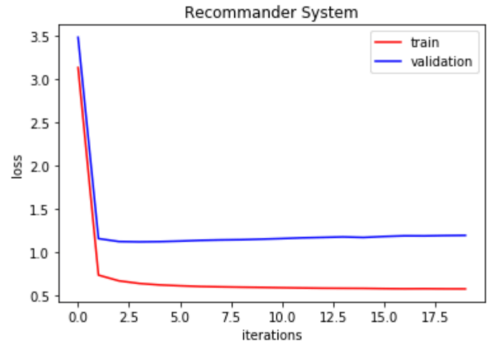
1. Read the data set and divide it (or use u1.base / u1.test to u5.base / u5.test directly). Populate the original scoring matrix against the raw data, and fill 0 for null values.
2. Initialize the user factor matrix and the item (movie) factor matrix , where is the number of potential features.
3. Determine 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 .

TABLE I

SIMULATION PARAMETERS

user factor matrix	P	P= P = np.random.normal(0,0.1,(n_user, K))
item (film) factor matrix	Q	Q = np.random.normal(0,0.1,(n_item, K))
number of potential features	K	K = 20

Result:



IV. CONCLUSION

In this experiment I have learned the ALS and SGD.

Least Squares and Gradient Descent methods are obtained by deriving the minimum value of the loss function, there are the differences between them.

Same:

1. Essentially the same: Both methods work out a general valuation function for dependent variables given the independent & dependent variables. The dependent variables for the given new data are then estimated.

2. The same goals: Both are within the framework of the known data, making the total variance of the estimated value and the actual value as small as possible (in fact not necessarily use the square), the formula of the total square deviation between the estimated value and the actual value is

$$\Delta = \frac{1}{2}\sum_{i=1}^{m}\left(f_\beta(\bar{x}_i) - y_i\right)^2$$

different:

1. Implementation methods and results are different: Least Squares method is directly on the Delta derivative to find the global minimum, is a non-iterative method. Gradient descent, on the other hand, is an iterative method, given a β , then β is adjusted to the direction in which β

declines fastest, and a local minimum is found after several iterations. The disadvantage of the gradient descent method is that the convergence rate slows to the minimum point and is extremely sensitive to the choice of the initial point, and the improvement is mostly in these two aspects.