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:
Yongliang Wen

Supervisor:
Qingyao Wu

Student ID:
201530613023

Grade:
Undergraduate

December 15, 2017

Logistic Regression, Linear Classification and Stochastic Gradient Descent

Abstract—The short abstract is intended to give the reader an overview of the experiment. It should be brief and to the point.

I. Introduction

TRY the stochastic gradient descent algorithm. Compare and understand the differences and relationships between Logistic regression and linear classification. Further understand the principles of SVM and practice on larger data.

II. Methods and Theory

A. SGD

Stochastic gradient descent (often shortened to SGD), also known as incremental gradient descent, is a stochastic approximation of the gradient descent optimization and iterative method for minimizing an objective function that is written as a sum of differentiable functions.

B. Logistic regression

Logistic regression was developed by statistician David Cox in 1958. The binary logistic model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features). It allows one to say that the presence of a risk factor increases the odds of a given outcome by a specific factor.

C. Logistic regression

A linear classifier achieves this by making a classification decision based on the value of a linear combination of the characteristics. We use :

$$f(x_i, W, b) = Wx_i + b$$

III. Experiments

A. Logistic Regression and Stochastic Gradient Descent

1) Dataset: Experiment uses a9a of LIBSVM Data, including 32561/16281(testing) samples and each sample has 123/123 (testing) features. Please download the training set and validation set.

2) Implementation: All detailed implementation in my experiment: initialization, process, results, all kinds of parameters.

TABLE I
NAG:Initialization Parameters

Learning rate	$lr = 0.01$
Max iteration number	$n\_estimator = 300$
Features weight w	$w = random \quad normal \quad distribution$
v	$v = n\_feature \quad size \quad zeros$
mu	$mu = 0.001$
Regularization parameters	$alpha = 1.$

TABLE II
RMSProp:Initialization Parameters

Learning rate	$lr = 0.01$
Max iteration number	$n\_estimator = 300$
Features weight w	$w = random \quad normal \quad distribution$
v	$v = n\_feature \quad size \quad zeros$
mu	$mu = 0.9$
sigma	$sigma = 1e - 5$
G	$G = 0.$
Regularization parameters	$alpha = 1.$

TABLE III
AdaDelta:Initialization Parameters

Learning rate	$lr = 0.01$
Max iteration number	$n\_estimator = 300$
Features weight w	$w = random \quad normal \quad distribution$
sigma	$sigma = 1e - 5$
G	$G = 0.$
dt	$dt = 0.$
mu	$mu = 0.001$
Regularization parameters	$alpha = 1.$

TABLE IV
Adam:Initialization Parameters

Learning rate	$eta = 0.1$
Max iteration number	$n\_estimator = 300$
Features weight w	$w = random \quad normal \quad distribution$
v	$v = n\_feature \quad size \quad zeros$
mu	$mu = 0.999$
sigma	$sigma = 1e - 5$
G	$G = 0.$
belta	$belta = 0.9$
Regularization parameters	$alpha = 1.$

B. Linear Classification and Stochastic Gradient Descent

1) Dataset: Experiment uses a9a of LIBSVM Data, including 32561/16281(testing) samples and each sample has 123/123 (testing) features. Please download the training set and validation set.

2) Implementation: All detailed implementation in my experiment: initialization, process, results, all kinds of parameters.

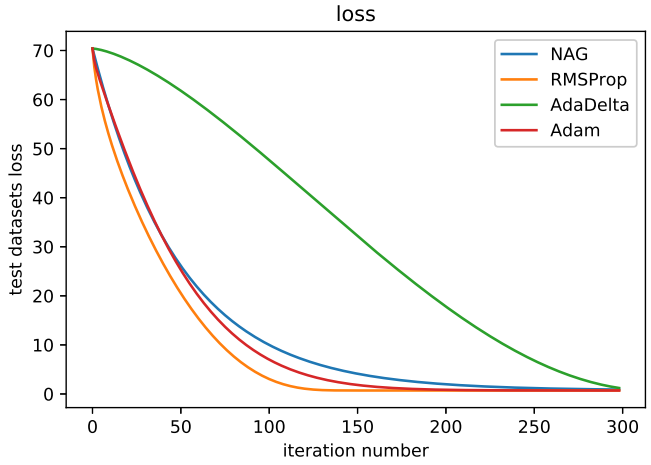

Fig. 1. Logistic Regression 4 types optimization method.

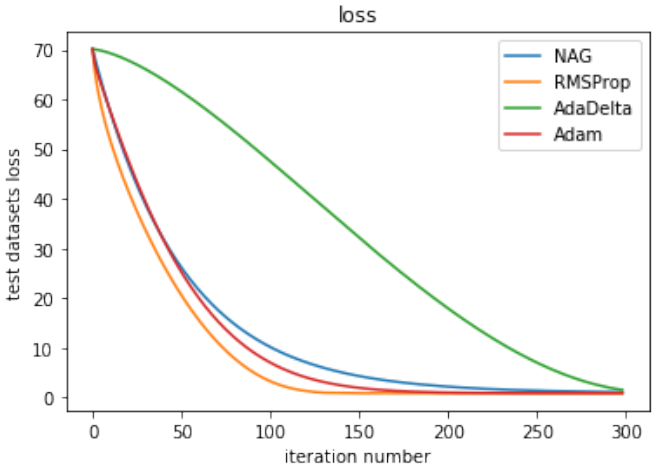

Fig. 2. Linear Classification 4 types optimization method.

TABLE V
NAG:Initialization Parameters

Learning rate	$lr = 0.01$
Max iteration number	$n\_estimator = 300$
Features weight w	$w = random \quad normal \quad distribution$
v	$v = n\_feature \quad size \quad zeros$
mu	$mu = 0.001$
Regularization parameters	$alpha = 1.$

TABLE VI
RMSProp:Initialization Parameters

Learning rate	$lr = 0.01$
Max iteration number	$n\_estimator = 300$
Features weight w	$w = random \quad normal \quad distribution$
v	$v = n\_feature \quad size \quad zeros$
mu	$mu = 0.9$
sigma	$sigma = 1e - 5$
G	$G = 0.$
Regularization parameters	$alpha = 1.$

TABLE VII
AdaDelta:Initialization Parameters

Learning rate	$lr = 0.01$
Max iteration number	$n\_estimator = 300$
Features weight w	$w = random \quad normal \quad distribution$
sigma	$sigma = 1e - 5$
G	$G = 0.$
dt	$dt = 0.$
mu	$mu = 0.001$
Regularization parameters	$alpha = 1.$

TABLE VIII
Adam:Initialization Parameters

Learning rate	$eta = 0.1$
Max iteration number	$n\_estimator = 300$
Features weight w	$w = random \quad normal \quad distribution$
v	$v = n\_feature \quad size \quad zeros$
mu	$mu = 0.999$
sigma	$sigma = 1e - 5$
G	$G = 0.$
belta	$belta = 0.9$
Regularization parameters	$alpha = 1.$

IV. Conclusion

I learn 4 type optimization algorithm to update the feature weight,and I pick up Stochastic Gradient Descent algorithm. Also I learn some Python tricks like Functional programming and List generator which make my code concise and efficient.