Homework 3

6.s955 Applied Numerical Analysis

Pitipat Wongsittikan

October 23rd, 2023

Problem 1

a. For simplification, we will work on the minimization of $||w||_2^2$ instead of $||w||_2$.
Note: Since X's columns are independent, X^TX is invertible, but XX^T **is not invertible**

$$\begin{aligned} \min_{w} \quad & ||w||_2^2 \\ \text{s.t.} \quad & X^T w = y \end{aligned} \tag{1}$$

b. Apply Lagrange multipliers, take their derivatives, and set them equal to 0 as follows,

$$\mathcal{L}(w, \lambda) = ||w||_2^2 + \lambda \cdot (X^T w - y) \tag{2}$$

$$\frac{\partial \mathcal{L}}{\partial w} = 2w + X\lambda \tag{3}$$

$$0 = 2w + X\lambda \tag{4}$$

$$w = -\frac{1}{2} X\lambda \tag{5}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = X^T w - y \tag{6}$$

$$0 = X^T w - y \tag{7}$$

$$X^T w = y \tag{8}$$

Substitute eq.(5) into eq.(8).

$$X^T(-\frac{1}{2} X\lambda) = y \tag{9}$$

$$\lambda = -2(X^TX)^{-1} y \tag{10}$$

Then, substitute eq.(10) into eq.(5).

$$w^* = -\frac{1}{2}X(-2(X^TX)^{-1}y) \tag{11}$$
$$= X(X^TX)^{-1}y \tag{12}$$

c. Since $w^* = X(X^TX)^{-1}y$, we can show that w^* and w_0 are perpendicular as follows,

$$w^* \cdot w_0 = w^{*T}w_0 \tag{13}$$
$$= y^T(X^TX)^{-T}X^Tw_0 \tag{14}$$

Since $w_0 \in ker(X^T)$, therefore, $X^Tw_0 = 0$. We will have,

$$w^* \cdot w_0 = y^T(X^TX)^{-T}[X^Tw_0] \tag{15}$$
$$= y^T(X^TX)^{-T}[0] \tag{16}$$
$$= 0 \tag{17}$$

Since they are perpendicular, $||w||_2 = ||w^* + w_0||_2 = ||w^*||_2 + ||w_0||_2$. Therefore, to minimize the term, w_0 must be $\mathbf{0}$. Hence, $||w||_2$ minimizes to $||w^*||_2$.

d. Substitute $X = QR$ into w^*.

$$w^* = X(X^TX)^{-1}y \tag{18}$$
$$= QR(R^TQ^TQR)^{-1}y \tag{19}$$
$$= QR(R^TR)^{-1}y \tag{20}$$
$$= QRR^{-1}R^{-T}y \tag{21}$$
$$= QR^{-T}y \tag{22}$$

Since Q is a d by n and R is a n by n upper triangular matrix, we can inverse R taking $O(n^2)$ time. Then multiply all of the terms from right to left, will take,$O(n^2)$ to compute w^*.

Problem 2

a. We apply Libschitz-continuous theorem on the gradient of $f(w)$ to find the bounds of the gradient steps.

$$\nabla f(w) = X(X^Tw - y) \tag{23}$$

A function $f(x)$ will be L-Lipschitz if there exists a constant L such that

$$||f(x_1) - f(x_2)||_2 \leq L||(x_1 - x_2)||_2 \tag{24}$$

We use $\nabla f(w)$ in place of $f(x)$.

$$\begin{aligned}
||\nabla f(w_1) - \nabla f(w_2)||_2 &= ||X(X^T w_1 - y) - X(X^T w_2 - y)||_2 && (25)\\
&\leq ||XX^T||_2||(w_1 - w_2)||_2 && (26)
\end{aligned}$$

Therefore, the gradient of the function f is $||XX^T||$-Libschitz. If we use a step size that is equal to $\frac{1}{||XX^T||_2}$, we can see that it's less than the gradient of f, which will guarantee convergence of the gradient descent of this problem.

b. If we start with $w_0 = \mathbf{0}$, we know that $\nabla f = X(X^T w - y)$. Therefore, we will have $\nabla f(w_0) = -Xy$ and $w_1 = \alpha Xy$ for a step size α. Now, we repeat the same steps to find w_k.

$$\begin{aligned}
& w_0 = \mathbf{0} \text{ and } \nabla f = 0 && (27)\\
& w_1 = \alpha Xy \text{ and } \nabla f = X(X^T(Xy) - y) && (28)\\
& w_2 = \alpha Xy - \alpha^2 X(X^T Xy - y) && (29)\\
& w_3 = X(\alpha y - \alpha^2(X^T Xy - y)). && (30)\\
& \quad . && (31)\\
& \quad . && (32)\\
& && (33)
\end{aligned}$$

We can see that, except w_0, w_k has the term X in front of it. Therefore, we can write w_k in terms of $\exists b : Xb$, which indicate that $w_k \in Im(X)$.

