Homework Set 2, CPSC 6430/4430

LastName, FirstName

Due 03/06/2024, 11:59PM EST

Gradient Descent

We reconsider Linear regression:

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \tag{1}$$

1. Please randomly generate $\mathbf{A} \in \mathbb{R}^{10000\times 5000}$, compare the time consumption of closed solution given by $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}$ and gradient descent method $\mathbf{x}^{k+1} = \mathbf{x}^k - \lambda\mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{y})$ by constant stepsize $\lambda = 1/\|\mathbf{A}\|_F^2$ assuming 1000 iterations, which one is faster?

2. Please randomly generate $\mathbf{A} \in \mathbb{R}^{100\times 50}$, compare the time consumption of closed solution given by $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}$ and gradient descent method $\mathbf{x}^{k+1} = \mathbf{x}^k - \lambda\mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{y})$ by constant stepsize $\lambda = 1/\|\mathbf{A}\|_F^2$ assuming 1000 iterations, which one is faster?

(For any matrix $\mathbf{Z}$, we define $\|\mathbf{Z}\|_F^2 := \sum_i \sum_j \mathbf{Z}_{ij}^2$.)

GD for Ridge Regression

For Ridge Regression:

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \gamma \|\mathbf{x}\|_2^2, \tag{2}$$

1. What is the derivative with respect to $\mathbf{x}$?
2. What is the closed solution of ridge regression?
3. Assume $\mathbf{A} \in \mathbb{R}^{100\times 50}$, and stepsize $\lambda = 0.5/(\gamma + \sigma_1(\mathbf{A}\mathbf{A}^T))$, where $\sigma_1(\mathbf{Z})$ denotes the largest singular value of $\mathbf{Z}$. Please change λ from $\{0.01, 0.1, 1, 10, 100\}$ and plot the objective changes with update via gradient descent method (100 iterations), what do you find?

Singular Value Decomposition

To save barbara.bmp image, usually we need a 512*512 matrix denoted by $\mathbf{X}$, which is large. We seek to store the image as the product of three matrices $\bar{\mathbf{A}}, \bar{\mathbf{Z}}, \bar{\mathbf{C}}$, such that $\bar{\mathbf{A}} * \bar{\mathbf{Z}} * \bar{\mathbf{C}}^T \approx \mathbf{X}$. To obtain the best $\bar{\mathbf{A}}, \bar{\mathbf{Z}}, \bar{\mathbf{C}}$, we can do SVD for $\mathbf{X}$ such that $\mathbf{A} * \mathbf{Z} * \mathbf{C}^T = \mathbf{X}$, so the best $\bar{\mathbf{A}}, \bar{\mathbf{Z}}, \bar{\mathbf{C}}$ can be obtained by $\bar{\mathbf{A}} = \mathbf{A}(:, 1 : r), \bar{\mathbf{Z}} = \mathbf{Z}(1 : r, 1 : r), \bar{\mathbf{C}} = \mathbf{C}(:, 1 : r)$. Now please choose different options of $r \in \{1, 5, 10, 20, 50, 100, 200, 300, 500, 512\}$ and plot the recovered $\bar{\mathbf{X}} = \bar{\mathbf{A}} * \bar{\mathbf{Z}} * \bar{\mathbf{C}}^T$. Choose the optimal r to you and determine the ratio of new storage over the original image $\mathbf{X}$.

Principal Components from Dataset

Please determine the top 10 components of olivettifaces.mat dataset via PCA and plot them.

Optimization for Lasso

In class, we mentioned Lasso model:

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_1, \tag{3}$$

where we define $\|\mathbf{z}\|_2^2 = \sum_i \mathbf{z}_i^2$, $\|\mathbf{x}\|_1 = \sum_i |\mathbf{x}_i|$. Different from linear regression or ridge regression, where we can obtain the optimal solution by taking the derivative with respect to $\mathbf{x}$ and set to 0, the main drawback of Lasso model is $\|\cdot\|_1$ is not differentiable. To address the issue, we turn to a different optimization framework which is summarized as below:

Algorithm 1 An algorithm to solve Eq. (3)

$\mathbf{x} \leftarrow 0$
$t \leftarrow 1/\sigma_1(\mathbf{A}\mathbf{A}^T)$
$k \leftarrow 1$
$K \leftarrow 100$
while $k \leq K$ **do**
    $\mathbf{z} = \mathbf{x} - t\mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{y})$
    $***for\ each\ element\ in\ \mathbf{z}\ and\ \mathbf{x}***$
    $\mathbf{x}_i = sign(\mathbf{z}_i) * max\{|\mathbf{z}_i| - t\lambda, 0\}$
end while

1. Please implement the algorithm and plot the objective changes with update (for $\lambda = 1$ and $\lambda = 2$, respectively) where x-axis is the iteration and y-axis is the objective in Eq. (3).

2. Please compare the output from Algorithm 1 with Lasso in sklearn and check whether they are close. You can randomly generate $\mathbf{A} \in \mathbb{R}^{100\times 50}, \mathbf{y} \in \mathbb{R}^{100}$.