The Experiment Report of *Machine Learning*

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Yawen Xu

Supervisor:
Mingkui Tan

Student ID:
201820137853

Grade:
Graduate

October 27, 2018

Linear Regression and Stochastic Gradient Descent

***Abstract*—The theme of this project is to have a deeper understanding of linear regression, closed-form solution and stochastic gradient descent.**

I. Introduction

THERE is a simple and traditional model called linear regression, which describes the linear relationship between a response variable and one or more predictor variables. We have two methods to achieve this model, one is closed-form solution, and the other is stochastic gradient descent. This project aims to implement these two methods by python and numpy, and conduct some experiments under a small scale dataset. The experimental results on the small scale dataset show that linear regression has a good imitative effect.

II. Methods and Theory

In statistics, linear regression is a linear approach to modelling the relationship between a scalar response and one or more explanatory variables. Linear regression plays and important role in the field of machine learning. The linear regression algorithm is one of the fundamental supervised machine-learning algorithms.

We denote X as the input, and Y as the output. Our goal is to learn a hypothesis function $f : X \to Y$. The linear regression function is defined as follows:

$$\begin{aligned} f(\boldsymbol{x}; w_0, \boldsymbol{w}) &= w_0 + w_1 x_1 + w_2 x_2 + ... + w_m x_m \\ &= \sum_{j=1}^{m} w_j x_j + w_0 \\ &= \boldsymbol{w}^T \boldsymbol{x} + w_0 \end{aligned} \tag{1}$$

where $\boldsymbol{w}$ and w_0 represents the parameters in this equation, m represents the number of features in X. We use a loss function to measure the "goodness" of a linear regression model. In this project, we use least square loss function, which is defined as follows:

$$\begin{aligned} \boldsymbol{L}_D(w) &= \frac{1}{2} \sum_{i}^{n} (y_i - f(x_i; \boldsymbol{w}))^2 \\ &= \frac{1}{2} \sum_{i}^{n} (y_i - \boldsymbol{w}^T) {x_i}^2 \\ &= \frac{1}{2} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) \end{aligned} \tag{2}$$

Our goal is to find the best "$\boldsymbol{w}$" which could minimize the least squared loss.

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} L_D(w) \tag{3}$$

III. Experiments

A. Dataset

This section represents the related information of datasets, such as the content, the number of data, the training set, the validation set and so on.

B. Implementation

All detailed implementation in your experiment: initialization, process, results, all kinds of parameters. In a word, describe clearly What you do and how you do.

Figures and tables should be labeled and numbered, such as in Table I and Fig. 1.

TABLE I
Simulation Parameters

Information message length	$k = 16000$ bit
Radio segment size	$b = 160$ bit
Rate of component codes	$R_{cc} = 1/3$
Polynomial of component encoders	$[1, 33/37, 25/37]_8$

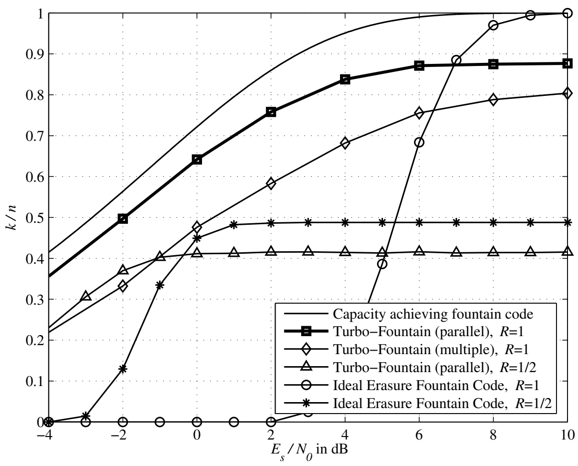

Fig. 1. Simulation results on the AWGN channel. Average throughput k/n vs E_s/N_0.

IV. Conclusion

This section summarizes the paper. In our experiments, you can also write your gains and inspirations in here.