c. From part b., we know that we can write $w_k = Xb$ for some vector b and if we select an appropriate value of step size t, which we also have shown in part 1 that it exists, it will converge to a solution. Therefore, if we iterate it enough, w_k will converge to w^*, which is also can be written in term of Xb for some b.
Moreover, $w^* = Xb$ will also be a solution of $X^T w = y$. Hence,

$$X^T(Xb) = y \tag{34}$$

Since $X^T X$ is invertible,

$$b = (X^T X)^{-1} y \tag{35}$$

Since $w^* = Xb$, we will have $w^* = X(X^T X)^{-1} y$ which is also the solution to the least norm to the solution of $X^T w = y$ shown previously.

Problem 3

a. From the hint, we can write the dot product between $k_{x_i}(\cdot)$ and $k_{x_j}(\cdot)$ as $\sum_{k=1}^{\infty} \exp(-\frac{||x_i-p_k||_2^2+||x_j-p_k||_2^2}{\sigma^2})$. Where p_k represents a vector that lies on a hyper line from ∞ to ∞.

b. To do this, first we consider the term inside exp as follows,

$$
\begin{aligned}
||x_i - p_k||_2^2 + ||x_j - p_k||_2^2 &= ||p_k - x_i||_2^2 + ||p_k - x_j||_2^2 && (36)\\
&= 2||p_k||_2^2 - 2p_k^T(x_i + x_j) + ||x_i||_2^2 + ||x_j||_2^2 && (37)\\
&= 2||p_k - \frac{1}{2}(x_i + x_j)||_2^2 + ||x_i||_2^2 + ||x_j||_2^2 - \frac{1}{2}||x_i + x_j||_2^2 && (38)\\
&= 2||p_k - \frac{1}{2}(x_i + x_j)||_2^2 + \frac{1}{2}||x_i - x_j||_2^2 && (39)
\end{aligned}
$$

Therefore, the terms become,

$$\sum_{k=1}^{\infty} \exp(-2\frac{||p_k - \frac{1}{2}(x_i + x_j)||_2^2 + \frac{1}{2}||x_i - x_j||_2^2}{\sigma^2}) \tag{40}$$

Which can be converted into an integral form,

$$\int_{-\infty}^{\infty} \exp(-2\frac{||p - \frac{1}{2}(x_i + x_j)||_2^2 + \frac{1}{2}||x_i - x_j||_2^2}{\sigma^2})dp \tag{41}$$

We know that, for gaussian integral,

$$\int_{-\infty}^{\infty} \exp(-a(x+b)^2) = \sqrt{\frac{\pi}{a}} \tag{42}$$

Therefore,

$$
\begin{aligned}
&\int_{-\infty}^{\infty} \exp(-2\frac{||p - \frac{1}{2}(x_i + x_j)||_2^2 + \frac{1}{2}||x_i - x_j||_2^2}{\sigma^2})dp\\
&\qquad = \exp(\frac{1}{2\sigma^2}||x_i - x_j||_2^2) \int_{-\infty}^{\infty} \exp(-2\frac{||p - \frac{1}{2}(x_i + x_j)||_2^2}{\sigma^2})\\
&\qquad\qquad = \exp(\frac{||x_i - x_j||_2^2}{2\sigma^2})\sqrt{\frac{\sigma^2\pi}{2}} && (43)
\end{aligned}
$$

(Sorry for a weird indent here, I have no idea how to deal with long equation in Latex (yet!).

Problem 4

a. Please see the attached hw3.py file.

b. First, we define C as a matrix with c's as its rows. For each iteration, we have a subset of K and y represent with $\hat{K}$ and $\hat{y}$ respectively. Therefore, we can write $f(C)$ as $f(C) = \frac{1}{2}||\hat{K}C - \hat{y}||^2$. Then, take derivative of $f(C)$ with respect to C.

$$\frac{\partial f}{\partial C} = \hat{K}^T(\hat{K}C - \hat{y}) \tag{44}$$

Therefore, using stochastic gradient descent, we can find C_{k+1} by

$$\begin{aligned} C_{k+1} &= C_k - t \cdot \frac{\partial f}{\partial C} && (45) \\ &= C_k - t \cdot \hat{K}^T(\hat{K}C - \hat{y}) && (46) \end{aligned}$$

$\hat{K}$ is a L by n matrix. C is a n by 10 matrix, and $\hat{y}$ is a L by 10 matrix. Therefore, computing C_{k+1} will have at least $O(10Ln + 10L + 10Ln + 10L + 10L)$, which is roughly $O(20Ln)$. Note: the order of time complexity is from multiplying KC (L by n times n by 10), adding 10 by L matrices, multiplying K^T, multiplying constant t to a 10 by L matrix, and add that to the original C, respectively. together.

c. Please see the attached hw3.py file.