The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Yingting Huang

Supervisor:
Qingyao Wu

Student ID：
201720145136

Grade:
Graduate

December 12, 2017

Linear Regression, Linear Classification and Gradient Descent

Abstract—
To further understand of linear regression, Linear Classification and gradient descent, I conduct some experiments under small scale dataset. During the process of optimization and adjusting parameters, I relate theory with practice. Now I am more be acquainted with the models as well as the python.

Key word: linear regression, Linear Classification, gradient descent

I. Introduction

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.

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.

As for linear classification, it is the same as linear regression in essence. Both of them are trained to fit the model and related to prediction, where regression predicts a value from a continuous set, whereas classification predicts the 'belonging' to the class. Given the following：

F: x-->y

If y is discrete/categorical variable, then this is classification problem.

If y is real number/continuous, then this is a regression problem.

II. Methods and Theory

A. The selected loss function and its derivatives

1. Linear Regression and Gradient Descent

1) Loss Function：

$$L = \|W^T X - y\|^2 \tag{1}$$

2) Gradient Matrix：

$$\frac{\partial L}{\partial W} = 2(W^T X - y)X \tag{2}$$

2. Linear Classification and Gradient Descent

1) Loss Function：

$$L = \frac{1}{N}\sum_i \sum_{j \neq y_i}[max(0, f(x_i; W)_j - f(x_i; W)_{y_i} + \Delta] + \lambda \sum_k \sum_l W_{k,l}^2 \tag{3}$$

2) Gradient Matrix：

$$\frac{\partial L}{\partial w_{y_i}} = -(\sum_{j \neq y_i} 1(w_j^T x_i - w_{y_i}^T x_i + \Delta > 0))x_i \qquad (j = y_i)$$
$$\frac{\partial L}{\partial w_j} = 1(w_j^T x_i - w_{y_i}^T x_i + \Delta > 0))x_i \qquad (j \neq y_i) \tag{4,5}$$

Where l(x) is an indicator function：

$$l(x == T) = 1$$
$$l(x == F) = 0 \tag{6,7}$$

III. Experiment

A. Data sets and data analysis

Linear Regression uses Housing in LIBSVM Data, including 506 samples and each sample has 13 features.

Linear classification uses Australian in LIBSVM Data, including 690 samples and each sample has 14 features.

Both od the data I download has been scaled to [-1,1]. After that, I divide the data into training set, validation set. The proportion of training set is 60% while validation set accounts for 40%.

B. Experimental steps

1 Linear Regression and Gradient Descent

1) Load the experiment data.
2) Divide dataset into training set and validation set.
3) Initialize linear model parameters.
4) Choose loss function and derivation.
5) Calculate gradient toward loss function from all samples.
6) Denote the opposite direction of gradient G as D.

7) Update model: $W_t = W_{t-1} + \eta D$.
8) Get the loss L_{train} under the training set and $L_{\text{validation}}$ by validating under validation set.
9) Repeat step 5 to 8 for several times, and drawing graph of L_{train} as well as $L_{\text{validation}}$ with the number of iterations.

2 Linear Classification and Gradient Descent

1) Load the experiment data.
2) Divide dataset into training set and validation set.
3) Initialize SVM model parameters.
4) Choose loss function and derivation.
5) Calculate gradient toward loss function from all samples.
6) Denote the opposite direction of gradient G as D.
7) Update model: $W_t = W_{t-1} + \eta D$.
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 training set and $L_{\text{validation}}$ by validating under validation set.
9) Repeat step 5 to 8 for several times, and drawing graph of L_{train} as well as $L_{\text{validation}}$ with the number of iterations

C. Selection of validation

1 Linear Regression and Gradient Descent

The evaluating method of linear regression and gradient descent I select is *simple cross validation*. I divide the data into training set, validation set randomly. And the proportion of training set is 60% while validation set accounts for 40%. At the end of the experiment, I choose the RMSE, R-squared, adjusted R-squared to assess the result of the linear model.

2 Linear Classification and Gradient Descent

The evaluating method of linear classification and gradient descent I select is *simple cross validation*. I divide the data into training set, validation set randomly. And the proportion of training set is 60% while validation set accounts for 40%. At the end of the experiment, I choose the accuracy to assess the result of the SVM model.

D. The initialization method of model parameters

1 Linear Regression and Gradient Descent

All parameters are set into zero in the linear model.

2 Linear Classification and Gradient Descent

All parameters are set into zero in the SVM model.

