The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Hu Lin

Supervisor:
Mingkui Tan or Qingyao Wu

Student ID:
201721045497

Grade:
Graduate

December 14, 2017

Linear Regression, Linear Classification and Gradient Descent

***Abstract*—This is the experiment report one of the Machine Learning. The main motivation of experiment is further understanding of linear regression and gradient descent , conduct some experiments under small scale dataset to realize the process of optimization and adjusting parameters.**

I. Introduction

The main motivation of experiment is further understanding of linear regression and gradient descent , conduct some experiments under small scale dataset to realize the process of optimization and adjusting parameters.

II. Methods and Theory

In statistics, 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.)

In linear regression, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models. Most commonly, the conditional mean of y given the value of X is assumed to be an affine function of X; less commonly, the median or some other quantile of the conditional distribution of y given X is expressed as a linear function of X. Like all forms of regression analysis, linear regression focuses on the conditional probability distribution of y given X, rather than on the joint probability distribution of y and X, which is the domain of multivariate analysis.

Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications. This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine.

The loss function of liner regression is LSL(Least squared loss), and it’s defined as,

$$J(\boldsymbol{\theta}) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

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.

A (linear) support vector machine (SVM) just solves the canonical machine learning optimization problem using hinge loss and linear hypothesis, plus an additional regularization term,

$$\text{minimize}_\theta \quad \sum_{i=1}^{m}\max\{1 - y^{(i)}\cdot\theta^T x^{(i)}, 0\} + \frac{\lambda}{2}\|\theta\|_2^2$$

Updates using gradient descent:

$$\theta := \theta \ - \alpha\sum_{i=1}^{m} -y^{(i)}x^{(i)}1\{y^{(i)}\cdot\theta^T x^{(i)} \le 1\} - \alpha\lambda\theta$$

III. Experiment

A. Dataset

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.

B. Environment for Experiment

python3, at least including following python package: sklearn, numpy, jupyter, matplotlib. It is recommended to install anaconda3 directly, which has built-in python package above.

C. Experiment Step

The experimental code and drawing are completed on jupyter.

Linear Regression and Gradient Descent

1. Load the experiment data. You can use load_svmlight_file function in sklearn library.
2. 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.
3. Initialize linear model parameters. You can choose to set all parameter into zero, initialize it randomly or with normal distribution.
4. Choose loss function and derivation: Find more detail in PPT.
5. Calculate gradient G toward loss function from all sam-

ples.

6. Denote the opposite direction of gradient G as D.
7. Update model: $W_t = W_{t-1} + \eta D$. η is learning rate, a hyper-parameter that we can adjust.
8. Get the loss L_{train} under the training set and $L_{validation}$ by validating under validation set.
9. Repeat step 5 to 8 for several times, and drawing graph of L_{train} as well as L_{train} with the number of iterations.

Linear Classification and Gradient Descent

1. Load the experiment data.
2. Divide dataset into training set and validation set.
3. Initialize SVM model parameters. You can choose to set all parameter into zero, initialize it randomly or with normal distribution.
4. Choose loss function and derivation: Find more detail in PPT.
5. Calculate gradient G toward loss function from all samples.
6. Denote the opposite direction of gradient G as D .
7. Update model: $W_t = W_{t-1} + \eta D$. η is learning rate, a hyper-parameter that we can adjust.
8. Select the appropriate threshold, mark the sample whose predict scores greater than the threshold as positive, on the contrary as negative. Get the loss L_{train} under the trainin set and $L_{validation}$ by validating under validation set.
9. Repeat step 5 to 8 for several times, and drawing graph of L_{train} as well as L_{train} with the number of iterations.

D. *Experiment Result*

We divide the dataset into two halves, one for train where the other one for validation.

Linear Regression and Gradient Descent

The parameter theta is initial as zero, and the learning rate is 0.01, epoch is 100. In each iteration, the theta will be used to computer loss in both train set and validation set. The final loss curve of L_{train} as well as L_{train} is show as Fig. 1.

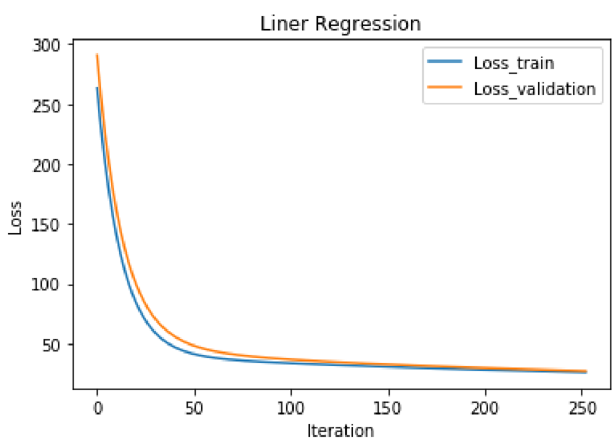

Fig. 1. Curve of Liner Regression loss on the train set and validation set

Linear Classification and Gradient Descent

The parameter theta is initial as zero, and the learning rate is 0.1, lambda is 0.01, and epoch is 100. In each iteration, the theta will be used to computer loss in both train set and validation set. The final accuracy of this classifier is 80.87%. Loss curve of L_{train} as well as L_{train} is show as Fig. 2.

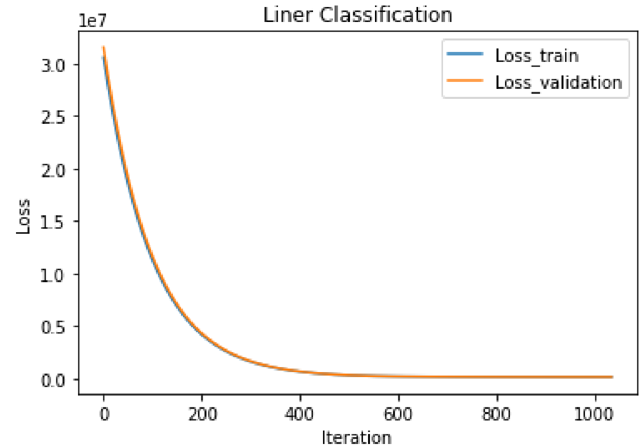

Fig. 2. Curve of Liner Classification loss on the train set and validation set

Overall, the liner regression and liner classification are essentially the same, that is, the fitting (matching) of the model. However, the **y** value (also known as label) of the classification problem is more discretized, and the same **y** value may correspond to a large number of **x**, which is of a certain range.

Therefore, the classification problem is some x in a certain region corresponds to a singe **y**, and the model of regression problem is more inclined to map x in a very small region or x in general to **y**.

IV. CONCLUSION

In this experiment, through the practice of small-scale data sets, I realized the process of optimization and reference, and further understand the principle of linear regression and gradient descent. It also raised my level of coding for machine learning.