South China University of Technology

The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Ran Yang

Supervisor:
Qingyao Wu

Student ID：
201721046128

Grade:
Graduate

November 14, 2017

Linear Regression, Linear Classification and Gradient Descent

Abstract—The experiment for Liner Regression， Linear Classification and Gradient Descent is to further understand the relationship between linear regression and Linear classification.In linear regression, the evaluation metric is often MSE(mean square error).In linear classification, the evaluation metric is often accuracy. And the threshold is a important factor to influence the accuracy.Gradient descent is one of the ways to solve linear regression.

I. Introduction

Linear regression is a linear approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X. The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression.(This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable.)

Linear classification is a classification algorithm (Classifier) that makes its classification based on a linear predictor function combining a set of weights with the feature vector. The decision boundaries of linear classification is flat. In the field of machine learning, the goal of statistical classification is to use an object's characteristics to identify which class (or group) it belongs to. A linear classifier achieves this by making a classification decision based on the value of a linear combination of the characteristics. An object's characteristics are also known as feature values and are typically presented to the machine in a vector called a feature vector. Such classifiers work well for practical problems such as document classification, and more generally for problems with many variables (features), reaching accuracy levels comparable to non-linear classifiers while taking less time to train and use.

Gradient descent is a first-order iterative optimization algorithm for finding the minimum of a function. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point. If instead one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent.Gradient descent is also known as steepest descent. However, gradient descent should not be confused with the method of steepest descent for approximating integrals.

The motivation of this experiment is as follows:

1.Further understand of linear regression and gradient descent.

2.Conduct some experiments under small scale dataset.

3.Realize the process of optimization and adjusting parameters.

II. Methods and Theory

Gradient descent is an optimization algorithm, popularly speaking, is along the direction of gradient down to find the minimum value of a function. Then we have learned in advanced mathematics, for some of the function equations we know, we can find the first and second derivative, such as the quadratic function. However, when we deal with the problem not all of the familiar functions, and since it is machine learning should let the machine itself to learn how to solve it, obviously we need to change the way of thinking. So we use the gradient descent, continuous iteration, along the direction of gradient down to move, find the minimum value.

For a linear model, follow.

h (x) is a function that needs to be fitted.

J (θ) is called mean square error or cost function. It is a measure of how well a training sample fits into a linear model.

m is the number of training samples.

θ is the parameter that we ultimately need to get through the gradient descent method.

Hypothesis:

h θ (x)= θ 0+ θ 1x

Parameters:

θ 0, θ 1

Cost Function:

J(θ 0, θ 1)=12m∑m1(h θ (x(i))−y(i))2

Goal:

minimize θ 0, θ 1 J(θ 0, θ 1)

Gradient descent algorithm

repeat until convergence {

θ j:= θ j− α ∂∂ θ jJ(θ 0, θ 1)(forj=0andj=1)

}

Correct: Simultaneous update

temp0:= θ 0− α ∂∂ θ 0J(θ 0, θ 1)

temp1:= θ 1− α ∂∂ θ 1J(θ 0, θ 1)

θ 0:=temp0

θ 1:=temp1

III. Experiment

Linear Regression uses Housing in LIBSVM Data, including 506 samples and each sample has 13 features. You are expected to download scaled edition. After downloading, you are supposed to divide it into training set, validation set. Linear classification uses Australian in LIBSVM Data, including 690 samples and each sample has 14 features. You are expected to download scaled edition. After downloading, you are supposed to divide it into training set, validation set.

Linear Regression and Gradient Descent

a)Load the experiment data. You can use load_svmlight_file function in sklearn library.

b)Devide dataset. You should divide dataset into training set and validation set using train_test_split function. Test set is not required in this experiment.

c)Initialize linear model parameters. You can choose to set all parameter into zero, initialize it randomly or with normal distribution.

d)Choose loss function and derivation: Find more detail in PPT.

e)Calculate gradient G toward loss function from all samples.

f)Denote the opposite direction of gradient G as D .

g)Update model: . η is learning rate, a hyper-parameter that we can adjust.

h)Get the loss under the training set and by validating under validation set.

i)Repeat step 5 to 8 for several times, and drawing graph of as well as with the number of iterations.

Linear Classification and Gradient Descent

a)Load the experiment data.

b)Divide dataset into training set and validation set.

c)Initialize SVM model parameters. You can choose to set all parameter into zero, initialize it randomly or with normal distribution.

d)Choose loss function and derivation: Find more detail in PPT.

e)Calculate gradient G toward loss function from all samples.

f)Denote the opposite direction of gradient G as D.

g)Update model: . η is learning rate, a hyper-parameter that we can adjust.

h)Select the appropriate threshold, mark the sample whose predict scores greater than the threshold as positive, on the contrary as negative. Get the loss under the trainin set and by validating under validation set.

i)Repeat step 5 to 8 for several times, and drawing graph of as well as with the number of iterations.

Hyper-parameter selection： η =0.1

Assessment Results (based on selected validation)：the loss function value on validation data set declines over iterations and finally converge to an optimal value.Predicted Results (Best Results): the value of loss function on training dataset declines as the number of iterations, and the same as the loss on validation dataset.

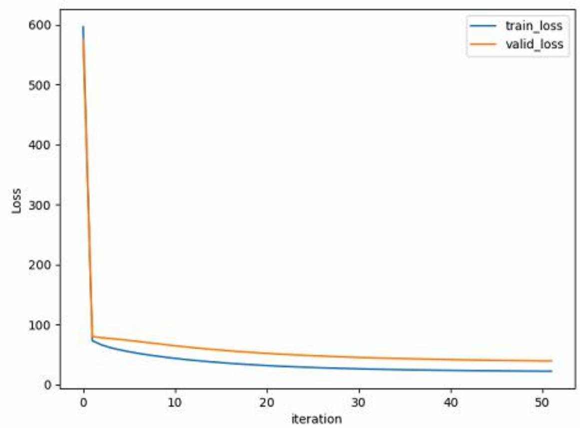

Fig.1 loss curve

IV. CONCLUSION

Linear Regression is used to slove the regression problem with the continuous label. Linear Classification is used to slove the binary classification problem with the discrete label. The learning rate represent the step size of descent. The small learning rate slow down the learning speed. The large learning rate may jump out of the minimum, so the learning curve will look like a wave.As the training times ascending, the predicted results is more and more closed to the true result.