E. Experimental results and curve

1 Linear Regression and Gradient Descent

1) Hyper-parameter selection
- ✓ *eta*, the learning rate, is set to 0.01
- ✓ *maxIterations*, the maximum number of iterations, is set to 1000

2) Assessment Results
- ✓ *RMSE_train*, the root mean squared error of training data, is equal to 4.96136623475
- ✓ *R-squared_train*, the coefficient of determination of the training data, is equal to 0.724127965617
- ✓ *Adjusted R-squared_train*, the degree-of-freedom adjusted coefficient of determination of the training data, is equal to 0.710717519501

3) Predicted Results (Best Results):
- ✓ *RMSE_validation*, the root mean squared error of validationing data, is equal to 4.89733305645
- ✓ *R-squared_validation*, the coefficient of determination of the validationing data, is equal to 0.692970318205
- ✓ *Adjusted R-squared_validation*, the degree-of-freedom adjusted coefficient of determination of the validationing data, is equal to 0.67010640573

4) Loss curve:

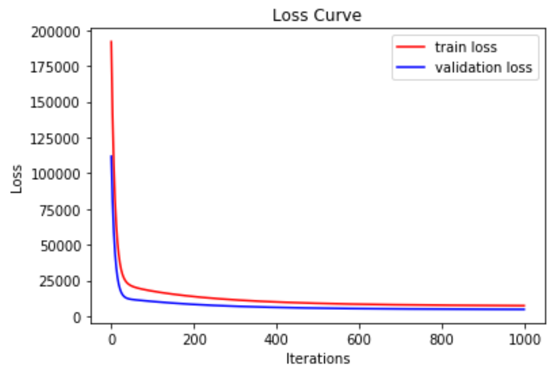

Fig 1

2 Linear Classification and Gradient Descent

1) Hyper-parameter selection:
- ✓ *eta*, the learning rate, is set to 0.01
- ✓ *maxIterations*, the maximum number of iterations, is set to 1000
- ✓ *threshold*, marks the sample whose predict scores greater than the threshold as positive, on the contrary as negative., is set to 0.
- ✓ *reg*, the regularization strength, is set to 0.1

2) Assessment Results (based on selected validation):
- ✓ *training_accuracy*, the accuracy of training data, is equal to 0.857487922705314

3) Predicted Results (Best Results):
- ✓ *validation_accuracy*, the accuracy of validation data, is equal to 0.8514492753623188

4) Loss curve:

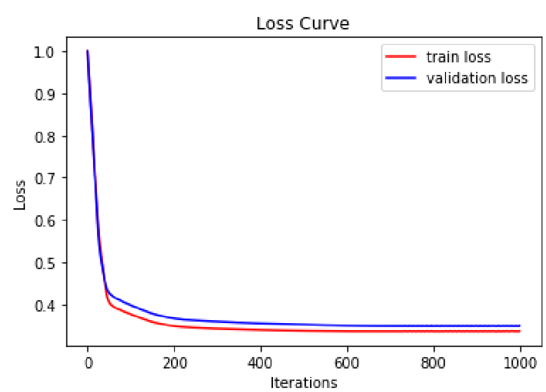

Fig 2

F. Results analysis

1 Linear Regression and Gradient Descent

From the Assessment Results in E.1 2) and the predicted results in E.1 3), we can see that the accuracy is high, which mean the predicting effect of the model is good.

From the loss curve in Fig 1, the train loss and the validation loss converge to a small number nears zero with the number of iterations. That is to say, the model we have trained is robust.

2 Linear Classification and Gradient Descent

From the Assessment Results in E.2 2) and the predicted results in E.2 3), we can see that the root mean squared error is little and the degree-of-freedom adjusted coefficient of determination is close to 1, which mean the predicting effect of the Model is good.

From the loss curve in Fig 2, the train loss and the validation loss converge to a small number with the number of iterations. That is to say, the model we have trained is robust.

IV. CONCLUSION

In the end, I think it is important to be clear when using terms like regression, classification and prediction to discriminate between the task we are performing and the method used to perform it. A classification task involves taking an input and labelling it as belonging to a given class, so the output is categorical. On the other hand, a prediction task involves predicting a continuous valued output.

Methods for achieving these tasks include regression, in which a continuous valued output is estimated (or, rather, the expected value of a distribution on a continuous variable is estimated, conditional on a given set of input values). This can be used to carry out a prediction task, as you would expect. It can also be used to carry out a classification task, for example using logistic regression to estimate the log odds of the input pattern belonging to a given class. In this case, the task is classification, the method is regression.

Classification methods simply generate a class label rather than estimating a distribution parameter. By the way, K nearest neighbor (KNN) is a good example where the task and the method are both called classification.