Homework 3

Due Feb 28 at 11 pm

1. (Augmented dataset) Ridge regression is equivalent to applying OLS on an expanded dataset that has additional examples. Describe these additional examples in detail. Intuitively, what effect do these additional examples have?

2. (Correlated features) Consider a regression problem where the response only depends on one feature, but we don't know it, so we incorporate an additional feature into the model that happens to be very correlated with the first feature. More specifically, let $y \in \mathbb{R}^n$ be defined by

$$y := \beta_{\text{true}} w_1 + z, \tag{1}$$

where $\beta_{\text{true}} \in \mathbb{R}$ is the true coefficient, $w_1 \in \mathbb{R}^n$ is the first feature vector, and $z \in \mathbb{R}^n$ is additive noise. The second feature vector is given by $w_2 \in \mathbb{R}^n$ and can be decomposed into

$$w_2 = \alpha w_1 + \sqrt{1-\alpha^2} w_\perp, \tag{2}$$

where $w_\perp$ is orthogonal to w_1. The vectors w_1, w_2, $w_\perp$ and z all have unit ℓ_2 norm. In addition, we assume

$$w_1^T z = 0.1, \tag{3}$$
$$w_\perp^T z = 0.1. \tag{4}$$

We fit a linear regression model to y using the feature matrix

$$X = \begin{bmatrix} w_1^T \\ w_2^T \end{bmatrix}. \tag{5}$$

   (a) What does the OLS estimator of the coefficients β_{OLS} equal to when $\alpha \to 1$? Explain what is happening.
   Hint: Use the fact that for any a, b, c, and d such that $ad \neq bc$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}. \tag{6}$$

   (b) What does the corresponding estimate of the response $y_{\text{OLS}} := X^T \beta_{\text{OLS}}$ equal to when $\alpha \to 1$? Is it collinear with the true feature w_1 when $\alpha \to 1$? Explain what is happening.

   (c) What does the ridge regression estimator of the coefficients β_{RR} equal to when $\alpha \to 1$ and the regularization parameter $\lambda > 0$ is fixed? Describe the difference with the OLS estimate.

   (d) What does the corresponding estimate of the response $y_{\text{RR}} := X^T \beta_{\text{RR}}$ equal to when $\alpha \to 1$? Is it collinear with the true feature w_1?

3. (Prior knowledge) Consider a linear regression problem where we have prior information indicating that the coefficients should be close to a certain value β_{prior}.

   (a) How can you incorporate this prior knowledge if you are using ridge regression? Write the corresponding optimization problem.

   (b) Assume that the data are generated according to a linear model $\tilde{y} := X^T \beta_{\text{true}} + \tilde{z}$, where $\beta_{true} \in \mathbb{R}^p$ and $X \in \mathbb{R}^{p \times n}$ are fixed and $\tilde{z}$ is an iid Gaussian random vector with zero mean and variance σ^2. Does the modification change the mean or the covariance matrix of the coefficient estimate with respect to the ridge-regression estimate? If so, report the new value.

   (c) How can you incorporate this prior knowledge if you are using gradient descent with early stopping? Write the corresponding update equation as a function of β_{prior}.

   (d) Assume that the data are generated according to the linear model described above. Does the modification change the mean or the covariance matrix of the coefficient estimate with respect to the gradient descent estimate initialized at the origin? If so, report the new value.

4. (Stochastic Gradient Descent) In class we saw how to use gradient descent to solve an optimization problem and applied it to ordinary least squares and ridge regression. However, gradient descent is often not used in practice. When the training data set is very large, computing the gradient of the objective function can take a long time as we need to go through the whole dataset for a single gradient step. When the objective function takes the form of an average of many values, such as in the case of linear regression

$$J(\beta) = \frac{1}{m} \sum_{i=1}^{m} \left(\beta^T x_i - y_i\right)^2$$

   stochastic gradient descent (SGD) can be very effective.

   (a) Here In SGD, rather than taking $-\nabla J(\beta)$ as our step direction (as in gradient descent), we take the gradient of objective function assuming there there is only data point i chosen uniformly at random from $\{1, \ldots, m\}$. Show that the SGD gradient is an unbiased estimator of the real gradient $-\nabla J(\beta)$.

   (b) Derive the SGD update rule for the linear regression

   (c) Use `np.linalg.lstsq` to find the least squares solution in `sgd.ipynb`.

   For our analysis of SGD, we assumed that we pick a datapoint at random at each step. The However, typically in practice, we we go through the whole dataset in a random order instead. One pass through the whole dataset is called an epoch. We recommend that your implementation for the next problem follow the strategy of going through all data points in a random order (ie shuffle dataset before every epoch) rather than picking a point uniformly at random for each step.

   (d) Use SGD to find β^* that minimizes the linear regression objective in `sgd.ipynb`. Note that our gradient estimate in each step is quite noisy (even though in expectation

it matches the full graident). Since our steps are noisy, it is important to pick the learning rate carefully. For this question, try a few fixed step sizes (at least try $\eta_t \in \{0.05, .005, 0.005\}$ and step sizes that decrease with the step number according to the following schedules: $\eta_t = \frac{C}{t}$ and $\eta_t = \frac{C}{\sqrt{t}}$, $C \leq 1$. Please include atleast $C = 0.01$ in your submissions. For each step size rule, plot the value of train loss and validation loss (or the log of it if that is more clear) as a function of epoch (or step number, if you prefer) for each of the approaches to step size. How do the results compare?

(e) Compute the error on the test set for the best β value obtained using SGD and `np.linalg.lstsq` in part (c).

The notebooks on ridge regression and gradient descent maybe useful for this assignment.