CS 560 Statistical Machine Learning: Homework 4

Instructor: Jie Shen

March 31, 2019

Notation: The solution $\boldsymbol{w} \in \mathbb{R}^d$ is always a column vector, and the feature vector $\boldsymbol{x} \in \mathbb{R}^d$ is always a row vector.

Linear Regression: A Statistical Perspective

Suppose we are given a data set $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, where each $\boldsymbol{x}_i \in \mathbb{R}^d \times \mathbb{R}$ is a row vector. In order to learn a good model $\boldsymbol{w}$ from the data, it typically boils down to solving the following *least-squares* program:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \; F(\boldsymbol{w}) := \frac{1}{2} \left\| \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} \right\|_2^2, \tag{1}$$

where $\boldsymbol{X}$ is the data matrix with the ith row being $\boldsymbol{x}_i$, and $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^\top$.

Optimization. Denote

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \; F(\boldsymbol{w}). \tag{2}$$

In Homework 3, we have studied how to make use of gradient descent to obtain an iterate $\boldsymbol{w}^t$ such that

$$\left\| \boldsymbol{w}^t - \boldsymbol{w}^* \right\|_2 \leq \epsilon \tag{3}$$

for any user-specified error parameter $\epsilon \in (0, 1)$. The paradigm of getting the above approximate solution is called optimization.

Statistical Estimation. Of the central interest in statistical machine learning (SML) is estimating the *groundtruth* model from the training data. That is, SML assumes that there exists a true model $\boldsymbol{w}_{\text{true}}$ that generates all the observations in the following way:

$$y_i = \langle \boldsymbol{x}_i, \boldsymbol{w}_{\text{true}} \rangle + e_i, \quad e_i \sim N(0, \sigma^2), \quad i = 1, \ldots, n.$$

In the above expression, e_i is random Gaussian noise with mean zero and variance σ^2.

Since $\boldsymbol{w}^*$ is the one that best fits the training data, a fundamental question raised in SML is when $\boldsymbol{w}^* \approx \boldsymbol{w}_{\text{true}}$.

Parameter. Fix $d = 1000$ and $\sigma = 0.1$.

Estimation in Low-Dimensions

Randomly generate the groundtruth model $\boldsymbol{w}_{\text{true}} \in \mathbb{R}^d$. You can use the Gaussian distribution, or any other distributions such as uniform distribution over $[-1000, 1000]$. *You need to save the model and never change it throughout the experiments.*

Now let $n_{\max} = 10000$. Randomly generate the $n_{\max}$-by-d data matrix $\boldsymbol{X}$ using the built-in python API `numpy.random.randn`. Make sure that the variance of this Gaussian distribution is 1. Use the same python API to generate the random Gaussian noise $\boldsymbol{e} = (e_1, \ldots, e_{n_{\max}})^\top$, but with variance 0.01 (i.e. $\sigma = 0.1$). Finally you will have the response vector $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}_{\text{true}} + \boldsymbol{e}$.

We aim to study how the estimation error $\|\boldsymbol{w}^* - \boldsymbol{w}_{\text{true}}\|_2$ changes with the sample size n. To this end, we increase n from 1000 to $n_{\max}$, with a step size 500. For each n,

1. Collect $(\boldsymbol{x}_1, y_1), \ldots (\boldsymbol{x}_n, y_n)$ and solve the program (1) with gradient descent. As in Homework 3, you should be able to calculate a good learning rate η_0 and run GD for 100 iterations.
2. Record the ℓ_2 distance between the final iterate produced by GD and $\boldsymbol{w}_{\text{true}}$. Let us denote it as z_n.

Plot the curve z_n v.s. n and summarize your findings.

Estimation in High-Dimensions

In the preceding experiment, n is always greater than $d = 1000$ which is referred to as low-dimensional regime. Now let us increase n from 100 to 1000 with a step size 100, and for each n we calculate z_n as before. Plot the curve z_n v.s. n and summarize your findings.

Structural Estimation in High-Dimensions

Let the sparsity parameter $k = 20$. Randomly generate a d-dimensional vector $\boldsymbol{w}'_{\text{true}}$ as before, then keep k coordinates and set others to zero. This way we obtain a $\boldsymbol{w}_{\text{true}} \in \mathbb{R}^d$ with sparsity k ($d = 1000$). The sparsity structure in $\boldsymbol{w}_{\text{true}}$ is the primary difference.

Let $n_{\max} = 1000$. Generate the $n_{\max}$-by-d data matrix $\boldsymbol{X}$ and $n_{\max}$-dimensional noise $\boldsymbol{e}$ as before. Finally you will have the response vector $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}_{\text{true}} + \boldsymbol{e}$ where $\|\boldsymbol{w}_{\text{true}}\|_0 = k$.

Estimation without prior knowledge. Let us increase n from 100 to 1000 with a step size 100, and for each n we run GD and calculate z_n as before. Plot the curve z_n v.s. n and summarize your findings.

Estimation with prior knowledge. Let us increase n from 100 to 1000 with a step size 100, and for each n we run *projected GD* and calculate z_n as before. Note that in order to run projected GD, you need to implement the hard thresholding operator. Plot the curve z_n v.s. n and summarize your findings.